HP WebInspect Tools

for WebInspect and WebInspect Enterprise

Software Version: 10.20

Tools Guide for WebInspect Products

Document Release Date: April 2014
Software Release Date: April 2014

Legal Notices

Copyright Notice

Restricted Rights Legend

Disclaimer of Warranty

Trademark Acknowledgements

Documentation Updates

This guide's title page contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates, or to verify that you are using the most recent edition of a document, visit the following URL:

http://h20230.www2.hp.com/selfsolve/manuals

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

You can open a support case for HP Fortify Security products via email, online, or by telephone. These options are designed to provide easier access and improved customer satisfaction.

Email (Preferred Method)

Send an email to fortifytechsupport@hp.com describing your issue. Please include the product name so we can help you faster.

Online (Fortify Support Portal)

Access your account at the Fortify Support Portal at **https://support.fortify.com**.

If you do not have an account, you forgot your user name or password, or you need any assistance regarding your account, please contact us at fortifytechsupport@hp.com or (650) 735-2215.

Telephone

Call our automated processing service at (650) 735-2215. Please clearly provide your name, telephone number, the name of the product, and a brief description of the issue.

Contents

1 Welcome to WebInspect Tools

About WebInspect Tools

WebInspect Tools is a robust set of diagnostic and penetration testing tools and configuration utilities packaged with WebInspect and WebInspect Enterprise.

The tools provided in WebInspect Enterprise are a subset of the tools provided in WebInspect. The chapters in this guide that describe tools that are provided in WebInspect but not in WebInspect Enterprise have titles that end with "(WebInspect Only)."

WebInspect includes an HP Support tool, described in Chapter 6, HP Support Tool (WebInspect Only), that provides a quick and simple method for uploading files that may help HP Support personnel analyze and resolve any problems you may encounter while using WebInspect.

Note: When using tools that incorporate a proxy you may encounter servers that do not ask for a client certificate even though a client certificate is required. To accommodate this situation you must edit the `SPI.Net.Proxy.Config` file.

2 Audit Inputs Editor

About the Audit Inputs Editor Tool

This tool allows you to create or edit inputs to the audit engines and to a distinct set of checks.

There are two ways to access the Audit Inputs Editor:

- From the Policy Manager (using the Policy Manager's **Tools** menu). Use this method to create or modify an inputs file (<filename>.inputs). You can then specify this file when modifying scan settings.

  To modify an inputs file, click the Open icon on the Audit Inputs Editor's toolbar or select **File** → **Open**.

- From the Default or Current Settings (by clicking the Audit Inputs Editor button on the Attack Exclusions settings). Using this method, you can modify the Default settings file directly, but you cannot create a separate inputs file.

If you access the Audit Inputs Editor from Default or Current Settings, the check inputs you create or modify become part of the settings file.

However, if you accessed the Audit Inputs Editor from the Policy Manager, you must import into WebInspect the saved file containing your check input modifications. To do so:

1. On the WebInspect menu bar, click **Edit** → **Default Settings**.
2. Under Audit Settings, select **Attack Exclusions**.
3. Click **Import Audit Inputs**.
4. Select the file you created and click **Open**.

When accessed through the *Current Settings* or the *Default Settings* window, Attack Exclusions panel, the Audit Inputs Editor does not contain a menu bar or toolbar.

Engine Inputs

To create or modify inputs to audit engines.

1. Click the **Engine Inputs** tab.
2. Click the drop-down arrow.
   a. To apply your modifications to all audit engines, select **<Default>**. The Default parameters are extracted from the WebInspect default Audit Settings - Attack Exclusions.
   b. To modify inputs for a specific audit engine, select one from the list.
3. Select an engine input.
4. If you selected one of the following:
   - Excluded Query Parameters
   - Excluded Post Parameters

- Excluded Cookies
- Excluded Headers
- Root Directories

a. To add an item to the list, click **Add**.

b. To edit an item, select an item and click **Edit**.

c. To delete an item, select the item and click **Remove**.

d. If you selected a specific engine (rather than Defaults), select one of the following options:

— **Merge with defaults** - The parameters you specified are added to the Defaults list, which apply to all engines.

— **Replace defaults** - The engine will use the parameters you specified instead of those in the Defaults list.

Note: If you specify a Root Directory, then the engine will attack the object in the directory you specify, rather than the actual root. For example, if an engine normally attacks filename.txt in the default root directory rootdir (/rootdir/filename.txt), then if you specify a root directory of /foobar/, the engine will attack /foobar/filename.txt.

5. If you selected one of the following:
   - Header Audit Rules
   - Cookie Audit Rules

   a. Unselect the **Use value from defaults** check box.

   b. Select an option from the drop-down list.
6. Click **OK** (if you launched the Audit Inputs Editor from WebInspect's Default or Current Settings) or click the **File** menu and select **Save** or **Save As** (if you launched the Audit Inputs Editor from the Policy Manager).

Check Inputs

Certain checks require inputs that accommodate the specific design of the target website. WebInspect conducts these checks using default values, which you may need to change.

To create or modify inputs for specific checks.

1. Click the **Check Inputs** tab.
2. Select a check (see list below).
3. Enter the requested input values.
4. Click **OK** (if you launched the Audit Inputs Editor from Default or Current Settings) or click **File** → **Save** (or **Save As**, if you launched the Audit Inputs Editor from the Policy Manager).

4719: IIS Mapping

Microsoft IIS extension handlers historically have been the source of many vulnerabilities. This check probes for each known IIS extension, and flags a vulnerability for each extension/handler that is found to be enabled. However, in certain cases, an extension handler may be legitimately enabled and used by the target website.

Required Input: One or more extensions that identify the handlers that are legitimately enabled and which should be excluded. Valid input is printer, idc, idq, ida, htr, htw, stm, shtm, and shtml.

4721: Admin Section Must Require Authentication

Any area of the website or web application that contains sensitive information or access to privileged functionality (such as remote site administration) requires authentication before allowing access. This check attempts to access a sensitive directory that should require authentication. The default check input is /admin.

Required Input: The directory (relative to the root) containing administrative or sensitive data.

4722: Logins Sent Over Unencrypted Connection

Any area of a web application that possibly contains sensitive information or access to privileged functionality (such as remote site administration functionality) should utilize SSL or another form of encryption to prevent login information from being sniffed or otherwise intercepted or stolen.

Required Input: Login forms. The name of file containing login form.

4723: Logins Sent Over Query

Information in query strings is directly visible to the end user via the browser interface, which can cause security issues. Recommendations include performing server-side input validation to ensure data received from the client matches expectations.

Required Input: Login forms. The name of file containing login form.

4724: Password Field Masked

Basic web application security measures include “masking” all passwords entered by a user when logging on to a web application. Normally, each character in a password entered by a user is instead represented with an asterisk. Recommendations include requiring all password fields in your web application be masked to prevent other users from seeing this information.

Required Input: The name attribute of input controls containing a password.

4726: Secure Section Only Accessible Via SSL

Any area of the website or web application that contains sensitive information or access to privileged functionality (such as remote site administration) requires that the pages under the secure section of the site are only accessible via SSL.

Required Input: The name of the secure directory, relative to the root. The default is /secure.

4728: Persistent Cookies

Persistent cookies are stored on the browser’s hard drive. This can cause security and privacy issues depending on the information stored in the cookie and how it is accessed. This check calculates how many seconds until a received cookie is set to expire. If the expiration date/time is less than the specified number of seconds (default: 600), the check considers the cookie’s life span to be excessive, increasing the chances of session ID recovery and session hijacking.

Required Input: The lifetime allowed for cookies (in seconds).

4729: User supplied data without POST

An area of the web application that possibly contains sensitive information or access to privileged functionality (such as remote site administration functionality) uses query strings to pass information between pages. Information in query strings is directly visible to the end user via the browser interface, which can cause security issues. The input value for this check is a space-separated list of regular expressions that are used to identify sensitive URL parameter names when used in GET queries. Generally, information such as passwords, social security numbers, etc., should not be sent as parameters to GET queries, since the GET query (and thus the sensitive information) can persist in web server and proxy logs and the web browser's history. You will need to adjust the regular expressions accordingly to specify the parameter names your application typically uses to denote sensitive information.

Required Input: Sensitive parameter (a regular expression). An example is:

p|P]ass(word)? [u|U]ser_?([N|n]ame)? [s|S][s|S][n|N]

4731: Script Directory Check

A directory containing an object referenced in a post request or query string should not have a name that could easily be guessed by an attacker. The primary danger from an attacker discovering this directory would arise from the information he could gather from its contents, such as what language was used to code the web application. This check is used to determine if a dynamic form action points to a file/URL that is in a directory whose name is included in the list.

Required Input: Names of directories containing scripts.

4732: Script File Extension Disclosure

Any area of the website or web application that contains sensitive information or access to privileged functionality (such as remote site administration) requires the file extension of all scripts to be checked as it may lead to information disclosure related to the technology used by the application. The use of certain CGI-related file extensions can indicate certain types of technology in use, which results in a mild information disclosure. The default list of check input values is generally applicable, but some sites may legitimately use a certain technology (such as Perl) and this check may incorrectly elicit false-positive issues in flagging all Perl extensions (.pl). In such cases, you should remove the legitimate extensions from the list.

Required Input: File extensions of scripts used in the web application (such as cgi, pl, and py).

5151: Arbitrary Remote File Include

This check attempts to discover if the web application can fetch and incorporate data from arbitrary URLs supplied by an attacker.

This is the most complex check to configure, because its extreme flexibility enables it to work in many environments and topologies. Basically, the check injects URL values into application parameters, attempting to force the application to make an HTTP request to the supplied URL. This activity looks for "remote file inclusion" vulnerabilities caused by the application attempting to remotely retrieve the specified file/URL and include the response into the application's processing. In certain extreme circumstances found in PHP environments, the application will remotely retrieve the file and execute any PHP script contained therein, making the activity capable of arbitrary code/script execution.

Check 5151 can operate in two modes: static and server (controlled by the "Audit Mode" parameter).

Static Mode

You specify the target external URL as the **Static Mode Target URL**, and a corresponding regular expression signature as the **Static Mode Signature**. If you want to use external targets, then you should use static mode. By default, the check uses static mode and the test URL of "http://15.216.12.12/serverinclude.html?" which is a special page hosted on an HP web server located on the public Internet at IP address 15.216.12.12. The signature contains a specific value that is returned by the indicated test URL. If you do not want to use the HP web server (particularly if the target server cannot access the Internet), then you should adjust the test URL (and corresponding signature) to a URL hosted by a server. When configuring static mode:

- Specify a full, absolute URL (i.e., it should begin with "http://").
- For best results, use non-SSL URLs (although SSL URLs are allowed).
- Include a question mark (?) at the end of your URL to ensure the URL is not affected if the application appends additional data to the end of the URL.

Server Mode

In this mode, WebInspect runs its own web server and attempts to get the target/scanned server to connect to the WebInspect scanning system. The added benefit of Server mode is that it can detect "blind" remote file inclusion vulnerabilities, resulting in potentially fewer false negatives. To use Server mode, the check conceptually needs three pieces of information:

Server Mode Target IP -- The IP address the server/target should use to access the host (particularly if the scanning system's network IP is different than what the server would need to access, due to a firewall or a multi-homed scanning system). The default value is empty/blank, meaning that it uses the same IP address ultimately used or determined by the Server Mode Server IP.

- Server Mode Server Port -- The port number to run the listening web server on. Using a specific port may be necessary due to network/access restrictions. The default value is 8181. If you leave this value blank, then the Remote File Include engine will dynamically choose a port between 25000 and 25100.
- Server Mode Server IP -- The local IP address of the scanning system to bind the web server on, if the system is multi-homed and/or you do not want to bind the web server listening on the first local IP address. The default value is "0.0.0.0", which instructs WebInspect to use the first available IP address on the system.

Although the default values fit most configurations, certain circumstances require specific modification.

- If your system has multiple IP addresses (due to multiple network adapters), then you may need to specify the explicit IP address to bind to (i.e., the one that is most appropriate for receiving requests from the system you are scanning). You can determine the list of your system's IP addresses by running "biconvex" from a Windows command prompt.
- If you are running multiple scans from the same scanning system using server mode, then you should leave the **Server Mode Server Port** value blank, causing WebInspect to dynamically pick the port. This is because two scans cannot run two separate web servers listening on the same port. One specific port can only be used by one scan at a time.
- If your system is behind a firewall and you are using port-forwarding to receive the incoming HTTP requests, or you are on a network that uses NAT, then the IP address used by the server to access your system will be different from the IP address actually assigned to your system. In this case, you will need to specify the IP address the target server should use for the **Server Mode Target IP**.

Required Inputs: Static mode target URL, Audit mode (static or server), Server mode server IP, Server mode Server port, Server mode target IP, static mode signature (a regular expression)

5546: Privacy Policy Not Present

This check is associated with WebInspect's compliance policies. Many legislative initiatives require organizations to place a publicly accessible document within their web application that defines their information privacy policy. If WebInspect does not find the specified file, it creates a vulnerability in the Best Practices category.

Required Inputs: The relative directory and file name of the privacy policy.

5649: Cross-Site Scripting

Abnormal String List for XSS Engine

This check input defines the list of potentially malicious characters that are often used in Cross-Site Scripting attack strings. The WebInspect Cross-Site Scripting engine will use these characters to test any potential encoding being applied to these characters. The test results will help guide the Cross-Site Scripting engine in forming the attack strings most likely to lead to a successful injection.

Do Partial Encoding of XSS Attack String

Insecurely designed anti-XSS, blacklist filters can sometimes be bypassed by applying certain encoding transformations to the attack string. The type and degree of encoding can affect the success of an attack. This check input, when set to true, extends the XSS engine behavior to send partially encoded attacks (for example, %3cscript%3ealert(1)%3c%2fscript%3e) in addition to fully encoded attacks (%3e%3c%73%43%72%49%70%54%3e%61%6c%65%72%74%28%31%29%3c%2f%73%43%72%49%70%54%3e).

5650: Cross-Site Scripting (User Interaction)

Abnormal String List for XSS Engine

This check input defines the list of potentially malicious characters that are often used in Cross-Site Scripting attack strings. The WebInspect Cross-Site Scripting engine will use these characters to test any potential encoding being applied to these characters. The test results will help guide the Cross-Site Scripting engine in forming the attack strings most likely to lead to a successful injection.

Do Partial Encoding of XSS Attack String

Insecurely designed anti-XSS, blacklist filters can sometimes be bypassed by applying certain encoding transformations to the attack string. The type and degree of encoding can affect the success of an attack. This check input, when set to true, extends the XSS engine behavior to send partially encoded attacks (for example, %3cscript%3ealert(1)%3c%2fscript%3e) in addition to fully encoded attacks (%3e%3c%73%43%72%49%70%54%3e%61%6c%65%72%74%28%31%29%3c%2f%73%43%72%49%70%54%3e).

10044: HTML Tag Injection

Abnormal String List for XSS Engine

This check input defines the list of potentially malicious characters that are frequently used in Cross-Site Scripting attack strings. The WebInspect Cross-Site Scripting engine will use this list to detect any potential encoding transformations being applied to these characters by the target application. The test results will help guide the Cross-Site Scripting engine to form the attack strings most likely to discover an injection flaw.

10167: Password in Query or Cookie Data

Transmitting password information in a query string or cookie value makes it easy for an attacker to see and tamper with login values. Recommendations include ensuring that login information is sent with a POST request over an encrypted connection and that sensitive account information is stored on the server.

Required Inputs:

- Password Field Names - List of Query or Cookie parameter names containing a password.
- Possible Username List - List of Query or Cookie parameter names containing a username.

10183: Allowed Top-Level Domain

Certain organizations (especially branches of the U.S. federal government) must use a restricted set of DNS top-level domain names (TLDs), such as .gov, .mil, or .fed. This check ensures that all allowed hosts encountered during the scan use one of the specified TLDs. Most public corporations arbitrarily use any TLD they desire (.com, .net, .org, etc.); those corporations should either disable this check (preferable) or change the default values to include .com, .net, and .org (and/or any other appropriate TLDs).

Required Inputs: All allowed top-level domains.

10274: Proxy CONNECT Access

Some proxy servers accept the CONNECT method to make an HTTP connection to another server. Usually, this method should be restricted to internal use only. If it is not restricted, your server can be used by an attacker on the Internet to disguise himself as your own server. Thus, any attack will appear to come from your server. This type of vulnerability is usually caused by not properly configuring the proxy server. Attackers can masquerade as your proxy server when conducting other attacks. Attackers may be able to access internal machines through the CONNECT proxy. This attack can also be used to enumerate your local network.

This check attempts to treat the target server as a proxy server for SSL requests. The check issues a CONNECT request to the target server, which essentially asks the server to make a connection to another external site. You can control which external site is used via the input values for this check. By default, the value “https://www.google.com/” is used, causing the server to make an external request to the host www.google.com on port 443. You may wish to modify this value to point to a more appropriate internal host. If so:

Use a server that has SSL enabled on the standard SSL port 443, if possible. Some proxies refuse connections to ports other than 443 due to explicit configuration.

Use the https:// URL format.

If you need to specify a port other than 443, use the normal URL format to specify a port after the host name (e.g., https://example.com:8443/).

Only the host name and port number are used; the remainder of the URL is ignored.

10275: Proxy GET Access

This check is virtually identical to check 10274, except it issues a proxy-qualified GET request to the target server instead of a CONNECT request. There are many servers that are willing to take a proxy-qualified GET request and treat it as a normal GET request (ignoring the proxy-specific aspects of the request), so it is necessary for the check to evaluate the response content to ensure the response is truly from the external server and not a normal response from the target. That is why check 10275 has two check inputs: one for specifying the external target host, and one for specifying a regular expression to match against the response content. By default the check attempts to access "http://www.google.com/" and looks for the phrase "Google Search" in the response. You will need to adjust the check input values if you need to use a different external host or an internal host. You can change the external target simply by adjusting the target check input value, and then specifying a unique value from the target page as the check input regex value.

- The URL target must begin with http:// or https://. For best results, use http://.
- If you need to specify a specific port other than 80/443, use the normal URL format to specify a port after the host name (such as http://example.com:8080).
- Unlike check 10274, the target URL you specify for 10275 is used in its entirety; if you specify a specific page/URL, then that specific page/URL will be requested.
- Try to select a unique value/phrase from the target URL to use as the response regex value, one that is not likely to appear elsewhere on the target scanned site; using the value in the <title> tags usually is sufficient (you can also include the "<title>" tags in the regex value itself).
- Remember to properly escape any regex-specific metacharacters (periods, parentheses, etc.).
- The check does not follow redirects (HTTP 302 responses), so you will need to specify an explicit final URL destination.

Required Inputs: Proxy GET target and Proxy GET target response (regular expression).

10280: Price-Related Form Fields

Forms containing price-related field names could harbor price manipulation vulnerabilities that would allow the attacker to change the price of the product.

Required Inputs: Names of price-related fields.

10287: Local File Include

Mode

The Mode parameter relates to the platform assumptions made by the engine. The default Mode value, **Auto**, causes the engine to look for both "c:\windows\win.ini", "c:\boot.ini" (Windows) and "/etc/passwd" (Unix) files and to use both Windows and Unix parent directory references accordingly. If the engine gets a visual response that explicitly indicates the underlying platform (Windows vs. Unix), it will automatically switch to using only the values for the appropriate target platform for the remainder of the auditing for that application parameter value. If you already know what the underlying platform is before you scan (that is, Windows vs. Unix), you can change the mode to **Windows** or **Unix**, which can save scan time since it reduces the number of values that need to be sent. At this time the engine does not support platforms that do not use a Windows ("\") or Unix ("/") path separator.

User-Specified File

If you want to use a specific target file, specify it here. There are occasions when the default file name values ("c:\windows\win.ini", "c:\boot.ini" and "/etc/passwd") may not work in your environment. For example, your web application can be hosted on a Windows drive other than 'C:', or your web application could be operating out of a Unix chroot environment. In both cases, parent directory references will not be able to locate the specified target files even if a vulnerability does exist. For this situation you should either use an existing file that is in the root directory of the same drive/chroot of the web application, or explicitly create a text file in the root directory of the drive/chroot used by your web application and place a unique value inside the text file. Then you inform the LFI engine to look for your specific file by setting the **UserOnly** mode option, and specifying the absolute path to your target file in the **User Specified File** check input. You will also need to specify a corresponding **User Specified File Regex** check input value; the regex value should uniquely identify/match the contents of your specified file while not matching any content typically found on the scanned website. You can also select the **UserAndAuto** mode, which would let you specify a file and still use the default "c:\windows\win.ini" and "/etc/passwd" values.

User-Specified File Regex

If you use a specific target file, enter a regular expression that matches the contents of that target file.

Audit Disposition

The Audit Disposition parameter default value **Adaptive** treats web application parameters in one of two ways: parameters with existing values that resemble file names receive significant (aggressive) scrutiny, while all other parameters receive basic scrutiny. The premise is that if the parameter has a value that resembles a filename, then there is a higher likelihood that the value is used in a file system operation; because of that higher likelihood, it makes sense for the engine to try more variations (particularly minor variations) to ensure that is not the case. However, trying additional minor variations can extend scan time, because it results in more attacks to be sent. That is why the **Adaptive** disposition tries to determine when it seems appropriate to spend the extra effort in auditing a particular parameter. However, if you desire the utmost level of scrutiny for all parameters, change the Audit Disposition value to **Aggressive**.

Suspicious Parameters

All the QUERY and POST parameters whose names match the patterns specified in this check input will be considered to have a higher likelihood of being vulnerable to Local File Inclusion (LFI) attacks and hence will be subjected to exhaustive testing.

Score on Error Message

By default when a Local File Inclusion (LFI) attack results in an HTTP error status code, the LFI engine will not report the finding. Occasionally, however, an error message could be masking a real vulnerability. To ensure that such potential scenarios are reported, set this check input value to true.

10551: Possible Username or Password Disclosure

Exposing login information on publicly accessible sections of a web Application could allow an attacker to access sensitive applications and information on a site, or to perform functions according to the privilege level of the login information. Gaining information critical to the success of escalated attacks would also be a likely impact of exploitation. Recommendations include purging the information from publicly accessible content, if possible, or otherwise ensuring proper access controls are in place.

Required Inputs:

- Password field names - Names of client-side script variables containing a password.
- Possible Username List - Names of client-side script variables containing a username.

10940: Persistent Cross-Site Scripting (XSS)

Abnormal String List for XSS Engine

This check input defines the list of potentially malicious characters that are often used in Cross-Site Scripting attack strings. The WebInspect Cross-Site Scripting engine will use these characters to test any potential encoding being applied to these characters. The test results will help guide the Cross-Site Scripting engine in forming the attack strings most likely to lead to a successful injection.

10962: Blind SQL Injection (confirmed)

One of the several techniques used by the SQL Injection engine is to force the application to execute a time intensive query. By analyzing the time taken by the application to return a response, the engine can detect if the WebInspect-supplied query was successfully injected.

Databases To Exclude

WebInspect is capable of detecting SQL injection vulnerabilities in applications using DB2, Microsoft SQL Server, MySQL, Oracle, PostgreSQL, and Informix. If the database server used by the application is known, tests against the remaining database servers can be disabled using this check input to reduce the scan time.

SQL Query Time

To conduct time-based SQL injection tests, a noticeable difference between the application's response time against normal requests and that against attack requests should exist. If the normal response time is known to be high or fluctuate considerably, this check input can be used to increase the delay that the SQL injection engine introduces via the SQL attack queries by selecting the Heavy Query option. This will ensure that even for slow responding applications that are vulnerable to SQL injection, the forced delay will result in an even slower response.

10963: Cross-Site Request Forgery

Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute unwanted actions on a web application in which he/she is currently authenticated. With a little help of social engineering (like sending a link via E-mail/chat), an attacker may force the users of a web application to execute actions of the attacker's choosing. A successful CSRF exploit can compromise end user data and operation in case of normal user. If the targeted end user is the administrator account, this can compromise the entire web application.

Criteria for identifying Cross-Site Request Forgery (CSRF)

- This check is only run against POST requests.
- The page must be either a login page or a page in restricted session (i.e., an authenticated session).

  Note: To avoid testing every POST request made during authenticated sessions, the check is run against a URL one time. This means that forms with multiple parameters will be tested one time only and not multiple times like a cross-site scripting or parameter injection check.
- The page is not a re-authentication page. This is to avoid cases where a user is asked to either change a password or provide a password when already in an authenticated session. A re-authentication page is not CSRF vulnerable.
- The page does not contain CAPTCHA. A CAPTACHA page is not vulnerable to CSRF.
- The page is not an error page or an invalid page from the server.

Check inputs are used as heuristics to help the CSRF agent refine detected results. There are a number of criteria used for CSRF detection that help to avoid false positives.

Required Inputs

- Password field names - This field is used to help identify login pages. The matches are string matches.
- Possible Username List - This field is used to help identify login pages. The matches here are string matches.

Optional Inputs

- CSRF Request Black List - This field is used to identify pages that are NOT to be flagged as vulnerable to CSRF. Matching values are identified for the name values in POST parameters.
- CSRF Response Black List - This field is used to identify error pages or invalid pages. The default value here is a combination of two regular expressions and also a string value (CAPTCHA). Matching values are identified on the response body.
- CSRF Response White List - This field is used to elevate the risk associated with this vulnerability for specific pages. By default, CSRF findings are a Medium severity. A match for values in this field will result in the finding being rated as a High severity. Matching values are identified in the response body.

10965: User Data in Query or Cookie

Transmitting password information in a query string or cookie value makes it easy for an attacker to see and tamper with login values. Recommendations include ensuring that login information is sent with a POST request over an encrypted connection and that sensitive account information is stored on the server.

Required Inputs:

- Password Field Names - List of Query or Cookie parameter names containing a password.

Possible Username List - List of Query or Cookie parameter names containing a username.

11201: Session Fixation

Password field names

List of Query or Cookie parameter names containing a password.

Possible Username List

List of Query or Cookie parameter names containing a username.

WellKnownSessionTokens

List of application parameters that have a high likelihood of containing authentication session-related information (for example, PHPSESSID).

11269: Persistent Cross-Site Scripting

Abnormal String List for XSS Engine

This check input defines the list of potentially malicious characters that are often used in Cross-Site Scripting attack strings. The WebInspect Cross-Site Scripting engine will use these characters to test any potential encoding being applied to these characters. The test results will help guide the Cross-Site Scripting engine in forming the attack strings most likely to lead to a successful injection.

Do Partial Encoding of XSS Attack String

Insecurely designed anti-XSS, blacklist filters can sometimes be bypassed by applying certain encoding transformations to the attack string. The type and degree of encoding can affect the success of an attack. This check input, when set to true, extends the XSS engine behavior to send partially encoded attacks (for example, %3cscript%3ealert(1)%3c%2fscript%3e) in addition to fully encoded attacks (%3e%3c%73%43%72%49%70%54%3e%61%6c%65%72%74%28%31%29%3c%2f%73%43%72%49%70%54%3e).

11270: Persistent Cross-Site Scripting (User Interaction)

Abnormal String List for XSS Engine

This check input defines the list of potentially malicious characters that are often used in Cross-Site Scripting attack strings. The WebInspect Cross-Site Scripting engine will use these characters to test any potential encoding being applied to these characters. The test results will help guide the Cross-Site Scripting engine in forming the attack strings most likely to lead to a successful injection.

Do Partial Encoding of XSS Attack String

Insecurely designed anti-XSS, blacklist filters can sometimes be bypassed by applying certain encoding transformations to the attack string. The type and degree of encoding can affect the success of an attack. This check input, when set to true, extends the XSS engine behavior to send partially encoded attacks (for example, %3cscript%3ealert(1)%3c%2fscript%3e) in addition to fully encoded attacks (%3e%3c%73%43%72%49%70%54%3e%61%6c%65%72%74%28%31%29%3c%2f%73%43%72%49%70%54%3e).

11277: Mobile Attack Surface Enumeration

Mobile User Agents

Each individual user agent string specified in this check input, when included in an HTTP request, will indicate to an application that the HTTP requests are originating from a mobile device. This behavior will allow WebInspect to discover additional attack surface that is exposed by an application only when it is accessed over a mobile device.

11287: Session Token Discovery

WellKnownSessionTokens

List of application parameters that have a high likelihood of containing authentication session-related information (for example, PHPSESSID).

SessionTokenIncludeList

List of application parameters and cookie parameters with a high likelihood of containing authentication session-related token values.

SessionTokenExcludeList

List of application parameters and cookie parameters that should not be treated as session-related values.

CrawlingLevelsToSearchForStatefulCookies

This check input allows you to achieve an acceptable balance between audit coverage and the scan performance by controlling the crawl depth and in turn the number of discovered URLs to test. A higher number will increase the attack surface to be tested and hence will result in longer scans.

11293: Missing Cross-Frame Scripting Protection

Password Field Names

List of Query or Cookie parameter names containing a password.

Possible Username List

List of Query or Cookie parameter names containing a username.

11307: Reliance on X-Content-Type-Options

Low Privilege Content Sniffing Triggers

This check input lists values indicative of response content that is considered to be highly susceptible to insecure treatment by web applications as well as web browsers. HTTP responses with the Content-Type header values matching one of more of the listed strings could be exploited by attackers to potentially bypass file upload filters and serve malicious content to the users.

11327: Local File Inclusion (Tomcat)

Suspicious Parameters

All the QUERY and POST parameters whose names match the patterns specified in this check input will be considered to have a higher likelihood of being vulnerable to Local File Inclusion (LFI) attacks and hence will be subjected to exhaustive testing.

11331: Ruby XML YAML Remote Code Execution

Aggressive Audit

To perform an exhaustive audit of the application, set this input to 1. When set to a value of 1, the input will force the engine to audit every single URL path crawled on a target host and port. By default, the engine is set to audit once per host and port configuration.

11349: Fortify Agent Probe Engine

Fortify Agent Suggestion Compatible Categories

This check input defines all the categories supported by the WebInspect Agent's active mode.

11351: Mobile Sensitive Information Disclosure Over HTTP

Sensitive information in mobile requests

Request content that successfully matches against these patterns will be assumed to contain highly sensitive material and will result in a finding if the request is being transmitted over an insecure channel.

11352: Information Leakage via BREACH Vulnerability

The BREACH attack exploits the peculiar behavior of an application configured to transmit information over SSL/TLS connections while using the Gzip compression algorithm to reduce the size of the transmitted content. Through the analysis of this behavior, an attacker can extract sensitive information from the application's response content.

Classified Information Indicators

This check input lists the keywords and patterns to be matched against form input names and JavaScript code snippets to identify presence of classified information within response text that needs to be protected against theft while in transit.

3 Compliance Manager (WebInspect Only)

About the Compliance Manager Tool

WebInspect employs an extensive arsenal of attack agents designed to detect security flaws in web-based applications. It probes your system with thousands of HTTP requests and evaluates each individual response. This session-based scan reports each vulnerability, pinpoints its location in the application, and recommends corrective actions you should take. It is, basically, a quantitative analysis of your system.

WebInspect can also perform a qualitative analysis by grading how well your application complies with certain government-mandated regulations or corporate-defined guidelines. For example, the Health Insurance Portability and Accountability Act (HIPAA) requires healthcare providers using web-based applications to provide "procedures for creating, changing, and safeguarding passwords." With WebInspect, you can assess your application and then generate a Compliance Report that measures how well your application satisfies this HIPPA rule.

How It Works

You create a compliance template that associates requirements with one or more attack agents or vulnerabilities. For example, you might include the statement (or question) "The application will not use any 'hidden' fields." The attack agent that tests for compliance to this requirement is Hidden Form Value, ID #4727 (which is one of the agents in the General Text Searching group).

Compliance templates are completely flexible. You can enable or disable individual requirements. You can also modify requirements by adding or removing attack agents or threat classes. For maximum flexibility, you can even create your own agents and associate them with a user-defined requirement.

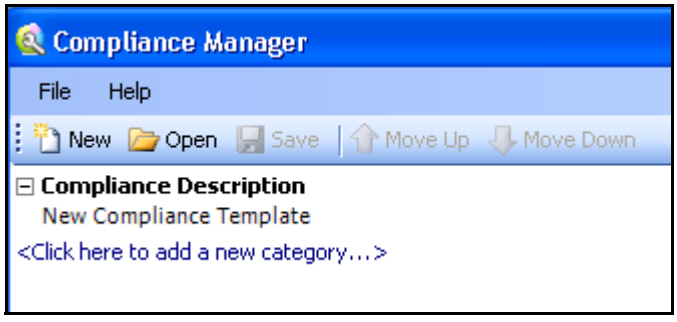
WebInspect includes sample compliance templates that you can edit to fit your company's specific requirements.

Creating a Compliance Template

To create a compliance template.

1 On the WebInspect menu bar, click **Tools** → **Compliance Manager**.

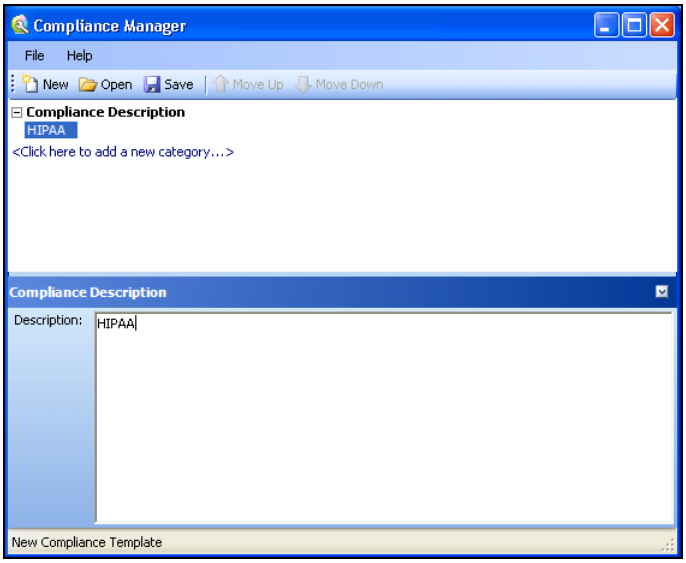
The *Compliance Manager* window opens, displaying the outline of a new template.



2 Click the phrase “New Compliance Template.”

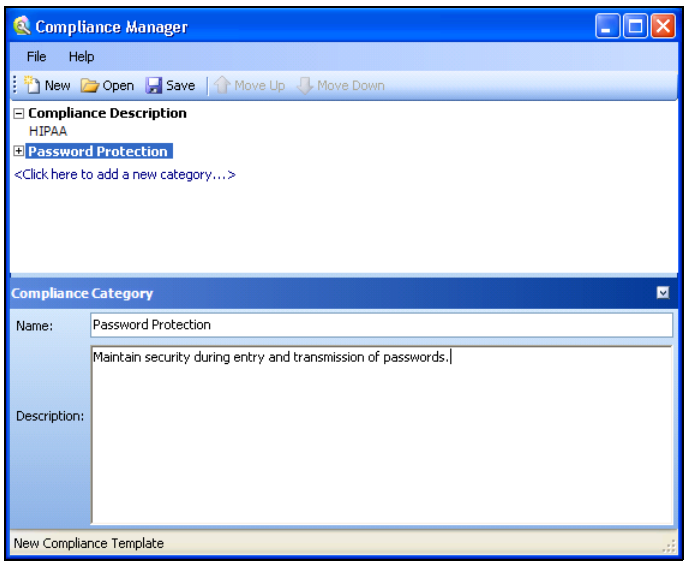
The Compliance Manager creates an editing area in the lower half of the window.

3 In the editing area, replace the phrase “New Compliance Template” with a description of the template you are creating (“HIPAA” in this example).



4 Click the phrase “<Click here to add a new category...>.”

5 In the editing area, enter the name and description of the new category. In this example, the name is “Password Protection” and the description is “Maintain security during entry and transmission of passwords.”



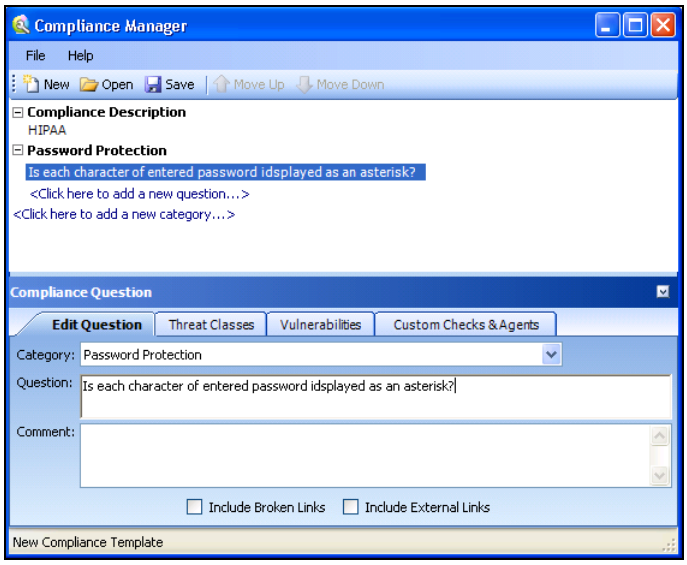
6 Click the plus sign ⊞ to expand the node labeled Password Protection.

7 Click the phrase “<Click here to add a new question...>.”

8 Click the phase “New Question.”

The editing area displays tabs allowing you to create a question related to the category “Password Protection.”

9 In the **Question** area, type a question related to the category. This example asks the question, “Is each character of entered password displayed as an asterisk?”

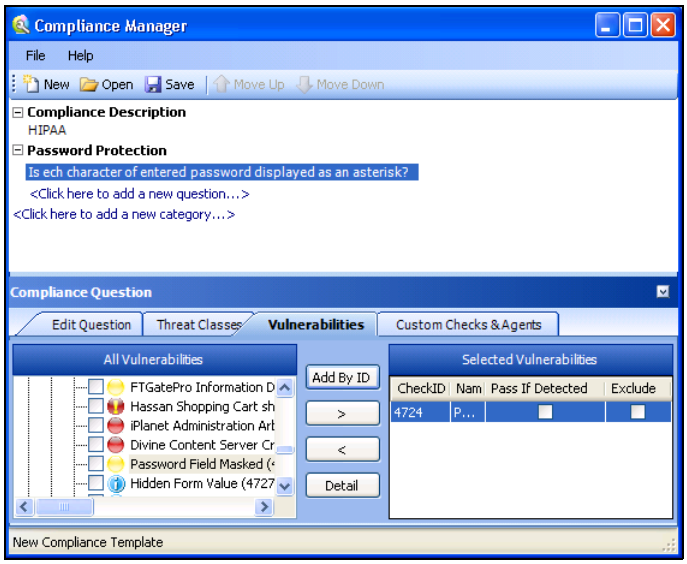


10 You can associate this question with threat classes, vulnerabilities defined by HP, or a custom check or agent that you previously created. For this example, click the **Vulnerabilities** tab and then click **Add By ID**.

You can also select a vulnerability and click  to include it in the **Selected Vulnerabilities** section for this question.

11 On the *Add Check By ID* window, enter 4724 and click **OK**. [4724 is the ID number of the “Password Field Not Masked” check.] Note: You can add multiple IDs (one per line).

The check you specified appears in the **Selected Vulnerabilities** area



12 The **Selected Vulnerabilities** area contains two check boxes:

- **Pass If Detected**—Select this option if the check is designed to confirm an attribute that contributes to application security. You might use this if, for example, you develop a custom check that checks for the existence of a file (such as Privacy Policy.html) that is part of your compliance program.
- **Exclude**—Select this option if you add a group of checks, but want to exclude specific ones.

In this example, do not select either check box.

13 Continue adding threat classes, vulnerabilities, or custom checks until you have included all that sufficiently test your application for the compliance question.

14 Create additional questions and categories using the above procedures until the compliance template is complete.

15 Click **Save**.

Usage Notes

To rearrange categories or items, select an item and click **Move Up** or **Move Down**.

To insert categories or items, you can alternatively right-click a category/question and select **Insert** from the shortcut menu. The item will be inserted above the selected item.

You can add an HTML link to any description or question, as depicted in the following illustration.

Testing for Compliance

To test your website for compliance:

1 Create a compliance template.

2 Scan your website.

3 On the WebInspect **Start** page, click **Generate a Report**.

The *Generate a Report* window appears.

4 If the scan data is stored in a different database, click **Change DB** and then select a database.

5 Select a scan (designated by name, URL, or IP address).

6 Click **Next**.

7 Select **Compliance**.

8 If you want to produce individual reports on separate tabs (rather than combining all reports on one tab), select **Open Each Report in a Separate Tab**.

9 Select either **Adobe PDF** or **HTML** as the report format.

Adobe Reader 7 or newer is required to read reports in portable data format (pdf).

10 Specify a compliance template. You can select a default template from the list, click the browse button  to browse for templates you have created, or open the Compliance Manager and create a custom template.

11 Click **Finished**.

12 After WebInspect generates the report and displays it on a tab, you can save a report by clicking the Save Report icon on the toolbar.

4 Cookie Cruncher

About the Cookie Cruncher Tool

The Cookie Cruncher analyzes cookies to determine the relative ease with which an attacker could predict or determine the value of a session ID generated by a server and delivered to a client via a cookie.

Background

The web's Hypertext Transfer Protocol (HTTP) is stateless, meaning that each communication is discrete and unrelated to those that precede or follow. Because there is no continuity inherent in the protocol, application designers introduced the concept of "session." A session is defined as all activity by a user with a unique IP address on a website during a specified period of time. When a user logs into an application, a session is created on the server to maintain the state for other requests originating from the same user.

Each session has a unique identifier (session ID). This text string is transmitted between the client and the server, and may be stored in cookies, URLs, or hidden fields of web pages. One problem with session IDs, however, is that many websites generate them using algorithms based on easily predictable variables, such as time or IP address. This predictability makes the websites vulnerable to session hijacking.

Session hijacking involves an attacker using session IDs to seize control of a legitimate user's session while that session is still in progress. The attacker can then gain complete access to the user's data, and can perform all operations that are normally available to the legitimate owner of the session.

Using the Cookie Cruncher

To use the Cookie Cruncher:

1. In the **URL** box, enter the URL of the site you want to test.

   If you are using the Cookie Cruncher to examine a site you have scanned with WebInspect, follow these steps:

   a. In the WebInspect navigation pane, click the cookies icon 

   . All HTTP responses containing a "Set-Cookie:" header are listed in the information pane.

   b. Double-click one of the listed responses.

   c. Click **Request**.

   d. Copy the request and paste it into the Cookie Cruncher's **Request** area.

2. In the **Sample** box, enter the number of requests the Cookie Cruncher should send to the server (expecting a cookie to be returned). A higher number of samples increases processing time, but produces more reliable result; a minimum of 100 is suggested.

3. Click **Sample**.

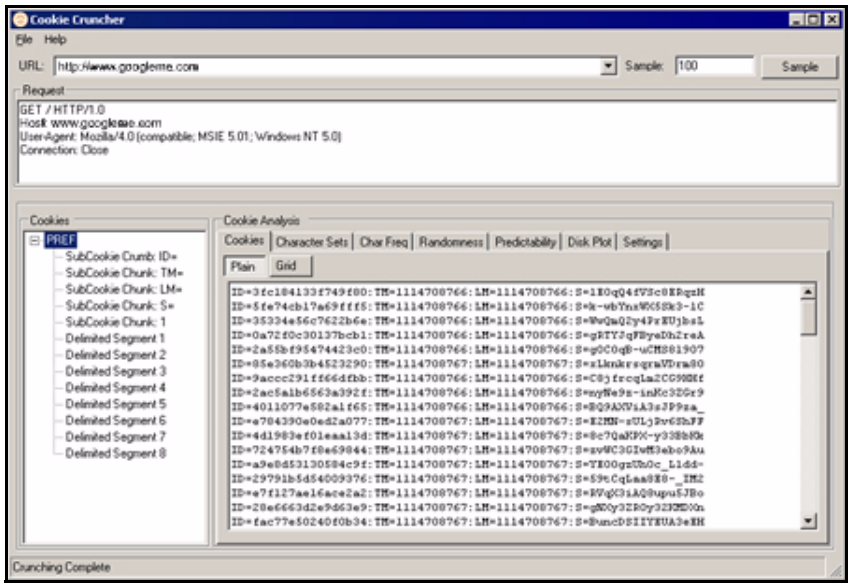
   As cookies are collected, the Cookie Cruncher organizes them into a tree hierarchy displayed in the vertical pane on the left side of the window.

4 Click a cookie in the tree hierarchy to analyze it. If subcookies are found, the Cookie Cruncher modifies the tree hierarchy; click the plus sign ⊞ to expand the level. Repeat as necessary.

5 To view the analysis, select a cookie or subcookie and click the various tabs.

6 To save the sampled cookies for future analysis, click **File** → **Save**.

Cookie Cruncher cannot open and display a saved cookie file (.sck) if it contains fewer than four cookies.



Subcookies

Subcookies are either portions of cookie values that are common to many cookies, or interpreted values.

When the same string of characters appears in multiple cookies, you can choose that as a subcookie. The recurring expression will be eliminated from the cookies that contain it, and those cookies will be re-analyzed. The portion that is removed (the recurring expression) is called a “subcookie crumb.”

In the following sample, “086-” would be detected as a recurring expression:

086-1123

086-1127

087-6281

086-1132

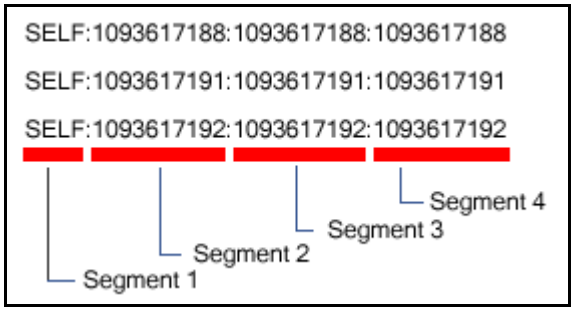
088-0518

087-6282

Analysis of those cookies containing the recurring expression (1123, 1127, 1132) would reveal the (most likely) incrementing cookie values that were interleaved with values from some other source.

If the detected character set of a sample consists of just 10 characters (Q-Z), these characters could possibly represent the digits 0-9. Choosing the re-encode option would run the cookies through an appropriate decoder algorithm (base-10, base-16, base-64, etc.) and re-analyze the cookies.

The “Delimited Segment” option(s) allow you to select the delimited portions of cookies. For example, the following subcookies contain four delimited segments.



To analyze the second segment of all subcookies, you would click the **Select Subcookie** list and select **Delimited Segment 2**.

Cookie Cruncher Tabs

Use the Cookie Cruncher tabs to analyze the sampled cookies. The tabs are:

- Cookies
- Character Sets
- Char Freq
- Randomness
- Predictability
- Disk Plot

Cookies Tab

This tab lists all cookies received from the server. You can view them either in plain or grid format by clicking the appropriate button.

Character Sets Tab

This tab displays the character set used to format the cookie:

A = alphabetic character (letters A-Z)

N = numeric character (numbers 0-9)

H = hexadecimal character (0-F)

T = Text A-Z, a-z

I = Illegal (anything else)

D = delimiter

Char Freq Tab

This tab displays a graph showing the number of times each ASCII character appeared in the total sample of cookies. A pale blue dot indicates an ASCII character whose number of appearances equals the number of cookies. A highlighted character indicates that it may be a delimiter (which is usually a character such as a comma, colon, or semicolon, but could also be something unusual such as “Z”).

Randomness Tab

This tab attempts to differentiate between random and non-random portions of cookies, based on the sample obtained.

Use the Grid view to illustrate the analysis of each column. The color key is:

Red = No randomness (or very little)
Orange = Somewhat random
White = Random

The top row of the grid indicates the numeric position of each character.

The second row displays, for each character position, a number representing the relative randomness of the character. This is actually the average number of bits that change per column from one cookie to the next.

Use the Graph view to illustrate the randomness level in a graphic format. The dashed green line represents the optimum (best practice) level of randomness. The red line represents the randomness of the cookies in the sample. In a well designed cookie, the red line should follow the green line. When the graph view is selected, you can save the graph (in BMP, GIF, PNG, or JPG format) using the **Save Graph** command in the **File** menu.

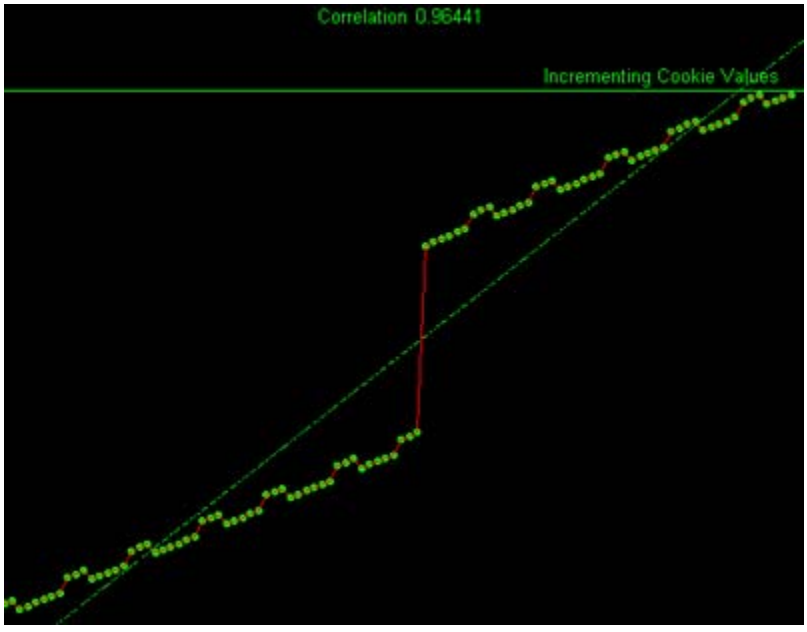
Predictability Tab

The Cookie Cruncher analysis produces a correlation value ranging from 0 to 1 and displays it at the top of the graph. A low value indicates that cookie generation is more random; a higher value indicates greater predictability.

The value of each cookie is plotted (on the Y axis) against the time the cookie was received (on the X axis). A scattered distribution indicates randomness, whereas a pattern approaching a line indicates predictability.

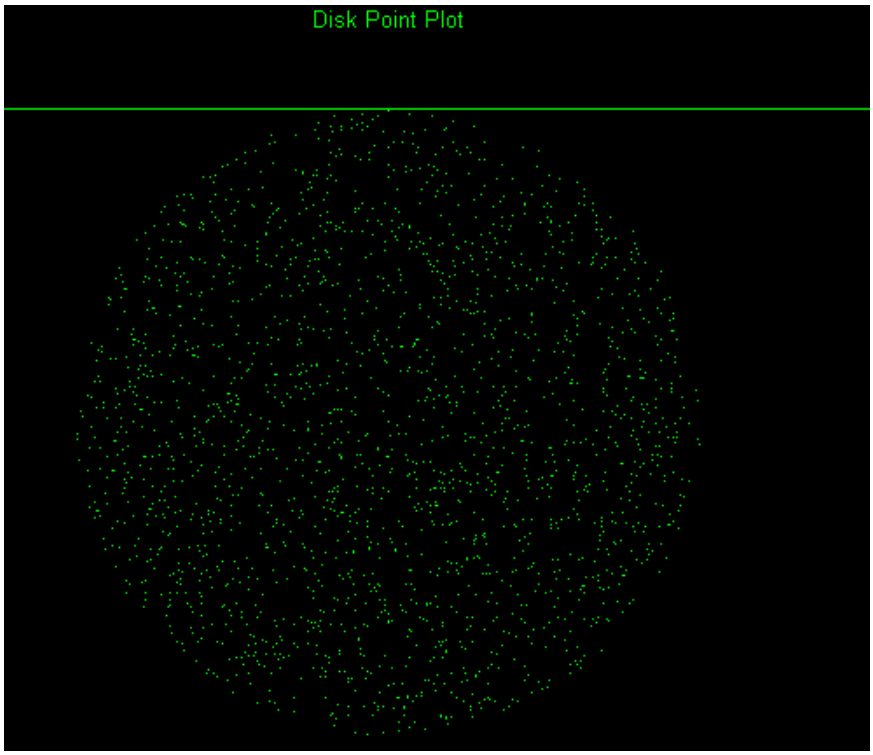
If the correlation is .9 or greater, the graph displays the header “Incrementing Cookie Values” or “Decremented Cookie Values” and draws a “best fit” line.

Only decimal or hexadecimal values can be plotted.



Disk Plot Tab

This graph plots a cookie's value against the sine and cosine functions. When random data is plotted, the points are evenly distributed around the plotting area. Only decimal or hexadecimal values can be plotted.



Cookie Cruncher Settings

To modify the Cookie Cruncher settings:

1. Click **Edit** → **Settings**.
2. Select either the **General**, **Authentication**, or **Proxy** category and enter the settings described in the following sections.
3. Click **OK**.

General

Thread Count

Specify the maximum number of threads that can be created. The Cookie Cruncher can be configured to send up to 75 concurrent HTTP requests before waiting for an HTTP response to the first request. The default setting is 10. Increasing the thread count will increase the speed of the process, but might also exhaust your system resources as well as those of the server you are scanning. While most servers can handle a large number of requests, servers in development environments sometimes have licensing limitations that allow only five or fewer users to be connected at a single time. In such cases, you should reduce the thread count to be less than 5.

Socket Timeout

Specify the maximum number of open sockets permitted. A higher number of open sockets results in a faster process. However, a setting that exceeds a server's threshold may result in false positives.

If the Cookie Cruncher runs on Windows XP with Service Pack 2 (SP2), the number of open sockets should be set to 10.

Custom Delimiters

The Cookie Cruncher interprets certain characters (such as /.-!,:;=) as delimiters. In some cases, you may want to substitute your own list. For example, a cookie having a value of "ABC123456-C:Program" contains two default delimiters — a dash (-) and a colon (:) — and would therefore be split into three parts. However, if you specify only the dash as a delimiter, the cookie would be split into just two parts.

The user-specified list, if present, will cause an extra subcookie type to appear in the tree, in addition to the regularly parsed subcookie types. The subcookie item may not appear when the number of cookies having the delimiter(s) is less than 10 percent of the total cookie sample.

To create a list of custom delimiters, select the **Parse with Custom Delimiters** check box and then enter one or more delimiters in the **Characters** box.

Authentication

Authentication Method

If authentication is required, select a type from the **Authentication** list:

Authentication	Description
Automatic	Allow the Cookie Cruncher to determine the correct authentication type. Note that automatic detection slows the scanning process. If you know and specify one of the other authentication methods, scanning performance is noticeably improved.
HTTP Basic	A widely used, industry-standard method for collecting user name and password information. The web browser displays a window for a user to enter a previously assigned user name and password. If the web server verifies that the user name and password correspond to a valid user account, a connection is established. The advantage of Basic authentication is that it is part of the HTTP specification and is supported by most browsers. The disadvantage is that web browsers using Basic authentication transmit passwords in an unencrypted form. By monitoring communications on your network, an attacker can easily intercept and decode these passwords using publicly available tools. Therefore, Basic authentication is not recommended unless you are confident that the connection between the user and your web server is secure.
NT LAN Manager (NTLM)	NTLM is an authentication process used by all members of the Windows NT family of products. It uses a challenge/response process to prove the client's identity without requiring that either a password or a hashed password be sent across the network. Use NTLM authentication for servers running IIS.

Authentication Credentials

Type a user ID in the **User name** box and the user's password in the **Password** box. To guard against mistyping, repeat the password in the **Confirm Password** box.

Proxy

Use these settings to access the Cookie Cruncher through a proxy server.

Direct Connection (proxy disabled)

Select this option if you are not using a proxy server.

Auto detect proxy settings

Use the Web Proxy Autodiscovery (WPAD) protocol to locate a proxy autoconfig file and configure the browser's web proxy settings.

Use Internet Explorer proxy settings

Import your proxy server information from Internet Explorer.

Use Firefox proxy settings

Import your proxy server information from Firefox.

Configure a proxy using a PAC file

Load proxy settings from the Proxy Automatic Configuration (PAC) file in the location you specify in the **URL** box.

Explicitly configure proxy

Configure a proxy by entering the requested information.

1. In the **Server** box, type the URL or IP address of your proxy server, followed (in the **Port** box) by the port number (for example, 8080).
2. Select a protocol for handling TCP traffic through a proxy server: SOCKS4, SOCKS5, or standard.
3. If authentication is required, select a type from the **Authentication** list. For information on the authentication types available to you, see Authentication Types on page 44.
4. If your proxy server requires authentication, enter the qualifying user name and password.
5. If you do not need to use a proxy server to access certain IP addresses (such as internal testing sites), enter the addresses or URLs in the **Bypass Proxy For** box. Use commas to separate entries.

Authentication Types

The following table lists the available authentication types:

Authentication	Description
HTTP Basic	A widely used, industry-standard method for collecting user name and password information. The Web browser displays a window for a user to enter a previously assigned user name and password. If the Web server verifies that the user name and password correspond to a valid user account, a connection is established. The advantage of Basic authentication is that it is part of the HTTP specification and is supported by most browsers. The disadvantage is that Web browsers using Basic authentication transmit passwords in an unencrypted form. By monitoring communications on your network, an attacker can easily intercept and decode these passwords using publicly available tools. Therefore, Basic authentication is not recommended unless you are confident that the connection between the user and your Web server is secure.
NT LAN Manager (NTLM)	NTLM is an authentication process used by all members of the Windows NT family of products. It uses a challenge/response process to prove the client's identity without requiring that either a password or a hashed password be sent across the network. Use NTLM authentication for servers running IIS.
Kerberos	Kerberos uses the Needham-Schroeder protocol as its basis. It uses a trusted third party, termed a Key Distribution Center (KDC), which consists of two logically separate parts: an Authentication Server (AS) and a Ticket Granting Server (TGS). The client authenticates itself to AS, then demonstrates to the TGS that it is authorized to receive a ticket for a service (and receives it). The client then demonstrates to a Service Server that it has been approved to receive the service.
Digest	The Windows Server 2003 operating system implements the Digest Authentication protocol as a security support provider (SSP), a dynamic-link library (DLL) that is supplied with the operating system. Using digest authentication, your password is never sent across the network in the clear, but is always transmitted as an MD5 digest of the user's password. In this way, the password cannot be determined by sniffing network traffic.
Automatic	Allow the Web Form Editor to determine the correct authentication type. Note that automatic detection slows the scanning process. If you know and specify one of the other authentication methods, scanning performance is noticeably improved.
Negotiate	If both the server and client are using Windows 2000 or later, Kerberos authentication is used. Otherwise, NTLM authentication is used. This method is also known as Integrated Windows authentication.

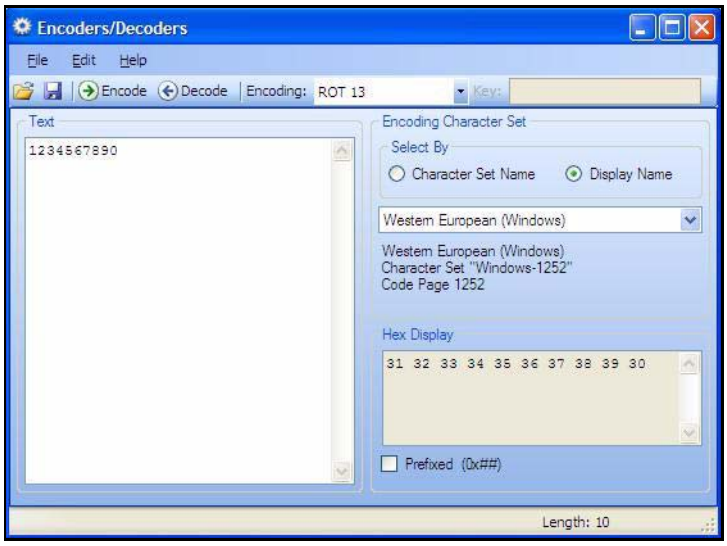
HTTPS Proxy Settings

For proxy servers accepting HTTPS connections, select **Specify Alternative Proxy for HTTPS** and provide the requested information.

5 Encoders / Decoders

About the Encoders/Decoders Tool

This tool allows you to encode and decode values using Base64, hexadecimal, MD5, and other schemes. You can also encode a string into a Unicode string and use special characters in URL construction. During the analysis of your scan results, when you encounter a string that you suspect is in an encoded or encrypted format, you can simply copy the string, paste it into the Encoders/Decoders tool, and then click **Decode**.



Encoding a String

To encode a string:

1. Type (or paste) a string into the **Text** area, or load the contents of a file by selecting **File** → **Open**.
2. Select an encoding character set using either the **Character Set Name** or the **Display Name**.
3. Select a cipher type from the **Encoding** list. For more information, see Encoding Types on page 46.
4. If necessary, type a key in the **Key** box. When a valid key is entered, the **Encode** and **Decode** buttons become enabled.
5. Click **Encode**.

