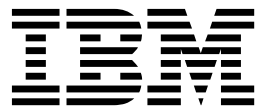


IBM Security AppScan Standard
Version 9.0.3.10

User Guide



Contents

Chapter 1. Introduction 1

Product overview	1
Supported technologies	1
What's new	2
Contact and support information	3

Chapter 2. Installing 5

System requirements	5
Flash Player upgrade	7
Flash Player configuration	7
Install	8
Silent install	8
Uninstall	9
License	9
Load a node-locked license	10
Load a floating or token license	11
Updates	11
Temp file location	12

Chapter 3. Getting started 13

How an automatic scan works	13
Web applications vs. web services	14
Basic workflow	15
Workflow description	16
Tour of the main window	17
View selector	17
Application Tree	18
Result List	20
Detail Pane	21
Scan panels	21
Status bar	21
Tutorial	22
Step 1: Configuring the scan	22
Step 2: Running the scan	23
Step 3: Reviewing Scan Results	24
Step 4: Communicating results	25
Sample scans	25

Chapter 4. Configuring 27

Scan configuration wizard	27
Launching the Scan Configuration Wizard	28
Wizard for exploring with AppScan	29
Wizard for exploring with an external device or client	33
Wizard for exploring with GSC	37
Scan configuration dialog box	39
URL and Servers view	40
Login Management view	44
Environment Definition view	56
Exclude Paths and Files view	57
Explore Options view	62
Parameters and Cookies view	67
Automatic Form Fill view	80
Error Pages view	82
Multi-Step Operations view	84

Content-Based Results view	90
Glass Box view	92
Communication and Proxy view	93
HTTP Authentication view	94
Test Policy view	95
Test Options view	100
Privilege Escalation view	102
Malware view	103
Scan Expert view	104
Advanced Configuration view	106
Scan file structure	121
Scan templates	121
Predefined templates	121
User-Defined Scan Templates	123
Loading scan templates	123
Editing Scan Templates	123
Changing the configuration during a scan	124

Chapter 5. Manual Exploring 125

Using AppScan	125
Recording a Manual Explore	126
Exporting Manual Explore data	128
Importing Manual Explore data	129
Using AppScan as a proxy server	130
Using AppScan as recording proxy	131
Exploring with the External Traffic Recorder	131
External Login Recorder	133
External Traffic Recorder	133
Using GSC	134
Exploring with GSC	135
Scanning a site that includes a SOAP web service as <i>part</i> of the site	136

Chapter 6. Scanning 137

Starting scans	137
Starting scans from the Scan Configuration Wizard	137
Starting scans from the Scan menu or the toolbar	137
Starting scans from the Welcome dialog box	138
Starting scans from the New Scan dialog box	139
Scan progress	139
Pausing and continuing scans	140
Scans stopped due to connectivity issues	140
Scans stopped due to application issues	140
Saving and loading scans	141
Saving scans	141
Automatic scan save	141
Loading saved scans	142
Importing a legacy scan template	142
Automatic scan	142
Automatic multiphase scanning	143
Scan Expert	143
Scan Expert recommendations	144
Glass box scanning	144

Overview.	144
For Java platforms.	145
For .NET platforms	164
Partial scans.	169
Scan Multi-Step Operations Only.	170
Changing the configuration during a scan.	170
Exporting scan results	170
Generating scan result DB and XML files	171
Firebird database structure	171

Chapter 7. Results: Application Data 177

Application Data: Application Tree	177
Application Data: Result List	177
Requests	178
Parameters	178
Cookies	179
Failed Requests.	180
Filtered URLs	180
User Interaction Needed.	181
Comments	182
JavaScript	182
Application Data: Detail Pane	183

Chapter 8. Results: Security Issues 185

Security Issues: Application Tree	185
Exclude URL from scan	185
Security Issues: Result List	186
Severity levels	186
Issue state: Open or Noise	187
Resending tests.	188
Right-click menu	189
Filtering Security Issues in Result List	189
Sorting the Result List	190
Security Issues: Detail Pane.	190
Issue Information tab.	190
Advisory tab	194
Fix Recommendations tab	196
Request/Response tab	197
Report false positive test results	199
Manual tests.	200
Non-vulnerable variants.	203
Saving all non-vulnerable variants	203
Defining variants as non-vulnerable.	203
Non-Vulnerable Variants List	203
Deleting variants	204

Chapter 9. Results: Remediation

Tasks 205

Remediation Tasks: Application Tree.	205
Remediation Tasks: Result List	205
Searching Remediation Tasks in Result List	206
Sorting Remediation Tasks	207
Manipulating priority levels	207
Deleting Remediation Tasks from the Result List	207
Remediation Tasks: Detail Pane	208

Chapter 10. Reports 209

Report overview	209
Configuring report layout	209
Viewing and saving reports	210

Creating partial reports	210
Earlier versions of report templates	211
Security reports.	211
Limiting the size of Security reports.	213
Industry Standard and Compliance reports	214
Industry Standard reports	214
Regulatory Compliance reports	216
User-defined reports	219
Delta Analysis reports	226
Template-based reports	228
Creating a custom report template	229
Importing a custom template	233

Chapter 11. Tools. 235

Options dialog box	235
Scan Options tab	235
Preferences tab	236
Recording Proxy tab	237
General tab	240
Advanced tab	240
Web Services Configuration wizard	241
Description files	241
Domains	242
Login Management	242
Sequences	243
Parameters	243
Complete.	244
Scan Scheduler	244
Schedule a new scan	244
Edit scheduled scan configuration	245
Delete a scheduled scan	245
Schedule a Test stage only	245
Schedule a scan in installments	246
Scheduled task command line parameters	247
User-Defined Tests	248
User-Defined Test wizard	248
Test Type.	249
Filter	249
Modification.	250
Validation	251
Advisory	251
Completing the wizard	251
PowerTools	251
Authentication Tester.	251
Connection Test	258
Encode/Decode	260
Expression Test.	261
HTTP Request Editor.	262
Generic Service Client (GSC)	266
Customizing the Tools menu	267
Adjust the order of the PowerTools	267
Add programs to the Tools menu.	267
Extensions	268
Extension Manager	268
Pyscan	269
Explore Optimization module	270
Logs	274
Scan Log	274
AppScan Log	275
Update Log	275
Traffic Log	276

Searching Results	276
-----------------------------	-----

Chapter 12. Integrations 277

AppScan Enterprise	277
Importing AppScan Enterprise license permissions	277
Publishing to AppScan Enterprise	278
Creating a job in AppScan Enterprise	278
Creating a scan template in AppScan Enterprise	280
Automation Frameworks	280
Creating the batch command	281
Application Security on Cloud	282
Uploading to Application Security on Cloud	282

Chapter 13. Best practices and FAQ 285

Workflow for advanced users	285
Initial Configuration	287
Initial Automatic Explore	288
Improve site coverage manually	288
Evaluate Explore results	290
Additional configuration	292
Sites that use parameter-based navigation	293
The challenge of parameter-based navigation sites	293
Scanning live production environments.	294
Flash content	296
Frequently Asked Questions	297

Chapter 14. Troubleshooting 301

Troubleshooting features.	301
License troubleshooting	301
Insufficient disk space	302
Digital signature troubleshooting	302
Importing a legacy scan template.	302
Reporting false positive results	303
Troubleshooting the Report False Positive feature	303
Extended Support Mode.	303
Changing the default browser	304
Login troubleshooting	305
Action-based login troubleshooting	307
Request-based login troubleshooting.	307

Advanced login troubleshooting workflow	308
Long or never-ending Explore stage	310
Flash movie troubleshooting	310
Some Flash movies are not scanned	311
Restore Adobe Flash Player settings	313
Multi-step operation troubleshooting	313
Replacing unsigned extensions	314
Scan Log messages	314
AppScan Log messages	323
Flash Log messages	329
Glass box troubleshooting	330

Chapter 15. CLI 331

Command structure	331
Commands	331
Exec command	331
Report command	334
Delta analysis report command	336
Other commands	336
Exit Status codes	337
Launching AppScan from the command line	337

Chapter 16. Menus, toolbars and keyboard shortcuts 339

File menu	339
Edit menu	340
View menu	340
Scan menu	341
Tools menu	342
Help menu	343
Main toolbar	343
Browser toolbar	344
Keyboard shortcuts	345
Accessibility controls	345

Chapter 17. Glossary 349

Chapter 18. Notices. 361

Trademarks	363
Terms and conditions for product documentation	363
IBM Online Privacy Statement.	363

Chapter 1. Introduction

A product overview, summary of what's new in this version and contact information.

Product overview

IBM Security AppScan® Standard is a security vulnerability testing tool for web applications and web services. It features the most advanced testing methods to help protect your site from the threat of cyber-attack, together with a full range of application data output options.

IBM Security AppScan Standard employs three distinct testing techniques that complement and enhance each other:

Dynamic Analysis ("black-box scanning")

This is the primary method, testing and evaluating application responses during run-time.

Static Analysis ("white-box scanning")

This is a unique technology that analyzes JavaScript code in the context of the full web page.

Interactive Analysis ("glass box scanning")

The dynamic test engine can interact with a dedicated glass-box agent which resides on the web-server itself, enabling AppScan to identify more issues, and with greater accuracy, than by conventional dynamic testing alone

AppScan's advanced capabilities include:

- General and regulatory compliance reporting, with over 40 different templates available out-of-the-box
- Customization and extensibility through the AppScan eXtension Framework, or by direct integration into existing systems using the AppScan SDK
- Link categorization capabilities that go beyond application security to identify risks posed to users from links to malicious or other unwanted sites

AppScan Standard helps you decrease the risk of web application attacks and data breaches both before site deployment and for ongoing risk assessment in production.

Supported technologies

Helps you understand which technologies used by your site might affect AppScan's ability to scan it.

Some technologies used by your site might affect AppScan's ability to scan it, while others do not affect the scan at all.

- AppScan is a "Black-Box" (DAST) tool, and scans your site using the same mechanisms as a browser. Therefore, in general, server-side technologies that are transparent to a browser are also transparent to AppScan, and do not affect the scan.
- Client-side technologies such as JavaScript and the HTTP protocol itself, do affect AppScan. Unlike a browser, AppScan needs to understand these technologies at a level that allows automatic crawling, session maintenance, and of course testing. In these cases you need to configure AppScan to scan correctly.

An AppScan scan consists of two main stages: Explore and Test. For each stage, the table below offers guidelines for understanding which server-side and client-side technologies might affect the scan, and in which cases configuration is needed.

	Server-side technologies	Client-side technologies
Explore stage	Any server-side technology that does not affect the client – such as the specific database used - does not affect the scan in any way. Many mechanisms that <i>do</i> affect the client (like session management) will not limit the scan as long as AppScan is configured correctly. For example, web servers and application servers affect how session IDs are managed, and AppScan must be able to track these IDs. Many common session IDs are predefined or can be automatically detected by AppScan and do not require additional configuration. However, additional configuration may still be required for some custom mechanisms. AppScan specifically supports WebSphere Portal custom URLs. WSP encodes the URLs in a way that makes it difficult to track them as they appear. AppScan decodes the URLs so they can be understood and tuned. Glass box scanning is supported for Java and .NET only.	The two main client-side technologies used today are HTML5 and JavaScript, and both affect the Explore stage of the scan: AppScan supports HTML in the Explore stage. This means links can be extracted, forms can be understood and filled, etc. AppScan supports (executes) plain JavaScript. Several major frameworks are specifically supported, including JQuery, AngularJS, and PrototypeJS. Many other JS frameworks though not specifically supported, do not limit the scan in any way. If the automatic Explore stage misses pages due to a specific technology, the pages can be added to the scan by exploring the site manually after the automatic Explore stage, and before the Test stage.
Test stage	AppScan is designed to test the application and not its supporting technologies, therefore they do not affect testing. To consider databases again: AppScan's suite of SQL Injection tests are independent of the database used. It also offers specific tests for 3rd Party testing (Common Vulnerabilities testing).	Client-side testing is performed only on JavaScript code. Currently only plain JS vulnerabilities are detected. JS Frameworks are not supported, and therefore JS code that uses a framework may not be properly analyzed. HTML5 is fully supported.

What's new

This section describes new product features and enhancements in this fix pack.

A complete list of fixes can be found at: <http://www.ibm.com/support/docview.wss?uid=swg27021374>

New in IBM Security AppScan Standard 9.0.3.10

Web Services Configuration wizard

The new Web Services Configuration Wizard helps you configure a scan based on the service's Open API definition files (JSON and/or YAML). The wizard helps *create* the Explore stage traffic (rather than record it), and the configuration is then used to run an automatic scan.

Contact and support information

AppScan contact information for technical support, to report false positive test results, and for technical, sales and general information.

Item	Details
Documentation	The AppScan Standard Publications Library links to all online user documentation, including: • PDF version of this User Guide • Readme file, containing any last minute information that could not be included in this Help • Fix List, detailing APARs fixed by version • System requirements • Known issues of general interest in the current version (updated as issues are discovered and as they are resolved in fix packs) • AppScan Standard download instructions http://www.ibm.com/support/docview.wss?uid=swg27024868
AppScan Standard Support Portal	http://www.ibm.com/support/entry/portal/product/software/security_systems/ibm_security_appscan_standard
AppScan Standard Forum	https://developer.ibm.com/answers/topics/appscan-standard/
To open a service request	https://www.ibm.com/support/servicerequest/Home.action
To report "false positive" results	http://www.ibm.com/support/docview.wss?uid=swg21295428 For more details see: "Report false positive test results" on page 199
AppScan eXtensions framework	http://www.ibm.com/developerworks/rational/downloads/08/appscan_ext_framework/ For more details see: "Extensions" on page 268
Support resources	http://www.ibm.com/support/docview.wss?uid=swg21672099
Sales and general information	http://www.ibm.com/software/rational/offerings/testing/webapplicationsecurity/

When calling or submitting a problem to AppScan Support about a particular service request, have the following information ready:

- The operation you performed, and error message you received
- Background information to help us understand the issue
- Version of AppScan Standard you are using
- Impact of the issue on your organization, schedule, deadlines
- Upload logs, data, and screen captures to the ticket

Chapter 2. Installing

Installation and license procedures.

System requirements

A summary of the minimum hardware and software required *for the machine that runs AppScan Standard*.

Important: A more complete list, which may include updates added after the product was released, can be found online at: <http://www.ibm.com/support/docview.wss?uid=swg27024155>

For a discussion of technologies *used by your site* that might affect AppScan's ability to scan it, see "Supported technologies" on page 1

Hardware requirements

Hardware	Minimum Requirement
Processor	Core 2 Duo 2 GHz (or equivalent)
Memory	4 GB RAM
Disk Space	30 GB
Network	1 NIC 100 Mbps for network communication with configured TCP/IP

Operating system and software requirements

Software	Details
Operating System	Supported operating systems: • Microsoft Windows Server 2016: Standard and Datacenter • Microsoft Windows Server 2012: Essentials, Standard and Datacenter • Microsoft Windows Server 2012 R2: Essentials, Standard and Datacenter • Microsoft Windows Server 2008 R2: Standard and Enterprise, with or without SP1 • Microsoft Windows 10: Pro and Enterprise • Microsoft Windows 8.1: Pro and Enterprise • Microsoft Windows 8: Standard, Pro and Enterprise • Microsoft Windows 7: Enterprise, Professional and Ultimate, with or without SP1 Note: Both 32-bit and 64-bit editions are supported, but 64-bit is preferred. All fix packs are supported.
Browser	Microsoft Internet Explorer Version 11 Recommended: Internet Explorer Version 11.0.9600.18537, Update Versions 11.0.38 KB3203621
Other	Microsoft .NET Framework 4.6.2 If using floating or token licenses: Rational® License Key Server 8.1.1, 8.1.2, 8.1.3, 8.1.4, 8.1.5 (Optional) Adobe Flash Player for Internet Explorer is required for Flash execution (and for viewing instructional videos in some of the advisories). Versions 9.0.124.0 up to 14.0.0.125 are supported. Earlier versions are not supported, and some versions may require configuration. For details, see "Flash content" on page 296. (Optional) Microsoft Word 2007, 2010, 2013 for custom report templates.

Important: Customers without a local license on their machine require a network connection to their license key server when using AppScan.

Important: A personal firewall running on the same computer as AppScan can block communication and result in inaccurate findings and reduced performance. For best results do not run a personal firewall on the computer that runs AppScan.

Glass box server requirements

The glass box scanning feature requires a glass box agent to be installed on the application server. For more details, see “Installing the glass box agent” on page 145.

Java platforms: On Java platforms the following server platforms and technologies are supported.

Software	Details
JRE	Versions 6 and 7 are supported. JRE 8 is <i>not</i> supported.
Operating System	Supported Microsoft Windows systems (both 32-bit and 64-bit editions): • Microsoft Windows Server 2012 • Microsoft Windows Server 2012 R2 • Microsoft Windows Server 2008 R2 Supported Linux systems: • Linux RHEL 5, 6, 6.1, 6.2, 6.3, 6.4 Supported UNIX systems: • UNIX AIX® 6.1, 7.1 • UNIX Solaris (SPARC) 10, 11
Java™ EE container	JBoss AS 6, 7; JBoss EAP 6.1; Tomcat 6.0, 7.0; WebLogic 10, 11, 12; WebSphere 7.0, 8.0, 8.5, 8.5.5

.NET platforms: On .NET platforms the following systems and technologies are supported:

Item	Details
Operating System	Supported operating systems (both 32-bit and 64-bit editions): • Microsoft Windows Server 2012 • Microsoft Windows Server 2012 R2 • Microsoft Windows Server 2008 R2
Other	Microsoft IIS 7.0 or later Microsoft .NET Framework 4.0 or 4.5 must be installed, and IIS must be configured at the root level to work with this version of ASP.net

Note: User must have administrator privileges when running the application on the server.

Note: The agent should be installed *after* the application you want to test is successfully installed on the server.

Cookie usage

Although AppScan tracks cookies set by the application it is scanning, it does not set cookies of its own for authentication or any other purpose.

Supported languages

The AppScan user interface can run in the following languages: Chinese (Simplified), Chinese (Traditional), English (United States), French, German, Italian*, Japanese, Korean*, Portuguese (Brazil), Russian*, Spanish (Spain). To change the user interface language go to **Tools > Options > General tab**

Note: For the Italian and Korean user interfaces, all documentation is in English. For the Russian interface the Help is translated, but all other documentation is in English.

Related concepts:

“Supported technologies” on page 1

Helps you understand which technologies used by your site might affect AppScan’s ability to scan it.

Flash Player upgrade

About this task

In order for AppScan to be able to execute Adobe Flash content during scanning, you must have a supported version of the Adobe Flash Player for Internet Explorer installed. Versions 9.0.124.0 up to 14.0.0.125 are supported. Earlier versions are not supported, and some supported versions may require configuration. You must have a supported version in order to execute Flash files during scanning.

Note: Until you upgrade Flash Execution will not run even though the Flash Execution check box (Scan Configuration > Explore Options) is selected.

Procedure

1. Close AppScan and any Microsoft Internet Explorer windows.
2. To download and install the latest Flash Player, go to <http://get.adobe.com/flashplayer/>

Flash Player configuration

About this task

In order for AppScan to execute Adobe Flash content during scanning, you must have a supported version of Adobe Flash Player for Internet Explorer installed. In some cases Flash Player Version 10.1 or higher may require configuration to work with AppScan. If you get a message that your Flash Player requires configuration, follow the procedure below.

Note: Without configuration Flash Execution will not run even though the Flash Execution check box (Scan Configuration > Explore Options) is selected.

Note: This procedure requires Administrator permissions.

Procedure

1. Close AppScan.
2. *With Administrator permissions*, open the folder containing the Flash installation files.
 - For 32-bit systems the path is usually:
C:\WINDOWS\System32\Macromed\Flash
 - For 64-bit systems the path is usually:
C:\WINDOWS\SysWow64\Macromed\Flash
3. Look for a file named `mms.cfg` in the Flash folder. If there is no such file, create an empty TXT file with this name.
4. Open `mms.cfg` with a text editor such as Microsoft Notepad, and search for the entry `FullFramerateWhenInvisible`
 - If it exists, set its value to 1

- If it does not exist, add the following line, *as a separate line*, after any existing content in the file:
FullFramerateWhenInvisible = 1

5. Save.

The Flash Player is now configured for AppScan Flash Execution.

Install

The installation wizard guides you through the fast and simple process.

Procedure

1. If you have on your machine a version of AppScan Standard later than 8.5, you must uninstall it before installing this version.
2. Close any Microsoft Office applications that are open.
3. Start AppScan setup.

The InstallShield Wizard starts, and checks that your workstation meets the minimum installation requirements. Then the AppScan installation wizard welcome screen appears.

4. Follow the wizard instructions to complete AppScan installation.

Note: You will be asked to install or download GSC (Generic Service Client). This is needed for exploring web services in order to configure a web services scan, but is not needed if you will not be scanning web services.

Silent install

Instructions for unattended installation, using the command line.

You can install AppScan "silently", using the command line and the following parameters:

AppScan_Setup.exe /l"LanguageCode" /s /v"/qn INSTALLDIR=\"InstallPath\""

Parameter	Function
/l	Language code. Options are: • English: 1033 • Chinese (Traditional): 1028 • Chinese (Simplified): 2052 • French: 1036 • German: 1031 • Italian: 1040 • Japanese: 1041 • Korean: 1042 • Portuguese: 1033 • Spanish: 1034
/s	Activates "Silent Mode" (otherwise the regular installation will be launched). Note: Must be used in conjunction with /v"/qn" (see next row)

Parameter	Function
/v	Sets additional MSI properties such as UI mode and the path where AppScan will be installed. UI Mode: For "Silent Mode", include /qn as a parameter (enclosed in quotes). Path: If you do not define an install path, installation uses the default path: ...Program Files\IBM\AppScan Standard\ To define a different install path, add INSTALLDIR="InstallPath\" as a parameter (enclosed in quotes). The path may include spaces. Example: /v"/qn INSTALLDIR="D:\Program Files\AppScan\""

Examples:

- To silently install an English version of AppScan in the default directory enter:
AppScan_Setup.exe /s /v"/qn"
- To silently install Japanese versions of AppScan in the default directory enter:
AppScan_Setup.exe /l"1041" /s /v"/qn"
- To silently install a Korean version of AppScan in D:\Program Files\AppScan\ enter:
AppScan_Setup.exe /l"1042" /s /v"/qn INSTALLDIR="D:\Program Files\AppScan\""

Uninstall

Instructions for uninstalling AppScan from your computer.

About this task

You can run the uninstall wizard from the Windows Start menu (described below), or using the Add/Remove Programs facility in the Windows Control Panel. The exact procedures may vary depending on which version of Windows is installed.

Procedure

1. Go to Start > All Programs > IBM Security AppScan Standard
2. Select Uninstall IBM Security AppScan Standard and follow the instructions.

Note: The uninstall wizard does not delete scan files or reports created with AppScan. If you want to delete these files, do so manually.

License

This section describes license installation and management.

The AppScan installation includes a default license that allows you to scan IBM's custom designed AppScan testing website (demo.testfire.net), but no other sites. In order to scan your own site you must install a valid license supplied by IBM®. Until this is done AppScan will load and save scans and scan templates, but it will not run new scans on your site.

IBM Security AppScan Standard licenses

There are three types of license:

"Node-locked" licenses

These are installed locally onto the machine on which AppScan runs. Each license is assigned to a single machine.

"Floating" licenses

These are installed onto the IBM Rational License Key Server (which can be the same as the machine on which AppScan runs). Any server on which AppScan is used must have a network connection with the license key server. Each time a user opens AppScan a licence is checked out, and when AppScan is closed the license is checked back in.

"Token" licenses

These are installed onto the IBM Rational License Key Server (which can be the same as the machine on which AppScan runs). Any server on which AppScan is used must have a network connection with the license key server. Each time a user opens AppScan the required number of tokens are checked out, and when AppScan is closed they are checked back in.

License status

To view and edit AppScan licenses, click **Help > License**. There are three options:

Open AppScan Standard License Manager	Opens the list of currently loaded licenses, and lets you: • Add or remove node-locked licenses • Set the license key server(s) for floating or token licenses
Add AppScan Enterprise License	If your organization has an AppScan Enterprise license that allows scanning additional sites to those allowed by your local AppScan Standard license, you can import these permissions to use on your local machine in addition to your existing license. Note: This option is available only when a full AppScan Standard license (not a demo license) is loaded. See "Importing AppScan Enterprise license permissions" on page 277.
View License Agreement	Click here to see the license agreement.

Note: You can refresh the license information displayed in the dialog box by clicking 

Note: If a floating or token license has been verified, but the license key server later becomes unavailable, AppScan can run in "Disconnected Mode" for up to three days. During this time you can scan your application as usual.

See also:

"License troubleshooting" on page 301

"Importing AppScan Enterprise license permissions" on page 277

Load a node-locked license

How to load a node-locked license.

About this task

"Node-locked" licenses are installed on the machine on which AppScan Standard runs, not a separate server.

Procedure

1. Download your license file from Rational License Key Center, and save it on your machine.
2. In AppScan Standard, click **Help > License > Open AppScan Standard License Manager**.
3. Click **License Configuration**.
4. Above the *upper* pane (Node-Locked License Files), click .
5. Browse to the license file.
6. (Optional:) If you add more than one file, you can use the **Up/Down Arrow** buttons to promote/demote licenses in the list.
7. In the AppScan license dialog box, click  to load your license.

Note: The license name must contain only ASCII characters. If necessary, rename the file in order to load it.

Load a floating or token license

How to load a floating or token license for use with AppScan Standard Edition.

About this task

In order to install a floating or token license you must first have a license key server with IBM Rational License Key Server (version 8.1.2 or later) installed on it. The license key server can be a different machine, or the same machine that AppScan is installed on. The procedure below describes setting up the server and then loading a floating license.

Procedure

1. Download IBM Rational License Key Server (version 8.1.2 or later) from Passport Advantage.
2. Install the License Key Server. This can be on the same computer as AppScan, or on your central network license server.
3. Download your license files from Rational License Key Center, and save them to the machine on which you installed Rational License Key Server.
4. Click **Start > Programs > IBM Rational > License Key Administrator (version)**, and use the Import a Rational license file wizard to import the license file to the license key server.
5. In AppScan Standard, click **Help > License > Open AppScan Standard License Manager**.
6. Click **License Configuration**.
7. Above the *lower* pane (License servers), click .
8. In the dialog box that opens, enter the Host and Port for the license.
9. (Optional:) If you add more than one license server, you can use the **Up/Down Arrow** buttons to promote/demote servers in the list.
10. In the AppScan license dialog box, click  to load your license.

Updates

Keeping your installation up-to-date.

About this task

Subscription updates include new types of web application exploitation techniques and bug fixes. It is recommended that you install these files as soon as you receive notification of their availability.

AppScan periodically checks for updates on the IBM website and notifies you when new updates are available. You may also initiate an update search.

After AppScan has detected new available updates you are given the opportunity to download and install the new update files on your machine.

Procedure

1. On the Toolbar, click .
AppScan checks for updates. If an update is available the **Install** button becomes active. (If your version of AppScan is up-to-date the button remains grayed out.)
2. To install the update, click **Install**.

What to do next

You can check update status in the “Update Log” on page 275.

Temp file location

Describes where AppScan saves its temporary files during normal operation, and how to change the location.

By default AppScan stores its temporary files in:

C:\Documents and Settings\All Users\Application Data\IBM\AppScan Standard\temp

If you need to override this default location for any reason, edit the path for the environment variable APPSCAN_TEMP as required. (Environment variables are accessed by right-clicking **My Computer**, and then selecting **Properties > Advanced > Environmental Variables**.)

Restriction: There must be no Unicode characters anywhere in the path to the new location.

Chapter 3. Getting started

This section provides a short tour of basic product features and procedures.

How an automatic scan works

This topic explains the difference between the "stages" and "phases" of a scan.

An AppScan Full Scan consists of two stages: Explore and Test. It is useful to understand the principal behind this, even though most of the scan process is in fact seamless to the user, and little user input is required until the scan is complete.

Explore stage

During the first stage, AppScan explores the site (web application or web service) by simulating a web user clicking on links and completing form fields. This is the Explore stage.

AppScan analyzes the responses to each request it sends, looking for any indication of a potential vulnerability. When AppScan receives a response that may indicate a security vulnerability, it automatically creates a test (or tests) based on the response, as well as noting the validation rules needed to determine which results constitute vulnerability, and the level of security risk involved.

Before sending the site-specific tests created, AppScan sends several malformed requests to the application to determine the manner in which it generates error responses. This information is then used to increase the precision of AppScan's automatic test validation process.

Test stage

During the second stage, AppScan sends thousands of custom test requests that it created during the Explore stage. It records and analyzes the application's response to each test using the custom validation rules. These rules both identify security problems within the application and also rank their level of security risk.

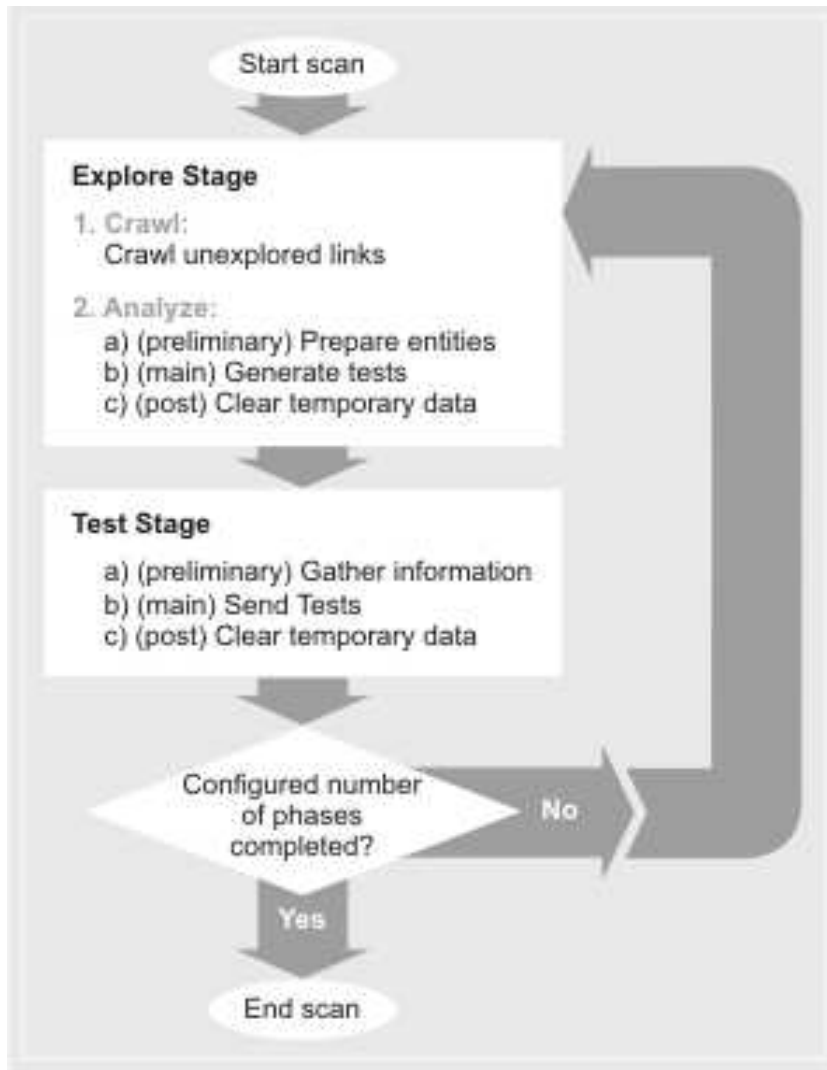
Scan phases

In practice, the Test stage frequently reveals new links within a site, and more potential security risks. Therefore, after completing the first "phase" of Explore and Test, AppScan automatically begins a second "phase" to deal with the new information. If new links are discovered during the second phase, a third phase is run, and so on.

After completing the configured number of scan phases (user configurable; default four), scanning stops and the completed results are available to the user.

Illustration of automatic scan flow

The following diagram illustrates the stages and phases of automatic scan flow. Note that this process requires no action from the user, but you may come across them referred to in the AppScan log.



Web applications vs. web services

This topic explains the different methods available for exploring sites, before AppScan tests them.

A site is scanned by first exploring it, and then, based on the data gathered, testing it. "Explore data" can be gathered using one or more different Explore methods. In all cases, once the Explore data is gathered AppScan is used to create and send tests to the site during the Test stage.

Exploring web applications (sites with a user interface)

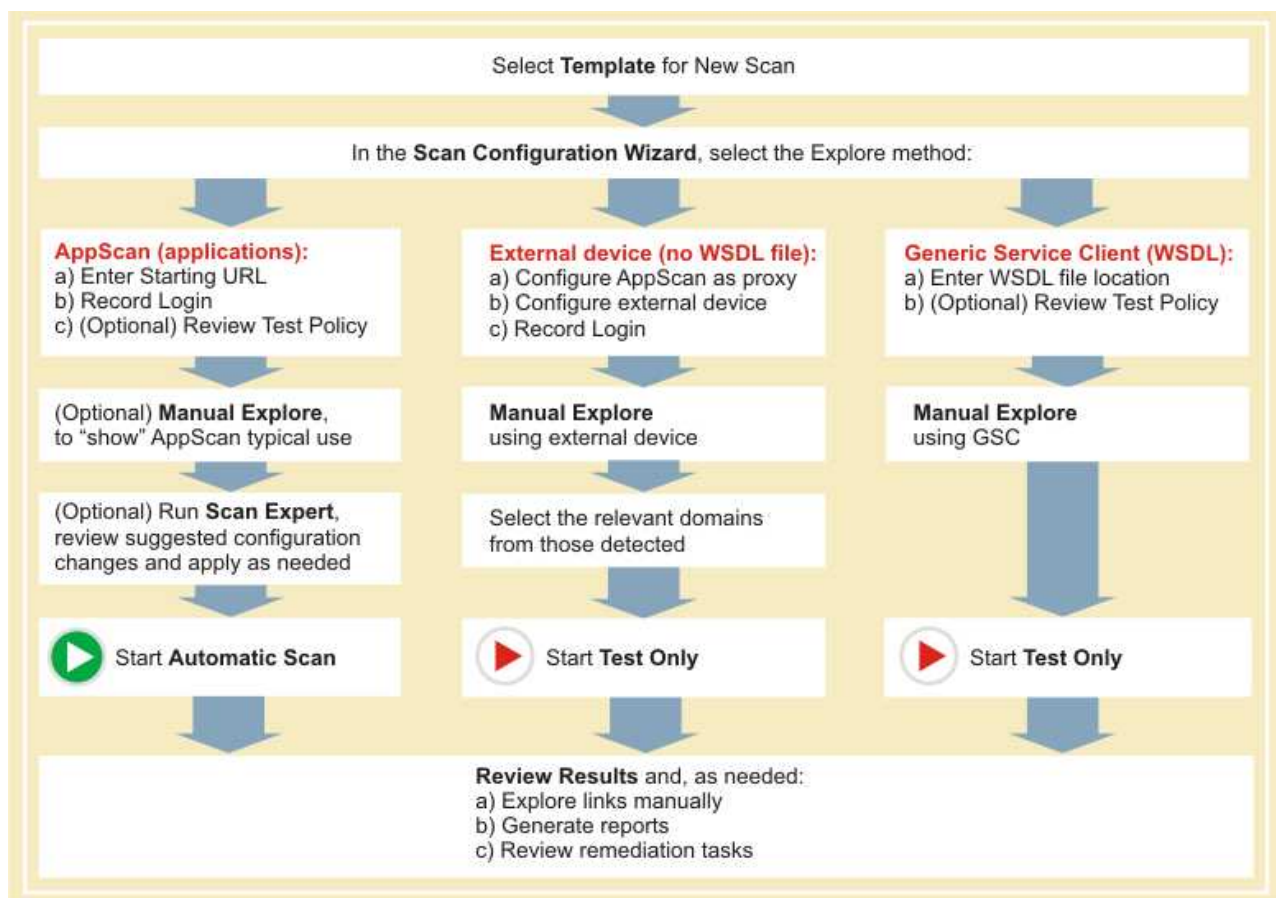
- In the case of applications (sites) without web services it is often sufficient to supply AppScan with the start URL and login authentication credentials for it to be able to test the site.
- If necessary you can manually explore the site *through AppScan*, in order to get access to areas that can only be reached through specific user input.
- For pages that can be reached only by accessing pages in a specific order, you can record a multi-step operation for AppScan to use.
- While the Configuration Wizard lets you configure and start your scan in a few steps, for complex sites the Configuration Dialog Box lets you fine-tune and customize many more settings.

Exploring web services

- You can set up AppScan as a recording proxy for the device (such as a mobile phone or simulator) you use to explore the service. That way AppScan can analyze the Explore data collected, and send appropriate tests. You can also use AppScan to record traffic using external tool, such as a web services functional tester. See “Using AppScan as recording proxy” on page 131.
- If you have Open API description files (JSON or YAML) for your web service, you can use the Web Services Wizard extension to configure a scan, and the multi-step sequences needed to use the service. AppScan will then automatically scan the service.
- If you cannot use the first two methods, and have a WSDL file for your web service (such as a SOAP web service), the AppScan installation optionally includes a separate tool that lets users view the various methods incorporated in the web service, manipulate input data, and examine feedback from the service. You first need to give AppScan the URL of the service. The integrated “Generic Service Client” (GSC) uses the WSDL file to display the individual methods available in a tree format, and create a user-friendly GUI for sending requests to the service. You can use this interface to enter parameters and view the results. The process is “recorded” by AppScan and used to create tests for the service when AppScan scans the site. GSC can also be used as client for REST requests, without parsing a WSDL file, as a simple HTTP client. See “Using GSC” on page 134.

Basic workflow

A diagram showing a simple AppScan workflow using the scan configuration wizard.



For more details of the basic workflow, see “Workflow description” on page 16

Users with experience in the field of web security, see “Workflow for advanced users” on page 285

Workflow description

AppScan provides a comprehensive assessment of your web application. It runs thousands of tests based on all levels of typical user techniques as well as unauthorized access and code injections.

When you run a scan on your application, the tests are sent by AppScan to your web application. The results of the tests are provided by AppScan's site-smart engine and result in expansive reports and fix recommendations, available for enhanced review and manipulation.

AppScan is an interactive tool: you decide on the configuration of the scan and determine what is to be done with the results.

The AppScan workflow includes the following stages:

1. **Select a Template:** A predefined scan configuration is a *scan template*. You can load the Regular Scan template, another predefined template, or a template that you previously saved. (You can later adjust the configuration as required for the current scan.)
2. **Application or Web Service Scan:** Scanning web services requires some manual input by the user, using GSC (Generic Service Client), to show AppScan how to *use* the service.
 - **AppScan:** If you are *not* scanning a web service, or if you want to scan parts of the application *other* than its web services, leave this default option selected.
 - **External device/client:** Select this option if you want to scan a service for which you do not have a WSDL file. You will configure AppScan as recording proxy, and send requests from your external client through AppScan.
 - **Generic Service Client:** Select this option if you want to scan a service. GSC (Generic Service Client) will later open to let you send requests to the service, and collect results, for AppScan to analyze and use to create tests.
3. **Scan Configuration:** Configure the scan, taking into account details of your site, your environment, and other requirements.
4. (Optional) **Manual Explore:** Log in to the site, and click links and fill in forms as a user would. This is a good way of "showing" AppScan how a typical user might browse the site, ensuring that important parts of the site are scanned, and providing data for filling forms.
5. (web services only) **Send requests using GSC:** Open GSC and send some valid requests to the service.
6. (Optional) **Run Scan Expert:** This is a short pre-scan of your site to evaluate the configuration. Scan Expert may suggest changes to increase the efficiency of the main scan.
7. **Scan the Application or Service:** This is the main scan, and consists of Explore and Test stages.

Explore Stage: AppScan crawls your site, visiting links as a regular user would and records the responses. It creates a hierarchy of the URLs, directories, files, and so on, that it finds on your application. This list is displayed in the Application Tree (see "Application Tree" on page 18).

The Explore stage can be done automatically, manually, or as a combination of both. You can also import an Explore Data File (see "Exporting Manual Explore data" on page 128), which consists of a previously recorded manual explore sequence. AppScan then analyzes the data it has collected from the site, and based on it, creates tests for the site. These tests are designed to reveal weaknesses both in infrastructure (such as security weaknesses in commercial, 3rd Party products or Internet systems), and the application itself.

Test Stage: During the Test stage, AppScan tests your application, based on the responses it received during the Explore stage, to reveal vulnerabilities and assess their severity.

An up-to-date list of all tests included in your current version of AppScan can be seen in the Scan Configuration dialog box (see "Test Policy view" on page 95).

You can also create user-defined tests in addition to the tests that AppScan automatically creates and runs (see "User-Defined Tests" on page 248). Your tests can supplement those generated by AppScan and can verify the results that it found.

Test results are displayed in the Result List, from where you can view and modify them. Full details of the results are displayed in the Detail Pane.

8. (Optional) **Run Malware Test:** This analyzes pages and links found on your site for malicious and otherwise unwanted content.

Note: Although a Malware Test can in principle be performed at this stage (in which case it will use the Explore stage results of the main scan), in practice a Malware Test is usually run on a *live* site, whereas a regular scan is usually run on a *test* site (because of the risk of disrupting a *live* site by scanning it).

9. **Review Results** to evaluate the security status of the site. You may also want to:
 - Explore additional links manually
 - Review Remediation Tasks
 - Print Reports
 - Adjust the scan configuration, if necessary based on your review of the results, and scan again

Note: For a simplified illustration of this workflow, see “Basic workflow” on page 15.

Tour of the main window

Describes the components of the AppScan main window, and all menus and toolbars.



You can resize the panes by clicking-and-dragging on the dotted section of any splitter bar (the bar between panes).

You can arrange the two right-hand panes vertically or horizontally, by clicking **View > Layout > Vertical/Horizontal**.

See also: Chapter 16, “Menus, toolbars and keyboard shortcuts,” on page 339.

View selector

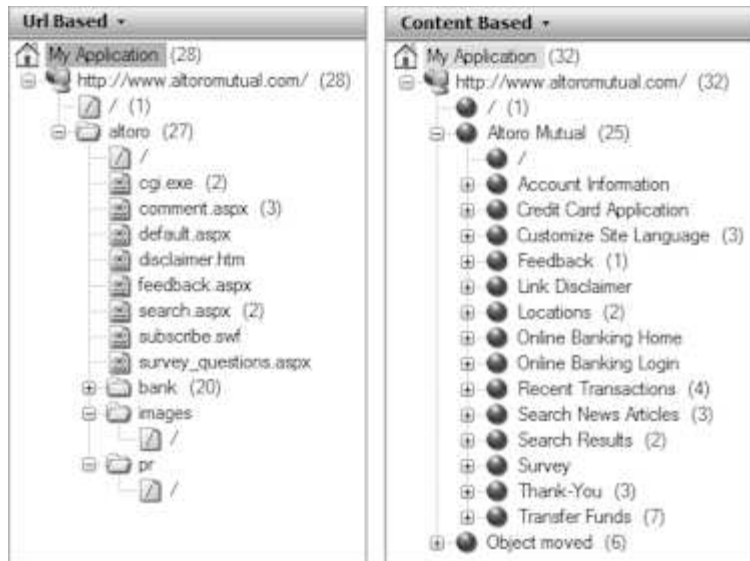
The View Selector icons on the right-hand side of the toolbar toggle between the different result views.

When you select a different view in the View Selector, the information displayed in the Application Tree, Result List and Detail Pane changes. The three parts of the screen are summarized in the table following.

	Data view	Shows script parameters, interactive URLs, visited URLs, broken links, filtered URLs, comments, JavaScripts and cookies from the Explore stage. Application Tree: Complete application tree. Result List: Select a filter from the pop-up list at the top of the Result List, to determine which information is displayed. Detail Pane: Filtered list of: script parameters, interactive URLs, visited URLs, broken links, filtered URLs, comments, JavaScripts and cookies. Unlike the other two views, Application data view is available even if AppScan has only completed the Explore stage. Use the pop-up list at the top of the Result list to filter the data. Keyboard shortcut: F2 Chapter 7, “Results: Application Data,” on page 177
	Issues view	Shows the actual issues discovered, from overview level down to individual requests/responses. This is the default view. Application Tree: Complete application tree. Counters next to each item show the number of issues found for the item. Result List: Lists issues for the selected node in the application tree, and the severity of each issue. Detail Pane: Shows advisory, fix recommendations and request/response (including all variants used) for the issue selected in the Result List. Keyboard shortcut: F3 Chapter 8, “Results: Security Issues,” on page 185
	Tasks view	Provides a To Do list of specific remediation tasks to fix the issues found by the scan. Application Tree: Complete application tree. Counters next to each item show the number of fix recommendations for that item. Result List: Lists remediation tasks for the selected node in the application tree, and the priority of each task. Detail Pane: Shows details of the remediation task selected in the Result List, and all the issues that this remediation will solve. Keyboard shortcut: F4 Chapter 9, “Results: Remediation Tasks,” on page 205

Application Tree

The Application Tree is a tree-view display of the folders, URLs, and files that AppScan found on your application.



Note: In the case of applications without a hierarchical URL structure, such as single entry-point applications (e.g. MVC), or applications where the hierarchical structure is not logical, you can create a "content-based" application tree, by defining a set of regular expressions which extract the logical path from the pages. (See "Content-Based Results view" on page 90.)

Application Tree icons

Each of the various node types in the Application Tree is indicated by its own icon.

Icon	Indicates
	My Application, root node.
	Scanned host. If you set up Additional Servers or Domains ("URL and Servers view" on page 40), or if you have a server on different ports, you may have multiple nodes of this type.
	Folder (path) found in your application.
	Slash; results of tests for the parent folder.
	File found in your application.
	A red X on any icon in the Application Tree (file icon illustrated here) indicates that the node and all its child nodes have been excluded from scans by the User. (To re-include such a node in future scans, right-click and select Include in Scan .) Note: A child node can be included even if its parent node is excluded.

Note: If a URL in the application tree contains *only* error responses, the URL is displayed with strike through formatting (with a line through it).

Application Tree counters

The Application Tree counters (the numbers in parentheses next to each node of the tree) change according to the view you select in the View Selector:

- **Security Issues:** Counters indicate the number of issues relevant to the node and all its child nodes. (The total number of issues is shown at the top of the “Result List.”)
- **Remediation Tasks:** Counters indicate the number of remediation tasks relevant to the node and all its child nodes.
- **Application Data:** No counters.

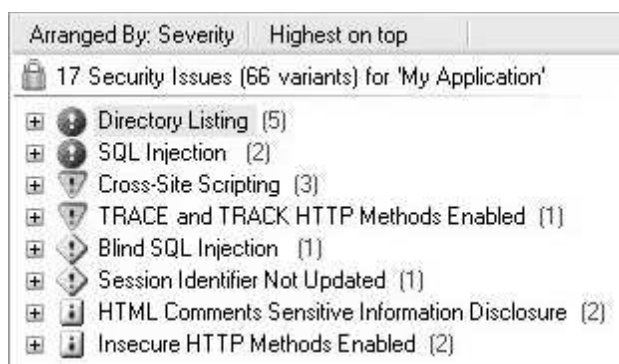
Application Tree right-click menu

If you right-click an item (other than My Application) in the Application Tree, the context-sensitive menu offers some or all of the following options:

Menu Item	Function
Show in Browser	Opens the embedded browser to the selected URL.
Manual Explore	Starts a Manual Explore of the selected item. See “Using AppScan” on page 125 for details.
Manual Test	Opens the Manual Test dialog box. See “Manual tests” on page 200 for details
Record Multi-Step Operation	Record a multi-step operation sequence for testing parts of the site that can only be reached by clicking links in a specific order. See “Multi-Step Operations view” on page 84 for details.
Copy URL	Copies the selected URL to the clipboard. (Where there are more than one sub-nodes the URL of the first sub-node is copied.)
Report for this node	Create a report for the currently selected node in the application tree (including all its child nodes). See “Creating partial reports” on page 210 for details.
Exclude from Scan	Excludes the selected URL (or all URLs under the selected node) from scans. (Toggles with “Include in Scan”) See “Exclude URL from scan” on page 185 for details.

Result List

The results of the scan on your web application are displayed in the Result List. The total number of issues and variants appears at the top of the list.



Result List counters

The Result List counters (the numbers in parentheses next to each node) change according to the view you select in the View Selector:

- **Security Issues:** Counters indicate the number of issues relevant to the node and all its child nodes.

Note: The total number of security issues (at the top of the Result List) is a measure of the vulnerable locations in the site, and depends in part on how site is structured. If you define a content-based

structure, the total number of issues in the application tree may not be the same as it is for the URL-based application tree (for the same results). When site structure is content-based (rather than URL-based), and content-based view is configured correctly, the issue count in content-based view represents more accurately the number of "vulnerable locations" that exist in the site. The total number of *variants* (at the top of the Result List in parentheses) is independent of site structure, and does not change between content-based and URL-based views.

- **Remediation Tasks:** Counters indicate the number of remediation tasks relevant to the node and all its child nodes.
- **Application Data:** No counters.

See also: "Content-Based Results view" on page 90

Detail Pane

This pane shows details of the item selected in the Result List.

The information available in the Detail Pane depends on the selected item and the selected view:

- "Application Data: Detail Pane" on page 183
- "Security Issues: Detail Pane" on page 190
- "Remediation Tasks: Detail Pane" on page 208

Scan panels

The Progress and Notice panels

Progress panel



The Progress panel appears below the Result List during a scan. It appears when you start a scan and displays progress information while the scan continues.

Notice panel

The Notice panel appears in place of the Progress panel, if the AppScan is unable to connect with your web server or if the scan was stopped before it finished. It displays brief information (such as "Server Down", or "Scan is Incomplete"), with a link to open a dialog with details.

Status bar

The Status Bar at the bottom of the main window shows information for the scan that is currently running or loaded.

Icon	Indicates
	Visited Pages: Number of pages visited / Total number of pages to be visited The second number may increase and then decrease during the scan, as pages are discovered and then rejected as not requiring scanning. By the end of the scan the two numbers should be equal.

Icon	Indicates
	Tested Elements: Number of elements tested / Total number of elements to be tested The <i>second</i> number will increase during the Explore stage, as elements for testing are discovered. During the Test stage the <i>first</i> number will increase. By the end of the scan the two numbers should be equal.
	HTTP Requests Sent This number represents all requests sent, including in-session detection requests, server-down detection requests, login requests, multi-step operations and test requests. During the scan it is therefore an indicator that AppScan is working, but the actual number is not of any particular significance either during or after the scan.
	Security Issues Total number of security issues found, followed by the number in each category: High, Medium, Low, and Informational.

Tutorial

This simple tutorial goes through the steps of configuring a simple application scan using the Scan Configuration wizard, running the scan, and reviewing the results.

This simple tutorial uses the Scan Configuration wizard to scan the "AltoroMutual Bank" website, which has been created for demonstration purposes.

Experienced users may prefer to follow the more advanced workflow (also using the wizard) described in "Workflow for advanced users" on page 285, or to configure in greater detail using the "Scan configuration dialog box" on page 39.

- "Step 1: Configuring the scan"
- "Step 2: Running the scan" on page 23
- "Step 3: Reviewing Scan Results" on page 24
- "Step 4: Communicating results" on page 25

You may like to work through this tutorial using the "AltoroMutual Bank" website, which has been created for demonstration purposes:

URL	http://demo.testfire.net/
Username	jsmith
Password	demo1234

Note: If you are using an evaluation copy of AppScan, the AltoroMutual Bank website is the *only* site you can scan.

Note: This tutorial is only a quick view of the basic steps for running a scan. For explanations and full instructions, see Chapter 4, "Configuring," on page 27. See also "Workflow for advanced users" on page 285.

Step 1: Configuring the scan

Step one of the tutorial.

About this task

The Scan Configuration Wizard provides an easy way to configure a scan, when you do not need to change many of the default configuration settings.

Procedure

1. Launch AppScan, to open the Welcome Screen or, if AppScan is open, click **File > New** for a similar dialog box.
2. Verify that the **Launch Scan Configuration Wizard** check box is selected, then select the **Regular Scan** template.

The Scan Configuration Wizard Welcome opens.

3. Select the **web Application Scan** radio button, then click **Next**.

The URL and Servers step of the wizard (1/3) appears.

Note: If you select the web *Services* Scan option, the flow is slightly different, and when the wizard closes the Generic Service Client (GSC) opens, enabling you to input parameters that AppScan will use for the Test stage of its scan. (For details, see “Wizard for exploring with GSC” on page 37.)

4. Enter the URL of your application in the text box, then click **Next**.

The Login Management step (2/3) appears.

5. Click **Record Login**.

The AppScan browser opens, to the starting URL that you set up in the previous step. Your browsing is now being recorded by AppScan.

6. Log into your application with an authorized username and password.

7. When you have successfully logged-in, close the browser.

The “Login Sequence” (the sequence of links that achieved the logged-in state) is displayed (see “Recording a login” on page 45 for more details), and the gray key icon turns green, indicating that in-session detection is active.

This:  changes to this: 

8. Click **Next**.

The Test Policy step (3/3) appears.

9. Click **Next**.

The final step of the wizard appears. You are now ready to run the scan (see “Step 2: Running the scan”).

Note: Although it is possible to start the automatic scan at this stage, in many cases a better result will be achieved by manually exploring the application first, as a regular user would (see “Using AppScan” on page 125).

Step 2: Running the scan

Step two of the tutorial.

About this task

When configuration is complete, you can run the scan.

Procedure

1. Select **Start a full automatic scan**, then click **Finish**.

The wizard closes and Scan Expert starts to evaluate the effectiveness of the current configuration for your site. When the evaluation is complete, a checklist of suggested configuration changes appears.

Note: If there are changes that require user input, their check boxes are grayed out and deselected. To provide the required input for these changes, click the link for that change.

2. Click **Apply Recommendations**.

The selected configuration changes are applied and scan starts. The Progress Panel opens, and Application Data and Issues are updated in real-time.

During the Explore stage, AppScan crawls your application to discover its pages and content. The **Application Data** tree is updated as this happens, and eventually shows a complete tree of the site. Then, during the Test stage, AppScan runs thousands of tests on the site and reports the issues found and fix recommendations. During this part of the scan, Security Issues view is selected automatically and the Result List displays a dynamically updated list of discovered issues.

A scan can have multiple phases (a phase is a cycle of Explore followed by Test). This happens when AppScan discovers new links during the test stage, and needs to a scan, it will create new tests based on these links, and run an additional scan phase. The subsequent phase is usually shorter than the previous one, as only the new links are scanned. AppScan will add phases as long as it finds new URLs, or until it has reached the configured "scan limit". The default limit is four phases.

When the scan is complete, the Progress Panel closes, and you can review the results (see "Step 3: Reviewing Scan Results").

Step 3: Reviewing Scan Results

Step three of the tutorial.

About this task

When the scan is complete, the results are displayed in the main window, in three areas: the Application Tree, the Result List, and the Detail Pane. The type of information presented in each area depends on the View selected (default is Security Issues View).

Procedure

To access a view, click the relevant icon in the View Selector on the left side of the screen.

View	Description
	Data View provides lists of content items found in your application. This can be a useful way of verifying that the scan is configured in a way that will cover your application, <i>before</i> you begin the Test stage. • Application Tree: Shows the URL and folder nodes. • Result List: Shows application data, sorted by scan results (see "Application Data: Result List" on page 177). For example, you can choose to see a list of broken links, of JavaScript, of cookies, and so on. • Detail Pane: Shows the request sent to a page and the response that was received. To learn more about the Application Data view, see Chapter 7, "Results: Application Data," on page 177.

View	Description
	Issues View provides comprehensive data of the security issues discovered by the scan. • Application Tree: Shows the folders, URLs, and files that AppScan discovered in your application are listed. The number next to each node in the tree indicates how many issues were found. • Result List: Shows an icon per issue, indicating the severity value that is assigned to this issue (see “Security Issues: Result List” on page 186). Each issue also holds a consolidation of the URLs that are vulnerable to this issue. Under each URL, is a listing of the vulnerable data. • Detail Pane: Displays information to enable you to understand why this is an issue, what it can do if gone untreated, fix recommendations, the variants of the test requests that were sent, the response of your application for which AppScan marked the test as an issue, and more. To learn more about the Issues view, see Chapter 8, “Results: Security Issues,” on page 185.
	Tasks View displays design remediations to address and to prevent security issues. It provides an easy and efficient way for you to communicate with personnel about the issues that your application has today and, in concise and accurate language, how to fix the design of your application. • Application Tree: Shows the number next to each node in the tree indicates how many remediation tasks are relevant for each item. • Result List: Shows an icon per remediation task, indicating the priority value that is assigned to this task (see “Remediation Tasks: Result List” on page 205). • Detail Pane: Shows details of the remediation task and lists the issues that will be addressed with this task. To learn more about the Remediations view, see Chapter 9, “Results: Remediation Tasks,” on page 205.

See “Step 4: Communicating results.”

Step 4: Communicating results

Step four of the tutorial.

About this task

There are two basic ways of communicating scan results to your teams:

Reports:

You can choose a report template based on the intended report: make one report for QA, one for executives, one for developers, and so on. As you get further along the development lifecycle of your application, the compliance reports will become more and more essential. You can get a full report of how your application stands up to selected government and industry standards and regulations. See Chapter 10, “Reports,” on page 209 for details.

Individual issues:

If there is a specific issue that is to be addressed by a specific team or individual, you can send the tests and their results to the relevant party. See “Report false positive test results” on page 199

Sample scans

The sample scans can help give you a feel for using AppScan and what scan results look like.

Four sample scans are saved to your machine when AppScan is installed. You can open them to see how they are configured, and how the results are displayed in AppScan. They can be found in your main AppScan Standard folder, whose default location is:

The scans are:

demo.testfire.net.scan

This is a scan of the AppScan demonstration test site. You can review the configuration and results. You can also send additional requests to the site and continue the scan with the new data.

Glass_Box_DotNet_Demo.scan and Glass_Box_Java_Demo.scan

These two scans are examples of glass box scans using a .NET application server and a Java server respectively. You can review the configurations and drill down to individual issues. For issues discovered by glass box the information includes the actual problematic source code.

Note: Glass box scanning requires access to an agent located on the server of the application being scanned, and you do not have access to the agent that was used for this scan, you cannot continue the scan.

GSC_demo.testfire.scan

This is a web services scan of the AppScan demonstration test site. You can review the configuration and results. If you have installed GSC (Generic Service Client) you can use it to send additional requests to the site and continue the scan with the new data.

Chapter 4. Configuring

You configure a scan by choosing settings that best describe your application, and the kind of testing you want.

You can choose between two methods of configuring regular scans:

- “Scan configuration wizard” is useful if:
 - You are new to AppScan, or
 - Most of the settings you require are standard, or
 - You do not need to change settings between scans, and have already saved a scan template with your own specifications
- “Scan configuration dialog box” on page 39 is useful if:
 - You need to customize many settings for a scan, or
 - You want to create and save a custom scan template

In both cases you start with a scan template (see “Scan templates” on page 121) and can optionally change settings as necessary. However, some advanced settings cannot be changed using the wizard, but only using the dialog box.

You can base your scan on the **Regular Scan template**, or you can load a previously saved template of your own. (A “scan template” is simply a set of scan configuration settings that have been saved.) Saving and loading your own scan templates is the most efficient method when you regularly use the same settings for your scans.

Note: After you have configured the scan, but before you start it, you can utilize “Scan Expert” on page 143 to evaluate the configuration for your particular application, and suggest changes to optimize it.

See also: “Workflow description” on page 16

Scan configuration wizard

You can quickly configure basic scans using the wizard.

The **Scan Configuration Wizard** is the simplest way to configure and start a scan when you do not need to change many of the default options. However, if you need to change advanced options, you may prefer to use the “Scan configuration dialog box” on page 39.

The wizard gives you access to the most common configuration options, the main difference being the way the options are arranged. (In both the table below and the sections following, cross-references to options that are the same in both the wizard and the dialog box point to the dialog box section of this guide.)

Explore with	Description
AppScan (automatically or manually)	Select this option for most web application scans. The application is explored manually and/or automatically with requests sent from AppScan to the application.
External device/client (with AppScan as recording proxy)	Select this option to use AppScan's External Traffic Recorder as a recording proxy, and manually explore RESTful or other non-SOAP web services - or SOAP services that do not require security envelopes - using a mobile phone, simulator, or emulator. AppScan displays the domains and requests in its External Traffic Recorder, and create appropriate tests from the input.

Explore with	Description
Generic Service Client (WSDL)	Select this option for web services with a WSDL file. Generic Service Client (GSC) uses your web service's WSDL file to display a simple interface showing the services available, and lets you enter parameters and view the results. Use the GSC interface to explore your web service manually, so that AppScan can use your input to create appropriate tests. Note: This option is only available if GSC (Generic Service Client) is installed on your machine. You can click the blue link to download GSC to your computer, and then double-click on the .EXE file to install it.

Launching the Scan Configuration Wizard

About this task

You can launch the **Scan Configuration wizard** from the AppScan Welcome Screen (which appears when you launch AppScan), or by clicking the New Scan icon on the toolbar

Note: If the **Welcome** screen does not appear when AppScan starts, your **Preference** settings have been changed. To change them back, see “General tab” on page 240. (You can also open the Welcome Screen at any time from the *View* menu.



Procedure

- To launch the wizard do one of the following:
 - On the Welcome Screen, click **Create New Scan**.
 - If AppScan is already open, click the **New Scan** icon  on the toolbar.
- In the New Scan dialog box, verify that the **Launch Scan Configuration Wizard** check box is selected.
- In the Predefined Templates list, select a scan template by clicking on it. If you do not need a *specific* template, select the **Regular Scan** template (for details see “Predefined templates” on page 121).
The Scan Configuration Welcome appears.
- In the **Scan Configuration Wizard** Welcome screen, select the required scanning option.

Explore with	Description
AppScan (automatically or manually)	Select this option for most web application scans. The application is explored manually and/or automatically with requests sent from AppScan to the application.

Explore with	Description
External device/client (with AppScan as recording proxy)	Select this option to use AppScan's External Traffic Recorder as a recording proxy, and manually explore RESTful or other non-SOAP web services - or SOAP services that do not require security envelopes - using a mobile phone, simulator, or emulator. AppScan displays the domains and requests in its External Traffic Recorder, and create appropriate tests from the input.
Generic Service Client (WSDL)	Select this option for web services with a WSDL file. Generic Service Client (GSC) uses your web service's WSDL file to display a simple interface showing the services available, and lets you enter parameters and view the results. Use the GSC interface to explore your web service manually, so that AppScan can use your input to create appropriate tests. Note: This option is only available if GSC (Generic Service Client) is installed on your machine. You can click the blue link to download GSC to your computer, and then double-click on the .EXE file to install it.

Wizard for exploring with AppScan

Lists the steps of this wizard.

1. "URL and Servers"
2. (Optional:) "Connection Settings" on page 30
3. "Login Management" on page 30
4. (Optional:) "Login Management Details" on page 31
5. "Test Policy" on page 32
6. "Complete" on page 32

URL and Servers

Define the starting URL for the scan and any additional servers and domains that need to be included.

About this task

If you selected **Web Application Scan** in the Welcome step, the first step of the wizard is to define the Starting URL and server settings for the scan.

Procedure

1. **Starting URL:** Type the URL of your application. The scan will start at this URL.

AppScan attempts to reach the URL, and if it succeeds a green checkmark and "Connected to server" confirmation appear underneath the Starting URL. If the problem can be corrected with a configuration change, AppScan suggests what change is needed and links to the appropriate view in the Configuration dialog box.



Opens the page you entered in a browser. If necessary you can browse to a different URL and use the toolbar to set the new page as the Start URL for the scan. Close the browser to continue with scan configuration.



Tests the Starting URL again. Click this button if the server was down when you configured, and later resumed.

Tip: If you change the domain, scheme or port of the Starting URL *after configuring the scan*, changes to the configuration may be needed. You will be asked if you want AppScan to try to do this automatically. For more details, see "Changing the Starting URL host" on page 43

2. **Case Sensitive Path:** When this check box is selected (default), links that differ from each other by case only are treated as different pages. For example, "ReadMe.as" would be considered different to "readme.as". In most cases you should check the check box for Unix-based servers, and uncheck the check box for Windows-based servers.
3. **Additional Servers and Domains:** If your application includes servers or domains other than those of the Starting URL, *and your AppScan license includes them*, you must add them here so that they will be included in the scan:
 - a. Click the plus icon  to open the Add Servers and Domains dialog box
 - b. Type in the hostname or IP address of the server, or the domain name (e.g. sitetoscan.com), then click **OK**.The new item is added to the list.
4. **I need to configure additional connectivity settings:** By default AppScan uses Internet Explorer proxy settings. Select this check box only if you want AppScan to use a different proxy. This will open an additional step, "Connection Settings," when you click the Next button.
5. Click **Next** to continue to the next step of the wizard.

What to do next

"Login Management"

If you selected "I need to configure additional connectivity settings", continue with "Connection Settings"

Connection Settings

(Optional:) Configure proxy settings and server-level authentication if needed.

About this task

By default AppScan uses Internet Explorer proxy settings. This step opens only if you select the Additional Connection Settings check box in the URL and Servers step of the wizard, and lets you configure AppScan to use different proxy settings, or none at all.

Procedure

1. Select one of the three proxy options:
 - **Use Internet Explorer proxy settings:** (Default) When selected AppScan will use the address and port of the Internet Explorer connection when connecting to the application.
 - **Don't use proxy:** Select this if you do not want AppScan to use proxy settings at all.
 - **Use custom proxy settings:** Select this if you want to use an address and port other than those used by Internet Explorer.
2. If you selected one of the options that uses a proxy, you can optionally set the Username, Password and Domain for AppScan to use when connecting. Click the **Configure** button to do this.
3. If your platform requires server-level authentication, set the Username, Password and Domain in the **HTTP Authentication** area.
4. Click **Next**.

What to do next

"Login Management"

Login Management

Configure the login method, and record the login procedure if needed.

About this task

The Login Management step of the wizard lets you select one of three methods for AppScan to use when it encounters login pages during a scan:

- **Recorded Login:** (Recommended method) If you select this option, AppScan uses a login procedure that you record, completing fields and clicking on links like a real user.

For more details see “Recording a login” on page 45.

If login requires human interaction each time (such as Two-Factor Authentication, One-Time Passwords, or CAPTCHA), select the Prompt option.

- **Prompt:** In this case you must still record a login procedure. Although AppScan will not attempt to log in using the procedure you record, it needs the procedure as a reference to know when it is logged out.
- **Automatic Login:** If AppScan will be able to log in to the site using a name and password only, without a special procedure, select this option and enter the Username and Password.
- **No Login:** Select this option only if the application does not require logging in, or if for some other reason you do not want AppScan to log in.