   The **Text** area displays the encoded string; the **Hex Display** area displays the hexadecimal value of each character in the encoded string (formatted in the character set that you select).

Decoding a String

To decode a string:

1. Type (or paste) a string in the text area, or load the contents of a file by selecting **File** → **Open**.
2. Select an encoding character set using either the **Character Set Name** or the **Display Name**.
3. Select a cipher type from the **Encoding** list.
4. If necessary, type a key in the **Key** box.
5. Click **Decode**.

You can also use WebInspect's encoding and decoding capabilities in the HTTP Editor. Right-click while editing a session to access encoding and decoding options.

Manipulating Encoded Strings

The encoded form of a string may contain characters that are non-printable. This often occurs when using a hash-based encoding scheme or any encoding scheme that requires a key. Since non-printable characters cannot be copied to the Windows clipboard, you cannot simply copy from or paste into the Encoder/Decoder. However, there are three methods you can use to work around this limitation:

- Save the encoded string to a file and, when you want to decode it, select **File** → **Open** to load it into the Encoder tool. Then decode it using the original method and (if applicable) key.
- Also, after encoding the string using the chosen encoding method and key, you can encode the resulting string using the base 64 method; then copy the string to the clipboard, paste the clipboard contents, decode using base 64, and decode again using the original method and (if applicable) key.

Encoding Types

The Encoder/Decoder allows you to select the encoding types described below.

- 3DES is a mode of the DES encryption algorithm that encrypts data three times (the string is encrypted, then the encryption is encrypted, and the resulting cipher text is encrypted a third time). The key must be 128 or 192 bits (16 or 24 characters).
- Base64 encodes and decodes triplets of 8-bit octets as groups of four characters, each representing 6 bits of the source 24 bits. Only characters present in all variants of ASCII and EBCDIC are used, avoiding incompatibilities in other forms of encoding.
- Blowfish is an encryption algorithm that can be used as a replacement for the DES algorithm.
- DES (Data Encryption Standard) is a widely-used method of data encryption that can use more than 72 quadrillion different private (and secret) encryption keys. Both the sender and the user must use the same private key.
- HEX is hexadecimal.
- MD5 produces a 128-bit "fingerprint" or "message digest" of whatever data you enter.
- RC2 is a variable key-size block cipher designed by Ronald Rivest. It has a block size of 64 bits and is about two to three times faster than DES in software.
- RC4 is a stream cipher designed by Ronald Rivest. It is a variable key-size stream cipher with byte-oriented operations. Used for file encryption in products such as RSA SecurPC and also used for secure communications, as in the encryption of traffic to and from secure websites using the SSL protocol.

- ROT13 is a simple Caesar cipher used for obscuring text by replacing each letter with the letter thirteen places down the alphabet.
- SHA1 is Secure Hash Algorithm, a one-way hash function developed by NIST and defined in standard FIPS 180. SHA-1 is a revision published in 1994; it is also described in ANSI standard X9.30 (part 2).
- SHA256 uses 256-bit encryption.
- SHA384 uses 384-bit encryption.
- SHA512 uses 512-bit encryption.
- ToLower changes upper-case letters to lower-case.
- ToUpper changes lower-case letters to upper-case.
- TwoFish is an encryption algorithm based on an earlier Blowfish.
- Unicode provides a unique number for every character, regardless of the platform, program, or language.
- URL creates values that can be used for URL-encoding non-standard letters and characters for display in browsers and plug-ins that support them.
- XHTML encapsulates the entered data with text tags: <text>data</text>
- XOR performs an Exclusive OR operation. You must provide a key. If the length of the key string is only one character, that character is ORed against each character in the encode/decode string.

Prefixed

C and languages with a similar syntax (such as C++, C#, Java and JavaScript) prefix hexadecimal numerals with “0x” (for example, 0x5A3). The leading zero allows the parser to recognize a number, and the “x” stands for hexadecimal.

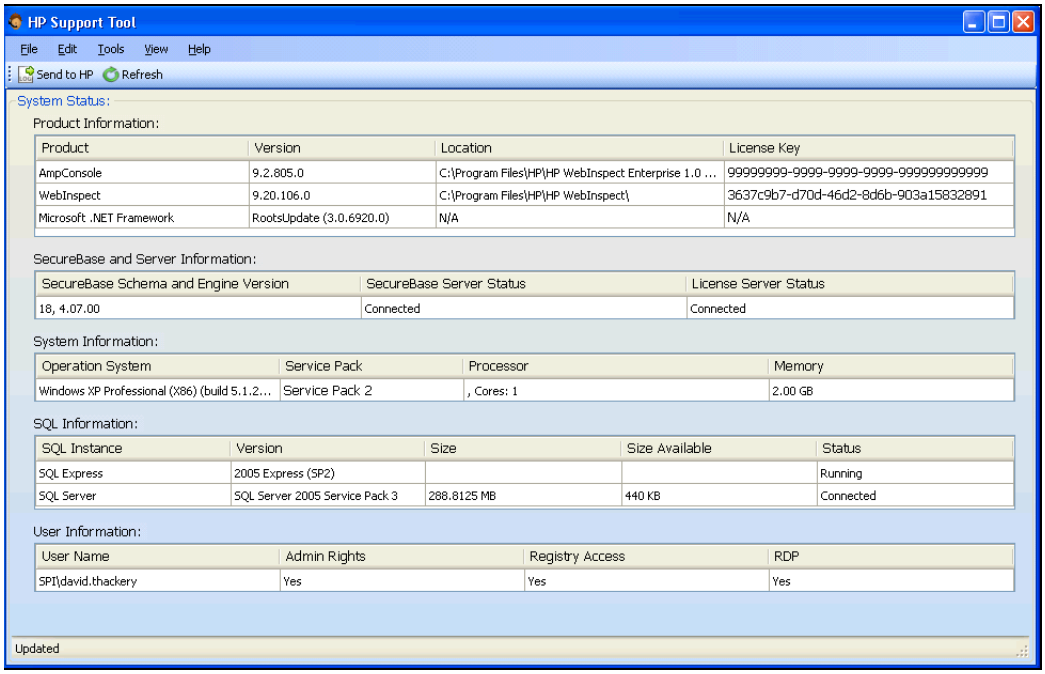
6 HP Support Tool (WebInspect Only)

About the HP Support Tool

The HP Support tool provides a quick and simple method for uploading files that may help HP support personnel to analyze and resolve any problems you encounter while using Application Security Center products. All communication uses Secure Sockets Layer (SSL) or FTP Secure (FTPS) protocol.

To launch the Support tool, click **Start** → **All Programs** → **HP** → **WebInspect** → **HP Support Tool**.

When first opened, the Support Tool displays information about ASC products and related system components. If data is not displayed, click **Refresh**.



Before sending data to HP, you may want to perform the following functions, which are available from the Tools menu.

- **Refresh WebInspect scans** - Deletes the scans.xml file and regenerates it, thereby refreshing the list of WebInspect scans that are displayed on the Manage Scans page. This is intended for use if scans are not displaying properly. Use this function only if directed by HP Support personnel.
- **Restore WebInspect SecureBase** - Replaces the current SecureBase with the factory default version. You can replace the Main SecureBase, the Scheduler SecureBase, or both. Use this function only if directed by HP Support personnel.
- **Check Services** - Displays a list of services associated with Application Security Center (ASC) products, allowing you to start, stop, or restart the services you select.

To send data to HP:

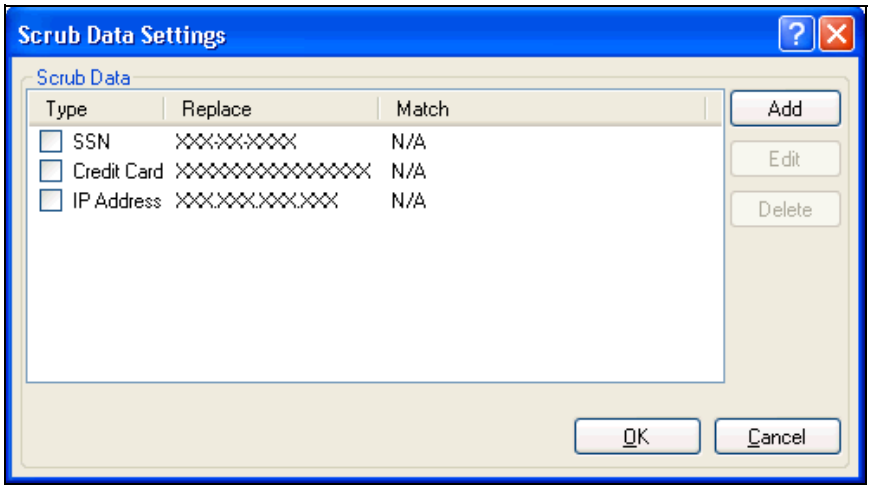
1. Click **Send to HP**.

   The *Send to HP* dialog appears.

2. Select one or more installed products.
3. Choose which product-related items you want to send to HP Support.
4. If you select **Include an additional directory**, click **Browse** and identify the directory. The contents of that directory (and all subdirectories) will be uploaded to HP Support.
5. If you include scans, select the appropriate scan export options.
   - **Include scan logs**: Include log files associated with the selected scans.
   - **Only export scan logs (local scans only)**: Do not include scan data.
   - **Scrub scan data**: Use a "scrubbing" feature that excludes sensitive data from the exported scan. To select specific scrubbing functions, click the Configure hyperlink to the right of the check box; see Scrubbing Data for instructions.
6. Click **Next**.
7. Enter the case number you obtained from HP support personnel (required).
8. To enter or modify customer information, click the **Customer Contact Information** hyperlink. First name, last name, and e-mail address are required.
9. Select an option from the Communication Settings group:
   - Send to HP Support via FTP
   - Send to HP Support via Secure Channel
   - Send the files to a local directory
10. Click **Send**.

Scrubbing Data

The Scrub Data Settings contain, by default, three non-editable regular expression functions that will substitute an X for each digit in a string formatted as a Social Security number, credit card number, or IP address.



To include these search-and-replace functions:

1. Select **Scrub Scan Data** in the **Scan Export Options** section.
2. Click **Configure**.
3. On the *Scrub Data Settings* window, select one or more of the functions in the **Type** column.
4. To create a Scrub Data function:
   a. Click **Add**.
   b. On the *Add Scrub Entry* window, select either **Regex** or **Literal** from the **Type** list.
   c. In the **Match** box, enter the string (or a regular expression representing a string) that you want to locate. If using a regular expression, you can click the browse button to open the Regular Expression Editor, with which you can create and test your regular expression.
   d. In the **Replace** box, enter the string that will replace the target specified by the **Match** string.
   e. Click **OK**.

Support Settings

Proxy

If you are not using a proxy server, select **Direct Connection** (proxy disabled).

If you are required to use a proxy server, select one of the following.

- **Auto detect proxy settings**: Use the Web Proxy Autodiscovery (WPAD) protocol to locate a proxy autoconfig file and configure the browser's web proxy settings.
- **Use Internet Explorer proxy settings**: Import your proxy server information from Internet Explorer.
- **Use Firefox proxy settings**: Import your proxy server information from Firefox.

  Electing to use browser proxy settings does not guarantee that you will access the Internet through a proxy server. If the Firefox browser connection settings are configured for "No proxy," or if the Internet Explorer setting "Use a proxy server for your LAN" is not selected, then a proxy will not be used. To access browser proxy settings:

  Internet Explorer: **Tools** → **Internet Options** → **Connections** → **LAN Settings**

  Firefox: **Tools** → **Options** → **Advanced** → **Network** → **Settings**

- **Configure a proxy using a PAC file**: Load proxy settings from the Proxy Automatic Configuration (PAC) file in the location you specify in the URL box.
- **Explicitly configure proxy**: Configure a proxy by entering the requested information.

SQL Server

To override SQL Server settings defined in WebInspect's application settings, select **Define local SQL Server settings** and supply the requested information. This feature is used most often to collect data from a different computer (that is, a machine other than the one on which this Support Channel software is running).

Server name

Enter or select the name of the server that will store WebInspect data.

Log on to the server

Specify the type of authentication used for the selected server:

- **Use Windows Authentication**—Log on by submitting the user's Windows account name and password.
- **Use SQL Server Authentication—**Use SQL Server authentication, which relies on the internal user list maintained by the SQL Server computer. Enter the user name and password.

Connect to a database

After supplying a server name, enter or select a specific database.

Advanced

Support Channel URL

If you are instructed to change the default Support Channel URL, do it here.

Communication Settings

Select one of the available protocols for sending files to the HP servers:

- **FTPS**—FTP Secure is an extension to the commonly used File Transfer Protocol (FTP) that adds support for the Transport Layer Security (TLS) and the Secure Sockets Layer (SSL) cryptographic protocols.
- **HTTPS**—Hypertext Transfer Protocol Secure is a combination of the Hypertext Transfer Protocol with the SSL/TLS protocol to provide encryption and secure identification of the server.

Log Level

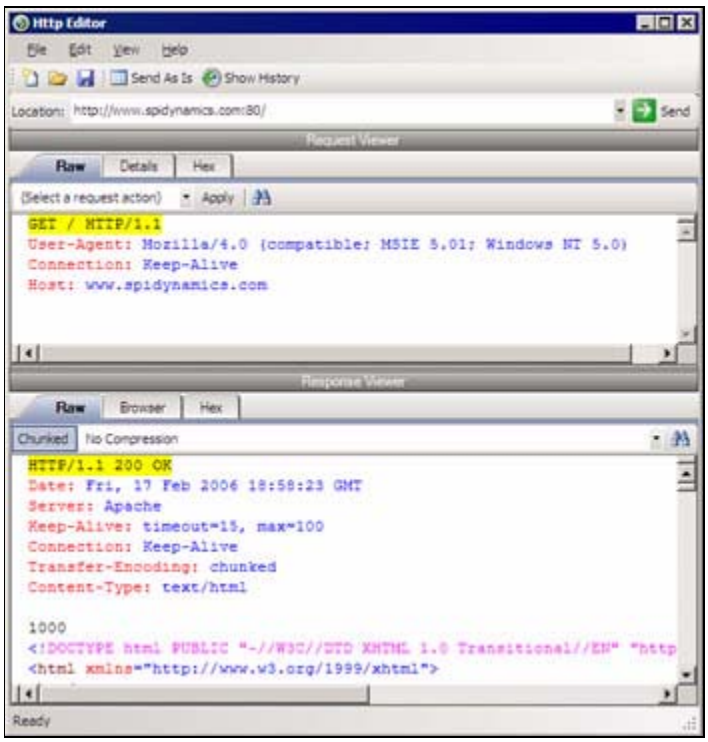
Specify how different functions and events that occur within the Support Tool should be logged. The choices are (from most verbose to least verbose) Debug, Info, Warn, and Error.

7 HTTP Editor

About the HTTP Editor Tool

Use the HTTP Editor to create or edit requests, send them to a server, and view the response either in raw HTML or as rendered in a browser. The HTTP Editor is a manual hacking tool that requires a working knowledge of HTML, HTTP, and request methods.

To set proxy and authorization parameters, if necessary, select **Edit** → **Settings**.



Request Viewer

The Request Viewer pane contains the HTTP request message, which you can view in three different formats using the following tabs:

- **Raw**—Depicts the line-by-line textual format of the request message.
- **Details**—Displays the header names and field values in a table format.
- **Hex**—Displays the hexadecimal and ASCII representation of the message.
- **XML**—Displays any XML content in the message body (Note: This tab appears only if the request contains XML-formatted data).

Response Viewer

The Response Viewer pane contains the HTTP response message, which you can also view in three different formats using the following tabs:

- **Raw**—Depicts the line-by-line textual format of the response message.
- **Browser**—Displays the response message as rendered in a browser.
- **Hex**—Displays the hexadecimal and ASCII representation of the response message.
- **XML**—Displays any XML content in the message body (Note: This tab appears only if the response contains XML-formatted data).

HTTP Editor Menus

File Menu

The **File** menu contains the following commands:

- **New Request**—Deletes all information from previous sessions and resets the Location URL.
- **Open Request**—Allows you to load a file containing an HTTP request saved during a previous session.
- **Save Request**—Allows you to save an HTTP request.
- **Save Request As**—Allows you to save an HTTP request.
- **URL Synchronization**—When selected, any characters you type into the Address combo box are added to the Request-URI of the HTTP request line.
- **Send As Is**—If you select this option, the HTTP Editor will not modify the request, regardless of any other settings you may select. This allows you to send a purposely malformed message. Authentication and proxy settings are disabled when using this option.

  Note: You may manually edit the request to go through a proxy, but most standard HTTP proxy servers cannot process non-compliant HTTP requests.
- **Exit**—Closes the HTTP Editor.

Edit Menu

The **Edit** menu contains the following commands:

- **Cut**—Deletes selected text and saves it to the clipboard.
- **Copy**—Saves the selected text to the clipboard.
- **Paste**—Inserts text from the clipboard
- **Find**—Displays a window that allows you to search for text that you specify.
- **Settings**—Allows you to configure request, authentication, and proxy parameters for the HTTP Editor.

View Menu

The View menu contains the following commands:

- **Show History**—Displays a pane listing all HTTP requests sent.
- **Word Wrap**—Causes all text to fit within the defined margins.

Help Menu

The Help menu contains the following commands:

- **HTTP Editor Help**—Opens the Help file with the Contents tab active.
- **Index**—Opens the Help file with the Index tab active.
- **Search**—Opens the Help file with the Search tab active.
- **About HTTP Editor**—Displays information about the HTTP Editor.

Request Actions

The following options are available from the **Request Action** list in the Request Viewer pane.

PUT File Upload

The PUT method requests that the enclosed entity be stored under the supplied Request-URI.

To write a file to a server:

1. Select **PUT File Upload** from the drop-down list on the Request Viewer pane.
2. In the text box that appears to the right of the list, type the full path to a file
   - or -
   Click the Open Folder icon and select the file you want to upload.
3. Click **Apply**. This will also recalculate the content length.

Change Content-Length

In normal mode, if you edit the message body of the request, the HTTP Editor recalculates the content length and substitutes the appropriate value in the Content-length header. However, when using the Send As Is option, the HTTP Editor does not modify the content length. You can force this recalculation before sending the request by selecting **Change Content-Length** and clicking **Apply**.

URL Encode/Decode Param Values

The specification for URLs (RFC 1738, Dec. ’94) limits the use of characters in URLs to a subset of the US-ASCII character set. HTML, on the other hand, allows the entire range of the ISO-8859-1 (ISO-Latin) character set to be used in documents, and HTML4 expands the allowable range to include the complete Unicode character set as well. To circumvent this limitation, you can encode non-standard letters and characters for display in browsers and plug-ins that support them.

URL encoding of a character consists of a “%” symbol, followed by the two-digit hexadecimal representation of the ISO-Latin code point for the character. For example:

- The asterisk symbol (*) = 42 decimal in the ISO-Latin set
- 42 decimal = 2A hexadecimal
- URL code for asterisk = %2A

You can use URL encoding to bypass an intruder detection system (IDS) that inspects request messages for certain keywords using only the ISO-Latin character set. For example, the IDS may search for “login” (in ISO-Latin), but not “%4C%4F%47%49%4E” (the URL-encoded equivalent).

To substitute URL code for parameters throughout the entire message, select **URL Encode Param Values** and click **Apply**.

To translate URL-encoded parameters to ISO-Latin, select **URL Decode Param Values** and click **Apply**.

Unicode Encode/Decode Request

The Unicode Worldwide Character Standard includes letters, digits, diacritics, punctuation marks, and technical symbols for all the world's principal written languages, using a uniform encoding scheme. Incorporating Unicode into client-server applications and websites offers significant cost savings over the use of legacy character sets. Unicode enables a single software product or a single website to be targeted across multiple platforms, languages and countries without re-engineering. It allows data to be transported through many different systems without corruption.

To translate the entire request message into Unicode, select **Unicode Encode Request** and click **Apply**.

To translate the entire request message from Unicode into ISO-Latin, select **Unicode Decode Request** and click **Apply**.

Create MultiPart Post

The POST method is used to request that the origin server accept the entity enclosed in the request as a new subordinate of the resource identified by the Request-URI in the Request-Line. You can attempt to upload data by manipulating a POST request message.

To insert data from a file:

1. Select **Create MultiPart Post** from the **Action** list on the Request pane.
2. In the text box to the right of the **Action** list, type the full path to a file
   - or -
   Click the Open Folder icon and select the file you want to insert.
3. Click **Apply**.

Remove MultiPart Post

To remove a file that is part of a multipart request, select **Remove MultiPart Post** from the **Action** list on the Request pane.

Response Actions

The area immediately below the tabs on the Response Viewer pane contains three controls:

- a **Chunked** button
- a **Content Coding** drop-down list
- a button that launches the *Find In Response* window, allowing you to search the response for the text string you specify.

Chunked

If a server starts sending a response before knowing its total length, it might break the complete response into smaller chunks and send them in series. Such a response contains the "Transfer-Encoding: chunked" header. A chunked message body contains a series of chunks, followed by a line with "0" (zero), followed by optional footers and a blank line. Each chunk consists of two parts:

- A line with the size of the chunk data, in hex, possibly followed by a semicolon and extra parameters you can ignore (none are currently standard), and ending with CRLF.
- The data itself, followed by CRLF.

Content Codings

If the HTTP response contains compressed data, you can decompress the data using one of the options from the list.

- GZIP—A compression utility written for the GNU project.
- Deflate—The “zlib” format defined in RFC 1950 [31] in combination with the “deflate” compression mechanism described in RFC 1951 [29].

Editing and Sending Requests

To edit and send a request.

1. Modify the request message in the Request Viewer pane.

   To change certain features of the request, select an item from the **Action** list and click **Apply**.

2. Click **Send** to send the HTTP request message.

   The Response Viewer pane displays the HTTP response message when it is received.

3. To view the response as rendered in a browser, click the **Browser** tab.

4. You can prepare your next HTTP request using the HTML or JavaScript controls rendered on the **Browser** tab. To use this feature, you must select the **Interactive Navigation** option (click **Edit** → **Settings**).

5. To save a request, select **File** → **Save Requests**.

Searching for Text

To search for text in the request or response

1. Click  in either the Request Viewer or Response Viewer pane.

2. Using either the *Find in Request* or *Find in Response* window, type or select a string or regular expression.

3. If using a regular expression as the search string, select the **Regex** check box.

4. Click **Find**.

HTTP Editor Settings

To modify the HTTP Editor settings:

1. Click **Edit** → **Settings**.

2. Select either the **Options**, **Authentication**, or **Proxy** category and enter the settings described in the following sections.

3. Click **OK**.

Options

Send As Is

If you select this option, the HTTP Editor will not modify the request, regardless of any other settings you may select. This allows you to send a purposely malformed message. Authentication and proxy settings are disabled when using this option.

Note: You may manually edit the request to go through a proxy, but most standard HTTP proxy servers cannot process non-compliant HTTP requests.

Manipulate Request

If you select this option, the HTTP Editor will modify requests to accommodate the following parameters:

- **Apply State** — If your application uses cookies, URL rewriting, or post data techniques to maintain state within a session, the HTTP Editor will attempt to identify the method and modify the response accordingly.
- **Apply Proxy** — If you select this option, the HTTP Editor will modify the request according to the proxy settings you specify.
- **Apply Filter** — This option appears only when you invoke the HTTP Editor while using WebInspect and a scan tab has focus (that is, after opening or while conducting a scan). If this option is selected, the HTTP Editor applies the Filters settings from WebInspect's Current Scan Settings to add search-and-replace rules for HTTP requests and responses. Note that changing the Current Scan Settings before invoking the HTTP Editor has no effect; the HTTP Editor will use the settings that were in effect when the scan began.
- **Apply Header** — This option appears only when you invoke the HTTP Editor while using WebInspect and a scan tab has focus (that is, after opening or while conducting a scan). If this option is selected, the HTTP Editor applies the Cookies/Headers settings from WebInspect's Current Scan Settings for HTTP requests. Note that changing the Current Scan Settings before invoking the HTTP Editor has no effect; the HTTP Editor will use the settings that were in effect when the scan began.

Enable Active Content

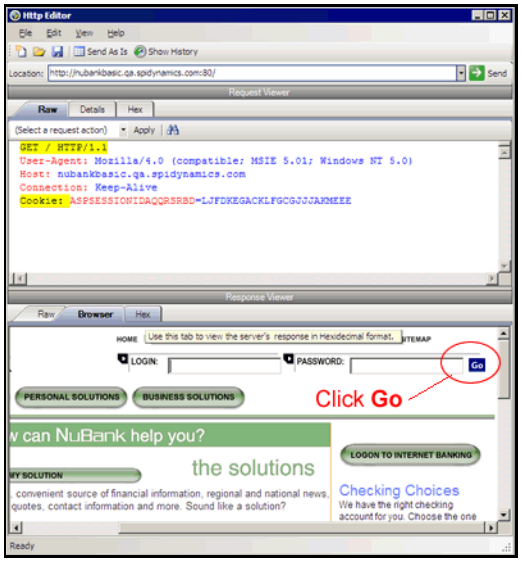
Select this option to allow execution of JavaScript and other dynamic content in all browser windows.

Navigation

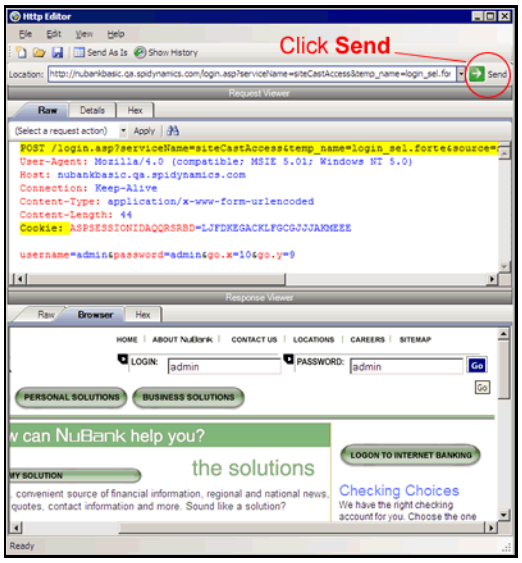
In the **Navigation** group, select either **None**, **Interactive**, or **Browser Mode**.

You can view the server's response as rendered in a browser by selecting the **Browser** tab in the Response Viewer (the lower pane). If the **Interactive** feature is enabled, you can prepare your next HTTP request using the HTML or JavaScript controls rendered in the browser.

For example, using the logon page at nubankbasic.qa.spidynamics.com (shown below), you could enter a user name (“admin”) and password (“admin”), and then click **Go**.



The HTTP Editor formats the request (which uses the POST method to the login1.asp resource) and displays it in the Request Viewer, as illustrated below. You could then edit the logon message (if required) or simply send it to the server by clicking **Send**.



If you select the **Browser Mode** option, then Interactive mode is enabled, but the HTTP Editor will send the request immediately, without first placing it in the Request Viewer and allowing you to edit it.

Advanced HTTP Parsing

Most web pages contain information that tells the browser what character set to use. This is accomplished by using the Content-Type response header (or a META tag with an HTTP-EQUIV attribute) in the HEAD section of the HTML document. For pages that do not announce their character set, you can specify which character set the HTTP Editor should use.

Authentication

If authentication is required, select a type from the **Authentication** list. See Authentication Types on page 44 for a description of the available authentication types.

After selecting an authentication method, enter a user name and password. To avoid typographic errors, re-enter the password in the **Confirm Password** box.

Proxy

Use these settings to access the HTTP Editor through a proxy server.

Direct Connection (proxy disabled)

Select this option if you are not using a proxy server.

Auto detect proxy settings

Use the Web Proxy Autodiscovery (WPAD) protocol to locate a proxy autoconfig file and configure the browser's web proxy settings.

Use Internet Explorer proxy settings

Import your proxy server information from Internet Explorer.

Use Firefox proxy settings

Import your proxy server information from Firefox.

Configure a proxy using a PAC file

Load proxy settings from the Proxy Automatic Configuration (PAC) file in the location you specify in the **URL** box.

Explicitly configure proxy

Configure a proxy by entering the requested information.

1. In the **Server** box, type the URL or IP address of your proxy server, followed (in the **Port** box) by the port number (for example, 8080).
2. Select a protocol for handling TCP traffic through a proxy server: SOCKS4, SOCKS5, or standard.
3. If authentication is required, select a type from the **Authentication** list. See Authentication Types on page 44 for a description of the available authentication types.
4. If your proxy server requires authentication, enter the qualifying user name and password.
5. If you do not need to use a proxy server to access certain IP addresses (such as internal testing sites), enter the addresses or URLs in the **Bypass Proxy For** box. Use commas to separate entries.

HTTPS Proxy Settings

For proxy servers accepting HTTPS connections, select **Specify Alternative Proxy for HTTPS** and provide the requested information.

8 Log Viewer (WebInspect Only)

About the Log Viewer Tool

Use the Log Viewer to inspect the various logs maintained by WebInspect. This feature is used mainly by the HP Product Support group to investigate reported incidents.

Viewing Logs

To view a log:

1. Click **Tools** → **Log Viewer**.

   If you open the Log Viewer when a tab containing a scan has the focus, the program assumes you want to view logs for that scan. Go to step 4.

2. Click **Open Scan**.
3. On the *Open Scan* window, select the scan whose logs you want to view and click **Open**. To open scans in a different database, click **Change Database**.
4. Select a log from the **Log Type** list. The available types depend on the logging level that was selected for the scan (in WebInspect's Application settings). They include:
5. To locate text within the log, click **Find** on the toolbar or select **Edit** → **Find**.
6. To view logs that are not related to a specific scan, click **WebInspect Logs** (on the toolbar).

9 Policy Manager (WebInspect Only)

About the Policy Manager Tool

A policy is a collection of audit engines and attack agents that WebInspect uses when auditing or crawling your Web application. Each component has a specific task, such as testing for susceptibility to cross-site scripting, building the site tree, probing for known server vulnerabilities, etc. These components are organized into the following groups:

- Audit Engines
- General Application Testing
- General Text Searching
- Third-Party Web Applications
- Web Frameworks/Languages
- Web Servers
- Web Site Discovery
- Custom Checks

All these components (except for the Audit Engines) are known collectively as attack groups. Each attack group contains subgroups of individual modules (called attack agents) that check your website for vulnerabilities.

WebInspect contains several prepackaged policies designed to accommodate the majority of users. All policies contain all possible audit engines and agents, but each policy has a different subset of these components enabled. You edit a policy by enabling or disabling audit engines and/or individual attack agents (or groups of agents). You create a policy by editing an existing policy and saving it with a new name.

Views

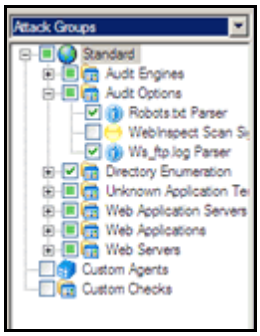
The Policy Manager has two different views, selectable from the **View** menu or by clicking icons on the toolbar. They are:

- Standard
- Search

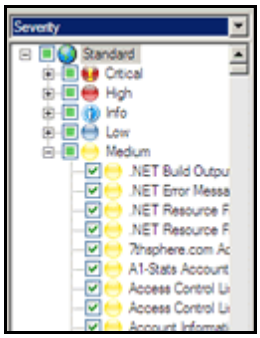
Standard View

This view displays, by default, a list of checks categorized by threat class (according to classifications established by the Web Application Security Consortium). Alternatively, a drop-down list allows you to display all attack agents by severity, or a list of audit engines and attack groups.

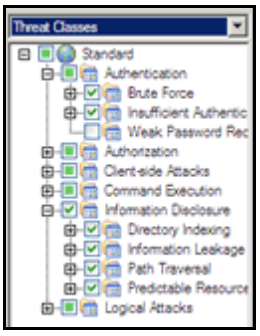
You enable or disable a component by selecting or clearing its associated check box.



Attack Groups



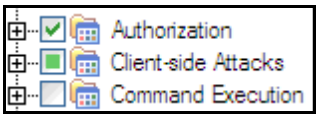
Severity



Threat Classes

The check box next to an unexpanded node indicates the “selected” status of the objects within the node.

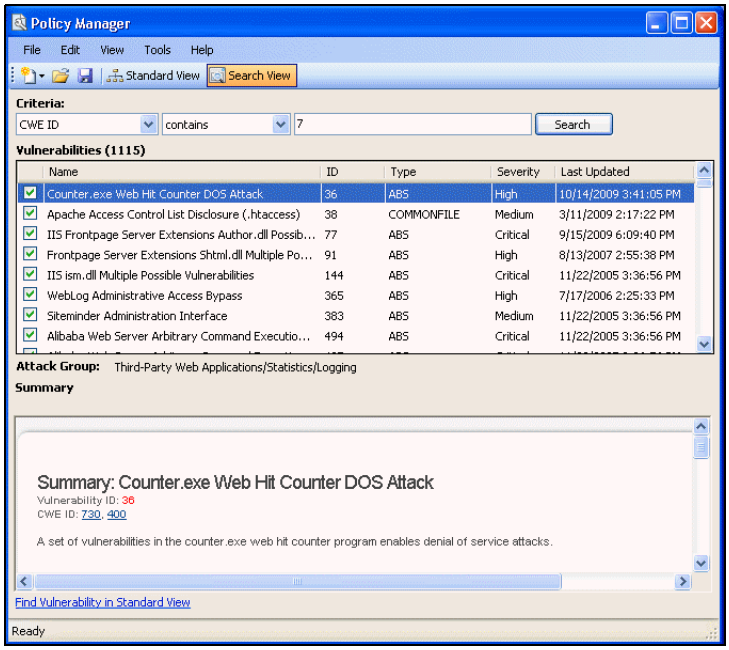
- A check means all objects are selected.
- A green square means some objects are selected.
- An empty box means no objects are selected.



Click the plus sign ⊞ to expand a node.

Search View

The Search view allows you to locate attack agents containing the text you specify in a selected report field (i.e., summary, implication, execution, recommendation, and fix).



This feature is used most often to identify checks that you want to disable. For example, if you are scanning an application that does not contain PHP scripting, you could search summary fields for "PHP." When the Policy Manager lists the attack agents that match your search criteria, you could disable an agent by clearing its associated check box. Then, you can either save the modified policy (making the policy changes permanent) or simply apply the modified policy to the current scan.

Creating or Editing a Policy

WebInspect contains a number of prepackaged policies designed to accommodate the majority of users. You cannot permanently change these policies. However, you may open any of them as a template, modify their contents to create a custom policy, and save the customized policy under a new name. A custom policy may be edited and saved without changing its name.

To edit a policy:

1. On the toolbar, click **Policy Manager**
   - or -
   select **Tools** → **Policy Manager**.

   The Policy Manager opens and loads, by default, the Standard policy.
2. To edit a policy that you previously created (i.e., a custom policy), select **File** → **Open** and select the policy.
3. To create a policy based on a prepackaged policy, select **File** → **New** (or click the New Policy icon) and select the policy on which the new one will be modeled.
4. Disable (or enable) an attack group by clearing (or selecting) its associated check box. To disable or enable an individual agent within a group, first expand the group and then edit its check box.
5. To rename an attack group:
   a. Right-click the attack group.
   b. Choose **Rename** from the shortcut menu.
6. To add an attack group:
   a. Right-click any existing attack group and choose **New Attack Group** from the shortcut menu. A highlighted entry named New Attack Group appears.
   b. Right-click the new group and choose **Rename**.
   c. Populate the group by dragging and dropping attack agents onto it.
7. You may also create a custom check. See Creating a Custom Check on page 66 for more information.
8. If you select the **Auto Update** check box, WebInspect determines if any updated or new attack agents downloaded from the HP database should be enabled or disabled, based on the analysis of its sibling agents. For example, if you disable attack agents targeting Microsoft's Internet Information Server (IIS), and you select **Auto Update**, then WebInspect will not enable any IIS-related attack agent that it downloads to your system. Conversely, any new or updated attack agents that are related to agents that are enabled in your policy will also be enabled.
9. Select **File** → **Save As**. Type a name for your custom policy in the **File name** box and then click **Save** to save the new policy in WebInspect's *.policy format. You cannot save a policy using the name of a default WebInspect policy (Assault, Blank, Standard, etc.).

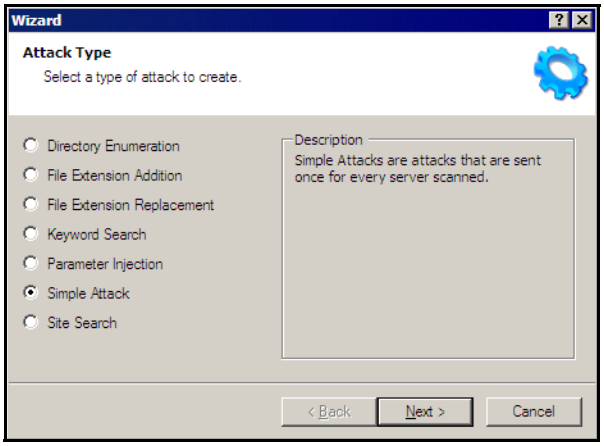
Creating a Custom Check

Although WebInspect rigorously inspects your entire website for real and potential security vulnerabilities, you may require a custom check to detect vulnerabilities that are unique to your application.

To create a custom check:

1. On the WebInspect toolbar, click **Policy Manager**
   - or -
   click **Tools** → **Policy Manager**.

   The Policy Manager opens and loads, by default, the Standard policy.
2. To edit a policy that you previously created, select **File** → **Open** and select the policy.
3. To create a policy based on one of the prepackaged policies, select **File** → **New** (or click the New Policy icon) and select the policy on which the new one will be modeled.
4. Make sure the Standard view is selected, with attack groups listed in the left pane.
5. Right-click on **Custom Checks** and select **New Custom Check** from the shortcut menu.
6. When the Custom Check Wizard appears, select an attack type.



The attack types are listed below. See step 8 on page 69 and step 9 on page 70 for entering attack and signature information.

- **Directory enumeration**

  This type of check searches for a directory of the name you specify.

Attack Type:	Directory Enumeration
Attack:	/directory_name/ [where directory_name is the name of the directory you want to find]
Signature:	[STATUSCODE]3\d\d OR [STATUSCODE]2\d\d OR [STATUSCODE]40[13]

- **File extension addition**

  This type of check searches for files with a file extension that you specify.

  During the crawl, whenever WebInspect encounters a file of any name and any extension (for example, global.asa), it sends an HTTP request for a file of the same name plus the found extension plus an extension that you specify. For example, if you specify a file extension of .backup, then when WebInspect discovers a file named global.asa, it will subsequently search for a file named global.asa.backup.

A server would normally deny any request for the global.asa file, but if a programmer has left a backup file on the server and the file has a different extension (such as global.asa.backup), then the server might return the file (which contains the full source of the global.asa file).

To create a custom check that searches for files with a specific added extension, enter the following in the Custom Check Wizard:

Attack Type:	File Extension Addition
Attack:	.ext [where ext is the file extension of files you want to locate]. You must include the leading dot or period (.)
Signature:	[STATUSCODE]200 AND ([HEADERS]Content-Type:\stext/plain OR [HEADERS]Content-Type:\sapplication/octet-stream)

- **File extension replacement**

  This type of check searches for files with a file extension that you specify.

  For example, WebInspect contains a standard check that searches for files having an extension of "old." During the crawl, whenever it encounters a file of any name and any extension (for example, startup.asp), it sends an HTTP request for a file of the same name but with an extension of "old" (for example, startup.old).

  To create a custom check that searches for files with a specific extension, enter the following in the Custom Check Wizard:

Attack Type:	File Extension Replacement
Attack:	ext [where ext is the file extension of files you want to locate]. Do NOT include a leading dot or period (.)
Signature:	[STATUSCODE]200 AND ([HEADERS]Content-Type:\stext/plain OR [HEADERS]Coontent-Type:\sapplication/octet-stream)

- **Keyword search**

  This type of check determines if a specified word or phrase (defined by a regular expression) exists anywhere in the HTTP response.

  The following example searches the HTTP response for a nine-digit number formatted as a social security number (\d = any digit).

Attack Type:	Keyword Search
Attack:	N/A
Signature:	[BODY]\d\d\d-\d\d-\d\d\d\d

- **Parameter injection**

  This type of attack replaces an argument value with an attack string.

  Example:

  http://www.samplesite.com/webapp.asp?ValidParameter=ValidArgument

  will be changed to

  http://www.samplesite.com/webapp.asp?ValidParameter=AttackArgument

There are several variations.

– Command Execution

  A command execution check combines strings composed of special characters with operating system-level commands. It is an attempt to make the web application execute the command using the provided string (if the application fails to check for and prohibit the input).

  The following example tests for parameter injection by providing spurious input to a program named support_page.cgi; if the HTTP response contains data that matches the regular expression, then the application is vulnerable to command execution.

  Attack Type: Parameter Injection

  Attack: /support_page.cgi?file_name=|id|

  Signature: [BODY]uid= AND [BODY]gid=

– SQL Injection

  SQL injection is the act of passing SQL code into an application. These attack strings are composed of fragments of SQL syntax that will be executed on the database server if the web application uses the string when forming a SQL statement without first filtering out certain characters.

  Attack Type: Parameter Injection

  Attack: ’ [an apostrophe]

  Signature: [[STATUSCODE]5\d\d

– Cross-Site Scripting

  This issue occurs when dynamically generated web pages display input that is not properly validated. This allows an attacker to embed malicious JavaScript into the generated page, enabling him to execute the script on the machine of any user who views the malicious page. Any site that allows users to post text messages can be vulnerable to an attack such as this.

  The following example tests for cross-site scripting in the Fusion News application:

  Attack Type: Parameter Injection

  Attack: /fullnews.php?id=<script>alert(document.cookie)</script>

  Signature: [ALL]Powered\sby\sFusion\sNews And
  [ALL]<script>alert\(document\.cookie\)</script>

– Directory Traversal

  Directory traversal entails sending malformed URL strings to access non-public portions of the web server’s content. An attacker will try to access different files on a server by using relative hyperlinks. For example, by adding triplets of two periods and a forward slash (../) to the target URL and by varying the number of directories to traverse, an attacker might find and gain access to a system password file such as www.server.com/../../../../password.

  The following example searches for the boot.ini file:

  Attack Type: Parameter Injection

  Attack: /../../../../../../../../../../../boot.ini

  Signature: [ALL]\[boot\sloader\]

– Abnormal Input

  Abnormal input attack strings are composed of characters that can cause unhandled exceptions (errors the program is not coded to handle) in web applications where unexpected input is not prohibited. Unhandled exceptions often cause servers to display error messages that disclose sensitive information about the application's internal mechanics. Source code may even be disclosed.

  The following example sends an extraordinarily long string in an attempt to create a buffer overflow.

Attack Type:	Parameter Injection
Attack:	AAAAA...AAAAA [1000 repetitions of the letter "A"]
Signature:	[STATUSCODE]5\d\d

- **Simple attack**

  This type of attack is sent once for every server scanned.

  The following example attempts to obtain a UNIX password file by appending the attack string to the target URL or IP address:

Attack Type:	Simple Attack
Attack:	/etc/passwd
Signature:	[ALL]root: AND [ALL]:0:0

- **Site search**

  This type of attack is designed to find files commonly left on a web server. For example, WebInspect check ID #279 searches for a file named log.htm.

  The following example searches for a file named xanadu.html by appending the attack string to the target URL or IP address:

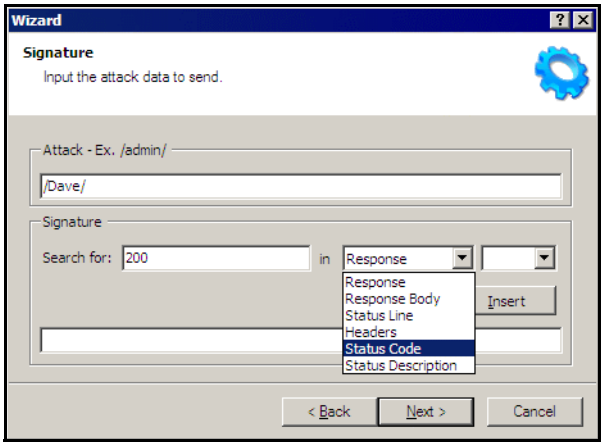
Attack Type:	Site Search
Attack:	xanadu.html
Signature:	[STATUSCODE]2\d\d OR [STATUSCODE]40[1]

  To create a custom check that searches for a file named confidential.txt, enter the following in the Custom Check Wizard:

Attack Type:	Site Search
Attack:	confidential.txt
Signature:	[STATUSCODE]2\d\d AND ([HEADERS]Content-Type:\stext/plain OR [HEADERS]Content-Type:\sapplication/octet-stream)

7 Click **Next**.

8 In the **Attack** box, enter the data you want to use for the attack. In the following example of directory enumeration, the check will search for a directory named "Dave" by appending the attack string (/Dave/) to the target URL or IP address.



9 You must specify a signature, which is simply a regular expression (i.e., a special text string for describing a search pattern). When WebInspect searches the HTTP response and finds the text described by the signature, WebInspect flags the session as a vulnerability. You can use the **Search for** box and drop-down lists to help you create the regular expression, or you can type the regular expression directly into the text box at the bottom of the window.

To use the **Search for** box:

a Enter the text you want to locate.

Enter only text; do not enter a regular expression.

b In this example (searching for a directory named "Dave"), the server would return a status code of 200 if the directory exists, so enter "200" in the **Search for** box. Realistically, however, you might also accept any status code in the 200 or 300 series, or a status code of 401 or 403.

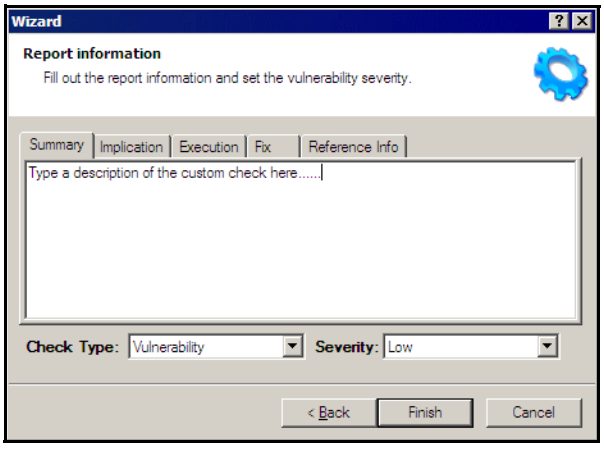
c Click the drop-down arrow to specify the section of the HTTP response that should be searched.

d (optional) To create a complex search, click the second drop-down and select a Boolean operator (AND, OR, or NOT).

e Click **Insert**.

f (optional) For complex searches, repeat step a through step d as needed. You can also edit or replace the regular expression that appears in the bottom text box.

10 Click **Next**.



11 On the Report Information panel, click each tab and enter the text that will appear in the vulnerability description.

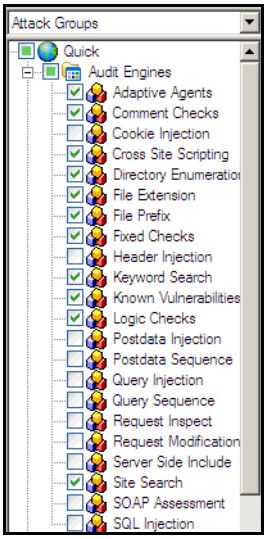
12 Select an entry from the **Check Type** list.

13 Select a severity level from the **Severity** list.

14 Click **Finish**.

15 Change the default name “New Custom Check” to reflect the purpose of the check.

16 Click ⊞ to expand the Audit Engines folder.



17 Ensure that the appropriate audit engine is enabled (with a check mark) for the type of check you created, according to the following table:

Correlation of Attack Type to Audit Engine

This Attack Type…	Uses this Audit Engine…
Simple Attack	Fixed Checks
Parameter Injection	Post Data Injection
Site Search	Site Search
File Extension Replacement	File Extension
File Extension Addition	File Extension
Directory Enumeration	Directory Enumeration
Keyword Search	Keyword Search

18 Click **File** → **Save**.

19 Enter a name for the new policy and click **Save**.

WebInspect adds all custom checks to every policy, but does not enable them. To enable the custom check in other polices, see Creating or Editing a Policy on page 65.

Disabling a Custom Check

To disable a custom check:

1 Select a custom check.

2 Clear its associated check box.

Deleting a Custom Check

To delete a custom check:

1 Right-click a custom check.

2 Select **Delete** from the short-cut menu.

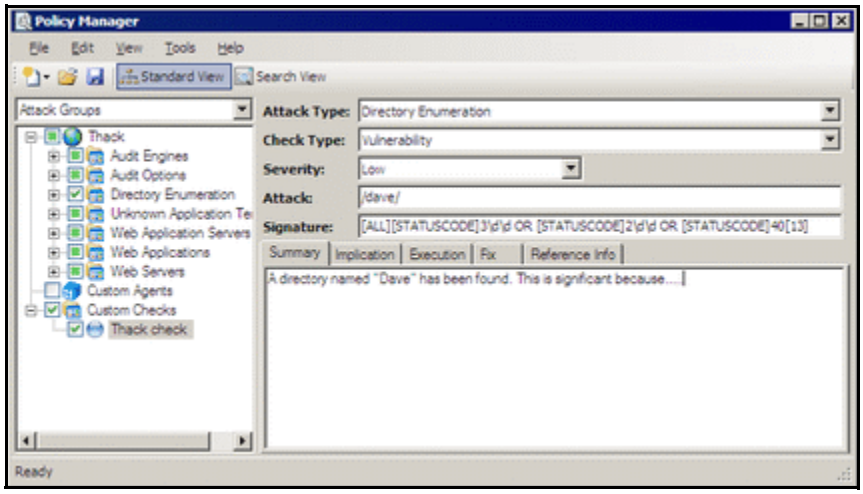
Editing a Custom Check

To edit a custom check:

1 Open a policy.

2 Select a custom check.

3 Using the right pane of the Policy Editor, modify the custom check properties.



4 Click the Save icon.

Searching for Attack Agents

Use the Search view on the Policy Manager to locate specific vulnerability checks (attack agents). You can then elect to include or exclude individual agents.

To search for attack agents:

1 On the toolbar, click **Policy Manager**
- or -
click **Tools** → **Policy Manager**.

2 If you do not have a policy selected, choose a policy from the *Open Policy* window and click **OK**.

3 Click **View** → **Search**.

4 From the **Criteria** list, select the property that you want to search.

The description of every attack agent contains “report fields” such as summary, implication, execution, fix, and reference information. The Search feature allows you to locate attack agents that contain the text you specify in a selected report field. In addition, you can search for a vulnerability ID, vulnerability name, engine type, or the date when last updated.

5 Choose an operator from the drop-down list (is, is greater than, is less than, contains).

6 In the text box, type the text or number you want to find.

7 Click **Search**.

The Policy Manager lists in the **Checks** area all attack agents that match your search criteria. An active agent will have a check mark next to its name. Select (or clear) a check box to activate (or deactivate) an agent.

8 Click **Save** to save the revised policy.

Policy Manager Icons

The following table illustrates and describes icons that are used in the Policy Manager tree view.

Policy Manager Icons

Icon	Definition
	The policy.
	Attack Group Folder: Contains vulnerability assessments.
	Audit Methodology: A set of checks that compose an audit methodology. For example, Site Search is part of the Audit methodology.
	A critical vulnerability wherein an attacker might have the ability to execute commands on the server or retrieve and modify private information.
	A high vulnerability. Generally, the ability to view source code, files out of the Web root, and sensitive error messages.
	A medium vulnerability. Indicates non-HTML errors or issues that could be sensitive.
	A low vulnerability. Indicates interesting issues, or issues that could potentially become higher ones.

10 Regular Expression Editor

About the Regular Expression Editor Tool

A regular expression is a pattern that describes a set of strings. Regular expressions are constructed similarly to mathematical expressions by using various operators to combine smaller expressions. Only advanced users with a working knowledge of regular expressions should use this feature.

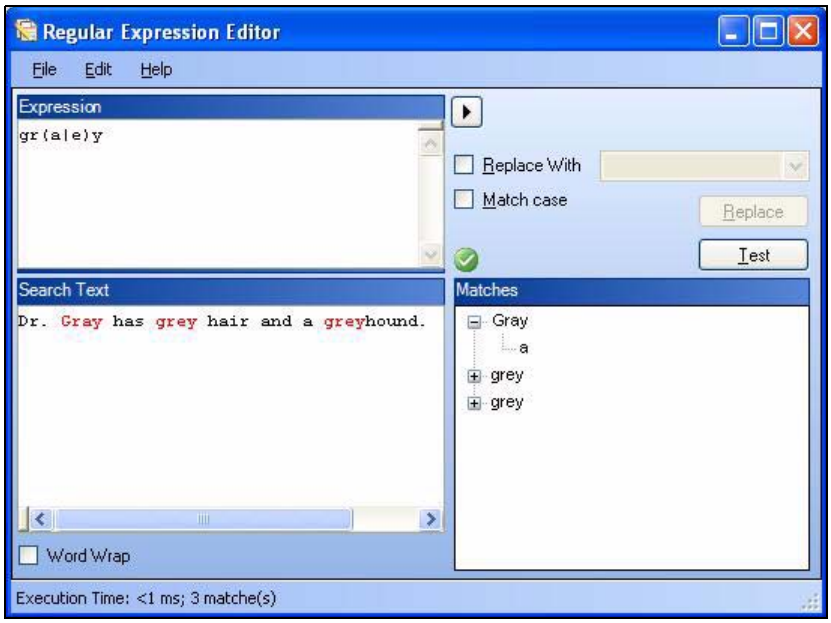
Testing a Regular Expression

Use the Regular Expression Editor to verify regular expressions.

To use the Regular Expression Editor:

1 Click **Tools** → **Regular Expression Editor**.

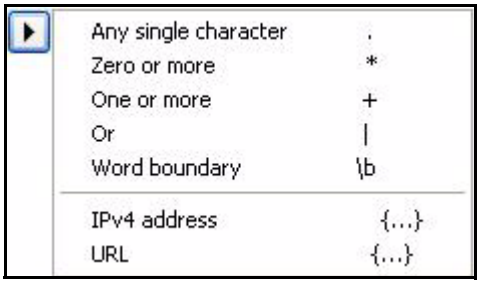
The *Regular Expression Editor* window opens.



2 In the **Expression** box, type or paste a regular expression that you believe will find the text for which you are searching.

For assistance, click  to reveal a list of objects. These include metacharacters and regular expressions that define a URL and an IP address. Click an object to insert it.

Note: You can also use special Regular Expression Extensions to restrict your search to certain areas of an HTTP message.



The Regular Expression Editor examines the syntax of the entered expression and displays (if valid) or (if invalid).

3 In the **Search Text** box, type (or paste) the text through which you want to search.

Alternatively, you can load an HTTP request or response message that you previously saved using the HTTP Editor. To do so:

a Click **File** → **Open Request**.

The Request file is actually a session containing data for both the HTTP request and response.

b Using the standard file-selection window, choose the file containing the saved session.

c Select either **Request** or **Response**.

d Click **OK**.

4 To find only those occurrences matching the case of the expression, select the **Match Case** check box.

5 If you want to substitute the string identified by the regular expression with a different string:

a Select the **Replace With** check box.

b Type or select a string using the drop-down combo box.

6 Click **Test** to search the target text for strings that match the regular expression. Matches will be highlighted in red.

7 If you selected the **Replace** option, click **Replace** to substitute all found strings with the replacement string.

Regular Expressions

Special characters and sequences are used in writing patterns for regular expressions. The following table describes some of these characters and includes short examples showing how the characters are used.

Characters Used in Regular Expressions

Character	Description
\	Marks the next character as special. /n/ matches the character “n”. The sequence /\n/ matches a linefeed or newline character.
^	Matches the beginning of input or line. Also used with character classes as a negation character. For example, to exclude everything in the content directory except /content/en and /content/ca, use: /content/([^ec].*|e[^n].*|c[^a].*|.{3,})[/][./]* Also see \S \D \W.
$	Matches the end of input or line.
*	Matches the preceding character zero or more times. /zo*/ matches either “z” or “zoo.”
+	Matches the preceding character one or more times. /zo+/ matches “zoo” but not “z.”
?	Matches the preceding character zero or one time. /a?ve?/ matches the “ve” in “never.”
.	Matches any single character except a newline character.
[xyz]	A character set. Matches any one of the enclosed characters. /[abc]/ matches the “a” in “plain.”
\b	Matches a word boundary, such as a space. /ea*r\b/ matches the “er” in “never early.”
\B	Matches a nonword boundary. /ea*r\B/ matches the “ear” in “never early.”
\d	Matches a digit character. Equivalent to [0-9].
\D	Matches a nondigit character. Equivalent to [^0-9].
\f	Matches a form-feed character.
\n	Matches a linefeed character.
\r	Matches a carriage return character.
\s	Matches any white space including space, tab, form-feed, and so on. Equivalent to [\f\n\r\t\v]
\S	Matches any nonwhite space character. Equivalent to [^ \f\n\r\t\v]
\w	Matches any word character including underscore. Equivalent to [A-Za-z0-9_].
\W	Matches any nonword character. Equivalent to [^A-Za-z0-9_].

Regular Expression Extensions

Hewlett-Packard engineers have developed and implemented extensions to the normal regular expression syntax. When building a regular expression, you can use the following tags and operators:

Regular Expression Tags

- [BODY]
- [STATUSCODE]
- [STATUSDESCRIPTION]
- [STATUSLINE]
- [HEADERS]
- [ALL]
- [COOKIES]
- [SETCOOKIES]
- [METHOD]
- [REQUESTLINE]
- [VERSION]
- [POSTDATA]
- [URI]
- [TEXT]

Regular Expression Operators

- AND
- OR
- NOT
- [ ]
- ( )

Examples

To detect a response in which (a) the status line contains a status code of “200” and (b) the phrase “logged out” appears anywhere in the message body, use the following regular expression:

```
[STATUSCODE]200 AND [BODY]logged\sout
```

To detect a response indicating that the requested resource resides temporarily under a different URI (redirection) and having a reference to the path “/Login.asp” anywhere in the response, use the following:

```
[STATUSCODE]302 AND [ALL]Login.asp
```

To detect a response containing either (a) a status code of “200” and the phrase “logged out” or “session expired” anywhere in the body, or (b) a status code of “302” and a reference to the path “/Login.asp” anywhere in the response, use the following regular expression:

```
( [STATUSCODE]200 AND [BODY]logged\sout OR [BODY]session\sexpired ) OR
( [STATUSCODE]302 AND [ALL]Login.asp )
```

Note that you must include a space (ASCII 32) before and after an “open” or “close” parenthesis; otherwise, the parenthesis will be erroneously considered as part of the regular expression.

To detect a redirection response where “login.aspx” appears anywhere in the redirection Location header, use the following regular expression:

[STATUSCODE]302 AND [HEADERS]Location:\slogin.aspx

To detect a response containing a specific string (such as “Please Authenticate”) in the Reason-Phrase portion of the status line, use the following regular expression:

[STATUSDESCRIPTION]Please\sAuthenticate

11 Report Designer (WebInspect Only)

About the Report Designer Tool

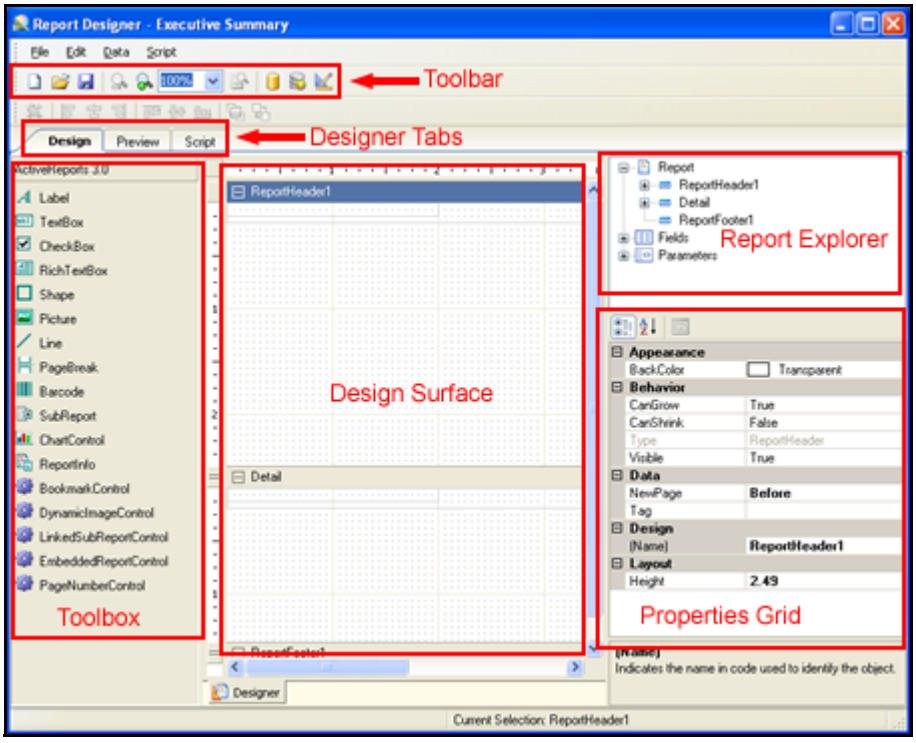
The Report Designer is an HP integration of the ActiveReports® 3.0 report designer developed by Grape City - Data Dynamics. It provides the ability to create and modify reports.