Procedure

1. Select the radio button for the required Login method.
2. Do one of the following:
 - For Recorded Login or Prompt, click **Record** or **Import** set the a login sequence (see “Login tab” on page 44 for details).
 - For Automatic Login, simply type in **Username** and **Password**.

Note: If you record a login, when you finish recording a dialog box may open asking you to confirm that the login data AppScan extracted is correct. Fill in or correct the parameters and values as necessary, then click **OK**.

If you have recorded a valid login sequence, the key icon turns from gray  to green , indicating that the an in-session page has been identified.

Note: If the key icon turns red , AppScan attempted to but was unable to identify any pattern in the in-session page that it can use during scanning to verify that it has not been logged out. To remedy this you should open the extra step of the wizard and either provide AppScan with an identifier manually (see next step).

3. If you select the **I want to configure In-Session Detection Options** check box, an additional wizard step, “Login Management Details,” will open when you click **Next**. Select this only if you need to edit the login sequence (see note to previous step).
4. Click **Next**.

What to do next

“Test Policy” on page 32

If you selected “I want to configure In-Session Detection Options”, continue with “Login Management Details”

Login Management Details

(Optional:) Review and edit the login sequence you recorded.

About this task

The additional Login Management settings dialog box lets you review and edit the login sequence that you recorded, and the pattern that AppScan uses during scanning to verify that it is logged in ("in-session detection pattern").

Procedure

1. Review and edit the login sequence, and make sure the session detection pattern is valid. Options are described in "Review & Validate tab" on page 47
2. Click **Next**.

What to do next

"Test Policy"

Test Policy

Limiting the scan to the specific types of tests you want can reduce scan time.

About this task

The number of tests which AppScan sends during a scan can reach the thousands. Sometimes it is preferable to reduce scan time by limiting the scan to certain types only. This is Test Policy.

AppScan comes with a Default Test Policy, and with some additional Test Policy configurations that you can select. You can also use your own User-Defined Test Policies.

The Test Policy step of the wizard shows the name of the Test Policy that the current policy is based on, and its description.

Procedure

1. Check that the Test policy is appropriate for your needs. (If you are in doubt, leave the Default Test Policy.)
2. To load a different Test Policy, click on one of the Pre-Defined Policies or Recent Policies in the **Policy Files** pane. For details see "Test Policy view" on page 95.
3. **Send tests on login and logout pages:** By default, AppScan will test your login and logout pages along with the rest of the application. You should leave this default configuration, unless:
 - Your application has safeguards that lock out users who provide illegal input on these pages, or
 - Your application flow would be altered if these pages were tested

If you are unsure how your application will respond to these tests, leave this option selected.

4. **Do not send session identifiers when testing login pages:** (This check box is active, and selected by default, only if the previous check box is selected.) It is recommended to leave this check box selected, since session identifiers could limit test success when testing login pages. Clear it only if you are sure that valid session tokens are necessary to test your login pages.

If you are unsure how your application will respond, leave this option selected.

5. Click **Next** to proceed to the final stage of the wizard.

What to do next

"Complete"

Complete

Having completed the steps of the wizard, you now decide how and when to start the scan that you have configured.

About this task

If you select one of the options that starts a scan right away, you can also elect to run *Scan Expert* before the main scan starts. Scan Expert logs into your application and performs a short, preliminary scan to evaluate the settings you have configured. It then suggests configuration changes if necessary. Incorporating these suggestions (automatically or manually) can greatly increase the efficiency of the main scan.

Procedure

1. Select one of the options:
 - **Start a full automatic scan:** Start a full scan of the application (Explore followed immediately by Test).
 - **Start with automatic Explore only:** Explore the application, but do not proceed to the Test stage. (You can run the Test stage later.)
 - **Start with Manual Explore:** The browser will open and you can manually explore your site by clicking on links and completing fields. AppScan will record the results to use in the Test stage.
 - **I will start the scan later:** Close the wizard without starting a scan. The next time you start a scan, this template will be used.
2. **Start Scan Expert when Scan Configuration is Complete:** (Active only if one of the first three scan options has been selected.) Select this check box in you want Scan Expert to evaluate your configuration before starting the main scan.

What to do next

See: “Scan progress” on page 139

See also:

“Starting scans” on page 137

Chapter 8, “Results: Security Issues,” on page 185

Wizard for exploring with an external device or client

Lists the steps of this wizard.

1. “Recording Proxy”
2. (Optional:) “Connection Settings” on page 34
3. “SSL Certificate” on page 34
4. “Login Management” on page 35
5. (Optional:) “Login Management Details” on page 35
6. “Test Policy” on page 36
7. “Complete” on page 36

Recording Proxy

Configure proxy port and client type.

Procedure

1. **Recording Proxy port:** Configure the port on which AppScan will receive the Manual Explore requests you send from your remote device or external client. You can let AppScan choose an available port automatically.

Tip: It is better to choose the port yourself, as the automatically chosen port may change from session to session. However, you must not specify a port if you will be opening more than one instance of AppScan at the same time; if you need to use more than one instance, configure AppScan to choose the port automatically.

2. **Record from:** Define whether you will be recording from an external client on the same machine as AppScan (such as a simulator or emulator), or a remote device (such as a mobile phone).
3. Configure your remote device or external client to use the IP and port shown in the information pane.
4. **I need to configure additional connectivity settings:** By default AppScan uses Internet Explorer proxy settings. Select this check box only if you want AppScan to use a different proxy. This will open an additional step when you click the Next button.
5. Click **Next** to continue to the next step of the wizard.

What to do next

“SSL Certificate”

If you selected "I need to configure additional connectivity settings", continue to “Connection Settings”

Connection Settings

(Optional:) Configure proxy settings and server-level authentication if needed.

Procedure

1. Select one of the three proxy options:
 - **Use Internet Explorer proxy settings:** (Default) When selected AppScan will use the address and port of the Internet Explorer connection when connecting to the application.
 - **Don't use proxy:** Select this if you do not want AppScan to use proxy settings at all.
 - **Use custom proxy settings:** Select this if you want to use an address and port other than those used by Internet Explorer.
2. If you selected one of the options that uses a proxy, you can optionally set the Username, Password and Domain for AppScan to use when connecting. Click the **Configure** button to do this.
3. If your platform requires server-level authentication, set the Username, Password and Domain in the **HTTP Authentication** area.
4. Click **Next**.

What to do next

“SSL Certificate”

SSL Certificate

If your server uses HTTPS, you need to add the AppScan SSL root certificate (so that requests sent using AppScan as proxy will be accepted).

Procedure

1. Click **Install AppScan certificate locally**, and then click **Yes** on the Windows dialog box that appears to allow the change.

Note: Once the certificate is installed, the button is replaced with a confirmation message, and the button will not appear the next time you use the wizard (unless the certificate is uninstalled).

Note: To uninstall, go to **Tools > Options > Recording Proxy**, and click **Remove**.

2. If you will be exploring from a remote device (mobile phone or emulator), you must also install the certificate on that device:

- a. On the device, browse to: <http://appscan>
- b. On the device, click **Install AppScan SSL Certificate**.

You are now ready to record the login from your device, with AppScan as recording proxy.

3. Click **Next**.

What to do next

“Login Management”

Login Management

Record the login procedure.

Before you begin



The orange key icon indicates that the login has not yet been recorded.

Procedure

1. On your device, browse to your application's login page.
2. In AppScan, click  **Record**.
The External Login Recorder opens, and should indicate that it is connected to your external device.
3. On your device, log into the application.
The login requests are listed in the External Login Recorder.
4. Once you are logged in to your application, click **Stop recording** (in the External Login Recorder).
5. Optionally review the list for unnecessary requests (for example, requests to a different domain), and delete them by selecting and clicking .
6. Click **OK** to close the recorder.



The green key icon indicates that in-session status has been detected.

7. If you select the **I want to configure In-Session Detection Options** check box, an additional wizard step, “Login Management Details,” will open when you click **Next**. Select this only if you need to edit the login sequence or activate/deactivate In-Session detection (see note to previous step).
8. In the wizard, click **Next**.

Related topic:

“External Login Recorder” on page 133

What to do next

“Test Policy” on page 36

If you selected “I want to configure In-Session Detection Options”, continue to “Login Management Details” on page 31

Login Management Details

(Optional:) Review and edit the login sequence you recorded.

Procedure

1. Review and edit the login sequence, and make sure the session detection pattern is valid. Options are described in “Review & Validate tab” on page 47

2. Click **Next**.

What to do next

“Test Policy”

Test Policy

Limiting the scan to the specific types of tests you want can reduce scan time.

About this task

The number of tests which AppScan sends during a scan can reach the thousands. Sometimes it is preferable to reduce scan time by limiting the scan to certain types only. This is Test Policy.

AppScan comes with a Default Test Policy, and with some additional Test Policy configurations that you can select. You can also use your own User-Defined Test Policies.

The Test Policy step of the wizard shows the name of the Test Policy that the current policy is based on, and its description.

Procedure

1. Check that the Test policy is appropriate for your needs. (If you are in doubt, leave the Default Test Policy.)
2. To load a different Test Policy, click on one of the Pre-Defined Policies or Recent Policies in the **Policy Files** pane. For details see “Test Policy view” on page 95.
3. **Send tests on login and logout pages:** By default, AppScan will test your login and logout pages along with the rest of the application. You should leave this default configuration, unless:
 - Your application has safeguards that lock out users who provide illegal input on these pages, or
 - Your application flow would be altered if these pages were testedIf you are unsure how your application will respond to these tests, leave this option selected.
4. **Do not send session identifiers when testing login pages:** (This check box is active, and selected by default, only if the previous check box is selected.) It is recommended to leave this check box selected, since session identifiers could limit test success when testing login pages. Clear it only if you are sure that valid session tokens are necessary to test your login pages.
If you are unsure how your application will respond, leave this option selected.
5. Click **Next** to proceed to the final stage of the wizard.

What to do next

“Complete”

Complete

AppScan is now configured as a recording proxy for your device to send requests to the application. You are now ready to start your Manual Explore of the application.

Procedure

1. Click **Finish**.
The External Traffic Recorder opens with a status message indicating that it is connected to your device.
2. When the External Traffic Recorder opens with status "Waiting for incoming connections, manually Explore the web service from your device/application:
 - a. Using your device or application, explore the web service.

As you explore, domains detected are listed in the left pane of the recorder, and URLs are listed in the right pane.

- b. When finished, in AppScan click **Stop Recording**.
3. Review and edit the Manual Explore data:

Domains detected

All domains to which requests were sent are listed, and by default selected for adding to the list of Additional Servers and Domains (Configuration > URLs and Servers > Additional Servers and Domains) so they can be included in the scan. You can deselect any you do not want included in the scan.

Tip: You should deselect any domains that belong to other companies.

Requests sent

All requests sent by the device to domains *that are selected* (in the left pane), are listed. If you select/clear domains in the left pane, the requests list is updated. You can delete specific requests if they are not needed.

Tip: If the total number of filtered requests is more than 200, deleting some of them may produce a more efficient scan.

Note: At this stage you can click **Export** to save the Explore data for use on another machine.

4. Click OK to close the recorder.
AppScan takes a few moments to process and display the data.
5. To start the Test stage, click **Scan > Test Only**
The Test stage starts and when complete the scan results are displayed.

Related topics:

-
- “Recording Proxy tab” on page 237
- Chapter 8, “Results: Security Issues,” on page 185

Wizard for exploring with GSC

Lists the steps of this wizard.

1. “URL and Servers”
2. (Optional:) “Connection Settings” on page 38
3. “Test Policy” on page 38
4. “Complete” on page 39

URL and Servers

Supply the URL of the WSDL file for the service, and define any additional servers and domains to include in the scan.

Procedure

1. **WSDL URL:** Type in the URL of the WSDL file for the service.
Example: <http://www.sitetoscan.com/Service1.asmx?wsdl>
2. To check that you have entered the URL correctly, you can open the URL you have entered in the AppScan browser:
 - a. Click  (This button is enabled only after you have entered the URL in the text box.) The embedded browser opens to the URL you entered.
 - b. If necessary correct the URL.
 - c. Close the browser to continue with scan configuration.

3. **Case Sensitive Path:** When this check box is selected (default), links that differ from each other by case only are treated as different pages. For example, "ReadMe.as" would be considered different to "readme.as". In most cases you should check the check box for Unix-based servers, and uncheck the check box for Windows-based servers.
4. **Additional Servers and Domains:** If your application includes servers or domains other than those of the Starting URL, *and your AppScan license includes them*, you must add them here so that they will be included in the scan:
 - a. Click  icon to open the *Add Servers and Domains* dialog box.
 - b. Type in the hostname or IP address of the server, or the domain name (e.g. sitetoscan.com), then click **OK**The new item is added to the list.
5. **I need to configure additional connectivity settings:** By default AppScan uses Internet Explorer proxy settings. Select this check box only if you want AppScan to use a different proxy. This will open an additional step when you click the Next button.
6. Click **Next** to continue to the next step of the wizard.

What to do next

"Test Policy"

or, if you selected the **I need to configure...** check box:

"Connection Settings"

Connection Settings

(Optional:) Configure proxy settings and server-level authentication if needed.

About this task

By default AppScan uses Internet Explorer proxy settings. The additional URL and Servers dialog box lets you configure AppScan to use different proxy settings, or none at all.

Procedure

1. Select one of the three proxy options:
 - **Use Internet Explorer proxy settings:** (Default) When selected AppScan will use the address and port of the Internet Explorer connection when connecting to the application.
 - **Don't use proxy:** Select this if you do not want AppScan to use proxy settings at all.
 - **Use custom proxy settings:** Select this if you want to use an address and port other than those used by Internet Explorer.
2. If you selected one of the options that uses a proxy, you can optionally set the Username, Password and Domain for AppScan to use when connecting. Click the **Configure** button to do this.
3. If your platform requires server-level authentication, set the Username, Password and Domain in the **HTTP Authentication** area.
4. Click **Next** to proceed to the Test Policy Settings step.

What to do next

"Test Policy"

Test Policy

Define which kinds of tests AppScan will send to the service when your Manual Explore is complete.

Procedure

1. In the **WSDL Scan Configuration Wizard > Platform Authentication** dialog box, click **Next**.
The **WSDL Test Policy step** of the wizard appears (same as “Test Policy” on page 32).
2. Check that the Test Policy is appropriate for your needs. (Unless you have defined a custom policy, the web Services Test Policy is recommended.)
 - To load a different Test Policy, click **Load**.
The **Load Test Policy** dialog box opens, letting you select one of the pre-defined or user-defined policies.
 - To edit the current Test Policy, click **Edit**.
The **Test Policy Manager** opens, letting you configure exactly which tests are included in the scan.
For details see “Importing a Test Policy” on page 98
3. If necessary you can click the **Advanced** Test Settings button to configure additional options. For details see “Importing a Test Policy” on page 98. Otherwise, click **Next** to proceed to the final stage of the wizard.

What to do next

“Complete”

Complete

You are now ready to start manually exploring the service using GSC.

Procedure

Click **Finish**.

A prompt that Generic Service Client (GSC) is about to open may appear (if so, click **OK**), and then the program opens.

What to do next

“Using GSC” on page 134

Scan configuration dialog box

About this task

The Scan Configuration dialog box provides many options for configuring your scans. The main options are also available via the Scan Configuration Wizard, but using the dialog box is recommended when you want to change many of the default settings, or tweak an existing configuration.

Procedure

To open the Scan Configuration dialog box, click the **Configuration** icon  on the toolbar (or press **F10**).

The Scan Configuration dialog box has different views, divided into four groups, that are accessed by clicking the relevant item in the view selection pane, on the left-hand side.

Note: If you stop a scan and change the configuration, changes will not affect requests already sent. To apply changes to the entire scan you must start a new scan.

Tip: You can change configuration options in more than one of the views, and then click OK to save all changes. (Your changes are kept as you navigate between the views, but only saved when you click OK.)

View	Select to configure:
<i>Explore</i>	
"URL and Servers view"	Starting URL, system type, and additional servers
"Login Management view" on page 44	Set login method, record a login sequence (optional), and configure in-session detection
"Environment Definition view" on page 56	Provide information about the application environment
"Exclude Paths and Files view" on page 57	Paths and file types to exclude from the scan
"Explore Options view" on page 62	Scan limits, link extraction methods and general Explore method
"Parameters and Cookies view" on page 67	Identify session IDs and list parameters to exclude from the scan
"Automatic Form Fill view" on page 80	Provide AppScan with valid parameter values for filling forms
"Error Pages view" on page 82	Add strings, regexps and URLs to identify custom error pages
"Multi-Step Operations view" on page 84	Record and manage multi-step operations that are required to reach parts of the application
"Content-Based Results view" on page 90	For applications that do not have a hierarchical URL structure, such as single entry-point applications, define how AppScan arranges the site tree.
"Glass Box view" on page 92	If the AppScan glass box agent is installed on your application server, configure glass box scanning here.
<i>Connection</i>	
"Communication and Proxy view" on page 93	Configure communication timeout and proxy server settings
"HTTP Authentication view" on page 94	Add server-level authentication and client-side certificates, if required by the application
<i>Test</i>	
"Test Policy view" on page 95	Define and edit test policy (which tests are sent to the application)
"Test Options view" on page 100	Additional test options
"Privilege Escalation view" on page 102	Refer AppScan to scans run using different user privileges, to discover privileged resources that are available to users with insufficient privileges
"Malware view" on page 103	Test for malicious links.
<i>General</i>	
"Scan Expert view" on page 104	Configure Scan Expert behavior and modules
"Advanced Configuration view" on page 106	Configure advanced scan options

URL and Servers view

URL and Servers view of the Scan Configuration dialog box.

You *must* define the URL from which to start the scan. The remaining settings are optional.

Setting	Details
Starting URL	Provide the URL of your application to be the starting point of the scan. AppScan attempts to reach the URL, and if it succeeds a green checkmark and "Connected to server" confirmation appear underneath the Starting URL. If the problem can be corrected with a configuration change, AppScan suggests what change is needed and links to the appropriate view in the Configuration dialog box.  Opens the page you entered in a browser. If necessary you can browse to a different URL and use the toolbar to set the new page as the Start URL for the scan. Close the browser to continue with scan configuration. Note: The default browser is used. You can set the default browser to be either of the two built-in browsers (IE or Chromium), or a supported external browser, in Tools > Options > Preferences tab  Tests the Starting URL again. Click this button if the server was down when you configured, and later resumed. Tip: If you change the domain, scheme or port of the Starting URL <i>after configuring the scan</i>, changes to the configuration may be needed. You will be asked if you want AppScan to try to do this automatically. For more details, see "Changing the Starting URL host" on page 43
Scan only links in and below this directory.	When selected, the scan is limited to pages below the starting URL. Links to other URLs will not be scanned, even if they are included in the list of Additional Servers and Domains (below). For details see "Limiting scan to the Starting URL folder"
Treat all paths as case-sensitive	When selected, links that differ from each other by case will be considered different pages. For example, "ReadMe.as" would be considered different to "readme.as". When deselected, all URLs appear in lowercase. Select this check box if the file system on your application host is case sensitive. In most cases, you will want to check the check box for Unix-based servers, and uncheck the check box for Windows-based servers.
Additional Servers and Domains	If your application includes links to domains other than the domain of the Starting URL, you must add them here in order for them to be included them in the scan. Note: You can only add servers/domains that are included in your AppScan license. For details see "Additional Servers and Domains" on page 43

Limiting scan to the Starting URL folder

You can easily limit a scan to within and below the folder of the Starting URL.

About this task

The check box under the Starting URL field automatically creates the necessary filters to limit your scan to within and below a particular directory.

Procedure

1. Open **Scan Configuration > URL and Servers**.
2. Type or paste-in the URL of the directory you want to restrict the scan to.
3. Select the **Scan only links in and below this directory** check box.

The scan will now be limited to paths under this URL. Links outside this range will not be scanned.

Example

If the Starting URL is defined as `http://main/bank/`

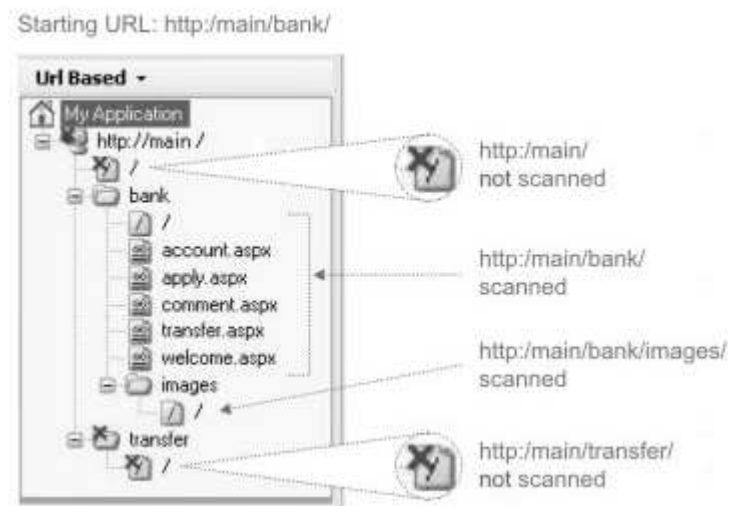
- Links to the following will be scanned:
 - `http://main/bank/transfer.aspx/`
 - `http://main/bank/transfer/page_1.aspx/`
- Links to the following will be ignored:
 - `http://main/transfer.aspx/`
 - `http://main/transfer/page_1.aspx/`

When you configure the Starting URL, and select this check box, the following two items are added at the top of the Exclude Paths and Files table (Scan Configuration > Exclude Paths and Files):

Item Type	Path	Matching	Behavior
Exclude (Start URL)	.*	Regular Expression	Always first in the table. Cannot be demoted. Cannot be edited or deleted. However, if the next item (Exception) is deleted, this is deleted too.
Exception (Start URL)	<code>http://main/bank</code>	Full Path	Always second in the table. Cannot be promoted or demoted. <i>Can</i> be edited. (This is to allow for editing the exception in rare cases where AppScan misidentifies the Starting URL directory.) If deleted, the previous item (Exclude) is also deleted, and the Starting URL check box is deselected.

Note: Unlike other exclusions and exceptions, these two have a gray background, indicating their special status.

When the scan starts, and the Application Tree fills in the left pane, links to parts of the application that are not below the Starting URL are shown with a red X, indicating that they were not scanned.



What to do next

You can verify that the Exception added is correct, or edit it, from **Exclude Paths and Files** view (see “Limiting scan to a specific folder” on page 60).

Additional Servers and Domains

You can include domains other than that of the Starting URL in the scan.

About this task

If your application includes links to domains other than the domain of the Starting URL, you must add them here in order for AppScan to include them in the scan.

Note: You can only add servers/domains that are included in your AppScan license.

Procedure

- To add a server, click  and type in the server/domain name.
Example: demo.testfire.net or 65.61.137.117
The server is added to the list.
- To edit a server in the list, select it and click 
- To delete a server from the list, select it and click 

Changing the Starting URL host

You can change the host, scheme, or port in the Starting URL of a configured scan, without re-recording the login, multi-step operations, or manual explore data.

About this task

If you have already recorded a login, multi-step operations, and/or a Manual Explore - and then the host, scheme or port of your Starting URL changes - requests and responses in these recordings must be updated and verified. Click **Scan > Change Host/Scheme/Port** to open a dialog box from which you can change the URL, and get AppScan to automatically update, verify and confirm the necessary changes.

The dialog box shows the steps being performed, and indicates when each step is successful. If the update process does not complete successfully, the dialog box indicates which step failed, and gives you the option to save the changes and proceed manually, or undo all changes.

Important: In some cases AppScan may update responses incorrectly, and part or all of the scan will fail. If that happens you will need to re-record the problematic procedures.

Note: Although *Manual* Explore data is updated, *Automatic* Explore data, and scan results, are deleted when you change the Starting URL.

Note: This option can be used to change *only* the host, scheme, or port of *only* the Starting URL. If you need to make *other* changes to the Starting URL, or change the host, scheme or port for one of the *Additional* Domains in the scan, you cannot use this option. Instead, save the scan as a template, and use that to create a new scan.

Procedure

1. Open the scan (.scan) or scan template (.scant) file and then:
 - **For a scan template:** On the toolbar, click **Scan Configuration > URL and Servers**.
 - **For a configured scan:** On the menubar, click **Scan > Change Host/Scheme/Port**.

2. Change the host, scheme or port of Starting URL as needed.
AppScan attempts to connect to the server and a green check mark indicates that this succeeded.
3. Click **OK**.
AppScan attempts to update the configuration, and a green check mark indicates each step that completes successfully.
4. **If all steps succeed:** Click **OK** to save the configuration changes.
If one of the steps fails:
 - To save the configuration changes, complete the failed step manually, and try again, click **OK**.
 - To undo all changes and revert to the original Starting URL, click **Cancel**.

Important: In some cases AppScan may update responses incorrectly, and part or all of the scan will fail. If that happens you will need to re-record the problematic procedures.

Login Management view

Login Management view of the Scan Configuration dialog box.

Login Management view of the **Scan Configuration** dialog box is used for configuring how AppScan logs in to your application, and how it recognizes when it has been logged out.

AppScan can automatically detect login requests and fills in the username and password parameters. If your application has a non-standard login sequence of actions, you can record these actions for AppScan to use.

Login Management view has these tabs:

- Login
- Review & Validate
- Session IDs
- Advanced

Login tab

Scan Configuration > Login Management > Login tab.

The **Login Management** view of the **Scan Configuration** dialog box is used to define how AppScan logs in to your application, and how it can recognize when it gets logged out.

AppScan can automatically detect login requests and fills in the username and password parameters. If your application has a non-standard login sequence of actions, you can record these actions for AppScan to use.

Setting	Details
<i>Select Login Method</i>	
Recorded (Recommended)	(Default method) Select this method to open the browser and record a login sequence (both HTTP requests and user actions are recorded). AppScan will use this sequence whenever it needs to login to the application.  Record is used to record the sequence. For web applications, see “Recording a login” on page 45; for RESTful (and other) web services, see “Exploring with the External Traffic Recorder” on page 131
Automatic Login	Select this method to let AppScan automatically detect the login form of your application and use the username and password you supply. (This method can be less reliable than the Recorded Login method.)

Setting	Details
Prompt	Select this method if login requires human interaction each time (such as Two-Factor Authentication, One-Time Passwords, or CAPCHA). Note that even if you select this option, you must record a login sequence. This is to provide AppScan with an in-session page that it can later use to verify that it is logged-in. For details see “Recording a login”
None	Select this option if the application does not require users to log in.
<i>Login Validation Status Indicator</i>	
Key icon	The key icon indicates the status of In-Session Detection:  Enabled and configured. (An in-session page has been identified in login sequence, either automatically or by the user.)  Enabled but not fully configured.  Enabled but not configuration failed.  Disabled. See “Select Detection Pattern dialog box” on page 53 for details.
<i>Import or Export Login Settings</i>	
Import	When you record a login sequence it is saved as part of the scan. If you save the scan as a template, the login sequence is saved as part of the template. To import a login sequence that was previously saved as a *.login file, click the Import button.
Export	To export the login sequence by itself, to use in future scans, click the Export button. The sequence is saved as a *.login file.

Recording a login: Before you begin

Before you can record a login sequence the Starting URL must be defined (either in Configuration > URL and Server view or in the wizard).

About this task

Recorded Login lets you teach AppScan the procedure for logging in to your site: which links to click, which text to input in forms, and the order in which to do them. As soon as you have recorded this, AppScan will attempt to identify an in-session pattern that it can use in future to verify that it is logged in. Once this is done, AppScan can use the login sequence to log itself back in during the scan, whenever it detects that it has been logged out.

AppScan must know at all times whether it is logged into or out of the site, so it can evaluate the site's responses correctly. During the scan, AppScan sends the In-Session Detection Request repeatedly, and checks that the response contains the In-Session Detection Pattern, to verify that it is still logged in. If AppScan does not find the pattern in the page's response, AppScan assumes it has been logged out, and

attempts to log in again by replaying the login sequence. It follows that the login sequence is typically played many times during a scan. It is therefore best that it contains as few steps as possible. It is also helpful if the In-Session page is a small page, and does not contain tracked parameters or cookies, since these can also increase scan time significantly.

Procedure

1. In **Scan Configuration > Login Management > Login tab**, select **Recorded**.

2. Click  > **AppScan IE browser**

The browser opens to the Starting URL and begins recording your actions.

Note: If your website login does not support Internet Explorer, click > **Use AppScan Chromium browser** instead.

Note: If the Starting URL has not yet been defined you are warned that you must define it before you can proceed (see “URL and Servers view” on page 40).

Note: If a login procedure has previously been recorded, you are warned that the new recording will overwrite the existing one.

Note: If you have configured AppScan to use an external browser for scanning (Tools > Options > Use External Browser > Select Browser), you will be given the choice of recording the login with either the AppScan browser or the external browser. If possible it is recommended to use the AppScan browser for the login recording (even if using a different browser for scanning), as it records extra information that improves login success during scanning. If recording the login with the AppScan browser does not work for your application, use the external browser.

Note: If your application does not support Internet Explorer you must configure AppScan to work with an external browser.

3. Log in to the site, completing forms and clicking on links as necessary.

Tip: By default, the page you reach when you have logged in will be used by AppScan as the in-session URL. AppScan sends this URL every few seconds during the scan, to check that it is still logged in. If the page sends a large response, or if it includes tracked parameters or cookies, you can improve scan performance by clicking on one or more additional links until you reach a page with a smaller response (while still logged in) and without tracked parameters or cookies. Then, after you close the browser, go to the **Review & Validate** tab and select the *later* page as the "in-session URL".

4. When you have successfully logged in to the site, click **I am logged in to the site**.

AppScan attempts to extract the login information from your login request, for use during scanning.

Note: Sometimes the login page does not provide enough information, and AppScan may ask you to click an additional step after you are logged in, or to log out of the site.

Note: If the login mechanism manipulates the login data using JavaScript, a dialog box may open asking you to confirm that the login data AppScan extracted is correct. Fill in or correct the parameters and values as necessary, then click **OK**.

The Session Information dialog box opens displaying the login requests you recorded, and



changes to , indicating that in-session detection is active.



Note: If the key icon turns red, AppScan attempted but was unable to identify any pattern in the in-session page that it can use during scanning to verify that it has not been logged out. If this happens, you need to identify the "in-session pattern" for AppScan, see "Select Detection Pattern dialog box" on page 53 for details. In some cases a more specific message may appear, with a link to a page in this Help for troubleshooting the problem, see "Login troubleshooting" on page 305.

5. To make changes to the recorded sequence (for example to remove unnecessary steps), refer to "Review & Validate tab."

Tip: Generally speaking the URL which logs the user in (and whose response is the first to include an in-session pattern), should be the one marked In-Session. However, sometimes you may want to select a later URL, that also includes the in-session pattern, but which has the advantage of being a smaller page or of not including tracked parameters or cookies. Additionally, sometimes the POST request with the user credentials is the request which logs you in and first contains the in-session pattern, this is a poor choice for the in-session page, since the in-session check would send the credentials each time, leading to a false positive in session response. See "Optimizing In-Session Detection" on page 54

6. To save the new login sequence, click **OK**.

Tip: If you are sure that the in-session page contains no tracked parameters or cookies, you can improve scan performance by changing the **Advanced Configuration > Session Management: Parse in-session page** setting to "False". See "Advanced Configuration view" on page 106.

Review & Validate tab

Scan Configuration > Login Management > Review & Validate tab

When you record a login sequence, AppScan records both the actions and the requests. These are shown on the two sub tabs: Actions and Requests. When replaying the login AppScan attempts (by default) to reproduce the action-based login; if this is unsuccessful it uses the request-based login.

This tab is used to review and edit:

- Action-based version of the login sequence
- Request-based version of the login sequence
- In-Session Detection Request
- In-Session (or Out-of-Session) Detection Pattern

It is also used to:

- Validate the current settings

Table 1. "Review & Validate tab" settings

Setting	Details
Login Playback	This section appears only if Recorded Login is the selected login method

Table 1. "Review & Validate tab" settings (continued)

Setting	Details
Login Playback Method	AppScan saves two versions of the login sequence you record: one based on the actions you performed, and the other on the HTTP requests actually sent. • Action-Based: (Used by default whenever possible:) AppScan will attempt to log in using action-based login, replaying the clicks and keystrokes of the user. –  Replay: Opens the Action-Based Player and replays the recorded login sequence in its browser. – Edit: Opens the Action-Based editor to view and edit details of the login recording. • Request-Based: If the first method fails, AppScan will use the request-based version, which re-sends the raw HTTP requests from the login recording. – Edit: Opens the Request-Based editor to view and edit details of the login recording. If a message indicates that one of the methods failed, use the other method. Note: If you select Action-Based Login and it fails during the scan, AppScan will try Request-Based Login. If that succeeds, the setting here will be changed automatically to Request-Based.
Automatic Login	<i>This section appears only if Automatic Login is the selected login method</i>
Auto-Detect In-Session Configuration button	Click for AppScan to perform the following actions: • Attempt to log in to the site using the credentials you supplied • Identify an In-Session Detection Pattern on the login page (see below) • Configure session identifiers (see "Session IDs tab" on page 55)
Session Detection	AppScan must know at all times whether it is logged into or out of the site, so it can evaluate the site's responses correctly. During the scan, AppScan sends the In-Session Detection Request repeatedly, and checks that the response contains the In-Session Detection Pattern, to verify that it is still logged in. If AppScan does not find the pattern in the page's response, AppScan assumes it has been logged out, and attempts to log in again by replaying the login sequence. It follows that the login sequence is typically played many times during a scan. It is therefore best that it contains as few steps as possible. It is also helpful if the In-Session page is a small page, and does not contain tracked parameters or cookies, since these can also increase scan time significantly.
In-Session Detection Request	This is the request used by AppScan to verify that it is still in-session. This request should be one that produces different responses depending on whether or not the user is logged in. AppScan attempts to identify valid in-session requests, and you can select one of them from the drop-down list. If none are found, or suitable, you can select your own using the Advanced Request Selection button.
Advanced request selection button	This button opens a dialog box in which you can review requests in the login sequence, and select an In-Session Detection Request. For details, see "Advanced In-Session Request Selection dialog" on page 52
In-Session Detection Pattern	(Active only when an In-Session Detection Request is selected:) This field shows a pattern found in the selected In-Session Detection Request, which indicates that the user is in-session (or out-of-session if that option is selected). The drop-down list lets you select a detection pattern from candidates that AppScan has identified in the Login recording, and the green or red message below the pattern indicates whether the current pattern is valid or invalid. Note: It is usually preferable to use an <i>in-session</i> pattern. However, in rare cases where the in-session pattern is not always returned following an in-session request, or where it is complicated to define, you can use an <i>out-of-session</i> pattern instead. If AppScan was unable to identify any valid, or if you need to select a different one, use the Advanced pattern selection button (next row in this table). RegExp: Select this check box to enter a regular expression for identifying the pattern.