For complete information on using ActiveReports, refer to the ActiveReports User Guide and Class Library.

User Interface

The Report Designer contains six main components, as depicted in the following illustration:

- Toolbar
- Designer Tabs
- Toolbox
- Design Surface
- Report Explorer
- Properties Grid



Toolbar

The Report Designer toolbar is illustrated below.

Report Designer Toolbar

Icon	Function	Description
	New	Opens the *Create Report Definition* window, allowing you to select the queries to be included in the report.
	Open	Opens the *Open a Report* window, allowing you to select a report or subreport for editing.
	Save	Saves the open report.
	Zoom In	Increases the magnification of the design surface at 50 percent increments.
	Zoom Out	Decreases the magnification of the design surface at 50 percent increments.
	Magnification Percentage	Allows you to select a magnification setting for the design surface.
	Actual Size	Returns the magnification of the design surface to 100 percent.
	Set Data Source	Allows you to specify the scan that will provide the data.
	Set Custom Data Source	Allows you to specify a custom data source.
	Parameter Designer	Opens the Parameter Designer tool.

Menus

The Report Designer contains the following menus:

Report Designer Menus

Menu	Command	Description
File	New	Opens the *Create Report Definition* window, allowing you to select a definition for a new report.
	Open	Opens the *Open a Report* dialog, allowing you to select a report for editing.
	Save	Saves the open report.
	Save As	Saves the open report to a file you specify.
	Export	Saves the report in a format you specify.
	Enable Console Output	If enabled, WebInspect presents a pane (at the bottom of the window) that displays the status of each report page being generated. If a problem is encountered, this pane displays an exception message and stack trace. This pane is also visible on the Preview tab of the Report Designer.
	Exit	Terminates the Report Designer.
Edit	Parameter Designer	Opens the Parameter Designer tool.
	Modify/Create Report	Opens the *Modify Report Definition* dialog, allowing you to change the report definition.
	Delete	Deletes the selected object.
	Cut	Deletes the selected object and saves it to the clipboard.
	Copy	Copies the selected object to the clipboard.
	Paste	Inserts the contents of the clipboard.
	Undo	Reverses the last operation performed.
	Redo	Reverses the last Undo operation.
Data	Set Scan and Report Inputs	Allows you to select a scan and specify report parameters.
	Set Custom Data Source	Opens the *Report Data Source* dialog, allowing you to connect to various sources.
	Edit Global Styles	Opens the Report Styles Editor. Use this to create or modify a stylesheet.
	Edit Report Styles	Opens the Report Styles Editor. Use this to create or modify styles for the report on which you are currently working
	Edit Report Settings	Opens the *Report Settings* dialog, allowing you to modify many facets of your report.
Script	Import	Allows you to select a script from the script library to import into the designer.

Report Designer Menus (cont'd)

Menu	Command	Description
	Compile	Compiles the script.
	Find	Opens the **Script** tab and presents the *Find/Replace* dialog, allowing you to search for the text you specify.
	Script Editor	Opens the Script Editor.

Designer Tabs

The Report Designer contains the following three tabs.

Design Tab

By default, when you create or open a report, the Design tab is selected. Use this area to perform all design-time and run-time functions associated with your report, such as creating a layout, binding to data sources, creating event-handling methods, and more.

Script Tab

Selecting the Script tab opens the script editor, which gives you the ability to add scripting to your report. The Script editor allows you to create event-handling methods. In the Report Events tab on the right, there is a combo box where you can select any report section to attach an event-handling method.

Preview Tab

The Preview tab allows you to view what your report looks like at run time with actual scan data. This makes it easy to quickly see the run-time impact of changes you make in the designer or the code-behind. Use the Preview toolbar to navigate the report and add annotations.

Toolbox

The toolbox displays a variety of controls. To add a control, drag it from the toolbox and drop it on the design surface (canvas), where you can modify its size, position, alignment, and properties.

- Barcode — Inserts an ActiveReports Barcode control; can be bound to a database field.
- ChartControl — Inserts a chart in any of a variety of styles.
- Checkbox — Inserts a check box; can be bound to a database field.
- Label — Inserts a new static label control; can be bound to a database field.
- Line — Inserts a line control.
- PageBreak — Inserts a page break within a selection.
- Picture — Inserts an image loaded from a file; can be bound to a database field.
- ReportInfo — Displays report information in a number of format strings such as {PageNumber} of {PageCount}: can be bound to a database field.
- Textbox — Inserts a textbox; can be bound to a database field
- Shape — Inserts a rectangle, circle or square shape.
- Subreport — Inserts a Subreport control to link to another report.
- RichTextBox — Inserts an ActiveReports RichTextBox control; can be bound to a database field.

- BookmarkControl — Inserts a hyperlink in the table of contents; clicking the hyperlink navigates to the bookmark.

  Note: Bookmark text can formatted as follows:

  {=MainReportName}\<static-text>\{=<field-name>}

  where

  MainReportName is optional (and doesn't need to appear first)

  \ indicates the beginning of a hierarchical level

  <static-text> is any text you assign to the bookmark

  <field-name> is the name of a bound or calculated field

- DynamicImageControl — Allows you to associate an image selector control with an image (using the Parameter Designer), so the user can select an image a run time. Can be bound to a database field.
- LinkedSubreportControl — Creates a link to the subreport you select. Use the AssociatedFields property to pass values to the subreport.
- EmbeddedReportControl — Allows you to design a subreport "on the fly" (rather than using a LinkedSubreportControl) using the DataTableField property.
- PageNumberControl — Allows you to place a page number in the report (usually in the page footer).

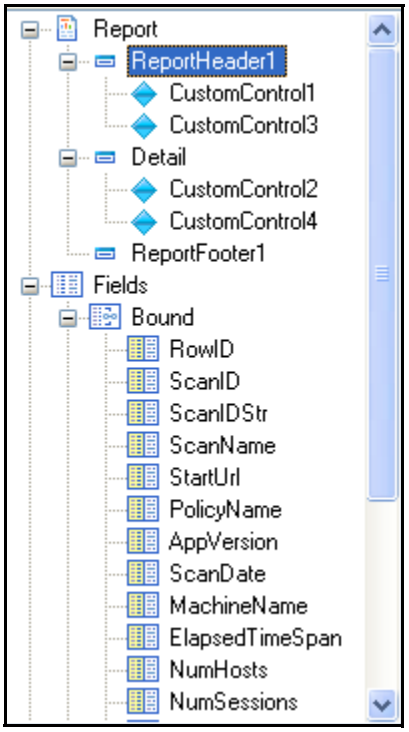
Design Surface

The default design surface contains the following base components:

- PageHeader section--This section can be used to print column headers, page numbers, page titles, or any information that needs to be printed once at the top of each page. Bound controls in the PageHeader or PageFooter are not supported. The data in such controls may not be in synch with the data displayed in other sections on the page.
- Detail section--This section is the body of the report that prints once for each record in the data source. A report's layout may contain only one Detail section.
- PageFooter section--This section can be used to print page totals, page numbers or any other information that needs to be printed once at the bottom of each page.
- Designer/Script/Preview tabs--The Designer and Script tabs can be clicked to toggle between design and script views, while the Preview tab allows for a fully functional design-time preview of how a report will look and behave at run time.

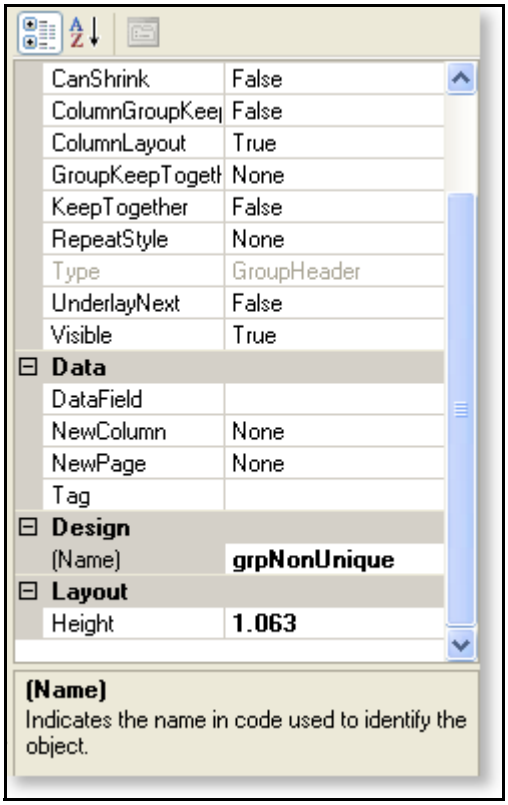
Report Explorer

The Report Explorer serves as the information focal point for your report. From it, you can gain a quick overview of the elements that compose the report, remove individual controls, add parameters and calculated fields, bind data fields to text box controls, and modify properties and report behavior via the Properties grid.



Properties Grid

The Properties Grid allows you to view or modify properties for an object selected on either the Design Surface or the Report Explorer.



Creating a Report

1. Open or create a report definition.

   To create a report definition:

   a. Click **File** → **New** (or click the New icon on the toolbar).

   b. Create a report definition.

   c. Enter a name and (optionally) a brief description for the report.

   d. Select a report context: either **Scan** or **Session**.

      When a scan is open, users can generate a session report by right-clicking a session and selecting **Generate Session Report** from the context menu.

   e. If you want the report name to be included in the list of WebInspect reports, select **Exposed in Product**.

      Typically, you do not select this option if you are creating a subreport.

   f. If you are creating a header/footer template, select **Header/Footer Template**.

   g. Select one or more views from the View Name list. To see the view parameters and fields, click the view name.

   h. Click **OK**.

   To open a report definition:

   a. Click **File** → **Open** (or click the Open icon on the toolbar).

   b. Select a report or subreport.

   c. Click **OK**.

2. Design your report. For complete information on using ActiveReports, refer to the ActiveReports User Guide and Class Library.
3. To modify the script associated with this report, click the **Script** tab.
4. To modify or create parameters associated with this report, click **Edit** →**Parameter Designer**.
5. To modify the styles associated with this report, click **Data** → **Edit Report Styles**.
6. To preview your work:

   a. Click the **Preview** tab.

   b. On the *Generate a Report* dialog, select a scan and click **Next**.

   c. If the report includes parameters, select parameters.

   d. Click **Finish**.

Report Script Editor

Use the Report Script Editor to create or modify scripts maintained in a script library. You can then import these scripts into reports.

All scripts must be written using the C# language.

The Report Script Editor menu bar contains the following menus:

Report Script Editor Menus

Menu	Command	Description
File	Save	Save the script to a library.
	Refresh	Redisplay the script.
	Exit	Terminate the Script Editor.
Edit	Find	Open a *Find/Replace* dialog, allowing you to search for and optionally replace text in the script.
Script	Import	Incorporate a script library into the script you are developing.
	Compile	Compile the script.
Help	Help	Open the Help file.

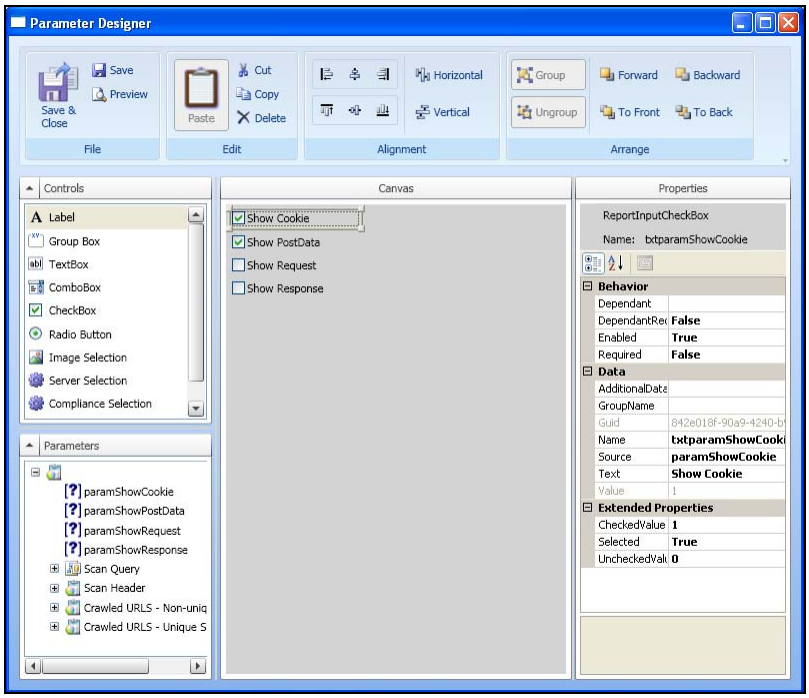
Parameter Designer

Reports have three types of inputs that can be used for filtering data or supplying custom content to reports. They are:

- **Data View parameters (query parameters)**—Data View parameters are used to pass values to the underlying Data View of the report for filtering data. Parameter names begin with @.
- **Report Parameters**—Report parameters are used to pass values entered by the user to the report. These values are then used by the report to alter report behavior or format.
- **Replacements**—Replacements are tokens that exist in the data view. Replacement inputs are used to pass values to these tokens. Replacements are used to change the sort order of a data view or to provide additional criteria to the data view.

Users have the opportunity to provide values for these inputs when generating a report. Before a user can be prompted to enter inputs, however, report designers must specify which inputs will be displayed to the user and how they will be presented. This is accomplished by using the Parameter Designer.

To open the Parameter Designer, from an open report in the Report Designer, click the Parameter Designer icon  on the toolbar or choose **Parameter Designer** from the **Edit** menu.



The Parameter Designer has five areas.

Toolbar

The toolbar provides easy access to all of the functions of the designer:

- **Save and Close**—Saves the current design to the report and closes the parameter designer window.
- **Save**—Saves the current design to the report.
- **Preview**—Opens a window showing what the designed inputs will look like at run time.
- **Cut, Copy, Paste, Delete**—Manipulate controls on the canvas.
- **Alignment**—Align one or more selected controls on the canvas.
- **Group/Ungroup**—A designer can group two or more selected controls on the canvas. When controls are grouped together, they can be moved together on the canvas.
- **Forward**—Bring the selected control forward one layer.
- **Backward**—Send the selected control backward one layer.
- **To Front**—Bring the selected control to the top most layer.
- **To Back**—Send the selected control to the bottom most layer.

Canvas

The canvas is the design area, which constitutes a visual representation of the parameters that are presented at run-time. Controls can be added, modified, and deleted from the canvas.

Properties Grid Pane

This area displays the properties of object(s) selected in the design canvas or the Parameters pane, whichever has the focus.

Controls Toolbox

The Controls toolbox lists the types of controls that may be added to the report. They include, in addition to the standard self-explanatory controls, the following special controls:

- **Server Selection**—A drop-down list of available servers in the selected scan.
- **Compliance Selection**—A list of compliance templates; suitable for compliance reports only.
- **Sort Control**—Allows you to select how you want the report data to be sorted.

To add a control, drag it from the toolbox and drop it on the canvas.

Report Parameters Pane

This pane displays a hierarchical representation of all parameters available to the current report and its subreports. Icons indicate the parameter type.

Query 

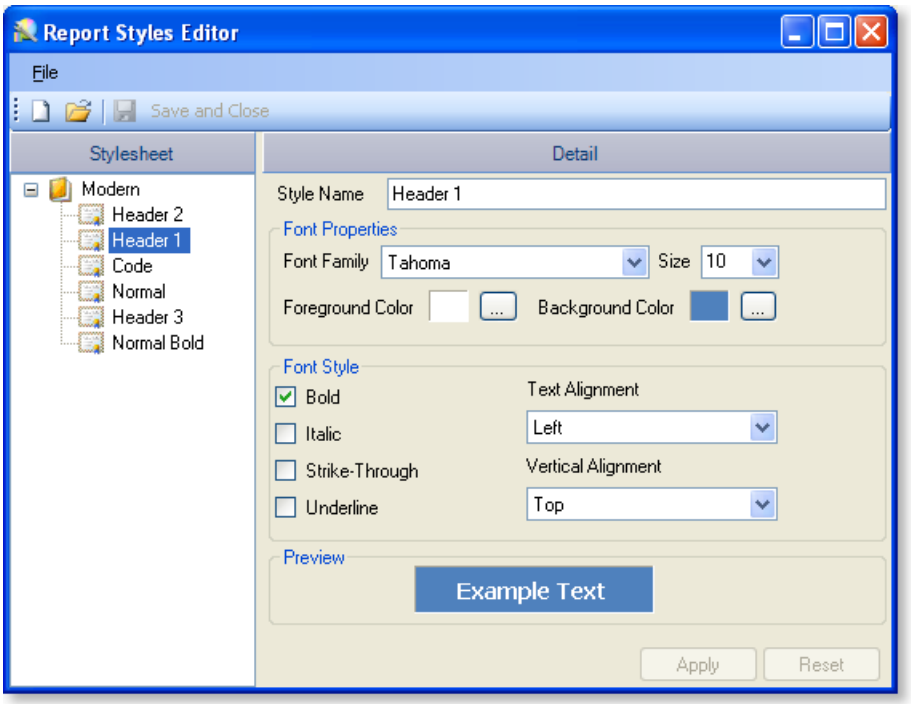
Report 

Replacement 

Report Styles Editor

When creating or modifying a report, the Report Designer uses the stylesheet that is specified as the default. If you want to create or modify styles for the report on which you are currently working, select **Edit Report Styles** from the **Data** menu. New styles will be added to the report; modified styles will override the default definition for this report.

Conversely, if you want to create or modify a stylesheet, select **Edit Global Styles** from the **Data** menu. You can then edit or create stylesheets, and specify the stylesheet that will be initially assigned to all reports as the default.



Report Structure

Report Structure

Report Structure

A report section contains a group of controls that are processed and printed at the same time as a single unit. ActiveReports defines the following section types.

Report Header

A report can have one report header section that prints at the beginning of the report. This section generally is used to print a report title, a summary table, a chart or any information that needs only to appear once at the report's start.

Report Footer

A report can have one report footer section that prints at the end of the report. This section is used to print a summary of the report, grand totals, or any information that needs to print once at the report's end.

Page Header

A report can have one page header section that prints at the top of each page. Unless the page contains a report header section, the page header will be the first section that prints on the page. The page header section is used to print column headers, page numbers, a page title, or any information that needs to appear at the top of each page in the report.

Page Footer

A report can have one page footer section that prints at the bottom of each page. It is used to print page totals, page numbers, or any other information that needs to appear at the bottom of each page.

Group Header/Footer

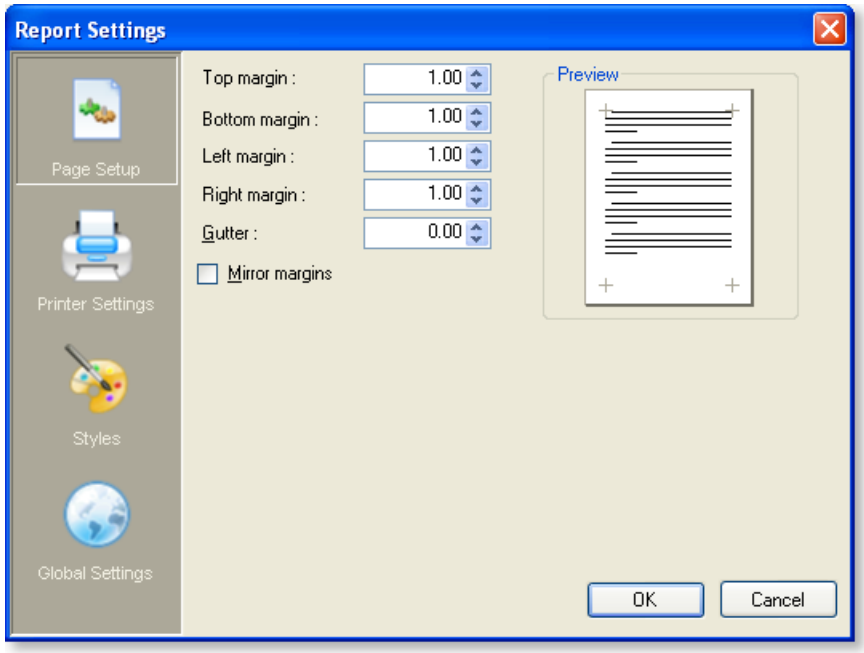
A report can consist of single or multiple nested groups, with each group having its own header and footer sections. The header section is inserted and printed immediately before the detail section. The footer section is inserted and printed immediately after the detail section.

Detail

A report has one detail section. The detail section is, in some cases, the body of the report and one instance of the section is created for each record in the report.

Report Settings

You can modify facets of your report, such as the page setup, printer settings, styles, and global settings of your report at design time. To make changes, access the *Report Settings* dialog by selecting Data > Edit Report Settings.



Charts

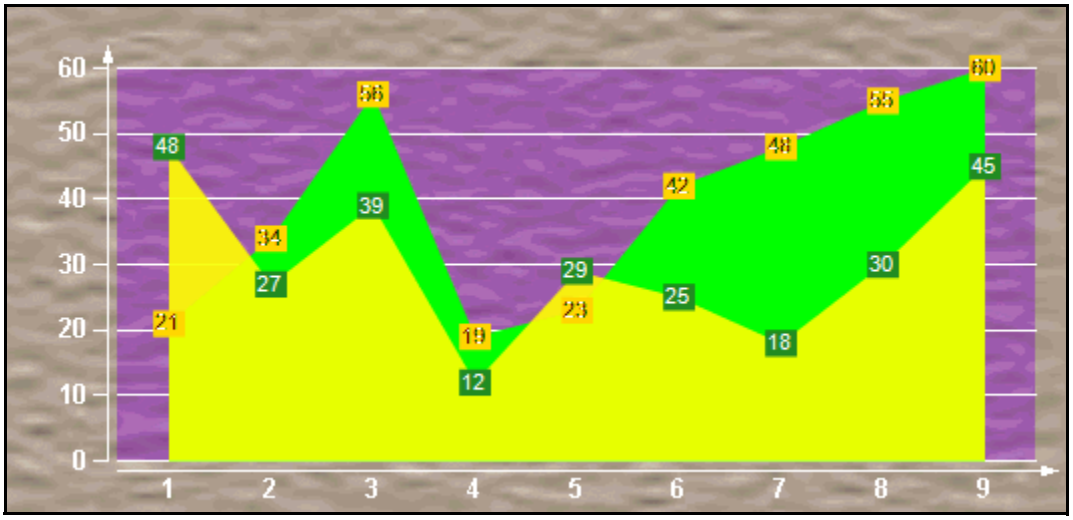
Chart Types

Chart types include Common Charts, 3D Charts, and XY Charts. See the online Help for more extensive illustrations of chart types.

Common Charts

- **Area Charts**

  Use an area chart to compare trends over a period of time or in specific categories.



  Number of Y values/data points: 1

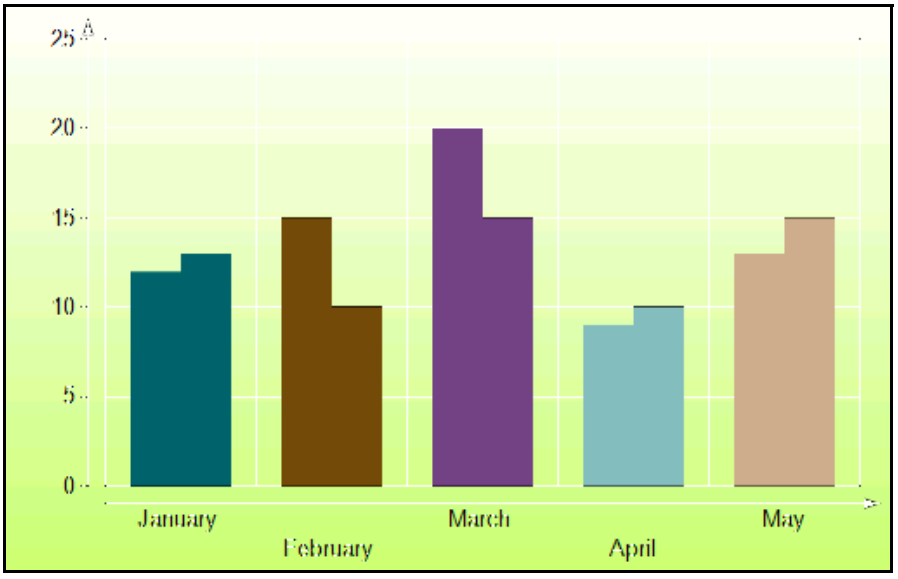
  Number of Series: 1 or more

  Marker Support: Series or Data Point

  Custom Properties: None

- **Bar2D Charts**

  Use a bar chart to compare values of items across categories.



  Number of Y values/data point: 1

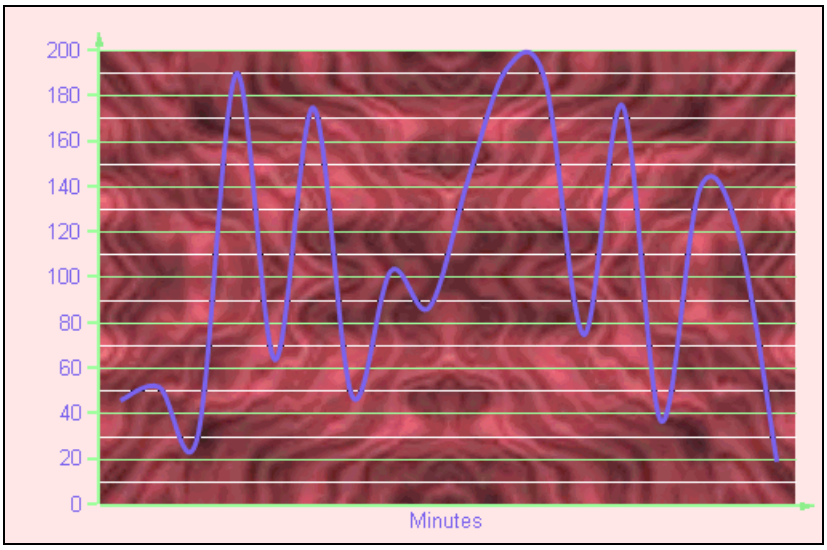
  Number of Series: 1 or more

  Marker Support: Series or Data Point

  Custom Properties: Gap gets or sets the space between the bars of each X axis value

- **Bezier Charts**

  Use a Bezier or spline chart to compare trends over a period of time or in certain categories. It is a line chart that plots curves through the data points in a series.



  Number of Y values/data point: 1

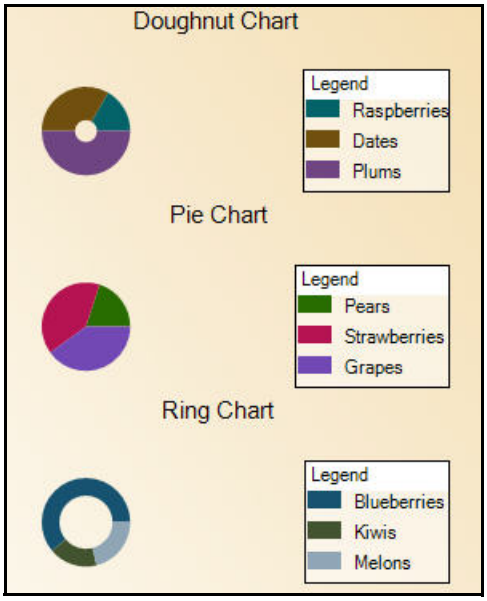
  Number of Series: 1 or more

  Marker Support: Series or Data Point

  Custom Properties: None

- **Doughnut/Pie Charts**

  A doughnut chart shows how the percentage of each data item contributes to the total.



  Number of Y values/data point: 1

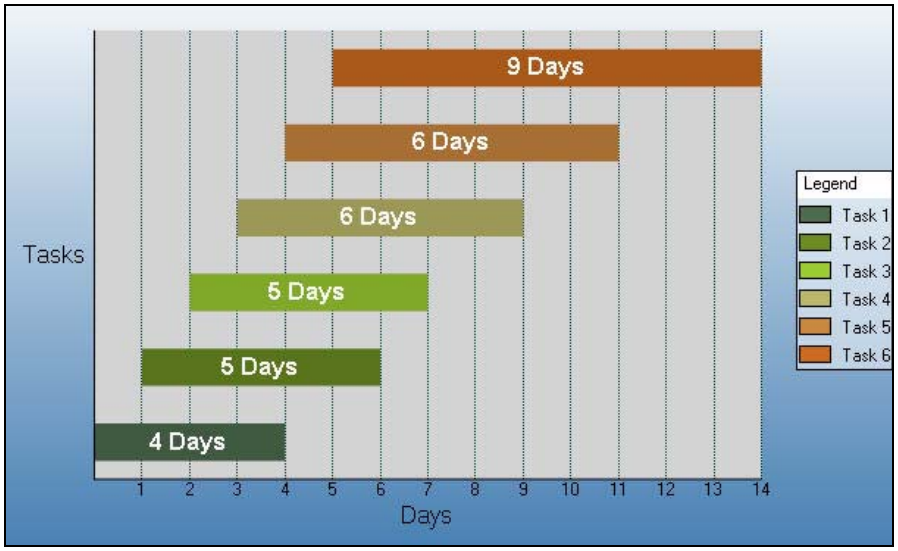
  Number of Series: 1

  Marker Support: Series or Data Point

Custom Properties: ExplodeFactor gets or sets the amount of separation between data point values. HoleSize gets or sets the inner radius of the chart. OutsideLabels gets or sets a value indicating whether the data point labels appear outside the chart. StartAngle gets or sets the horizontal start angle for the series.

- **Gantt Charts**

  The Gantt chart is a project management tool used to chart the progress of individual project tasks. The chart compares project task completion to the task schedule.



  Number of Y values/data point: 2

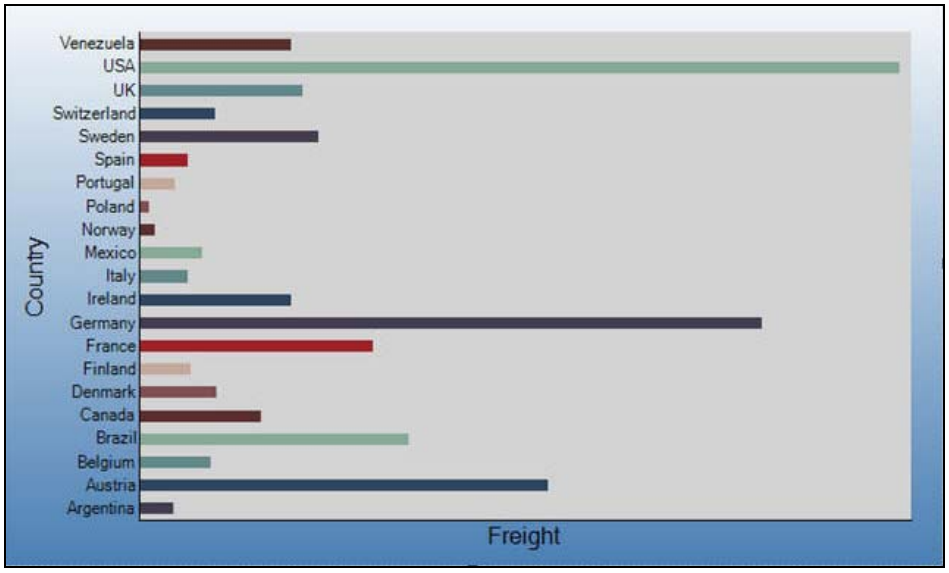
  Number of Series: 1 or more

  Marker Support: Series or Data Point

  Custom Properties: Gap gets or sets the space between the bars of each X axis value.

- **Horizontal Bar Charts**

  Use a horizontal bar chart to compare values of items across categories with the axes reversed.



Number of Y values/data point: 1

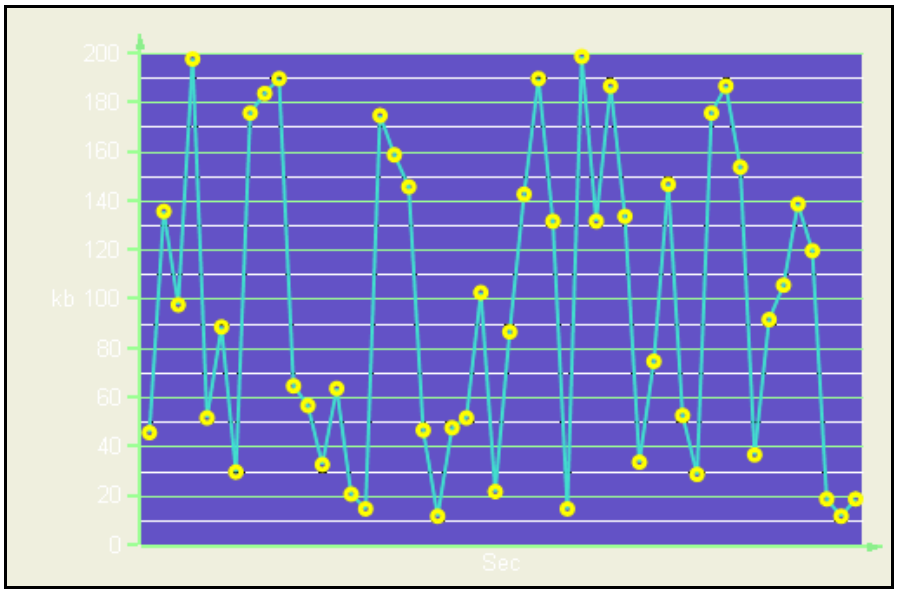
Number of Series: 1 or more

Marker Support: Series or Data Point

Custom Properties: Gap gets or sets the space between the bars of each X axis value.

- **Line Charts**

  Use a line chart to compare trends over a period of time or in certain categories.



  Number of Y values/data point: 1

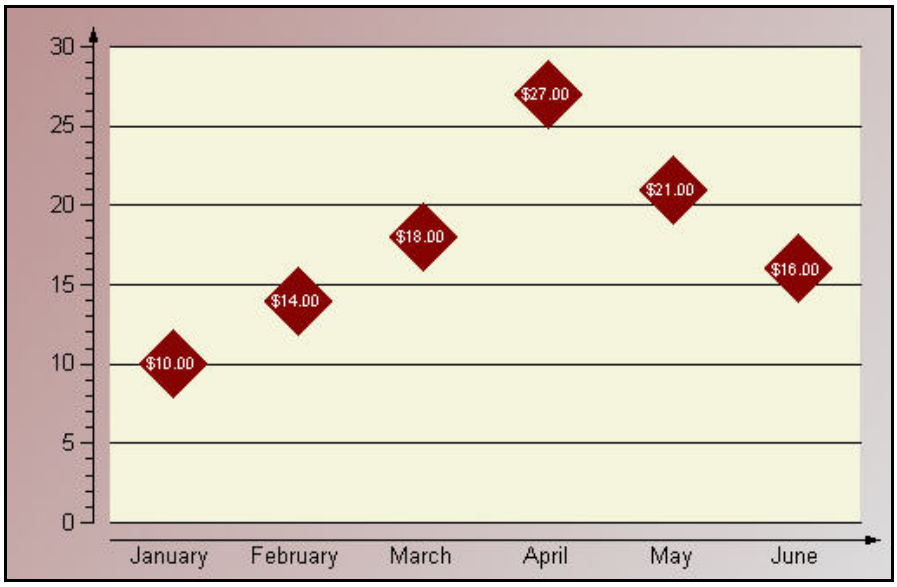
  Number of Series: 1 or more

  Marker Support: Series or Data Point

  Custom Properties: None

- **Scatter Charts**

  Use a scatter chart to compare values across certain categories.



Number of Y values/data point: 1

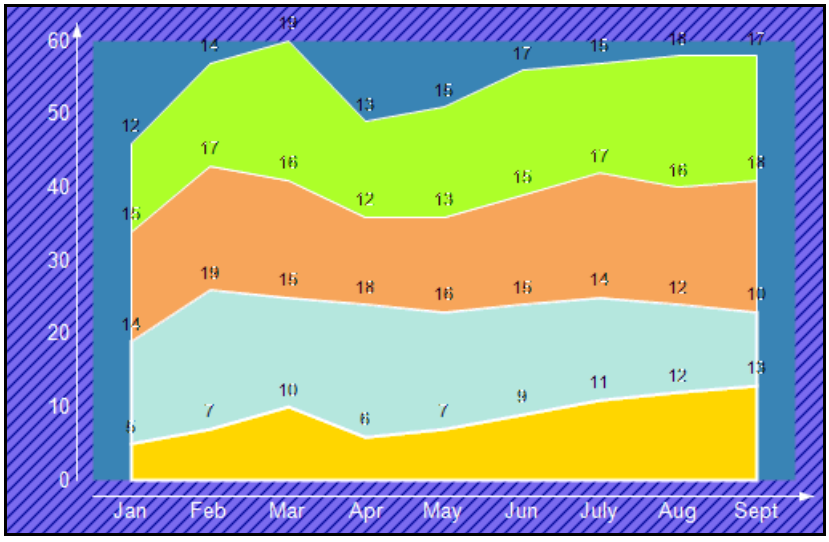
Number of Series: 1 or more

Marker Support: Series or Data Point

Custom Properties: None

- **StackedArea Charts**

  A stacked area chart is an area chart with two or more data series stacked one on top of the other. Use this chart to show how each value contributes to a total.



Number of Y values/data point: 1

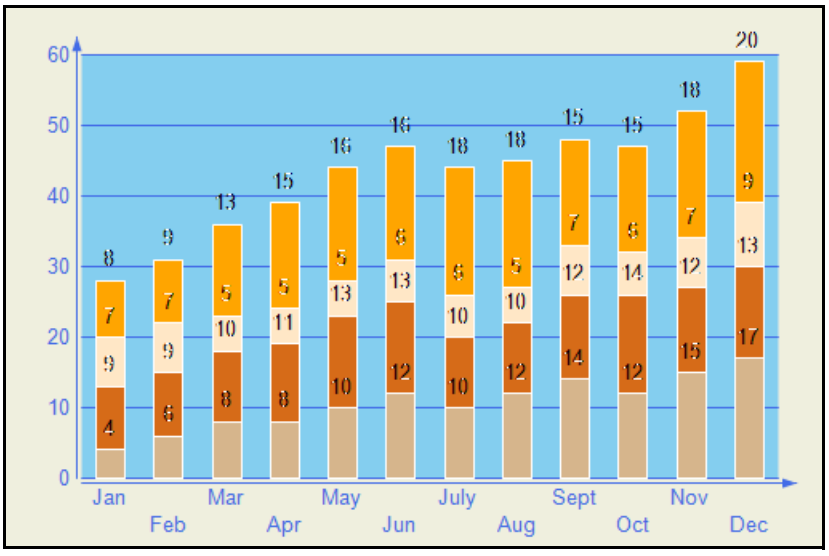
Number of Series: 1 or more

Marker Support: Series or Data Point

Custom Properties: None

- **StackedBar Charts**

  A stacked bar chart is a bar chart with two or more data series stacked one on top of the other. Use this chart to show how each value contributes to a total.



Number of Y values/data point: 1

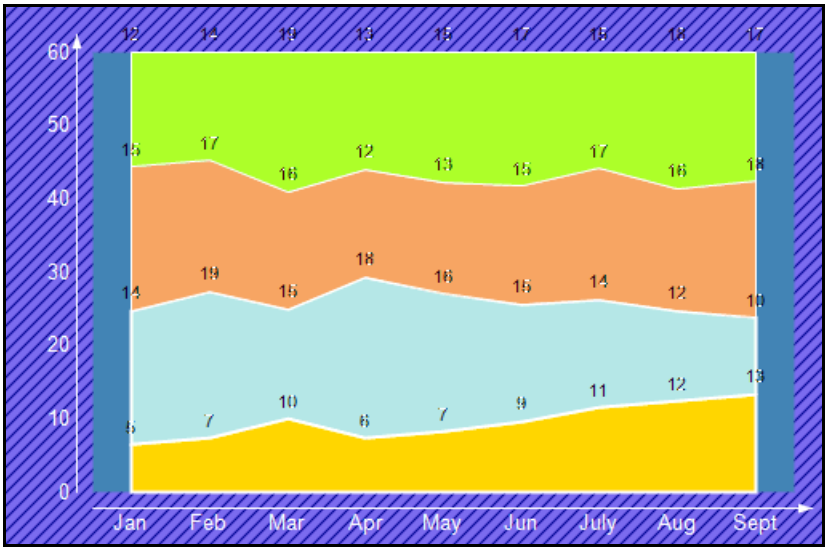
Number of Series: 1 or more

Marker Support: Series or Data Point

Custom Properties: Gap gets or sets the space between the bars of each X axis value

- **StackedArea100Pct Charts**

  A stacked area 100 percent chart is an area chart with two or more data series stacked one on top of the other to sum up to 100 percent. Use this chart to show how each value contributes to a total with the relative size of each series representing its contribution to the total.



Number of Y values/data point: 1

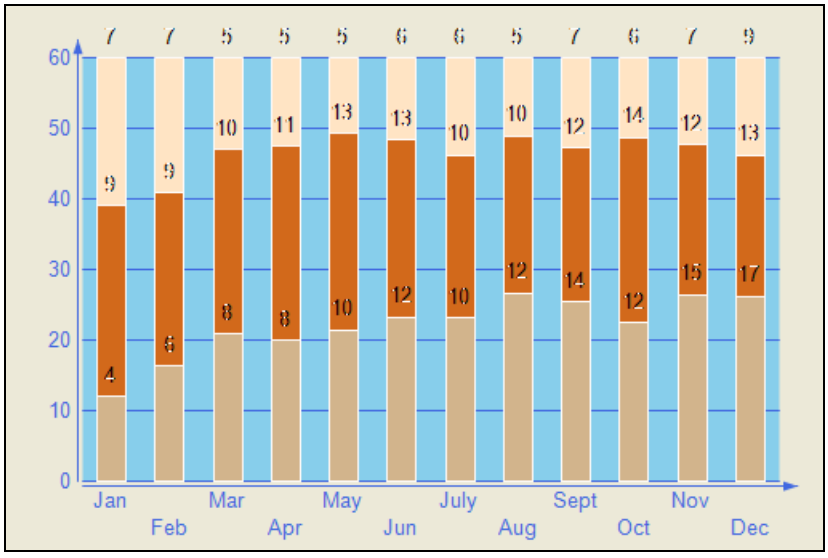
Number of Series: 1 or more

Marker Support: Series or Data Point

Custom Properties: None

- **StackedBar100Pct Charts**

  A StackedBar100Pct chart is a bar chart with two or more data series stacked one on top of the other to sum up to 100 percent. Use this chart to show how each value contributes to a total with the relative size of each series representing its contribution to the total.



  Number of Y values/data point: 1

  Number of Series: 1 or more

  Marker Support: Series or Data Point

  Custom Properties: Gap gets or sets the space between the bars of each X axis value

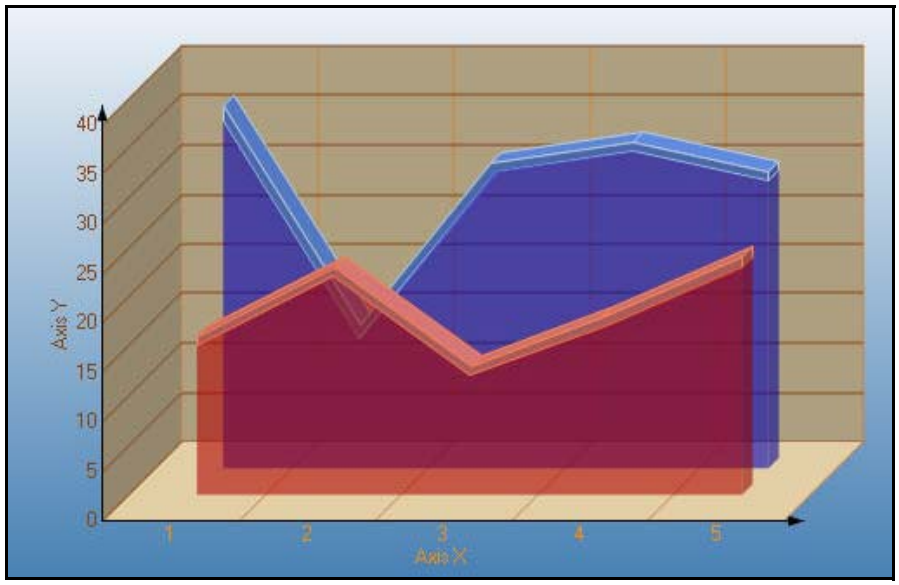
3D Charts

This topic illustrates some of the three dimensional chart types that you can create with the Chart control.

Note: To see a chart in three dimensions, open the ChartArea Collection dialog, and in the Projection section, change the ProjectionType from Identical to Orthogonal.

- **Area3D Charts**

  Use a 3D area chart to compare trends in two or more data series over a period of time or in specific categories, allowing the data to be viewed side by side.



  Number of Y values/data point: 1

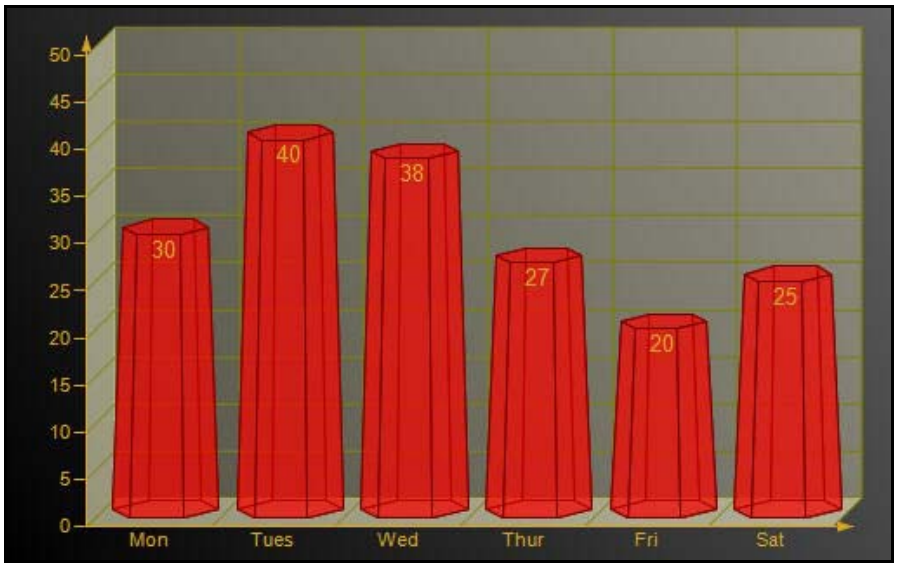
  Number of Series: 1 or more

  Marker Support: Series or Data Point

  Custom Properties: LineBackdrop gets or sets the backdrop information for the 3D line. Thickness gets or sets the thickness of the 3D line. Width gets or sets the width of the 3D line.

- **Bar3D Charts**

  Use a 3D bar chart to compare values of items across categories, allowing the data to be viewed conveniently in a 3D format.



  Number of Y values/data point: 1

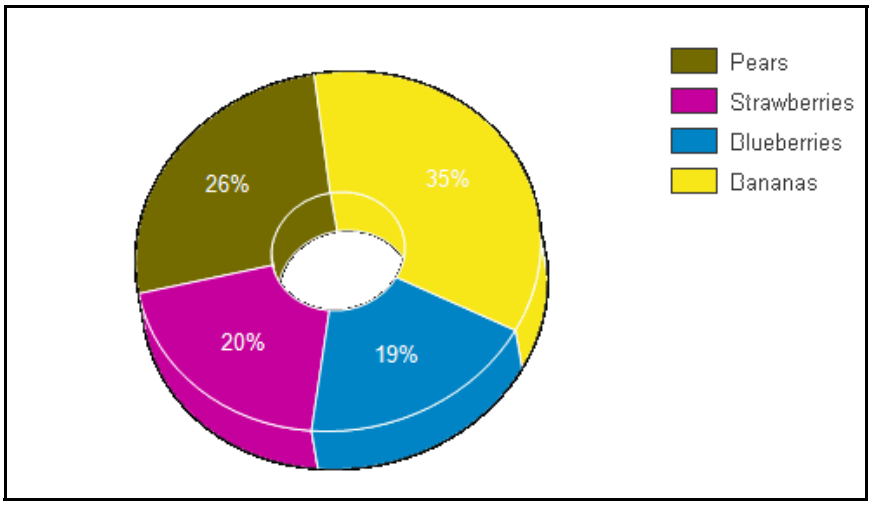
  Number of Series: 1 or more

Marker Support: Series or Data Point

Custom Properties: BarTopPercent gets or sets the percentage of the top of the bar that is shown for Cone or Custom BarTypes. BarType gets or sets the type of bars that is displayed. Gap gets or sets the space between the bars of each X axis value. RotationAngle gets or sets the starting horizontal angle for custom 3D bar shapes. Can only be used with the Custom BarType. VertexNumber gets or sets the number of vertices for the data point, used to create custom 3D bar shapes. Can only be used with the CustomBarType. Bars must contain 3 or more vertices.

- **Doughtnut3D Pie Charts**

  A 3D doughnut chart shows how the percentage of each data item contributes to a total percentage, allowing the data to be viewed in a 3D format.



  A 3D doughnut chart shows how the percentage of each data item contributes to a total percentage, allowing the data to be viewed in a 3D format.

  Number of Y values/data point: 1

  Number of Series: 1

  Marker Support: Series or Data Point

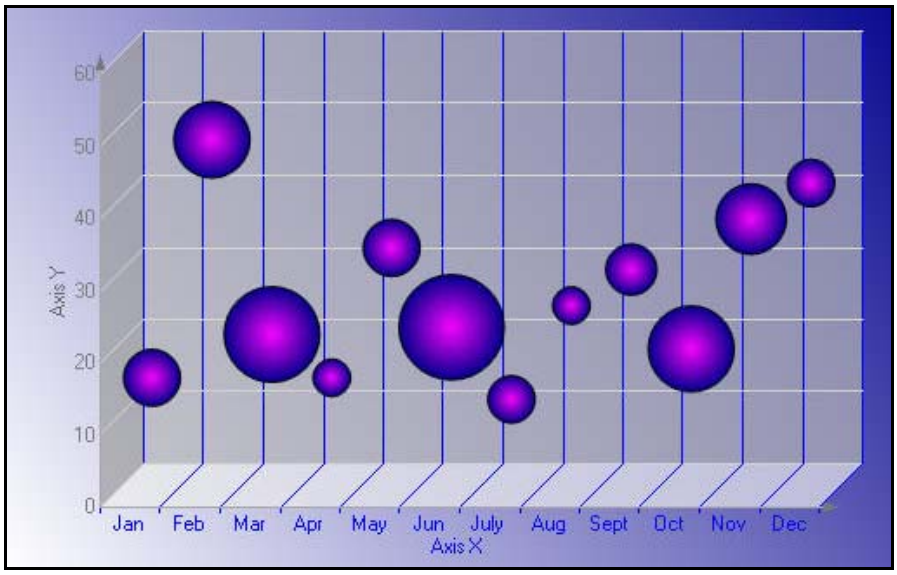
  Custom Properties: ExplodeFactor gets or sets the amount of separation between data point values. The value must be less than or equal to 1. To explode one section of the doughnut chart, set ExplodeFactor on the data point instead of on the series. HoleSize gets or sets the inner radius of the chart. If set to 0, the chart will look like a pie chart. The value must be less than or equal to 1. OutsideLabels gets or sets a value indicating whether the data point labels appear outside of the graph. StartAngle gets or sets the horizontal start angle for the series data points.

XY Charts

Some of the XY chart types you can create with the Chart control are described below.

- **Bubble Charts**

  The Bubble chart is an XY chart in which bubbles represent data points. The first Y value is used to plot the bubble along the Y axis, and the second Y value is used to set the size of the bubble. The bubble shape can be changed using the series Shape property.



  Number of Y values/data point: 2

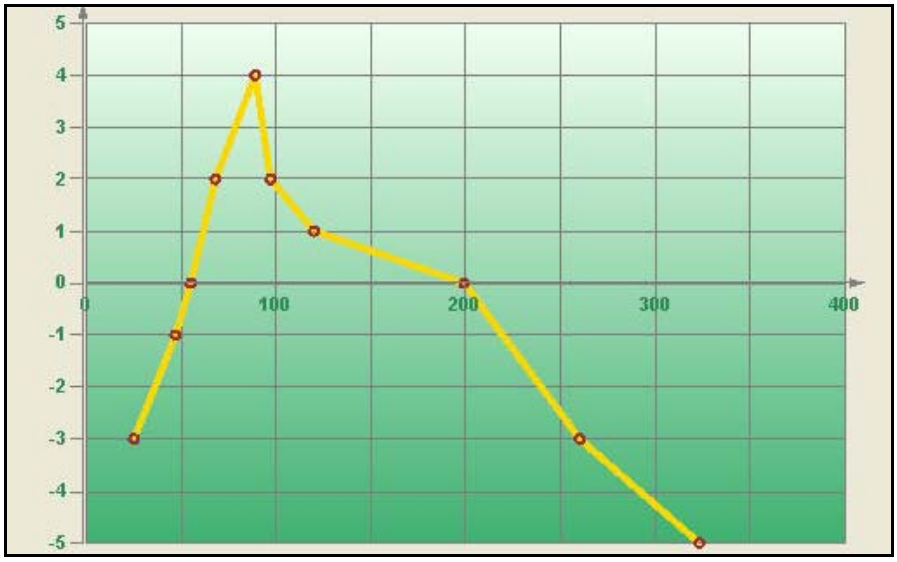
  Number of Series: 1 or more

  Marker Support: Series or Data Point. Marker labels use the second Y value as the default value.

  Custom Properties: MaxSizeFactor gets or sets the maximum size of the bubble radius. Values must be less than or equal to 1. Default is .25. MaxValue gets or sets the bubble size that is used as the maximum. MinValue gets or sets the bubble size that is used as the minimum. Shape gets or sets the shape of the bubbles. Uses or returns a valid MarkerStyle enumeration value.

- **LineXY Charts**

  A line XY chart plots points on the X and Y axes as one series and uses a line to connect points to each other.



Number of Y values/data point: 1

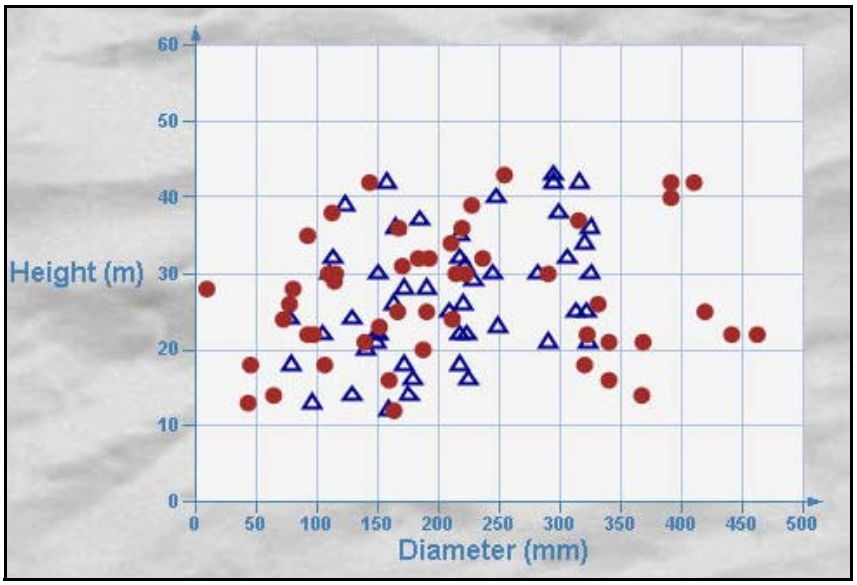
Number of Series: 1 or more

Marker Support: Series or Data Point

Custom Properties: None

- **PlotXY Charts**

  A plot XY chart shows the relationships between numeric values in two or more series sets of XY values.



  Number of Y values/data point: 1

  Number of Series: 1 or more

  Marker Support: Series or Data Point

  Custom Properties: None

Chart Data

Data-Bound Charts

The Chart control provides several ways to bind your charts to data at design time.

- Adding Data with the Wizard

  To open the Chart Wizard, right-click the chart and select Wizard. In the Chart Wizard, once you have added a series, you can create a data adapter to contain the data for your chart, if needed. When a data source is available, the Value X and Y values can be set for the series in the chart wizard from the expressions and/or data columns retrieved from the data source.

- Adding Data with the Chart Designer

  Once a data source is set up, you can easily bind data to a series using the Chart Designer. To open the Chart Designer, click the Customize verb below the *Properties* window. Choose the Series section on the left, and on the General tab, after a series has been added to the chart, set the ValueY property by selecting the name of the data expression you wish to assign to the series.

- Adding Data through the *Chart Data Source* Dialog

  To set the data source for the chart through the *Chart Data Source* dialog, click the DataSource property.

  After the DataSource for the chart is set, add a series to the chart. To do this, open the *Series Collection Editor* dialog by clicking the browse button  which appears when you click next to the Series property in the *Properties* window, then click the **Add** button. To bind the series to an expression or dataset column returned by your data source, set the ValueMembersY or ValueMembersX property of the series by selecting it from the drop-down list.

Unbound Charts

The Chart control makes it easy to set the data source for a chart control, series, or data points collection at run time.

Below is a list of objects that can be used as data sources.

- dataset
- dataset Column
- Data Table
- SqlCommand/OleDbCommand
- SqlDataAdapter/OleDbDataAdapter
- Array

Below are some examples of binding to different data sources at run time.

dataset

The Chart control's DataSource property can be set to a dataset at run time. The following code demonstrates setting up a dataset, setting the DataSource property to the dataset, creating a series, and setting the ValueMembersY property to the dataset expression at run time.

```
// C#

// create the series

DataDynamics.ActiveReports.Chart.Series s = new DataDynamics.ActiveReports.Chart.Series();

string m_cnnString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:/Northwind.mdb;Persist
    Security Info=False";

System.Data.OleDb.OleDbConnection m_cnn = new
System.Data.OleDb.OleDbConnection(m_cnnString);

System.Data.OleDb.OleDbDataAdapter oDBAdapter;

 // create the dataset

System.Data.DataSet oDS;

oDBAdapter = new System.Data.OleDb.OleDbDataAdapter("SELECT ShipCountry, SUM(Freight) AS
    Expr1 FROM Orders GROUP BY ShipCountry", m_cnnString);

oDS = new System.Data.DataSet();

oDBAdapter.Fill(oDS, "Expr1");

// set the DataSource and ValueMembersY properties

this.ChartControl1.DataSource = oDS;
```

```
s.ValueMembersY = "Expr1";

this.ChartControl1.Series.Add(s);
```

dataset Column

In the Chart control, the ValueMembersX and ValueMembersY properties of a series can be set to a dataset column. The following code demonstrates creating a series, setting up a dataset, setting the DataSource property to the dataset, and setting the ValueMembersY and ValueMembersX properties to dataset columns at run time.

```
// C#

// create the series

DataDynamics.ActiveReports.Chart.Series s = new DataDynamics.ActiveReports.Chart.Series();

string m_cnnString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:/Northwind.mdb;Persist
    Security Info=False";

System.Data.OleDb.OleDbConnection m_cnn = new
System.Data.OleDb.OleDbConnection(m_cnnString);

System.Data.OleDb.OleDbDataAdapter oDBAdapter;

 // create the dataset

System.Data.DataSet oDS;

oDBAdapter = new System.Data.OleDb.OleDbDataAdapter("SELECT * from Orders WHERE
OrderDate
    < #08/17/1994#", m_cnnString);

oDS = new System.Data.DataSet();

oDBAdapter.Fill(oDS, "Orders");

 // set the DataSource, ValueMembersY, and ValueMembersX properties

this.ChartControl1.DataSource = oDS;

this.ChartControl1.Series.Add(s);

this.ChartControl1.Series[0].ValueMembersY = oDS.Tables["Orders"].Columns[7].ColumnName;

this.ChartControl1.Series[0].ValueMemberX = oDS.Tables["Orders"].Columns[8].ColumnName;
```

Data Command

A chart's data source can be set to a SqlCommand or OleDbCommand. The following code demonstrates creating a series, creating an OleDbCommand, setting the DataSource property to the data command, and setting the ValueMembersY property for the series at run time.

```
// C#

// create the series

DataDynamics.ActiveReports.Chart.Series s = new DataDynamics.ActiveReports.Chart.Series();

string m_cnnString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:/Northwind.mdb;Persist
    Security Info=False";

System.Data.Oledb.OleDbConnection m_cnn = new
System.Data.Oledb.OleDbConnection(m_cnnString);

string query = "SELECT ShipCountry, SUM(Freight) AS Expr1 FROM Orders GROUP BY
ShipCountry";
```

```
// create the OleDbCommand and open the connection
System.Data.Oledb.OleDbCommand command = new System.Data.Oledb.OleDbCommand(query, m_cnn);
command.Connection.Open();
// set the DataSource and ValueMembersY properties
this.ChartControl1.DataSource = command;
this.ChartControl1.Series.Add(s);
this.ChartControl1.Series[0].ValueMembersY = "Expr1";
// close the connection
m_cnn.Close();
```

Array

The Chart control allows the data source for the data points collection to be set to an array. The following code demonstrates creating a series, creating an array, and using the DataBindY method to set the data source for the data points collection at run time.

```
// C#
// create the series
DataDynamics.ActiveReports.Chart.Series s = new DataDynamics.ActiveReports.Chart.Series();
// create the array
double [] a = {1,4,2,6,3,3,4,7};
// set the data source for the data points collection
this.ChartControl1.Series.Add(s);
this.ChartControl1.Series[0].Points.DataBindY(a);
```

Calculated and Sequence Series Charts

The Chart control allows you to bind a formula to the ValueMembersY property of a series to create a calculated or sequence series for your chart.

Calculated Series

You can easily create a calculated series based on the values of one or more series by setting the ValueMembersY property of a series to a formula. To reference a series in the formula, use the name of the series. The following code demonstrates creating two series, one bound to a data array and the other bound to a formula based on the Y values of the first series.

```
// C#
DataDynamics.ActiveReports.Chart.Series s = new DataDynamics.ActiveReports.Chart.Series();
DataDynamics.ActiveReports.Chart.Series cS = new DataDynamics.ActiveReports.Chart.Series();
double [] a = { 1,4,2,6,3,3,4,7};
this.ChartControl1.Series.AddRange(new DataDynamics.SharpGraph.Windows.Series[] {s, cS});
this.ChartControl1.Series[0].Name = "Series1";
this.ChartControl1.Series[0].Points.DataBindY(a);
this.ChartControl1.Series[1].ValueMembersY = "Series1.Y[0]+10";
```

Sequence Series

Set a sequence series by specifying the minimum value, maximum value, and step for the series. The following code shows how to set the ValueMembersY property at run time to create a sequence series.

```
// C#

DataDynamics.ActiveReports.Chart.Series s = new DataDynamics.ActiveReports.Chart.Series();

this.ChartControl1.Series.Add(s);

this.ChartControl1.Series[0].ValueMembersY = "sequence(12,48,4)";
```

Chart Effects

Colors

In the Chart control, colors can be used in different ways to enhance the chart's appearance, distinguish different series, point out or draw attention to data information such as averages, and more.

Color Palettes

The Chart control includes several pre-defined color palettes that can be used to automatically set the colors for data values in a series. The pre-defined palettes are as follows:

- Cascade (default): A cascade of eight cool colors ranging from deep teal down through pale orchid.
- Confetti: A sprinkling of bright and pastel colors.
- Iceberg: A range of the soft blues and greys found in an iceberg.
- Springtime: The colors of spring, in deep green, two vivid colors and five pastels.
- None: All data is drawn using the same teal color.

These enumerated values are accessed through the Series class with code like the following.

```
// C#

this.ChartControl1.Series[0].ColorPalette = DataDynamics.ActiveReports.Chart.

ColorPalette.Iceburg;
```

Gradients

Gradients can be used in object backdrops to enhance the visual appearance of various chart items. Gradients can be used in the following chart sections:

- Chart backdrop
- Chart area backdrops
- Wall backdrops
- Title backdrops
- Legend backdrops
- Legend item backdrops (for custom legend items)
- WallRange backdrops
- Series backdrops
- Data point backdrops
- Marker backdrops
- Marker label backdrops
- Annotation TextBar backdrops

3D Effects

Using the projection and viewpoint settings, you have the ability to display your 3D chart at or from any angle needed to provide the desired view or call attention to a specific chart section.

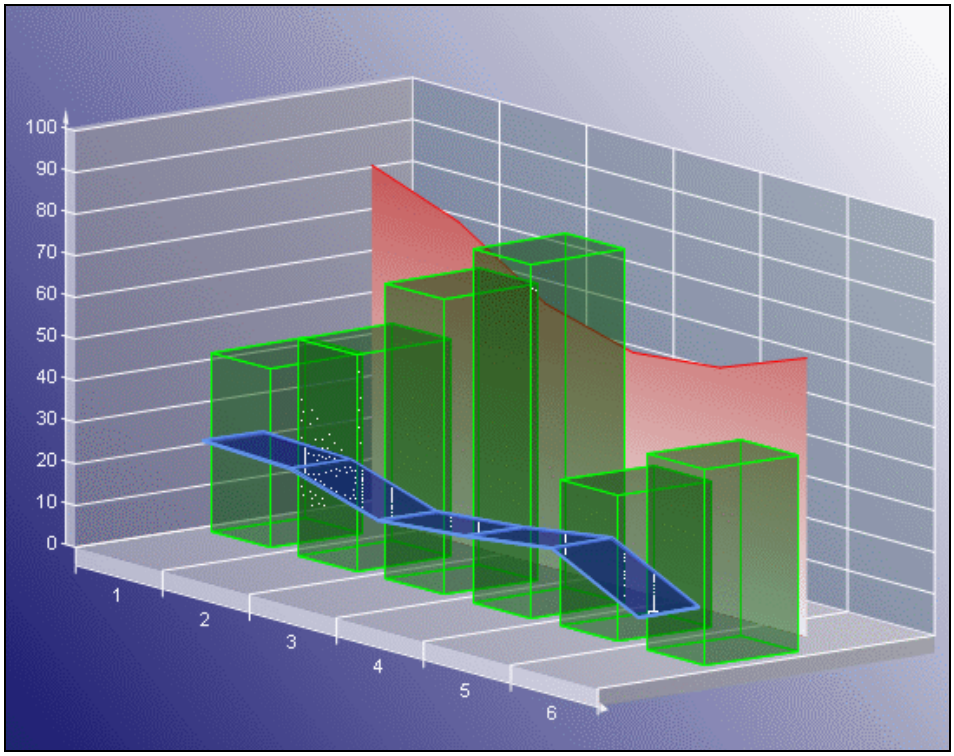
Projection

Determine the projection for a 3D chart using three factors: the ZDepth ratio, the projection type, and the projection DX and DY values.

- ZDepth ratio The Z depth ratio is the level of depth the Z axis has in the chart. Values range from 0 (for a 2D chart) to 1.0.
- ProjectionType The type of projection used for the chart. In order to show charts three dimensionally, the ProjectionType in the ChartArea Collection editor must be set to Orthogonal. To access this dialog box, click the browse button ... next to the ChartAreas (Collection) property in the *Properties* window.
- ProjectionDX The origin position of the Z axis in relation to the X axis. This property is valid only when the ProjectionType is Orthogonal.
- ProjectionDY The origin position of the Z axis in relation to the Y axis. This property is valid only when the ProjectionType is Orthogonal.
- HorizontalRotation The HorizontalRotation property allows you to set the degree (-90° to 90°) of horizontal rotation from which the chart is seen.
- VerticalRotation The VerticalRotation property allows you to set the degree (-90° to 90°) of vertical rotation from which the chart is seen.

Lighting

The Chart control provides the ability to completely customize lighting options for 3D charts.



Directional Light Ratio

Using the DirectionalLightRatio property, you can control the directional or ambient intensity ratio.

Light Type

By setting the Type property to one of the enumerated LightType values, you can control the type of lighting used in the chart. The settings are as follows:

- Ambient An ambient light source is used. It is equal to DirectionalLightRatio = 0.
- InfiniteDirectional An infinite directional light source (like the sun) is used.
- FiniteDirectional A point light source is used.

Light Source

You can also set the Source property to a Point3d object, which controls the location of the light source.

Alpha Blending

The Backdrop class in the Chart control has an Alpha property which employs GDI+, and is used to set the transparency level of each object's backdrop. GDI+ uses 32 bits overall and 8 bits per alpha, red, green, and blue channels respectively to indicate the transparency and color of an object. Like a color channel's levels of color, the alpha channel represents 256 levels of transparency.

The default value of the Alpha property is 255, which represents a fully opaque color. For a fully transparent color, set this value to 0. To blend the color of the object's backdrop with the background color, use a setting between 0 and 255.

In the Chart control, you can use the Color.FromArgb method to set the alpha and color levels for a particular chart element. The following example shows how you can use the method to set the alpha and color values for the chart backdrop.

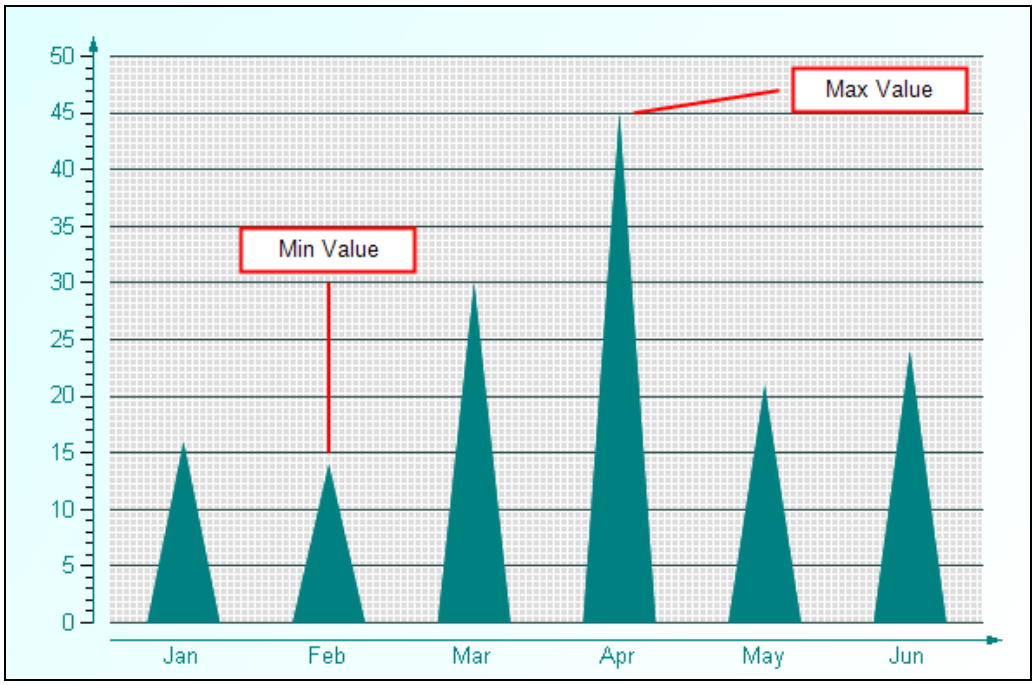
```
// C#

this.ChartControl1.Backdrop = new DataDynamics.ActiveReports.Chart.

BackdropItem(Color.FromArgb(100, 0, 11, 220));
```

Changing the alpha level of a chart element reveals other items that are beneath the object. Because you can set the alpha level for any chart element that supports color, you can create custom effects for any chart. For example, you can use alpha blending to combine background images with a semi-transparent chart backdrop to create a watermark look.

Chart Control Items

Annotations

The Chart control offers a built-in annotation tool to allow you to include floating text bars or images in your charts or call attention to specific items or values in your charts using the line and text bar controls included in the Annotation Collection Editor.

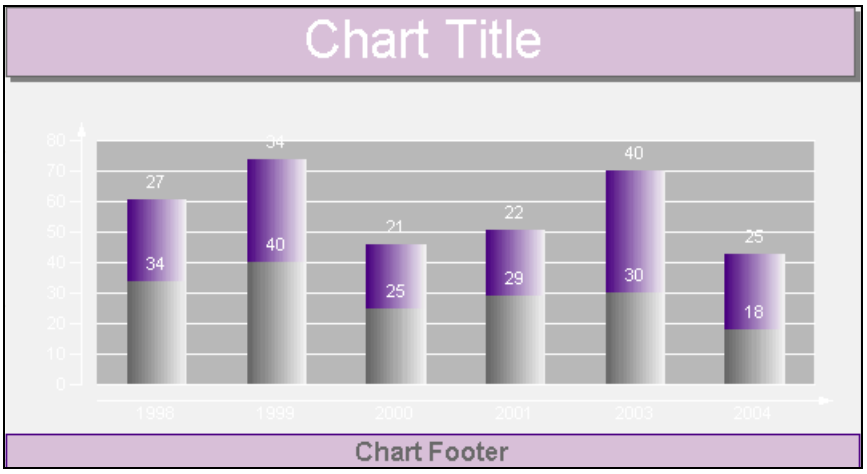


The following properties are important when setting up annotations for your chart:

- Start Point: sets the starting point (X and Y axis values) for an annotation line.
- End Point: sets the end point (X and Y axis values) for an annotation line.
- Anchor Placement: sets the position of the anchor point for the text bar on the chart surface.
- Anchor Point: sets the point (X and Y axis values) where the text bar will be anchored based on the anchor placement selected.

Titles and Footers

The Chart control allows you to add custom titles to your charts. The Titles collection is accessible from the SharpGraph object. With the ability to add as many titles as needed, dock them to any side of a chart area, change all of the font properties, add borders and shadows, make the background look the way you want it, and change the location of the text, you can easily make your titles look the way you want them to look.

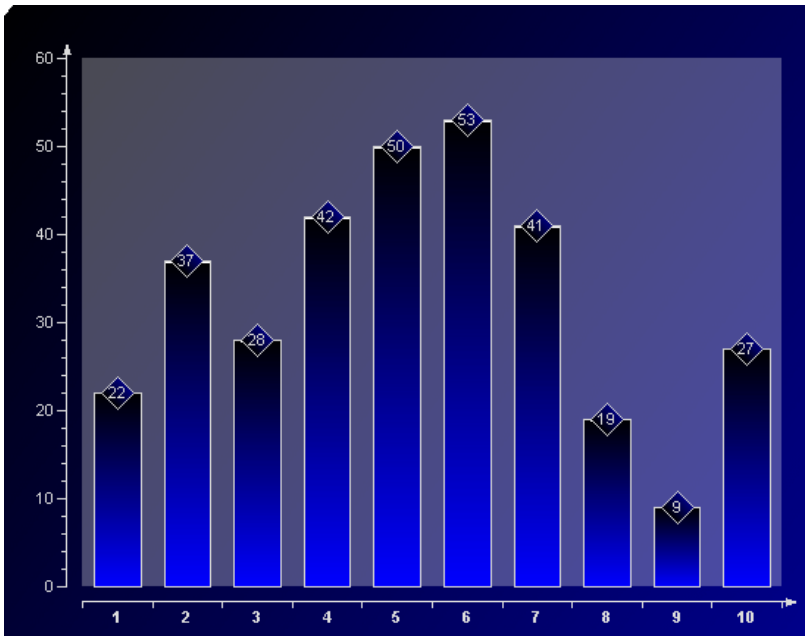


Legends

The Chart control automatically creates a legend item for each series added to a chart at design time and sets the Legend property for each series by default. However, the legend's Visible property must be set to True for the legend to show with the chart. The text for each default legend entry is taken from the Name property on the series. Each Series to be shown in the Legend must have a Name. If the Name property is not set, the Series does not show up in the Legend.

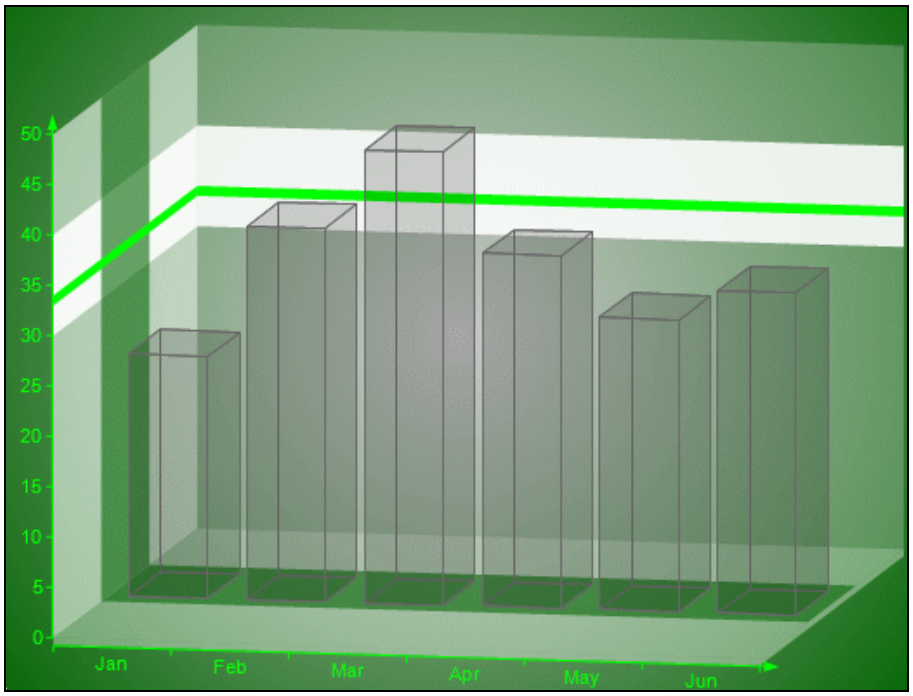
Markers

Markers are used to show specific data series values in a chart.



Constant Lines and Stripes

The Chart control supports constant lines and stripes through the use of the WallRanges collection. It allows you to display horizontal or vertical lines or stripes in a chart to highlight certain areas. For example, you could draw a stripe in a chart to draw attention to a high level in the data or draw a line to show the average value of the data presented.



Important properties

- EndValue--Sets the end value on the primary axis for the wall range.
- StartValue --Sets the start value on the primary axis for the wall range.
- PrimaryAxis--Sets the axis on which the wall range should appear.

Chart Axes and Walls

Standard Axes

The Chart control provides the means to change axis settings at design time or run time. Chart axes make it possible to view and understand the data plotted in a graph.

Axis Types

Most 2D charts contain a numerical axis (AxisY) and a categorical axis (AxisX). 3D charts include another numerical axis (AxisZ). These axes are accessible at run time from the ChartArea object and allow you to control the settings for each, including scaling, labels, and various formatting properties. For any of the scaling or labeling properties you set to show up at run time, you will need to set the Visible property of the axis to True.

Changing Axis Settings

Axis settings can be changed at design time by clicking on a Chart control and using the *Properties* window or at run time in code from the chart's ChartArea object.

Scaling

For normal linear scaling on a numeric axis, you will need to set the Max and Min properties for the axis, which correspond to the numerical values in the chart's data series. You will also need to set the Step property of the MajorTick to show the major numerical unit values. The Step property controls where labels and/or tick marks are shown on the numerical axis.

```
// C#

this.ChartControl1.ChartAreas[0].Axes["AxisY"].Max = 100;

this.ChartControl1.ChartAreas[0].Axes["AxisY"].Min = 0;

this.ChartControl1.ChartAreas[0].Axes["AxisY"].MajorTick.Step = 10;
```

The Chart control also supports logarithmic scaling which allows you to show the vertical spacing between two points that corresponds to the percentage of change between those numbers. You can set your numeric axis to scale logarithmically by setting the IsLogarithmic property on the axis to True and setting the Max and Min properties of the axis.

Labeling

To show labels on an axis, you will need to specify the value for the LabelsGap property, set your LabelsFont properties, and set LabelsVisible to True. These properties can be set in the AxisBase Collection editor, which is accessed at design time by clicking the browse button  next to the ChartAreas (Collection) property, then the Axes (Collection) property of the ChartArea.

NOTE: Labels render first, and then the chart fills in the remaining area, so be sure to make the chart large enough if you use angled labels.

You can specify strings to be used for the labels instead of numerical values on an axis by using the Labels collection property at design time or assigning a string array to the Labels property at run time. You can also specify whether you want your axis labels to appear on the outside or inside of the axis line using the LabelsInside property. By default, labels appear outside the axis line.

Secondary Axes

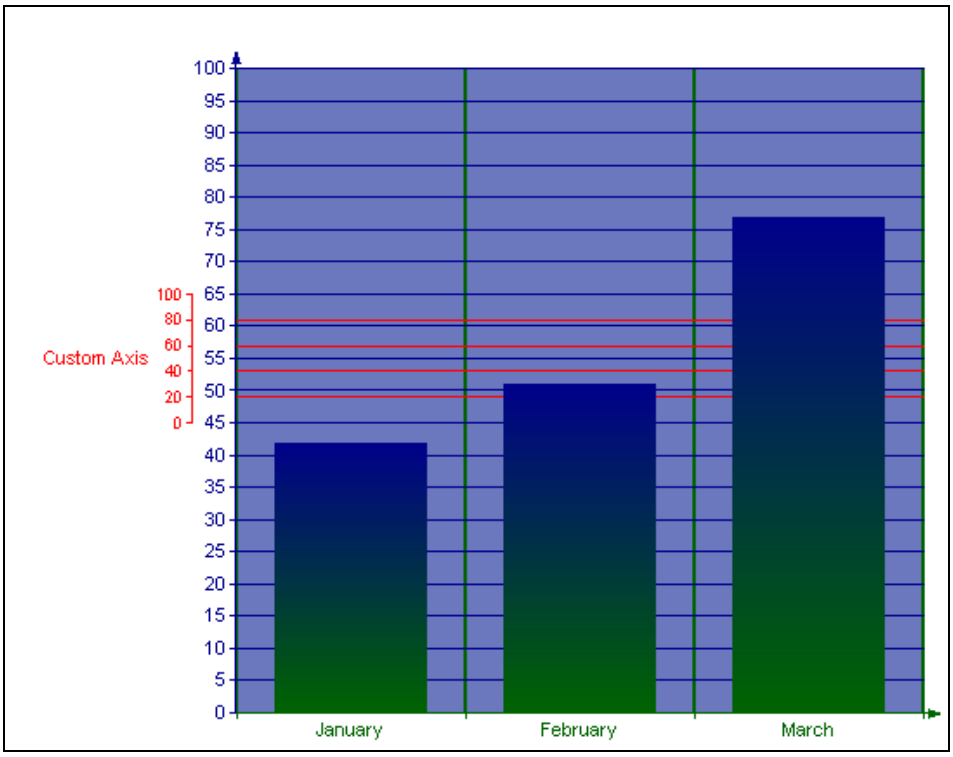
By default, a Chart object includes secondary X and Y axes (AxisX2 and AxisY2). At design time or run time, you can specify a secondary axis to plot data against by setting all of the appropriate properties for AxisX2 or AxisY2, including the Visible property.

If you want to use two axes to show the same data as it appears on two different scales, you can set the primary axis to show the actual data value scale, for example, and set the secondary axis to show a logarithmic scale.

Custom Axes

The Chart control supports the creation of additional custom axes through the use of the chart's CustomAxes collection. Once a custom axis has been added to the collection, in addition to setting the normal axis properties, you will need to set the following properties:

- Parent—The Parent property allows you to choose the primary or secondary axis on which your custom axis resides.
- PlacementLength—The PlacementLength property allows you to set the length of the custom axis in proportion to the Min and Max property values you have already set for the parent axis.
- PlacementLocation—The PlacementLocation property allows you to set the starting location value for the custom axis to appear in relation to the parent axis.



Gridlines and Tick Marks

Gridlines and tick marks are generally used to help increase the readability of a chart.

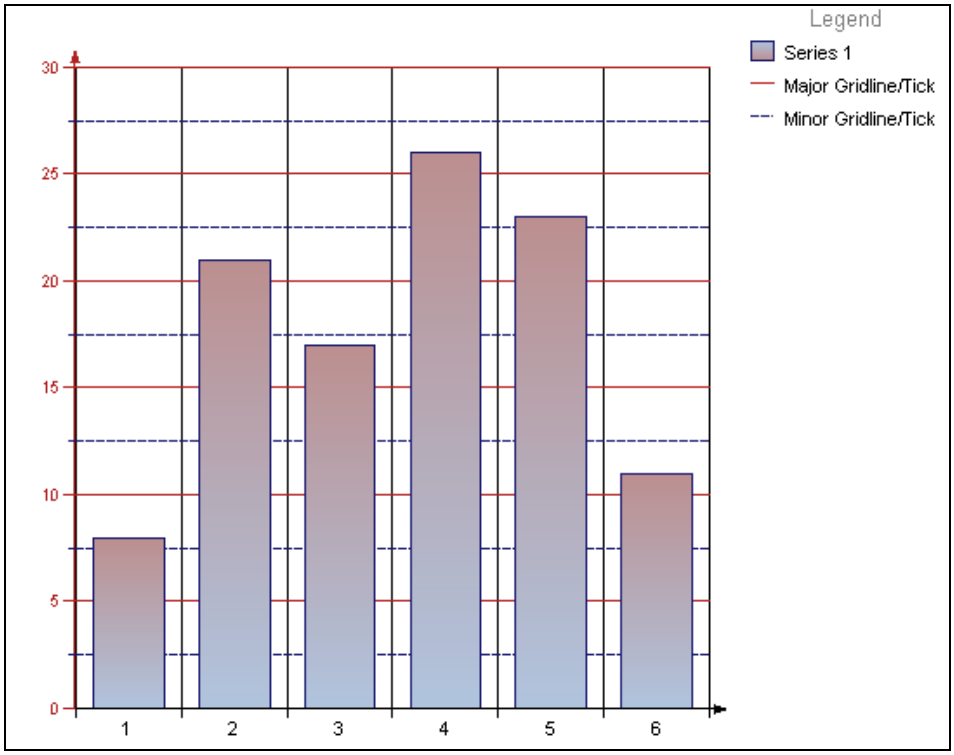
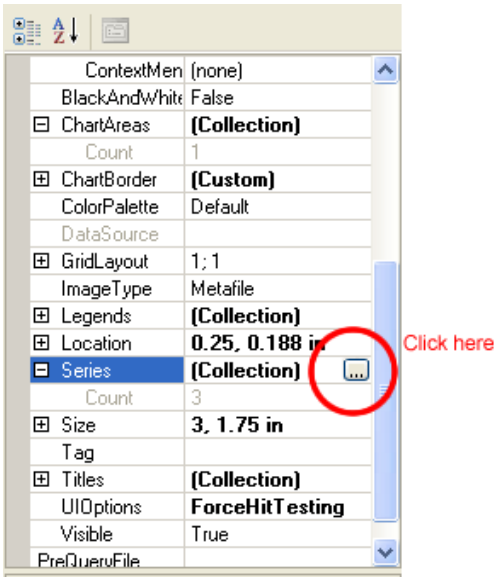


Chart-Specific Properties

Each chart type in the Chart control contains specific properties that apply to it. Set the chart type and chart-specific properties in the *Series Collection Editor* dialog box accessed through the Series property in the property grid and in the *DataPoint Collection* dialog box accessed through the Points property in the *Series* dialog box.



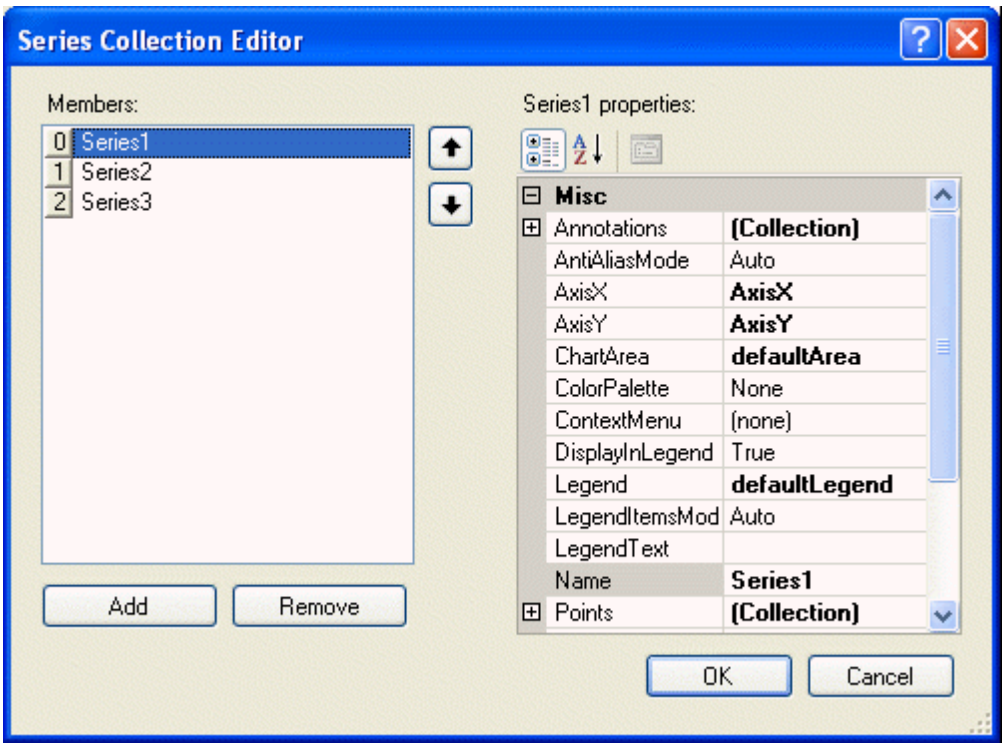
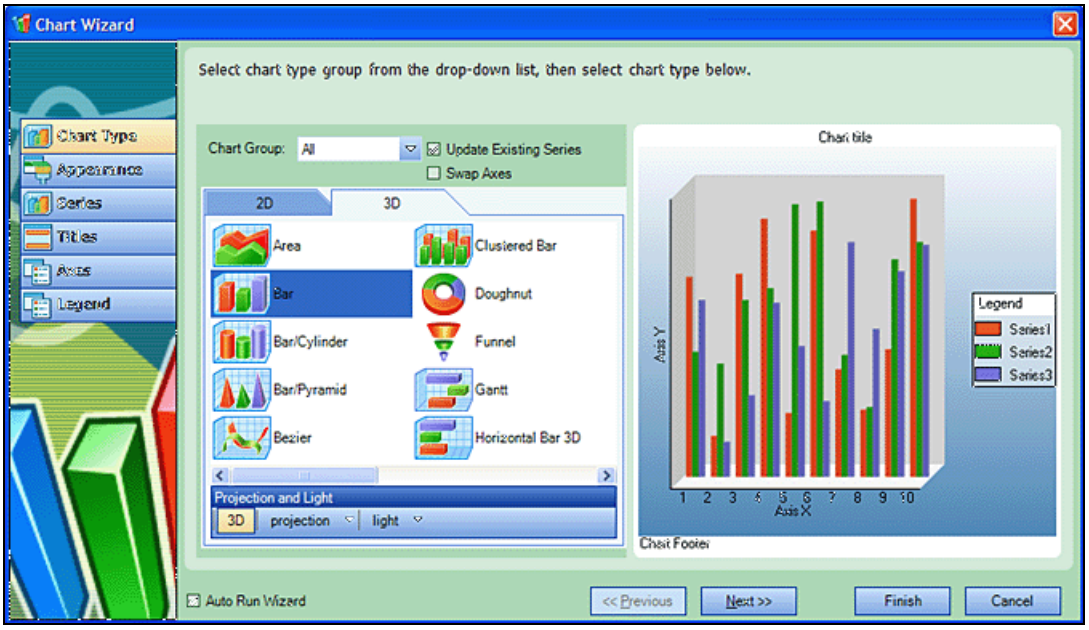


Chart Wizard

The chart control features an easy-to-use wizard. The chart wizard automatically runs when you first add a chart control to a report. If you prefer not to have the wizard run automatically, uncheck the Auto Run Wizard check box at the bottom of the wizard. You can also access the wizard through the Wizard verb that appears below the *Properties* window when the chart is selected on the report.



Walk-Through: Creating a Report

In this exercise, you will build a report similar to the vulnerability (classic) report, but not as intricate.

Task 1: Build the master report

1. Click **File** → **New**.
2. On the *Create Report Definition* window, enter the following:

   Name: My vulnerability report

   Description: Sample report
3. In the Context list, select **Scan**.
4. Select **Exposed in Product**.
5. In the **View Name** list, select **Basic - Server Information**.
6. Click **OK**.
7. Right-click the PageHeader caption and select Delete.
8. Right-click the Detail caption and select **Insert** → **Report Header/Footer**.
9. In the toolbox, drag LinkedSubReportControl into the ReportHeader section.
10. On the *Choose a Report* dialog, expand the Vulnerability (classic) group, select **ScanHeader**, and click **OK**.
11. Position the element and extend it to the right margin.

12 Click the ReportHeader caption.

13 In the Properties grid, set CanShrink = True.

14 Click the Detail caption.

15 In the Properties grid, set CanShrink = True.

Task 2: Add a Link to a Subreport

1 In the toolbox, drag a LinkedSubReportControl into the Detail section.

2 On the *Choose a Report* dialog, expand the Vulnerability (classic) group, select **ServerHeader**, and click **OK**.

3 Position the element and extend it to the right margin.

4 With the ServerHeader selected, in the Properties grid under Associated Fields, click **@ServerID** and select ServerID.

5 Click the **Preview** tab.

6 When prompted to design parameters, select **No**.

7 Select a scan and click **Next**.

8 When prompted to select a report, click **Finish**.

9 Click **File** → **Save**.

Task 3: Create a Subreport

1 Click **File** → **New**.

2 On the *Create Report Definition* window, enter or select the following:

a For the Name, enter “My vulnerability by server.”

b For the Description, enter “Sample report.”

3 From the **Context** list, select **Scan**.

4 Clear the **Exposed in Product** check mark.

5 In the **View Name** list, select *Basic - Vulnerability by Session*.

6 Click **OK**.

7 Delete the PageHeader caption (right-click the caption and select **Delete**).

8 In the Properties grid, set CanShrink = True.

9 Right-click the Detail caption and select **Insert** → **Group Header/Footer**.

10 In the Properties grid:

a Set CanShrink = True.

b Change the name to GroupServer.

c For the DataField, select Server.

11 Drag a BookmarkControl to the GroupServer area.

12 In the Properties grid, select BookmarkText and enter the following:

{=MainReportName}\{=Server}

Task 4: Add a chart to the report

1. Click **Edit** → **Modify/Create Report**.
2. Select **Aggregate - Severity Summary by Server** and click **OK**. This query will be used to generate a chart.
3. Drag a ChartControl onto the design area.
4. On the Chart Wizard, click the **2D** tab and select **Bar**.
5. Click **Finish**.
6. Resize the chart and arrange it to your liking.
7. With the chart selected, go to the Properties grid, click **AssociatedQuery**, and select the query you just added: Aggregate - Severity Summary by Server.
8. Right-click the chart and select Wizard.
9. Select **Series** from the list in the left-hand pane.
10. Assign a series to each severity category: critical, high, medium, low, informational, and best practice.
    a. Select **Series1**, and in the Series Properties area and enter "Critical" for the Name.
    b. In the Data Binding area, select the Y axis and select **Critical** from the drop-down list.
    c. Repeat this process for each series; click **Add New Item** where necessary.
11. Click **Finish**.
12. With the chart selected, go to the Properties grid and click **@ServerID** under AssociatedFields and select VulnerabilityCount.

Task 5: Add a section for the Check ID, Check Severity, and Check Name, and Summary

1. Right-click the Detail caption and select **Insert** → **Group Header/Footer**.
2. Collapse the footer.
3. Click the GroupHeader.
4. In the Properties grid:
    a. Change the name to "groupCheck."
    b. Set CanShrink = True
    c. For the DataField, select "checkid."
5. Drag a TextBox to the groupCheck section.
6. In the Properties grid:
    a. For Name, enter txtSeverity
    b. For DataField, select "checkseverity."
7. Drag another TextBox into the groupCheck section and place it to the right of the first TextBox.
8. In the Properties grid:
    a. Change the name to "txtCheckName."
    b. Set CanShrink = True
    c. For the DataField, select "checkname."
    d. For ClassName, select Normal Bold.
9. Drag a Label into the area.

10. In the Properties grid:
    a. Change the name to lblSummary.
    b. For Text, enter Summary.
11. Drag a RichTextBox onto the canvas; place it below the summary label and extend it to the right.
12. In the Properties grid:
    a. Change the name to txtSummary.
    b. For the DataField, select ReportSection_Summary.
13. Drag a BookmarkControl and place it anywhere on the groupCheck canvas
14. On the Properties grid:
    a. For BookMarkText, enter {=MainReportName}\Checks\{=Checkid}.
    b. For the Name, enter BookmarkChecks.

Task 6: Add an area for the HTTP Request

1. Right-click on the Detail caption and select **Insert** → **Group Header/Footer**.
2. Collapse the group footer.
3. On the Properties grid:
    a. Set CanShrink =True.
    b. For the Name, enter groupRequest.
4. Drag a RichTextBox to the groupRequest canvas. Size it and extend it to the right margin.
5. On the Properties grid:
    a. Set CanShrink =True.
    b. For the Name, enter txtRequest.
    c. For the DataField, select RequestText.
    d. For TruncateVulnerability, select True.
    e. For HighlightVulnerability, select True

Task 7: Add an area for the HTTP Response

1. Drag a RichTextBox to the groupRequest canvas. Size it and extend it to the right margin.
2. On the Properties grid:
    a. Set CanShrink =True.
    b. For the Name, enter txtResponse.
    c. For the DataField, select ResponseText.
    d. For TruncateVulnerability, select True.
    e. For HighlightVulnerability, select True

Populate the Detail section

1. Drag the bound field “fullURL” to the Detail section.
2. Click the Parameter Designer icon on the toolbar.

3 In the Parameter Designer Canvas area, delete all parameters (click in the area, press Ctrl + a, and then press **Delete**).

4 Click **Save and Close**.

Task 8: Add/Modify the script

1 Click the **Script** tab on the Report Designer.

2 Change the method name “myEventHandler” to “onGroupCheckFormat.”

3 Delete all the script and replace with the following:

```
using System;

using DataDynamics.ActiveReports;

using HP.AppSec.Reporting.ReportScript;

namespace Script.Events

{
    public class MyEventClass
    {
        /*
         * You can declare fields, events and methods just like in c#...
         * in fact this is C#!
         */
         /*
         * Script event handlers, MUST have this method signature
         */
       public void OnGroupCheckFormat(ScriptReportObject report, EventArgs
ea)
        {
                    int nSeverity = (int)report.Fields["checkseverity"];
                    TextBox txtSeverity =
report.CurrentSection.Controls["txtSeverity"] as TextBox;
                    if (nSeverity <= 10)
                    {
                          txtSeverity.Text = "Informational";
                    }
                    else if( 10 < nSeverity && nSeverity <= 25)
                    {
                          txtSeverity.Text = "Low";
                    }
                    else if( 25 < nSeverity && nSeverity <= 50)
                    {
                          txtSeverity.Text = "Medium";
                    }
                    else if( 50 < nSeverity && nSeverity <= 75)
                    {
                          txtSeverity.Text = "High";
                    }
                    else if( 75 <  nSeverity && nSeverity <= 100)
                    {
                          txtSeverity.Text = "Critical";
                    }
        }
    }
}
```

4. After entering the script, click the **Report Events** tab (in the lower right) and select **groupCheck** from the drop-down list.
5. For the Section Format Event, select Script.Events.MyEventClass.onGroupCheckFormat.
6. Save the report.

Task 9: Add a pre-query to the master report

1. Open MyVulnerability report (listed under Custom Reports on the *Open a Report* dialog).
2. Click **Edit** → **Modify/Create Report**.
3. From the **View Name** list, select **PreQuery - Vulnerability**.

   A pre-query improves performance by first determining if any data is available for the report.
4. Drag a LinkedSubReportControl onto the Detail area.
5. From the *Choose a Report* dialog, select My vulnerability by server and click **OK**.
6. Position the control and extend it to the right margin.
7. On the Properties grid:
   a. Under AssociatedFields, click **@serverID** and select serverID.
   b. For PreQueryFile, select PreQuery - Vulnerability.
8. Click **Save**.
9. Click the **Preview** tab.
10. Note and correct any improperly positioned controls, then save your work.

12 Server Analyzer

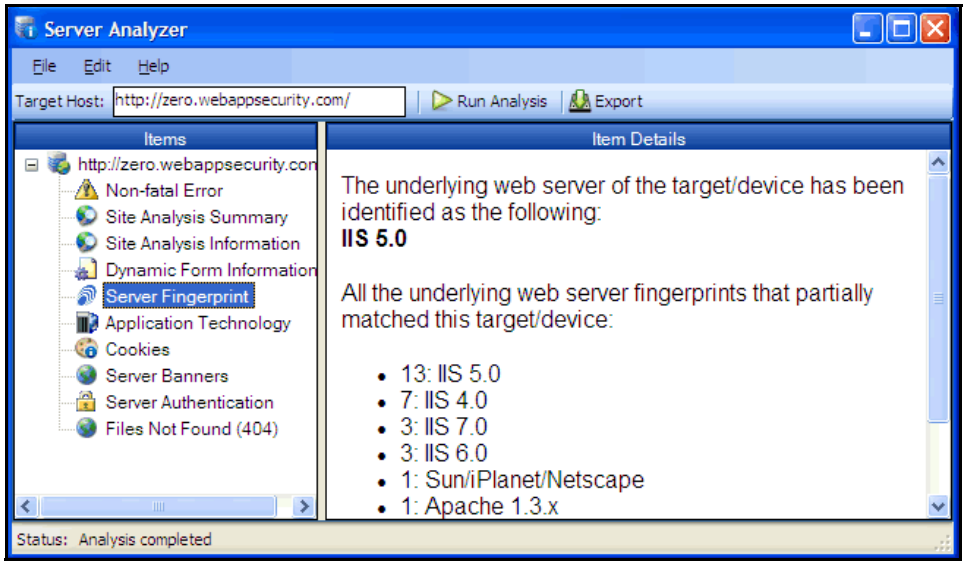
About the Server Analyzer Tool

The Server Analyzer interrogates a server to determine the server's operating system, banners, cookies, and other information.

Analyzing a Server

To analyze a server:

1. In the **Target Host** box, enter the URL or IP address of the target server.
2. If host authentication is required, or if you are accessing the host through a proxy server, select **Edit** → **Settings** and provide the requested information. See Server Analyzer Settings for detailed information.
3. Click the **Run Analysis** icon.



Server Analyzer Settings

To modify the Server Analyzer settings:

1. Click **Edit** → **Settings**.
2. Select either the **Host Authentication** or **Proxy** category and enter the settings described in the following sections.
3. Click **OK**.

Authentication Method

If authentication is required, select a type from the **Authentication** list. See Authentication Types on page 44 for a description of the available authentication types.

Authentication Credentials

Type a user ID in the **User name** box and the user's password in the **Password** box. To guard against mistyping, repeat the password in the **Confirm Password** box.

To use these credentials whenever the Server Analyzer encounters a password input control, select **Submit these credentials to forms with password input fields**.

Proxy

Use these settings to access the Server Analyzer through a proxy server.

Direct Connection (proxy disabled)

Select this option if you are not using a proxy server.

Auto detect proxy settings

Use the Web Proxy Autodiscovery (WPAD) protocol to locate a proxy autoconfig file and configure the browser's web proxy settings.

Use Internet Explorer proxy settings

Import your proxy server information from Internet Explorer.

Use Firefox proxy settings

Import your proxy server information from Firefox.

Configure a proxy using a PAC file

Load proxy settings from the Proxy Automatic Configuration (PAC) file in the location you specify in the **URL** box.

Explicitly configure proxy

Configure a proxy by entering the requested information.

1. In the **Server** box, type the URL or IP address of your proxy server, followed (in the **Port** box) by the port number (for example, 8080).
2. Select a protocol for handling TCP traffic through a proxy server: SOCKS4, SOCKS5, or standard.
3. If authentication is required, select a type from the **Authentication** list. See Authentication Types on page 44 for a description of the available authentication types.
4. If your proxy server requires authentication, enter the qualifying user name and password.
5. If you do not need to use a proxy server to access certain IP addresses (such as internal testing sites), enter the addresses or URLs in the **Bypass Proxy For** box. Use commas to separate entries.

HTTPS Proxy Settings

For proxy servers accepting HTTPS connections, select **Specify Alternative Proxy for HTTPS** and provide the requested information.

Exporting Results

To export the results of the analysis to an HTML file:

1. Click **File** → **Export**.
2. On the *Export File* window, select or enter a location and file name.

Click **Save**.

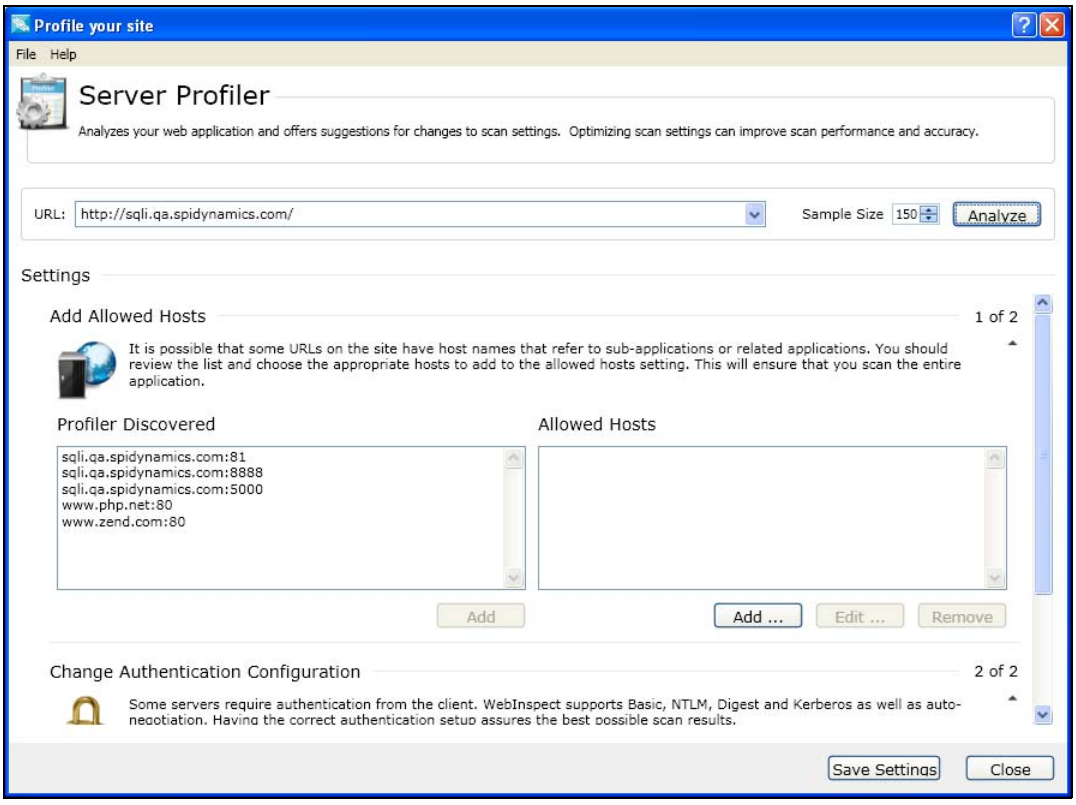
13 Server Profiler (WebInspect Only)

About the Server Profiler Tool

Use the Server Profiler to conduct a preliminary examination of a website to determine if certain WebInspect settings should be modified. If changes appear to be required, the Server Profiler returns a list of suggestions, which you may accept or reject.

For example, the Server Profiler may detect that authorization is required to enter the site, but you have not specified a valid user name and password. Rather than proceed with a scan that would return significantly diminished results, you could follow the Server Profiler's prompt to configure the required information before continuing.

Similarly, your settings may specify that WebInspect should not conduct "file-not-found" detection. This process is useful for websites that do not return a status "404 Not Found" when a client requests a resource that does not exist (they may instead return a status "200 OK," but the response contains a message that the file cannot be found). If the Server Profiler determines that such a scheme has been implemented in the target site, it would suggest that you modify the WebInspect setting to accommodate this feature.



You can use either of two methods to invoke the Server Profiler:

- Launch the Server Profiler as a standalone tool.
- Invoke the Server Profiler automatically when starting a scan.

Launching the Server Profiler as a Tool

To launch the Server Profiler tool:

1. On the WebInspect menu bar, click **Tools** → **Server Profiler**.
2. In the **URL** box, enter or select a URL or IP address.
3. (Optional) If necessary, modify the Sample Size. Large websites may require more than the default number of sessions to sufficiently analyze the requirements.
4. Click **Analyze**.

   The Server Profiler returns a list of suggestions (or a statement that no modifications are necessary).
5. To reject a suggestion, clear its associated check box.
6. For suggestions that require user input, provide the requested information.
7. (Optional) To save the modified settings to a file:
   a. Click **Save Settings**.
   b. Using a standard file-selection window, save the settings to a file in your Settings directory.

Invoking the Server Profiler when Starting a Scan

To launch the Server Profiler when beginning a scan:

1. Start a scan using one of the following methods:
   - On the WebInspect **Start** page, click **Start a Basic Scan**.
   - Click **File** → **New** → **Basic Scan**.
   - Click the drop-down arrow on the **New** icon (on the toolbar) and select **Basic Scan**.
   - On the WebInspect **Start** page, click **Manage Schedule**, click **Add**, and then select **Web Site Scan**.
2. On step 4 of the Scan Wizard (Detailed Scan Configuration), click **Profile** (unless **Run Profiler Automatically** is selected).

   The Profiler returns a list of suggestions (or a statement that no modifications are necessary).
3. To reject a suggestion, clear its associated check box.
4. For suggestions that require user input, provide the requested information.
5. Click **Next**.

14 SmartUpdate

About the SmartUpdate Tool

Use the SmartUpdate tool to download the latest adaptive agents and programs, as well as vulnerability and policy information. Smart Update also ensures that you are using the latest version of WebInspect or WebInspect Enterprise, and it prompts you if a newer version of the product is available for download. New vulnerability checks downloaded via Smart Update are not added automatically to any custom policies you may have created

Caution: For WebInspect Enterprise installations, if Smart Update changes or replaces certain WebInspect Enterprise-related files used by WebInspect, the sensor service may stop and the sensor will display a status of “off line.” You must launch the WebInspect application and restart the service. To do so, choose Configure from the AMP menu, and then click the **Start** button on the **Sensor Service** tab of the *AMP Configuration* window.

WebInspect Smart Update

To manually initiate a Smart Update in WebInspect:

1. Do any of the following:
   - From the toolbar, click **SmartUpdate.**Select **SmartUpdate** from the **Tools** menu
   - Select **Start SmartUpdate** from the WebInspect Start Page.
2. If updates are available, the *Smart Update* window displays up to three separate collapsible panes for downloading the following:
   - New and updated checks
   - WebInspect software
   - Smart Update software

   Select the check box associated with one or more of the download options.
3. To install the updates, click **Download**.

If you download checks without also downloading available new versions of WebInspect, HP will continue to offer updates to your installed knowledgebase for only 10 days. Beyond that period, updates will not be available to you until you download the new WebInspect software.

Checking for Updates Automatically

You can force WebInspect to check for new vulnerabilities or program components every time you open the application. This is the simplest method of ensuring that your SecureBase vulnerabilities database remains accurate and includes the latest information. To enable this option, select **Application Settings** from the **Edit** menu and choose **Smart Update**.

WebInspect Enterprise Smart Update

Each time you log in to the WebInspect Enterprise Administrative Console, it contacts the server and downloads any available console binary updates.

You can obtain updates to the SecureBase, as well as binary updates for WebInspect Enterprise-connected products such as WebInspect, through either a manual or scheduled process.

To manually initiate a Smart Update in WebInspect Enterprise, do either of the following in the Administrative Console:

- Click **SmartUpdate**.
- Select **SmartUpdate** from the **Tools** menu.

15 SQL Injector

About the SQL Injector Tool

SQL injection is a technique for exploiting web applications that use client-supplied data in SQL queries without first removing potentially harmful characters. The SQL Injector supports MS-SQL, Oracle, Postgress, MySQL, and DB2 as database types and also supports multiple language systems including Japanese.

This tool tests for SQL injection vulnerabilities by creating and submitting HTTP requests that may be processed by your SQL server. If your web application allows database records to be updated or created using data supplied by the user, the SQL Injector may create spurious records. To avoid this possibility, do not test against your production database. Instead, use a copy of the database, or use a test account that does not have access to the production data, or exclude from audit any pages that may update or delete data from the database. If these alternatives are not feasible, back up your production database before testing at a time when the site has little or no customer traffic.

Using the SQL Injector

To test for susceptibility to SQL injection:

1. If using a proxy server or if the target site requires authentication, click the **Settings** tab and enter the appropriate information. See SQL Injector Settings on page 134 for additional information.
2. Select **File** → **New**
   - or -
   click the New Request icon.
3. In the **Location** box, type or paste the URL that you suspect is susceptible to SQL injection. See examples below.
   - GET method (query parameters are embedded in the URL):
     http://172.16.61.10/Myweb/MSSQL/Welcome.asp?login=aaa&password=bbb
   - POST method (query parameters are included in message body):
     http://172.16.61.10:80/Myweb/MSSQL/Welcome.asp

     Because the SQL Injector defaults to the GET method, you must also edit POST requests on the **Raw** tab (visible if you select **View** → **Show Request**). The edited request would be similar to the following:

     POST /Myweb/MSSQL/POST/2.asp HTTP/1.1

     User-Agent: Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)

     Host: 172.16.61.10

     Content-Length: 22

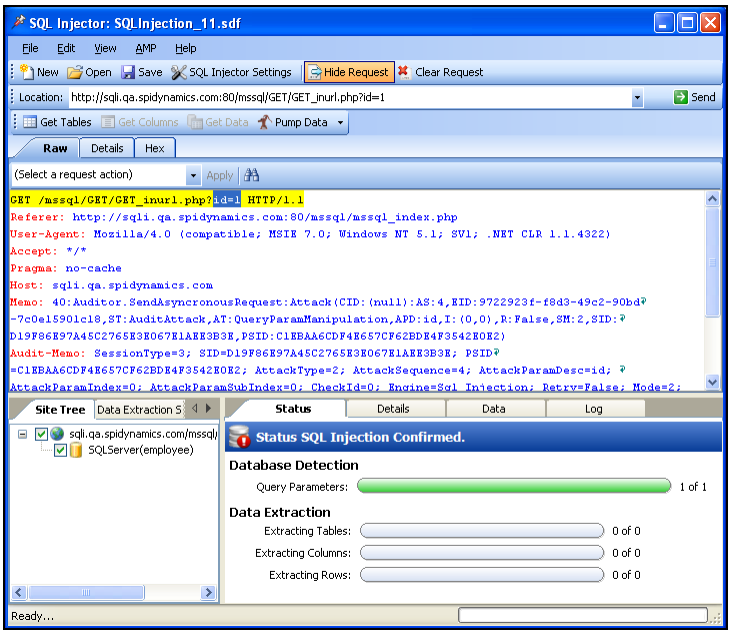
     Content-Type: application/x-www-form-urlencoded

     login=qqq&password=aaa

If WebInspect has detected a SQL injection vulnerability, you can right-click the vulnerable session in WebInspect's navigation pane (or right-click the vulnerable URL on the **Vulnerabilities** tab of the summary pane) and select **Tools** → **SQL Injector** from the shortcut menu.

4 Click **Send**.

If SQL injection is successful, "SQL Injection Confirmed" appears on the **Status** tab and the beginnings of a data hierarchy tree appear on the **Site Tree** tab in the lower left pane.



5 To extract all the data from all tables, click **Pump Data**.

Alternatively, you can selectively investigate tables and columns using the following procedure:

a Select **Get Tables**.

The SQL Injector returns the names of all tables in the targeted database.

b Choose tables by selecting or clearing their associated check box.

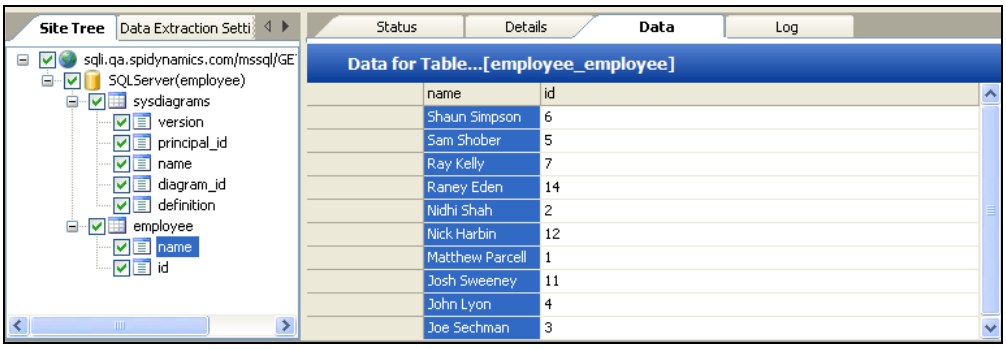
c Click **Get Columns**.

The SQL Injector returns the names of all columns in the selected tables.

d Choose a column by selecting or clearing its associated check box.

e Click **Get Data**.

6 Select a column and click the **Data** tab to column values.



SQL Injector Tabs

Request Pane

The Request pane contains three tabs:

- **Raw** - Displays the text of the HTTP request.
- **Details** - Displays the request segmented by method, request URI, and protocol. Also lists the request header fields and their associated values.
- **Hex** - Displays a hexadecimal representation of the HTTP request.

To toggle the display of the Request pane, click **Show Request**/**Hide Request**.

To delete the request, replacing it with the default http://localhost:80/, click **Clear Request**.

Database Pane

The lower left pane contains two tabs:

- **Site Tree** - Displays the URL, databases, tables, and columns.
- **Data Extraction Settings** - Displays the maximum number of tables, columns, and rows to return when extracting data. These values are extracted from the settings, but can be modified here or in the settings dialog.

Information Pane

The lower right pane contains four tabs:

- **Status** - Displays progress bars for detection and extraction functions.
- **Details** - Displays database information and injectable parameter details.
- **Data** - Displays data extracted from the selected tables and columns.
- **Log** - Displays a synopsis of pertinent functions and the time at which they occurred.

SQL Injector Settings

To modify the SQL Injector settings:

1. Click **Edit** → **Settings**.
2. Select either the **Options**, **Authentication**, or **Proxy** category and enter the settings described in the following sections.
3. Click **OK**.

Options Tab

Timeout in Seconds

Specify the number of seconds that the SQL Injector will wait for a response before terminating the session.

Apply State

If your application uses cookies, URL rewriting, or post data techniques to maintain state within a session, the SQL Injector will attempt to identify the method and modify the response accordingly.

Apply Proxy

If you select this option, the SQL Injector will modify the request according to the proxy settings you specify.

Logging

Select the events you want to log:

- Requests
- Responses
- Errors
- Debug Messages

Log files are stored in xml format in My Documents\SPI dynamics\Tools\SQLInjector\logs.

The beginning of each file name is formatted as YYYY_MM_DD<current-process-id>. The remainder of the name is formatted as follows:

_sqli_debug.log: Contains debugging messages for that session.

_errors.log: Contains errors and exceptions that occurred for that session.

_RequestsResponses.log: Contains all the requests and responses sent and received by the SQL Injector.

Data Extraction

Specify the maximum number of tables, columns, and rows that should be returned when extracting data through a URL that is vulnerable to SQL injection. These values are also displayed in the Database pane on the **Data Extraction Settings** tab. You can change these values using either this tab or the *Settings* dialog.

Also specify the maximum number of concurrent threads that should be used for data extraction.

Inferential/Time-Based Extraction

The SQL Injector can use two different techniques for extracting data when a SQL injection vulnerability is discovered. All attempts are conducted using the inferential technique, which examines the content of the HTTP responses. If this method fails, you can force the tool to use a second technique called time-based extraction. Instead of extracting table data, this method attempts to retrieve the name of the database by sending 4-5 long-running database queries for each character in the database name. Since this can be a rather time-consuming exercise, you can specify the number of characters required to confirm the existence of the SQL injection vulnerability.