Table 1. "Review & Validate tab" settings (continued)

Setting	Details
Advanced pattern selection button	(Active only when an In-Session Detection Request is selected:) This button opens the Select Detection Pattern dialog box, showing the content of in-session and out-of-session responses to requests in the Login sequence you recorded (based on the selected detection pattern). It lets you see the selected detection pattern in the context of the response, and define a detection pattern that is <i>not</i> listed in the combo box. The dialog lets you toggle through all recorded responses. In the upper part of the box you can also see the in-session and out-of-session requests that AppScan sent.
<i>Validation</i>	
Validate button	(Active only if the current login sequence has not been verified yet:) Click to validate the sequence and the session detection pattern.
Key icon	The key icon indicates In-Session Detection configuration status:  Enabled and configured. (An in-session page has been identified in login sequence, either automatically or by the user.)  Enabled but not fully configured.  Enabled but not configuration failed.  Disabled. See "Select Detection Pattern dialog box" on page 53 for details.

Edit Action-Based Login dialog box:

Dialog that opens from **Config > Login > Review & Validate > (Action-Based) Edit**, can be used to troubleshoot the login procedure if validation fails.

Even if Request-Based login succeeds, it is good to troubleshoot the Action-Based Login if possible. In this dialog box you can:

- Click **Replay**, to replay the login in the selected browser.
- Change the selected browser and **Replay** in a different browser.
- Click **Select > Wait Actions**, and increase the wait time between requests.
- Review the "Type" of specific requests in the procedure and change them if needed.
- Click **Edit playback actions** to view and edit in XML format.
- Increase the timeout for a single login attempt.

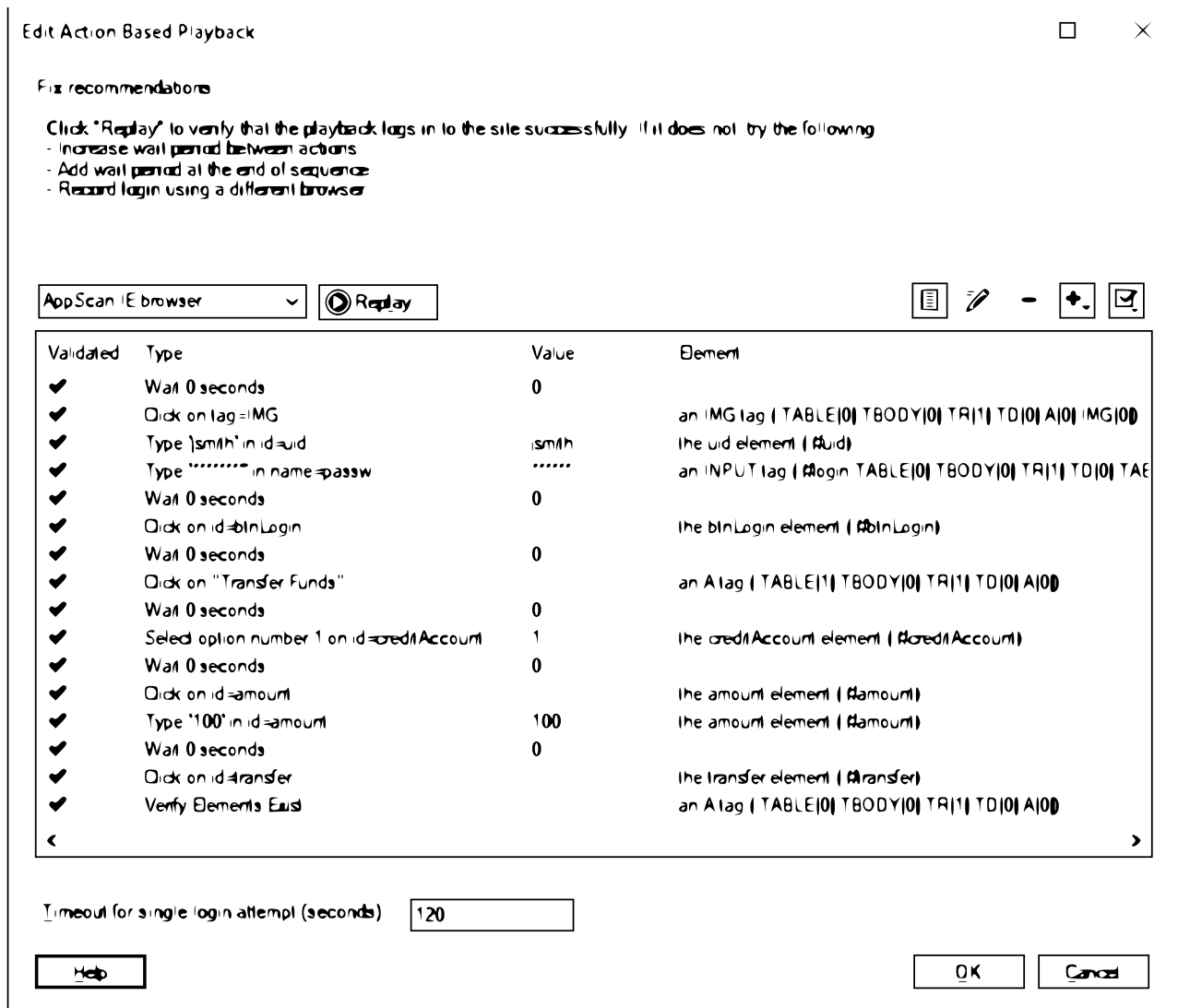


Figure 1. Edit Action-Based Playback dialog box

Table 2. "Edit Action-Based Playback" settings

Item	Description
Browser field	Shows the currently selected browser for login playback. You can change this from the drop-down list.
Replay button	Replays the recorded login procedure in the selected browser.

Table 2. "Edit Action-Based Playback" settings (continued)

Item	Description
List of requests, and wait times between requests.	Shows the following for each action: Validated: Green check mark for actions successfully performed; red X for unsuccessful actions Type: Description of the action (such as "wait", "click" or "set" (value) Value: For wait actions: time in seconds; for user actions: name of the action Element: HTML element on which the action is performed For any selected action the icons above the table let you:  View and edit playback actions as XML  : Edit action value  : Delete action  : Add a wait time between actions  : Add a "wait" action before or after the selected action
Timeout for single login attempt (minutes)	If your login procedure needs more time you can increase this setting.

Edit Request-Based Login dialog box:

Dialog that opens from **Config > Login > Review & Validate > (Request-Based) Edit**, can be used to troubleshoot the login procedure if validation fails.

This is a simplified version of the Advanced In-Session Request Selection dialog.

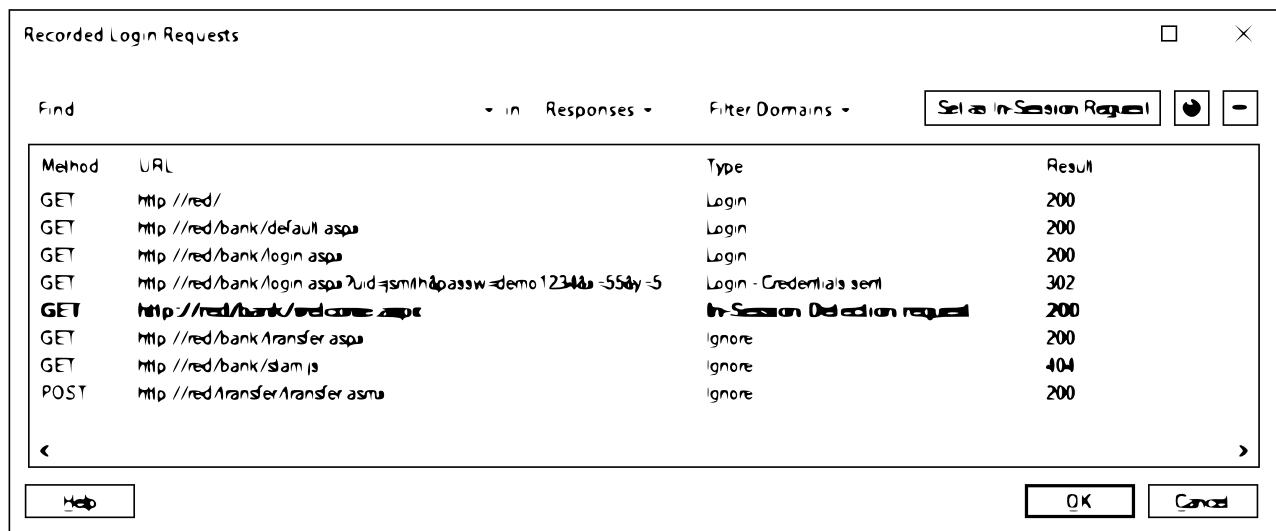


Figure 2. Edit Request-Based Playback dialog box

Table 3. "Recorded Login Requests" settings

Setting	Details
Main list	Shows all requests the recorded login procedure.
Find	Show only requests that contain the text string you enter, in URL, Request, Response or All.
Filter Domains	Show only requests from domains selected in the drop-down list.Click for AppScan to perform the following actions:
Set as In-Session Request button	Sets the selected request as the In-Session Request, that will be used by AppScan during the scan to verify that it is still logged in. You can also do this by right-clicking on a request in the list.
	Show the response received to the selected request when the login was recorded. The window that opens has two tabs: The Browser tab shows the response received, and the Request/Response tab shows the raw data for both the request and the response.
	Delete the selected request from the login sequence.

Advanced In-Session Request Selection dialog:

Advanced In-Session Request Selection dialog box, that opens from **Config > Login > Review & Validate > Advanced Request Selection**

This is a version of the Edit Request-Based Login dialog box with more options. In this dialog box you can:

- See the sequence of requests you sent when logging in
- See the In-Session Detection Request

Note: The page marked "In-Session" should be the first page to be highlighted. If an earlier, "Login", page is highlighted then either the in-session pattern is wrong, or the wrong page is marked as "In-Session".

- View any URL in the sequence in a browser.
- Set a *different* request as the In-Session Request, and select a new In-Session Detection Pattern from this new request.
- Delete unnecessary requests *before* the "In-session" URL, to save AppScan repeating these unnecessary requests many times during a scan
- See requests sent *after* the In-Session Detection Request, that contain the In-Session Detection Pattern and are marked "Ignore"
- Search the requests in the sequence
- Show only requests from specific domains
- Open the Select Detection Pattern dialog box to select a pattern not suggested by AppScan

Table 4. "Advanced In-Session Request Selection" settings

Setting	Details
Main list	Shows all requests the recorded login procedure.
Find	Show only requests that contain the text string you enter, in URL, Request, Response or All.
Show Domains	Show only requests from domains selected in the drop-down list.Click for AppScan to perform the following actions:
Set as In-Session Request button	Sets the selected request as the In-Session Request, that will be used by AppScan during the scan to verify that it is still logged in. You can also do this by right-clicking on a request in the list.

Table 4. "Advanced In-Session Request Selection" settings (continued)

Setting	Details
Advanced pattern selection button	Opens the Select Detection Pattern dialog box, showing the content of in-session and out-of-session responses to requests in the Login sequence you recorded (based on the selected detection pattern). It lets you see the selected detection pattern in the context of the response, and define a detection pattern that is <i>not</i> listed in the combo box. The dialog lets you toggle through all recorded responses. In the upper part of the box you can also see the in-session and out-of-session requests that AppScan sent. You can also do this by right-clicking on a request in the list.
	Show the response received to the selected request when the login was recorded. The window that opens has two tabs: The Browser tab shows the response received, and the Request/Response tab shows the raw data for both the request and the response.
	Delete the selected request from the login sequence.
Detection Pattern	This field shows a pattern found in the selected In-Session Detection Request, which indicates that the user is in-session (or out-of-session if that option is selected). The drop-down list lets you select a detection pattern from candidates that AppScan has identified in the Login recording, and the green or red shading indicates whether the pattern is valid or invalid. Note: It is usually preferable to use an <i>in-session</i> pattern. However, in rare cases where the in-session pattern is not always returned following an in-session request, or where it is complicated to define, you can use an <i>out-of-session</i> pattern instead..If AppScan was unable to identify any valid pattern, or if you need to select a different one, use the Advanced pattern selection button to select your own.

Select Detection Pattern dialog box:

Use this dialog to compare in-session and out-of-session responses to login requests, to help you decide on the detection pattern that is best for your application. Opened by clicking

About this task

This dialog box opens when you click **Config > Login Management > Review & Validate > Advanced pattern selection**.

In this dialog you can see:

- Current session detection pattern, and its status
- Four panes showing the current request and response when in-session and out-of-session
- Difference between the requests are highlighted in green
- In the In-Session Response the selected pattern is highlighted in green
- **Save pattern** button lets you select a different pattern and set it
- Page toggle at the top right of the dialog lets you toggle between requests

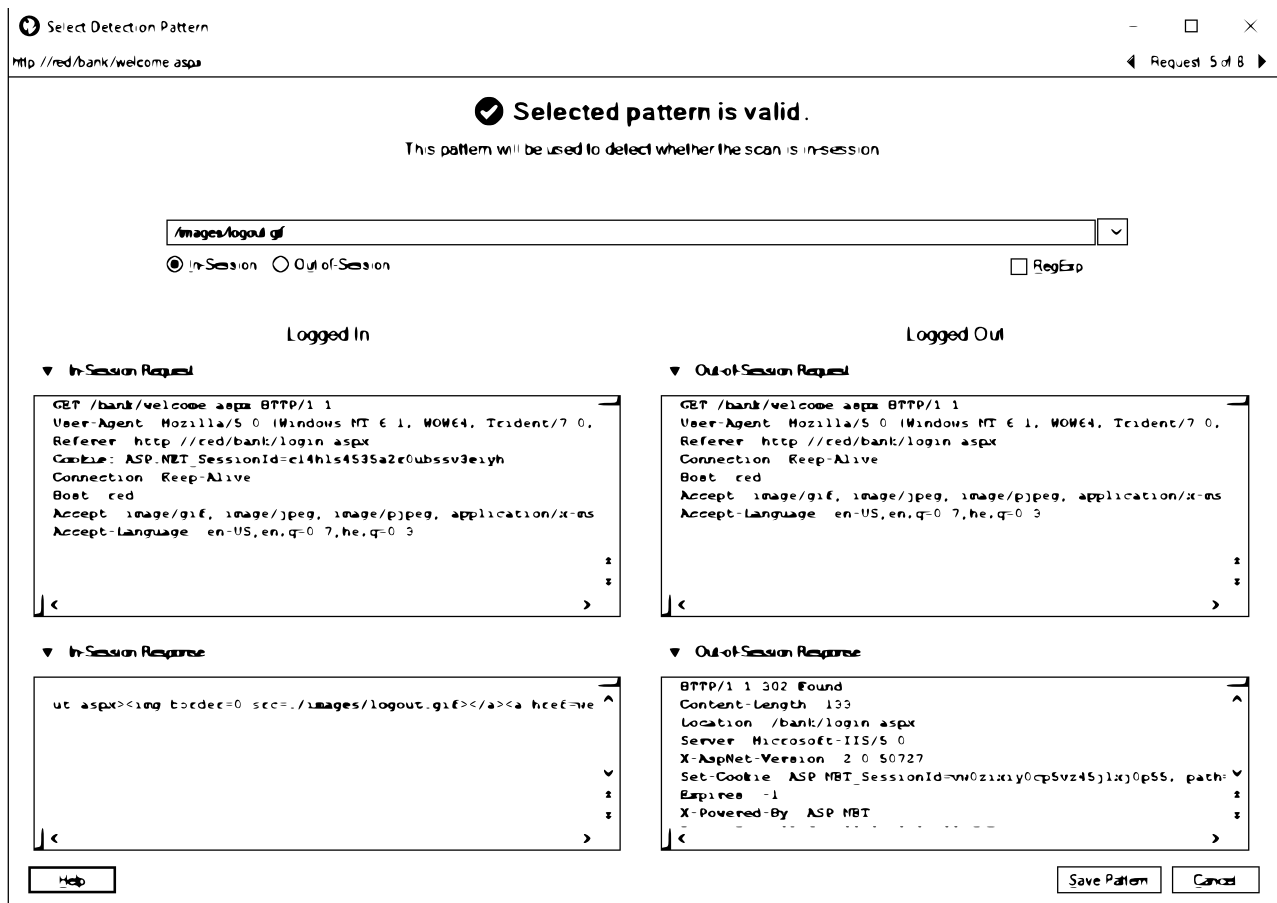


Figure 3. Select Detection Pattern dialog box

Procedure

1. Review the selected pattern where it occurs within the In-Session (or Out-of-Session) response.
2. View and compare the two responses to find other valid patterns.
3. View and compare the In-Session and Out-of-Session *requests* (above the response fields) that generated these responses.
4. Toggle through the requests and responses for *all* requests in the recorded login, using the arrows in the upper right corner of the dialog box.
5. Select and save a new pattern of your choice.

Optimizing In-Session Detection:

Reviewing the login sequence can help troubleshoot and optimize in-session detection.

About this task

AppScan attempts to identify automatically an "In-Session Detection Pattern" that occurs on the in-session page, that it can use during the scan to verify that it is still logged in. This should be a pattern that occurs in the page's response *only* when you are logged in. An example might be text that reads: "Click here to log out".

During the scan, AppScan sends the in-session request repeatedly, and checks that the response contains the In-Session Detection Pattern. If AppScan does not find the pattern in the page's response, AppScan assumes it has been logged out, and attempts to log in again by replaying the login sequence. It follows

that the login sequence is typically played many times during a scan. It is therefore best that it contains as few steps as possible. It is also helpful if the In-Session page is a small page, and does not contain tracked parameters or cookies, since these can also increase scan time significantly.

When the defined in-session pattern is detected in the in-session request, (the request immediately following the POST request), it is highlighted in green.

Procedure

1. Verify that the In-Session Detection Pattern that was automatically selected is in fact an indication that the user is logged in. If necessary, change it.
2. Verify that there are no unnecessary steps in the login procedure. If there are, delete them.
3. Verify that the in-session response is not a large one, and - if possible - does not include tracked parameters or cookies. If necessary, add one or more steps till you reach a smaller page or one without tracked items.
4. If you succeeded in selecting an in-session page without tracked parameters or cookies, there is no need for AppScan to check for these each time it logs in. Go to **Advanced Configuration > Session Management: Parse In-Session Page**, and change the setting to False.
5. If none of these succeed, you can try to identify an out-of-session pattern instead, and then change the detection method.

Session IDs tab

Scan Configuration > Login Management > Session IDs tab.

The **Session IDs** tab of **Login Management** view is used to review and manage the tracking of variables (session IDs) received during the recorded login.

The tab lists all variables received during the login sequence, and automatically designates Session IDs as Tracked (a check-mark appears next to the variable in the Tracked column).

You can select variables in this list and use the **Track** and **Stop Tracking** buttons to change their status. All "tracked" parameters are added to the global list of parameters and cookies whose status is tracked during scanning (see "Parameters and Cookies view" on page 67).

Tip: If you are not sure which items are session identifiers and you're scan is going out-of-session, contact your application's developers to see if they can supply a list of parameters and cookies the application uses to maintain session.

Advanced tab

Scan Configuration > Login Management > Advanced tab.

The **Advanced** tab of **Login Management** view is used to advanced login settings and logout page detection.

Setting	Details
Advanced Login Settings	Allow login even if the application is already logged in: To save scan time, AppScan sends multiple login requests without logging out in between. Deselect this check box only if your application does not allow this. Number of failed login attempts before user is locked out: If your application will lock a user out after a certain number of failed login attempts, select this check box and configure the number. AppScan Enterprise will send valid login requests between failed requests to ensure this threshold is never reached, as further scanning would then be impossible.

Setting	Details
Logout Page Detection	AppScan uses a regexp to identify log out pages. This helps it to scan more efficiently by trying to avoid getting logged out too often, and having to log in again. It is also used to identify logout pages when you configure the scan to not test login/logout pages (see “Test Options view” on page 100), and to logout when needed as a part of some security tests. This is the default regexp: <pre>(logout signout logoff signoff exit quit invalidate)</pre> If any of the indicators in this regexp appear in the URL, AppScan assumes the page is a logout page, and therefore that it is currently logged in to the application. Note: AppScan may add to this expression when you record a Login procedure, if it identifies additional indicators. You can add further indicators as necessary but be sure to follow the regular expression syntax rules. Note: The Expression Test PowerTool (Tools > Expression Test) can be useful to verify the syntax of your regular expressions. If you need additional help you may find the following link useful: http://www.regular-expressions.info/quickstart.html

Environment Definition view

Environment Definition view of the Configuration dialog box.

Environment definition is not essential, but enables AppScan to safely refrain from sending non-relevant tests during the scan, resulting in a faster and more accurate scan.

Note: Where relevant, some of the list boxes allow you to select more than one option by pressing the **Ctrl** key when selecting items in a list.

Metric	Comments
Operating System	Operating System of application being scanned.
Web Server	Select <i>all</i> applicable answers. To select more than one option use [Ctrl] + Click.
Application Server (if any)	Select <i>all</i> applicable answers. To select more than one option use [Ctrl] + Click.
Type of Database (if any)	Select <i>all</i> applicable answers. To select more than one option use [Ctrl] + Click.
Third-Party Component (if any)	Select <i>all</i> applicable answers. To select more than one option use [Ctrl] + Click.
Location of Site	Whether the site is remote or local.
Type of Site	Whether it is a test site or a live production site.
Deployment Method	Whether the site is deployed internally (private site), or externally (on the Internet).
Collateral Damage Potential	The potential for damage or theft if the application is vulnerable.
Target Distribution	The proportion of systems in the environment that are potential targets.
Availability Requirement	The relative importance of availability (of information).
Confidentiality Requirement	The relative importance of confidentiality (of user information).
Integrity Requirement	The relative importance of integrity (accuracy) of information.

Note: The last five items are the Environmental CVSS metrics for the site. If you define the relative importance of these metrics in your application environment, AppScan will take these definitions into

account when assigning severity values to vulnerabilities it finds during the scan. These definitions are global. (You can adjust the Environmental Metrics for a *specific* issue from the Detail Pane > Severity > Open CVSS Panel, see “CVSS settings” on page 192.)

Exclude Paths and Files view

Exclude Paths and Files view of the Configuration dialog box.

You can configure AppScan to ignore certain paths in the application, or specific types of file. However, you should apply exclusions with caution, as they may have important issues. Changes here apply only to request-based exploring, not action-based exploring (as configured in Config > Explore Options > Explore method).

Setting	Details
Exclude Paths	You can filter the scope of the automatic Explore stage by adding URLs (full paths, which may include queries) or Regular Expressions to the Exclude or Include Paths list. For details see “Exclude Paths”
Exclude File Types	You can configure AppScan to ignore specific types of file during the scan. For example, you could make the scan run faster if you exclude graphics files. However, you should exclude files with caution, as they may contain important issues. For details see “Excluding file types” on page 61

“Limiting scan to the Starting URL folder” on page 41

Exclude Paths

Exclude paths and files view of the Configuration dialog box.

You can filter the scope of the automatic Explore stage by adding URLs (full paths, which may include queries) or Regular Expressions to the **Exclude Paths** list. Possible reasons for doing this may be:

- Because they are still under development and you know they are problematic and do not want to scan them right now
- Because you know they are *not* problematic and you want to reduce scan time
- To reduce scan time by restricting the scan to certain parts of the application

For any path defined, you can optionally limit the filter to one or more specific parameters. A possible reason for doing this may be:

- For megascript applications (applications contained in a URL, and controlled by its parameters) where you want to exclude certain parameters (such as the login or log out parameter) from scans.

There are two types of entry in this table:

- **Exclude:** Listed paths are excluded from the scan.

Any link that matches a path configured to be an exclusion is filtered out of the scan.

Note: You can also exclude a path by right-clicking on it in the Application Tree, and selecting **Exclude from Scan**.

- **Exception:** Use to include a specific directory within a path that was excluded *higher up in the list*.

Note: The **Exception** function is only needed if you want to include a directory that is within a path that you have excluded. For example, if you have excluded: `http://demo.testfire.net/bank` you could then add `http://demo.testfire.net/bank/transfer.aspx` as an Inclusion, lower down in the list, to include that subdirectory in scans.

Note: If you add Exclusions between the Explore and Test stages of scanning, AppScan will not test the excluded paths even though they were explored.

“Adding new exclusions or exceptions”

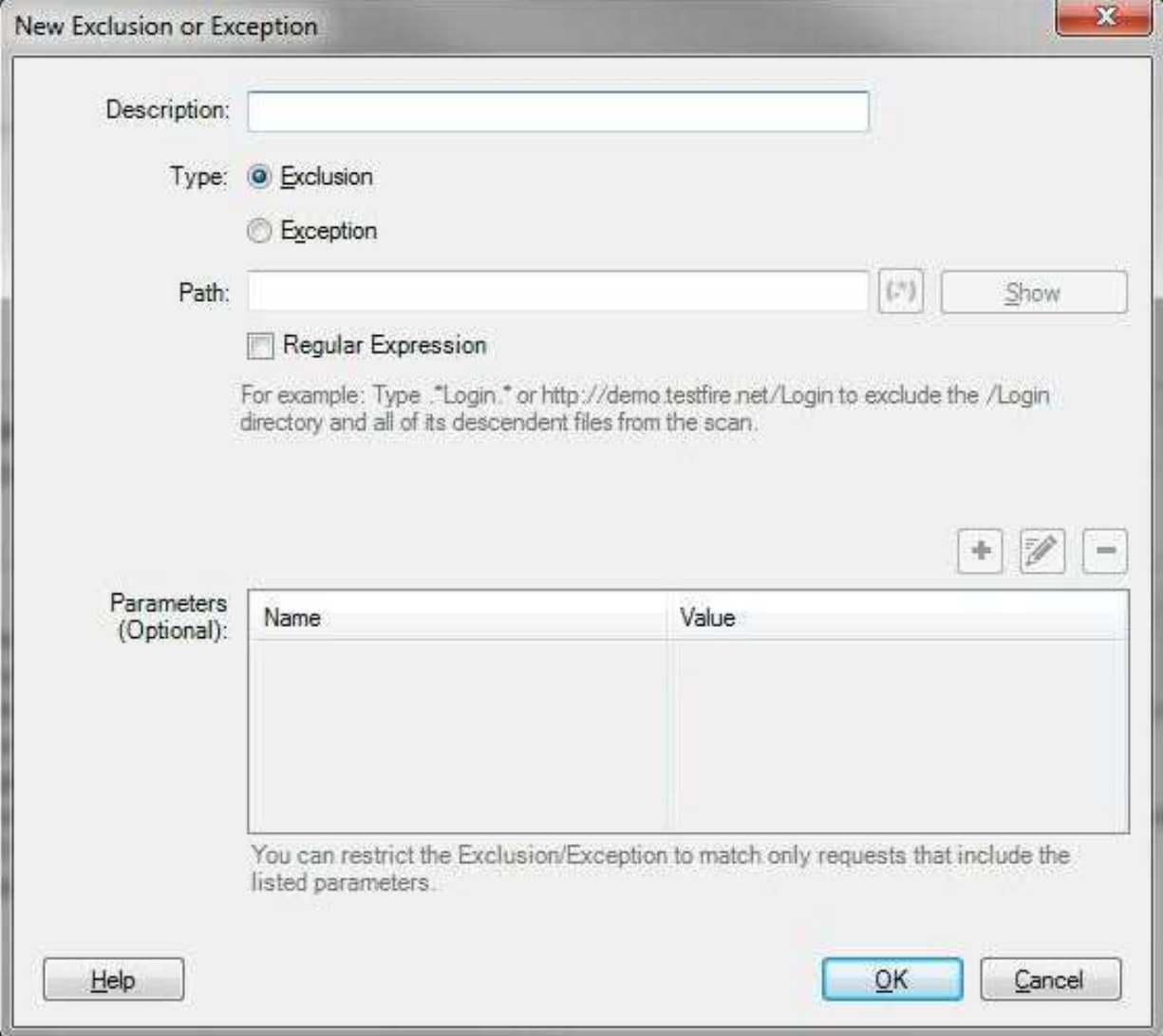
Adding new exclusions or exceptions: Procedure

1. In the Exclude Paths area of the Exclude Paths and Files tab of the Configuration dialog box, click



, select Exclusion or Exception, enter the full path or Regexp., and click **OK**. (In the **Scan Configuration Wizard > Starting URL > Advanced**, click **Explore**.)

The **New Exclusion or Exception** dialog box opens.



The dialog box titled "New Exclusion or Exception" contains the following elements:

- Description:** A text input field.
- Type:** Two radio buttons: **Exclusion** (selected) and **Exception**.
- Path:** A text input field followed by a button with a wildcard symbol (*) and a **Show** button.
- Regular Expression:** A checkbox.
- Example:** Text stating: "For example: Type *.Login.* or http://demo.testfire.net/Login to exclude the /Login directory and all of its descendent files from the scan."
- Parameters (Optional):** A table with two columns: **Name** and **Value**.

Name	Value
------	-------
- Buttons:** **Help**, **OK**, and **Cancel** at the bottom.
- Icons:** A set of three icons (+, edit, -) to the right of the parameters table.

2. Select the radio button for the filter you want to create:
 - **Exclusion:** Do not scan URLs matching this item.
 - **Exception:** Include URLs matching this item even though they are excluded by an Exclusion higher up on the list.

Note: The **Exception** function is only needed if you want to include a directory that is within a path that you have excluded. For example, if you have excluded: `http://demo.testfire.net/bank` you could then add `http://demo.testfire.net/bank/transfer.aspx` as an Inclusion, lower down in the list, to include that subdirectory in scans.

Note: If you add Exclusions between the Explore and Test stages of scanning, AppScan will not test the excluded paths even though they were explored.

3. Optionally add a description that will appear in the Exclude Paths list.
4. Type a path or regular expression that will match a set of directories into the Path field (see examples in table following), and for a regular expression select the check box.

Note: A regular expression (regexp.) is a string that describes a set of strings, according to certain syntax rules. Click  to open the Expression Test PowerTool, which can help you verify the syntax of your regular expressions.

If you need additional help writing regular expressions you may find the following link useful:
<http://www.regular-expressions.info/quickstart.html>

5. To apply the exclusion or exception to paths that include specific parameters only, click  to add one or more parameters to the lower pane of the dialog box.

Note: This feature is designed for "megascript" applications, where the entire application is contained in a URL and controlled by its parameters. Filtering out the URL would disable the scan, but you can filter out specific parameters, or even specific parameter values (such as those that login or logout).

6. Click **OK**.

The new item is added to the bottom of the list.

Note: Where there is a conflict between two items in the list, the *lower* item takes priority. Use the **Up/Down** buttons to adjust the order of items as required. If an exclusion or inclusion is made redundant due to another exclusion or inclusion higher in the list, the redundant item will be deleted from the list when you click OK.

Example

Type	Example and Function
Exclude	http://demo.testfire.net/transfer or http://demo.testfire.net/transfer/ Filters out the specified URL and all sub-directories and files
Exclude	.*private.* Exclude all URLs containing the string private
Exclude	.*_bk.aspx Exclude all URLs ending in _bk.aspx
Exception	http://demo.testfire.net/transfer/customize.aspx When an earlier Exclusion (such as the first one in this table) excludes sub-directories and files, this Exception includes this particular path in the scan. Note that the Exception must appear <i>below</i> the Exclusion to take effect

Editing exclusions or inclusions:

Procedure

1. Select an item in the **Exclude or Include Paths** list.
2. Click **Edit**.

The **Edit Exclusion or Inclusion** dialog box appears, displaying the properties of the selected item.

3. Change as required and click **OK**.

What to do next

See also: "Limiting scan to a specific folder"

"Limiting scan to the Starting URL folder" on page 41

Limiting scan to a specific folder

Using Exclusions and Exceptions to limit the range of a scan.

About this task

You can limit automatic scanning to a *specific* folder or folders by excluding all URLs, and then including the directory or directories required. (To limit the scan to the *Starting URL folder*, see "Limiting scan to the Starting URL folder" on page 41.)

Procedure

1. Open **Scan Configuration > Exclude Paths and Folders**.
2. Add an **Exclude** item with path of your site, such as **http://www.mysite.com/** (see "Exclude Paths" on page 57).
3. Add one or more **Exception** items *below* the **Exclude** item, with the path(s) you want to scan.

Note: The **Exception** item(s) must appear *below* the **Exclude** item in order to take effect. If necessary, use the **Up/Down** buttons to adjust the order before closing.