Use a macro

Select this option to use a workflow macro; then click  to select, edit, or create a macro.

Database File Path

This read-only text box displays the path to the database created by the SQL Injector tool to store attack data and replicate portions of the attacked database.

Authentication Tab

Authentication Method

If the site does not require authentication, select **None**. Otherwise, select a type from the **Authentication** list:

Authentication	Description
Automatic	Allow the SQL Injector to determine the correct authentication type. Note that automatic detection slows the scanning process. If you know and specify one of the other authentication methods, scanning performance is noticeably improved.
HTTP Basic	A widely used, industry-standard method for collecting user name and password information. The web browser displays a window for a user to enter a previously assigned user name and password. If the web server verifies that the user name and password correspond to a valid user account, a connection is established.
NT LAN Manager (NTLM)	NTLM is an authentication process used by all members of the Windows NT family of products. It uses a challenge/response process to prove the client's identity without requiring that either a password or a hashed password be sent across the network. Use NTLM authentication for servers running IIS.

Authentication Credentials

Type a user ID in the **User name** box and the user's password in the **Password** box. To guard against mistyping, repeat the password in the **Confirm Password** box.

Proxy Tab

Use these settings to access the SQL Injector through a proxy server.

Direct Connection (proxy disabled)

Select this option if you are not using a proxy server.

Auto detect proxy settings

Use the Web Proxy Autodiscovery (WPAD) protocol to locate a proxy autoconfig file and configure the browser's web proxy settings.

Use Internet Explorer proxy settings

Import your proxy server information from Internet Explorer.

Use Firefox proxy settings

Import your proxy server information from Firefox.

Configure a proxy using a PAC file

Load proxy settings from the Proxy Automatic Configuration (PAC) file in the location you specify in the **URL** box.

Explicitly configure proxy

Configure a proxy by entering the requested information.

1. In the **Server** box, type the URL or IP address of your proxy server, followed (in the **Port** box) by the port number (for example, 8080).
2. Select a protocol for handling TCP traffic through a proxy server: SOCKS4, SOCKS5, or standard.
3. If authentication is required, select a type from the **Authentication** list. See Authentication Types on page 44 for a description of the available authentication types.
4. If your proxy server requires authentication, enter the qualifying user name and password.
5. If you do not need to use a proxy server to access certain IP addresses (such as internal testing sites), enter the addresses or URLs in the **Bypass Proxy For** box. Use commas to separate entries.

HTTPS Proxy Settings

For proxy servers accepting HTTPS connections, select **Specify Alternative Proxy for HTTPS** and provide the requested information.

16 SWFScan (WebInspect Only)

About the SWFScan Tool

The HP Web Security Research Group developed SWFScan to help organizations secure applications developed using the Adobe Flash platform. This innovative tool identifies many of the vulnerabilities that affect Flash applications and provides definitive insight on how to remove or avoid them.

SWFScan uniquely supports all versions of Adobe Flash and ActionScript, including ActionScript 2 and 3 (Flash versions 9 and 10).

When you point SWFScan at a Flash file on the Internet or intranet, or load a Flash file from your local computer, SWFScan decompiles the SWF bytecode, generates ActionScript source code, and performs static analysis. You can then generate reports that include:

- Identification of the source code that caused the vulnerability
- Implications of each specific vulnerability
- “Best practice” guidelines to help with remediation

SWFScan also provides additional key information (such as networking calls, external domain requests, etc.) that may be useful for manual inspection of your Flash applications.

Vulnerability Detection

HP SWFScan tests for the following Flash security vulnerabilities. Checks for additional vulnerabilities will be added to SecureBase (through Smart Update) as they are developed.

ActionScript 3 Vulnerabilities Detected by SWFScan

HP SWFScan finds the following vulnerabilities in applications built on Flash 9 and above.

Insecure Programming Practice

- Insecure Security.allowInsecureDomain() usage
- Insecure Security.allowDomain() usage
- Insecure LocalConnection.allowDomain() usage
- Insecure Flash Storage Object usage
- Shared Flash Storage Object usage
- Possible Malicious Activity (LoadBytes)
- Interesting Package/Class/Function Names
  - Potential User Account Information
  - Possible Commerce Information

- Possible Cryptographic Data
- Potential Personal Information
- Possible Application Information
- Potentially Interesting Name Encountered

Insecure Application Deployment

- Complete Flash Application Source Available
- Debugging Information (trace function)
- Debugging Information (Source file disclosure)
- Remote Flash Debugging Enabled

Adobe Best Practices Violation

- Minimum Stage Size For Security Dialogs
- Utilize AllowScript Privileges
- Utilize AllowNetworking Privileges
- Utilize AllowFullscreen Privileges

Information Disclosure

- Possible Social Security Number
- Possible Credit Card Number Disclosure
- Internal IP Disclosure
- Path Disclosure (win32)
- Path Disclosure (unix)
- PGP Public Key Block Detected
- PGP Private Key Block Detected
- MD5 Hash Detected
- SHA-0/SHA-1 Hash Detected
- SQL Query Detected
- LDAP Query Detected
- XPath Query Detected
- Data Connection String
  - Generic
  - MSSQL ODBC Trusted Connection
  - MSSQL OleDb Trusted Connection
  - MSSQL via IP Address
  - MSSQL .NET DataProvider Standard Connection or Sybase .NET DataProvider
  - MSSQL .NET DataProvider Trusted Connection
  - MSSQL .NET DataProvider via IP Address

- Access and Oracle ODBC -- Standard Security for MS Access and ODBC Oracle Driver
- Access ODBC Workgroup - System Database
- Access OleDb with MS Jet Workgroup - System Database
- Access OleDb with MS Jet With Password
- Oracle ODBC New Microsoft Driver
- Oracle ODBC Old Microsoft Driver
- Oracle OleDb Microsoft Driver and Oracle Driver - possible trusted connection
- Oracle OleDb Oracle Driver - Trusted Connection
- Oracle .NET DataProvider from Microsoft and Oracle - Standard Connection
- Oracle .NET DataProvider from Microsoft and Oracle - Trusted Connection
- IBM DB2 ODBC without DSN and OleDb IBM Driver
- IBM DB2 OleDb Microsoft Driver
- IBM DB2 .NET DataProvider from IBM
- MySQL ODBC MyODBC Driver - local database
- MySQL ODBC MyODBC Driver - remote database
- MySQL .NET DataProvider from CoreLab
- Sybase ODBC Sybase System 12 (12.5) ODBC Driver
- Sybase ODBC Sybase System 11 ODBC Driver or Intersolv 3.10 ODBC Driver
- Sybase ODBC SQL Anywhere
- Sybase OleDb Sybase Adaptive Server Enterprise (ASE)
- Informix ODBC DSN INFORMIX 3.30 ODBC Driver
- Informix ODBC without DSN INFORMIX 3.30 ODBC Driver
- Informix OleDb IBM Informix OleDb Provider

ActionScript 1 and 2 Vulnerabilities Detected by SWFScan

HP SWFScan finds the following vulnerabilities in applications built on Flash 8 and below.

Possible Cross-Site Scripting

- Identifying undefined global variables
- Identifying injection points for cross-site scripting vectors
  - getURL
  - XML.load
  - loadMovie/loadMovieNum
  - htmlText
  - ExternalInterface

Dangerous functions accepting user supplied data

- loadVariables/LoadVars
- System.Security.loadPolicyFiles
- Insecure Programming Practice
- Insecure Security.allowInsecureDomain() usage
- Insecure Security.allowDomain() usage
- Insecure LocalConnection.allowDomain() usage
- Insecure Flash Storage Object usage
- Shared Flash Storage Object usage
- Possible Malicious Activity (LoadBytes)
- Interesting Package/Class/Function Names
  - Potential User Account Information
  - Possible Commerce Information
  - Possible Cryptographic Data
  - Potential Personal Information
  - Possible Application Information
  - Potentially Interesting Name Encountered

Insecure Application Deployment

- Debugging Information (trace function)
- Debugging Information (Source file disclosure)
- Remote Flash Debugging Enabled

Information Disclosure

- Possible Social Security Number
- Possible Credit Card Number Disclosure
- Internal IP Disclosure
- Path Disclosure (win32)
- Path Disclosure (unix)
- PGP Public Key Block Detected
- PGP Private Key Block Detected
- MD5 Hash Detected
- SHA-0/SHA-1 Hash Detected
- SQL Query Detected
- LDAP Query Detected
- XPath Query Detected

- Data Connection String
  - Generic
  - MSSQL ODBC Trusted Connection
  - MSSQL OleDb Trusted Connection
  - MSSQL via IP Address
  - MSSQL .NET DataProvider Standard Connection or Sybase .NET DataProvider
  - MSSQL .NET DataProvider Trusted Connection
  - MSSQL .NET DataProvider via IP Address
  - Access and Oracle ODBC -- Standard Security for MS Access and ODBC Oracle Driver
  - Access ODBC Workgroup - System Database
  - Access OleDb with MS Jet Workgroup - System Database
  - Access OleDb with MS Jet With Password
  - Oracle ODBC New Microsoft Driver
  - Oracle ODBC Old Microsoft Driver
  - Oracle OleDb Microsoft Driver and Oracle Driver - possible trusted connection
  - Oracle OleDb Oracle Driver - Trusted Connection
  - Oracle .NET DataProvider from Microsoft and Oracle - Standard Connection
  - Oracle .NET DataProvider from Microsoft and Oracle - Trusted Connection
  - IBM DB2 ODBC without DSN and OleDb IBM Driver
  - IBM DB2 OleDb Microsoft Driver
  - IBM DB2 .NET DataProvider from IBM
  - MySQL ODBC MyODBC Driver - local database
  - MySQL ODBC MyODBC Driver - remote database
  - MySQL .NET DataProvider from CoreLab
  - Sybase ODBC Sybase System 12 (12.5) ODBC Driver
  - Sybase ODBC Sybase System 11 ODBC Driver or Intersolv 3.10 ODBC Driver
  - Sybase ODBC SQL Anywhere
  - Sybase OleDb Sybase Adaptive Server Enterprise (ASE)
  - Informix ODBC DSN INFORMIX 3.30 ODBC Driver
  - Informix ODBC without DSN INFORMIX 3.30 ODBC Driver
  - Informix OleDb IBM Informix OleDb Provider

Analyzing Flash Files

You can use SWFScan as a standalone tool or as an integrated component of WebInspect.

Analyze a Flash file using SWFScan as a standalone tool

1 Launch SWFScan:

Click **Start** → **All Programs** → **HP** → **HP Security Toolkit** → **SwfScan**.

2 Specify the Flash file (.swf) you want to analyze.

- In the **Path or URL** combo box, enter or select the full path to a Flash file and click 

on the SWFScan toolbar.

  - or -

- Click **File** → **Open**, select a Flash file from a local storage device, and click **Open**.

SWFScan loads and decompiles the selected file.

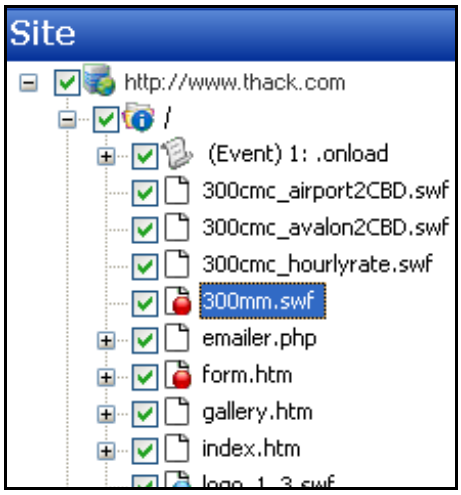
3 Click 

on the SWFScan toolbar.

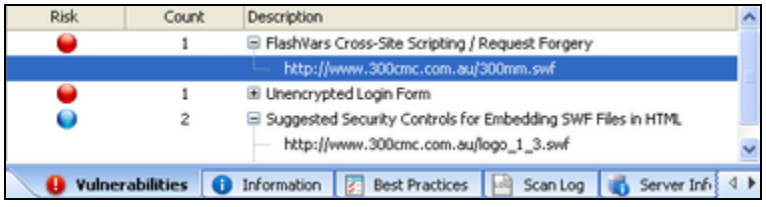
Analyze a Flash file using SWFScan as an integrated component of WebInspect

1 Do one of the following while or after conducting a scan:

- Locate a Flash file (.swf) in the navigation pane, then right-click the file name and select **Tools** → **SWFScan** from the shortcut menu.



- Locate a Flash vulnerability on the **Vulnerabilities** tab, then right-click an associated URL and select **Tools** → **SWFScan** from the shortcut menu.



The SWFScan tool launches and loads the decompiled source code.

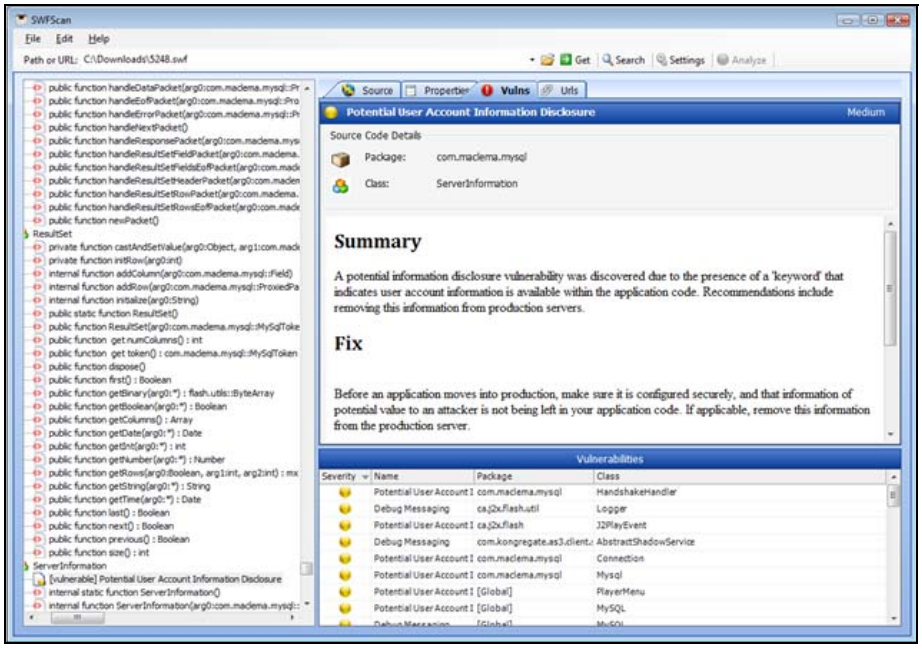
2 Click Analyze on the SWFScan toolbar.

WebInspect analyzes Flash files if this function is enabled in the Default settings (located in **Scan Settings** → **Content Analyzers**). However, SWFScan offers more functionality and control by allowing you to configure independent settings, export source code and discovered URLs, and generate individual reports for each file. You can also search the source code (or specific portions of it).

Examining Results

SWFScan displays a list of detected vulnerabilities in the lower right pane.

Click an item in the list to display information about the vulnerability and to locate (in the left pane) the module in which the vulnerability was detected.



Searching Source Code

You can search for specific text strings or text strings that match the regular expression you specify.

1 In the **Search For** box, enter a text string or regular expression.

2 To find only those occurrences matching the case of the text string or regular expression, select the **Match Case** check box.

3 To identify the string as a regular expression, select **RegEx**.

4 Choose the specific area that you want to search.

For ActionScript 2 files:

- All Source Code—The decompiled source code.
- Specific Movie Clip—Select a clip from the list.

- Specific Frame—Select a clip and a frame.
- Specific Class—Select a class from the list.
- Specific Method—Select a class and a method.

For ActionScript 3 files:

- All Source Code—The decompiled source code.
- Specific Package—Select a package from the list.
- Specific Class—Select a package and class.
- Specific Method—Select a package, class, and method.

5 Click **Search**.

The results appear on the **Search Results** tab, with matches highlighted.

SWFScan Settings

Use the following procedure to configure SWFScan settings.

1 Click 

on the SWFScan toolbar.

2 Click any of the four tabs presented, which are described below.

AS2 Exclusions

You can exclude ActionScript 2 packages (namespaces) from analysis by selecting the **Enabled** check box associated with a particular package.

Clear the check box if you want to include the package in your analysis.

To add an exclusion to the list:

a Click **Add**.

b On the Add *Exclusion Rule* window, enter a name for the rule and a regular expression that describes the package.

c Click **OK**.

You can also edit or remove any rules that you add, but you cannot modify the default rule (the Flash Standard Library).

AS3 Exclusions

You can exclude ActionScript 3 packages (namespaces and classes) from analysis by selecting the **Enabled** check box associated with a particular package or class.

Clear the check box if you want to include the package or class in your analysis.

To add packages and classes to the exclusion list:

a Click **Add**.

b On the *Add Exclusion Rule* window, enter a name for the rule and a regular expression that describes the package or class.

c Click **OK**.

You can also edit or remove any rules that you add, but you cannot modify the default rules.

Proxy

Select from the following options.

- **Direct Connection (proxy disabled)**—Select this option if you are not using a proxy server.
- **Auto detect proxy settings**—If you select this option, SWFScan uses the Web Proxy Autodiscovery Protocol (WPAD) to locate a proxy autoconfig file and configure the browser's web proxy settings.
- **Use Internet Explorer proxy settings**—Select this option to import your proxy server information from Internet Explorer.
- **Use Firefox proxy settings**—Select this option to import your proxy server information from Firefox.
- **Configure a proxy using a PAC file**—Select this option to load proxy settings from a Proxy Automatic Configuration (PAC) file. Then specify the file location in the URL box.
- **Explicitly configure proxy**—Select this option to configure a proxy manually, and then enter the requested information.
- **Specify Alternative Proxy for HTTPS**—For proxy servers accepting HTTPS connections, select the **Specify Alternative Proxy for HTTPS** check box and provide the requested information.

Checks

This tab lists all attacks that check for specific vulnerabilities in the decompiled code. You can enable or disable a check by selecting or clearing its associated check box.

3 When complete, click **OK**.

Changed settings are persisted, but cannot be applied retroactively. To analyze a Flash file after changing settings, you must click 

.

17 Web Brute

About the Web Brute Tool

This tool will determine if your users are employing user names and passwords that an unauthorized intruder might be able to guess easily. For example, if one of your customers is accessing your website by using a username of “customer” and a password of “password,” you might want to warn that user about his susceptibility and suggest that he change his password and/or username.

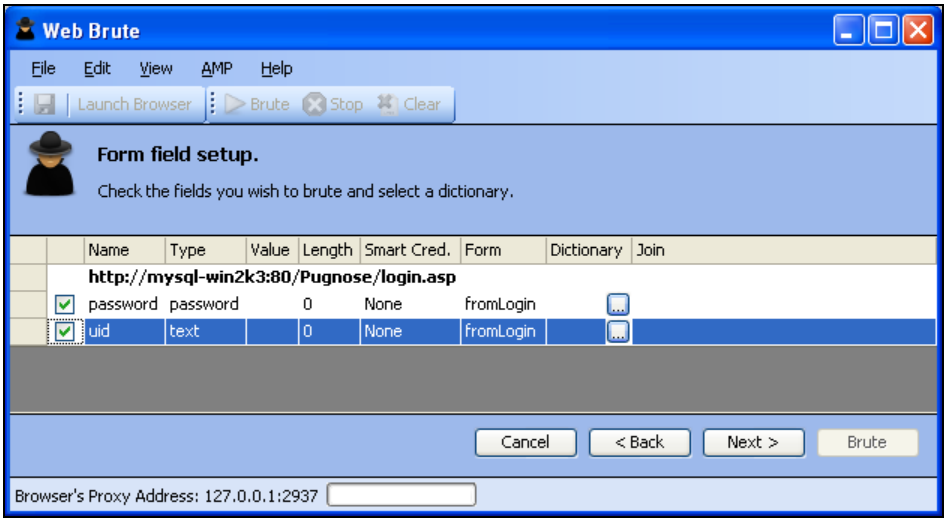
Web Brute will attempt a “brute force” attack of a login form or authentication page, using two prepared lists of user names and passwords.

This is an intrusive attack and can break into a secure area. Brute force attacks are intended for testing purposes only, and should not be used against unsuspecting websites.

Mounting a Brute Force Attack

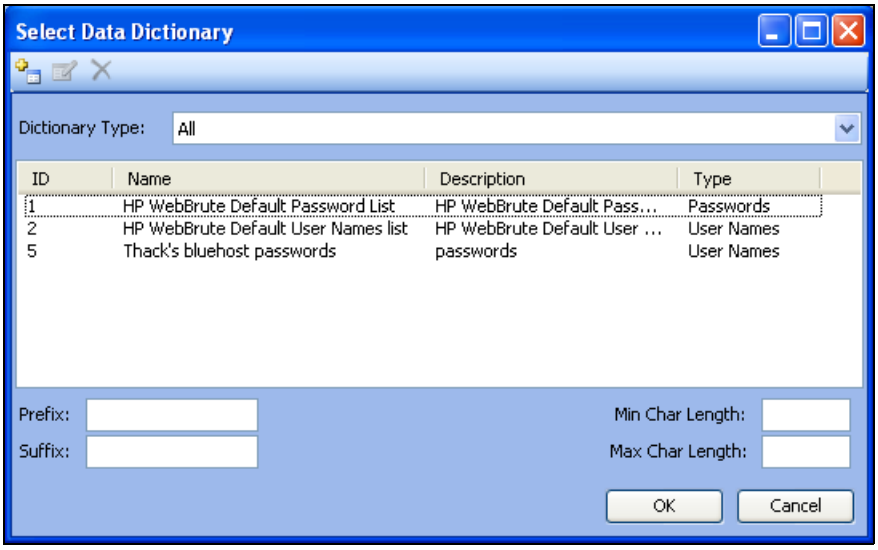
To use a brute force authentication attack:

1. On the WebInspect menu bar, click **Tools** → **Web Brute**.
2. In the **Enter URL** box, type the URL of the site you want attack and click **Next**.
3. Select the authentication type used by the target site. See Authentication Types on page 44 for a description of the available authentication types.
4. If necessary, use the **Domain** box to specify the domain that should be used for authentication. Web Brute will prefix this string to each user name that it submits. Do not include a backslash.
5. Click **Next**.
6. If you selected **Web Form** in step 3, a web browser opens. If necessary, navigate to the login page.
7. On Web Brute’s **Form Field Setup** panel, select (check) the fields you want to brute force. If you already know the value that should be entered for a field, remove the check mark, double-click the cell in the **Value** column for that field, and enter the value.



8 For fields you have selected (checked), click [...] in the **Dictionary** column to select a list of names or passwords to be submitted.

The *Select Data Dictionary* window appears, listing all currently defined dictionaries. You can limit the display of dictionary names by selecting an entry in the **Dictionary Type** list.



These dictionaries are in a database that is not directly accessible. To create your own dictionary or merge a list into an existing dictionary, see Creating and Importing Lists on page 149.

9 Select a list.

10 (Optional) Enter the following:

- **Prefix**—A string that will be added to the beginning of each entry in the list.
- **Suffix**—A string that will be added to the end of each entry in the list.
- **Min Char Length**—The minimum number of characters allowed for each entry; entries that are shorter will not be submitted.
- **Max Char Length**—The maximum number of characters allowed for each entry; entries that are longer will not be submitted.

11 Click **OK**.

12 Repeat step 7 through step 11 for each authentication field to be submitted.

13 If you want to "join" two or more lists, click the **Join** column associated with each list.

If a list of user names is joined with a list of passwords, then Web Brute will submit user names with passwords in the order in which they appear in the lists. That is, the first name in the user name list will be submitted with the first password in the password list, the second name will be submitted with the second password, etc.

If the two lists are not joined, then Web Brute submits each user name with all passwords. This feature is used most often for web form authentication where the user must re-enter the password. In this case, Web Brute would use two lists, but the password list would be specified for both the "password" and "confirm password" fields. You would then join these fields, forcing the same password to be submitted for each field.

14 To modify the parameters that Web Brute uses during an authentication attack, select **Edit** → **Settings**. See Web Brute Settings on page 150 for more information.

15 Click **Next**.

16 To see a list of failed name/password attempts (in addition to successful attempts), select **Show Failed**.

17 Click **Brute**.

Web Brute attacks the site and displays the results. If you double-click a result (either successful or failed), Web Brute opens the HTTP Editor, allowing you to inspect both the HTTP request and response.

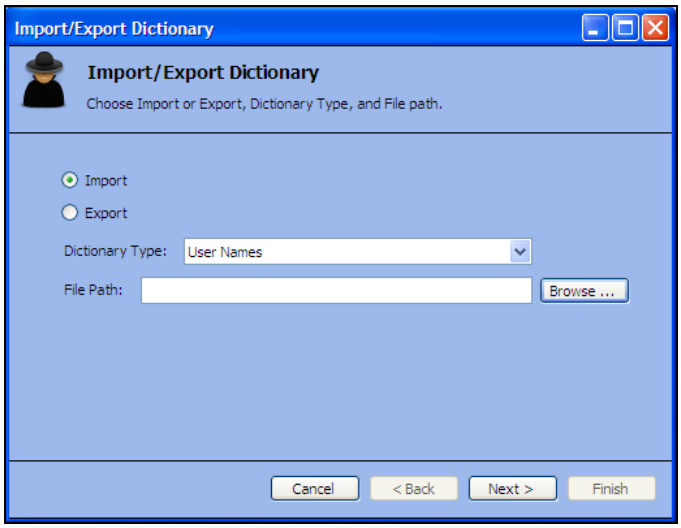
Creating and Importing Lists

To use your own list of passwords or user names, you must first create a list and then import it into Web Brute as a "dictionary," using the following procedure:

1 Create a text file where each entry is delimited by a carriage return and line feed.

2 Click **File** → **Import/Export Dictionary**.

3 On the *Import/Export Dictionary* window, select **Import**.



4 From the **Dictionary Type** list, select either **User Names**, **Passwords**, or **E-mails**.

5 Click **Browse** and select the file containing the list you want to import.

6 Click **Next**.

7 On the *Import Dictionary* window, specify a name for the dictionary and enter a description.

8 Click **Next**.

9 Click **Finish**.

Exporting Dictionaries

Use the following procedure to create a text file from a Web Brute dictionary:

1 Click **File** → **Import/Export Dictionary**.

2 On the *Import/Export Dictionary* window, select **Export**.

3 In the **File Path** box, enter the path and name of the text file in which the dictionary contents will be saved, or click **Browse** and use the *Save As* window to specify the name and path.

4 Click **Next**.

5 On the *Export Dictionary* window, select a dictionary type from the list.

6 Select a dictionary.

7 Click **Next**.

8 Click **Finish**.

9 Click **Done**.

Web Brute Settings

To modify the Web Brute settings:

1 Click **Edit** → **Settings**.

2 Select either the **Options**, **Authentication**, or **Proxy** tab and enter the settings described in the following sections.

3 Click **OK**.

Options

Timeout in seconds

Enter the number of seconds that Web Brute will wait for a response. If a response is not received during this period, Web Brute will resend the request, up to the number of times specified in the Retry Count setting.

Retry Count

Enter the number of times that Web Brute will resend a request that has not been acknowledged.

Apply State

If you select this option, Web Brute will attempt to maintain state during the procedure.

Apply Proxy

If you select this option, Web Brute will use the settings on the Proxy tab to connect to the target site (if the Direct Connection option is not selected).

Logging

Select the types of messages that should be logged.

Max Concurrent Threads

Enter or select the number of requests that Web Brute may send before requiring a response to the first request.

Advanced HTTP Parsing

Most web pages contain information that tells the browser what character set to use. This is accomplished by using the Content-Type response header (or a META tag with an HTTP-EQUIV attribute) in the HEAD section of the HTML document. For pages that do not announce their character set, you can specify which character set the Web Brute should use.

Authentication

If required, select an authentication method and provide credentials. The methods are:

- **None**—Select this option if the site does not require authentication.
- **Automatic Authentication**—This allows Web Brute to determine the correct authentication type.
- **HTTP Basic Authentication**—This is a widely used, industry-standard method for collecting user name and password information. Normally, a web browser displays a window for a user to enter a previously assigned user name and password, also known as credentials. The web browser then attempts to establish a connection to a server using the user's credentials.
- **NTLM Authentication**—NTLM (NT LanMan) is an authentication process that is used by all members of the Windows NT family of products. Like its predecessor LanMan, NTLM uses a challenge/response process to prove the client's identity without requiring that either a password or a hashed password be sent across the network.

After selecting an authentication method, enter a user name and password. To avoid typographic errors, re-enter the password in the **Confirm Password** box.

Proxy

Use these settings to access the Web Brute through a proxy server.

Direct Connection (proxy disabled)

Select this option if you are not using a proxy server.

Auto detect proxy settings

Use the Web Proxy Autodiscovery (WPAD) protocol to locate a proxy autoconfig file and configure the browser's Web proxy settings.

Use Internet Explorer proxy settings

Import your proxy server information from Internet Explorer.

Use Firefox proxy settings

Import your proxy server information from Firefox.

Configure a proxy using a PAC file

Load proxy settings from the Proxy Automatic Configuration (PAC) file in the location you specify in the URL box.

Explicitly configure proxy

Select this option to access the Internet through a proxy server and then enter the requested information.

1. In the **Server** box, type the URL or IP address of your proxy server, followed (in the Port box) by the port number (for example, 8080).
2. Select a protocol for handling TCP traffic through a proxy server: SOCKS4, SOCKS5. or standard.
3. If authentication is required, select a type from the Authentication list; see Authentication Types on page 44 for a description of the available authentication types.
4. If your proxy server requires authentication, enter the qualifying user name and password.
5. If you do not need ko use a proxy server to access certain IP addresses (such as internal testing sites), enter the addresses or URLs in the Bypass Proxy For box Use commas to separate entries.

Specify Alternative Proxy for HTTPS

For proxy servers accepting HTTPS connections, select Specify Alternative Proxy for HTTP and provide the requested information.

18 Web Discovery

About the Web Discovery Tool

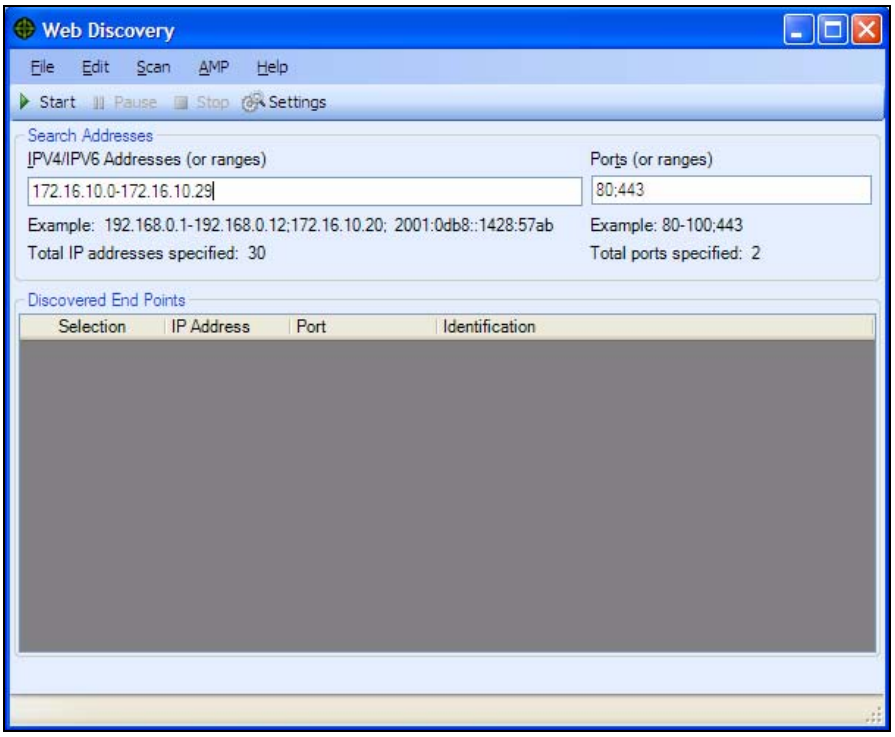
Use Web Discovery to find all open hosts in your enterprise environment.

Web Discovery sends packets to all the open ports (in a range of IP addresses and ports that you specify), searches the server's response for specific information, and then displays the results. There are two predefined packets included with Web Discovery: Web Server and SSL Web Server. They both contain the following HTTP request:

GET / HTTP/1.0

Web Discovery searches the HTTP response for the string "HTTP"; if it finds the string, it displays the IP address, port number, and the text "WebServer," followed by the results of a regular expression search designed to reveal the server's name and version number.

You can save the list of discovered servers in a text file.



Discovering Sites

To discover sites using Web Discovery:

1. In the **IPV4/IPV6 Addresses (or ranges)** box, type one or more IP addresses (or a range of IP addresses).

   - Use a semicolon to separate multiple addresses.
     Example: 172.16.10.3;172.16.10.44;188.23.102.5
   - Use a dash or hyphen to separate the starting and ending IP addresses in a range.
     Example: 10.2.1.70-10.2.1.90.

   Note: IPV6 addresses must be enclosed in brackets. For example:

   • http://[::1]
   WebInspect scans "localhost."

   • http://[fe80::20c:29ff:fe32:bae1]/subfolder/
   WebInspect scans the host at the specified address starting in the "subfolder" directory.

   • http://[fe80::20c:29ff:fe32:bae1]:8080/subfolder/
   WebInspect scans a server running on port 8080 starting in "subfolder."

2. In the **Ports (or ranges)** box, type the ports you want to scan.

   - Use a semicolon to separate multiple ports.
     Example: 80;8080;443
   - Use a dash or hyphen to separate the starting and ending ports in a range.
     Example: 80-8080.

3. To modify Web Discovery settings, click **Settings**. See Web Discovery Settings on page 154 for more information.

4. Click **Start** to initiate the discovery process.

   Results display in the Discovered EndPoints area.

5. Click an entry in the **IP Address** column to view that site in a browser.

6. Click an entry in the **Identification** column to open the *Settings Properties* window and view the raw request and response.

To save the list of discovered servers:

1. Click **File** → **Export**.

   If you export the data to a `.csv` file, the IP addresses become default SSC project names. You can edit those projects and their associated data in Microsoft Excel. In WebInspect Enterprise, you can then import the projects into SSC. For more information, see the *WebInspect Enterprise User Guide* or the WebInspect Enterprise online Help.

2. Use the standard file-selection window to name and save the file.

Web Discovery Settings

To modify the Web Discovery settings:

1. Click **Edit** → **Settings**.
2. Enter the settings described in the following sections.
3. Click **OK**.

Select Protocols

Choose the packet type you want to send by selecting or clearing the check box next to the protocol name.

Logging

Select the elements you want to log:

- **Log Open Ports**—Logs all available ports found open on the host; saves only web server information in log file.
- **Log Services**—Logs all services identified during the discovery.
- **Log Web Servers**—Logs web servers identified.

Enter the file location in the **Log To** box, or click the browse button  and use the standard file-selection window to specify the file in which the log entries should be recorded.

Connectivity

Set the following timeouts (in milliseconds):

- **Connection**—The period of time that Web Discovery will wait before stopping a port scan when no information has been returned from an IP address.
- **Send**—When sending a message to the remote IP endpoint, the transmission is divided into smaller packets. If the IP endpoint does not acknowledge receipt of a sent packet within the specified period of time, the socket is closed and the discovery for that endpoint reports no services.
- **Receive**—When sending a message to the remote IP endpoint, the transmission is divided into smaller packets. If the Web Discovery tool does not receive the sent packet within the specified period of time, the socket is closed and the discovery for that endpoint reports no services.

Adjust the number of open sockets using the **Sockets** box. A higher number of open sockets results in a faster scan. However, a setting that exceeds a server's threshold may result in false positives

If you are using Windows XP with Service Pack 2 (SP2), your **Open Sockets** setting is set to 10. If you increase this setting, you will increase the likelihood that the scanner will fail to detect servers. This occurs because Microsoft has limited the number of open sockets to 10.

19 Web Form Editor

About the Web Form Editor Tool

Most web applications contain forms composed of input controls (text boxes, buttons, drop-down lists, etc.). Users generally "complete" a form by modifying its controls (such as entering text or checking boxes) before submitting the form to an agent for processing. Usually, this processing will lead the user to another page or section of the application. For example, after completing a login form, the user will proceed to the application's beginning page.

Some sites (such as WebInspect's example banking application zero.webappsecurity.com) contain many different forms for completing a variety of transactions. If WebInspect is to navigate through all possible links in the application, it must be able to submit appropriate data for each form.

With the Web Form Editor, you can create or modify a file containing the names of all input controls and the associated values that need to be submitted during a scan of your website. These entries are categorized by URL, so even if different controls on different pages have the same name, the Web Form Editor can discriminate between them. Alternatively, you can designate a form entry as "global," meaning that its value will be submitted for any input control having the same name attribute, regardless of the URL at which it occurs.

During a scan, if WebInspect encounters an input control whose name attribute is not matched in the file you create, it will submit a default value (12345).

If you are using a proxy server, the WebForm Editor will not use the default settings from WebInspect. You must first configure Internet Explorer to use the desired proxy.

There are two ways to create a list of form values:

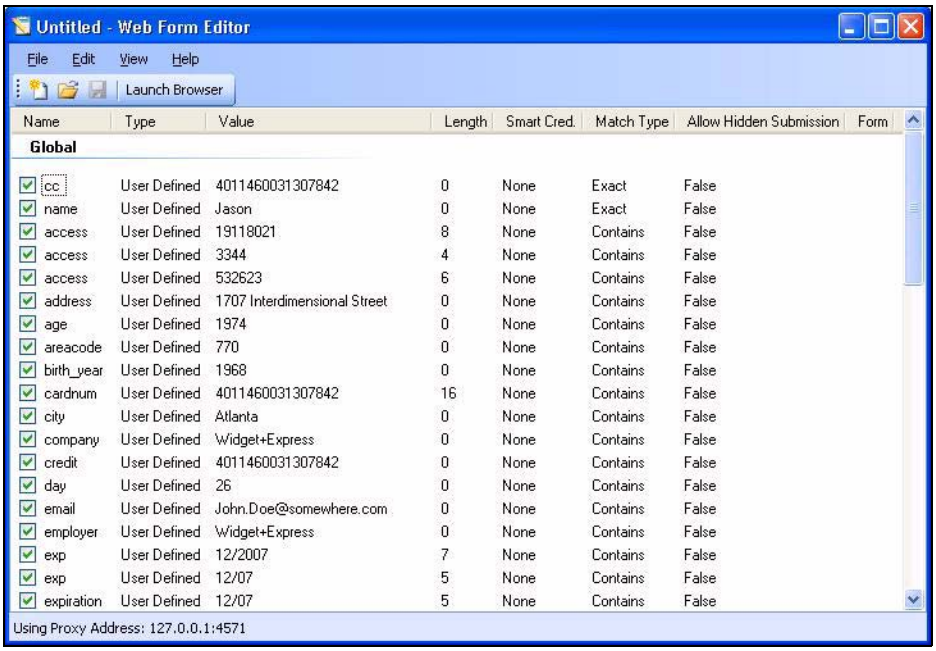
- Create the list manually.
- Record the values as you navigate through the application.

Manually Creating a Web Form List

Use the following procedure to create a Web Form list manually.

1 Click **Tools** → **Web Form Editor**.

The *Web Form Editor* window appears.



The Web Form Editor loads a prepackaged default file.

a To load a different file, select **File** → **Open**.

b To create a new file, select **File** → **New**.

2 Do one of the following:

- To add a web form value, right-click anywhere in the Web Form Editor's work area and select **Add Global Form Input** from the shortcut (pop-up) menu.
- To modify a value, right-click an entry and select **Modify** from the shortcut (pop-up) menu.

The *Add User-Defined Input* or the *Modify Input* window appears.

3 In the **Name** box, type (or modify) the name attribute of the input element.

4 In the **Length** box, enter either:

- the value that must be specified by the size attribute, or
- zero, for input elements that do not specify a size attribute.

For example, to submit data for the following HTML fragment…

<INPUT TYPE="password" NAME="accessID" MAXLENGTH="6">

…you must create an entry consisting of accessID (Name) and specify a size of "6" (Length).

5 In the **Value** box, type the data that should be associated with the input element (for example, a password).

6 Use the **Match** list to specify how the scanner should determine if this entry qualifies to be submitted for a particular input control. The options are:

- **Exact**—The name attribute of the input control must match exactly the name assigned to this entry.
- **Starts with**—The name attribute of the input control must begin with the name assigned to this entry.
- **Contains**—The name attribute of the input control must contain the name assigned to this entry.

7 Programmers sometimes use input controls with type= “hidden” to store information between client/server exchanges that would otherwise be lost due to the stateless nature of HTTP. Although the Web Form Editor will collect and display the attributes for hidden controls, the scanner will not submit values for hidden controls unless you select **Allow Hidden Submission**.

8 Click **Add** (or **Modify**).

9 If necessary, you can assign additional attributes by right-clicking an entry and using the shortcut (pop-up) menu.

- To submit the form value for any input control having the same name attribute, regardless of the URL at which it occurs, choose **Make Global**.
- To remove an entry, choose **Unselect**. This clears the check mark and removes the entry from processing, but does not delete it from the file.
- To activate an entry, choose **Select**. This creates a check mark and includes the entry for processing.
- To delete an entry, choose **Delete**.
- To designate an entry as a smart credential, select either **Smart Credential Username** or **Smart Credential Password**.

  When recording web form values, you will often encounter a log-on form requiring you to enter a user name and password. You can safely use your own user name and password, provided that you designate those entries as “Smart Credentials” before saving the file. Your actual password and user name are not saved.

  When scanning the page containing the input control associated with this entry, the scanner will substitute the password specified in the product’s Authentication options. This would be a known user name and password that does not require security. Alternatively, if no user name or password is specified, the scanner will submit the string “FormFillText.”
- If you select **Mark As Interactive Input**, the scanner will pause the scan and display a window prompting the user to enter a value for this entry (if the scan options include the settings **Prompt For Web Form Values During Scan** and **Only Prompt Tagged Inputs**).

  It is not necessary to tag passwords with **Mark As Interactive Input**.

Recording Web Form Values

The Web Form Editor serves as a proxy that handles HTTP traffic between a browser and a target website. By default, it uses the local IP address 127.0.0.1 and any available port. However, you can specify a different IP address and port by clicking **Edit** → **Settings**.

Use the following procedure to capture names and values of input controls on a website.

1 To create a list of form values, select **File** → **New** (or click the New icon on the toolbar).

2 To add form values to an existing list, select **File** → **Open** (or click the Open icon on the toolbar) and choose a file using the standard file-selection dialog.

3 Click **Launch Browser**.

4 Using the browser’s **Address** bar, enter or select a URL and navigate to a page containing a form.

5 Complete the form and submit it (usually by clicking a button such as **Log In**, **Submit**, **Go**, etc.).

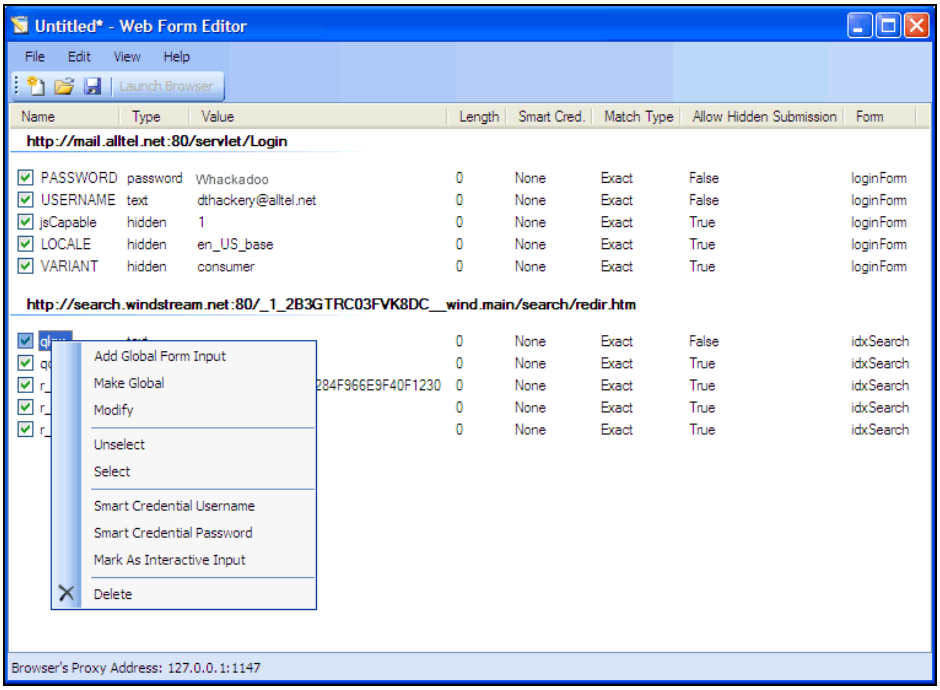
6 Navigate to additional pages and submit forms until you have traversed all the links you wish to follow.

7 The Web Form Editor displays a list of name and value attributes for all input controls found in all forms on the pages you visited.

For example, the first two entries in the following illustration were derived from the following HTML fragment…

```
<form name="loginForm" action="/servlet/Login" method="POST">
<input type="password" size="16" name="PASSWORD">
<input type="text" size="16" name="USERNAME" value="">
<input type="SUBMIT" value="Submit"></form>
```

…and the user entered his name and password.



8 If necessary, you can modify items by right-clicking an entry and using the shortcut (pop-up) menu.

- To submit the form value for any input control having the same name attribute, regardless of the URL at which it occurs, choose **Make Global**.
- To edit an entry, select **Modify**.
- To add an entry, select **Add Global Form Input**. A Global entry is one not associated with a specific URL.
- To remove an entry, choose **Unselect**. This removes the entry from processing, but does not delete it from the file.
- To delete an entry, choose **Delete**.
- To designate an entry as a smart credential, select either **Smart Credential Username** or **Smart Credential Password**.
- To force the scanner to pause and display a window prompting the user to enter a value for this entry, select **Mark As Interactive Input**.

  When a scanner encounters an HTTP or JavaScript form, it will pause the scan and display a window that allows you to enter values for input controls within the form, provided that the scanner's option to **Prompt For Web Form Values** is selected. However, if the scanner's option to **Only Prompt Tagged Inputs** is also selected, WebInspect will not pause for user input unless a specific input control has been designated **Mark As Interactive Input** (except for passwords, which always cause the scanner to pause for input).

9 Click **File** → **Save** (or **Save As**).

Importing a Web Form File

You can import a file that was designed and created for earlier versions of WebInspect, QAInspect, or DevInspect and convert it to a file that can be used by the current Web Form Editor.

1. Click **File** → **Import**.

   The *Convert Web Form Values* window appears.

2. Click the browse button … next to **Select File To Import**.
3. Using a standard file-selection window, locate the XML file created by an earlier version of the Web Form Editor.
4. Click the browse button … next to **Select Target File**.
5. Using a standard file-selection window, specify a file name and location for the converted file.
6. Click **OK**.

Scanning with a Web Form File

When scanning a site, you specify which Web Form file you want to use by selecting **Auto-fill web forms during crawl** (step 4 of a Basic Scan) and then selecting a file.

You can also designate a specific file as the default by using the following procedure:

1. On the WebInspect menu bar, click **Edit** → **Default Settings**.

   The *Default Settings* window opens.

2. In the **Scan Settings** section, select **Method**.
3. In the **Scan Behavior** group, select **Auto-fill Web Forms During Crawl**.
4. To select a previously recorded file:

   a. Click the browse button ….

   b. Using the standard file-selection window, select a file containing the web form value you want to use and click **Open**.

   c. (Optional) Edit the contents by right-clicking an entry and selecting an option from the context menu.

5. To record web form values:

   a. Click **Create New Web Form Values**.

   b. Click the Web Form Editor's **File** menu and select **New.**

   c. Click **Launch Browser**.

   d. See Recording Web Form Values on page 159 for further instructions.

6. To edit web form values for the selected file:

   a. Click **Edit Current Web Form Values**.

   b. See Recording Web Form Values on page 159 for further instructions.

Web Form Editor Settings

To modify the Web Form Editor settings:

1. Click **Edit** → **Settings**.
2. Select either the **General** or **Proxy** category and enter the settings described in the following sections.
3. Click **OK**.

General

Proxy Listener

The Web Form Editor serves as a proxy that handles HTTP traffic between a browser and a target website. By default, it uses the local IP address 127.0.0.1 and any available port. However, you can specify a different IP address and port by selecting **Edit** → **Settings**.

To avoid the possibility of specifying a port that is already in use, select **Automatically Assign Port**.

Advanced HTTP Parsing

Most web pages contain information that tells the browser what character set to use. This is accomplished by using the Content-Type response header (or a META tag with an HTTP-EQUIV attribute) in the HEAD section of the HTML document. For pages that do not announce their character set, you can specify which character set the Web Form Editor should use by selecting an entry from the **Assumed 'charset' Encoding** list.

Proxy

Use these settings to access the Web Form Editor through a proxy server.

Direct Connection (proxy disabled)

Select this option if you are not using a proxy server.

Auto detect proxy settings

Use the Web Proxy Autodiscovery (WPAD) protocol to locate a proxy autoconfig file and configure the browser's web proxy settings.

Use Internet Explorer proxy settings

Import your proxy server information from Internet Explorer.

Use Firefox proxy settings

Import your proxy server information from Firefox.

Configure a proxy using a PAC file

Load proxy settings from the Proxy Automatic Configuration (PAC) file in the location you specify in the **URL** box.

Explicitly configure proxy

Configure a proxy by entering the requested information.

1. In the **Server** box, type the URL or IP address of your proxy server, followed (in the **Port** box) by the port number (for example, 8080).
2. Select a protocol for handling TCP traffic through a proxy server: SOCKS4, SOCKS5, or standard.
3. If authentication is required, select a type from the **Authentication** list. See Authentication Types on page 44 for a description of the available authentication types.
4. If your proxy server requires authentication, enter the qualifying user name and password.
5. If you do not need to use a proxy server to access certain IP addresses (such as internal testing sites), enter the addresses or URLs in the **Bypass Proxy For** box. Use commas to separate entries.

HTTPS Proxy Settings

For proxy servers accepting HTTPS connections, select **Specify Alternative Proxy for HTTPS** and provide the requested information.

Web Form Logic

When crawling a web application and submitting web form values, WebInspect analyzes the entries in the web form values file to determine if a value should be submitted. The logic for determining a match is represented in the following table, ordered from “most preferred” to “least preferred.”

Rules for Matching Web Form Values

Page-specific form values	Exact Match. Name exact match. Length exact match.	The specific web page, web form name, and value length detected on the crawled web page exactly match a single record in the webformvalues.xml selected for the scan.
	Partial Match. Name-only match. Length allows wildcard.	The specific web page and web form name detected on the crawled web page match a single record in the webformvalues.xml selected for the scan. The field length associated with that form value allows for submission to any field input length (wildcard field length match).
Global form values	Exact Match. Name exact match. Length exact match.	The web form name and value length detected on the crawled web page match a single record in the Global web form values section of the webformvalues.xml selected for the scan.
	Partial Match 1. Name exact match. Length allows wildcard.	The web form name detected on the crawled web page exactly matches a form name found in the global values section of the webformvalues.xml selected for the scan. The field length associated with that form value allows for submission to any field input length (wildcard field length match).

Rules for Matching Web Form Values (cont'd)

	Partial Match 2. Field name starts with Name value. Length exact match.	A web form value in the file partially matches the field name found. All characters in the web form value match the beginning of the web page field name and the field length detected on the crawled web page match the record in the Global web form values section of the webformvalues.xml selected for the scan.
	Partial Match 3. Field name starts with Name value. Length allows wildcard.	A web form value in the file partially matches the field name found. All characters in the web form value match the beginning of the web page field name and the field length for the record allows for submission to any field length (wildcard field length match).
	Partial Match 4. Name value included in field name. Length exact match.	A web form value in the file partially matches the field name found. All characters in the web form value match a portion of the web page field name and the field length for the record allows for submission to any field length (wildcard field length match).
	Partial Match 5. Name value included in field name. Length allows wildcard.	A web form value in the file partially matches the field name found. All characters in the web form value match a portion of the web page field name and the field length for the record allows for submission to any field length (wildcard field length match).
No match	Field name has no exact or partial matches to web form values.	No web form value match was found. Submit the specified default value (Default).
No default value	The web form values file has no default value specified.	No web form value match was made and the default value is not in the web form values file. Submit "not found."

20 Web Fuzzer

About the Web Fuzzer Tool

"Fuzzing" is an automated software testing technique that generates and submits random or sequential data to various areas of an application in an attempt to uncover security vulnerabilities. For example, when searching for buffer overflows, a tester can simply generate data of various sizes and send it to one of the application entry points to observe how the application handles it.

The Web Fuzzer lets you run several automated tests for common classes of web application security vulnerabilities such as SQL injection, format strings, cross-site scripting, path traversal, odd characters, and buffer overflows, as well as protocol implementation problems.

Using the Web Fuzzer

To use the Web Fuzzer:

1. Click **Edit** → **Server**.
2. Enter the fully qualified domain name or IP address of a website, along with other server configuration information, and click **OK**.
3. Click **Edit** → **Settings**.
4. Configure the settings and click **OK**. For more information, see Web Fuzzer Settings on page 171.
5. To create a session, click **Session** and select either **Create** or **Raw Create**.
   a. If you select **Create**, Web Fuzzer displays a tabbed property sheet that identifies each section of an HTTP request and allows you to replace an HTTP element with generated data or with text that you enter. This structured approach is recommended for novice users. For detailed information, see Using the Session Editor on page 168.
   b. If you select **Raw Create**, Web Fuzzer displays a standard GET request formatted as regular text. You can edit the request. You can also place the cursor anywhere in the request, right-click to invoke a shortcut menu, and then insert a generator that will fuzz the selected HTTP element. If you highlight any portion of the request, the highlighted portion will be replaced by the generator.

Fuzzer Generators

Generator	Function
Number	Inserts a whole number, within the range you specify, in each request; you specify the starting and ending number, the increment, and the number of times to loop through the series.
ASCII	Inserts one ASCII character, within the range you specify, in each request; you specify the starting and ending character, and the number of times to loop through the series.

Fuzzer Generators **(cont'd)**

Generator	Function
Character	Generates the character you specify and inserts multiple numbers of the character into each request; you specify the minimum and maximum number of characters, and an increment.
Decimal Number	Inserts a fractional number, within the range you specify, in each request; you specify the starting and ending number, the increment, and the number of times to loop through the series.
Guid	Inserts a random Globally Unique Identifier (a 128-bit number) in each request; you specify the number of requests.
WordList Reader	Inserts a string from a text file you specify; the number of requests is determined by the number of paragraphs in the file; all characters in the paragraph are inserted.
SQL Injection	Inserts a string from a text file you specify; the number of requests is determined by the number of paragraphs in the file; all characters in the paragraph are inserted. The default file (sqlinjections.txt) contains the following two entries: ' or 1=1 ' or like '%
Text	Inserts the text you specify in a single request.
Cross-Site Scripting	Inserts a string from a text file you specify; the number of requests is determined by the number of paragraphs in the file; all characters in the paragraph are inserted. The default file (xssinjections.txt) contains the following entry: <script>alert('test')</script>
Method	Inserts a method (GET, POST, PUT, etc.); you specify the protocol version (0.9, 1.0, 1.1, or all).

6 After creating the request, click **OK**.

7 You can use filters so that only those server responses meeting criteria you specify will be displayed.

8 On the *Web Fuzzer Request* window, click **Start**.

The **Sessions** area lists each session (request and response) generated by the tool.

9 To examine the results, click an entry in the **Sessions** list.

- The HTTP request for the selected session appears in the **Request** area.
- The server's response appears on both the **Browser View** and **Raw Response** tabs.

10 To edit the request that you constructed, select a session in the **Sessions** group, then click the **Session** menu and choose either **Edit** or **Raw Edit**.

Filters

A filter consists of a name, description, and rule. The rule is a regular expression that defines the text you want to locate in a particular section of the server's response. For example, if you want to display only those responses that contain the word "error" in the response body and where the response also specifies a status code between 500 and 599, then use the following rule:

[STATUSCODE]5\d\d AND [BODY]\serror\s

Use the following notation to specify a response section:

- [HEADERS]
- [STATUSLINE]
- [STATUSCODE]
- [STATUSDESCRIPTION]
- [ALL]
- [SETCOOKIES]
- [BODY]

You access the *Filters* window by selecting **Filters** → **Edit**.

In addition to enabling a specific rule, you must also enable the use of rules in general by selecting **Filters** → **Enable**.

Creating a Filter

To create a filter:

1. Click **Add**.

   The tool creates a rule named Default Rule.
2. Modify the Name, Description, and Rule.
3. Click **Apply** to save the definition.

Using a Filter

To use a filter in a session:

1. Select a filter from the **Filters** list.
2. Select the **Enable** check box.

Deleting a Filter

To delete a filter:

1. Select a filter from the **Filters** list.
2. Click **Delete**.

Editing a Filter

To edit a filter:

1. Select a filter from the **Filters** list.
2. Modify the Name, Description, or Rule.
3. Click **Apply** to save the modifications.

Using the Session Editor

Use this tabbed property sheet to change specific sections of an HTTP request. You can replace an HTTP element with text that you type or paste into a text box, or you can insert a generator that will create multiple requests containing generated data.

To use the Session Editor:

1. Click a tab.
2. You can either:
   - Edit the data appearing in text boxes, or
   - Select the **Use Generator** check box and click **Generator** to insert a generator.
3. To change other areas, click a different tab.
4. After configuring the areas you want to change, click **OK**.
5. When you return to the *Web Fuzzer* window, click **Start**.

Creating a Query String

To create a query string:

1. Click **Add**.

   The text “name=value” appears in the list, representing the query string you are creating.
2. Click the **Name** tab.

   You can edit the parameter named “name” or you can substitute a generator for it (select the **Use Generator** check box and click **Generator**).
3. Click the **Separator** tab.

   You can edit the character that separates the name from the value (usually an equals sign) or you can substitute a generator for it (select the **Use Generator** check box and click **Generator**).
4. Click the **Value** tab.

   You can edit the value in the equation or you can substitute a generator for it (select the **Use Generator** check box and click **Generator**).
5. Click the **Format** tab.

   You can edit the order in which the equation elements appear, or you can introduce characters between them.
6. In the **Name Value Separator** group, you can edit the character that separates parameters (usually an ampersand) or you can substitute a generator for it (select the **Use Generator** check box and click **Generator**).
7. To add another parameter, click **Add** and repeat step 2 through step 6.

Session Editor Tabs

Method Tab

The GET method is specified by default. You can replace it with any text, or you can insert the Method generator.

Path Tab

You can fuzz three elements related to the path: the name of the file, the file extension, and the character that designates a directory level (usually the forward slash /). You can replace these elements with any text, or you can insert generators.

Query Tab

Some HTTP requests include a query string, with each parameter formatted as parameter=value and separated by an ampersand (&). The resource is separated from the query by a delimiter character (usually a question mark, although other characters can be used depending on the application). For example:

http://www.website.com/category.cfm?model_ID=0&category_ID=12.

Version Tab

The version indicates to the server which HTTP version to use for interpreting the request. Valid versions are 0.9, 1.0 and 1.1. The version information is formatted as “HTTP/version,” which is a name-value pair separated by a forward slash (/). You can fuzz all three sections: Protocol, Separator, and Version. You can also fuzz the format by rearranging the order or introducing extraneous characters.

Headers Tab

Headers contain basic information issued by the client to help the server or application handle the request. Common headers are Host and User-Agent. Each header is defined by using the “name: value” syntax. This name-value structure also can be separated into four fuzzing opportunities.

Creating Headers

To create headers:

1. Click **Add**.

   The text “name:value” appears in the list, representing the header you are creating.

2. Click the **Name** tab. You can edit the parameter named “name” or you can substitute a generator for it (select the **Use Generator** check box and click **Generator**).

3. Click the **Separator** tab. You can edit the character that separates the name from the value (usually a colon) or you can substitute a generator for it.

4. Click the **Value** tab. You can edit the “value” text or you can substitute a generator for it.

5. Click the **Format** tab. You can edit the order in which the header elements appear, or you can introduce characters between them.

6. In the **Name Value Separator** group, you can edit the character that separates headers or you can substitute a generator for it.

7. To add another header, click **Add** and repeat step 2 through step 6.

Cookies Tab

Cookies are special headers that contain parameters used by the application to manage users and states. The format of a cookie definition is:

Cookie: name=value;name=value

Each parameter is a name-value pair that can be independently fuzzed.

Creating Cookies

To create cookies:

1. In the **Cookies** group, click **Add**.

   "Cookie:" appears in the list, representing the cookie you are creating.

2. Click **Cookie:** (in the Cookies list) and then click **Add** (in the **Cookie** group).

   The text "name=value" appears.

3. In the **Cookie** group, click the **Cookie Name** tab. You can edit the name or you can substitute a generator for it.

4. Click the **Separator** tab. You can edit the character that separates the name from the value (usually an equals sign) or you can substitute a generator for it.

5. Click the **Value** tab. You can edit the "value" text or you can substitute a generator for it.

6. Click the **Format** tab. You can edit the order in which the header elements appear, or you can introduce characters between them.

7. In the **Name Value Separator** group, you can edit the character that separates headers or you can substitute a generator for it.

8. To add another cookie, repeat step 1 through step 7.

Post Data Tab

While a query can be appended to the Request-URI, post data is added to the end of the request. The format is similar to the URI query and is mostly used with the POST method. When post data are used, the request must contain a Content-Length header that indicates the size of the post data. You can fuzz not only the post data, but also the Content-Length value to test how the server or application handles the differences.

When fuzzing the HTTP request message, you affect two main layers of the application environment: server protocol implementation and web application.

Creating POST Data

To create post data:

1. Click **Add**.

   The text "name=value" appears in the list, representing the post data you are creating.

2. Click the **Name** tab. You can edit the parameter named "name" or you can substitute a generator for it.

3. Click the **Separator** tab. You can edit the character that separates the name from the value (usually a colon) or you can substitute a generator for it.

4. Click the **Value** tab. You can edit the "value" text or you can substitute a generator for it.

5. Click the **Format** tab. You can edit the order in which the header elements appear, or you can introduce characters between them.

6. In the **Name Value Separator** group, you can edit the character that separates headers or you can substitute a generator for it.

7. To add another post data element, click **Add** and repeat step 2 through step 6.

Web Fuzzer Settings

To modify the Web Fuzzer settings:

1. Click **Edit** → **Settings**.
2. Select either the **General** or **Proxy** category and enter the settings described in the following sections.
3. Click **OK**.

General

Enable Filters

Select this option to enable filter support.

Auto scroll view

Select this option to enable automatic scrolling in the **Sessions List** view. This will force the view to scroll down to the latest session automatically.

Show ToolTips

Select this option to enable the display of tool tips when you hover your mouse pointer over certain controls.

Sockets

Enter the maximum number of sockets and the sockets send timeout (in seconds).

Protocol Compliance

Select **Enforce Content-Length** to automatically adjust the Content-Length value in the request when needed. If this feature is enabled, you cannot fuzz the content-length header.

Select **Enforce Host header** to include the Host header in all requests. If this feature is enabled, you cannot fuzz the host header.

Proxy

Use these settings to access the Web Fuzzer through a proxy server.

Direct Connection (proxy disabled)

Select this option if you are not using a proxy server.

Auto detect proxy settings

Use the Web Proxy Autodiscovery (WPAD) protocol to locate a proxy autoconfig file and configure the browser's Web proxy settings.

Use Internet Explorer proxy settings

Import your proxy server information from Internet Explorer.

Use Firefox proxy settings

Import your proxy server information from Firefox.

Configure a proxy using a PAC file

Load proxy settings from the Proxy Automatic Configuration (PAC) file in the location you specify in the **URL** box.

Explicitly configure proxy

Configure a proxy by entering the requested information.

1. In the **Server** box, type the URL or IP address of your proxy server, followed (in the **Port** box) by the port number (for example, 8080).
2. Select a protocol for handling TCP traffic through a proxy server: SOCKS4, SOCKS5, or standard.
3. If authentication is required, select a type from the **Authentication** list. See Authentication Types on page 44 for a description of the available authentication types.
4. If your proxy server requires authentication, enter the qualifying user name and password.
5. If you do not need to use a proxy server to access certain IP addresses (such as internal testing sites), enter the addresses or URLs in the **Bypass Proxy For** box. Use commas to separate entries.

HTTPS Proxy Settings

For proxy servers accepting HTTPS connections, select **Specify Alternative Proxy for HTTPS** and provide the requested information.

21 Unified Web Macro Recorder

About the Unified Web Macro Recorder Tool

When you open the Login Macro Recorder or the Workflow Macro Recorder from the **Tools** menu or when you configure a scan in WebInspect or WebInspect Enterprise, you launch the Unified Web Macro Recorder.

In this chapter, the term “scanner” is often used instead of “WebInspect and WebInspect Enterprise” where the information applies to both products.

A login macro is a recording of the events that occur when you access and log in to a website. You can subsequently instruct the HP scanner to begin a scan using this recording. A workflow macro is a recording of login steps (as needed) and specific URLs on a site.

WebInspect and WebInspect Enterprise include one “Unified” Web Macro Recorder tool. By default, it uses event-based functionality and Firefox browser technology to record new macros.

The separate Event-Based IE Compatible Web Macro Recorder that was provided in some earlier versions is no longer *directly* accessible in WebInspect or WebInspect Enterprise *menus*. In effect, it is hidden. However, the Unified Web Macro Recorder allows you to indirectly open, play back, and edit existing event-based macros that were created in earlier versions of WebInspect or WebInspect Enterprise (or Assessment Management Platform—AMP), and create new macros, using the earlier Event-Based IE Compatible Web Macro Recorder. For information about how you can access and use macros recorded with the Event-Based IE Compatible Web Macro Recorder of earlier versions of WebInspect and WebInspect Enterprise (or AMP), see Chapter 22, Event-Based IE Compatible Web Macro Recorder.

HP strongly recommends that you use the Unified Web Macro Recorder to record all new login macros and workflow macros.

Guided Scan is the preferred method for scanning a site using WebInspect or WebInspect Enterprise. WebInspect also continues to provide the legacy scan now known as Basic Scan, and WebInspect Enterprise continues to provide the legacy Web Site Scan.

The Web Macro Recorder can be launched in several ways—while configuring a Guided Scan, a Basic Scan in WebInspect, a Web Site Scan in WebInspect Enterprise, or outside of any scan in what is known as “stand-alone” mode. For more information, see Accessing the Web Macro Recorder on page 177.

Macros that were *recorded* in a Guided Scan, a Basic Scan in WebInspect, or a Web Site Scan in WebInspect Enterprise can be *used* in any of those scans.

By default, the Web Macro Recorder uses underlying Firefox browser technology to record and play macros. It can also use Internet Explorer browser technology (also referred to here as IE technology) to record and display web traffic data.

Notes:

- The Web Macro Recorder does not support the recording of Flash or Silverlight applications.
- The TruClient technology used in the Web Macro Recorder is an adaptation of the Ajax TruClient technology originally developed for use with HP LoadRunner and HP Performance Center. It does not incorporate or support all the capabilities of the fully-featured version in those products.

- When you play a macro, the HP scanner does not send any cookie headers that may have been incorporated in the recorded macro.
- If a URL is in a macro, the request is always sent when the macro is played, regardless of any exclusion rules in scan settings.
- When launching the Web Macro Recorder, you may receive the following error message:

  "Exc in ev handl: TypeError: this.oRoot.enable is not a function."

  This can occur if the McAfee SiteAdvisor is installed. Simply acknowledge the message and continue.

Login Macros

A login macro is a recording of the activity that is required to access and log in to a website or web application, typically by entering a user name and password and clicking a button such as Log In or Log On. When you configure a scan, you usually specify a previously recorded login macro or record a new one at the time for the scan to use.

To prevent the scanner from terminating prematurely if it gets logged out of your application, a login macro should also specify at least one logout condition that definitively indicates that a logout has occurred. During a scan, the scanner can get logged out for a variety of reasons, including:

- Normal logout driven by the target site
- An error condition in the target site such as a timeout
- An error in the macro itself, such as an invalid parameter

Specifying a logout condition as part of the login macro makes it unnecessary for users to manually log back in, perhaps repeatedly, when unexpected logouts occur during a scan. When scanning a site, the scanner analyzes every target site response to determine the state. If the scanner determines at any time that it is logged out, it runs the login macro to log back in, and then it resumes crawling or auditing the site at the point where the logout occurred.

As the final step in recording a login macro, the Unified Web Macro Recorder uses sophisticated analysis to try to *automatically* detect a logout condition and specify it in the login macro. In most cases you do not have to identify a logout condition manually. However, you can add or edit logout conditions as described in this chapter.

Additionally, for Guided Scan login macros, to prevent potential timing issues in macro playback, you can specify that a particular object must appear in the initial Web page that appears immediately after login.

You can specify multiple logout conditions, and if any of them are met, WebInspect or WebInspect Enterprise plays the login macro to log the scanner back in and resume the scan where it left off.

Workflow Macros

A workflow macro is a recording of the login steps (as needed) and the specific URLs to which you manually navigate on a site. When you configure a Guided Scan, a Basic Scan in WebInspect, or a Web Site Scan in WebInspect Enterprise, you specify a previously recorded workflow macro or record a new one at the time for the scan to use. WebInspect or WebInspect Enterprise audits only the URLs that are recorded in the workflow macro and does not take any hyperlinks encountered during the audit. This type of macro is used most often to focus on a particular subsection of an application. In terms of the macro recording process, the essential differences from login macros are that:

- Workflow macros include only the specific URLs to which a user navigated while recording them, and upon replay workflow macros access only those URLs.

Workflow macros do not require logout conditions, so the macro recorder user interface excludes logout condition functionality when recording workflow macros.

If your website requires authentication, do not record login steps in a workflow macro. Instead, record a separate login macro to log in to your website.

Upgrade Impacts

WebInspect and WebInspect Enterprise versions 10.00 and later include only one directly accessible "Unified" Web Macro Recorder tool. It enhances the functionality of the three web macro recorders that were available in earlier versions, and its enhancements make login macro recording more automatic, more complete, and more successful.

Guided Scan is the preferred method for scanning a site using WebInspect or WebInspect Enterprise. WebInspect also continues to provide the legacy Basic Scan, and WebInspect Enterprise continues to provide the legacy Web Site Scan.

If you have upgraded from an earlier version of WebInspect or WebInspect Enterprise, review the following aspects of the Unified Web Macro Recorder, as compared to the web macro recorders of versions 9.30 and earlier:

- The Unified Web Macro Recorder includes and enhances the TruClient Web Macro Recorder functionality of earlier versions, and by default it uses this enhanced functionality to record a new macro. The automatic detection of logout conditions was significantly improved from earlier versions. As a result, usually you should not need to manually identify a logout condition as part of recording a login macro.

  Macros that were recorded using the TruClient Web Macro Recorder in any earlier version can be used in a Guided Scan, a Basic Scan in WebInspect, or a Web Site Scan in WebInspect Enterprise version 10.00 or later.

- When using Internet Explorer browser technology (also referred to here as IE technology), the Unified Web Macro Recorder can open macros that were created in the Traffic-Mode Web Macro Recorder of earlier versions of WebInspect or WebInspect Enterprise (or AMP). For more information, see Opening Macros Recorded with the Traffic-Mode Web Macro Recorder on page 176.

  Also, if the Web Macro Recorder cannot successfully record a new macro using the default Firefox browser technology, it automatically switches to IE technology to record the macro. IE technology can also be manually selected. For more information, see Internet Explorer Browser Technology on page 188.

- The Unified Web Macro Recorder does not support opening existing macros that were created in the Event-Based IE Compatible Web Macro Recorder tool of earlier versions of WebInspect or WebInspect Enterprise (or AMP). However, in version 10.00 and later you can indirectly access the Event-Based IE Compatible Web Macro Recorder tool to open and edit existing event-based macros and even to create new ones. For more information, see About the Unified Web Macro Recorder Tool on page 173 and Opening Macros Recorded with the Event-Based IE Compatible Web Macro Recorder on page 176. For recording new event-based macros, HP strongly recommends using the default Firefox technology of the Unified Web Macro Recorder.

- Login and workflow macros created in any of the web macro recorders in any version of WebInspect or WebInspect Enterprise have the file extension `.webmacro`. When you open any macro recorded using any of the types of web macro recorder, WebInspect or WebInspect Enterprise uses information in the macro to determine the appropriate type of macro recorder and functionality to use, within the limitations stated above.

Opening Macros Recorded with the Traffic-Mode Web Macro Recorder

Note that all of the information in this section applies to both workflow macros and login macros that were recorded using the formerly separate Traffic-Mode Web Macro Recorder tool used in earlier versions of WebInspect or WebInspect Enterprise (or AMP).

The Traffic-Mode Web Macro Recorder tool is not provided in WebInspect or WebInspect Enterprise versions 10.00 or later. However, the Unified Web Macro Recorder, when using its built-in IE technology, allows you to open, play back, and edit existing traffic-mode macros created in earlier versions, and create new ones. When recording new macros, the Unified Web Macro Recorder first uses event-based functionality based on Firefox technology by default. If that fails for some reason, the Web Macro Recorder automatically switches to using its IE technology as an alternative recording method. This displays web traffic data in the Web Macro Recorder interface.

The **Rendering engine** button in the toolbar at the top of the Web Macro Recorder allows you to see and specify whether the underlying technology used by the Web Macro Recorder is based on Firefox (the recommended option) or Internet Explorer.

When a macro was created using the Traffic-Mode Web Macro Recorder of earlier WebInspect and WebInspect Enterprise versions and you open it in any of the following ways, it opens using IE technology:

- When configuring a Guided Scan
- When configuring a Basic Scan in WebInspect
- When configuring a Web Site Scan in WebInspect Enterprise
- In the stand-alone Web Macro Recorder (see Accessing the Web Macro Recorder on page 177)
- From Windows Explorer

In all cases you can edit the macro after you open it when the Web Macro Recorder is using IE technology, and you can use the macro in a Guided Scan, a Basic Scan in WebInspect, or a Web Site Scan in WebInspect Enterprise.

Opening Macros Recorded with the Event-Based IE Compatible Web Macro Recorder

The Event-Based IE Compatible Web Macro Recorder tool is not directly accessible as a separate tool in WebInspect or WebInspect Enterprise version 10.00 or later. However, these versions include a copy of the tool so that you can open existing login macros that were created using the tool in earlier versions. The capabilities that are available for such an existing login macro depend on how it is accessed, as follows:

- When configuring a Guided Scan, you can select the login macro and use it in the scan, but you cannot open or edit it.
- When configuring a Basic Scan in WebInspect or a Web Site Scan in WebInspect Enterprise, you can select and optionally edit the login macro in the Event-Based IE Compatible Web Macro Recorder, and use it in the scan.
- In the stand-alone Unified Web Macro Recorder (see Accessing the Web Macro Recorder on page 177), when you try to open the login macro, the macro recorder opens a dialog box that asks if you would like to “switch to event mode.”
  - If you answer **Yes**, the macro opens in the Event-Based IE Compatible Web Macro Recorder and you can edit it.
  - If you answer **No**, the dialog box closes and the macro does not open.
- You can select and edit the login macro in the Event-Based IE Compatible Web Macro Recorder if you do any of the following:
  - Open the login macro by double-clicking it from Windows Explorer.

— In WebInspect default scan settings for site authentication, click **Edit** → **Default Scan Settings**, click **Authentication**, and in the Site Authentication section select **Use a login macro for forms authentication.**In the WebInspect Enterprise Scan Wizard, click **Advanced Settings**, click **Method** under SCAN SETTINGS, and select **Use a login macro for forms authentication** (for existing macros only).

Once the Event-Based IE Compatible Web Macro Recorder is open, you can create new login macros with it. However, HP strongly recommends that you create new macros using the Unified Web Macro Recorder, which uses Firefox technology by default and IE technology as necessary.

For more information about the Event-Based IE Compatible Web Macro Recorder tool, see About the Unified Web Macro Recorder Tool on page 173.

Accessing the Web Macro Recorder

The Web Macro Recorder can be launched in several ways—while configuring a Guided Scan, a Basic Scan in WebInspect, a Web Site Scan in WebInspect Enterprise, or outside of a scan in what is known as "stand-alone" mode. For login macros and workflow macros, the following sections describe how you can record a new macro or select (and optionally edit) an existing macro that was recorded in WebInspect or WebInspect Enterprise version 10.00 or later.

Login Macros

You can record a new login macro or select (and optionally edit) an existing login macro that was recorded in WebInspect or WebInspect Enterprise version 10.00 or later in the following ways:

- When configuring a Guided Scan, specify that the target site requires a login macro, and click **Create** to record a new login macro or select (and optionally edit) an existing login macro.
- When configuring a Basic Scan in WebInspect or a Web Site Scan in WebInspect Enterprise, in step 2 select **Site Authentication** and record a new login macro or select (and optionally edit) an existing login macro.
- In WebInspect default scan settings (**Edit** → **Default Scan Settings**), click **Authentication**, select **Use a login macro for forms authentication**, and record a new login macro or select (and optionally edit) an existing login macro.
- On the WebInspect toolbar, click **Tools** → **Login Macro Recorder** to run the Web Macro Recorder in stand-alone mode, and record a new login macro or open (and optionally edit) an existing login macro.
- In the WebInspect Enterprise Scan Wizard, click **Advanced Settings**, click **Method** under SCAN SETTINGS, and select **Use a login macro for forms authentication** (for existing macros only).
- In WebInspect Enterprise, on the Administrative Console toolbar, click **Tools** → **Login Macro Recorder** to open the Web Macro Recorder in stand-alone mode, and record a new login macro or open (and optionally edit) an existing login macro.
- Click **Start** → **All Programs** → **HP** → **HP Security Toolkit** → **Login Macro Recorder** to run the Login Macro Recorder in stand-alone mode, and record a new login macro or open (and optionally edit) an existing login macro.
- From Windows Explorer, navigate to a particular recorded login macro and double-click it to open it in the Web Macro Recorder in stand-alone mode, and record a new login macro or open (and optionally edit) an existing login macro.

Workflow Macros

You can record a new workflow macro or select (and optionally edit) an existing workflow macro that was recorded in WebInspect or WebInspect Enterprise version 10.00 or later in the following ways:

- When configuring a Guided Scan, specify that the **Scan Type** is **Workflows** and later, in the **Workflows** → **1. Manage Workflows** step, record a new workflow macro or import (and optionally edit) an existing workflow macro.
- When configuring a Basic Scan in WebInspect, in step 1 select **Workflow-Driven Scan** and click **Record** or **Manage** to record a new workflow macro or select (and optionally edit) an existing workflow macro.
- When configuring a Web Site Scan in WebInspect Enterprise, in step 1 select **Workflow-Driven Scan** and click **Import** or **Manage** to select an existing workflow macro.
- In WebInspect Enterprise, on the Administrative Console toolbar, click **Tools** → **Workflow Macro Recorder** to open the Web Macro Recorder in stand-alone mode, and record a new workflow macro or open (and optionally edit) an existing workflow macro.
- Using the HP Security Toolkit, click **Start** → **All Programs** → **HP** → **HP Security Toolkit** → **Workflow Macro Recorder** to run the Workflow Macro Recorder in stand-alone mode, and record a new workflow macro or open (and optionally edit) an existing workflow macro.

Recording or Editing a Macro

This section describes the tasks involved in interactively recording or editing login macros and workflow macros, using the Web Macro Recorder in stand-alone mode (that is, launched from the **Tools** menu), or as it is invoked when recording or editing a macro during configuration of a Guided Scan, a Basic Scan in WebInspect, or a Web Site Scan in WebInspect Enterprise.

For information about accessing the Web Macro Recorder under a variety of circumstances, see Accessing the Web Macro Recorder on page 177.

HP strongly recommends initially using the default Firefox technology of the Web Macro Recorder to record a macro. However, if Firefox has not worked for you to successfully record a macro, you can try using IE technology, which displays web traffic data in the Web Macro Recorder interface as you record and play the macro. To record a macro using IE technology, go to Using IE Technology to Record Web Traffic on page 189 and return to this procedure when instructed to do so.

In the Web Macro Recorder, step-by-step guidance is provided near the top of the screen in a yellow instruction bar. Above that, when you begin the specific process to record or edit a macro for Guided Scan, a Basic Scan in WebInspect, a Web Site Scan in WebInspect Enterprise, or stand-alone Web Macro Recorder operation, a toolbar with the following buttons in the Record/Edit Login Macro group appears, except as noted:

- **New**. Starts creating a new macro.
- **Import** (Guided Scan only). Allows you to select an existing macro to play and edit.
- **Open** (Basic Scan for WebInspect, Web Site Scan for WebInspect Enterprise, and stand-alone Web Macro Recorder only). Allows you to select an existing macro to play and edit.
- **Export** (Guided Scan only). Saves the current macro under the same name or a new name.
- **Save** (Basic Scan for WebInspect, Web Site Scan for WebInspect Enterprise, and stand-alone Web Macro Recorder only). Saves the current macro under the same name or a new name.

- **Parameters Editor**. See Parameters Editor on page 194. (This option is not available for a macro that uses IE technology in the Web Macro Recorder or for a macro that was recorded using the Traffic-Mode Web Macro Recorder tool from earlier versions of WebInspect, WebInspect Enterprise, or AMP.)
- **Logout Conditions**. See Logout Condition Editor on page 187. (This option appears only for login macros, not workflow macros.)
- **Browser Settings** (Basic Scan for WebInspect, Web Site Scan for WebInspect Enterprise, and stand-alone Web Macro Recorder only). See Browser Settings on page 193.
- **Rendering engine: Firefox** or **Rendering engine: IE**. See Internet Explorer Browser Technology on page 188.

At times while you record or play back a macro in Guided Scan, at the bottom of the Guided Scan screen you can click the **Recorded Locations** button to expand (or later contract) the list of locations the macro has accessed. The list has the following columns:

- **Run**. Steps that are selected (checked) are played when you click **Play All**. All steps are stored in the macro when you save it, but only the selected steps are run whenever the macro is played.
- **Excluded**. Select **Url, Directory,** or **Page** to add that type of exclusion rule. The exclusion rule will apply to any requests made by a scan that uses this scan configuration. This column also displays, read only, the causes of any existing exclusions for requests—Custom, Disallowed Host, or, if **Restrict to folder** was selected at the start of configuring the scan, Outside Root.
- **Method**. The method of the request, for example, GET or POST.
- **Status**. The status code of the response to the request, for example, 302 or 200.
- **URL**. The URL of the request.

Task 1: Record or edit the macro

If your website requires authentication, do not record login steps in a workflow macro. Instead, record a separate login macro to log in to your website.

1 Select one of the following procedures, based on the activity you want to perform, and follow the on-screen guidance:

- **Use the Web Macro Recorder in stand-alone mode to record a login macro.**

  If you want to use the Web Macro Recorder in stand-alone mode (not in conjunction with running a scan) in order to record or edit a login macro, on the WebInspect or WebInspect Enterprise toolbar click **Tools** → **Login Macro Recorder.** Then proceed to step 2 on page 181 (*before* Task 2).

- **Use the Web Macro Recorder in stand-alone mode to record a workflow macro.**

  If you want to use the Web Macro Recorder in stand-alone mode (not in conjunction with running a scan) in order to record or edit a workflow macro, on the WebInspect or WebInspect Enterprise toolbar click **Tools** → **Workflow Macro Recorder** and proceed to step 2 on page 181 (*before* Task 2).

- **Use a login macro in a Guided Scan.**

In WebInspect Enterprise, the first time a user launches Guided Scan (or creates a report) from WebInspect Enterprise or Software Security Center, the WebInspect Enterprise Thin Client application, including an installation wizard and its own Help system, is automatically downloaded and installed on the user's computer. Then the interface for the selected function opens (and its Help becomes available). You can also refer to the *WebInspect Enterprise User Guide*.

If you are completing the **Application Authentication** → **1. Select Login Macro** step of a Guided Scan, select the **Use a login macro for this site** option, and click **Create** to record a new login macro, or click the ... (browse) button to navigate to, select, and optionally edit an existing login macro to use in the scan. To clear a previously selected macro from the text box, click the **X** at the right end in the text box. (You cannot edit a macro recorded using the Event-Based IE Compatible Web Macro Recorder tool from earlier versions of WebInspect or WebInspect Enterprise. For more information, see Opening Macros Recorded with the Event-Based IE Compatible Web Macro Recorder on page 176.)

If a particular login macro uses parameters, a table of those parameters is displayed when that login macro is selected. Edit the values of the parameters as needed.

Proceed as follows:

— If you are recording or editing a login macro, proceed to step 2 on page 181 (*before* Task 2).

— If you need to select an existing login macro, not record or edit a macro, after selecting a macro, click the **Next** button in the Guided Scan pane and continue configuring the scan.

- **Use workflow macros in a Guided Scan.**

In WebInspect Enterprise, the first time a user launches Guided Scan (or creates a report) from WebInspect Enterprise or Software Security Center, the WebInspect Enterprise Thin Client application, including an installation wizard and its own Help system, is automatically downloaded and installed on the user's computer. Then the interface for the selected function opens (and its Help becomes available). You can also refer to the *WebInspect Enterprise User Guide*.

If you are completing the **Start Parameters** → **2. Choose Scan Type** step of a Guided Scan, select the **Workflows** option in the Scan Type section. Later in the Guided Scan, in the **Workflows** → **1. Manage Workflows** step, workflow macro information is displayed in the right pane.

Click **Record** to record a new workflow macro or click **Import** to add an existing workflow macro to the list. When a macro in the list is selected, click **Edit** to edit it, **Delete** to remove it from the list, or **Export** to save it with a name and location you specify. (You cannot edit a macro recorded using the Event-Based IE Compatible Web Macro Recorder tool from earlier versions of WebInspect or WebInspect Enterprise. For more information, see Opening Macros Recorded with the Event-Based IE Compatible Web Macro Recorder on page 176.)

When the first workflow macro is added to the list, its name (or default name) appears in the Workflows table and an Allowed Hosts table is added to the pane. Adding another workflow macro can add more allowed hosts. Any host that is enabled is available to all the listed workflow macros, not just the workflow macro for which it was added. The Guided Scan will play all the listed workflow macros and make requests to all the listed allowed hosts, whether or not their check boxes are selected. If a check box for an allowed host is selected, the scanner will crawl or audit the responses from that host. If a check box is not selected, the scanner will not crawl or audit the responses from that host.

In addition, if a particular workflow macro uses parameters, a table of those parameters is displayed when that workflow macro is selected in the list. Edit the values of the parameters as needed.

Proceed as follows:

— If you are recording or editing a workflow macro, proceed to step 2 on page 181 (*before* Task 2).

— If you need to select one or more existing workflow macros, not record or edit any macros, after adding the macros to the Workflows table, click the **Next** button in the Guided Scan pane and continue configuring the scan.

- **Use a login macro in a Basic Scan in WebInspect or a Web Site Scan in WebInspect Enterprise.**

  If you are completing Step 2 of a Basic Scan in WebInspect or a Web Site Scan in WebInspect Enterprise and you select the **Site Authentication** option to use a login macro, click **Record** to record a new login macro or click the  (browse) button to navigate to, select, and optionally edit an existing login macro to use in the scan.

  If a particular login macro uses parameters, a table of those parameters is displayed when that login macro is selected. Edit the values of the parameters as needed.

  Proceed as follows:

  — If you are recording or editing a login macro, proceed to step 2 on page 181 (*before* Task 2).

  — If you need to select an existing login macro, not record or edit a macro, after selecting a macro complete Step 2 of the Basic Scan in WebInspect or Web Site Scan in WebInspect Enterprise and continue configuring the scan.

- **Use workflow macros in a Basic Scan in WebInspect or a Web Site Scan in WebInspect Enterprise.**

  Note: Parameters are not supported for workflow macros created or used in a Basic Scan in WebInspect or a Web Site Scan in WebInspect Enterprise. If you need to use parameters for your workflow macro, use Guided Scan.

  If you are completing Step 1 of a Basic Scan in WebInspect or a Web Site Scan in WebInspect Enterprise and you select the **Workflow-Driven Scan** option to use a workflow macro, click **Record** to record a new workflow macro or click **Manage** to navigate to, select, and optionally edit an existing workflow macro to use in the scan.

  If you click **Manage**, or after you record and save a new workflow macro, the *Select Workflow-Driven Scan Macros* dialog box appears. It displays workflow macro information for a list of workflow macros you select. Click **Import** to add an existing workflow macro to the list. (You can also click **Record** here to record a new workflow macro.) When a macro in the list is selected, click **Edit** to edit it, **Remove** to remove it from the list, or **Export** to save it with a name and location you specify.

  When the first workflow macro is added, its name (or default name) appears in the Macros table in the dialog box and a specific entry is added to the Allowed Hosts list. Adding another workflow macro can add more allowed hosts. Any host that is enabled is available to all the listed workflow macros, not just the workflow macro for which it was added. The scanner will play all the listed workflow macros and make requests to all the listed allowed hosts, whether or not their check boxes are selected. If a check box for an allowed host is selected, the scanner will crawl or audit the responses from that host. If a check box is not selected, the scanner will not crawl or audit the responses from that host.

  Proceed as follows:

  — If you are recording or editing a workflow macro, proceed to step 2 (*before* Task 2).

  — If you need to select one or more existing workflow macros, not record or edit any macros, after adding the macros to the Macros table complete Step 1 of the Basic Scan in WebInspect or the Web Site Scan in WebInspect Enterprise and continue configuring the scan.

2 Follow the guidance in the yellow instruction bar to record or edit the macro. All of your actions will be recorded and displayed in the macro steps pane on the right. You can stop recording at any time.

  The instructions differ somewhat between login macros and workflow macros. For example, in workflow macros, you choose where to navigate through the target site, and the URLs are recorded as you navigate, so that you can later replay them by running the macro. Also, workflow macros do not include any logout conditions.

If the website requires users to answer a variable set of questions in order to complete the login process, go to Recording a Macro for a Site with Multiple, Variable Login Questions on page 185 at the appropriate time to create the macro steps required for this case, and then return to Task 2 on page 182.

When you are instructed to play the macro, go to Task 2 on page 182.

Note: In a Guided Scan, if you record or edit a workflow macro for a site that already has a login macro that you specify (in **Application Authentication → 1. Select Login Macro**), for your convenience the login macro automatically plays before you begin recording the workflow macro to ensure that the site is accessible and to obtain and record state information from the site that the workflow macro uses whenever it is played.

Task 2: Play the macro

Play the macro, correcting any errors that occur during the process:

1. When you are instructed to do so, click the **Play** button in the instruction bar.

   The macro steps are highlighted as playback progresses. If the macro detects no errors (while using Firefox technology), "Replay succeeded" is displayed at the bottom of the right pane.

   If the macro had errors, see Debugging Macros on page 198.

2. Answer the question "Did the macro play correctly?" In other words, indicate whether the login macro successfully logged in to the target site or the workflow macro accessed all the recorded URLs. Successful replay of the macro in step 1 does not guarantee that the macro did what you intended.

   If you click Yes to indicate that the macro played correctly, do the following:

   - If you are recording a *workflow* macro, go to Task 5 on page 184. Workflows macros do not include logout conditions, so you are skipping tasks associated with logout conditions.
   - If you are recording a *login* macro but *not* in a Guided Scan, the macro recorder attempts to automatically detect a logout condition. If it succeeds, the macro is complete and you can proceed to Task 4 on page 184. If the macro recorder does not detect a logout condition, proceed to Task 3 on page 183.
   - If you are recording a *login* macro in a Guided Scan, the instruction bar in the macro recorder recommends that you select an object on the displayed Web page to specifically indicate successful login. Doing so adds a Wait step to the macro to wait for appearance of the object you will select. Without such a step in the macro, the macro operates on the assumption that the login fully succeeded when "document loading" of the initial website page (seen after login from the login page) completes. However, some websites transfer many updates to and from the browser (using AJAX technology, for example) before and/or after the browser renders the initial page. This can lead to timing problems in macro playback if the macro proceeds as though the initial page has been fully rendered when it has not and then the macro prematurely runs subsequent steps. To prevent these problems, it is best practice to click **Yes** and provide a consistent, positive indication of access to the initial page.

     — If you click **No** and do not specify an object that specifically indicates login succeeded, the macro recorder attempts to automatically detect a logout condition. If it succeeds, the macro is complete and you can proceed to Task 4 on page 184. If the macro recorder does not detect a logout condition, proceed to Task 3 on page 183.

     — If you click **Yes** to specify an object that specifically indicates login succeeded, the macro recorder asks you to select that object. A Log Out button or "Welcome" text on the initial website page are common examples of objects that indicate successful access to that page. After you select the object, the macro recorder asks you to click **Play** to verify the macro again, now including the added Wait step. When the macro recorder asks you again if the macro succeeded, successful playback now means that the initial page actually displayed the object you specified.

——— If you click **Yes** to indicate that the macro with the login object played correctly, the macro recorder attempts to automatically detect a logout condition. If it succeeds, the macro is complete and you can proceed to Task 4 on page 184. If the macro recorder does not detect a logout condition, proceed to Task 3 on page 183.

——— If you click **No** to indicate that the macro with the login object did not play correctly, continue to the following paragraphs.

If you click No to indicate that the macro did not play correctly, do the following:

- If this is the *first* time you click **No**, the macro recorder automatically:
  - Adjusts the Script Level slider to level 3 to display and play back *all* of the macro steps. For more information, see Modify Script Levels on page 199.
  - Plays the macro again.
  - Asks you again whether the macro succeeded.

  Return to the beginning of this step (step 2 on page 182).
- If this is the *second* time you click **No**, the macro recorder automatically switches to IE technology (as if you started recording the macro by selecting the IE option for the **Rendering engine** button). A new pane for locations appears at the bottom of the Web Macro Recorder interface, rather than a pane for macro steps on the right. You can adjust the height of the locations pane relative to the top (target site) pane. In a Guided Scan you can also expand and collapse the locations pane by clicking **Recorded Locations** below the pane. Go to Using IE Technology to Record Web Traffic on page 189.

Task 3: Identify a "logout" condition if automatic detection by the macro recorder fails

Note: For workflow macros, skip to Task 5 on page 184. Workflow macros do not have logout conditions.

At this point the target site is at a protected page, that is, a page that can be accessed only when a user is logged in.

To automatically resume a scan that gets logged out, a login macro needs to include at least one condition that represents getting logged out. Then when WebInspect or WebInspect Enterprise recognizes any of the logout conditions, it will automatically restart the macro to log back in and resume the scan where it left off.

After the macro plays back successfully, the Web Macro Recorder uses sophisticated analysis to try to automatically detect a logout condition. At that time, it displays "Detecting Logout Condition..." in the instruction bar. If the Web Macro Recorder succeeds in detecting a logout condition, the login macro is complete, as stated in the instruction bar, and you can proceed to Task 4 on page 184.

You can view the automatically detected logout condition and add other logout conditions by clicking **Logout Conditions** in the toolbar to open the Logout Condition Editor.

If the automatically detected logout condition is identified as the "Auto Redirect" type in the Logout Condition Editor, the Web Macro Recorder generated the displayed regular expression (regex), including the 'Location' header of a redirect (302), to represent the logout condition for the redirect when login state was lost.

If navigation parameters are specified in the scan settings, they are used as applicable at scan time to revise and uniquely identify the URL in the 'Location' header in the regex for the redirect. For information about navigation parameters, see the *WebInspect User Guide* or the *WebInspect Enterprise User Guide* for information about HTTP parsing in default scan settings.

If you later determine that the Auto Redirect regex does not work as well as necessary to automatically log back in to the site being scanned, you cannot edit the regex in place, but you can copy it, manually create a new Regex condition that you revise from the copy, and optionally delete the Auto Redirect regex. For more information, see Logout Condition Editor on page 187.

If the automatically detected logout condition is identified as the “Automatic” type in the Logout Condition Editor, the Web Macro Recorder detected a non-302 response, such as a 200.

If you later determine that the Automatic logout condition does not work as well as necessary to automatically log back in to the site being scanned, you can replace it by manually specifying a regular expression (Regex), an object (interface element), or a URL as a logout condition. For more information, see Logout Condition Editor on page 187.

If the Web Macro Recorder fails to detect a logout condition when you record a login macro, it presents an error message and offers to open the Logout Condition Editor, which is the same as clicking **Logout Conditions** in the toolbar. In the Logout Condition Editor, you can manually specify, in recommended order, a regular expression (Regex), an object (interface element), or a URL as a logout condition.

To specify a logout condition if the Web Macro Recorder could not detect one:

1. In the macro, navigate in the target site to a page that users consistently see when they get logged out.
2. Use the Logout Condition Editor to specify a logout condition that is unique to this page. See Logout Condition Editor on page 187.

Task 4: (Optional) Modify the logout conditions

If you have been recording a login macro, this task is optional. It does not apply to workflow macros.

To examine or modify the logout condition, click **Logout Conditions** in the toolbar. You can specify as many different logout conditions as you need, and if any of them is met during a scan, WebInspect or WebInspect Enterprise invokes the login macro to log back in. For more information, see Logout Condition Editor on page 187.

Task 5: (Optional) Parameterize the macro

This task is optional. To parameterize the login credentials or the URL, click **Parameters Editor** in the toolbar. For more information, see Parameters Editor on page 194.

Task 6: (Optional) Save the macro

To save the macro for future use, click **Export** in the toolbar for a Guided Scan, or click **Save** (and then **Save** or **Save As**) in the toolbar for a Basic Scan in WebInspect or a Web Site Scan in WebInspect Enterprise. This completes the macro recording procedure.

If you are configuring a Guided Scan, a Basic Scan in WebInspect, or a Web Site Scan in WebInspect Enterprise, close the Web Macro Recorder to return to the configuration process.

Recording a Macro for a Site with Multiple, Variable Login Questions

Challenge-response authentication is a family of protocols in which the server presents a question (the challenge) and the client must provide a valid answer (the response). In the simplest example, the challenge asks for a password and the valid response is the correct password.

Many websites now present multiple challenges to the user. Typically, when a user first registers with a website, the site presents a list of questions to which the user provides answers that will be used for subsequent authentication. For example:

- What is your favorite color?
- What was the name of your first pet?
- In what town or city were you born?
- What was the make of your first automobile?

When the user later attempts to log in, the website presents two or more of these challenges.

Some sites also create groups of challenges, and present different questions from the groups on each new login attempt, as demonstrated in the following example.

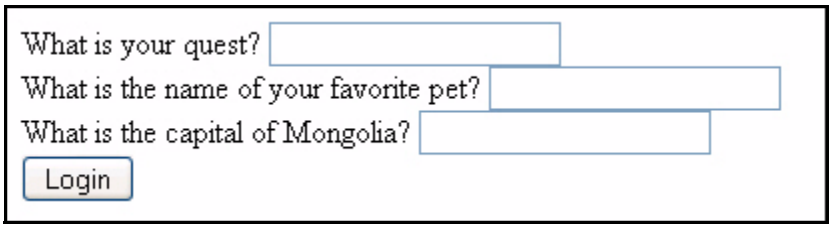
When registering for the following example website, the user is asked to provide answers to nine questions, which are arranged into three groups of three questions each, as follows.

Group 1
Q: What is your quest? A: happiness
Q: What is your name? A: Smith
Q: What is your favorite color? A: blue

Group 2
Q: What is the name of your favorite pet? A: Rusty
Q: What is your mother's maiden name? A: Jones
Q: In what state were you born? A: Delaware

Group 3
Q: What is the capital of Mongolia? A: Ulaanbaatar
Q: What is the name of a sea bird? A: Albatross
Q: What is your paternal grandmother's first name? A: Esther

The login page might look like this (using the first question from each group):

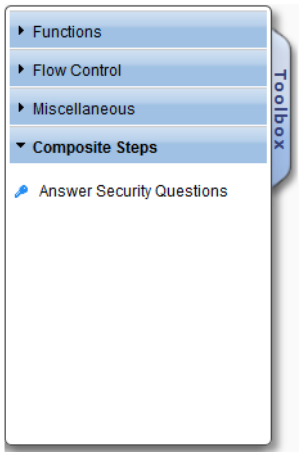


When recording a macro for a challenge/response type of login, you must know all possible question-and-answer combinations, even if only a subset of those combinations might be presented during any one login. You enter these combinations manually, as special steps while recording a macro.

At the point where the target site asks the challenge questions, usually after logging in with username and password credentials, use the following procedure to manually create the required steps for this hypothetical set of nine questions:

1 If you are recording a macro, click **Stop** on the instruction bar to stop the automatic macro recording process.

2 Click the **Toolbox** vertical tab on the left side of the macro steps pane. Click (expand) **Composite Steps**.



3 Click and drag the **Answer Security Questions** element to the right pane to create the next macro step.

4 Click the first “Click to choose an object” button in the new step and then, in the target site pane, click the object representing the first question (usually a label).

5 Click the second “Click to choose an object” button in the new step and then, in the target site pane, click the object representing the answer (usually a text box).

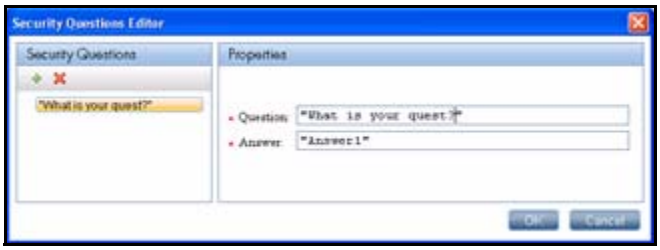
6 Place your mouse in the upper right corner of the step and click to open the Step Editor.

7 Click (expand) the **Security Questions** section. Click  to open the Security Questions Editor.

8 In the Security Questions pane of the Security Questions Editor, click  to add a new question.

A new question appears with the default name “Question1.” Its properties include the text box labelled **Question** (also shown with a default value of “Question1”) and the text box labelled **Answer**, with a default value of “Answer1.”

9 In the **Question** text box, type over the default text with the actual question exactly as it appears on the login page, including capitalization and punctuation. Be sure to enclose the text in quotation marks as shown below. The question in the left pane is simultaneously updated.



10 In the **Answer** text box, type the correct response in quotes.

11 Click **OK**.

The **Question** and **Answer** are added to a table in the **Security Questions** section in the macro step. (If you later need to edit a question or answer, reopen the Security Questions Editor.)

12 Repeat step 7 through step 11 to add the information for the second question that might appear in the same location on the web page (in this example, “What is the name of your favorite pet?”).

13 Repeat step 7 through step 11 to add the information for the third question that might appear in the same location on the web page (in this example, “What is the capital of Mongolia?”).

This completes the macro step for this particular location on the web page.

14 Refresh the web page until the second set of questions appears. Click in the target site pane and press F5 (or right-click and click **Reload**).

15 Repeat step 2 through step 13 to add another macro step for the second set of three questions and answers at the second location on the web page.

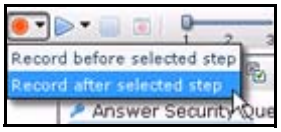
16 Refresh the web page until the third set of questions appears. Click in the target site pane and press F5 (or right-click and click **Reload**).

17 Repeat step 2 through step 13 to add another macro step for the third set of three questions and answers at the third location on the web page.

18 After creating macro steps for all possible question-and-answer combinations, if you need to record further macro steps:

a Select the last step you created.

b Click the drop-down arrow on the **Record** button in the macro steps pane and select **Record after selected step**.



c Add any further steps to the macro as needed.

d Click **Stop** on the instruction bar.

19 To play back the macro, return to Task 2 on page 182 or, if you are using IE technology, return to Task 2 on page 191.

On playback, if the macro cannot find a particular question object or answer object on the page, you can expand **Security Question Object** or **Security Answer Object** in the Step Editor for the security question and use the **Highlight** and **Replace** buttons to try to correct the failure. See Highlight an object on page 201 and Replace an object on page 205.

Logout Condition Editor

The Logout Condition Editor allows you to create or edit logout conditions for login macros. For introductory information, see Login Macros on page 174. You can specify as many different logout conditions as you need, and if any of them is met, WebInspect or WebInspect Enterprise invokes the login macro to log back in and resume a scan where it left off. The final set of all logout conditions should cover all the cases of becoming logged out during a scan of the target site.

When the Web Macro Recorder successfully detects a logout condition automatically, it categorizes the logout condition as one of the following types:

- **Auto Redirect**. This type of logout condition is created when the Web Macro Recorder detects that the target site responds with a 302 redirect. It takes the form of a regular expression (regex).
- **Automatic**. This type of logout condition is created when the Web Macro Recorder detects that the target site responds with anything other than a 302 redirect, for example, with a 200.

To add a new logout condition:

1. Click the **Logout Conditions** button in the toolbar.
2. Click  in the left pane (or click the drop-down arrow to the right of  and select **Manual**).
3. In the right pane specify the name of the new condition. (Notice that the name in the left column is simultaneously updated with your changes.)
4. Select which type of logout condition you want to use and complete the information required for that type. In order of recommended priority, the options are:
   - **Regex**. With this option, you will construct a regular expression (regex). A regular expression is a pattern that describes a set of strings. Regular expressions are constructed much like mathematical expressions by using various operators to combine smaller expressions. Only users with a working knowledge of regular expressions should use this feature.

     The regex must reflect the difference between a) the response to a logged-in user's request to access a protected page, and b) the response to the same request from the user, while *not* logged in, to access the same protected page. The general steps to construct the regex are as follows:

     — Start the Web Proxy tool to record web traffic. See Chapter 23, Web Proxy.

     — Log in to the target site legitimately and copy the URL of a protected page.

     — Log out and use the copied URL to try to access the protected page without logging in.

     — Compare the responses and identify a unique aspect of the response to the attempt to access the protected page without logging in.

     — Open the Regular Expression Editor. From the WebInspect or WebInspect Enterprise menu, select **Tools** → **Regular Expression Editor**. See Chapter 10, Regular Expression Editor.

     — Construct a regex that reflects the unique aspect of the response to the attempt to access the protected page without logging in.

     — Copy the regex into the **Regex** field of the Logout Condition Editor.
   - **Object**. After you select this option, click **Click to choose an object**, navigate to a page where the user is logged out and can log back in, and move your mouse over objects on the page until you find one that does not appear on any other page and that indicates that the user is logged out. As you mouse over objects, each one is highlighted in green until you select one or press Esc to stop the selection process. Once you select an object, if you click **Highlight** the Logout Condition Editor is hidden temporarily and the selected object is highlighted by a rapidly flashing red outline.
   - **URL**. When you select this option, the currently displayed web page is automatically used as the default value. You can specify a static URL to which the target site redirects users when it logs them out. Do not specify the target site's general login page.

In the Logout Condition Editor, if you click the drop-down arrow to the right of  , and if the **Automatic** option is available and you select it, the Logout Condition Editor closes and the Web Macro Recorder attempts to automatically detect a logout condition.

To delete a logout condition, select it in the left pane and click the red **X**.

Internet Explorer Browser Technology

By default, the Unified Web Macro Recorder tries to create a macro using Firefox technology. However, if it cannot successfully create the macro, it automatically tries again using its Internet Explorer browser technology (also referred to here as IE technology), which displays locations and web traffic data in the Web

Macro Recorder interface. For more information, see the procedure for recording a macro (Recording or Editing a Macro on page 178).

In the Unified Web Macro Recorder, you can also manually initiate the use of IE technology as you begin recording a new macro, in case the default Firefox technology of the Web Macro Recorder has not worked. HP strongly recommends that you start by trying the default Firefox technology.

The consequences of selecting IE technology (**Rendering engine: IE**) depend upon the circumstances.

- When recording a macro, selecting IE technology:
  - Switches the macro recorder from the use of custom Firefox technology to Internet Explorer browser control.
  - Records the macro in traffic mode.
- In a Guided Scan, at the **Start Parameters** → **Verify Web Site** step or at the **Optimization Tasks** → **Enhance coverage of your web site** step, selecting IE technology implies that using Firefox technology would not work for any aspect of the site, so it:
  - Switches the macro recorder from the use of custom Firefox technology to Internet Explorer browser control.
  - Changes the default script execution engine from the one introduced in WebInspect and Webinspect Enterprise 10.00 to an engine from earlier versions that is compatible with Internet Explorer.
  - Uses Internet Explorer as the default browser for subsequent steps in the scan such as recording a macro or discovering locations in **Active Learning**.

As you record a macro against a target site using IE technology, the Web Macro Recorder displays requested locations (as described in detail in subsequent sections). When you play a macro, the Web Macro Recorder also displays HTTP web traffic for both the requests and their associated responses, but the recorded macro includes only the requests.

For information about compatibility between IE technology in the Unified Web Macro Recorder tool of WebInspect and WebInspect Enterprise 10.00 and later, and macros that were recorded in earlier WebInspect or WebInspect Enterprise (or AMP) versions using the Traffic-Mode Web Macro Recorder, see Opening Macros Recorded with the Traffic-Mode Web Macro Recorder on page 176.

Using IE Technology to Record Web Traffic

HP strongly recommends initially using the default Firefox technology of the Web Macro Recorder to record a macro. However, the Web Macro Recorder can invoke IE technology automatically if Firefox technology fails or, if you have not been able to successfully record a macro with the Firefox technology, you can manually invoke IE technology. Using IE technology, the bottom pane of the Web Macro Recorder interface displays requested locations. The actions you take to record and test the macro, guided by the yellow instruction bar at the top of the screen, are essentially the same as for recording a macro using Firefox technology, but the user interface is different, as described in this section.

Proceed as follows, depending on how you are accessing IE technology:

- Proceed to Task 1 on page 190 to record the macro if one of the following applies:
  - You need to *initiate* the use of IE technology when you are just starting to use the stand-alone Web Macro Recorder or just starting to configure a Guided Scan, a Basic Scan in WebInspect, or a Web Site Scan in WebInspect Enterprise.
  - You need to edit a traffic-mode macro that was recorded in an earlier version of WebInspect, WebInspect Enterprise, or AMP.
- Go to Task 2 on page 191 if playback failed twice using Firefox technology and the macro recorder *automatically* invoked IE technology.

Task 1: Record the macro using IE technology

1 Select **Rendering engine: IE** in the Record/Edit Login Macro section of the Web Macro Recorder toolbar, the Guided Scan toolbar, the Basic Scan toolbar in WebInspect, or the Web Site Scan toolbar in WebInspect Enterprise, as applicable.

A new pane for locations appears at the bottom of the Web Macro Recorder interface, rather than a pane for macro steps on the right. You can adjust the height of the locations pane relative to the top (target site) pane. In a Guided Scan you can also expand and collapse the locations pane by clicking **Recorded Locations** below the pane.

The buttons, check box, and columns in the locations pane are described in step 2.

2 Follow the guidance in the yellow instruction bar to record the macro.

The instructions differ somewhat between login macros and workflow macros. For example, in workflow macros, you choose where to navigate through the target site, and the URLs are recorded as you navigate, so that you can later replay them by running the macro. Also, workflow macros do not include any logout conditions.

Note: IE technology does not support websites that require users to answer a variable set of questions in order to log in.

The following descriptions are provided for information and to assist with debugging. When the on-screen instructions tell you to play the macro, you can proceed to Task 2 on page 191.

From the first time you navigate to a URL, a table of request data is added to the locations pane. The locations pane has a button bar with the following buttons and check box, which become available as described:

- **Record** button.
- **Play Highlighted** button. Available after you highlight a single request (row) by clicking it. Plays the highlighted request if the associated check box in the Run column is selected. Other check boxes in the Run column do not matter.

  The first time you select a request, the locations pane splits into left and right panes. The left pane continues to display the table of request data for each location. In the right pane, the default Details tab is split, with HTTP request data above the associated response data.
- **Play All** button. Available after you click **Stop** in the instruction bar to stop recording the login steps. Plays only the requests that are selected (checked) in the Run column.

  Note: All steps are stored in the macro when you save it, but only the selected steps are run whenever the macro is played.
- **Stop** button. Available during playback after you have clicked the **Play All** button. Aborts playback upon completion of the current request.
- **Logout** button. (Does not appear for workflow macros.) Available after you click **Stop** in the instruction bar to stop recording the login steps. Logs you out of the site so that you can determine how the site responds to a subsequent request you play when logged out.
- **Delete Highlighted** button. Available immediately. Deletes the single request (row) you highlighted by clicking it.
- **Delete All** button. Available after you click **Stop** in the instruction bar to stop recording the login steps. Deletes all the requests, regardless of whether they are selected in the Run column.
- **Prompt for login (CAPTCHA)** check box. (Does not appear for workflow macros.) Available immediately. CAPTCHA is a challenge-and-response test designed to ensure that a login response is provided by a person, not generated by a computer. If your target site uses CAPTCHA, select this check box. The macro still detects a logout condition, but WebInspect or Webinspect Enterprise

users will need to log in manually at the beginning of a scan and whenever a logout occurs. Selecting this option disables selection of any of the listed requests and closes the right pane that displays HTTP traffic.

Below the button bar, the locations pane lists locations and has the following columns:

- **Run**. Steps that are selected (checked) are played when you click **Play All**. All steps are stored in the macro when you save it, but only the selected steps are run whenever the macro is played.
- **Excluded**. Select **Url, Directory,** or **Page** to add that type of exclusion rule. The exclusion rule will apply to any requests made by a scan that uses this scan configuration. This column also displays, read only, the causes of any existing exclusions for requests—Custom, Disallowed Host, or, if **Restrict to folder** was selected at the start of configuring the scan, Outside Root.
- **Method**. The method of the request, for example, GET or POST.
- **Status**. The status code of the response to the request, for example, 302 or 200.**URL**. The URL of the request.

The bottom right pane includes the following tabs:

- **Details** tab. For the selected (highlighted) request in the left pane, shows request data in the top right pane and associated response data in the bottom right pane.
- **State** tab. A collection of all the items that represent state or could represent state, that have been seen across all the locations that the macro has accessed. You can select them in any combination to characterize them as representing a state and you can manually add various types of items. Web applications can require that certain parameters be marked as "stateful."
- **Parameters** tab. (Does not appear for workflow macros.) For login macros, allows you to designate form input fields as being user name or password input so that the macro using IE technology can have user name and password parameters that can be specified at scan time, like macros that use Firefox technology.

Task 2: Play the macro using IE technology

Using IE technology, a pane for locations appears at the bottom of the Web Macro Recorder interface, rather than a pane for macro steps on the right. You can adjust the height of the locations pane relative to the top (target site) pane. In Guided Scan you can also expand and collapse the locations pane by clicking **Recorded Locations** at the bottom of the pane.

1 If the macro has not already been played automatically, click the **Play** button in the instruction bar.

The following descriptions are provided for information and to assist with debugging. You can proceed to step 2 on page 192.

The locations pane has a button bar. Below it, the left pane displays a table of data for each location. In the right pane, the default **Details** tab is split, with HTTP request data above the associated response data for the request selected in the left pane.

The button bar has the following buttons and check box:

- **Record** button.
- **Play Highlighted** button. Plays the single request (row) you highlighted by clicking it. Plays the highlighted request if the associated check box in the Run column is selected. Other check boxes in the Run column do not matter.
- **Play All** button. Plays only the requests that are selected (checked) in the Run column.

  Note: All steps are stored in the macro when you save it, but only the selected steps are run whenever the macro is played.
- **Stop** button. Available during playback after you have clicked the **Play All** button. Aborts playback upon completion of the current request.

- **Logout** button. (Does not appear for workflow macros.) Logs you out of the site so that you can determine how the site responds to a subsequent request you play when logged out.
- **Delete Highlighted** button. Deletes the single request (row) you highlighted by clicking it.
- **Delete All** button. Deletes all the requests, regardless of whether they are selected in the Run column.
- **Prompt for login (CAPTCHA)** check box. (Does not appear for workflow macros.) CAPTCHA is a challenge-and-response test designed to ensure that a login response is provided by a person, not generated by a computer. If your target site uses CAPTCHA, select this check box. The macro still detects a logout condition, but WebInspect or WebInspect Enterprise users will need to log in manually at the beginning of a scan and whenever a logout occurs. Selecting this option disables selection of any of the listed requests and closes the right pane that displays HTTP traffic.

The left pane lists locations and has the following columns:

- **Run**. Steps that are selected (checked) are played when you click **Play All**. All steps are stored in the macro when you save it but only the selected steps are run whenever the macro is played.
- **Excluded**. Select **Url, Directory,** or **Page** to add that type of exclusion rule. The exclusion rule will apply to any requests made by a scan that uses this scan configuration. This column also displays, read only, the causes of any existing exclusions for requests—Custom, Disallowed Host, or, if **Restrict to folder** was selected at the start of configuring the scan, Outside Root.
- **Status**. The status code of the response to the request, for example, 302 or 200.
- **Protected**. (Does not appear for workflow macros.) For login macros, a set of options with one selection allowed. The default is the request to the page that the Web Macro Recorder has identified as the most likely to be the protected page. This is also the page from which the Web Macro Recorder attempts to automatically determine a logout condition.
- **URL**. The URL of the request.

The bottom right pane includes the following tabs:

- **Details** tab. For the selected (highlighted) request in the left pane, shows request data in the top right pane and associated response data in the bottom right pane.
- **State** tab. A collection of all the items that represent state or could represent state, that have been seen across all the locations that the macro has accessed. You can select them in any combination to characterize them as representing a state and you can manually add various types of items. Web applications can require that certain parameters be marked as "stateful."
- **Parameters** tab. (Does not appear for workflow macros.) For login macros, allows you to designate form input fields as being user name or password input so that the macro using IE technology can have user name and password parameters that can be specified at scan time, like macros that use Firefox technology.

If the macro itself detects an inconsistency between an expected status code for a response as determined during macro recording and the actual status code during macro playback, the macro highlights the difference between expected and actual status in the bottom left pane. Investigate and address this condition.

2 Answer the question "Did the macro play correctly?" In other words, indicate whether the login macro successfully logged in to the target site or the workflow macro accessed all the recorded URLs. Successful replay of the macro in step 1 does not guarantee that the macro did what you intended.

If you click **Yes:**

- If you are recording a *login* macro, the macro recorder attempts to automatically detect a logout condition. Return to Task 3 on page 183 and from that task on, perform the same procedures as you would for the Web Macro Recorder using its default Firefox technology.

- If you are recording a *workflow* macro, return to Task 5 on page 184 and from that task on, perform the same procedures as you would for the Web Macro Recorder using its default Firefox technology. Workflow macros do not include logout conditions, so you are skipping tasks associated with logout conditions.

If you click **No**, the instructions advise you to create a new macro or use the Help. For example, see Debugging Macros on page 198 and Resolving Object Identification Issues on page 201.

Browser Settings

When using the Web Macro Recorder in stand-alone mode (click **Tools** → **Web Macro Recorder** or **Tools** → **Login Macro Recorder** in WebInspect in the WebInspect Enterprise Administrative Console), click the **Browser Settings** button in the toolbar to display the **Proxy Settings** and **Network Authentication** tabs, described in the following sections. For Guided Scans, Basic Scans in WebInspect, and Web Site Scans in WebInspect Enterprise, proxy settings and network authentication are configured as part of the scan.

Browser settings are not saved in macros.

Proxy Settings Tab

Select one of the following options:

- **Direct Connection (proxy disabled)**. Select this option if you are not using a proxy server.
- **Auto detect proxy settings**. Select this option to use the Web Proxy Autodiscovery (WPAD) protocol to locate a proxy autoconfig file and configure the browser's web proxy settings.
- **Use Internet Explorer proxy settings**. Select this option to import the proxy server information from Internet Explorer.
- **Use Firefox proxy settings**. Select this option to import the proxy server information from Firefox.
- **Configure proxy settings using a PAC file**. Select this option to load proxy settings from a Proxy Automatic Configuration (PAC) file in the location you specify in the URL box.
- **Explicitly configure proxy settings**. Select this option to configure a proxy by entering the requested information.
  - **Server**: Enter the URL or IP address of your proxy server.
  - **Port**: Enter the port number (for example, 8080).
  - **Type**: Select a protocol for handling TCP traffic through a proxy server—Standard, SOCKS4, or SOCKS5.
  - **Authentication**: Select an authentication method. For a description of authentication methods, see Authentication Types on page 44.
  - **User Name**: Specify a user name.
  - **Password**: Specify a password.
  - **Bypass proxy for**: If you do not need to use a proxy server to access certain IP addresses (such as internal testing sites), select this option and enter the addresses or URLs in the box. Use commas to separate entries.

Network Authentication Tab

If network authentication is required:

1 Click **Network Authentication**.

2 Select one of the methods. For a description of authentication methods, see Authentication Types on page 44.

3 Specify a **User Name** and **Password** for network authentication.

Select or clear the **Client Certificate** check box. If selected, complete the Certificate Store fields and select a certificate.