4. Verify that the starting URL (see "URL and Servers view" on page 40) is not within one of the excluded paths. (If it is the scan will not be able to start!)
5. Click **OK** to save changes and close the list.

Example 1:

About this task

Let's consider a case where the Starting URL for the scan (see "URL and Servers view" on page 40) is: **http://www.mysite.com/index.aspx** and we want to restrict the scan to the folder: **http://www.mysite.com/myfolder/**

Procedure

1. Add an **Exclude** item with path **http://www.mysite.com/** (see "Exclude Paths" on page 57).
2. Add an **Include** item *below* the **Exclude** item, with the path of the Starting URL: **http://www.mysite.com/index.aspx**
3. Add a *second* **Include** item below the **Exclude** item, with the path of the folder to be scanned: **http://www.mysite.com/myfolder/**

Example 2:

About this task

Let's now consider a case where the Starting URL for the scan is within the folder that is to be scanned. The Starting URL is: `http://www.mysite.com/myfolder/index.aspx` and we want to restrict the scan to the folder: `http://www.mysite.com/myfolder/`

In this case the folder to be scanned and the Starting URL are added to the scan in a single **Include** item.

Procedure

1. Add an **Exclude** item with path `http://www.mysite.com/`
2. Add an **Include** item *below* the **Exclude** item, with the path of the folder to be scanned: `http://www.mysite.com/myfolder/`

Excluding file types

Excluding certain types of files from the scan.

About this task

You can configure AppScan to ignore specific types of files, rather than complete pages, as in the Exclude Paths settings. For example, you could make the scan run faster if you exclude graphics. However, you should exclude files with caution, as they may contain important issues.

Procedure

1. Open the **Scan Configuration** dialog box > **Explore Settings** view.
2. In the **Exclude File Types** pane, make sure the check boxes of the file types that you do not want to be scanned are selected.

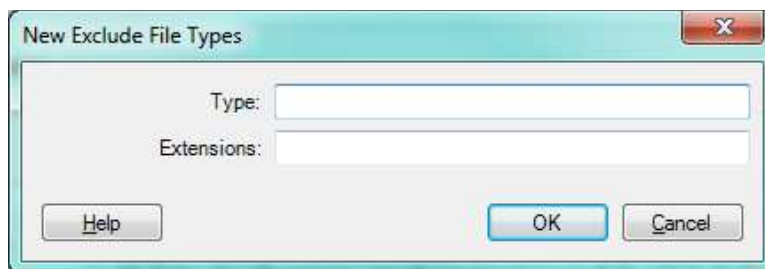
Excluding additional file types:

About this task

If you have a file extension in your application that you cannot find listed in the **Exclude File Types** dialog box, you can add these extensions to the list.

Procedure

1. In the **Exclude File Types** area, click 
The **New Exclude File Types** dialog box appears.



2. In the **Type** text box, enter a word or phrase to describe a set of file extensions.
3. In the **Extensions** text box, enter the list of extensions that fall under this type of file. (Separate multiple extensions with a comma, without a space.)
4. Click **OK**.

Editing the list of excluded file types:

Procedure

1. Select a row in the **Exclude File Types** dialog box.
2. Click **Edit**.

The **Edit Exclude File Type** dialog box appears, displaying the associated extensions of the selected exclusion.

3. Change the type name or extensions list as required
4. Click **OK**.

Explore Options view

Explore Options view of the Configuration dialog box.

Explore Options view has these tabs:

- The Main tab is used to select the method AppScan uses to explore the site.
- The Action-Based tab is used to configure settings specific to action-based exploring.
- The Request-Based tab is used to configure settings specific to request-based exploring.

Main tab

Scan Configuration > Explore Options > Main tab.

In this tab you select the Explore method AppScan will use for the scan, and configure options that apply to both methods.

Explore Method

AppScan uses two distinct methods for the Explore stage of the scan. You can select either one, or both. Of the two methods, Request-based Explore is usually faster than Action-based Explore. When both are selected (default, and recommended), Action-Based Explore runs first, with a 30 minute time limit, followed by Request-Based Explore.

Page Structure (DOM) Filtering

These can *greatly* reduce scan time by identifying pages that are similar enough to pages already scanned, that they can safely be ignored.

Scan Limits

These determine how deeply (or how quickly) AppScan explores your application.

Other Settings

These are for configuring the client to recognize a specific server encoding and to send a specific user-agent header.

Setting	Details
<i>Explore Method</i>	
Action-Based	A version of the Google Chrome browser is used to scan the site, as a user would, clicking the links that are visible in the browser. This method is particularly effective where new technologies such as JavaScript and Session Storage are used, and for sites that are RIA, Single-page Application (SPA), or AngularJS.
Request-Based	Requests are sent based on all page content that AppScan discovers. This includes content that is not visible to users using a browser, such as links in comments, which an attacker would find.
<i>Page Structure (DOM) Filtering</i>	

Setting	Details
Filter similar pages based on structure (DOM)	AppScan compares new pages with those already scanned, for structural (DOM) similarity, which indicates the new page contains no new links or content that require additional testing. For example, on a commercial site there may be a catalog with individual pages for a thousand different items, that are in all other ways identical. There is usually no need to scan all those pages. Filtering based on DOM similarity can <i>greatly</i> reduce scan time. By default both check boxes are selected. After the scan you should examine the Filtered tab of the scan results to see whether unique requests were mistakenly filtered out of the scan. If this happened you should try the "Filter less pages" option, which maintains a steady, lower level of filtering, or disable DOM filtering altogether. Three kinds of filtered items will be found in the Filtered tab of the results: • Similar DOM: This indicates a page that was filtered from the scan because its structure (DOM) is similar to that of a previously explored page, and probably contains no new elements to test. • Likely Similar DOM: This indicates a request that was not sent at all, because AppScan estimates that the response will have the same structure (DOM) as that of a previously explored page, and will contain no new elements to test. • Similar Body: This indicates a request (from a page that was <i>not</i> filtered due to Similar DOM) that was filtered from the scan because its response body content is similar to that of a request that was previously explored. After the scan you should examine the Filtered tab of the scan results to see whether unique requests were mistakenly filtered out of the scan. If this happened you should clear the "Filter likely duplicate pages" option (next option), or disable DOM filtering altogether by clearing this check box.
Filter pages that are likely to be similar based on structure (DOM)	This setting filters "Likely similar DOM" pages from the scan (see description above). If unique requests are mistakenly filtered out of the scan you should clear this check box.
<i>Scan Limits</i>	
Redundant Path Limit	AppScan will not access the same path more than the specified number of times. A particular path may be visited several times if it appears with different parameters. This limit is relevant mainly for scripts. It is deselected by default, as in most cases selecting the check box above, Filter duplicate pages based on structure (DOM), will sufficiently control scan time.
Click Depth Limit	AppScan will not scan pages that are accessed by clicking more than the specified number of links.
Total Page Limit	If selected, AppScan will access no more than the maximum number of pages defined. Note that there may be many URLs explored per page.
<i>Other Settings</i>	
Encoding	AppScan generally detects the application's encoding method automatically, and therefore Autodetect is selected by default. If the content of responses in the scan Results looks distorted, this may mean that the encoding method was not correctly identified. To solve this problem, select the correct encoding method from the drop-down list.

Setting	Details
User-Agent	The user-agent header in an HTTP request tells the server what kind of client sent the request, and this may affect the content that the server returns. For example, there may be content that is specific to mobile phones that is sent only when the user-agent is a mobile phone browser. In order for AppScan to be able to test such content, you need to configure it to send the appropriate user-agent header. AppScan generally detects the user agent automatically, and therefore Autodetect is selected by default. However, if you use a browser other than the built in browser, and you do not record a login procedure, a multi-step operation, or a manual explore, AppScan will be unable to autodetect the user agent, and you must select it manually. To change the user-agent, select an agent from the drop-down list. To enter custom content, click the Edit button and type in the content. When you close the dialog box the button name changes to Custom User Agent. For more details, see "User-agent headers." Note: If you change the default browser, refer to the conditions listed in "Changing the default browser" on page 304

User-agent headers:

This section describes the user-agent header sent for each user-agent, and how to work with "Auto-Detect".

If necessary, you can edit the content of any user-agent header by clicking the Edit button. The updated user-agent will then appear as "Custom User Agent".

You can select a header by clicking the User-Agent button, and edit the content by clicking the Edit button. Once you edit the content, the button name changes to "Custom User Agent". Custom User Agent content is only saved with the scan (and transferred to other computers along with the scan) if it is the selected user-agent when the scan is saved.

The table below shows the user-agent header sent for each user-agent.

User-Agent	Header
Auto Detect	This is the default setting. If you record a manual explore, AppScan will automatically detect the user-agent. If there is no manual explore, AppScan will use the Internet Explorer user-agent.
Custom User Agent	If you edit the content of the selected user-agent, the setting is saved as a "Custom User Agent". Important: Custom User Agent content is only saved with the scan (and transferred to other computers along with the scan) if it is the selected user-agent when the scan is saved.
<i>Desktop browsers</i>	
Internet Explorer	Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Firefox	Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.1
Chrome	Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2228.0 Safari/537.36
Microsoft Edge	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/42.0.2311.135 Safari/537.36 Edge/12.246
Opera	Opera/9.80 (Windows NT 6.1; U; es-ES) Presto/2.9.181 Version/12.00

User-Agent	Header
Safari	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_1) AppleWebKit/601.2.4 (KHTML, like Gecko) Version/9.0.1 Safari/601.2.4
<i>Mobile browsers</i>	
iPhone (Safari)	Mozilla/5.0 (iPhone; CPU iPhone OS 6_1_4 like Mac OS X) AppleWebKit/536.26 (KHTML, like Gecko) Version/6.0 Mobile/10B350 Safari/8536.25
iPad (Safari)	Mozilla/5.0 (iPad; U; CPU OS 4_3_5 like Mac OS X; nl-nl) AppleWebKit/533.17.9 (KHTML, like Gecko) Version/5.0.2 Mobile/8L1 Safari/6533.18.5
Google Android	Mozilla/5.0 (Linux; Android 5.1.1; A0001 Build/LMY48B) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/49.0.2623.105 Mobile Safari/537.36
Blackberry	Mozilla/5.0 (BB10; Kbd) AppleWebKit/537.10+ (KHTML, like Gecko) Version/10.1.0.4633 Mobile Safari/537.10+
Opera Mini	Opera/9.80 (Series 60; Opera Mini/7.1.32444/35.5706; U; en) Presto/2.8.119 Version/11.10
Opera Mobile	Opera/12.02 (Android 4.1; Linux; Opera Mobi/ADR-1111101157; U; en-US) Presto/2.9.201 Version/12.02
Microsoft IE 7 for mobile	Mozilla/4.0 (compatible; MSIE 7.0; Windows Phone OS 7.0; Trident/3.1; IEMobile/7.0) Asus;Galaxy6

Action-Based tab

Scan Configuration > Explore Options > Action-Based tab.

In this tab you configure settings that affect Action-Based exploring. This tab is active only if the Action-Based Explore Method is selected in the Main tab.

Setting	Details
<i>General</i>	
Explore Timeout (minutes)	The default time limit for Action-Based Explore of a site is 30 minutes, after this time the Explore stage stops even if the site has not been fully covered. If AppScan misses significant parts of the site in this time, you can increase this timeout.
Minimum wait before invoking actions on page (milliseconds)	AppScan attempts to identify that a page has loaded completely before it starts exploring it. If you add a minimum wait period here, AppScan always uses <i>this</i> setting as the minimum wait period (even if it detects that the page has loaded), but will wait more than this time if it detects that the page has <i>not</i> loaded. Tip: If when reviewing the Explore data, you see that AppScan failed to execute all the possible actions on a page, this can indicate that its dynamic wait time was too short. You can also see this during the scan if you enable the browser: 1. Go to Tools > Options > Advanced 2. Locate <code>SessionManagement:ShowActionBasedPlayerWindow</code>, and change its setting to True. 3. Run a scan. The browser will open when scanning, and you can watch as AppScan explores your site. If you notice that it continues to a new page before the current page is fully loaded, increasing the wait time may solve the problem. Note: Changing this setting might affect Explore time, so you may also want to consider increasing the Explore timeout (above).

Setting	Details
Auto-detect dynamic-page loading	By default, AppScan actively detects dynamic page content and treats the page as such. In rare cases this may prevent the page from loading correctly, and therefore affect scan coverage. Tip: To identify this issue: 1. Go to Tools > Options > Advanced 2. Locate <code>SessionManagement:ShowActionBasedPlayerWindow</code>, and change its setting to True. 3. Run a scan. The browser will open when scanning, and you can watch as AppScan explores your site. If you notice that pages which load successfully in a regular browser do not load correctly during the scan, clearing this check box may solve the problem.
<i>Filters</i>	
Skip actions on identical DOM elements	AppScan identifies actions that it has already executed on a previous page based on various criteria. If your site includes <i>different</i> actions that might seem identical due to their DOM element, AppScan might incorrectly ignore them. If this happens, clear this check box. Note: AppScan <i>does</i> actually repeat identical actions a few times, to make verify that they really are the same, before deciding to ignore future iterations.
Analyze and skip redundant actions	When AppScan identifies actions that seem similar, it compares their resulting pages. If after a few iterations the results all seem similar, AppScan ignores future similar actions as redundant. If your site includes many pages whose only difference is their content, such as a news site, this feature can drastically reduce scan time, and you should select the check box.
Actions to skip	This is a list of actions for AppScan to ignore as they could adversely affect the scan, or even the application. Actions to be skipped are identified based on the Id, name or ng-model attributes of the DOM element for the action. Any action whose DOM element attributes' Id, name or ng-model contain one of the words in the list, will be filtered out of the scan. You can Add to, Edit , and Delete items in this list.

Request-Based tab

Scan Configuration > Explore Options > Request-Based tab.

In this tab you configure settings that affect Request-Based Explore. This tab is active only if the Request-Based Explore Method is selected in the Main tab.

- **JavaScript** and **Flash** options determine whether AppScan should ignore or scan these scripts.
- **Explore Mode** determines whether AppScan explores all links on a page before continuing to the next page, or explores each new link as it is found.
- **WebSphere Portal** are for configuring the client to recognize a specific server encoding and to send a specific user-agent header.
- **Flash**

Setting	Details
<i>JavaScript</i>	
Parse JavaScript code to discover URLs	AppScan will parse JavaScript code as text data to collect links.
Execute JavaScript to discover URLs and dynamic content	AppScan will actually execute JavaScript code and analyze the results to collect links, including dynamic links that may not be discovered by parsing alone. (This utilizes more system resources than parsing.)
Execute JavaScript when replaying login	If the application's login page uses JavaScript code, this check box must be selected in order for AppScan to be able to login during scanning.
<i>Explore Mode</i>	

Setting	Details
Breadth First	(Default) AppScan explores page by page, exploring all links on one page before continuing to the next. It is recommended that you do not change the default selection of this option (Breadth First), unless you are aware of limitations in your application that demand that a user visits links in a specific order.
Depth First	AppScan explores link by link, exploring each new link as it is found. If you change the Explore Method to Depth First, you must also change AppScan to use only one thread during the Explore (in Configuration > Communication and Proxy view).
<i>WebSphere® Portal</i>	
Enable WebSphere Portal scanning	If the site is a WebSphere Portal site, AppScan will need to get URL decoding information from the site for more efficient scanning and to build a useful application tree. To enable decoding, select Enable WebSphere Portal scanning. If the context root URL does not follow the default format, click Add Context Root URL to add one or more context root URLs. Tip: If you are not sure what your portal's context root URL is: 1. On the computer where WebSphere Portal is installed, open the <code>wkplc.properties</code> file in the <code>wp_profile_root/ConfigEngine/properties</code> directory. 2. The context root value is specified by the <code>WpsContextRoot</code> property. Tip: When scanning a WebSphere Portal site, it is recommended to use the predefined WebSphere Portal scan template, which is configured for the purpose.
<i>Flash</i>	
Parse Flash to discover URLs	AppScan will parse Flash code as text data to collect links.
Execute Flash files to discover potential vulnerabilities	AppScan will actually <i>play</i> Flash files and analyze the results to collect links, including dynamic links that may not be discovered by parsing alone. (This utilizes more system resources than parsing.) Adobe Flash Player for Internet Explorer, Version 9.0.124.0 or higher is required. If a supported version is not installed a warning appears next to the check box when selected, and Flash Execution will not run. See "Flash content" on page 296. If you have Adobe Flash Player Version 10.1 or higher, you may get a message that it requires configuration to work with AppScan. See "Flash Player configuration" on page 7. When Flash Execution is selected, three Flash execution limits can also be configured. These are: • Depth Limit: The maximum number of "clicks away from the initial screen" that can be clicked before the scanning of any particular Flash movie is stopped and the scan moves on. • Click Limit: The maximum total number of clicks that can be made before the scanning of any particular Flash movie is stopped and the scan moves on. • Screen Limit: The maximum number of unique Flash states allowed before the scanning of any particular Flash movie is stopped and the scan moves on.

Parameters and Cookies view

Parameters and Cookies view of the Configuration dialog box.

This view is used to manage four main functions:

- Assign special treatment to specific parameters and cookies
- Control the default treatment of parameters and cookies ("redundancy tuning")

- Define parameters and cookies that have a special format which AppScan might not recognize on its own
- Define custom headers

Setting	Description	See
Parameters and Cookies tab	Lets you view, add, edit and delete global parameters that require non-default treatment. For example, your application may have parameters and cookies whose values you do not want AppScan to manipulate during tests. To make sure that AppScan does not change these parameters and cookies, exclude them from tests. For example, your application might lock a user session if certain cookie or parameter values are changed. You should exclude these parameters from manipulation. If you do not exclude them, AppScan may not be able to successfully complete the scan, as these cookies will lock AppScan out of the application. During the Explore stage, AppScan automatically detects cookies and HTML parameters that are likely to be session IDs and adds them to the list in this tab. You can manually add cookies and parameters that you know to be session IDs. The columns in this tab are defined in the table below. Note: The Hide/Show template items button lets you filter out items that originated in the scan template, which may not be relevant to the current scan.	"Parameter definition" on page 69
Custom Parameters tab	Lets you add, edit and delete parameters with a custom format that AppScan might not otherwise recognize as such.	"Custom Parameters tab" on page 76
Custom Headers tab	Lets you define non-standard (custom) HTTP header formats. AppScan must be able to identify parameters in response content and correctly add them to headers it sends to the site, in order to be able to test the site effectively.	"Custom Header tab" on page 78
Redundancy Tuning Defaults	This link (at the bottom of the Parameters and Cookies tab) lets you access and edit the <i>default</i> redundancy tuning applied to all parameters, whether discovered by AppScan or defined by the user. Note: Changing the specific redundancy tuning of an individual parameter is done as part of "Parameter definition" on page 69 Changes to the defaults are not applied retroactively to parameters that have already been defined. This must be done manually for each parameter.	"Redundancy tuning" on page 73

Parameters and Cookies tab fields

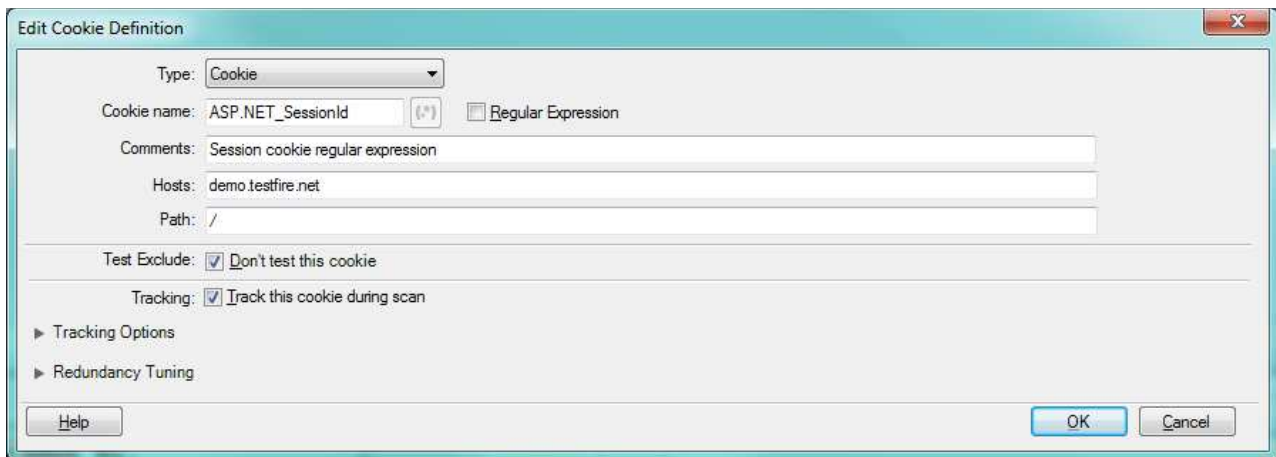
The following table summarizes the fields in this tab.

Heading	Options and description
Type	Parameter / cookie / custom parameter
Name	

Heading	Options and description
Tracking	How to track this parameter/cookie: • As a login value • As a dynamic value • As a fixed value • Don't track it at all
Test Exclude	Defines whether or not to exclude this parameter/cookie from testing during the Test stage of the scan.
Redundancy Tuning	• Default: The default redundancy tuning is applied to this item • Custom: The redundancy tuning for this item is different to the current default
Source	Shows from where AppScan obtained this item: • Scan template: Originated in the scan template • Login session ID: From the login sequence recorded by the user • Multi-step sequence variable: From a sequence recorded by the user • Scan Expert [module name]: From the specified Scan Expert module • Explore Optimizer: From the Explore Optimizer extension • User-defined

Parameter definition Procedure

To add a new definition, click  (or, to edit an existing parameter, select it and click ). The **Add Parameter Definition** dialog box appears.



Setting	Description
Type	Select a parameter type from the drop-down list: Parameter: All parameters matching this name are included in the definition. Cookie: All cookies matching this name are included in the definition. Custom Parameter: This is a custom parameter (select one of the custom parameters from the <i>Name</i> drop-down list)

Setting	Description
Name	The name of the parameter or cookie. Select the adjacent check box if the name you enter is a regular expression. If you do this you can also open the Expression Test PowerTool by clicking  , to help you verify the syntax of your regular expression. See "Parameter names" on page 71 for details.
Comments	You may optionally add a comment about the parameter in this field for your own reference.
Hosts	If a Host is specified: Use this parameter for the specified host only. If left blank: Use this session ID for all hosts.
Path	If the application supplies cookies of the same name from different parts of the application, you can differentiate between them by defining the path for each one. Blank or / will include all occurrences of the cookie.
Test Exclude	Select this check box only if you are sure you don't want AppScan to test this parameter at all.
Tracking	This setting tells AppScan that this parameter or session ID should be updated during the scan whenever a new value is set by the application, so that a valid cookie/parameter is always sent in requests to the application.
Tracking Options...	(Click the link to open this optional section of the dialog box.) These options let you fine-tune how the tracked parameter or cookie is treated. Track Type • Login Value: (Default, and Recommended) Requests sent to the application that include this parameter use the last value of the parameter received in the login process, <i>not including</i> the In-Session request itself. Tip: If you wanted to track the parameter <i>in the In-Session response</i>, you would need to set its type to Dynamic, <i>not Login</i>, and verify that Config > Advanced > Session Management: Parse in-session page is set to True (it's default value). • Dynamic Value: Requests sent to the application that include this parameter use the most recent value received from the application. • Fixed Value: Requests sent to the application that include this parameter always use the value that you enter in the Value field. See "Session IDs" on page 71 for more details. Send cookie on all requests: When selected, the cookie will be included in all requests, even if not explicitly set by the application. Treat as Group: If the cookie name is a regular expression, define whether to treat different cookie names that match the regexp. as a group (and therefore update the name as well as the value, when there are changes) or as separate cookies. Response Pattern: Generally, AppScan updates parameter or cookie values based on the content of links extracted from the response (parameters) or from the cookie header (cookies). If AppScan will not be able to extract the value unaided, you can supply the regexp. that AppScan can use to extract the value from the raw response. The regexp. must contain at least one group, and AppScan will extract the first match. • URL Filter: If you know that the parameter/cookie only appears in a specific URL, you can improve scan efficiency by defining the full URL path here. • Encoding: If the extracted value must be encoded when pasted into the request, define the method here. If you are unsure of the coding, select According to context; if you are sure, selecting the <i>correct</i> encoding is preferable. Options are: None, According to context, URL, XML, JSON. • Match: Select Header and Body (default) or Body only.

Setting	Description
Redundancy Tuning...	(Click the link to open this optional section of the dialog box.) These four check boxes let you fine-tune how AppScan relates to changes in the parameter (or even its existence) during the Explore and Test stages of the scan. See "Redundancy tuning" on page 73

Identifiers that define a parameter or cookie

A parameter or cookie is recognized as unique on the basis of certain identifiers. It follows that you cannot define two or more parameters or cookies with the same identifiers. The table below shows the identifiers for each kind of entry.

Parameter	Parameter name, whether a regular expression, host
Cookie	Parameter name, whether a regular expression, host, path
Custom parameter	Extracted name (if one exists), reference name, host, occurrence index

Parameter names Procedure

In the **Name** text box, enter the parameter or cookie name, or a regular expression that matches it:

- All text is case-sensitive; add (?i) to the regexp if you want it to be case-insensitive
- Strings in the regexp are considered partial; you do not need to add (.*)

A number of default definitions are provided.

Type	Value	Includes
Parameter	__VIEWSTATE	parameters with this string in their name
Parameter and Cookie	^CFID	parameters and cookies whose names begin with CFID or cfid
Parameter and Cookie	Token	parameters and cookies whose names include "Token"

Note: A regular expression (regexp) is a string that describes a set of strings, according to certain syntax rules. The Expression Test PowerTool (**Tools > Expression Test**) can be useful to verify the syntax of your regexps.

If you need additional help writing regexps you may find the following link useful: <http://www.regular-expressions.info/quickstart.html>

Session IDs

If the site uses time-restricted session IDs (in the form of cookies or parameters), the site will reject requests that contain expired tokens; causing the site tests to fail.

Therefore, AppScan must be able to recognize and handle the HTML parameters or cookies that are time-restricted *session IDs*. AppScan will assign session IDs the most recent value available, thus preventing the application session from expiring.

You can determine whether AppScan should automatically update the value of a session ID. Set the **Status** of a session ID:

-

Login Value: (Recommended) When sending a test request containing this parameter, AppScan automatically updates the session ID with the last value received from the application before the In-Session request.

Tip: To track the parameter *in the In-Session response*, you must set its type to **Dynamic**, *not Login*, and verify that **Config > Advanced > Session Management: Parse in-session page** is set to **True** (it's default value).

This status is recommended for most parameters and cookies, unless there is a specific need to set a specific value. However, when Login Value session IDs are used, the value might expire while it is in the database.

To update a tracked session ID in the database: Just before running the scan, visit the URL where the session ID is sent. A new session ID will be sent, with an updated value.

- **Dynamic:** AppScan automatically updates the session ID value during the Test stage, according to new values set by the web application in prior tests (for example, as with Shadow Cookies).

Select *Dynamic* only if you know that your web application enforces security measures that demand that a specific session ID be updated during certain usage procedures.

-

Fixed: Retains a fixed value. Set a fixed value for a session ID if your web application security needs this session ID to always have this value.

During the Explore stage, AppScan automatically detects cookies and HTML parameters that are likely to be session IDs and adds them to a list. You can manually add the cookie and parameters that you know to be session IDs when you configure the scan.

Session IDs within the URL:

How to track a Session ID that is embedded in the URL Path.

About this task

If AppScan does not track session IDs correctly it will frequently fall out of session. This section describes how to track a session ID that is embedded in the URL.

Session ID: abc34f3fa135

URL containing session ID: `http://domain.name/dir/subdir/abc34f3fa135/anotherdir?param=val`

Procedure

1. Create the rule for recognizing this custom parameter:
 - a. Open **Scan Config > Parameters and Cookies > Custom Parameters tab**
 - b. Click  to add a new custom parameter.
 - c. In the Reference Name field, enter a name for the custom parameter rule.
 - d. In the Pattern field, enter a regular expression that describes the parameter format. Example: `(abc[a-zA-Z0-9]+)`
 - e. Leave Value Group Index and Name Group Index unchanged.
 - f. In the Location field, select **Path**.
 - g. Click **OK** to save the changes.
2. Configure AppScan to track this custom parameter:
 - a. Open **Scan Config > Parameters and Cookies > Parameters and Cookies tab**
 - b. Click  to add a new parameter.

- c. Set the Type to **Custom Parameter**.
 - d. Select the Reference Name you assigned in the previous step.
 - e. Select the **Track this parameter during the scan** check box.
 - f. Set Track Type to **Login Value** or **Dynamic** as appropriate.
 - g. Click **OK** to save the changes.
3. If the recorded login sequence includes a URL that contains this in-path session ID, you must record the login again so that AppScan can track the Session ID.
 4. Run a full Re-Scan or Re-Explore as needed.

Redundancy tuning

Careful redundancy tuning can significantly reduce scan time.

AppScan tries to avoid sending more than one request when it is clear that no new information will be revealed. In many cases a difference in the value of a particular parameter is insignificant, and it is not necessary to send more than one request when all other parameters are the same and only that value is different.

Consider the following two requests:

```
.../doAction.pl?action=buy&timestamp=14:00&n=1
```

```
.../doAction.pl?action=buy&timestamp=15:30&n=1
```

The only difference between them is their Timestamp value.

In most cases it would be sufficient to send a single request using either of the configurations, and unnecessary to send them both. It is unlikely that the response to one of the requests would reveal any weakness that would not also be revealed by the other. You should therefore configure the Redundancy Tuning settings for the Timestamp parameter so that in cases like this only one of the requests will be sent.

See also:

“Redundancy tuning options”

“Redundancy tuning defaults” on page 74

“Changing redundancy tuning defaults” on page 75

“Changing redundancy tuning for a specific parameter” on page 75

Redundancy tuning options:

Redundancy tuning for parameters and cookies with special attributes, listed in Configuration > Parameters and Cookies.

check box	When selected...
Explore the URL again whenever this parameter/cookie is added or removed.	During the Explore Stage, treat URLs whose only difference is that one includes this parameter and one does not, as different URLs, and explore both. For example, in the case of the following two URLs, <i>both</i> will be explored: <pre>...page.jsp ...page.jsp?thisParam=Value</pre> If you deselect this check box, only one request will be sent in cases like this, and the other will be discarded.
Explore the URL again whenever the value of this parameter/cookie changes.	During the Explore Stage, treat URLs whose only difference is the value of this parameter/cookie, as different URLs, and explore both. For example, in the case of the following two URLs, <i>both</i> will be explored: <pre>...page.jsp?thisParam=Value1 ...page.jsp?thisParam=Value2</pre> If you deselect this check box, only one request will be sent in cases like this, and the other will be discarded. Note: This option is irrelevant when the parameter or cookie is tracked.
Repeat all adjacent parameter/cookie tests whenever this parameter/cookie is added or removed.	During the Test Stage, treat URLs whose only difference is that this parameter was added or removed, as different URLs, and test adjacent parameters again. For example, in the case of the following two URLs, <i>two full sets</i> of tests will be generated for the adjacent parameter, one set for each URL. <pre>...page.jsp?adjacentParam=<test_this> ...page.jsp?adjacentParam=<test_this>&thisParam=Value</pre> If you deselect this check box, only one set of tests will be generated for the adjacent parameter.
Repeat all adjacent parameter/cookie tests whenever the value of this parameter/cookie changes.	During the Test Stage, treat URLs whose only difference is the value of this parameter/cookie, as different URLs, and test adjacent parameters again. For example, in the case of the following two URLs, <i>two full sets</i> of tests will be generated for the adjacent parameter, one set for each URL. <pre>...page.jsp?adjacentParam=<test_this>&thisParam=Value1 ...page.jsp?adjacentParam=<test_this>&thisParam=Value2</pre> If you deselect this check box, only one set of tests will be generated for the adjacent parameter. Note: This option is irrelevant when the parameter or cookie is tracked.

See also:

“Redundancy tuning” on page 73

Redundancy tuning defaults:

Redundancy tuning for parameters and cookies with special attributes, listed in Configuration > Parameters and Cookies.

If the Explore Optimization eXtension is not run, the default redundancy tuning setting is this:

Check box	Default value
Explore the URL again whenever this parameter/cookie is added or removed.	

Check box	Default value
Explore the URL again whenever the value of this parameter/cookie changes.	
Repeat all adjacent parameter/cookie tests whenever this parameter/cookie is added or removed.	
Repeat all adjacent parameter/cookie tests whenever the value of this parameter/cookie changes.	

If the Explore Optimization eXtension is activated and run, the redundancy tuning setting for navigational parameters is *increased*, while the default setting (applied to all non-navigational parameters) is *decreased*.

Check box	Navigational	Default
Explore the URL again whenever this parameter/cookie is added or removed.		
Explore the URL again whenever the value of this parameter/cookie changes.		
Repeat all adjacent parameter/cookie tests whenever this parameter/cookie is added or removed.		
Repeat all adjacent parameter/cookie tests whenever the value of this parameter/cookie changes.		

See also:

“Redundancy tuning” on page 73

Changing redundancy tuning defaults:

The default redundancy tuning setting is applied to all new additions to the list in Configuration > Parameters and Cookies, unless manually changed.

Procedure

1. At the bottom of the Parameters and Cookies tab, click **Redundancy Tuning Defaults**.
The Redundancy Tuning Defaults dialog box opens. Its options are described in “Redundancy tuning options” on page 73.
2. Adjust the settings as necessary (see “Redundancy tuning options” on page 73 for details).
Changes you make here are applied to all new parameters as they are discovered by AppScan or defined by the user.

Note: Changes to the defaults are *not* applied retroactively to parameters that have already been defined. This must be done manually for each parameter.

What to do next

See: “Redundancy tuning options” on page 73

Changing redundancy tuning for a specific parameter:

Procedure

1. In the Parameters and Cookies tab, select the parameter and click .
The **Parameter Definition** dialog box appears (see “Parameter definition” on page 69 for details).
2. At the bottom of the dialog box, click **Additional Options**.
The Redundancy Tuning options open (see “Redundancy tuning options” on page 73 for details).

What to do next

See “Redundancy tuning options” on page 73

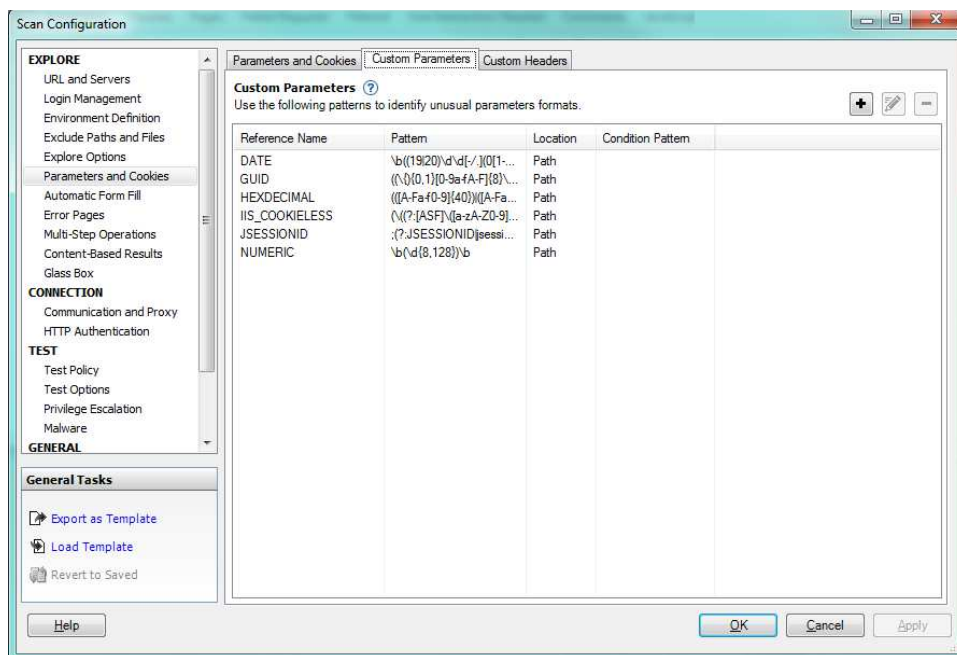
Custom Parameters tab

The Custom Parameters tab of Parameters and Cookies view in the Configuration dialog box.

About this task

The second tab of the Explore: Parameters and Cookies view lets you create and manage custom parameters with formats that AppScan would not be able to recognize automatically.

AppScan automatically recognizes parameters in standard HTML format, but if parameters are in other formats (for example within the path or within another parameter), you need to define them to AppScan so it can recognize, follow and manipulate them during scanning.



Procedure

To define a new custom parameter, click ; to modify an existing definition, select the parameter in the table and click .
The **Add/Edit Custom Parameter** dialog box opens. Its fields and options are described in the table following.

The screenshot shows the 'Edit Custom Parameter' dialog box. It has the following fields and controls:

- Reference Name:** A text box containing 'GUID'.
- Pattern:** A text box containing a complex regular expression: `((\{0,1}[0-9a-fA-F]{8}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{12})`. To the right of the text box is a small icon with a question mark inside a square.
- Name group index:** A dropdown menu currently set to 'None'.
- Value group index:** A dropdown menu currently set to '1'.
- Location:** A dropdown menu currently set to 'Path'.
- Condition Pattern:** An empty text box. To its right is a small icon with a question mark inside a square.
- Buttons:** 'Help', 'OK', and 'Cancel' buttons are at the bottom.

Setting	Description
Reference Name	Assign the parameter an easily recognizable name. Custom parameters appear on the Application Data tree with the prefix "__patternParameter__" followed by the parameter name.
Pattern	A regular expression containing one or more groups that defines the parameter. A "group" is a section of the regular expression delineated by parentheses. One of the groups contains the parameter value, and there may also be a group that contains the name. You can open the Expression Test PowerTool by clicking  to help you verify the syntax of the regular expression.
Name group index	(Optional) If the name is included in the regular expression, indicate which group (1,2,3...) contains it. AppScan uses this value to "count through the groups" and locate the parameter name (see example below).
Value group index	Indicate which group (1, 2, 3...) in the above regular expression contains the parameter's value. AppScan uses this value to "count through the groups" and locate the parameter value (see example below).
Location	Indicate which component of the request contains this parameter: Body / Path / Query. Note: The selection you make here will apply to both the Pattern and the Condition Pattern (if any), but not to the Response Pattern.
Condition Pattern	(Optional) You can enter a regular expression that defines the whole component (Body, Path or Query) containing the parameter. AppScan will create the parameter only when the whole component matches this pattern; saving scan time. For example, if the parameter is located in the Body, and the Body must be XML, you could set as the Condition Pattern a regex that verifies that the Body starts and ends with XML tags. In cases where it does not, AppScan will not create the parameter.

Note: Fields that require or accept a regular expression have this button: , which opens the Expression Test PowerTool, to help you verify the syntax of the regular expression.

Group indexes

To understand the indexing system for groups in the Pattern regular expression, consider the following example:

Pattern: (abc)((def)(ghi))

The groups in this expression would be indexed as follows:

Group 1: (abc)
Group 2: ((def)(ghi))
Group 3: (def)
Group 4: (ghi)

Use the **Name group index** and **Value group index** drop down lists to select the correct groups for the parameter. The selected group is highlighted in the Pattern field.

Note: If you change the pattern after indexes have been selected, and the selected index no longer exists in the pattern, a warning appears, but the value is not automatically changed, and you must change it manually.

Custom Header tab

The Custom Headers tab of Parameters and Cookies view in the Configuration dialog box.

About this task

The third tab of Explore: Parameters and Cookies view lets you define non-standard (custom) HTTP header formats. AppScan must be able to identify parameters in response content and correctly add them to headers it sends to the site, in order to be able to test the site effectively. It attempts to recognize custom headers automatically, but you can use this tab to add and modify the definitions. You can also activate or deactivate existing definitions (when deactivated the definition is saved with the scan but not used).

Procedure

To create new definition, click ; to modify an existing definition, select the header in the table and click .

The **Add/Edit Custom Header** dialog box opens. Its fields and options are described in the table following.

Add Custom Header

Select one of the headers found by AppScan in recorded requests, or define a new one:

Authorization: Bearer qPcTtUv14jX_c6F1r02wOWrr5Ur9trE0UdATAzAPMozOs-gFGSWPEv1WbfyyufDq3DKuvJSe

Header Name: Authorization

Header Value: ☐ Fixed ☒ Dynamic

Format: Bearer {0}

Regular expression that identifies the dynamic value(s) in site responses: "access_token""s*:"([w-.~+/@]+)"

► Regular expression review pane

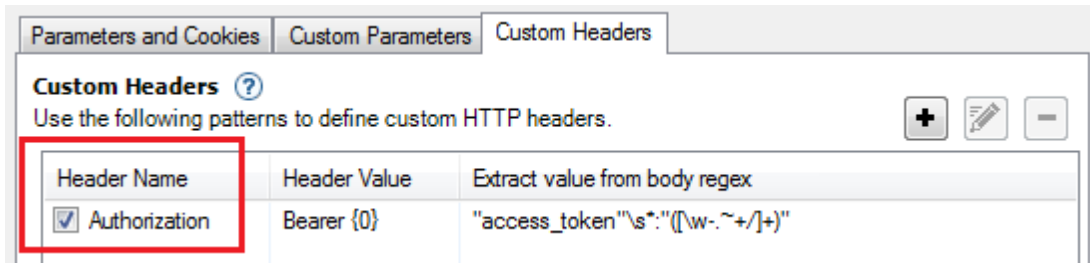
Header Validation: Authorization: Bearer qPcTtUv14jX_c6F1r02wOWrr5Ur9trE0UdATAzAPMozOs-gFGSWPEv1WbfyyufDq3DKuvJSe6xNBpYRtnonEoJNXdz4JQUyDc28L0WfRYR7sdG5JUIGsRK9MI-e_wYhXp-0NowAoOPnijkYkQrNdrbujg2rUMAYbBlmjSFkeRUZkUARXyjVPNQjr6DHgnqtGPghLFrFeSPVGfrA4jE8Af2PfuBv_NTJOjM_QpyuLX0tihxShB3EaDXEUynOjtwEy2gUnKTYLUaQSTog9aK4-U2ffkLmGyde84-bqNx-yszpPszPi_jVjdV8fw_clR3sWJ0XaKhIWNZN0IKjxplaCYISxi6prg8xTY1RVb8RpwL-YDvLiGikHUhNvi_Ud5vLK4gSLGu6KihMZPrN1F3nVJPG1105KXVZg5BATvgpZKSWxUNq5FdGCCnJmqe5ntkuo3xdMDa1Bq6RPvLuNXCHG1rYGI_rq1mM8-KQoEv5TzMVYpxPx_vCwKlyT5U22Xr14prdzXQLCc5ef8

Help OK Cancel

Setting	Description
Select header...	If requests have been recorded, and AppScan has identified a custom header, it will appear in a drop-down list at the top of the dialog box. If no custom headers have been found, this drop-down list does not appear. If you select a header from this list, the remaining fields are filled automatically.
Header Name	The HTTP header name.
<i>Track Type</i>	
Login/Dynamic/Fixed	• Login Value: (Default, and Recommended) Requests sent to the application that include this header use the value of the header received at the end of the login process. • Dynamic Value: Requests sent to the application that include this header use the most recent value received from the application. • Fixed Value: Requests sent to the application that include this header always use the value that you enter in the Value field.
Value	(Fixed only) Enter the value.
Format	(Dynamic only) Define the format of the full header, with one or more groups for the dynamic values, starting with {0} for the first group, {1} for the next, and so on.
Regular expression	(Dynamic only) Regexp defining the values in the site's response. Must include a group for each value defined in the Format field.
Regular expression review pane	(Dynamic only) Click to expand. Use this pane to validate your regexp. Enter the full response in the upper field, and the lower field will display groups identified, and their values.
Header Validation	Indicates whether or not the header definition has been successfully validated for use when scanning.

Results

When you add a custom header definition, it appears in the Custom Headers list, and the check box next to it is selected, indicating that the definition is activated. Deactivated entries are saved with the scan but not used.



Automatic Form Fill view

Automatic Form Fill view of the Configuration dialog box contains the values used to fill forms in your application.

Automatic Form Fill values are the values AppScan uses to fill forms in your application. There are default values for many of these, and they are automatically updated to include any values you enter during a “Recording a login” on page 45.

You can view, add to and edit these values from this view of the Scan Configuration dialog box.

Setting	Details
Enable Automatic Form Fill	When this check box is selected (default), AppScan attempts to automatically fill-in the user input forms of your application during the Explore stage.
Automatic Form Fill Properties List	List of values used by AppScan to fill in forms automatically during the scan. You can add to and edit this list. For details, see “Form properties”
Username and Password Parameters	Shows the strings, comma separated, that AppScan uses to recognize fields that must be filled with the Username or Password. You can edit this string as required. Add a comma without a space, between strings.
Username and Password Values	Shows the Name and Password used for login if Automatic Login is selected in Login view. You can edit it here. (For details see “Login tab” on page 44.)
Fill unknown fields with	You can type in a string that AppScan will use whenever it encounters a field that it does not recognize (the default string is 1234), or get AppScan to use a Random Value each time.

Form properties

Form Properties view of the Configuration dialog box lists the values that AppScan will use to fill in forms automatically.

About this task

You can add to or edit the **Form Properties** that AppScan uses to automatically fill forms during scanning.

Procedure

To add a new definition, click  (or, to edit an existing form property, select it and click ). The **Form Properties** dialog box appears.

Column	Description
Description	A name that describes the parameter.
Parameters	The name, or part of the name, that is used to define this parameter in the HTML. When AppScan finds a field with this parameter name it fills in the value supplied. To enter multiple parameter names in one row, use a comma without a space. Example: Description: Zip Code Parameters: zip,postal
Value	The user input that AppScan will send for this parameter.
Override	Sometimes the site fills in its own default value for a particular field. In such cases, by default AppScan will use the site's suggestion rather than the one entered here. Select this check box if you want AppScan to use the value defined here, <i>even when the site offers a different default value.</i>
Match Type	Defines how AppScan will search for the parameter defined here, and how it relates to field-filling options supplied by the site. Options are "Partial" or "Complete." Complete: The value will be input <i>only</i> for parameters that exactly match one of the text strings in the Parameters field. (For example, the parameter defined as addr, will not be used for a field named address.) Furthermore, if the field offers a drill-down list of options, this value will be used only if it <i>exactly</i> matches one of the options. Partial: This value will be input for parameters that match <i>or partially match</i> one of the parameter strings listed here. (For example, if the parameter is defined as addr, it will also be used for a fields named address, or ADDR.) Furthermore, if the field offers a drill-down list of options, and none of the options exactly match the value entered here, a close match will be used instead.
URL	If this field is left empty, this value will be used for this parameter regardless of URL. If a specific URL is given, this value will be used for only for parameters that occur in this URL. (You can therefore define a parameter one way for one URL and differently for another.) However, if there is no other row with the same parameters and an empty URL, the value of this row is default for all other URLs too.

Example

"Exporting and Importing Form Properties"

"Importing a saved ASFF file" on page 82

Exporting and Importing Form Properties: About this task

You can save form filler and authentication properties for future use, by exporting your **Form Properties** table as an XML file.

Procedure

1. In **Explore: Automatic Form Fill** view, click **Export**.

The **Save As** dialog box that appears lets you save the data as an ASFF file (AppScan Form Filler file). The content is in XML format.

2. Name your file and click **Save**.

Example of ASFF file content:

```
<FormFiller Version="1.0" Enabled="True" DefaultValue="1234"
UseDefaultValue="True">
  <Group Name="InternalAppScanUserName" Value="" MatchType="Partial" Action="">
    <MatchNames>
      <MatchName>user</MatchName>
      <MatchName>name</MatchName>
      <MatchName>uid</MatchName>
      <MatchName>login</MatchName>
      <MatchName>usr</MatchName>
    </MatchNames>
  </Group>
```

Importing a saved ASFF file:

Procedure

1. In **Explore: Automatic Form Fill** view, click **Import**.

A message appears, warning you that by importing an ASFF file, you will delete any current data in the table. If you want to continue, click **OK**.

2. Browse to the relevant ASFF file and click **Open**.

Form properties in Flash files

Form properties in Adobe Flash files may require special attention.

About this task

An important element in configuring scans is defining valid values for form fields, especially when they have unusual names. For example if an email field has an unexpected name, AppScan might send input that is not a valid email address format, resulting in the failure of the tests.

In the case of Flash files, the problem is greater, because of the inaccessibility of the instance names (form fields). Accessing these names and configuring valid values for them in “Automatic Form Fill view” on page 80 of the Scan Configuration dialog box.

Error Pages view

Error Pages view of the Configuration dialog box.

When AppScan gets a 404 error page in response to a test, it generally records the test as failed, since this response indicates that the site successfully recognized the request as illegal. In some cases the opposite is true, and the error page indicates a successful result. In both cases it is important that error pages are correctly defined so that AppScan can recognize them as such.

Web applications and servers often use customized or dynamically generated 404 error pages that may be hard to recognize automatically. AppScan attempts to recognize customized 404 error pages, but in some cases may not succeed in doing so. If it receives a custom error page and does not recognize it as such, it may record the result as positive, where in fact it should be negative, or the other way around. By default the Error Pages list includes standard error page definitions. For each definition type and value are shown.

If your application's error pages are not covered by the definitions in this list, you should add the necessary strings, regexps and URLs that will enable AppScan to recognize your error pages. By doing this you can reduce the number of “false positives” in your scan results. There are two ways you can do this:

- You can define the error page manually before scanning. See “Define a new error page” on page 83
- If you have run the Explore stage, you can set a URL that was discovered as an error page. See “Set an error page” on page 84

Important: Incorrect error page definitions may cause both "false positive" and "false negative" results, therefore when you add or delete error pages *after* the Test stage of a scan, the scan results should be updated.

- For tests where the previous definition indicated the success of a test, results can be updated by clicking **Apply to Current Results**
- For tests where the previous definition indicated the failure of a test, you must re-test.

See also:

"Applying error page changes" on page 84

Define a new error page Procedure

1. To add a new Custom Error Page definition, click  (or, to edit an existing one, select it and click ).

The **Custom Error Page** dialog box appears.

2. In the **Type** list, select:

- **String:** Match a string found in the HTML content of your error pages.
- **Regular Expression:** Match a regular expression found in the HTML content of your error pages.

You can open the Expression Test PowerTool by clicking  button, to help you verify the syntax of the regular expression.

- **URL:** Match the URL of the response page.
 - **Page:** Match the exact response page. (This method is used when you click **Set as Error Page** for a particular variant in the "Request/Response tab" on page 197.)
3. In the **Value** text box, enter the string, regular expression, or URL (relative path and filename).

Important: Incorrect error page definitions may cause both "false positive" and "false negative" results, therefore when you add or delete error pages *after* the Test stage of a scan, the scan results should be updated.

- For tests where the previous definition indicated the success of a test, results can be updated by clicking **Apply to Current Results**
 - For tests where the previous definition indicated the failure of a test, you must re-test.
4. Click **OK**.

Any response that matches the string, regular expression, or URL will be recognized by AppScan as an error page from now on.

Example

"Custom page filtering examples"

Custom page filtering examples:

The following table provides examples of how to add pages to the Custom Error Page list.

Type	Value	Adds
URL	/fileNotFound.aspx	Specific file
String	"page not found"	Any page with this string

Type	Value	Adds
Regular Expression	(?i)(URL page) (.*) not found	Any page with: "URL name not found", "url name not found", "Page page.ext not found", etc.

Note: A regular expression (regex) is a string that describes a set of strings, according to certain syntax rules. The Expression Test PowerTool (**Tools > Expression Test**) can be useful to verify the syntax of your regexps.

If you need additional help writing regexps you may find the following link useful: <http://www.regular-expressions.info/quickstart.html>

Set an error page

You can set a page listed in the Explore results as a custom error page.

About this task

You can view the list of URLs that were discovered in the Explore stage in Data view > Pages. If this wrongly lists one of your site's custom error pages, you can set the page as an error page.

Procedure

1. Open the Explore results in Application Data view (F2).
2. In the Result pane, click **Pages**.
3. Right-click on the URL and click **Set as Error Page**.

You are asked to confirm the update of the results to incorporate the new definition.

4. Click **Start**.

The URL is added to the Error Page list in the scan configuration, and the results are updated as needed.

Applying error page changes

About this task

If you edit or add to the list of Error Pages after scanning, you must update the scan results to reflect the changes (remove "false positives").

After editing the *Apply to Current® Results* button above the list becomes activated.

Procedure

Click **Apply to Current Results**.

The results are updated to take into account that the new/updated pages are error pages.

Multi-Step Operations view

Multi-Step Operations view of the Configuration dialog box is for testing parts of the site that can only be reached by clicking links in a specific order.

A multi-step operation is needed to explore parts of the site that can only be reached by clicking links *in a specific order*, such as an online shop where the user adds items to a cart before paying for them.

Consider the following three pages:

1. User adds one or more items to a shopping cart
2. User fills in payment and shipping details
3. User receives confirmation that the order is complete

Page 2 can be reached only via Page 1. Page 3 can be reached only via Page 1 followed by Page 2. This is a sequence. In order to be able to test Pages 2 and 3, AppScan must send the correct sequence of HTTP requests before each test.

In the case of the above example you would record a single sequence: Page 1 > Page 2 > Page 3. AppScan would extract the necessary sub-sequences from this sequence, as required. (When testing Page 2 it would send a Page 1 request first; when testing Page 3, it would send Page 1 followed by Page 2.)

Note: It is suggested that the number of multi-step operations be limited to five, with no more than 25 steps in any single operation, and no more than 70 steps altogether.

Note: Configuring multi-step operations should not be confused with manual exploring, and should only be used in cases like the one described above. For more details see Manual Explore using AppScan

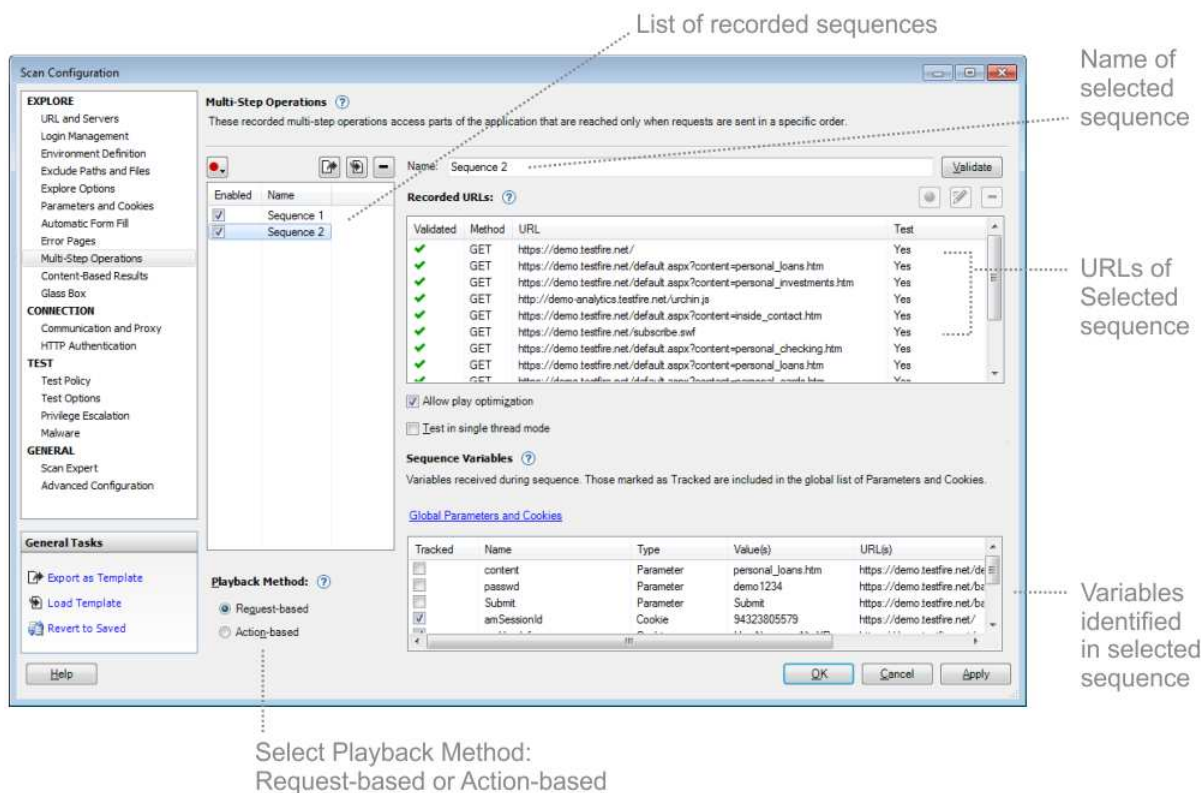


Table 5. Multi-Step Operations view options

Setting	Details
Record	Click to record a new sequence. If login details have been configured, you can click the down arrow to select: AppScan IE browser > Log in and then record AppScan will log in to the application automatically (using the login you recorded) before the browser opens. You can then record your multi-step operation without the login. AppScan IE browser > Record without login AppScan will begin recording the sequence without logging in. When the browser opens you can log in and then record your multi-step sequence. The login will be part of the sequence and will be replayed each time the sequence is played. AppScan Chromium browser AppScan will record <i>using the built-in Chromium-based browser</i>, without logging in. When the browser opens you can log in and then record your multi-step sequence. The login will be part of the sequence and will be replayed each time the sequence is played. For details, see “Record a Sequence” on page 87
  	Export a sequence (as an SEQ file) for use with a different scan; import a sequence (SEQ file) exported from a different scan; delete the selected sequence from the current scan.
Playback Method	When you record a multi-step operation, AppScan records both the actions and the requests. You can select which of them will be used for the scan: Request-based playback Sends the raw HTTP requests from the recording. This method is usually faster. Action-based playback Replays the clicks and keystrokes of the user. Reasons for selecting this method could be that the site includes a lot of JavaScript, or that some of the requests in the request-based playback were marked with a red X when you attempted to validate them. This method can increase scan time. Request-based playback is the default method. Note: If the scan is configured not to use a browser other than the embedded browser (Tools > Options > Use external browser), request-based playback is always used. Note: If you load a sequence that was recorded in a version of AppScan that did not support action-based playback, request-based playback is used <i>for that sequence</i>, even if action-based playback is selected. Note: If you select Action-based playback for a multi-step operation, you must also select Action-based as the login method. If necessary, record the Login sequence again (see “Login Management view” on page 44).
Sequence List	Lists all recorded Multi-Step Operations for this scan.
Sequence Name	The name of the sequence that is selected in the List of Sequences. The check box next to each one indicates if the sequence is enabled for this scan. Validate Click this to check that the sequence is valid. AppScan replays the sequence, and any requests that receive a response different to the original response are marked with a red X, indicating that they will not be tested. Tip: A common reason for requests receiving a different response is the presence of a dynamic sequence variable that needs to be defined, see “Sequence variables” on page 89. If this is not the problem, and the site contains JavaScript, changing to action-based playback may give better results.

Table 5. Multi-Step Operations view options (continued)

Setting	Details
Recorded URLs	Shows the links or actions in the selected sequence. Validated A green check mark indicates that the URL has been validated. A red X appears next to URLs that were not validated. Test Indicates whether this URL will be tested on its own (as well as in the Multi-Step Operation). Options are Yes/No. To change the setting right-click on the URL and select Test / Don't Test. Even if you select No the URL will still be played as part of the Multi-Step Operation. Play Sequence (Applies to tested URLs only) Indicates whether the previous steps in the sequence will be replayed <i>each time</i> this URL is tested. Options are Yes/No. To change the setting right-click and select Play sequence before testing request > Yes/No. • View any link in the sequence by selecting it and then clicking the browser button (you can delete individual requests by clicking the trash icon in the upper right of the dialog that opens) • Delete any link in the sequence by selecting it and clicking . After doing this click Validate to check that the updated sequence stays in-session.
Log in before sequence replay	If selected, each time a Multi-Step Operation is played, AppScan will log in first. This option is cleared if you record the login as part of the multi-step operation.
Allow play optimization	(Request-based playback only) When selected (default) AppScan attempts to optimize scan time by avoiding unnecessary playback. You should not disable this setting unless you find that AppScan is missing parts of the application due to play optimization. The “Scan Log” on page 274 can help in determining this.
Test in Single-Thread mode	AppScan may send two or more requests simultaneously, if they don't require the replaying of a sequence between them. If this results in parts of the application being missed, select this check box.
Sequence Variables	Lists variables that were received while recording the sequence(s), and indicates those that AppScan has determined should be tracked. These may be session IDs or other variables. You can change the status of variables in this list to improve how AppScan deals with them (for details see “Sequence variables” on page 89).

See also:

Manual Explore using AppScan

“Scan Multi-Step Operations Only” on page 170

Record a Sequence

About this task

If a login sequence has been configured (see “Login tab” on page 44), there are two options when recording a multi-step operation:

AppScan IE browser > Log in and then record

AppScan will log in to the application automatically (using the login you recorded) before the browser opens. You can then record your multi-step operation without the login.

AppScan IE browser > Record without login

AppScan will begin recording the sequence without logging in. When the browser opens you can log in and then record your multi-step sequence. The login will be part of the sequence and will be replayed each time the sequence is played.

AppScan Chromium browser

AppScan will record *using the built-in Chromium-based browser*, without logging in. When the browser opens you can log in and then record your multi-step sequence. The login will be part of the sequence and will be replayed each time the sequence is played.

If no login sequence has been configured there is just one option: **Record**.

Important: During playback of a multi-step operation, in-session detection is Off (see "Login tab" on page 44). This means that AppScan does not verify that it is logged in. Therefore, if the failure of the multi-step operation will cause the user to be logged out of the application, it is important that login be recorded as part of the sequence (so it will be replayed each time the sequence runs). If this is not done the multi-step operation may fail.

Note: If your website does not support Internet Explorer, click > **Use AppScan Chromium browser** instead.

Procedure

1. Click  and select one of the record options (see above).
The browser opens and begins recording.

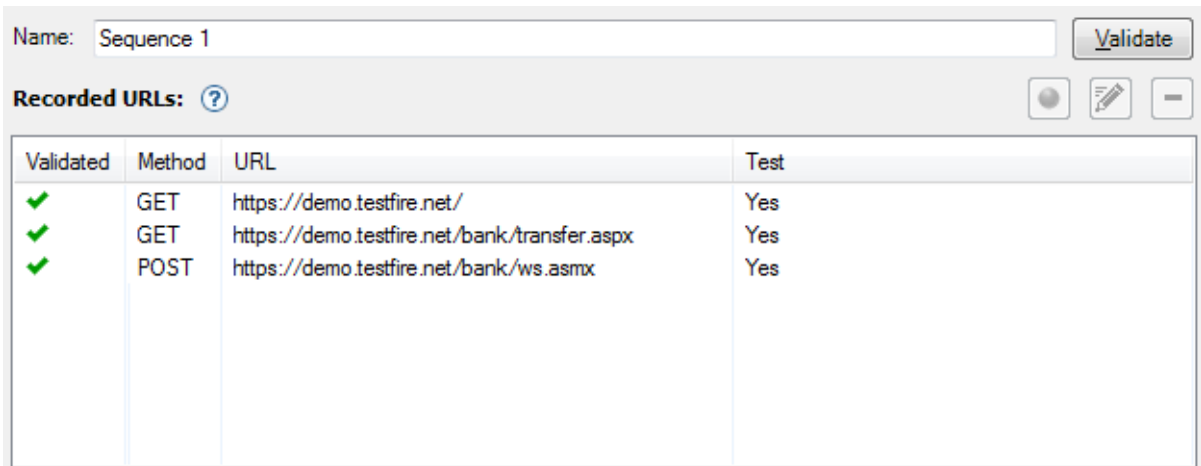
2. Click on links and fill in fields as necessary to reach the required pages. You can use the **Pause**  button if you want to click links without recording them as part of the operation.

3. Close the browser.

The sequence appears in the *Sequence* pane (upper right). Sequences are automatically named in order: "Sequence 1", "Sequence 2" etc., but you can rename by typing into the name field.

You can optionally change the Playback Method (bottom left of the dialog box):

- **Request-based** playback (default) sends the raw HTTP requests from the recording. This method is usually faster.



Validated	Method	URL	Test
✓	GET	https://demo.testfire.net/	Yes
✓	GET	https://demo.testfire.net/bank/transfer.aspx	Yes
✓	POST	https://demo.testfire.net/bank/ws.aspx	Yes

- **Action-based** playback replays the clicks and keystrokes of the user. Reasons for selecting this method could be that the site includes a lot of JavaScript, or that some of the requests in the request-based playback were marked with a red X when you attempted to validate them. This method can increase scan time.

Name: Validate

Recorded Browser Actions: ? ⏮ ✎ ⏭

Validated	Action	Method	URL	Test
✓	Click on Checking	GET	https://demo.testfire...	Yes
✓	Click on MY ACCOUNT			
✓	Click on Transfer Funds			
✓	Set 1001160140 Checking1001160141 Savings to option number 0	GET	https://demo.testfire...	Yes
✓	Click on 1001160140 Checking1001160141 Savings			
✓	Set 1001160140 Checking1001160141 Savings to option number 1			
✓	Click on 1001160140 Checking1001160141 Savings			
✓	Click on transferAmount			

Note: If the scan is configured to use a browser other than the embedded browser (Tools > Options > Use external browser), request-based playback is always used.

Note: If your site requires users to log in, and you selected Request-Based Login, you *must select* Request-based Multi-Step Operations too, otherwise the Multi-Step Operations will not be sent.

4. Click **Validate**.

AppScan replays the sequence, and a green check-mark appears next each request or action that is successfully replayed. If a request or action is not successful a red X appears next to it. Options:

- View any URL by selecting it and clicking 
- Remove any unnecessary step by selecting it and clicking . After doing this click the **Validate** button, to check that the sequence still keeps in-session.
- Right-click on a step in the sequence and set to **Don't Test**. The URL will still be included when playing the sequence, but will not be tested individually.
- Right-click on a step that is set to be tested individually, and select **Play sequence before testing request > No** if it is not necessary to play the previous steps in the sequence each time this URL is tested.

Sequence variables

The Sequence Variables pane lists all variables that were received during the recorded sequences. Those that AppScan recognizes as session IDs are automatically added to the global list of parameters and cookies that are tracked during the scan (see "Parameters and Cookies view" on page 67), and they are marked with a checkmark in the "Tracked" column, to indicate that they are being tracked.

Sequence Variables
Variables received during sequence. Those marked as Tracked are included in the global list of Parameters and Cookies.

[Global Parameters and Cookies](#)

Tracked	Name	Type	Value(s)
☐	context	Parameter	Random string
☐	lastAccounts	Parameter	1001160140
☒	ASP.NET_SessionId	Cookie	rchatyguaf2vzzae545...
☒	Limit	Cookie	10000
☒	Interest	Cookie	7.9
☒	CardType	Cookie	Gold
☒	Userinfo	Cookie	UserId=1001160140&...