Parameters Editor

When recording a macro, you can use the Parameters Editor for two different purposes:

- Creating parameters for the user name and password to allow testers to use their own authentication credentials when starting a scan. For procedures, see Using Name and Password Parameters on page 194.
- Creating a parameter for the URL to allow testers to designate an alternate URL when the macro runs. For example, suppose you record a macro for www.testsite.com. At a later point in time, you rename the site to www.testsite2.com. If you parameterize the URL when you record the macro, you do not need to record a new macro. You simply enter a new host name as the Start URL when you run a scan. For procedures, see Using a URL Parameter on page 196.

When the macro is played during a Guided Scan, a Basic Scan in WebInspect, or a Web Site Scan in WebInspect Enterprise, it asks the user to specify values for the parameters.

Using Name and Password Parameters

Task 1: Create Parameters

1 After recording and testing your macro, click **Parameters Editor** in the toolbar.

The Parameters Editor opens.

2 Click  to add a parameter.

3 In the **Name** text box, enter a name for the parameter (for example: **Username**).

4 In the **Value** text box, enter the default text (for example: **Enter user name**) that you want testers to see as the value for the parameter as they configure a scan. However, while you are developing the macro, for macro playback to succeed you will probably need to enter your own valid user name and change it to the default text after you verify the macro near the end of this procedure.

5 Click  to add a second parameter.

6 In the **Name** text box, enter a name for the parameter (for example: **Password**).

7 In the **Value** text box, enter the default text (for example: **Enter password here**) that you want testers to see as the value for the parameter as they configure a scan. However, while you are developing the macro, for macro playback to succeed you will probably need to enter your own valid password and change it to the default text after you verify the macro near the end of this procedure.

8 Select **Encrypted** if the Value should be encrypted before transmission to the web server.

9 Click **Close** to close the Parameters Editor.

Task 2: Assign Parameters to Steps

1 Select the macro step that contains the user name.

2 Place your mouse in the upper right corner of the step and click to open the Step Editor.

3 Click (expand) **Arguments**.

4 Highlight the entire contents of the **Value** text box, right-click the highlighted text, and select **Replace with a Parameter**.

5 On the *Enter Parameter Name* dialog, select the parameter (**Username** in this example) from the **Select Parameter** list and click **OK**.

The **Value** takes on the format of a parameter.

6 Select the macro step that contains the password.

7 Place your mouse in the upper right corner of the step and click to open the Step Editor.

8 Click (expand) **Arguments**.

9 Highlight the entire contents of the **Value** text box, right-click the highlighted text, and select **Replace with a Parameter**.

10 On the *Enter Parameter Name* dialog, select the parameter (**Password** in this example) from the **Select Parameter** list and click **OK**.

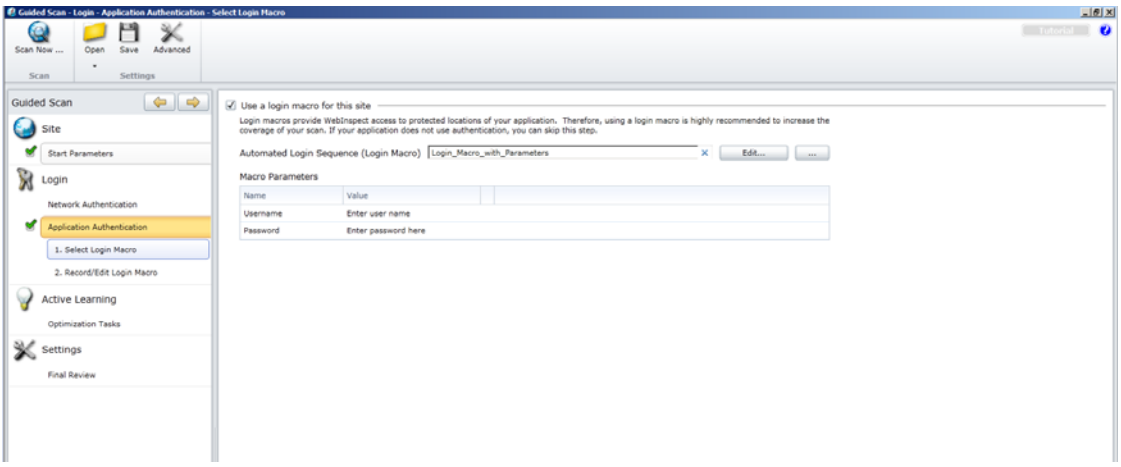
The **Value** takes on the format of a parameter.

11 Play the macro to verify that it logs in correctly.

12 If necessary, reopen the Parameters Editor and change the text in the **Value** text boxes to the default text that you want testers to see, as described in step 4 and step 7 under Task 1 on page 194.

13 Save the macro. (For Guided Scan, click **Export**. For a Basic Scan in WebInspect or a Web Site Scan in WebInspect Enterprise, click **Save**.)

When you start a Guided Scan, a Basic Scan in WebInspect, or a Web Site Scan in WebInspect Enterprise and select this macro, the parameters appear in the Macro Parameters table for a Guided Scan as shown below, or in the table below the name of the selected macro in step 2 of a Basic Scan or a Web Site Scan. The tester simply replaces the parameters with a valid user name and password.



Using a URL Parameter

Task 1: Create Parameter

1. After recording and testing your macro, click **Parameters Editor** in the toolbar.

   The Parameters Editor opens.

2. Click  to add a parameter.
3. In the **Name** text box, enter a name for the parameter (for example: **StartURL**).
4. In the **Value** text box, enter the default text, Host Name, or URL (for example: **www.Placeholder.com**) that you want testers to see as the value for the parameter as they configure a scan. However, while you are developing the macro, for macro playback to succeed or for security reasons you might need to use a different, temporary default text, Host Name, or URL and change it to the default after you verify the macro near the end of this procedure.
5. Click **Close** to close the Parameters Editor.

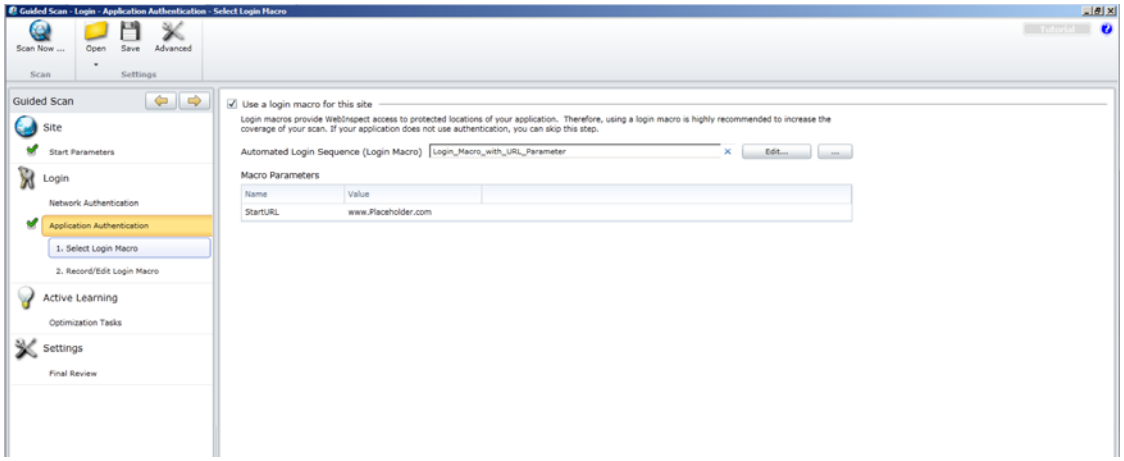
Task 2: Assign Parameters to Steps

1. Select the macro step that contains the URL ("Navigate to...").
2. Place your mouse in the upper right corner of the step and click to open the Step Editor.
3. Click **Arguments**.
4. Highlight the entire contents of the **Location** text box, right-click the highlighted text, and select **Replace with a Parameter**.
5. On the *Enter Parameter Name* dialog, select the parameter (**StartURL** in this example) from the **Select Parameter** list and click **OK**.

   The **Location** takes on the format of a parameter.

6. Play the macro to verify that it logs in correctly.
7. If necessary, reopen the Parameters Editor and change the text in the **Value** text box to the default text, Host Name, or URL that you want testers to see, as described in step 4 under Task 1 on page 196.
8. Save the macro. (For Guided Scan, click **Export**. For a Basic Scan in WebInspect or a Web Site Scan in WebInspect Enterprise, click **Save**.)

When you start a Guided Scan, a Basic Scan in WebInspect, or a Web Site Scan in WebInspect Enterprise and select this macro, the parameters appear in the Macro Parameters table for a Guided Scan as shown below, or in the table below the name of the selected macro in step 2 of a Basic Scan or a Web Site Scan. The tester either leaves the parameter unchanged (to access the original URL) or enters the URL of the target site.



Enhancing Macros

There are a number of optional enhancements that can be added to macros.

Modify Steps

Modify step arguments and objects by selecting the desired step and expanding the options. This expands the step and allows you to modify the objects and properties. For a detailed list of the step structure, see Toolbox on page 205.

Insert loops

A loop repeats a selected portion of the macro until certain criteria are met or for a specified number of times. To insert a loop, click the **Toolbox** vertical tab on the left side of the macro steps pane, click (expand) **Flow Control**, and click and drag the **For loop** element to the desired location among the macro steps. For more information, see Inserting and Modifying Loops on page 205.

Insert If blocks or If-else blocks and exit steps

To conditionalize a portion of the macro, you can insert If or If-else blocks. To insert an If block, click the **Toolbox** vertical tab on the left side of the macro steps pane, click (expand) **Flow Control**, and click and drag the **If block** element to the desired location among the macro steps. To add an else condition, click the **Add else** link next to the If step title. For more details, open the online Help and find the topic titled Step Arguments.

Exit steps cause a macro to exit the iteration or the entire macro. These can be used with If statements to exit a macro or iteration when a specified condition occurs. To insert an exit step, click **Toolbox**, click (expand) **Flow Control**, and click and drag the **Exit** element to the desired location among the macro steps.

Insert comments

To insert comments into your macro, click the **Toolbox** vertical tab on the left side of the macro steps pane, click (expand) **Miscellaneous**, and click and drag the **Comment** element to the desired location among the macro steps.

Insert Catch Error Steps

"Catch error" steps are group steps that run their contents if the previous step contains an error. (To group steps, use **Ctrl** + click to select multiple steps, right-click any of them, and click **Group Steps**.) Additionally, the error is "caught" and is not returned. You can define catch error steps to catch any error, or a specific type of error. If there are two catch error steps in a row, they both apply to the same step. To insert a catch error step, click the **Toolbox** vertical tab on the left side of the macro steps pane, click (expand) **Flow Control**, and click and drag the **Catch Error** element to the desired location among the macro steps.

Verify that an object exists

To verify that a string or object exists in the application, insert a verify step:

1. Click the **Toolbox** vertical tab on the left side of the macro steps pane, click (expand) **Functions**, and click and drag the **Verify** element to the desired location among the macro steps.
2. Click the object in the verify step.
3. Select the object you want to verify.

Insert generic steps

You can insert a blank step and manually configure it. To insert a generic step, click the **Toolbox** vertical tab on the left side of the macro steps pane, click (expand) **Functions**, and click and drag the **Generic Object Action** element or the **Generic Browser Action** element to the desired location among the macro steps. Expand the step, and enter the desired step properties. Generic Object Actions perform an unspecified action on an object. Generic Browser Actions perform an unspecified action on the browser such as go back, reload, switch tabs, etc.

Debugging Macros

This section describes the basic steps involved in interactively debugging a macro.

View Replay Errors in Browser

If any steps failed during replay, they are marked with an error icon  . Hover the mouse pointer over these icons to view descriptions of the errors.

Run the Macro Step by Step

The step-by-step replay allows you to view the sequence more slowly and in a controlled manner. To run the macro step by step, select the down arrow next to the **Replay** button in the right (macro steps) pane and select **Replay step by step**. Repeat this procedure after each step to continue the step-by-step replay.

Insert Breakpoints

Breakpoints instruct the macro to stop running during a replay when in interactive mode. They can be used to help debug your macro. To insert a breakpoint, select the desired step and click **Toggle breakpoints**  in the macro steps toolbar (or right-click on the step and click **Toggle Breakpoint** in the popup menu).

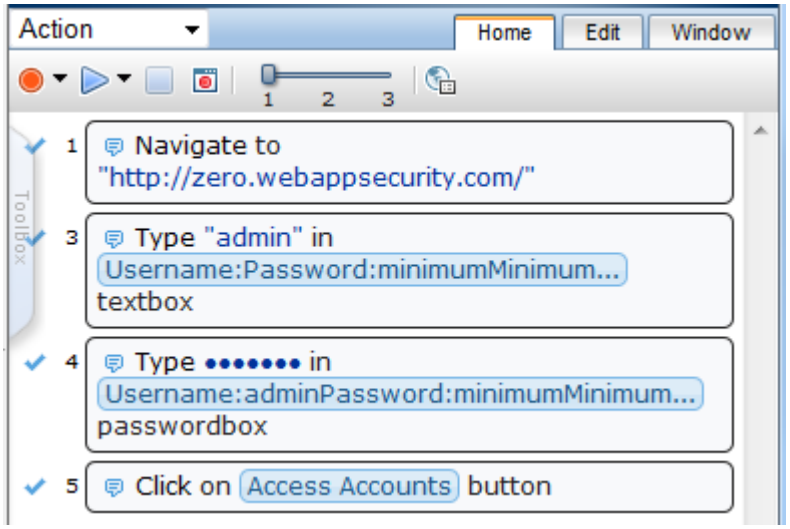
Modify Script Levels

As you record a macro, it assigns a level from 1 to 3 to each step. For example, a level 1 step is essential to the macro. A click step that occurs in an area of the application that has no effect is assigned to level 2. Mouse-over steps are generally considered unnecessary for the macro and are assigned to level 3.

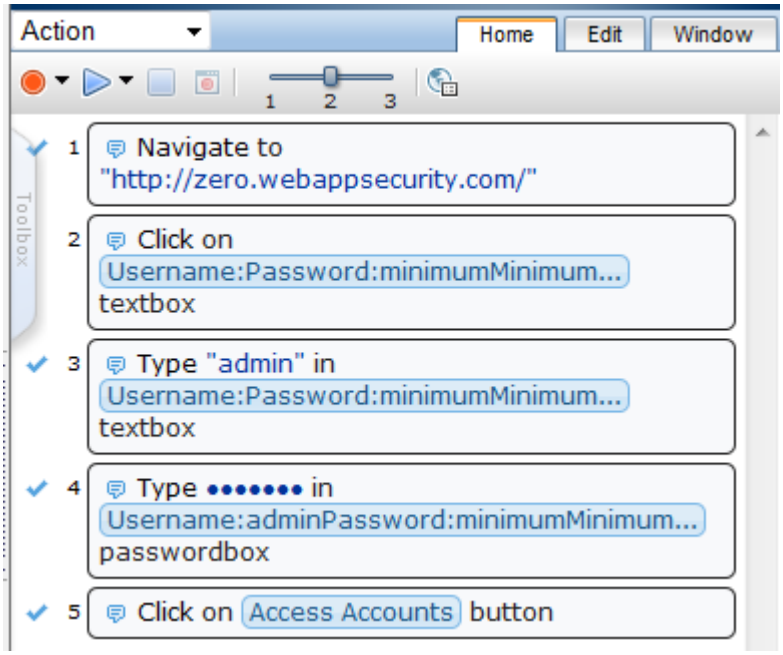
Macro steps are displayed *and played* with the granularity specified as level 1, 2, or 3 in the **Script Level** slider in the macro steps toolbar at the top of the Home tab. The highest granularity is level 3—setting the slider to level 3 displays and plays back all the steps at levels 1, 2, and 3. Using higher granularity might be required for successful playback, but it can cause the macro to take longer to run. By default, the **Script Level** is set to 1.

To modify a macro's replay level, drag the **Script Level** slider in the macro steps toolbar to the desired level.

The following illustration shows a macro for which step 2 is hidden at **Script Level** 1.



When the **Script Level** is changed to 2 as shown below (or if the **Script Level** were changed to 3), then macro step 2, which represents clicking in a text box and is assigned to **Script Level** 2, is also displayed and will run if the macro is replayed.



In certain cases, you may want to manually change the level of a particular step, not the entire macro. For example, you may want to display and play a particular mouse-over step. To change the level of a step:

1 Place your mouse in the upper right corner of the step and click to open the Step Editor for the step.

2 Move the slider at the top of that step (to the right of the step number) to the desired level.

If the step is part of a group step, both the group step and the individual step must be modified. (To group steps, use **Ctrl** + click to select multiple steps, right-click any of them, and click **Group Steps**.)

Insert Wait Steps

Wait steps cause the macro to pause for a specified amount of time before continuing with the next step. Wait for Object steps cause the macro to wait for a specified object to load before continuing with the next step. Wait steps begin after the End Event of the previous step is reached. This means that the previous step may continue to run after the wait step has been reached.

To insert a wait step, click the **Toolbox** vertical tab on the left side of the macro steps pane, click (expand) **Functions**, and click and drag the **Wait** element or the **Wait for Object** element to the desired location among the macro steps. Wait steps wait for a specified amount of time. Wait for Object steps wait until the specified object appears in the application. In Wait for Object steps, select the **Click to choose an object** button to select the target object in the application.

Disable/Enable Steps During Replay

To disable or re-enable a macro step during replay, place your mouse in the upper right corner of the step and click to open the Step Editor, and click the  icon in the toolbar for the step. Alternatively, to disable or re-enable one or more steps, use **Ctrl** + click to select them, right-click one of the steps, and click **Disable Steps** or **Enable Steps** on the popup menu.

Disabled steps remain in the macro and can be re-enabled in the future, but are not played.

Make a Step Optional

Some steps can be made optional. An optional step is skipped during replay if its object is not found. To make a step optional, place your mouse in the upper right corner of the step and click to open the Step Editor, and click the  icon in the toolbar for the step. To make a step non-optional again, click the icon again.

Play a Step

To play one step, place your mouse in the upper right corner of the step and click to open the Step Editor, and click the  icon in the toolbar for the step.

Play From a Step to End of Macro

To start playback at one particular step and continue until the end of the macro, select the starting step, right-click on the step, and click **Play From This Step** on the popup menu.

Resolving Object Identification Issues

In dynamic websites, objects that have been recorded can often move or change content. Object identification presents one of the biggest challenges with recording and replaying Web 2.0 applications. This can cause a macro to lose the ability to locate the object.

The Web Macro Recorder includes sophisticated mechanisms to overcome this challenge, including the Highlight, Improve Object Identification, Replace, and Related Object options within steps that have objects. Using these options requires that you select an object in the application. For cases where various actions are required in the application to make the object visible, such as mouse over and mouse click, use the **Ctrl+Alt+F4** option to suspend the object-selection mode until you bring the object into view and press **Ctrl+Alt+F4** again to select the object.

When identifying objects for applications that were recorded in windows, use the Windows tab to make sure that the correct window is selected.

After you perform any of the changes, first replay the single failed step in question and then replay the entire macro again. This will help verify whether the change has solved the issue you encountered.

The following sections describe ways to resolve object identification issues.

Highlight an object

Regardless of which method of object identification is used, place your mouse in the upper right corner of the step and click to open the Step Editor. Click (expand) **Object** and click the **Highlight** button  to check at any time whether an object is visible in the application. If the object is found, it is temporarily surrounded by a flashing red outline. If the object is not found, an error message is displayed. The error could be an issue of pacing and timing, or that the correct page to find the object is not displayed.

Improve Object Identification

If the Highlight option fails, click the  . icon (with tooltip “Improve object identification”) next to the selected ID Method for the object. This will let the Web Macro Recorder relearn the properties of the object and compare them to the properties learned during recording. Based on the differences, the necessary adjustments can be made. Depending on how dynamic the application is, you may need to use the Improve Object Identification function more than once.

Once you have done this, try replaying the step again to check whether the problem has been solved.

Consider Alternative Steps

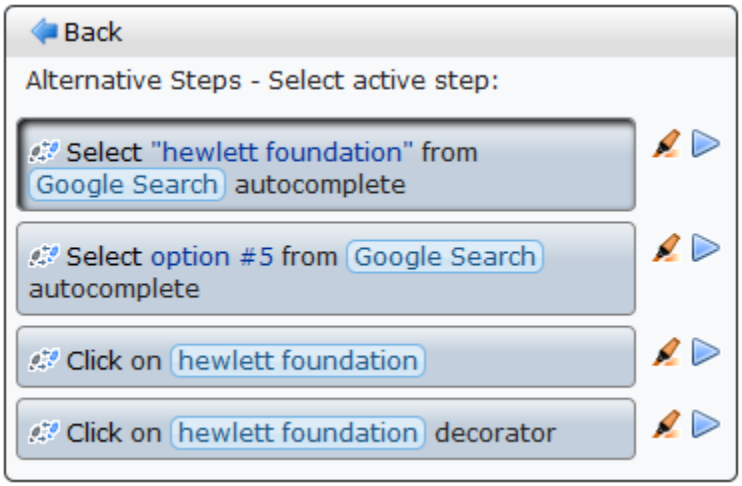
Alternative steps allow you to view multiple ways to perform the same action in a step, where it is possible. You can modify the step for the best or most consistent macro performance, or for debugging purposes.

For example, you may be clicking on an option in a drop down list in which the text changes based on some value. If you try to click based on the text, the step may fail. If you use an alternative step that selects the item in the list based on the ordinal value of the option within the list, the click will succeed regardless of the text.

Steps that have alternative options are labeled with an alternative step icon  on the left. Click the icon to view the alternative options for that step. (If the Step Editor is open, a button labelled **Alternative steps**, with the same icon, appears in the step’s toolbar and performs the same function.)

The following screens show an example of alternative steps. After performing a Google search on “hewlett” and selecting “hewlett foundation” as the fifth automatically generated item in the Google search box autocomplete, the step that has alternatives (expanded in the Step Editor) is shown below.

Clicking the  icon displays the alternative steps shown below.



The  icon to the right of each alternative has the tooltip "Highlight the object in the AUT," where AUT means application under test. This performs the same highlighting function as described in Highlight an object on page 201, with the convenience of being able to highlight each alternative one at a time within the macro step.

Each alternative also has a Play icon so that you can confirm that using an alternative takes the appropriate action for the macro step.

Macro replay succeeded for each alternative.

Click the desired alternative to make it active, and click **Back** to return to normal display of the macro step using the alternative you selected. Replay the macro to test it.

Modify the Object Identification Method

You can modify the way the Web Macro Recorder identifies the object by modifying the object identification method (ID method) in the Object section of the Step Editor. The following options are available:

- **Automatic**. The default and recommended object identification method. The Automatic method allows the Web Macro Recorder to use its internal advanced algorithms to locate the object. If this method does not successfully find the object during replay, click the  icon (with tooltip "Improve object identification") and replay the macro again.
- **XPath**. If Automatic identification fails, even after using Improve Object Identification or Related Objects (see Relate objects to other objects on page 204), try using the XPath identification method. This method identifies the object based on an XPath expression that defines the object in the DOM tree. For example, if you need to select the first search result, regardless of the term being searched for, using XPath identification may help.

  Click the drop-down arrow next to the **XPath** edit box to select a suggested XPath for the object. You can click the popup **Edit** button at the right end of the **XPath** edit box to open the XPath Editor and edit the suggested XPath.

  For the XPath ID method, the tooltip for the  icon changes to "Regenerate expression." When you click the icon, you can select an object in the interface and thereby create its associated XPath.

- **JavaScript**. JavaScript code that returns an object. For example: document.getElementById(“SearchButton”) returns an element that has a DOM ID attribute of “SearchButton.”

  Using the JavaScript identification method, you can write JavaScript code that references the returned document and you can use CSS selectors and other standard functions.

  For example, the page returned by the server contains multiple links with the same “title” attribute (search results) and we want the script to randomly click on one of the available links.

  Object identification for this case, using the JavaScript identification method, may look similar to the following:

  ```
  var my_results = document.querySelectorAll('a[title="SearchResult"]');

  random(my_results);
  ```

Modify the macro timing

Sometimes objects may not be found because of timing and synchronization issues. For example, the macro may be looking for an object that was in the application, but the macro replayed too quickly and already progressed to another page. If you suspect that the object is not being found because of a timing or synchronization issue, you can insert Wait steps. For more information, see Insert Wait Steps on page 200.

Relate objects to other objects

If the preceding options do not solve the issue, try using the Related Objects option.

If an object becomes difficult to identify on its own, you can label the object based on a different, more stable object. For example, you can select an object that is not dynamic and “relate it” to the target object. Relations are defined visually, relating objects according to their distance in pixels from other objects. Relations are defined per ID method, per object. If more than one relation is defined for an ID method of a given object, both relations must locate the same object for the step to pass. To use this function, place your mouse in the upper right corner of the step and click to open the Step Editor, click (expand) **Object,** click (expand) **Related Objects**, and click  . Follow the directions to create a relation. Verify that it has worked by highlighting both the object and its related object.

Tips:

- Use this feature only if other identification methods have failed, as it may be more resource intensive.
- Use the minimum search area to improve performance.
- Related Objects are sensitive to window sizing. Resizing may alter object positions and relationships. This should be taken into account.
- Each identification method (Automatic, XPath, and JavaScript) has its own set of related objects. These related objects are not shared among identification methods.
- If several relations exist, they all need to be found in order for the identification to succeed.

Replace an object

If you selected the wrong object during recording or an object has permanently changed, you can replace it with a different object without replacing the step. This effectively resets the step, deleting changes made to the original step such as relations. Place your mouse in the upper right corner of the step and click to open the Step Editor, click (expand) **Object**, and click **Replace**  . Select the new object and replay the macro.

Using this option tells the macro recorder that the object currently referenced in the step is incorrect. The macro recorder will remove any current knowledge of the object and learn the object you select. Therefore, you should only use the **Replace** option if the object you used during recording was the wrong one.

Inserting and Modifying Loops

Loops repeat selected portions of the macro until certain criteria are met or for a specified number of iterations. You can insert loops and loop modifiers from the **Flow Control** section of the **Toolbox**.

"For" Loops

"For" loops perform the steps surrounded by the loop until the end condition is met or the code reaches a break statement. Loop arguments use JavaScript syntax. To insert a For loop, click the **Toolbox** vertical tab on the left side of the macro steps pane, click (expand) **Flow Control**, and click and drag the **For loop** element to the desired location among the macro steps.

"Break" statements

Break statements indicate that the current loop should end immediately. For example, if a Break statement is encountered in the second of five iterations in a For loop, the loop will end immediately without completing the remaining iterations. To insert a Break statement, click the **Toolbox** vertical tab on the left side of the macro steps pane, click (expand) **Flow Control**, and click and drag the **Break** element to the desired location among the macro steps.

"Continue" statements

Continue statements indicate that the current loop iteration should end immediately. The loop condition is then checked to determine if the entire loop should end as well. For example, if a Continue statement is encountered in the second of five iterations in a For loop, the second iteration will end immediately and the third iteration will begin. To insert a Continue statement, click the **Toolbox** vertical tab on the left side of the macro steps pane, click (expand) **Flow Control**, and click and drag the **Continue** element to the desired location among the macro steps.

Toolbox

The toolbox, a vertical tab on the left side of the macro steps pane, enables you to add steps to macros. When you click **Toolbox** and click (expand) one of the headings such as **Functions,** you can click and drag a particular element such as **Verify** to add it to the macro steps.

You can click and drag **Toolbox** to move the toolbox up or down. To close the toolbox, click **Toolbox** again.

User interface elements are described in the following table.

Toolbox User Interface Elements

UI Element	Description
Functions	**Verify**. Verify that an object exists in the application. **Wait**. Wait for a specified number of seconds before continuing with the next step. **Wait for Object**. Wait for an object to load before continuing with the next step. **Generic Object Action** or **Generic Browser Action**. Blank steps that can be inserted and manually configured. See Insert generic steps on page 198.
Flow Control	**For Loop**. A logical structure that repeats the steps contained in the loop a specified number of times. **If Block**. A logical structure that runs the steps contained in the block if the condition is met. • **Add else**. Click the **Add else** link to add an **else** section to your **If** block. If the condition is not met, the steps included in the **else** section run. • **Remove else**. Removes the **else** section from the **If** block. Note: If the **else** section contains steps and you click **Remove else**, the steps are deleted. Copy and paste them into the main body of your macro to save them. **Break**. Causes the loop to end immediately without completing the current or remaining iterations. **Continue**. Causes the current loop iteration to end immediately. The macro continues with the next iteration. **Catch Error**. Catches an error in the step immediately preceding and runs the contents of the catch error step. For more information, see Insert Catch Error Steps on page 198. **Exit**. Exits the iteration or the entire macro depending on the specified setting.
Miscellaneous	**Evaluate JavaScript**. Runs the JavaScript code contained in the step. **Evaluate JS on Object**. Runs the JavaScript code contained in the step after the specified object is loaded in the application. **Comment**. A blank step that allows you to write comments in your macro.
Composite Steps	**Answer Security Questions.** Allows you to select the interface object (usually a label) that asks a security question and the interface object (usually a text box) where the user provides the answer. Then you specify the text of the question and the answer.

General Settings

Click **General Settings**  in the toolbar of the macro steps pane to open the *General Settings* dialog.

Snapshot Generation

A snapshot is an image of the browser taken at the times specified by the following options:

- **Recording snapshots generation**. Select **Never** (the default) or **Always**. During macro recording, any snapshots that are taken are saved in the same folder as the macro.
- **Replay snapshots generation**. Select **Never**, **On Error** (the default), or **Always**. During a scan, any snapshots that are taken are saved in the log directory.

Replay Options

Specify the following options:

- **Maximum time for object-not-found (seconds)**. Specify the maximum time (in seconds) that the macro recorder will wait for the target object of a replay step to appear.
- **Inter-step interval (milliseconds)**. Specify the minimum interval (in milliseconds) between steps.
- **End-of-network identification timeout (milliseconds)**. Specify the timeout (in milliseconds). The end-of-network timeout for a step is recognized when the specified time has elapsed with no network activity.
- **Clean image cache per user**. If you select this option, the image cache will be cleared during replay.

Log Level

Select one of the following options:

- **Standard logging**. Log only warnings and high-level informational messages.
- **Extended logging**. Log low-level messages, warnings, and high-level informational messages.

Logout Detection

In the **XPath depth** option, specify the depth used for XPath in logout detection by element. The depth determines the number of xpath locators (parents) from the element up to its ancestors.

An element can be located (found) in a page using a path to its location. For example, in the following HTML, to locate , the search can use the following: find body, then find div with id painterId, or the search can use find body-> then find second div.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"

"http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
  <head>
    <title>colors</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />

  </head>
    <body>
      <div class="container">
        <div class="box">
          <div class="caret" id="red">
            <span></span>
          </div>
        </div>
        <div class="number" id="redNumber">0</div>
        <div class="box">
          <div class="caret" id="green">
            <span></span>
          </div>
        </div>
        <div class="number" id="greenNumber">0</div>
        <div class="box">
          <div class="caret" id="blue">
            <span></span>
          </div>
        </div>
        <div class="number" id="blueNumber">0</div>
      </div>
      <div class="painter" id="painterId">Color</div>
      <script type="text/javascript" language="javascript">initialize();</script>
    </body>

  </html>
```

So, when searching through larger html files with more complex structures, the process can use either a rigid full xpath, or a loose short xpath. The default value is 3.

Encryption

If you select the **Encrypt Macro** option, the entire macro file is encrypted when saved. Otherwise, the file is saved in plain text, which exposes user names and passwords. This option is selected (ON) by default.

22 Event-Based IE Compatible Web Macro Recorder

About the Event-Based IE Compatible Web Macro Recorder (Hidden) Tool

In this chapter, the term “scanner” is often used instead of “WebInspect and WebInspect Enterprise” where the information applies to both products.

A macro is a recording of the events that occur when you access and log in to a website. You can subsequently instruct the HP scanner to begin a scan using this recording.

WebInspect and WebInspect Enterprise include one “Unified” Web Macro Recorder tool. By default, it uses event-based functionality and Firefox browser technology to record new macros. The separate Event-Based IE Compatible Web Macro Recorder that was provided in some earlier versions is no longer *directly* accessible in WebInspect or WebInspect Enterprise *menus*. In effect, it is hidden. However, as described in this chapter, the Unified Web Macro Recorder allows you to indirectly open, play back, and edit existing event-based macros that were created in earlier versions of WebInspect or WebInspect Enterprise (or Assessment Management Platform—AMP), and create new macros, using the earlier Event-Based IE Compatible Web Macro Recorder. For information about the Unified Web Macro Recorder, see Chapter 21, Unified Web Macro Recorder. For information about how you can access and use macros recorded with the Event-Based IE Compatible Web Macro Recorder of earlier versions of WebInspect and WebInspect Enterprise (or AMP), see Opening Macros Recorded with the Event-Based IE Compatible Web Macro Recorder on page 176.

HP strongly recommends that you use the Unified Web Macro Recorder to record all new login macros and workflow macros.

Login Macros

A login macro is a recording of the activity that is required to access and log in to a website or web application, typically by entering a user name and password and clicking a button such as Log In or Log On. When you configure a scan, you usually specify a previously recorded login macro or record a new one at the time for the scan to use.

To prevent the scanner from terminating prematurely if it gets logged out of your application, a login macro should also specify at least one logout condition that definitively indicates that a logout has occurred. During a scan, the scanner can get logged out for a variety of reasons, including:

- Normal logout driven by the target site
- An error condition in the target site such as a timeout
- An error in the macro itself, such as an invalid parameter

Specifying a logout condition as part of the login macro makes it unnecessary for users to manually log back in, perhaps repeatedly, when unexpected logouts occur during a scan. When scanning a site, the scanner analyzes every target site response to determine the state. If the scanner determines at any time that it is logged out, it runs the login macro to log back in, and then it resumes crawling or auditing the site at the point where the logout occurred.

For the procedure to record a login macro, see Recording a Login Macro on page 210.

1 Workflow Macros

Workflow macros cannot be recorded using this Event-Based IE Compatible Web Macro Recorder tool.

If you open an existing *login* macro in this tool (see Opening Macros Recorded with the Event-Based IE Compatible Web Macro Recorder on page 176, except for Guided Scan) and select **File** → **New** → **Workflow Macro**, the Unified Web Macro Recorder opens, replacing the Event-Based IE Compatible Web Macro Recorder. You can then record a workflow macro using the Unified Web Macro Recorder (see Chapter 21, Unified Web Macro Recorder). Of course, it is simpler to just use the Unified Web Macro Recorder in the first place.

Recording a Login Macro

To record a new login macro in the Event-Based IE Compatible Web Macro Recorder (from WebInspect 9.30 or earlier, WebInspect Enterprise 9.30 or earlier, or AMP versions):

1. Open and edit an existing login macro that was created in the Event-Based IE Compatible Web Macro Recorder. See Opening Macros Recorded with the Event-Based IE Compatible Web Macro Recorder on page 176.
2. Select **File** → **New** → **Login Macro**.
3. Click **Record**.
4. In the **Address** box, enter the URL of the target website and click  (or press **Enter**).

   The Web Macro Recorder renders the resource like a browser and records each event on the Events tab in the dockable pane positioned (by default) at the bottom of the window.
5. If necessary, navigate to the login screen.
6. Enter a valid user name and password, and submit the credentials (usually by clicking a button such as Log On, Go, Submit, etc.).
7. Click **Stop** (to the right of the Address bar) or **Stop Recording** (on the Status bar).
8. When prompted to play your macro, click **OK**.

   The macro plays by sequentially executing each enabled event listed on the Events tab. A message prompts you to either confirm the success of the macro and specify a logout condition or (assuming that the macro was not successful) troubleshoot the macro.
9. Do one of the following:
   - To specify a logout condition, select **Yes** and click **Finished**. Go to Specifying a Logout Condition.
   - To troubleshoot, select **No** and click **Next**. Go to Troubleshooting a Macro on page 211.

Specifying a Logout Condition

1. Navigate to a page where you are logged out (usually by clicking a button such as **Log Out**, **Log Off**, or **Exit**).
2. Do one of the following:
   - If the browser always displays this page when you log out, click **This page displays when I have logged out** (on the Selection Mode bar that appears directly under the Web Macro Recorder toolbar).

- If the browser displays a page that contains an element or control that appears only when you are logged out, click **Select Logout Indication** (on the Selection Mode bar) and then click the element or control. For example, if a Login button appears when you have logged out, click **Select Logout Indication** and then click the Login button. Your selection appears on the **Logout Conditions** tab.
- If you want the scanner to search each page for a condition that matches a regular expression that you create, click **Add Logout Regex**. See Regular Expression Editor on page 75 for details.

3 Select **File** → **Save** (or **Save As**).

Note: You can specify a logout condition at any time by clicking **Actions** → **Add Logout Condition**.

Specifying a Confirmation Element

After recording the macro, you may optionally identify a "confirmation element" that indicates that you have logged in successfully. This is particularly useful for those sites that, following a successful login, display a specific element or control on every page. Some sites, for example, always present a "Log Out" button after the user has logged in. Identifying this confirmation element increases the probability that the scanner will be able to recognize the "logged in" condition.

Once you identify a confirmation element, if the scanner does not detect that element on the page, it assumes the macro has failed and will attempt to replay the macro up to three times. If the confirmation hint is not detected during one of these playbacks, the scanner produces an error and stops trying to use the macro.

1 Navigate to a page that appears after you log in.

2 Click **Actions** → **Add Confirmation Element**.

3 Do one of the following:

- If this page always appears after you log in, select **This page displays when I have logged in**.
- Click **Select Confirmation Element** and then click an element on the page that appears only when you are logged in.

Troubleshooting a Macro

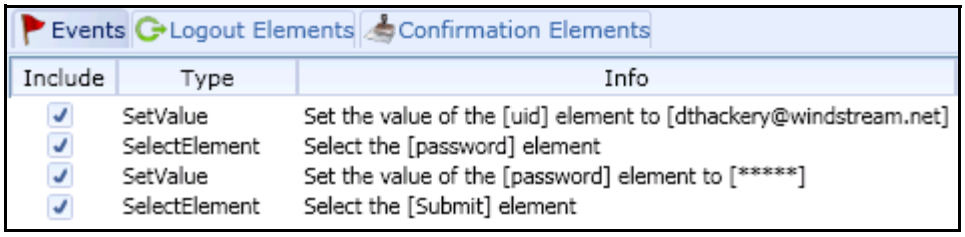
When troubleshooting your recorded macro, you have the following choices:

- **Replay Macro**. Try this solution first. The Web Macro Recorder normally plays the macro at the fastest possible speed, which may compromise performance. Use the slider to select either **Fast** (which is half the speed at which the macro was recorded) or **Original** (which mimics the speed at which the macro was originally recorded).
- **Switch to Unified Macro Recorder**. This closes the Event-Based IE Compatible Web Macro Recorder and opens the Unified Web Macro Recorder. See Chapter 21, Unified Web Macro Recorder.
- **Adjust macro hints**. Allows you to add or change confirmation elements and/or logout conditions.
- **Re-record Macro**. This choice deletes all data and returns you to the beginning point, where you can try again to create a successful macro.

Editing a macro

After recording a macro, you can modify its contents by excluding certain events.

For example, if you entered the wrong validation credentials while attempting to log in, and then entered the correct credentials, you can remove the erroneous login events simply by clearing the check box (in the Include column of the **Events** tab) next to the event you want to exclude.



Usually, the best practice is to re-record the macro instead of editing it. However, for an extremely lengthy or complex macro, you can first attempt to modify it. Excluded events are not actually removed until you save the macro, so be sure to test the modified macro (by playing it) before you save it.

You might also need to add events for those situations where events are not recorded (such as login elements located in an iframe).

The Web Macro Recorder events are defined in the following table.

Event	Definition
WaitForPageLoad	Wait for the browser to complete the processing of pages.
NavigateTo	Navigate to the specified URL.
WaitForElement	Wait for element to be rendered on current page. This is used most often to render cascading menus.
WaitNumberOfSeconds	Pause for a specific number of seconds.
Click	Simulate a mouse click on an element.
MouseUp	Simulate any mouse button being released over an element.
MouseDown	Simulate any mouse button being pressed while the pointer is over an element.
SetValue	Simulate entering a value associated with an element.
JavaScript	Execute JavaScript.
Blur	Lose focus on the element.
SelectElement	Select the specified element.

Example: Adding Elements for Iframe Login

The most frequently encountered failure to record a login macro occurs when the login elements are contained within an iframe. During recording, you might enter a user name and password, and then click the Sign In button, but nothing occurs when you play the macro.

You can edit the recorded events or you can begin by recording a new macro. If you edit the recording:

1. Click **Stop** (on the Status bar).
2. Deselect (remove the checks marks next to) those events that occur after the page is loaded.

Create an event for the user name element

1. Right-click the WaitForPageLoad event and select **Insert new event here** from the shortcut menu.
2. On the *Event Properties* dialog, click the drop-down arrow on the **Type** list and select **Click**.
3. Click **Locate an Element** (at the bottom of the dialog).
4. Move the mouse pointer to and click on the user name element (which may be labeled “name,” “user,” “email address” or other such identifier).
5. When the *Event Properties* dialog displays information about the element, click **OK**.

   Note that the event is added after (following) the event on which you clicked.

Add a value to the user name element

1. Right-click the event that you just added and select **Insert new event here** from the shortcut menu.
2. On the *Event Properties* dialog, select **SetValue** from the **Type** list.
3. Click **Locate an Element** (at the bottom of the dialog).
4. Click the user name element.
5. On the *Event Properties* dialog, enter a user name in the **Value** box and click **OK**.

Create an event for the password element

1. Right-click the event that you just added and select **Insert new event here** from the shortcut menu.
2. On the *Event Properties* dialog, select **Click** from the **Type** list.
3. Click **Locate an Element** (at the bottom of the dialog).
4. Click the password element.
5. When the *Event Properties* dialog displays information about the element, click **OK**.

Add a value to the password element

1. Right-click the event that you just added and select **Insert new event here** from the shortcut menu.
2. On the *Event Properties* dialog, select **SetValue** from the **Type** list.
3. Click **Locate an Element** (at the bottom of the dialog).
4. Click the password element.
5. On the *Event Properties* dialog, enter a password in the **Value** box and click **OK**.

Submit the user name and password

1. Right-click the event that you just added and select **Insert new event here** from the shortcut menu.
2. On the *Event Properties* dialog, select **Click** from the **Type** list.
3. Click **Locate an Element** (at the bottom of the dialog).
4. Click the submit element (which may be labeled “Submit,” “Sign In,” or other such identifier).
5. On the *Event Properties* dialog, click **OK**.

Dynamic Challenge-Response Authentication

Challenge-response authentication is a family of protocols in which the server presents a question (the challenge) and the client must provide a valid answer (the response). In the simplest example, the challenge asks for a password and the valid response is the correct password.

Many websites now present multiple challenges to the user. Typically, when a user first registers with a website, the site presents a list of questions to which the user provides answers that will be used for subsequent authentication. For example:

- What is your favorite color?
- What was the name of your first pet?
- In what town or city were your born?
- What was the make of your first automobile?

When the user later attempts to log in, the website presents two or more of these challenges.

Some sites also create groups of challenges, and present questions from different groups on each subsequent login attempt, as demonstrated in the following example.

When registering for the following example website, the user is asked to provide answers to nine questions, which are arranged into three groups of three questions each, as follows.

Group 1

“What is your name?”, “Smith”
“What is your favorite color?”, “blue”
“What is the name of your first grade teacher?”, “Williams”

Group 2

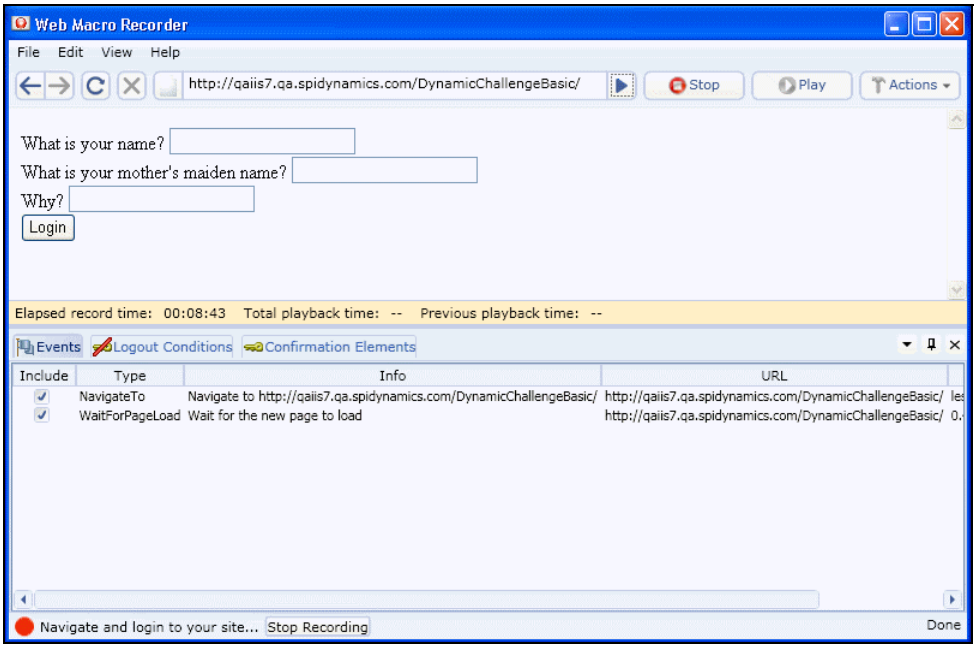
“What is your mother's maiden name?”, “Larrimore”
“In what state were you born?”, “Delaware”
“What is the name of your favorite pet?”, “Rusty”

Group 3

“Why?”, “Albatross”
“What is your paternal grandmother's first name?”, “Esther”
“What is the capital of the state you live in?”, “Atlanta”

In this example, the application randomly selects a number between 1 and 3 (inclusive) and then displays the corresponding ordinal question (first, second, or third) from each group.

1 Start the Web Macro Recorder, click **Record**, and enter the URL of the login page.



The source code for the pertinent area of the form is:

<label for="Q1"> What is your name?</Label><input id="Q1" name="Q1" />

<label for="Q2"> What is your mother's maiden name?</Label><input id="Q2" name="Q2" />

<label for="Q3"> Why?</Label><input id="Q3" name ="Q3" />

<input type="submit" value="Login" />

This illustrates that the label for each question is Q1, Q2, and Q3; similarly, the ID and name for each text box into which the user enters the response is Q1, Q2, and Q3.

2 On the login page, enter a value for each input element and click **Login**.

3 Assuming that you logged in correctly, click **Stop**.

4 When prompted to play your macro, click **Cancel**.

To modify the macro so that it accommodates a random presentation of authentication questions:

1 Navigate to the login page.

2 Click the **Events** tab.

3 Right-click the first SetValue element and choose **Select security question for this element**.

a Click **Select Security Question** (just below the toolbar).

b Click on the label for the first security question (in this example, “What is your name?”).

The *Question-Answer Groups* dialog appears.

c In this example, we know that the first question is a member of the Q1 group. So click the **Add** button, enter “Q1” in the Group Name box, and click **OK**.

Note: If your program does not divide questions and answers into groups, but presents the same set of questions at each login attempt, ignore the Group Name controls.

d Click **Click here to add new question/answer pair**.

e Enter the first question and answer pair. In this example:

Question: What is your name?

Answer: Smith

f Repeat step d and step e, entering the second and third question/answer pairs in Group 1.

g Click **OK**.

Note that a **Sec. Questions** column is added to the **Events** tab.

h Click the drop-down arrow for this element in the **Sec. Questions** column on the **Events** tab and select Q1.

4 Right-click the second SetValue element and choose **Select security question for this element**.

a Click **Select Security Question** (just below the toolbar).

b Click on the label for the second security question (in this example, "What is your mother's maiden name?").

c Click the drop-down arrow for this element in the **Sec. Questions** column on the **Events** tab and select **Manage**.

d On the *Question-Answer Groups* dialog, click **Add**, enter a group name of "Q2" and click **OK**.

e Add the three security question/answer pairs for the Q2 group, following the procedure in step 3.

5 Right-click the third SetValue element and choose **Select security question for this element**.

a Click **Select Security Question** (just below the toolbar).

b Click on the label for the third security question (in this example, "Why?").

c Click the drop-down arrow for this element in the **Sec. Questions** column on the **Events** tab and select **Manage**.

d On the *Question-Answer Groups* dialog, click **Add**, enter a group name of "Q3" and click **OK**.

e Add the three security question/answer pairs for the Q3 group, following the procedure in step 3.

6 Click **Play** to test the macro.

When troubleshooting the macro, it is usually helpful to right-click an entry on the **Events** tab and select **Playback macro to this event**.

Logout Elements

When the *Playback Successful?* dialog appears, the first of three messages at the bottom of the dialog pertains to logout conditions. These are elements, pages, or regular expressions that indicate to the Web Macro Recorder (and the scanner) that the user is no longer logged in to the site or application.

If the message is "Logout hints have been specified for this macro," the Web Macro Recorder has recognized the logout condition you specified.

However, if the message is "Unable to auto-detect logout hints (please add manually)," then one of the following occurred:

- You did not instruct the Web Macro Recorder to automatically detect logout elements (see Event-Based IE Compatible Web Macro Recorder Settings on page 219).
- The Web Macro Recorder was unable to auto-detect elements.

- You did not manually specify a logout condition.

To correct this error, click **Edit** → **Settings** → **Auto-Detection** and do one of the following:

- Select **Auto-detect logout conditions** and choose one or more of the standard logout elements (or create a custom logout element).
- Clear **Auto-detect logout conditions**, click **OK** to save the settings, and then:

  a Click 

  and select **Add Logout Condition**.

  b Use the Forward and Back buttons  to navigate to a page that contains a logout element.

  c Do one of the following:

  – Click 

  and then click the page element that appears only when you are in a “logged out” condition.

  – If the entire page appears only after the user has logged out, click 

  .

  – If you want the scanner to search each page for a logout condition that matches a regular expression that you create, click 

  .

To delete a logout condition from the macro, click the **Logout Conditions** tab (in the Web Macro Recorder's lower pane), right-click a condition, and select **Delete**.

Using a Regular Expression for Logout Detection

If you want the scanner (and the Event-Based IE Compatible Web Macro Recorder) to use a regular expression to detect a logged out condition:

1 Select **Add Logout Regex**.

The Regular Expression Editor opens.

2 Enter a regular expression that identifies a unique text or phrase that occurs in the server's HTTP response when a user logs off or when a user who is not logged on requests access to a protected URL.

For example, if your server returns a message such as “Have a nice day” when a user logs off your application, then enter “Have\sa\snice\sday” as the regular expression (“\s” is used in regular expressions to denote a space).

The scanner can also detect that it has logged off if the server sends a specific message in response to the scanner's attempt to access a password-protected URL. For example, the server may respond with a status code of “302 Object moved.” In this case,

“[STATUSCODE]302 AND [ALL] http://login.myco.com/config/mail?”

might be a typical regular expression. See Chapter 10, Regular Expression Editor for more information.

3 Click **OK**.

Confirmation Elements (Hints)

When the *Playback Successful?* dialog appears, the second of three messages at the bottom of the dialog pertains to confirmation elements. These are elements or pages that indicate to the Web Macro Recorder (and the scanner) that the user is logged in to the site or application.

If the message is “Confirmation hints have been specified for this macro,” the Web Macro Recorder has recognized the element that you specified as indicating that the user is logged in.

However, if the message is “Unable to auto-detect confirmation hints (please add manually),” then one of the following occurred:

- You did not instruct the Web Macro Recorder to automatically detect confirmation elements (see Event-Based IE Compatible Web Macro Recorder Settings on page 219).
- You instructed the Web Macro Recorder to automatically detect confirmation elements, but the Web Macro Recorder could not recognize the element you specified (or you failed to specify an element).
- You did not manually specify a confirmation element.

To correct this error, click **Edit** → **Settings** → **Auto-Detection** and do one of the following:

- Select **Auto-detect confirmation hints** and choose one or more of the standard elements (or create a custom element).
- Clear **Auto-detect confirmation hints**, click **OK** to save the settings, and then:

  a. Click 

  and select **Add Confirmation Element**.

  b. Use the Forward and Back buttons  to navigate to a page that contains a confirmation element.

  c. Do one of the following:

  – Click 

  and then click the page element that appears only when you are in a “logged in” condition.

  – If the entire page appears only after the user has logged in, click 

  .

Unsupported Elements

While recording your macro, the Event-Based IE Compatible Web Macro Recorder displays a warning if you click an unsupported element. These non-HTML elements include objects created using the following technologies:

- Applets
- ActiveX
- Silverlight
- Flash
- Cross-Domain Iframes

If these objects are not required components of your macro, there is no problem. The Event-Based IE Compatible Web Macro Recorder simply ignores the object and continues to record events as you generate them by navigating through the site.

However, if an unsupported element contains an essential component (such as a login form), the macro will not succeed.

You might avoid this issue by switching to the Unified Web Macro Recorder (see Chapter 21, Unified Web Macro Recorder).

Event-Based IE Compatible Web Macro Recorder Settings

To modify the Event-Based IE Compatible Web Macro Recorder settings:

1. Click **Edit** → **Settings**.
2. Select either the **Application** or **Macro** category (described below) and enter the settings.
3. Click **OK**.

Application Settings

General

Show startup window

The startup window appears when the Event-Based IE Compatible Web Macro Recorder is launched. It displays a shortcut menu that allows you to begin creating or editing a login macro.

Compress macro files

Applies a compression algorithm to reduce the size of the saved macro.

Encrypt macro file

Applies an encryption algorithm to the saved macro to provide security.

Network Authentication Credentials

If network authentication is required, provide a user name and password that will allow access to the network.

Troubleshooting

Highlight failed events

If you select this option, the program displays failed events with a background color.

- Red highlight: The macro event caused the macro to fail.
- Orange highlight: The event failed, but playback continued.

Ignore events after final page load

In most cases, the events that occur after loading the final page in the macro are not significant and do not affect the playback of the macro.

Auto-Detection

During the recording process, you can manually specify a logout element and a confirmation element (an object that appears on the page to indicate that you have logged in successfully). If auto-detection is enabled and the program automatically detects a logout element during the recording process, the wizard that appears once playback is complete will reflect this and you will not be prompted to select a logout element.

To instruct the Web Macro Recorder to automatically detect elements, select **Auto-detect logout conditions** and/or **Autodetect confirmation hints**.

To identify which of the standard elements will trigger automatic detection, select or clear the associated check box next to the element in the Standard list.

To create a custom element:

1. Click **Add**.
2. In the **Value** box, enter a text string that appears somewhere within the page.
3. Click **OK**.

   The element appears in the Custom list.
4. In the **Type** column, click the down arrow and select the element type: **Confirmation** or **Logout**.

Proxy

If you need to use a proxy server to access the target website:

1. Select **Use Proxy**.
2. Enter the IP address or host name of the server.
3. Enter the server's port number.

Macro Settings

General

Smart Credentials

If you start a scan using a macro that includes Smart Credentials, then when you scan the page containing the input elements associated with these entries, WebInspect or WebInspect Enterprise will substitute the user name and password specified here. This allows you to create the macro using your own user name and password, yet when other persons run the scan using this macro, they can substitute their own user names and passwords.

To enable this feature, you must first record a macro and then associate one SetValue event in the Events grid as a user name and another SetValue event as a password.

Replacement URL

If you select **Enable URL Replacement**, the host name entered as the Start URL in the Scan Wizard will be dynamically inserted into each URL for this macro. For example, suppose you record a macro for www.testsite.com. At a later point in time, www.testsite.com is renamed to www.testsite2.com. Instead of recording an entirely new macro, you could reuse the original one and enable URL replacement.

IE Dialogs

Microsoft Internet Explorer may sometimes display dialogs that are not related to the actual content of the web page. For example, the browser's security feature may present a modal dialog with the following message: “Do you want to view only the webpage content that was delivered securely?” If this occurs during playback of a macro, the scanner will halt until the user presses **Yes** or **No**. You can avoid this interruption by selecting **Use IE Dialog Suppression**.

Several conditions are defined by default. You may, however, define a condition that meets your specific requirements. To do so:

1. Click **Add**.
2. Enter the requested information.
   - **Dialog Caption:** Enter the text that appears on the title bar of the dialog box.
   - **Dialog Text:** Enter the text that appears as the message content.
   - **Button:** Enter the text that appears on the button that the macro should automatically “press.”

   The utility that performs this check is case-sensitive, so be sure to enter the text string exactly as it appears.

Click **OK**.

23 Web Proxy

About the Web Proxy Tool

Web Proxy is a stand-alone, self-contained proxy server that you can configure and run on your desktop. With it, you can monitor traffic from WebInspect, a browser, or any other tool that submits HTTP requests and receives responses from a server. It is a tool for debugging and penetration scanning; you can see every request and server response while browsing a site.

You can also create a Workflow macro or a Login macro that you can use with WebInspect.

Before using Web Proxy with your browser, you must configure your browser's proxy settings. If using Internet Explorer:

1. Click **Tools** → **Internet Options**.
2. Click the **Connections** tab.
3. Click **LAN Settings**.
4. On the *LAN Settings* window, select **Use a Proxy Server for your LAN** and enter the address and port of the proxy server you want to use. By default, Web Proxy uses your local host settings (127.0.0.1:8080)

You should also configure Microsoft Internet Explorer to use HTTP 1.1 through proxy connections. On Internet Explorer:

1. Click **Tools** → **Internet Options**.
2. Click the **Advanced** tab.
3. In the "HTTP1.1 settings" section, select **Use HTTP 1.1 through proxy connections**.

Using Web Proxy

To use Web Proxy with a browser:

1. Click **Tools** → **Web Proxy**.

   The *Web Proxy* window opens.

If you want to create a Workflow macro from a set of Burp proxy files, you can click **File** → **Open**, change the file type in the drop-down list from **Proxy Session File (*.Psf)** to **Burp proxy (*.*)**, and then navigate to and open the Burp proxy files. See Creating a Web Macro on page 225.

2. Click  or select **Proxy** → **Start**.

   "Listening on <server:port number>" displays in the Web Proxy status bar.

3. Click **Launch Browser**  .

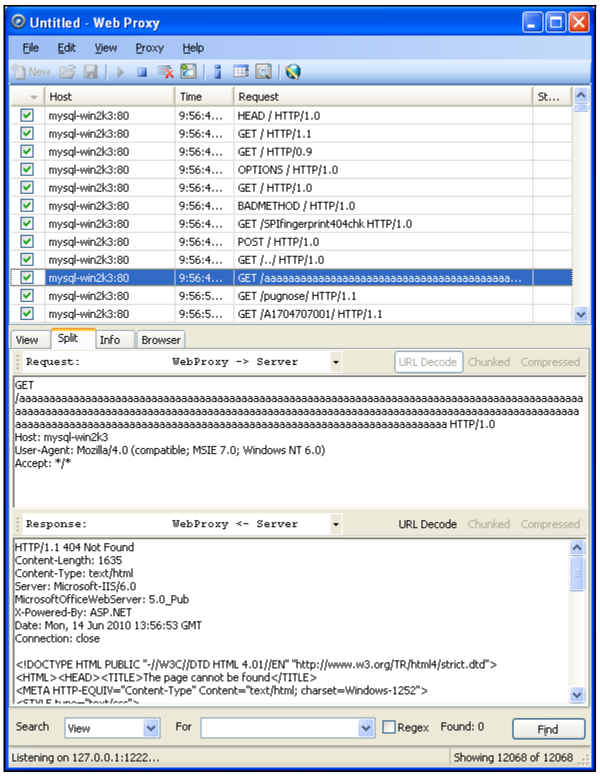
   This starts a web browser and configures it to communicate through Web Proxy.

4. Manually navigate the site for which you want to view requests/responses.

5 If Web Proxy receives a request for a certificate from a web server, it displays a dialog asking you to locate the certificate. The program then caches your selection on a “per server” basis. Therefore, if you subsequently want to use a different certificate for a particular server, you must clear the cache by stopping and then restarting Web Proxy.

6 When you have browsed to all necessary pages, return to Web Proxy and click (or click **Proxy** → **Stop**).

7 Each session (a request and matching response) you recorded is listed in the top pane. To view the actual HTTP message, select an entry. The message appears in the bottom frame. By default, the **View** tab is selected.



8 To change the format in which the message is displayed, select one of the tabs (**View**, **Split**, **Info**, or **Browser**).

When using the **View** or **Split** tabs, you can enable or disable URL decoding of requests and responses by selecting the **URL Decode** button. Since most WebInspect attack traffic is URL encoded, this feature makes it easier to analyze HTTP messages. To illustrate, compare the following URL encoded and decoded versions of the same GET request:

- GET /notes.asp?noteid=1%20union%20%20select%200%2c1%2c2%20from%20information_schema.tables%20order%20by%204%20desc%20limit%201 HTTP/1.1
- GET /notes.asp?noteid=1 union select 0,1,2 from information_schema.tables order by 4 desc limit 1 HTTP/1.1

The **Chunked** and **Compressed** buttons are enabled if a response is either chunked-encoded or compressed. This allows you to view the original response received by Web Proxy as well as the de-chunked or decompressed response.

9 To resend a request (with or without editing), select it from the list of displayed sessions and click the HTTP Editor icon (or right-click the request and select **HTTP Editor** from the context menu).

10 To clear sessions from the list, select one or more sessions and press the Delete key (or click **Edit** → **Clear Selected**). To clear all sessions, click  (or click **Edit** → **Clear All**).

Note: When you clear a session from the Web Proxy list, you also remove it from the captured data. For example, if you have 100 sessions in the list and clear 98 of them, and then save the sessions to a file, only the two remaining sessions will be included. When clearing sessions, ignore the check boxes.

Use the **File** menu to save selected requests to a proxy session file (.psf) and later load them for analysis (using the **File** → **Open** command). You can also save a sequence of requests as a Web Macro that you can use when conducting a WebInspect scan. All **File** menu commands apply to "check-marked" requests.

Click the top of any column to sort the requests by that selection. For example, to sort the requests by the time they were made, click the top border of the **Time** column.

You must stop Web Proxy when you want to change Web Proxy settings.

Creating a Web Macro

You can use either the Web Macro Recorder or Web Proxy to create a Workflow macro or a Login macro.

A Workflow macro is used most often to focus on a particular subsection of an application. It specifies URLs that an HP scanner will use to navigate to the area. It may also include login information, but does not contain logic that will prevent the scanner from logging out of your application.

A Login macro is used for Web form authentication, allowing the scanner to log in to an application. You can also incorporate logic that will prevent the scanner from inadvertently logging out of your application.

If you want to create a Workflow macro from a set of Burp proxy files, you can click **File** → **Open** in the menu bar of the Web Proxy tool, change the file type in the drop-down list from **Proxy Session File (*.Psf)** to **Burp proxy (*.*)**, and then navigate to and open the Burp proxy files.

To create a macro using sessions captured by Web Proxy:

1 Select the sessions you want to include in the macro by placing a check mark in the left column.

2 Click **File** → **Create Web Macro**.

3 (Optional) On the *Create Web Macro* window, select **Enable Check For Logout** and then enter a regular expression that identifies a unique text or phrase that occurs in the server's HTTP response when a user logs out or when a user who is not logged in requests access to a protected URL.

Background: During a normal scan, the scanner begins crawling your site at the home page. If it encounters a link to another resource (usually through an <A HREF> HTML tag), it will navigate to that URL and continue its scan. If it follows a link to a logout page (or if the server automatically "logs out" a client after a certain number of minutes), the scanner will not be able to visit additional resources where the client is required to be logged in. When this inadvertent logout occurs, the scanner must be able to log in again without user intervention. This process hinges on the scanner's ability to recognize when it is no longer logged in.

In some applications, if the user logs out (by clicking a button or some other control), the server responds with a unique message, such as "Have a nice day." If you specify this phrase as the server's logout condition, the scanner searches every response message for this phrase. Whenever it detects the phrase, the scanner attempts to log in again by sending an HTTP request containing the username and password.

The scanner can also detect that it has logged out if the server sends a specific message in response to the scanner's attempt to access a password-protected URL. For example, the server may respond with a status code of "302 Object moved." If the scanner knows specifically what to look for in this response, the program will recognize that it has been logged out and can re-establish a logged-in state.

Using the background example (above), if your server returns a message such as "Have a nice day" when a user logs out of your application, then enter "Have\sa\snice\sday" as the regular expression ("\s" is used in regular expressions to designate a space). As a more likely example, the server returns a 302 status code and references a new URL. In this case,

"[STATUSCODE]302 AND [ALL]http://login.myco.com/config/mail?"

might be a typical regex phrase.

4 Enter a name in the **Save macro as** box.

5 Click **OK**.

Web Proxy Tabs

Each HTTP session (a single request and the associated response) is listed in the top pane of Web Proxy. When you select a session, Web Proxy displays information about the session in the lower pane. The information displayed depends on which tab you select.

Web Proxy Tabs

Tab	Description
View	Use the **View** tab to select which HTTP messages you want to inspect. Options available from the drop-down list immediately below the tab are: • **Session**: view the complete session (both request and response) • **Request from browser to Web Proxy**: view only the request made by the browser to Web Proxy • **Request to server from Web Proxy**: view only the Web Proxy request to the server • **Response from server to Web Proxy**: view only the server response to Web Proxy • **Response to browser from Web Proxy**: view only the Web Proxy response to the browser
Split	Click the **Split** tab to create two information areas for a single session. For example, you could show the HTTP request message created by the browser (in one area) and the HTTP response generated by the server (in the second area). You can cut, paste, and copy the raw request, and right-click to see a shortcut menu of encoding options. However, you cannot save an edited request from the Web Proxy tool. Use the HTTP Editor to save an edited request.
Info	Use the **Info** tab to view detailed information about the requests. Information includes the number of forms found, header information, and the properties of the page.
Browser	Click the **Browser** tab to view the response as formatted in a browser.

Web Proxy Settings

To access this feature, click **Edit** → **Settings**.

You cannot change settings while Web Proxy is running. Click **Proxy** → **Stop**, change settings, and then restart Web Proxy.

Task 1: Configure General Settings

1. Select the **General** tab.
2. In the **Proxy Listener Configuration** group, enter an IP address and port number. By default, Web Proxy uses address 127.0.0.1 and port 8080, but you can change this if necessary.

   Both Web Proxy and your web browser must use the same IP address and port. If using Internet Explorer, click **Tools** → **Internet Options**; click the **Connections** tab and click **LAN Settings**; on the *LAN Settings* window, select **Use a Proxy Server for your LAN** and enter the address and port of the proxy server you want to use.

   To configure Web Proxy on your host to be used by another host, you will need to change the value of the Local IP Address. The default address of 127.0.0.1 is not available to outside hosts. If you change this value to your workstation's current IP address, remote stations can use your workstation as a proxy.
3. Use the **Do Not Record** option to create a regular expression filter that prevents files of specific types from being handled by Web Proxy. The most common types are already excluded as defaults, but other types (MPEG, PDF, etc.) can also be excluded. The purpose is to allow you to focus on HTTP request/response lines and headers by removing clutter from the message.
4. When using the interactive mode, you can force Web Proxy to pause when it:
   - Receives a request from the client.
   - Receives a response from the server.
   - Finds text that satisfies the search rules you create (using the **Flag** tab).

   If you select any of these options, Web Proxy will continue only when you click the **Allow** button.
5. In the **Logging** group, select the type of items you want to record in the log file and specify the directory in which the log file should be maintained. If you elect to record requests and/or responses, you can also choose to convert and log the data using Base 64 encoding. This can be useful when responses contain binary data (such as images or Flash files) that you want to examine.
   - Raw Request refers to the HTTP message sent from the client to Web Proxy.
   - Modified Request refers to the HTTP message sent from Web Proxy to the server.
   - Raw Response refers to the HTTP message sent from the server to Web Proxy.
   - Modified Response refers to the HTTP message sent from Web Proxy to the client.
6. Most web pages contain information that tells the browser what language encoding to use. This is accomplished by using a META tag with an HTTP-EQUIV attribute in the HEAD section of the HTML document, as in the following example:

   <meta http-equiv="Content-Type" content="text/html; charset=windows-1252">

   For pages that do not announce their character set, choose an option from the **Assumed 'charset' Encoding** list to select the language (and implied character set) that Web Proxy should use.

Task 2: Configure Proxy Servers Settings

1 Click the **Proxy Servers** tab.

Use this area to add one or more proxy servers through which Web Proxy will route all its requests. Distributing the attack across multiple servers makes detection and counter-measures more difficult, thus mimicking how a hacker might attempt to avoid an intrusion detection system.

If you use multiple proxy servers, Web Proxy will "round-robin" the requests (i.e., Web Proxy will sequence through the list of proxy servers, sending the first request to the first server, the second request to the second server, and so on).

2 In the **Proxy Address** box**,** type the IP address of the server through which you want to route Web Proxy requests.

3 Specify the port number in the **Proxy Port** box.

4 Select the type of proxy (standard, SOCKS4, or SOCKS5) from the **Proxy Type** list.

5 If this proxy server requires authentication, select an authentication type and enter your authentication credentials in the **Username** and **Password** boxes. See Authentication Types on page 44 for a description of the available authentication types.

6 Click **Add** to add the server and display its IP address in the **Available Proxy Servers** list.

You can also import a file containing a list of proxy servers by clicking **Import**. The file containing proxy information must be formatted as follows:

- Each line contains one record followed by a line feed and carriage return.
- Each field in the record is separated by a semicolon.
- The fields appear in the following order: address;port;proxytype;user name;password.
- The user name and the password are optional. However, if authorization is not used, you must include two semicolons as placeholders.

Examples:

- 128.121.4.5;8080;Standard;magician;abracadabra
- 127.153.0.3;80;socks4;;
- 128.121.6.9;443;socks5;myname;mypassword

7 If you do not need to use a proxy server to access certain URLs (such as internal testing sites), you can specify one or more hosts in the **Bypass Proxy List** area.

a Click **Add** in the **Bypass Proxy List** group.

The *Bypass Proxy* window appears.

b Enter the host portion of the HTTP URL that should be bypassed.

Do not include the protocol (such as http://).

For example, to bypass a proxy server for this URL

http://zero.webappsecurity.com/Page.html

enter this string

zero.webappsecurity.com

or this string

zero.*

You can also enter an IP address. Note that Web Proxy will not resolve host names to IP addresses. That is, if you specify an IP address and the HTTP request actually contains that numeric IP address, then Web Proxy will bypass a proxy server for that host. However, if the HTTP request contains a host name that resolves to the IP address that you specify, Web Proxy will still send the request to a proxy server.

c Click **OK**.

Task 3: Configure Search-and-Replace Settings

1 Click the **Search and Replace** tab.

Use this tab to create rules for locating and replacing text or values in HTTP messages. This feature provides a highly flexible tool for automating your simulated attacks. Some suggested uses include:

- Masking sensitive data, such as user names and passwords
- Appending a cookie to each request
- Modifying the Accept request-header field to add or delete media types that are acceptable for the response
- Replacing a variable in the Request-URI with a cross-site scripting attack

2 Click **Add** to create a default entry in the table.

3 Click the **Search Field** column of the entry.

4 Click the drop-down arrow and select the message area you want to search.

5 In the **Search For** column, type the data (or a regular expression representing the data) you want to find.

6 In the **Replace With** column, type the data you want to substitute for the found data.

7 Repeat this procedure to create additional search rules.

Search-and-replace rules are executed on request messages sent from Web Proxy to the Server and on response messages sent from Web Proxy to the Browser. You can observe the altered messages by choosing the **Info** tab, or by selecting either the **View** or **Split** tab and then choosing one of the following from the drop-down list immediately below the tab:

- Request: WebProxy -> Server
- Response: Browser <- WebProxy
- Session

The request/response rules are applied sequentially, in the order in which they appear. For example, if a rule changes HTTPS to SSL, and if a subsequent rule then changes SSL to SECURE, the result will be that HTTPS is changed to SECURE.

Task 4: Configure Flag Settings

1 Click the **Flag** tab.

This feature allows you to find and highlight keywords in requests or responses.

2 Click **Add** to create a default entry in the table.

3 Click the **Search Field** column of the entry.

4 Click the drop-down arrow and select the message area you want to search.

5 In the **Search** column, type the data (or a regular expression representing the data) you want to find.

6 Click the **Flag** column of the entry.

7 Click the drop-down arrow and select a color with which to highlight the data, if found.

Task 5: Configure Evasion Settings

Evasions are techniques that Web Proxy uses to circumvent intrusion detection systems, monitors, sniffers, firewalls, log parsers, or any device that attempts to shield systems from attack by filtering HTTP requests. Typically, these filters examine portions of the request, searching for "signatures" that indicate malicious threats or potential breeches of system security. If they detect these signatures, they reject the request.

To evade detection, Web Proxy modifies the HTTP request to obscure the signature for which the filter is searching, while retaining integrity sufficient for the message to be processed by the server. Of course, the techniques used by Web Proxy are not always successful. As developers become aware of methods that compromise their product's effectiveness, they incorporate procedures to combat them.

This feature is intended for use as a penetration testing tool. Do not use it or enable it when conducting vulnerability scans with WebInspect.

Use the following procedure to enable evasions:

1. Select the **Evasions** tab.
2. Select **Enable Evasions**.

Choose one or more evasion techniques, as described below.

Method Matching

Web Proxy replaces the GET method with HEAD. This is an attempt to defeat a filter that searches for a signature that begins with GET.

For example, the browser sends the following message to Web Proxy:

```
GET http://www.microsoft.com/ HTTP/1.1
```

Web Proxy sends the following message to the server:

```
HEAD http://www.microsoft.com/ HTTP/1.1
```

URL Encoding

Web Proxy converts characters in the URL to a "%" followed by two hexadecimal digits corresponding to the character values in the ISO-8859-1 character set.

For example, the browser sends the following message to Web Proxy:

```
GET http://zero.webappsecurity.com/cgi-bin/filename.cgi HTTP/1.1
```

Web Proxy sends the following message to the server:

```
GET %2f%63%67%69%2d%62%69%6e%2f%66%69%6c%65%
6e%61%6d%65%2e%63%67%69 HTTP/1.1

Host: zero.webappsecurity.com
```

If the device is looking for "cgi-bin" as the signature, it does not match the string "%63%67%69%2d%62%69%6e" and so the request is not rejected.

Double Slashes

Web proxy converts each forward slash (/) into a double forward slash (//).

For example, the browser sends the following message to Web Proxy:

```
GET http://www.microsoft.com/en/us/secrets.aspx HTTP/1.1
```

Web Proxy sends the following message to the server:

```
GET //en//us//secrets.aspx HTTP/1.1
Host: www.microsoft.com
```

If the device is looking for "/secrets.aspx" as the signature, it does not match the string "//secrets.aspx" and so the request is not rejected.

Reverse Traversal

This technique attempts to disguise a request for a certain resource by interjecting references to relative directories, which equates to the original request.

For example, the browser sends the following message to Web Proxy:

```
GET http://www.TargetSite.com/cgi-bin/some.cgi HTTP/1.1
```

Web Proxy sends the following message to the server:

`GET /d/../cgi-bin/d/../some.cgi HTTP/1.1` [which equates to GET/cgi-bin/some.cgi]

```
Host: www.TargetSite.com
```

Self-Reference Directories

Web Proxy uses the notation for parent directory (../) and current directory (./) to obfuscate the request.

For example, the browser sends the following message to Web Proxy:

```
GET http://www.TargetSite.com/cgi-bin/phf HTTP/1.1
```

Web Proxy sends the following message to the server:

```
GET /./cgi-bin/./phf HTTP/1.1 [which equates to GET /cgi-bin/phf]
Host: www.TargetSite.com
```

Parameter Hiding

A request can contain parameters that are used to build dynamic page content. These parameters are typically used when search requests or selections are made and take this form:

```
/anypage.php?attack=paramhiding&evasion=blackhat&success...
```

This technique is effective against a device that does not examine that portion of the request following the question mark (?). However, the parameter indicator can be used to potentially mask further relevant data.

For example, the browser sends the following message to Web Proxy:

```
GET /index.htm%3fparam=/../cgi -bin/test.cgi
```

Web Proxy sends the following message to the server:

```
GET /index.htm?param=/../cgi -bin/test.cgi
```

HTTP Misformatting

An HTTP request has a clearly defined structure:

Method<space>URI<space>HTTP/Version<CRLF><CRLF>

However, some web servers will accept a request that contains a tab character instead of a space, as in the following:

Method<tab>URI<tab>HTTP/Version<CRLF><CRLF>

Any filter that incorporates the space (between the three components) as part of the signature for which it searches will fail to reject the request.

Long URLs

This technique is directed toward devices that do not examine the entire request string, but concentrate only on a subset of a programmable length (such as the first 50 characters). Web Proxy inserts a large number of random characters at the beginning of the request so that the operative portion of the request is pushed beyond the area normally examined by the filter.

For example, the browser sends the following message to Web Proxy:

```
GET http://zero.webappsecurity.com/ HTTP/1.1
```

Web Proxy sends the following message to the server:

```
GET /YPVIFAHD[hundreds of characters]NIWCJBXZPXMP/../ HTTP/1.1
Host: zero.webappsecurity.com
```

DOS/Win Directory Syntax

A Windows-based filter that attempts to detect a specific signature (such as /cgi-bin/some.cgi) might be fooled if a backward slash is substituted for a forward slash (such as /cgi-bin\some.cgi). Windows-based web servers convert a forward slash to a backward slash when interpreting directory structures, so the notation is valid. However, HTTP rules require the first character of a URI to be a forward slash.

NULL Method Processing

This technique injects a URL-encoded NULL character immediately after the METHOD (such as GET%00). It is designed for a device that attempts to apply string operations on the request, and those string libraries use the NULL character to denote the end of a string. If this ploy is successful, detection of the NULL character prevents the device from examining the remainder of the message.

Case Sensitivity

This technique is designed to evade a filter that searches for a case-specific string.

For example, the browser sends the following message to Web Proxy:

```
GET http://zero.webappsecurity.com/cgi-bin/some.cgi HTTP/1.1
```

Web Proxy sends the following message to the server:

```
GET /CGI-BIN/SOME.CGI HTTP/1.1
Host: zero.webappsecurity.com
```

Web Proxy Interactive Mode

Use Interactive mode to view each browser request and each server response as the messages arrive at Web Proxy. The message will not continue toward its destination until you click **Allow**. This permits you to modify the message before it is delivered.

You can also prevent the message from being sent to the server by clicking **Deny**.

Using the **Proxy** tab on the *Web Proxy Settings* window, you can force Web Proxy to pause after each request, after each response, or after locating specific text in either the request or response.

To turn on interactive mode:

1. Click **Proxy** → **Stop**.
2. Click **Proxy** → **Interactive**
   -or-
   click  on the toolbar.
3. Click **Proxy** → **Start**.

When Web Proxy is in Interactive mode, a check mark appears next to the Interactive command on the **Proxy** menu and the Interactive icon is backlit. Clicking the icon or selecting the command will toggle the Interactive mode on or off.

24 Web Service Test Designer

About the Web Service Test Designer Tool

Web services are programs that communicate with other applications (rather than with users) and answer requests for information. Most web services use Simple Object Access Protocol (SOAP) to send XML data between the web service and the client web application that initiated the information request. Unlike HTML, which only describes how web pages are displayed, XML provides a framework to describe and contain structured data. The client web application can readily understand the returned data and display that information to the end user.

A client web application that accesses a web service receives a Web Services Definition Language (WSDL) document so that it can understand how to communicate with the service. The WSDL document describes the programmed procedures included in the web service, the parameters those procedures expect, and the type of return information the client web application will receive.

Use the Web Service Test Designer to create a Web Service Test Design file (filename.wsd) containing the values that WebInspect should submit when conducting a web service scan.

Although the following procedure invokes the Web Service Test Designer from the WebInspect **Tools** menu, you can also open the designer from the HP Security Toolkit or through the WebInspect Scan Wizard by selecting **Start a Web Service Scan** from the WebInspect Start page and, when prompted, electing to launch the designer.

When the Web Service Test Designer is launched from the WebInspect Scan Wizard, if the WSDL has not yet been configured, the designer will automatically import the WSDL, assign “auto values” to each parameter, and invoke all operations. This does not occur when you launch the tool from the WebInspect **Tools** menu or from the HP Security Toolkit.

1. Select **Tools** → **Web Service Test Designer**.
2. On the startup dialog, select one of the following:
   - **New Web Service Test** - Design a new web service test.
   - **Open Web Service Test** - Edit a design that you previously created.

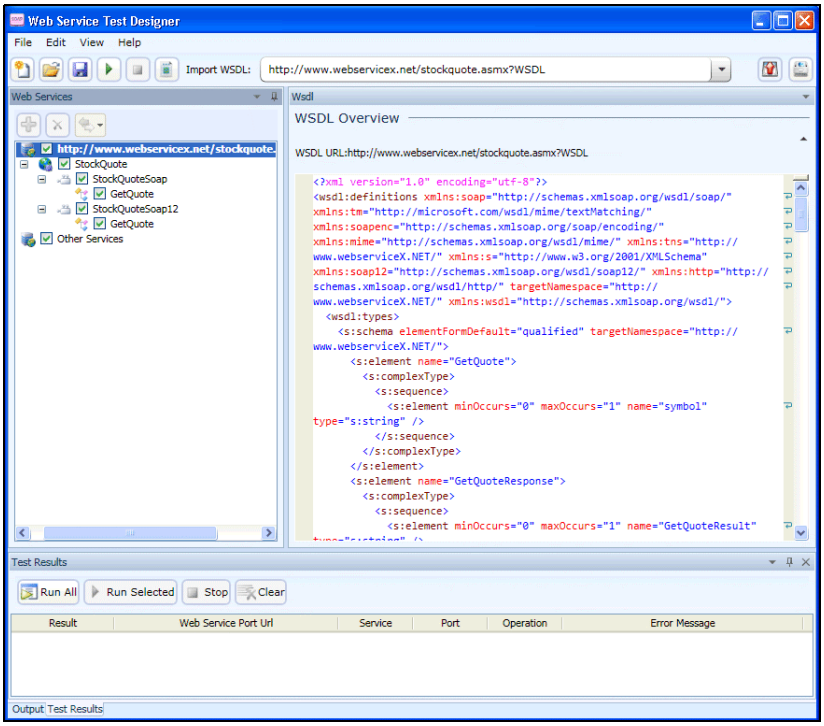
   The following procedure assumes that you are creating a design.
3. Do one of the following:
   - In the **Import WSDL** box, type or select the URL of the WSDL site and click **Import WSDL**  .

     Example: http://www.webservicex.net/stockquote.asmx?WSDL.
   - Click **Browse for WSDL**  and select a WSDL file that you previously saved locally.

   If authentication is required, or if SOAP requests need to be made through a proxy server, see Web Service Test Designer Settings on page 249 for more information.

   Also note that “Other Services” appears by default. This feature is used to add services manually when a service is not associated with a WSDL. See Manually Adding Services on page 244 for more information. Remove the check mark next to this item.

The WSDL endpoint (typically represented by a simple http URL string) appears in the left pane, followed by the service name and a hierarchical listing of the operations defined for that service. The right pane (by default) contains the WSDL URL and, when available, the namespace, binding namespace, and the port location.



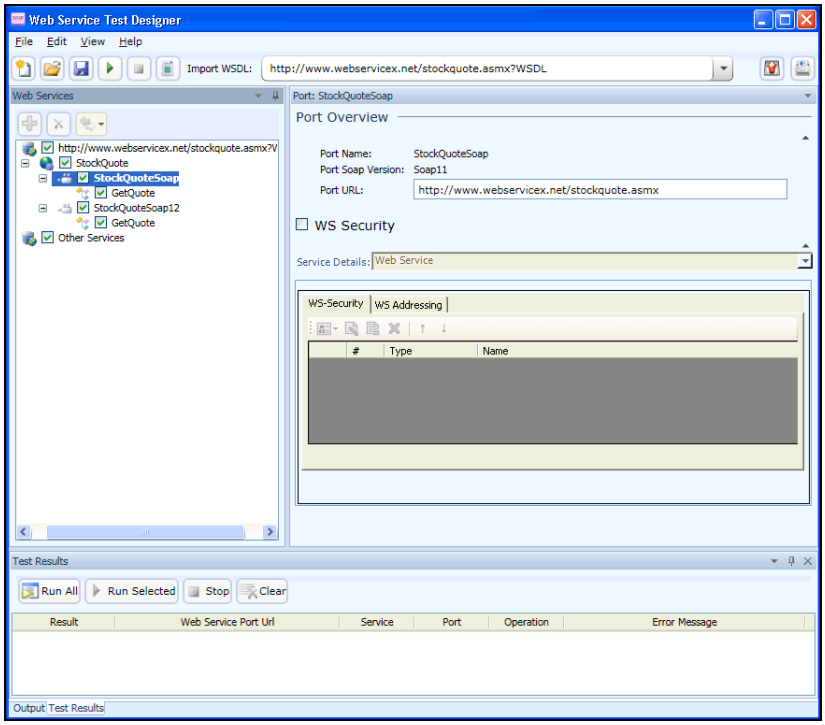
The above illustration shows a simple WSDL that returns the current stock price and other related information when the user submits a corporate symbol used by the New York Stock Exchange.

4 Select a service transport in the left pane to display the port information in the right pane. A port defines an individual endpoint by specifying an address for a binding. Note that if the description of the WSDL includes both SOAP version 1.1 and version 1.2, and if the operations in both descriptions are the same, the versions are assumed to be identical and the services in version 1.1 only are configured. If you wish to attack both versions, then you must select the check box for each version 1.2 operation.

Note: The Port Overview panel for SOAP version 1.2 contains an additional option to include SOAP action in the HTTP header.

Even though the SOAP specification states that the SOAP Action is optional for SOAP version 1.2, some architectures require it and some cannot accept it. You can choose to include or exclude the SOAP action for a SOAP 1.2 binding, depending on your specific environment. This check box appears for SOAP 1.2 ports only and defaults to true.

RPC-encoded services require manual configuration. The **Schema Fields** tab is populated using a default SOAP schema. You can obtain the desired SOAP message from a developer or a proxy capture, and then paste the message into the **XML** tab (or import the saved message from a file). You can then click **Send** to test the operation.



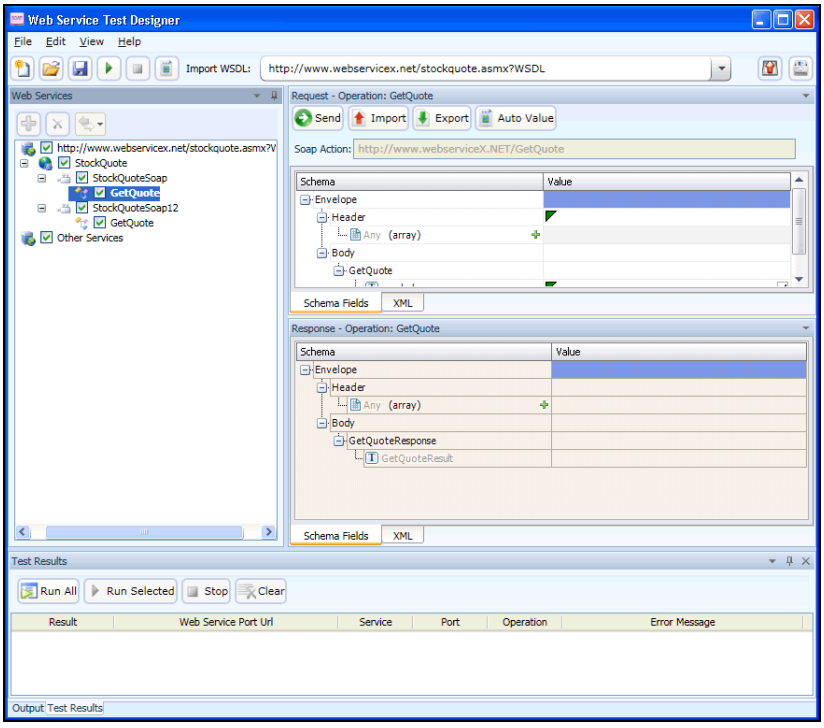
5 If security is required:

a Select **WS Security**.

b Select an option from the **Service Details** list.

c Provide the required information. For help with security settings, see WS Security Settings on page 238.

6 Click an operation to display schema for the request (in the top half of the right pane) and the response (in the lower half).



7 Enter a value for each parameter in the operation. In this example, the user entered HPQ (the NYSE symbol for Hewlett Packard).

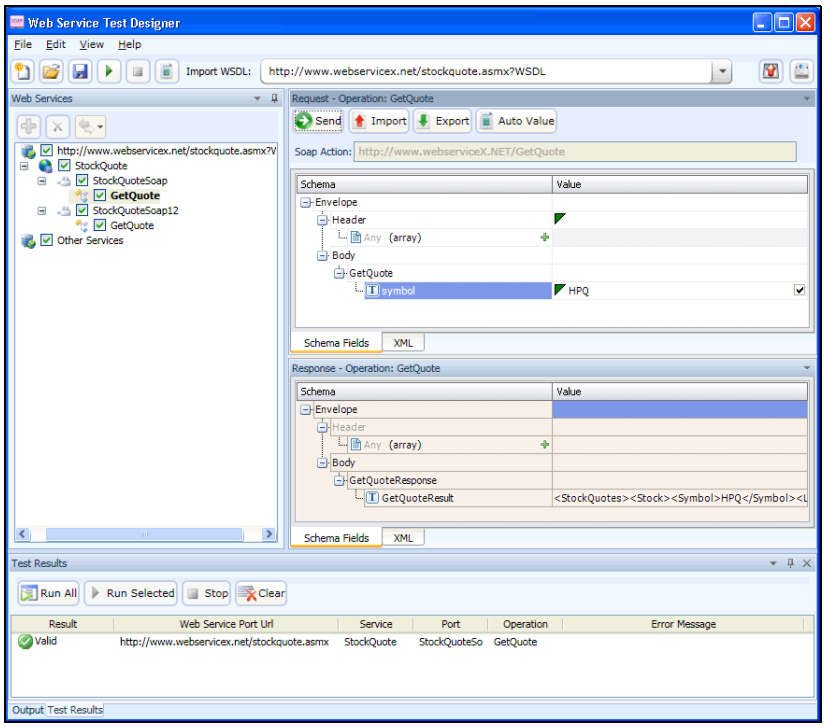
If you click **Auto Value**, the designer assigns a value to the operation. This value is either:

- Obtained from the GlobalValuesDefault.xpr file, if the file contains an entry that matches the name of the parameter; see Global Values Editor on page 245 for more information.
- Created by the designer, based on the data type. In this example, the designer would populate the parameter “symbol” with the value “symbol1.”

See Using Autovalues on page 246 for more information.

8 Click **Send**  .

Results appear in the lower portion. You can alternate between the Schema and XML views by clicking the appropriate tabs.



9 When you have assigned and tested values for each operation (although only one operation is depicted in this example):

a Click **File** → **Save**.

b Using the standard file-selection dialog, select a name and location for the Web Service Design file (.wsd).

If the WSDL contains multiple operations, data is saved for each operation regardless of whether or not the operation is checked. A check mark simply indicates that the operation will be used for auditing.

WS Security Settings

You can configure security settings for all operations in a web service port, using a variety of services:

- Web Service
- Windows Communication Foundation (WCF) Service (CustomBinding)

- WCF Service (Federation)
- WCF Service (WSHttpBinding)

Select an appropriate service from the Service Details list and then provide the requested information.

Web Service

When Security credentials, known as tokens, are placed in the SOAP request, the web server can verify that the credentials are authentic before allowing the web service to execute the application. To further secure web services, it is common to use digital signatures or encryption for the SOAP messages. Digitally signing a SOAP message verifies that the message has not been altered during transmission. Encrypting a SOAP message helps secure a web service by making it difficult for anyone other than the intended recipient to read the contents of the message.

WS-Security Tab

To add a security token, click  , select a token type, and provide the requested information.

UserName. This token specifies a user name and password. You can elect to include a nonce, specify how to send the password to the server for authentication (Text, None, or Hash) and indicate whether to include a timestamp.

X509 Certificate. This token is based on an X.509 certificate. You can purchase a certificate from a certificate authority, such as VeriSign, Inc., or set up your own certificate service to issue a certificate. Most Windows servers support the public key infrastructure (PKI), which enables you to create certificates. You can then have it signed by a certificate authority or use an unsigned certificate. Select a certificate and specify the reference type (BinaryCertificateToken or Reference).

Kerberos /Kerberos2. (For Windows 2003 or XP SP1 and later). The Kerberos protocol is used to mutually authenticate users and services on an open and unsecured network. Using shared secret keys, it encrypts and signs user credentials. A third party, known as a Kerberos Key Distribution Center (KDC), authenticates the credentials. After authentication, the user may request a service ticket to access one or more services on the network. The ticket includes the encrypted, authenticated identity of the user. The tickets are obtained using the current user's credentials. The primary difference between the Kerberos and Kerberos2 tokens is that Kerberos2 uses the Security Support Provider Interface (SSPI), so it does not require elevated privileges to impersonate the client's identity. In addition, the Kerberos2 security token can be used to secure SOAP messages sent to a web service running in a web farm. Specify the host and domain.

SAML Token. Security Assertion Markup Language (SAML) is an XML standard for exchanging security-related information, called assertions, between business partners over the Internet. The assertions can include attribute statements, authentication, decision statements, and authorization decision statements. Click Load from file to browse to a SAML certificate. Click Certificate to import a certificate. Finally, select a certificate reference type: X509 Data or RSA.

To add a message signature, click  and provide the requested information.

Signing token. The token to use for signing, usually an X.509 type. Select from the list of all added tokens.

Canonicalization algorithm. A URL for the algorithm to use for canonicalization. A drop-down list provides common algorithms. If you are unsure which value to use, keep the default.

Transform algorithm. A URL for the Transform algorithm to apply to the message signature. A drop-down list provides common algorithms. If you are unsure which value to use, keep the default.

Inclusive namespaces list. A list of comma-separated prefixes to be treated as inclusive (optional).

What to sign. The SOAP elements to sign: SOAP Body, Timestamp, and WS-Addressing.

XPath (optional). An XPath that specifies which parts in the message to sign. If left blank, the elements selected in the Signature options field are signed. For example, //*[local-name(.)='Body'].

Token (optional). The target token you want to sign. Select from the drop-down list of all added tokens. With most services, this field should be left empty.

To add message encryption, click  and provide the requested information.

Encrypting token. The token to use for encryption (usually an X.509 type). You can select from a list of all previously created tokens.

Encrypting type. Indicates whether to encrypt the whole destination Element or only its Content.

Key algorithm. The algorithm to use for the encryption of the session key: RSA15 or RSAOAEP.

Session algorithm. The algorithm to use for the encryption of the SOAP message. You can select from a list of common values.

XPath (optional). An XPath that indicates the parts of the message to encrypt. If left blank, only the SOAP body is encrypted.

Token (optional). The name of the encrypted token. A drop-down box provides a list of all added tokens. With most services, this field should be left empty.

Use the Up and Down arrows  to position the security elements in order of their priority.

WS Addressing

Use the **WS-Addressing** tab to indicate whether WS-Addressing is used by the service, and if so, its version number.

WCF Service (CustomBinding)

WCF Service (CustomBinding) enables the highest degree of customization. Since it is based on WCF customBinding standard, it allows you to test most WCF services, along with services on other platforms such as Java-based services that use the WS - <spec_name> specifications.

Transport. Select HTTP, HTTPS, or AutoSecuredHTTP. Named Pipes and TCP transport are not supported.

Encoding. Select Text, MTOM, or WCF Binary.

Security. Select an authentication mode and bootstrap policy from the appropriate list.

Net Security. The type of stream security: None, Windows stream security, or SSL stream security.

Reliable Messaging. Select Enabled to use reliable messaging and then select a format: either Ordered or Not Ordered.

Identities. Provide identity information for the bindings and certificate:

- **Username and Password**
- **Server Certificate/Client certificate**. A certificate that provides identity information for the server or client. Use the Browse button to open the Select Certificate Dialog Box.
- **Expected DNS, SPN, and UPN**. The expected identity of the server in terms of its DNS, SPN, or UPN. This can be localhost, an IP address, or a server name.

Client Windows Identity. Provide identity information for the client windows:

- **Current User**. The identity of the user logged onto the machine.
- **Custom User**. Specify the Username, Password, and Domain.

Click **Advanced** to open the *Advanced Settings* dialog. See Advanced Security Settings on page 242.

WCF Service (Federation)

When using WCF Service (Federation), the client authenticates against the Security Token Service (STS) to obtain a token. The client uses the token to authenticate against the application server.

Server

- **Transport**. The transport type: HTTP or HTTPS.
- **Encoding**. The server's encoding policy: Text or MTOM.

Security

- **Authentication mode**. A drop-down list of possible modes of authentication, such as AnonymousForCertificate, MutualCertificate, and so forth.
- **Bootstrap Policy**. A drop-down list of possible bootstrap policies for Secure Conversation authentication, such as SspiNegotiated, UserNameOverTransport, and so forth.

Identities. The identity information for the bindings and certificate:

- **Server certificate**. A certificate that provides identity information for the server. Use the Browse button to open the Select Certificate Dialog Box.
- **Expected DNS**. The expected identity of the server in terms of its DNS. This can be localhost, an IP address, or a server name.

STS (Security Token Service) Details

- **Endpoint address**. The endpoint address of the STS. This can be localhost, an IP address, or a server name.
- **Binding**. The scenario which references the binding that contacts the STS.

Click **Advanced** to open the *Advanced Settings* dialog. See Advanced Security Settings on page 242.

WCF Service (WSHttpBinding)

Using WCF Service (WSHttpBinding), you can choose from several types of authentication: None, Windows, Certificate, or Username (message protection).

None

Negotiate server credentials. Negotiates the web service's certificate with the server. You can also provide the server's DNS information.

Specify service certificate. The location of the service's certificate. If you select this option, the Negotiate service credentials option is not relevant.

Expected server DNS. The expected identity of the server in terms of its DNS. This can be localhost, an IP address, or a server name. It can also be the common name by which the certificate was issued.

Windows

Expected server identity. The service principal name (SPN) or user principal name (UPN). SPN ensures that the SPN and the specific Windows account associated with the SPN identify the service. UPN ensures that the service is running under a specific Windows user account; the user account can be either the current logged-on user or the service running under a particular user account.

Client Windows identity. The identity information for the client windows:

- **Current User**. Use the credentials of the user logged onto the machine.
- **Custom User**. Provide the user credentials (Username, Password, and Domain) and optionally select an impersonation level (which determines the operations a server can perform in the client's context).

Impersonation Level	Description
None	No level selected.
Anonymous	The server cannot impersonate or identify the client.
Identification	The server can get the identity and privileges of the client, but cannot impersonate the client.
Impersonation	The server can impersonate the client's security context on the local system.
Delegation	The server can impersonate the client's security context on remote systems.

Enable secure session. Allows a secure session using Windows type authentication.

Certificate

Client certificate. The location of the client certificate. The Browse button opens the Select Certificate Dialog Box.

Negotiate server credentials. Negotiates the web service's certificate with the server. You can also provide the server's DNS information.

Specify service certificate. The location of the service's certificate. If you select this option, the Negotiate server credentials option is disabled.

Expected server DNS. The expected identity of the server in terms of its DNS. This can be localhost, an IP address, or a server name. It can also be the common name by which the certificate was issued.

Enable secure session. Allows a secure session using Certificate type authentication.

Username (Message Protection)

Username, Password. The authentication credentials of the client.

Negotiate server credentials. Negotiates the web service's certificate with the server. You can also provide the server's DNS information.

Specify service certificate. The location of the service's certificate. If you select this option, the Negotiate server credentials option is disabled.

Expected server DNS. The expected identity of the server in terms of its DNS. This can be localhost, an IP address, or a server name. It can also be the common name by which the certificate was issued.

Enable secure session. Allows a secure session using Username type authentication.

Advanced Security Settings

This dialog box allows you to customize the security settings for your test on the following tabs.

Encoding Tab

Encoding. The encoding type to use for the messages: Text, MTOM, or WCF Binary.

WS-Addressing version. The version of WS-Addressing for the selected encoding: None, WSA 1.0, or WSA 04/08.

Advanced Standards Tab

Reliable messaging. Enables reliable messaging for services that implement the WS-ReliableMessaging specification. The encoding type to use for the messages: Text, MTOM, or WCF Binary.

Reliable messaging ordered. Indicates whether the reliable session should be ordered.

Reliable messaging version. The version to apply to the messages: WSReliableMessagingFebruary2005 or WSReliableMessaging11.

Specify via address. Sends a message to an intermediate service that submits it to the actual server. This may also apply when you send the message to a debugging proxy. This corresponds to the WCF clientVia behavior. This is useful to separate the physical address to which the message is actually sent, from the logical address for which the message is intended.

Via address. The logical address to which to send the message. It may be the physical of the final server or any name. It appears in the SOAP message as follows:

<wsa:Action>http://myLogicalAddress<wsa:Action>

The logical address is retrieved from the user interface. By default, it is the address specified in the WSDL. You can override this address using this field.

Drop down section Security Tab

Enable secure session. Establish a security context using the WS-SecureConversation standard.

Negotiate service credentials. Allow WCF proprietary negotiations to negotiate the service's security.

Default algorithm suite. The algorithm to use for symmetric/asymmetric encryption. The list of algorithms is populated from the SecurityAlgorithmSuite configuration in WCF.

Protection level. Indicates whether the SOAP Body should be encrypted/signed. The possible values are: None, Sign, and Encrypt And Sign (default)

Message protection order. The order for signing and encrypting. Choose from: Sign Before Encrypt, Sign Before Encrypt And Encrypt Signature, Encrypt Before Sign.

Message security version. The WS-Security security version. You can also indicate whether to require derived keys for the message.

Security header layout. The layout for the message header: Strict, Lax, Lax Timestamp First, or Lax Timestamp Last.

Key entropy mode. The entropy mode for the security key. The possible values are: Client Entropy, Security Entropy, and Combined Entropy.

Require security context cancellation. Indicates whether to require the cancellation of the security context. If you disable this option, stateful security tokens will be used in the WS-SecureConversation session, if they are enabled.

Include timestamp. Includes a timestamp in the header.

Allow serialized signing token on reply. Enables the reply to send a serialized signing token.

Require signature confirmation. Instructs the server to send a signature confirmation in the response.

Note: The next four options apply only when using an X.509 certificate.

X509 Inclusion Mode. Specifies when to include the X.509 certificate: Always to Recipient. Never, Once, Always To Initiator.

X509 Reference Style. Specify how to reference the certificate: Internal or External.

X509 require derived keys. Indicates whether X.509 certificates should require derived keys.

X509 key identifier clause type. The type of clause used to identify the X.509 key: Any, Thumbprint, Issuer Serial, Subject Key Identifier, Raw Data Key Identifier.

HTTP & Proxy Tab

This tab lets you set the HTTP and Proxy information for your test.

Transfer mode. The transfer method for requests/responses. The possible values are Buffered, Streamed, Streamed Request, and Streamed Response.

Max response size (KB). The maximum size of the response before being concatenated.

Allow cookies. Indicates whether to enable or disable cookies.

Keep-Alive enabled. Indicates whether to enable or disable keep-alive connections.

Authentication scheme. The HTTP authentication method: None, Digest, Negotiate, NTLM, Integrated Windows Authentication, Basic, or Anonymous.

Realm. The realm of the authentication scheme in the form of a URL.

Require client certificate. Indicates whether to require a certificate for SSL transport.

Use default web proxy. Indicates whether to use machine's default proxy settings.

Bypass proxy on local. Indicates whether to ignore the proxy when the service is on the local machine.

Proxy address. The URL of the proxy server.

Proxy authentication scheme. HTTP authentication method on Proxy: Digest, Negotiate, NTLM, Basic, or Anonymous.

Manually Adding Services

You may encounter a web service that does not have a WSDL associated with it.

For example, the WebInspect Recommendations module monitors scans to detect omissions, abnormalities, or anomalies that interfere with or diminish the thoroughness of a scan. If it detects SOAP requests during a Web Site scan, it suggests that you conduct a Web Service scan of that site and creates a Web Service Test Design file (filename.wsd) for that purpose. A WSDL file probably will not be available.

You may create a service manually, as shown in the following example.

1. Right-click the default "Other Services" service and select **Add Service**.

   New Service 1 appears in the Web Services tree in the left pane.

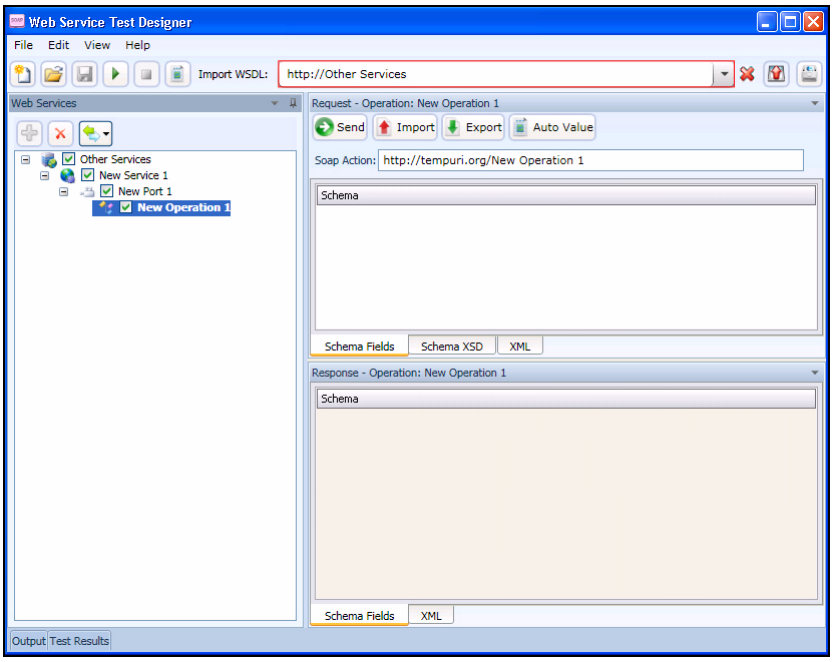
2. If authentication is required, select **WS Security** and provide the required credentials.

3. Right-click New Service 1, select **Add Port**. and then choose either **SOAP 1.1** or **SOAP 1.2**.

   New Port 1 appears in the Web Services tree.

4. In the **Port URL** box, enter the correct URL to the service.

5 Right-click New Port 1 and select **Add Operation**.



Note: To change service, port, or operation names, double-click the name.

6 You can import a file containing a SOAP envelope (possibly obtained using the Web Proxy tool) or you can copy and paste a SOAP envelope that you obtained from a developer onto the **XML** tab.

If importing from a proxy capture, the SOAP action will be in the HTTP header (Soapaction=<action_name>).

7 If necessary, modify the values using either the **Schema Fields** tab or the **XML** tab.

8 To test the service, click either **Send** or **Run All**.

Global Values Editor

You can create a library of name/value parameters for operations that you frequently encounter. After importing a WSDL file, if you click Set Auto Values  , the Web Service Test Designer searches the Global Values file for the names of parameters contained in the WSDL operations. If it finds a matching name, it inserts the associated value from the file into the parameter value field.

To add a global value:

1 Click **Edit** → **Global Values Editor**.

The Global Values Editor opens and displays the contents of the default xml parameter registry (xpr) file named GlobalValuesDefault.xpr.

2 Click **Add**.

This creates an entry with the default name of [Name] and a default value of [Value].

3 Click anywhere on the entry and substitute an actual name and value for the default.

4 Repeat step 2 and step 3 to create additional entries.

5 Do one of the following:

- Click **OK** to save and close the file.
- Click **Save As** to create and close the file using a different file name and/or location.

Importing and Exporting Operations

You can build a library of operations and their assigned values, allowing you to quickly modify other web service designs or exchange these components with other developers/testers. Each module is saved as an XML file, such as the following request used in the preceding example:

```
<Envelope xmlns="http://www.w3.org/2003/05/soap-envelope">
 <Header />
 <Body>
  <GetQuote xmlns="http://www.webserviceX.NET/">
   <symbol>HPQ</symbol>
  </GetQuote>
 </Body>
</Envelope>
```

To save or import an operation:

1. Select an operation in the left pane.
2. Click **Import Request**  to load the operation.
3. Click **Export Request**  to save the operation.

Using Autovalues

Use the Autovalues feature as an alternative to manually entering specific values for each parameter. The Web Service Test Designer analyzes each parameter and inserts a value that is likely to fulfill the service requirement. This can save considerable time when dealing with large web services.

After selecting a WSDL file:

1. Place a check mark next to each operation you want to autofill.
2. Click **Set Auto Values**.

   The following message appears: “Would you like the default values to be replaced with the defined global values?”

   If you click **Yes**, any values you may have entered manually will be erased. Also, if any parameter name in any operation matches a parameter name in the Global Values file, the associated value in the file will be substituted for the value that would normally be generated for the operation.

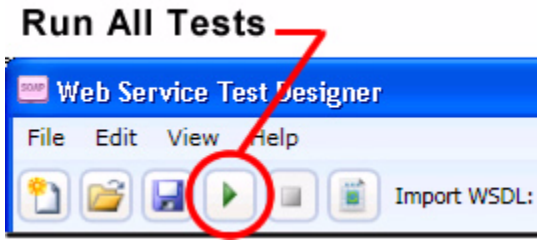
   If you click **No**, the function terminates.
3. Click **Yes**.
4. Click **Run All Tests**.

   The Web Service Test Designer submits the service request, with values inserted for each operation.
5. Click the **Test Results** tab (at the bottom of the window).
6. If an operation returned an error, double-click the operation to open it in the Request pane and manually provide a value.

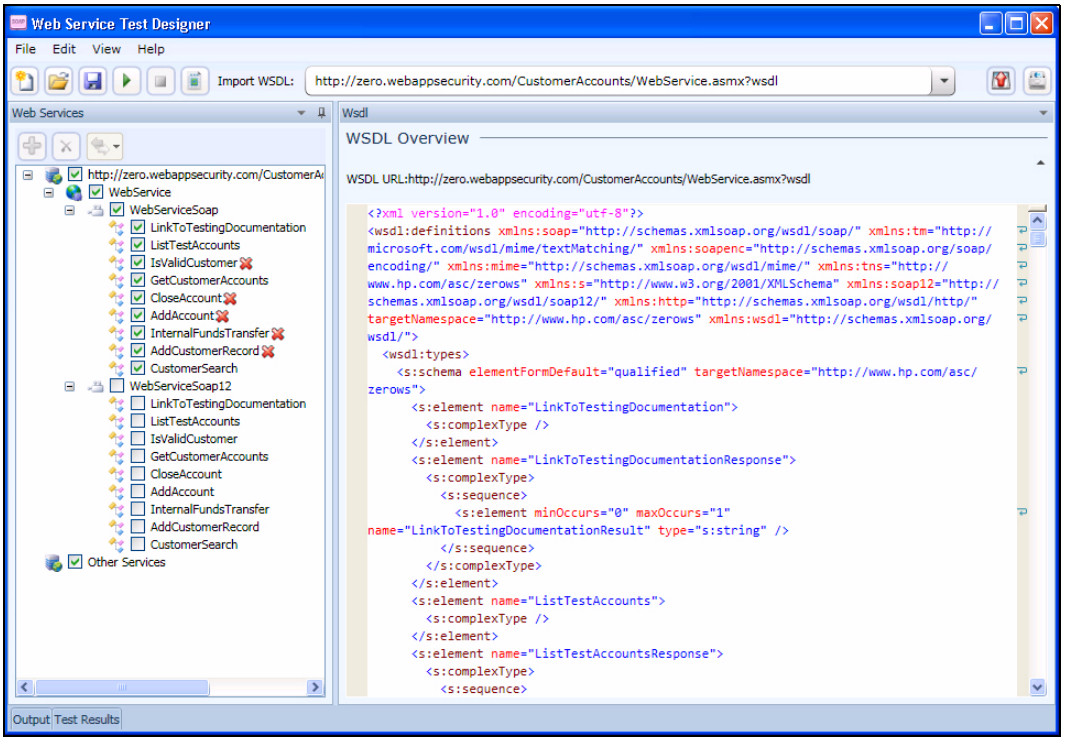
Testing Your Design

You can, at any time, test the configuration of any or all operations.

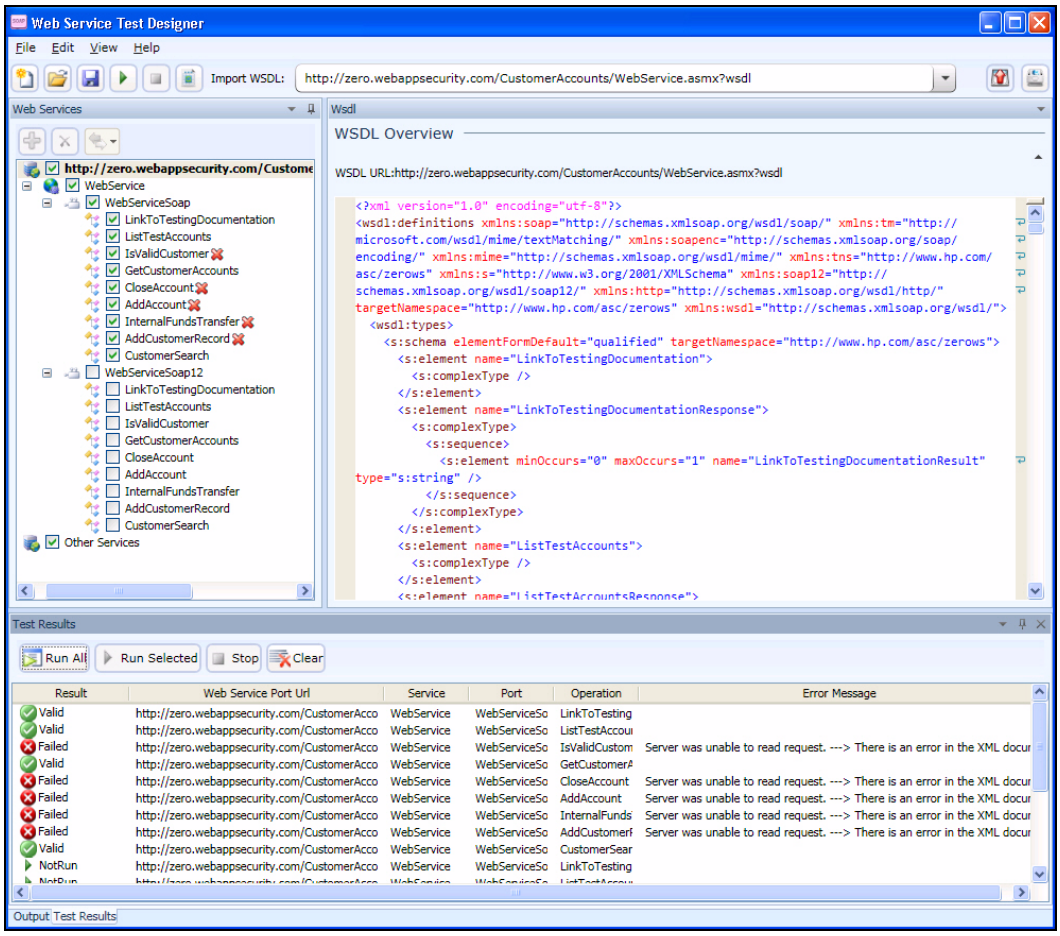
After importing the WSDL, click **Run All Tests**.



The designer attempts to submit all selected operations and displays the results.



To open the special Test Results pane, click **Test Results** on the Status bar.



The Test Results pane displays the following information:

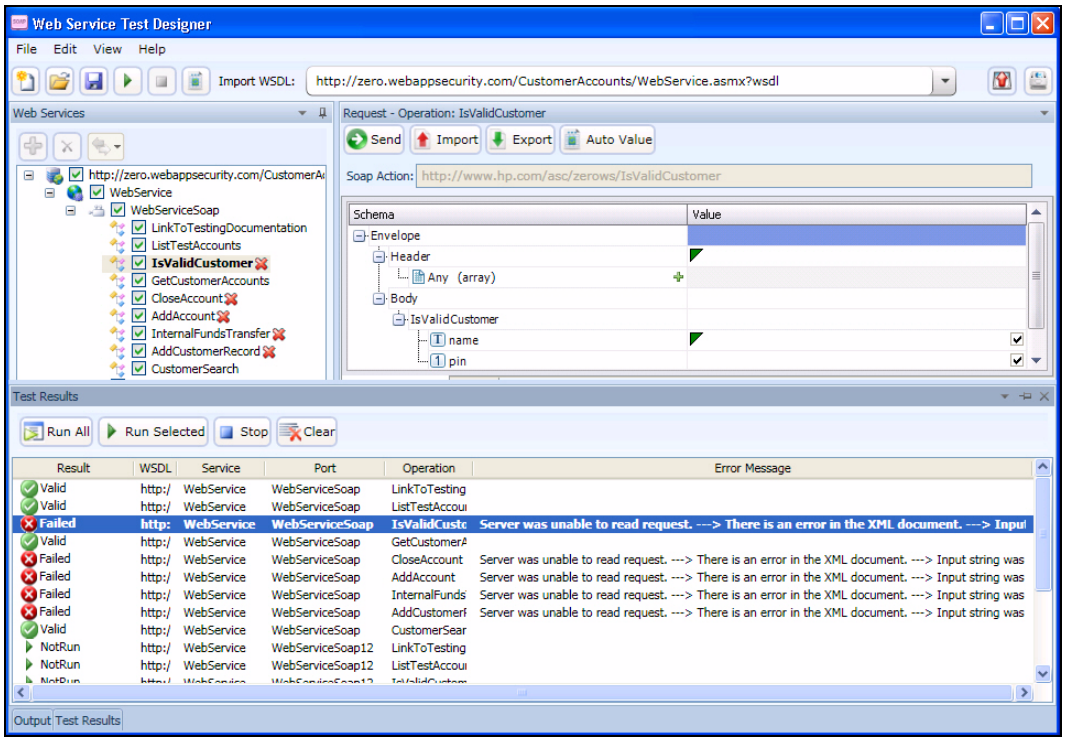
- Result – The test outcome. Possible values are:
  - Valid: The operation succeeded without a server error or SOAP fault.
  - Not Run: The operation was not submitted because it was not selected (no check mark) or the Stop button was pressed before the operation was submitted.
  - Pending: The Run button has been pressed but the operation has not yet been submitted.
  - Failed: The request was unsuccessful, the server returned an error message, or a SOAP fault was received.
- WSDL – The WSDL associated with the item
- Service – The service associated with the item
- Port – The port associated with the item
- Operation – The operation the item represents
- Error Message – Explanation for failure

The Test Results toolbar contains the following buttons:

- Run All – The designer submits the service request for each checked operation.
- Run Selected – The designer submits the service request for operations selected in the Test Results pane.

- Stop – cancels the sending of service request.
- Clear – Removes all items from the Test Results pane.

If you double-click an item in the Test Results pane, the designer highlights the related operation in the Schema Fields pane, where you can enter values for each parameter.



Web Service Test Designer Settings

The Web Services Designer has two categories of settings:

- Network Proxy
- Network Authentication

To access settings, click **Edit** → **Settings**.

Network Proxy

1 Select a profile from the Proxy Profile list:

- **Direct**: Do not use a proxy server.
- **Auto Detect**: Use the Web Proxy Autodiscovery Protocol (WPAD) to locate a proxy autoconfig file and use this to configure the browser's web proxy settings.
- **Use Internet Explorer**: Import your proxy server information from Internet Explorer.
- **Use PAC File**: Load proxy settings from a Proxy Automatic Configuration (PAC) file. Then specify the file location in the URL box.
- **Use Explicit Proxy Setting**s: Access the Internet through a proxy server using information you provide in the Explicitly Configure Proxy section.
- **Use Mozilla Firefox**: Import proxy server information from Firefox.

Note: Electing to use browser proxy settings does not guarantee that you will access the Internet through a proxy server. If the Firefox browser connection settings are configured for “No proxy,” or if the Internet Explorer setting “Use a proxy server for your LAN” is not selected, then a proxy will not be used.

2 If you selected **Use PAC File**, enter the location of the PAC file in the **URL** box.

3 If you selected **Use Explicit Proxy Settings**, provide the following information:

   a In the **Server** box, type the URL or IP address of your proxy server, followed (in the **Port** box) by the port number (for example, 8080).

   b From the **Type** list, select a protocol for handling TCP traffic through a proxy server: SOCKS4, SOCKS5, or standard.

   c If authentication is required, select a type from the **Authentication** list:

   d If your proxy server requires authentication, enter the qualifying user name and password.

   e If you do not need to use a proxy server to access certain IP addresses (such as internal testing sites), enter the addresses or URLs in the **Bypass Proxy For** box. Use commas to separate entries.

4 Click **Save**.

Network Authentication

If server authentication is not required, select **None** from the **Method** list. Otherwise, select an authentication method and enter your network credentials.