Not tracked → ☐ ☒

Dynamic value set as "Random string" → Random string

Tracking variables

You can change the status of any variable by selecting it and selecting or deselecting the **Tracked** check box. When a variable is "tracked" AppScan will always send the most recently received version, to help stay in-session.

Note that if you designate a variable as "not tracked" it is removed from the global list of parameters and cookies (see "Parameters and Cookies view" on page 67).

Defining a variable's format

You can define the parameter's format, to help AppScan send (during the Test stage) valid alternative versions of a parameter that varies. Right-click on the parameter and select one of the "Dynamic value" options:

- Random integer (between 1 - 1000)
- Decrementing integer (starts from 999999, and subtracts 1 each time)
- Incrementing integer (starts from 1, and adds 1 each time)
- Incrementing integer with leading zeros (starts from 000001, and adds 1 each time)
- Random string (5 random characters)
- Random alphabetic string (6 random alphabetic characters)
- Date and time (MMddyyHHmmss)
- Date, time and milliseconds (MMddyyHHmmssSSS)
- Random email address

If the sequence registers a new user, and the process requires a new email address to be entered each time (so that the site will initiate the registration process rather than an "Existing User" page), define the variable as a "Random email address". Then, each time a test request is sent that includes the variable, a different email address will be used, enabling AppScan to test the registration pages.

Content-Based Results view

Content-Based view of the Configuration dialog box. You can use this view to define a logical structure for the application tree, if AppScan will not be able to do this based on URL structure.

- If your site content is structured in such a way that the URLs reflect a folder-like hierarchy, the scan results will automatically reflect this, making them easy to navigate.
- If your site uses "breadcrumbs", or other "content-based" navigational methods, so that the URLs do not indicate the user's "location" within the site, it is recommended that you "teach" AppScan how the site is "logically" structured, so it can present the scan results in an easily understood format, rather than long lists of results under one or two URLs. This is not essential, *but will make it easier for you to navigate the results.*

For example, the code snippet below has a logical structure **Home | Buy | Books** and it would be useful to structure the results so that "Books" appears under "Buy", and "Buy" under "Home".

```
<td class="navigation">
  <a href="http://www.onlineshop.com/">Home</a> &gt;
  <a href="http://hub.onlineshop.com/buy?ssPageName=h:h:cat:US">Buy</a> &gt;
  <b>Books<b>
</td>
```

To do this, you define the rules that will enable AppScan to identify and extract the relevant content (in this case "Home", "Buy" and "Books") to construct a content-based tree.

Once you have defined the rules, you can select the Content-Based option in the Application Tree, to display the results using this information. (See "Security Issues: Application Tree" on page 185.)

Note: The total number of security issues (shown at the top of the “Result List” on page 20) is a measure of the vulnerable locations in the site, and depends in part on how site is structured. If you define a content-based structure, the total number of issues in the application tree may not be the same as it is for the URL-based application tree (for the same results). When site structure is content-based (rather than URL-based), and content-based view is configured correctly, the issue count in content-based view represents more accurately the number of “vulnerable locations” that exist in the site. The total number of *variants* (at the top of the Result List in parentheses) is independent of site structure, and does not change between content-based and URL-based views.

Adding new Content-Based view rules

Instructions and examples of adding rules to define the site's logical structure.

Before you begin

In Content-Based view of the Scan Configuration dialog box (Explore > Content-Based), you can define content-based structure using two types of definition:

- Logical Content Path (such as breadcrumbs)
- Custom (use regexps to define your own custom nodes)

Doing this enables AppScan to display a logical application tree, instead of listing large parts of the site under a single node. When more than one rule is defined, AppScan attempts to match each URL to the rules in the order they are listed. As soon as it finds a match it includes the URL in the Content-Based tree following that rule and proceeds to the next URL.

Procedure

1. Add new definitions by clicking  (or edit existing definitions by selecting a definition and clicking ). Refer to the table below for field descriptions.

Setting	Description
Name	Name for this rule.
Description	(Optional) Description for this rule.
Rule Type	Select Logical Content Path or Custom. The remaining fields vary depending on your selection.
<i>Logical Content Path</i>	
Content Path	Regexp that matches the content path in the HTML.
Separator	Regexp that matches the hierarchy separators in the content path in the HTML.
Node Display Name	Regexp that matches the node names (that appear to the application user and that will be used in the Application Tree).
Condition	(Optional) Regexp that defines the responses that are to be included in this node location.
<i>Custom</i>	
Node Location	Use standard URL format to describe where nodes that fulfil the condition for this rule should be placed in the application tree (for example: /Home/Buy/Books). If the path does not exist it will be created.
Condition	Regexp that defines the responses that are to be included in this node location. If the page content matches this condition it will be included in the Content-Based tree.

Note: Fields that require or accept a regular expression have the Expression Test button: , which opens the Expression Test PowerTool, to help you verify the syntax of the regular expression.

If you need additional help you may find the following link useful: <http://www.regular-expressions.info/quickstart.html>

2. Use the up/down arrows to arrange the definitions in the order they are to be applied.
3. Verify that the check box next to each rule that is to be applied, is selected.
4. Click **OK** to save changes.

Example

The following tables shows sample content for the two rule types.

Table 6. Content-Based View rule

Setting	Sample	Meaning
Content Path	<title>(.*?)</title>	All text between the title tags will be used as the basis for the node names.
Separator	[:\->]	These five characters will be treated as separators. Therefore: <title>Home:Accounts</title> and <title>Home:Plans</title> will appear in the application tree as a parent node Home with two child nodes: Accounts and Plans
Node Display Name	^\s*(.*?)\s*\$	Spaces before and after the text will be removed when naming the node.
Condition	Log out	Only pages containing the words "Log out" will be included in the application tree under this rule.

Table 7. Custom rule

Setting	Sample	Meaning
Node Location	/root/child/grandchild	All pages that fulfil the Condition are added to the application tree under /root/child/grandchild Note: If the parent nodes in the specified branch do not exist, they will be created.
Condition		As this field is empty, all content to which the rule is applied will be added at this node location. Therefore, if this rule appears after the rule above, all pages that do not contain the words "Log out" will be added under the node /root/child/grandchild

Glass Box view

Glass Box view of the Configuration dialog box.

Glass box scanning uses an agent installed on your application server. This agent monitors server-side activity during the scan, collecting source-code information and other data. This results in a faster and more accurate scan. The relevant glass box agent for the configured starting URL is selected by default, and both of the glass box scanning functions are enabled.

Glass box scanning can discover hidden URLs in the Explore stage, and additional issues and information during the Test stage.

Setting	Details
Use this glass box agent	If the glass box agent has been installed on your application server, and defined in AppScan, you can select it for use in the scan. If you have entered a Starting URL, AppScan attempts to select the appropriate agent automatically. When an agent is selected, AppScan attempts to connect to it, and indicates whether this was successful. Note: If you select an agent and get the message "Credentials needed", check that the credentials supplied in Tools > Glass Box Management are correct. If the required server does not appear in the drop-down list, you can define it by clicking the Glass box agent management link. Restriction: Only one glass box agent can be selected for use in a scan. If the application being scanned has more than one server, you must scan using each server agent separately.
Use glass box in the Explore stage	(Selected by default.) This function can increase coverage of the site, by examining the server-side source code for the presence of parameters that affect the behavior of the server, but do not appear in the response. Example server-side code: <pre>String debugOn = request.getParameter("debug"); if (debugOn == "true"){ response.getWriter().println(SECRET_SERVER_DATA); }</pre> In this example the developer has left the parameter "debug" in the code. It does not appear in any link on the site, but if an attacker were to send a request containing it, SECRET_SERVER_DATA could be obtained.
Use glass box in the Test stage	(Selected by default.) Select this check box to send glass box tests during the Test stage of the scan. This function can verify the success or failure of certain tests, such as Blind SQL Injection, with greater accuracy, and also reveal the existence of certain security issues that cannot be detected by black box techniques.
Skip equivalent black box tests	(Cleared by default.) This means that both glass box tests and black box tests for the same vulnerability (WASC Threat Classification) are sent. This is because although the glass box tests are generally both more accurate and give more detailed results, occasionally a glass box test may fail while the equivalent black box test succeeds. If the results for your application are unchanged when black box tests are skipped, you can reduce scan time by selecting this check box.

Note: By default the two main check boxes are selected. *Deselecting* both of them will disable glass box scanning.

See also:

"Glass box scanning" on page 144

"Installing the glass box agent" on page 145

"Defining the glass box agent in AppScan" on page 157

"Scanning with glass box" on page 159

Communication and Proxy view

Communication and Proxy (if AppScan needs a proxy to access the tested application) view of the Configuration dialog box.

Setting	Details
<i>Communication</i>	
Number of Threads	Set the maximum number of requests that will be sent simultaneously. By default this is set to its maximum possible value of 10. If your site does not allow simultaneous threads reduce this number to 1.
Adjust timeout automatically	In some case you may be able to reduce the length of the scan by selecting Adjust timeout automatically during the scan . When selected AppScan reviews and adjusts the timeout automatically throughout the scan, which may significantly reduce scan length.
Timeout	If the above check box is cleared, you can set the time limit (in seconds) for AppScan to wait for a response from the web server. By default, a timeout of 10 seconds is used.
Request rate limit	By default, AppScan sends its requests to the site as fast as possible. If this overloads your network or server, or if your site limits users to a maximum request rate, select this check box and reduce the limit. • If Action-Based Explore is used (Config > Explore Options > Explore method), this setting affects only the Test stage of the scan • If Request-Based Explore is used, this setting affects both the Explore and Test stages of the scan
<i>Proxy</i>	
Proxy settings	If AppScan needs a proxy to access your application, configure it here. By default AppScan is configured to use the Internet Explorer proxy settings. Use Internet Explorer proxy settings (Default) Select this to use the Address and Port of the Internet Explorer connection. If authentication is required, add username, password and domain below. Don't use Proxy Select if you do not want to allow AppScan to use a proxy. Use custom proxy settings Select to define your own proxy settings. Type in the address and port of for the proxy. If authentication is required by the proxy, add username, password and domain below.

HTTP Authentication view

HTTP Authentication view of the Configuration dialog box.

Use this view to configure platform authentication information and client-side certificate, if required. AppScan supports multiple certificates from the user's personal store, or a single certificate (per scan) in PKCS#12 (PFX) format.

Tip: PEM certificates are not supported, but you can convert them to PFX (see “Convert a PEM Certificate to PFX/P12 format” on page 95).

Setting	Details
HTTP Authentication	If your site requires Basic, Digest, NTLM, Negotiate, or Kerberos HTTP authentication, enter the Username, Password and Domain for AppScan to use during scanning here.

Setting	Details
Client-Side Certificate	If your site server uses client-side certificates to verify user identity, AppScan will need them for scanning. Don't use certificate (Default) PFX/P12 Use a single PFX or P12 certificate. Click  to select Certificate File, and type in the Password. Installed certificates (including smart card) Use certificates installed on this machine. - If you know which certificate is needed, or if you are using a smart card, it is preferable to add it manually: 1. Leave the Identify required certificates automatically check box unselected. 2. Click Add. 3. Select certificate (or certificates) from the Windows Personal Certificates Store (for the current user). 4. Click Add. The certificates are added to the table. 5. If the certificate requires a PIN, double-click in the column and type it in. - If you are not sure which certificates are needed: 1. Select the Identify required certificates automatically check box. 2. If a PIN is required and you know it, type it into the PIN field, otherwise you will be prompted to enter it when you start the scan. Tip: Smart cards that require a PIN may get locked out. If possible, add the certificate manually. Note: For most smart card certificates the card must be available in the reader during scanning.
Connection Status	Shows the status of the connection with the Starting URL configured in "URL and Servers view" on page 40.

Convert a PEM Certificate to PFX/P12 format

PEM certificates are not supported, they must be converted to PKCS#12 (PFX/P12) format.

About this task

This conversion can be done using an external tool such as OpenSSL, as described below.

Procedure

1. Go to <https://www.openssl.org/community/binaries.html>
2. Download and install version 1.0.1p.
3. Run the following command format from the OpenSSL installation bin folder.

```
openssl pkcs12 -export -out Cert.p12 -in cert.pem -inkey key.pem -passin pass:root -passout pass:root
```

Test Policy view

Test Policy view of the Configuration dialog box shows details of the current test policy.

The number of possible AppScan tests for a site can reach the thousands. Rather than manually filter the large number of tests and test variants, you can set a general policy for the type of test you do, or do not, want to be run on your application.

Use Test Policy view to view, edit and manage test policies, that define which tests are included in scans, and to define the policy for the current scan.

Tests are grouped and listed in the upper of the two panes. The Advisory and Fix Recommendation for the selected test appear in the lower pane.

In Test Policy view you can:

- View details of the current policy
- Edit the current policy to create a User-Defined Test Policy of your own
- Import a predefined policy, or a previously saved user-defined policy

Field/Pane/Option	Details
Test Policy	Shows the name of the current Test Policy. Tests are grouped and listed in the upper of the two panes. The Advisory and Fix Recommendation for the selected test appear in the lower pane.
Grouping method	Use the drop-down list to select a grouping method for the tests in the upper pane.
Filter	Use the drop-down list to filter the tests in the upper pane. You can choose: All, DAST Only (Dynamic analysis), IAST Only (Glass box runtime analysis), or SAST Only (Static analysis).
Search	Typing text into the Search field will display only tests that contain the search string. The Magnifying glass drop-down list lets you define whether to look for the string in <i>all</i> test fields, or only specific ones (such as Test Name or CVE ID).
Export	Click to save the current Test Policy so you can load it on another occasion.
Import	Click to load a predefined or user-defined Test Policy (see “Importing a Test Policy” on page 98).
Policy description	The upper-right pane shows the description of the current policy. For user-defined policies this field can be edited.
Test pane	The upper main pane lists all AppScan tests that meet the filter/search criteria. For each test the following information is listed: Name, Variant ID, CVE ID, CWE ID, Severity assigned to the issue (and whether the severity is CVSS or user-assigned), XFID (X-Force ID), Type, Invasiveness, and WASC threat classification. You can Sort tests by some of these fields, by clicking on the column header. Tests whose check box is selected are included in the current policy. You can edit the policy by selecting/deselecting tests (see “Editing a Test Policy.”
Update Settings link	This link opens a dialog box that lets you define which types of test can be added to this policy when new tests are added to the database. For details see “Test Policy Update Settings” on page 98
Advisory and Fix Recommendation tabs	The lower main pane shows the Advisory and Fix Recommendation for the selected test. You can Edit any Advisory to your own specifications, or Reset to Default an Advisory that has been edited (see “Editing advisories and fix recommendations” on page 99).
Policy files	Load an existing Test Policy by clicking one of the Recent Policies , or Predefined Policies , or by clicking Browse... and browsing to the required policy.

Editing a Test Policy

Test Policy view can be used to fine-tune your selected test policy.

About this task

You can fine-tune the current test policy by adding or deleting tests, and also export the changed configuration as a user-defined test policy for future use.

Procedure

1. In the Scan Configuration dialog box, click **Test Policy** (or in **Scan Configuration Wizard > Test Policy**).

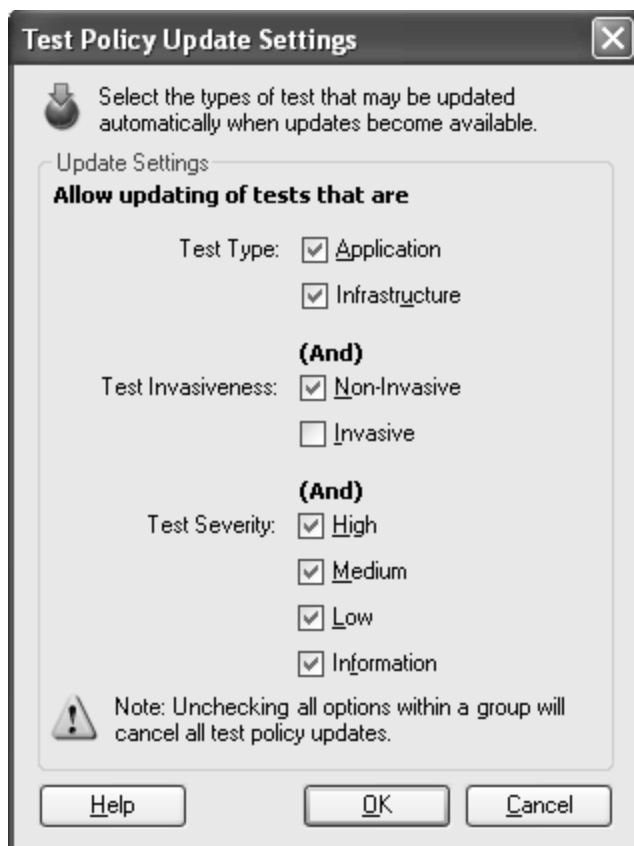
The upper area lists all AppScan tests and indicates which are included in the current scan (check box selected).

2. Include/exclude tests or variants by selecting/deselecting the check box(es). (To view individual variants, click the + icon next to a Test Name.)

Note: For each test the following information is listed: Name, Variant ID, CVE ID, CWE ID, Severity assigned to the issue (and whether the severity is CVSS or user-assigned), Type, Invasiveness, WASC threat classification, and XFID (X-Force ID). You can **Sort** tests by any of these fields, by clicking on the column header.

Note: The **Search** facility lets you search for tests using free text search.

3. In the Information field at the top right of the dialog, you can edit the description.
4. New tests are continually being added to AppScan's database of tests. By default, all new tests except Invasive tests are added to all user-defined test policies. However, you can define which groups in your policy will be updated: Click **Update Settings**, select/deselect check boxes in the **Test Policy Update Settings** dialog box as required, then click **OK**.



The dialog box contains three groups: Test Type, Test Invasiveness, and Test Severity. Only the tests that belong to a selected category *in all three groups* will be added to the current policy, when new

tests are added to your AppScan database of tests. For example: If you select **High Severity**, but deselect **Invasive**, high severity, invasive tests will *not* be added to this policy when updates become available.

5. You can optionally give the scan a name and save it for future use (click **Export**, and save in **.policy** format).
6. Click **OK** to save the changes to the current Test Policy.

Importing a Test Policy

About this task

You can load recent, predefined and user-defined test policies.

Procedure

Do one of the following:

- In the Policy Files area select one of the listed **Recent Policies** or **Predefined Test Policies** (see “Predefined Test Policies”)
- To open a user-defined policy that is not listed in the Policy Files area, click the **Import** button in the upper part of the screen (or click **Browse** in the Policy Files area).

Results

The selected policy loads, and its name and description appear in upper part of the dialog box.

Predefined Test Policies

The Policy Files pane at the lower left of Test Policy view lets you select one of the recently used policies, or one of the predefined ones. The predefined policies provide a range of useful policies for common requirements.

Policy Name	Description
Default	Includes all tests except invasive and port listener tests.
Application-Only	Includes all application level tests except invasive and port listener tests.
Infrastructure-Only	Includes all infrastructure level tests except invasive and port listener tests.
Third-Party-Only	Includes all third-party level tests except invasive and port listener tests.
Invasive	Includes all invasive tests (tests which might affect the server's stability).
Complete	Includes all AppScan tests.
Web Services	Includes all SOAP related tests except invasive and port listener tests.
The Vital Few	Includes a selection of tests that have a high probability of success. This can be useful for evaluating a site when time is limited.
Developer Essentials	Includes a selection of <i>application</i> tests that have a high probability of success. This can be useful for evaluating a site when time is limited.
Production Site	Excludes invasive tests that might damage the site, or tests that might result in Denial of Service to other users. Note: For more information about scanning a live site, see “Scanning live production environments” on page 294.

Test Policy Update Settings

This dialog box opens from Scan Configuration > Test Policy view.

Select which types of tests (if any) will be added to the current Test Policy when new tests are added to your AppScan tests database. Tests are updated when you load a scan or a template, or when you import a test policy file.

There are three groups: Test Type, Test Invasiveness, and Test Severity. Note that only tests that belong to a selected category *in all three groups* will be added to the current policy when new tests become available. For example: If you select **High Severity**, but deselect **Invasive**, high severity, invasive tests will not be added to this policy when updates become available.

Editing advisories and fix recommendations

About this task

You can change the text that appears in the Advisory and Fix Recommendation tabs of the scan results for a test (see “Advisory tab” on page 194). (When you do this the edited file is saved in .xml format on your computer. The exact location is configured in Tools | Options | Preferences Tab, see “Preferences tab” on page 236.) The Advisory and the Fix Recommendation text are stored in the same file. (You can restore the original text at any time using the *Reset to Default* button.)

Note: If your scan is sent to another computer, the original, *default* advisory for the issue, and not your edited advisory, will be displayed (unless you send the .xml file to the other computer, and have the user save it to the Custom Advisories folder *of that computer*; see “Preferences tab” on page 236).

Note: If IBM releases an updated version of an advisory/fix recommendation that you have edited, the updated version will not appear in your AppScan interface (unless you discard the changes you made by clicking the *Reset to Default* button).

Procedure

1. Select the required test.

The advisory and fix recommendation for the selected test appear in the lower pane.



2. Click the **Edit** icon above the Advisory tab. (The advisory and fix recommendations for any particular test are stored in the same file. Therefore it does not matter which of the two tabs is on top when you click **Edit**.)

The advisory file is opened in the embedded XML editor.

3. Edit the text as required, then click **OK**.

Note: If you use invalid XML syntax, or tags that are not part of the advisory file schema, a warning appears and the changes are not saved.

The XML Editor closes, and the changes appear in the current advisory.

Restore default advisories:

About this task

When the advisory for a particular test has been changed by the user, the Restore to Default button is active when that test is selected.

Procedure

Select the relevant test(s) in Test Policy Manager, and click **Restore to Default**.

Test Options view

Test Options view of the Configuration dialog box.

This view lets you configure various settings that affect the length and thoroughness of the scan. However, the default settings are sufficient in most cases.

Setting	Details
Test Options:	
Use Adaptive Testing	AppScan can send many thousands of tests to a site. However, in order to reduce scan time, it can send preliminary tests that intelligently determine which are the appropriate tests to send and which can be dispensed with. This is "Adaptive Testing" and it can greatly reduce scan time, without sacrificing efficiency. Clear this check box if you want AppScan to send <i>all</i> its tests to the site.
Allow Multiphase Scanning	AppScan analyzes responses to the tests that it sends your application. From this analysis, AppScan frequently discovers additional content, such as links that were invisible on the first "phase" of the scan. Multiphase scanning enables AppScan to repeat the Explore and Test stages on this newly detected content. (The additional phase is usually shorter, as it involves the new links only.) Multiphase Scanning is configured by default to allow a maximum of 4 scan phases. Note that multiphase scanning applies only when you run a Full Scan. If you use the Explore Only and Test Only functions, the result will be a single-phase scan.
Send Tests on Login and Logout pages	It is recommended to allow AppScan to test login and logout pages, unless your application locks out users who provide illegal input, or the application flow would be altered by AppScan testing these pages.
Do not send session identifiers when testing login pages	(Active only if previous check box is selected.) It is recommended to leave this check box selected, since session identifiers could limit test success when testing login pages. Clear it only if you are sure that valid session tokens are necessary to test your login pages. Note that even when this check box is cleared, <i>some</i> tests are still sent with session identifiers, to prevent false positive results.
Analyze results for inadvertently-triggered issues	When selected, AppScan analyzes each test response for additional security issues over-and-above the specific issue tested for. Deselect this option if the application is very large, or if scans produce a large number of false-positive results.
Include all variants of each issue	(Active only if previous check box is selected.) When selected, AppScan analyzes all variants of each inadvertently-triggered issue; when deselected, only one variant per issue is analyzed. Selecting this check box is not usually necessary, and can significantly increase scan time.

Setting	Details
Test for cookie security issues in form submission requests only	When selected (default), AppScan will submit cookie related tests only on cookies used in form submission requests. For higher accuracy (but increased scan time), deselect this check box, and AppScan will submit cookie tests on all relevant HTTP requests.
Save Non-Vulnerable Test Variant Information	During a scan, AppScan sends many thousands of test variants to the site it is testing. The responses to many of these indicate that they do not pose a security threat of any kind, and by default AppScan discards all these "non-vulnerable" results, considerably reducing the volume of the result data. If you select this check box AppScan will save all non-vulnerable variants. A warning will appear this option may reduce AppScan performance and significantly increase the disk space required. For more details see "Non-vulnerable variants" on page 203
Enable JavaScript Security Analysis	Enables static JavaScript analysis that can detect a range of client-side issues, particularly DOM-Based Cross Site Scripting, in client-side JavaScript code. For more details see "JavaScript analysis"
Issue Management:	
Apply previous noise classifications to this scan	If in a previous scan you classified one or more issues as "Noise" (not relevant to your application), the same settings are automatically applied to future scans, unless you deselect this check box. For more details see "Issue state: Open or Noise" on page 187

Note: If you make changes to Test Options *after* a scan, you may be prompted to re-scan, as not all changes can be applied to existing results.

JavaScript analysis

This section describes the basics of how static analysis of JavaScript works, and how the results are presented.

JavaScript analysis results

AppScan performs static analysis of JavaScript source code, therefore the issues found include source-level trace information showing the vulnerable source code. You can view this trace information in the Issue Information tab of the Detail pane.

Highlighted and numbered lines in the code show, step-by-step, from source to sink, how untrusted data that enters the application gets propagated until it is used in an insecure way.

```

http://demo.testfire.net/disclaimer.htm?url=http://www.netscape.com
19      }
20
21      var iPos = document.URL.indexOf("url=")+4;
22  ❶    var sDst = document.URL.substring(iPos,document.URL.length);
23      </script>
24      </head>
25      ...
31      <td>
32          <p>This hyperlink allows you to access a third party website:
33          <br /><br />
34  ❷    <b><script>document.write(unescape(sDst));</script></b>
35          <br /><br />
36          Please read the privacy policy of the linked website, which
37          may differ from the privacy policy of the Altoro Mutual website.

```

AppScan uses an advanced "string analysis" technology to eliminate false positives and classify issues more accurately.

The principals of string analysis

AppScan applies two stages for analyzing each page: taint (or data flow) analysis and string analysis. In the first, AppScan looks for a trace from source to sink that does not go through a sanitizer. If this is found, AppScan validates it in a second step, using string prefix analysis (SPA), a variant of string analysis.

In certain cases, where the source is a URL, or part of it, a trace is only exploitable when the host and path parts of the URL can be controlled by an attacker at the point of the sink. SPA validates these cases by tracking the string operations that manipulate the source URL. SPA assumes that the host and path of the URL are not controlled by the attacker, while the query and/or anchor part are:



At the point of the sink, SPA checks whether the host and path parts of the URL can be controlled by an attacker. If it is determined that the host and path parts are both fixed and not controlled by an attacker, the issue is eliminated.

If the redirect target begins with the prefix javascript: or mailto: the issue is classified more accurately as "DOM Based Cross-Site Scripting" or "DOM Based Email Spoofing" respectively.

Privilege Escalation view

Privilege Escalation view of the Configuration dialog box lets you compare results for different user levels.

About this task



AppScan can refer to scans that were run using different user privileges, in order to investigate the extent to which privileged resources are accessible to users with insufficient access permissions. This can be done in two ways:

- **By comparison with a higher privileged user:** You point AppScan to scan results that were produced using a higher level of access permissions than the current scan. During the scan AppScan attempts to access the additional links that were available to the higher level user, using the current (lower level) access permissions. The scan results indicate where these attempts were successful.
- **By comparison with a non-authenticated user:** You point AppScan to scan results that were produced *without* user authentication. AppScan then runs a scan using the current authentication and notes the new links it accesses. It then logs out and attempts to access these new links *without* authentication. The scan results indicate where these attempts were successful.

Important: Scans being compared must have the same scan configuration, and equivalent Explore data. For example, if the site was explored manually before testing in one of the scans, the *same* Manual Explore must be performed before the Test stage in the scans being compared with it.

Procedure

1. **(To compare with a higher privileged user:)** In the upper area ("Higher Privilege User Tests"), click  , and browse to a scan that was run with higher access permissions than the current scan.

2. Click **Open**.

3. Type in a name that represents the authentication level used in the scan (for example Guest or Administrator), then click **OK**.

The selected scan is added to the list, and its role (e.g. Admin, Operator, Visitor) appears in the left column.

4. Repeat these steps to add scans with different authentication levels, as required.

Note: You may add more than one scan for Higher Privilege User tests, one for each role. For example, if the current scan is configured with the Username and password of a normal user, you could add two scans to this list: one that was run with Administrator permissions, and one that was run with Supervisor permissions. The results will indicate which user's resources were found to be accessible to the Normal User.

5. **(To compare with a non-authenticated user:)** You can also optionally load the results of a scan run *without* authentication. To do this, in the lower area click  , and browse to the scan results.

Malware view

Malware view of the Configuration dialog box is for configuring malware testing.

A test for malware consists of two modules with various configuration options. They are configured using this view. Malware test configuration does not affect regular scans.

Setting	Details
<i>Malware Tests</i>	Check for links to malicious external web sites When this check box is selected, AppScan checks the application for unwanted external links, during the scan. An Internet connection is required.
<i>Scope</i>	You can create a list of regular expressions that define URLs you want to <i>exclude</i> from malware testing. Use  to add a regular expression, and in the dialog box that opens you can click , to open the Expression Test PowerTool which can help you verify the syntax of your regular expressions. If you need additional help writing regular expressions you may find the following link useful: http://www.regular-expressions.info/quickstart.html
<i>Additional Results</i>	The check boxes in this section let you widen the scope of the malware analysis. When AppScan analyzes results for links classified as "Malicious", it can also report on links classified as Unwanted and Harmless, as well as those Unclassified by ISS, that may nonetheless be problematic. Select any issue types you want included in the results. Note: Some links of issue type "Unclassified" may in fact be malicious. Note: Including Harmless links may increase the size of the results significantly.

Scan Expert view

Scan Expert view of the Scan Configuration dialog box.

Scan Expert runs a short Explore stage before the main scan starts. It analyzes the results of this Explore to evaluate the efficiency of the settings you have configured and suggest changes as necessary. The changes it suggests can then be implemented before the main scan starts, to optimize its efficiency. (See "Scan Expert" on page 143 for more details.)

The settings in this view let you decide how thoroughly Scan Expert explores, whether configuration changes are implemented automatically or manually, and which "modules" of the configuration are included in the evaluation.

Note: By default Scan Expert runs automatically before every full scan. You can check or change this setting in **Tools > Options... > Preferences**.

Setting	Details
Scan Expert Behavior	Evaluation Limit: Define how quickly (or deeply) Scan Expert explores your application. Deeper exploration may produce a more complete evaluation. You can define a maximum number of URLs to be explored during the evaluation, a time limit, or both. If you limit the evaluation by <i>both</i>, the evaluation will stop when the <i>first</i> of the two limits is reached. (Note: The scan may not always stop exactly at the limit, but shortly after it.) After Evaluation: Define whether Scan Expert's recommendations are implemented - and the scan started - automatically, or whether the system waits for you to review the recommendations and implement manually those you approve. (Note that some modules cannot be implemented automatically, so if you select "auto-enabled" the settings for these modules will not be evaluated.)

Setting	Details
Modules	The Modules pane lists all modules that Scan Expert is able to evaluate. Select/clear check boxes so that the selected modules are those you want included in the evaluation. Use the up/down arrows to rearrange the order of the modules (this is for convenience only, and does not affect they way they are run). For details, see “Scan Expert modules.”

See also: “Scan Expert” on page 143

Scan Expert modules

A table of Scan Expert modules and their descriptions.

The list is arranged in the order that the modules run. If there is a conflict between the recommendations of two or more modules, the one *lower* in the list prevails.

If you add a module of your own, it is added to the bottom of the list, so that if there are conflicts with existing modules, your new module will prevail.

If necessary you can use the up and down arrows to promote and demote modules in the list.

Module	Description
Parameter-Based Navigation Site Detection	Checks if the application uses parameter-based navigation, and helps configure the scan correctly for this.
Server Authentication	Checks for missing or failed NTLM and HTTP Authentication details.
Proxy server	Checks that no Proxy Connection or Authentication errors occurred.
Client-Side Certificate	Detects whether a client certificate is required.
Missing Recorded Login	If the scan is configured to detect login pages automatically, alerts user when no such page is detected.
Detect AJAX Frameworks	Looks for markers of common AJAX Frameworks, which require JavaScript execution.
Detect Session IDs	Checks whether Session IDs were found for the automatic or recorded Login Sequence.
Unfilled Forms	Detects unfilled forms.
Detect Untested Servers	Detects links to servers other than the Starting URL server, in the same domain as the Starting URL. If such links are found, the module will recommend adding to the list of Additional Servers and Domains (Configuration > URL and Servers).
Communication Timeout	Evaluates the accuracy of the Communication Timeout configuration.
Number of Threads	Assesses the ability of the application to withstand multiple scanning threads
Detect Error Pages	Detects custom error pages in the application.
Case Sensitive Path	Checks web application server for case sensitivity.
Suspicious URL	Detects suspicious URLs that should probably be excluded from the scan.
Logout Link Missing	Detects Logout pages.
Depth Limit	Checks if links are unreachable, due to depth limit. If so, it may suggest that depth limit be increased or disabled.
Detect Flash Objects	Detects the use of Flash within the web application.
web Services	Detects the existence of web services in the application.

Module	Description
Check Environmental Settings	If any global questions exist in the Environmental Settings tab, alerts the user if none of these questions have been answered.
WebSphere Portal Detection	Checks if the tested application is based on WebSphere Portal.
Scan Expert Evaluation	Checks whether Scan Expert evaluation was performed successfully.
Hacme Bank Detection	Checks if tested application is McAfee Foundstone Hacme Bank
WebGoat Detection	Checks if tested application is OWASP's WebGoat

“Scan Expert view” on page 104

Advanced Configuration view

The **Advanced** tab of the **Scan Configuration** dialog box is used to change advanced registry settings that affect specific scans (Scan > Scan Configuration > Advanced tab), and it should only be used by experienced AppScan users, or when instructed to do so by the support team to troubleshoot a problem.

Tip: Advanced registry settings that affect AppScan *generally* (rather than a specific scan) are located in the **Advanced** tab of the **Options** dialog box (Tools > Options > Advanced tab).

Note: Each setting has an ID that you can use when discussing settings with the Support team. The items in the grid can be sorted by Name or ID, by clicking the relevant column heading.

Note: Where the default setting is a regular expression, deleting it altogether will result in the setting being treated as undefined (and not as a regular expression that includes everything).

Name	Description	Possible use cases
<i>Action-Based:</i>		
Login playback browser	When you record the login sequence, the embedded AppScan browser is always used to make the action-based recording. However, you can configure the browser that AppScan uses when <i>playing back</i> the recording during the scan. Options are: • 0 = Internet Explorer (embedded version) • 1 = Chromium • 2 = Chrome • 3 = Firefox Default: 0	
Multi-step playback browser	When you record a multi-step sequence, the embedded AppScan browser is always used to make the action-based recording. However, you can configure the browser that AppScan uses when <i>playing back</i> the recording during the scan. Options are: • 1 = Chromium • 2 = Chrome • 3 = Firefox Default: 1	
Multi-step playback no-interaction timeout	The no-interaction timeout (in seconds) for stopping playback of a multi-step operation. Default: 10	

Name	Description	Possible use cases
Timeout for single login attempt	The time (in seconds) that AppScan waits for the browser to replay a single action-based login attempt before forcing the browser to close. Default: 120	
<i>Communication:</i>		
Accept-Language request-header value	The string sent for the Accept-Language header in all HTTP requests. If not defined by the user, AppScan will use the value that was sent by the browser the first time in this scan that the user opened it to record the login procedure, a multi-step operation, or to view a page. Note: If you change the default browser, refer to the conditions listed in "Changing the default browser" on page 304 Default: en-US	During the Explore stage AppScan might receive an unexpected response, due to the Internet Explorer header value. In such cases you should check which value should be used in the Accept-Language header when interacting with the site, and define it in this setting (or in Internet Explorer).
Custom headers	Allows you to define custom headers to add to all requests that AppScan sends the site. Default: empty	If your site expects specific header content (for example, because access to the site is via a specific client or browser plug-in), define the header(s) here. Each header must be preceded by a delimiter. Between the header and the value must be a colon and a single space. Format: delimiter header colon-and-single-space value Example1: ;Header: Value (In this example the delimiter is ;) Example2: ,Header1: Value1,Header2: Value2 (In this example the delimiter is ,)
Force an HTTP request without parameters to every form action	In some cases, server-side logic may behave differently when a form submission <i>without</i> parameters is received. When set to True, AppScan will send an additional request, without parameters, to every form. This may result in the return of custom error pages with links to additional web pages and functionality. Default: True	If you notice, when viewing traffic during the scan, that form submissions without parameters cause timeouts or crash the application, you may decide to set this option to False.
GSC SSL port	This setting defines the port number used in GSC for SSL communication. Default: 443	In the case of links supplied through GSC Explore, AppScan identifies HTTPS based on this port number. If your application uses a different port for SSL communication, define it here. If the correct SSL port is not defined, AppScan will send all tests as HTTP.

Name	Description	Possible use cases
Include AppScan debug headers in all requests	When set to True, an HTTP header is added to all requests sent by AppScan to the site. The header name is "X-AppScan-Debug", and its value includes information about AppScan's reason for sending this particular request (Explore, Test, Login Playback, Server Down Check, and so on). Default: False	Configuring the scan to send "X-AppScan-Debug" headers can be useful in tracking AppScan traffic in external tools such as web debuggers, proxies, analyzers and sniffers. Note: Some sites may reject any requests that include special headers such as this.
Maximum response length	AppScan truncates long responses to avoid memory consumption issues. This setting defines the maximum allowed response length in Megabytes. Longer responses are treated as errors. Default: 8	If AppScan seems to miss links or get out of session, and the application is known to send long responses, increasing the maximum response length may solve the problem.
Remove 'Accept-Encoding' header	AppScan does not support all encodings, and removes the encodings it does not support. If this setting is enabled AppScan will remove the entire header and not just the encodings it does not support. Default: True	If the server rejects AppScan's requests, returns unexpected responses, or AppScan is unable to maintain session, you should check the traffic log and compare the requests AppScan sends with those of your usual browser. If the Accept-Encoding header is different or missing in your browser, you should enable this setting.
Reuse server connections	By default, AppScan closes TCP connections after use, since open connections, and saved data, may affect scan results. When set to True, AppScan leaves connections open after use, and attempts to reuse open connections whenever possible. Default: False	If there are network resource exhaustion errors on the web server, changing this setting to True may solve the problem.
Security package order	AppScan supports Basic, Digest, NTLM, Negotiate, and Kerberos HTTP authentication. If you want to force AppScan to use or not use a specific method, or apply an order of preference for method selection when the site/proxy allows more than one, you can edit this value. For example, if you want to allow only NTLM and Basic, and prefer to use NTLM if available, edit this string to: ntlm, basic Default: basic, digest, ntlm, negotiate, kerberos	If your site uses a specific authentication method and AppScan is denied access, defining the required method as the only method can solve the problem. If you want to test your site with specific methods - say Basic and NTLM - you could configure one scan with Basic only, and another with NTLM only.
Slash Normalization	Normalize URLs by replacing two or more consecutive slashes with a single slash. Default: True	If your site URLs utilize consecutive slashes, deactivate this setting.

Name	Description	Possible use cases
Treat error response as valid	AppScan treats error pages differently to regular pages (for example, by not parsing for links). This setting lets you tell AppScan to treat error pages as regular pages for the starting URL only, or always. When set to 0, AppScan treats all error responses as invalid. When set to 1, AppScan treats all error responses to the starting URL (4xx and 5xx) as valid When set to 2, AppScan treats all error responses as valid for both regular pages and the starting URL. Default: 0	If your starting URL response is an error page, change the setting to 1. If you want the scan to extract data from error pages, and test them, change this setting to 2. Note that changing the default setting is likely to affect performance.
<i>Flash:</i>		
Coverage	Defines the "coverage level" for the scan. 1 = faster scan; 2 = more complete coverage Default: 1	For more complete (but slower) coverage of Flash content, this setting can be changed to 2.
Exclude Instances	Define problematic GUI instances to be excluded from Flash scan. Use this option only when instructed to by Support.	If scan fails or is endless due to repetitive operations. Instances should be comma separated.
File download string	In order to be able to scan Flash movies efficiently it is important that AppScan can identify controls that result in the downloading of a file. Define these here. Note that AppScan will pause after clicking a control containing the strings defined here, for the "Long Operation wait time", even if these strings are not included in the "Long operation string" setting. Default: Download	If the movie includes file downloads, they should be defined here. If there are more than one, separate them with semicolons. Example: Download;Save;Copy
File upload path	Path to a file that AppScan can upload to the application when the movie includes an option to upload a file. Default: empty	If the Flash move includes an option to upload a file, define the path to the file, including filename, and make sure the file exists there. Note that you must also define the "File upload string".
File upload string	In order to be able to scan Flash movies efficiently it is important that AppScan can identify controls that allow the user to upload files. Define these here. Note that AppScan will pause after clicking a control containing the strings defined here, for the "Long Operation wait time", even if these strings are not included in the "Long operation string" setting. Default: Upload;Browse	If the movie allows the user to upload files, the text of the links or controls that do this should be defined here. If there are more than one, separate them with semicolons. Example: Upload;Browse;Add Note that the "File upload path" must also be defined.
Flash movie dependency exists	Determines that the site has a Flash movie that contains another Flash movie on which the first movie depends. Default: False	If the site contains a Flash movie that loads another Flash movie, and the first depends on the second, set this to True.

Name	Description	Possible use cases
Frame rate factor	When playing Flash, increase the frame-rate by this factor (1,2,3,4). Default: 4	By default AppScan speeds up the frame rate by a factor of four when playing movies during scans. If this is too fast to imitate real use, you can decrease this.
Load movie wait time	The length of time (ms) that AppScan waits for a movie to load before starting to explore it. Default: 1600	If AppScan starts to explore a movie before the movie has loaded, the results will not be accurate. Therefore, if your movie takes longer than the default time to load you should increase this setting.
Long operation string	After most operations the Flash browser waits the amount of time defined in "Regular wait time" before proceeding, to allow the movie to reach its new state. In the case of operations that are likely to take longer, and that could therefore result in the scan continuing before the new state has been reached, you should define "Long operations" (such as uploading files or logging in) for which the browser will pause for a longer time. Default: empty	Operations you might want to define as "Long operations" could be "Upload" or "Login" links. Note that in the case of a file upload, the link you need to define here is the one that actually results in the uploading of the file, not the link that lets the user <i>select</i> a file for uploading. If you enter two or more strings, separate them with single spaces.
Long operation wait time	Time (ms) for the Flash browser to wait for operations defined in "Long operation string" to complete and reach the new state. Default: 5000	If a "Long operation" has been defined, you should define its "wait time" here.
Time between samples	Minimum time (ms) to wait between "user operations" (such as mouse clicks), to allow the movie to reach its new state. Default: 160	If the movie has a lot of animation, the wait required between user-clicks or filling forms may be longer than the default setting. You can increase the amount here.
<i>General:</i>		
AppScan Browser script-error-popup suppression	Suppresses script-error popups in the AppScan built-in browser during action-based login recording and playback, manual explore, multi-step recording, and Show-in-browser. Default: False	If irrelevant error popup messages interfere with action-based login recording and playback, you can suppress them by setting this value to True. Note that other popups such as "HTTP Authentication" errors and "Install ActiveX control" prompts will also be suppressed.
Merge redundant tests	When set to True, AppScan sends only a single set of tests on two (or more) requests that are identical except for additional cookies. If set to False, all such requests will be tested separately. Default: True	Changing this setting to False can impair performance; do so only if advised to by Support.

Name	Description	Possible use cases
Proxy file extension filter	A regular expression that defines file extensions which will be removed from the list of URLs that is saved when you record a login, Manual Explore or Multi-Step operation. If you remove an extension from the regexp., URLs ending with that extension will not be filtered out of recordings. Default: "\.(zip Z tar t?gz sit cab pdf ps doc ppt xls rtf dot mp(p t d e a 3 4 ga) m4p mdb csv pp(s a) xl(w a) dbf slk prn dif avi mpe?g mov(ie)? qt moov rmi? as(f x) m1v wm(v f a) wav ra au aiff midi? m3u gif jpe?g bmp png tif?f ico pcx css xml)\$"	In rare cases where you need a particular kind of file, such as a CAPTCHA image file, included in your Login recording for reference, you could remove its file extension (in this case jp?g) from the regular expression.
Sanitize logs	Removes sensitive information from logs. Default: False	If you need to remove sensitive information from logs, activate this option and define the pattern to be removed in the "Sensitive information pattern" option. Note that changing this setting does not affect logs already generated.
Sanitize reports	Removes sensitive information from reports. Default: False	If you need to remove sensitive information from reports, activate this option and define the pattern to be removed in the "Sensitive information pattern" option. Note that changing this setting does not affect reports already generated.
Send all tests through GSC	AppScan can use GSC to send tests on some or all links that were found by GSC. 0 = Send only SOAP messages using GSC 1 = Use GSC to send all tests on links found by GSC 2 = Do not send any tests using GSC Default: 0	If you didn't define any special security settings when exploring your site in GSC, letting AppScan (rather than GSC) send tests during the Test stage will significantly reduce scan time. However, if during the Test stage many tests get no response, or unexpected error responses, the problem may be due to the difference between how GSC and AppScan send requests. Sending such tests using GSC may solve the problem.
Sensitive Information pattern	A regular expression that defines one or more groups that will be filtered out of logs and reports, if the Sanitize Logs or Sanitize Reports options are activated. Default: empty	If you need to remove sensitive information from reports or logs, activate the relevant option ("Sanitize logs" or "Sanitize reports"), and define one or more groups in a regular expression here. The sensitive text is replaced with: **CONFIDENTIAL 1**, **CONFIDENTIAL 2**, and so on.
JavaScript:		

Name	Description	Possible use cases
Automatic JavaScript Execution in Manual Explore	You can configure AppScan to extract <i>all</i> JavaScript-generated links on pages that were manually explored, even those that were not triggered during the Manual Explore. Default: False	This can be a way of increasing coverage when manually exploring.
Clear cache	If set to True, JavaScript execution (if enabled), will not cache requests sent during login. This may increase scan time and file size. Default: False	
Fetch external links	When JavaScript Execution is enabled, allow AppScan to fetch external links even if their server is not configured in AppScan as an additional server. Default: False	HTML pages frequently link to external JavaScript source files, such as Dojo and jQuery source files. If you want JavaScript Execution to access all relevant links discovered during the Explore stage, without adding all the servers to the list of Additional Servers and Domains that AppScan tests, activate this setting. Note that AppScan will fetch the link, but will not test it, or parse it for new links.
JavaScript and Flash filter	This regular expression defines pages that on which JavaScript will not be executed and Flash not parsed, during the Explore stage of the scan. (It does not limit JavaScript <i>parsing</i> .) Default: empty	If AppScan repeatedly crashes or freezes on a specific page containing JavaScript or Flash (this can be confirmed by reference to the Traffic Log), defining the page here will solve the problem.
JavaScript link pattern	AppScan uses a variety of patterns to identify links in JavaScript code. If your site uses unusual patterns they should be defined in this regular expression. Default: empty	If AppScan seems to miss links from your JavaScript code, and your site uses unusual JavaScript link patterns, define one or more patterns here to tell AppScan what to search for.
<i>Localization:</i>		
HTML encoding	Overrides the encoding defined in your site's HTML responses. Default: empty	If the content of responses in the scan results looks distorted, this may mean that: 1) The encoding method was not correctly identified by AppScan, or 2) The encoding method is incorrectly defined in your site's HTML To solve 1: Select the correct method in the Explore Options drop-down list. To solve 2: Enter the correct encoding method here.
<i>Parameters & Cookies:</i>		
Exclude redundant JSON parameters from testing	JSON content type body can contain multiple values of a single parameter that need not be tested individually. When set to True, AppScan attempts to identify redundant values and limit testing to a subset, reducing scan time. Default: True	If you find that a particular, significant parameter was not tested, change the setting to False.

Name	Description	Possible use cases
Exclude redundant XML parameters from testing	XML content type body can contain multiple values of a single parameter that need not be tested individually. When set to True, AppScan attempts to identify redundant values and limit testing to a subset, reducing scan time. Default: True	If you find that a particular, significant parameter was not tested, change the setting to False.
Track custom parameters in headers	This setting applies only to scans saved with AppScan v. 8.7.0.1 or earlier. In later versions the default behavior changed to True, and the setting is controlled for individual parameters and cookies in: Configuration > Parameters and Cookies > Parameter Definition > Tracking Options > Match: Header and Body (default) or Body only (see "Parameter definition" on page 69). By default AppScan (8.7.0.1 and earlier) searches for custom parameters only in the body of responses, not in their headers. If you change this setting to True, AppScan will search in headers too. Default: False Note:	If AppScan gets out of session due to changes to a parameter in the response header, changing this setting may solve the problem. Note that this may increase scan time.
Track dynamic parameters in Test stage only when inline content exists	Tracking dynamic parameters during the Test stage may result in performance problems. Therefore, by default, dynamic parameters are tracked during the Test stage only in responses with inline content. Default: True	Change this setting to False only if this kind of tracking is essential.
<i>Server Down Detection:</i>		
Check for "server down" in Explore	Enables the sending of heartbeat requests to check for "Server Down" during Explore stage. Default: True	If AppScan gets server-down errors during the Explore stage, and the server is not down, this may be due to the server blocking the frequent heartbeat requests. If AppScan frequently gets out-of-session during scanning, this may be due to the Starting URL being sent to the server as a heartbeat, without cookies. Deactivating this setting may solve the problem, but note that AppScan will not be able to verify server status.
Check for "server down" in Test	Enables the sending of heartbeat requests to check for "Server Down" during Test stage. Default: True	If AppScan gets server-down errors during the Test stage, and the server is not down, this may be due to the server blocking the frequent heartbeat requests. If AppScan frequently gets out-of-session during scanning, this may be due to the Starting URL being sent to the server as a heartbeat, without cookies. Deactivating this setting may solve the problem, but note that AppScan will not be able to verify server status.

Name	Description	Possible use cases
Explore stage reconnect attempts	When AppScan is about to finish the Explore stage but several tests failed due to "server down", and the server is still down, AppScan will try to connect to the server several times. Default: 5	If you know that your server is sensitive, or you see that the scan stopped due to a communication error while several tests failed due to communication errors, you should increase this number.
Request retry interval	Interval in seconds before resending failed requests (including failed heartbeat requests). Default: 1	If you know that you have a poor connection or an unstable server (which would result in false negative results, or reduced coverage), you can increase this interval to reduce the impact.
Request retry limit	Number of times to retry sending failed requests. Default: 2	Increasing this setting may result in a more efficient scan if your server is unstable or communication is poor.
Server down timeout	When AppScan is unable to connect to the server or gets out-of-session, this setting defines (in seconds) for how long AppScan will try to reconnect or get back into session before stopping the scan. Default: 185	If you have a slow connection, or your server takes a long time to reload after down time, you might want to increase this setting.
Server-down heartbeat interval	Interval in seconds between "server down" heartbeats. Default: 3 s Max: 60 s	If AppScan gets server-down errors during the scan, it may be due to a poor connection or unstable server. Increasing this interval may solve the problem.
Test stage reconnect attempts	When AppScan is about to finish the Test stage but several tests failed due to "server down", and the server is still down, AppScan will try to connect to the server several times. Default: 5	If you know that your server is sensitive, or you see that the scan stopped due to a communication error while several tests failed due to communication errors, you should increase this number.
<i>Session Management:</i>		
Ad domains	Regular expression describing common web advert domains. Requests sent to these domains when the login sequence is recorded, will be discarded. Default: ad\d.google syndication doubleclick\.net coremetrics\. webtrends\. 112\.2o7\.net view.atdmt.com ad.yieldmanager.com ads.adbrite.com oasn04.247realmedia.com segment-pixel.invitemedia.com"	Since the login sequence is replayed continually during the scan, you can improve scan efficiency by filtering out these unnecessary requests. Note that if you delete the regexp altogether, no domains will be filtered out.
Analyze login recording	When you record a login sequence (Scan Configuration > Login Management), AppScan will analyze it and update in-session detection settings (in-session pattern, in-session request, and session IDs received during login). Default: True	If the analysis takes too long, you can change this setting to False. However, if you do this you will need to configure the in-session detection settings manually.
Clear cookies before playing login	Determines whether cookies are deleted before replaying the login sequence. Default: True	

Name	Description	Possible use cases
Common static parameter values	Common static parameter values. Used for the detection of non-random parameter values, which should not be tracked during login. Default: true false \bon\b \boff\b \bout\b checked enabled log\s?in log\s?out exit submit sign ever disabled agree	
Disable Explore stage in-session buffering	During the Explore stage: If the response to a request indicates that the user was out-of-session when it was sent, AppScan queues the request to send again. This insures that as much of the site as possible is scanned. Default: False	If your site throws the user out of session frequently, in-session buffering may result in the Explore stage continuing indefinitely. Setting this option to True will make the Explore stage faster, but may reduce site coverage.
In-session before multi-step operations	By default AppScan verifies in-session status before replaying multi-step operations. Default: True	If you want to test multi-step operations with a non-authenticated user, or if your multi-step sequence includes login steps, change this setting to False. Important: If Configuration > Login Management > Details > Activate In-session detection is <i>deselected</i>, and this advanced setting is set to True (default), the entire login sequence will be replayed before each multi-step operation.
In-session heartbeat interval	Interval in seconds between in-session heartbeats. Default: 5	If AppScan gets out-of-session during the scan, it may be due to a poor connection or an unstable server. Increasing this interval may solve the problem.
Login content type filter	Regular expression defining content types that should be filtered out of the login and multi-step operation sequences. When a login or multi-step operation sequence is recorded, requests whose responses include headers with these content types will be removed from the sequence. Therefore, when AppScan replays the sequence during scanning, requests whose responses contain headers with these content types will not be sent as part of the sequence. Default: text/javascript application/javascript application/x-javascript image text/css	If your site's login procedure, or one of the multi-step operations you have recorded, requires clicking on a link that contains a header with a content type listed here, you should remove it from the regular expression.
Login retry interval	Interval in seconds before re-sending failed login requests. Default: 3	If AppScan gets out of session, and repeated login retry attempts fail, this may be because the server is sensitive to frequent login attempts. Increasing this interval may solve the problem.

Name	Description	Possible use cases
Multipart Content Type Filter	To reduce unnecessary memory consumption, certain content types are automatically filtered out of multipart requests (requests that contain more than one content type). Only content types defined in this regular expression are included in multipart requests; all others are filtered out. Content that has <i>no</i> content type header, is included by default and defined by the value: content_without_content_type_header Default: text/ text/plain application/javascript application/json application/rtf application/xml text/xml content_without_content_type_header	If an important content type is filtered out of requests, add it to this regular expression. You may also be able to reduce memory consumption by removing unnecessary content types so they will not be sent.
Navigational parameter hosts	Regular expression describing hosts. Used for the detection of navigational parameters (by value), which should not be tracked during the login sequence. Default: https?://	If your site uses unusual hosts in navigational parameters, that are not filtered out by the default regexp, you can add them to improve scan efficiency. If you delete the item navigational parameters might not be identified properly.
Navigational parameter scripts	Regular expression describing server-side scripts used in the detection of navigational parameters (by parameter value) which should not be tracked during the login sequence. Default: /[^\.\.]+\.(htm jsp jsf ws dll asp php do)	If your site uses unusual server-side scripts in navigational parameters, that are not filtered out by the default regexp, you can add them to improve scan efficiency. If you delete the item navigational parameters might not be identified properly.
Navigational parameters	Regular expression describing navigational parameters, which should not be tracked during the login sequence. Default: \bnav url page step redirect request location target argument item article goto node action ctrl control source menu frame command	If your site uses unusual navigational parameters that are not filtered out by the default regexp, you can add them to improve scan efficiency. Modifying this regular expression might result in insufficient scan coverage or improper session tracking.
Parse in-session page	If set to False AppScan will not parse the in-session page, and will not update tracked parameters or cookies whose values were changed in the in-session page. Default: True	If your in-session page does not contain tracked cookies or parameters, you can improve performance by changing this setting to False. Note, that if set to False, AppScan will not update cookie/parameter values on the in-session page, which could result in getting out-of-session.
Requests between heartbeats	Following an in-session detection request, AppScan will send at least the number of requests defined here before sending another in-session detection request. Default: 1	In cases where a slow response from the server results in the scan consisting mostly of in-session detection requests (see Traffic Log), increasing this value can reduce scan time.
Timeout for single action-based login attempt	The time in seconds that AppScan waits for the browser to replay a single action-based login attempt, before forcing the browser to close. Default: 120 seconds	

Name	Description	Possible use cases
<i>Special Patterns:</i>		
Exclude from Automatic Form Fill	Parameter names listed here are excluded from the Automatic Form Filler. Default: ^CFID __EVENTVALIDATION __VIEWSTATE ^CFTOKEN __EVENTARGUMENT __EVENTTARGET ^BV_	Parameters with very long values may slow down the scan and increase file size. If your application uses parameters with long values, and they are not needed for filling forms, add them to this list.
<i>Tests:</i>		
CSRF: Pattern of meaningful request	By default AppScan tests POST requests, and requests whose response was "Transaction Successful", for Cross-Site Request Forgery. This setting lets you define additional requests as "meaningful" for Cross-Site Request Forgery vulnerability, in addition to POST requests. This definition is used in conjunction with "CSRF: Pattern of meaningful response". Default: ^POST	If you want to test for Cross-Site Request Forgery on GET requests too, change this regular expression.
CSRF: Pattern of meaningful response	By default AppScan tests POST requests, and requests whose response was "Transaction Successful", for Cross-Site Request Forgery. This setting lets you define additional <i>responses</i> as "meaningful" for Cross-Site Request Forgery vulnerability, in addition to "Transaction Successful". This definition is used in conjunction with "CSRF: Pattern of meaningful request". Default: Transaction Successful	If you want to test for Cross-Site Request Forgery on requests that receive other kinds of responses, define them in this regular expression.

Name	Description	Possible use cases
Difference threshold	AppScan often needs to compare two responses, and decide whether that are "similar" or "different", in order to know whether a test was successful or not. In these cases, AppScan uses a variety of algorithms to assign a Similarity Percentage (where 100% means the two responses are identical). In some cases it decides the test outcome based on whether the Similarity Percentage is above the "Similarity Threshold", and in others based on whether it is <i>below</i> the "Difference Threshold". Both thresholds can be configured. For most tests the default Similarity Threshold is 95%, and the default Difference Threshold is 75%. This means that: • For tests results whose outcome depends on <i>similarity</i>, a Similarity Percentage of 95% or more indicates the two pages are similar. • For tests results whose outcome depends on <i>difference</i>, a Similarity Percentage of 75% or less indicates the two pages are different. If you enter a value between 1 and 100 (percent) for this setting, it will override the default Difference Threshold for all tests. You may also want to adjust the Similarity Threshold. Default: 0 (Use AppScan thresholds)	If your site has no "dynamic" text that causes the similar responses to be slightly different, setting a value lower than 75 may reduce false positive results. Tip: You may also want to adjust the Similarity Threshold (see below).
Disable cookie testing	This setting is used to turn off cookie testing altogether. Default: False	If cookie testing for your application results in a very long scan, you might want to disable it. However, doing so might result in security issues being missed ("false negatives").
Disable cookie testing for static content	Don't test cookies in requests for pages with this extension. Default: ;htm;html;ahtm;ahtml; chtm;chtml;fhtm;fhtml;mht; mhtm;mhtml;css;css1;js;	In order to reduce scan time and memory consumption, you may want to exclude additional types of page extension. If so, add them to the list of extensions to exclude, separated by a semicolon.
Don't test directory or page	This option lets you define a regular expression to exclude specific directories or pages from attacks during the Test stage. Note that this will only exclude the directories or pages defined, and not any subdirectories or files. Default: /wps/[^/]*!/ut/	If you know that certain directories or pages are not vulnerable, or are concerned that testing them might harm site stability, you can exclude them from the scan by defining them in this regular expression. For excluding a folder <i>and all its sub-folders</i>, see "Exclude Paths and Files view" on page 57
Extract links from all responses	By default during the test stage AppScan will only search for new links in vulnerable responses. Default: False	If you think AppScan might miss links, or that its coverage isn't sufficient, you can enable this setting, though doing so will increase scan time and file size.

Name	Description	Possible use cases
Follow all automatic links	By default AppScan only follows automatic links* that are likely to contain vulnerabilities. These are: iFrame, Frame and Redirect. You can configure it to follow <i>all</i> types of automatic link. Note that requests that match the regular expression defined in "Automatic links to ignore" will <i>never</i> be sent, regardless of this setting. Default: False	If you think your site may contain a vulnerability in other types of automatic link, such as scripts, enable this setting. This will increase scan time and file size.
Login after test	Send tests in a single thread, and verify in-session, or send login sequence, after every test. 0 = False 1 = Send tests in a single thread, and verify in-session after every test. If out-of-session, send login sequence. 2 = Send tests in a single thread, and sent login sequence after every test. Default: 0	Settings 1 or 2 may be needed for applications with a sensitive session, or that require frequent logouts to avoid session or memory issues. They significantly increase scan time.
Multi-step Operation: Validation limit	The maximum number of consecutive requests from a Multi-Step Operation sequence that will be validated in Cross-Site Scripting tests. Default: 0	
Pattern to ignore in response	This regular expression defines sections of the response that AppScan will ignore when analyzing test responses. When comparing responses to decide if a test succeeded, AppScan measures the percentage of change in the entire response. If the response is very long, and the change very small, AppScan might ignore the difference and miss the vulnerability. Default: <input[^>]+(__VIEWSTATE __EVENTTARGET __EVENTARGUMENT __EVENTVALIDATION)) [^>]+>	If your site sends responses that include long sections that are not important, defining them here can improve scan accuracy and performance.

Name	Description	Possible use cases
Refresh original response interval	Interval in seconds before refreshing the original response (by sending the request again) during the Test stage. One of the ways AppScan decides whether a Test response reveals a vulnerability is by comparing it with the Explore response. When an Explore response is older than the value set here, the Explore request will be sent again, before sending tests, so that an updated Explore response can be used for the comparison. This is important for cases where the Explore response is likely to vary with time, and comparing the Test response to the outdated Explore response might result in a false positive. Default: 30 (seconds)	If you are sure that your application's responses will never become outdated in this way, you can change this setting to zero to reduce scan time. Explore stage requests will then never be resent.
Send port listener tests	By default AppScan doesn't send port listener tests because of the likelihood of failure and the time it takes to validate. Default: False	If the external site is part of your network, so that it is aware of local IP addresses, you might want to activate this type of blind SQL injection test.
Similarity threshold	AppScan often needs to compare two responses, and decide whether that are "similar" or "different", in order to know whether a test was successful or not. In these cases, AppScan uses a variety of algorithms to assign a Similarity Percentage (where 100% means the two responses are identical). In some cases it decides the test outcome based on whether the Similarity Percentage is above the "Similarity Threshold", and in others based on whether it is <i>below</i> the "Difference Threshold". Both thresholds can be configured. For most tests the default Similarity Threshold is 95%, and the default Difference Threshold is 75%. This means that: - For tests results whose outcome depends on <i>similarity</i>, a Similarity Percentage of 95% or more indicates the two pages are similar. - For tests results whose outcome depends on <i>difference</i>, a Similarity Percentage of 75% or less indicates the two pages are different. If you enter a value between 1 and 100 (percent) for this setting, it will override the Similarity Threshold for all tests. Default: 0 (Use AppScan thresholds)	If your site has no "dynamic" text that causes the similar responses to be slightly different, increasing this percentage may reduce false positive results. Tip: You may also want to adjust the Difference Threshold (see above).
XSS: Test all reflected probes	Usually multiple occurrences of the payload text in a response from the site have the same level of vulnerability, therefore AppScan tests only one of them. Default: False	Set this to True if you want to test <i>all</i> occurrences of the payload text in a single response.

* **Automatic link:** A link on the web page that the browser sends automatically, without any user interaction.

Scan file structure

Explains the basic structure of an AppScan Standard SCAN file.

When you save an AppScan Standard scan, the data is saved to a file with the extension SCAN. This file is a ZIP archive containing several components, including:

RESULTSDB.FDB

The scan results, as displayed in Data view.

templateConfig.xml

The scan template (configuration). The content is identical to the AppScan Standard SCANT file for the same configuration.

Manual_Explore_#.exd files

One or more numbered files for Manual Explore sequences. These can be imported into other scans.

Scan templates

A scan template is simply a scan *configuration* that has been saved so that you can use it again.

- The Regular Scan template can be used to run a scan without changing any of its configurations (however, you must at least set the Starting URL for the scan).
- You can use one of the supplied “Predefined templates” designed for testing AppScan on a specific test site, or a specific type of site.
- You can create custom “User-Defined Scan Templates” on page 123, configured for your own specific requirements.

When you save a *scan template*, you save the configuration definitions of the scan (for future use); when you save a *scan*, you save both the configuration *and* the scan results.

Predefined templates

About this task

AppScan comes with some predefined scan templates that are configured to produce the best results when testing AppScan on a test site. These templates save you the need to adjust the many configuration options to optimize your scan. (Updates to these templates may be included in AppScan updates from time to time.)

- Regular Scan
- Quick and Light Scan (configured to produce useful results in a short time)
- Parameter-Based navigation
- WebSphere Commerce
- WebSphere Portal
- demo.testfire.net (for scanning the Altoro Mutual Bank website, which has been created for demonstration purposes)
- Production Site (configured for use with live production sites, see “Scanning live production environments” on page 294 for more details)
- Hacme Bank
- WebGoat v5
- Worklight (for scanning in an IBM Worklight server environment)

The table below shows some basic configuration details of some of the predefined templates.

Test Application	Path Exclusion	Path Limit	Explore Method*	Case Sensitive	Login
WebGoat	.*attack\?Num=.*	Off	Depth First	Yes	Username: guest Password: guest
demo.testfire.net	none	5	Breadth First	No	Username: jsmith Password: demo1234

* For details of *Explore Method* refer to “Explore Options view” on page 62

To scan with a predefined template:

Procedure

1. Select a template:
 - On the *Welcome Screen*, select one of the *Predefined Templates* area, or
 - On the main screen, click  (or **File > New**), and then in the *Predefined Templates* area select the required template.
2. Click **Save As**, type in a name for the scan, and save the scan.
3. Define the Starting URL for the scan (see “URL and Servers view” on page 40).
4. If applicable, record the login procedure, or supply username and password (see “Login tab” on page 44).

5. Click  Scan

The Paramater-Based Navigation template

Explains the configuration of this predefined template.

The Parameter-Based Navigation template includes the following settings that answer the challenges to scanning of this type of site.

Item	Location	Setting	Comments
Redundant Path Limit	Explore Options	500	Can be further increased if necessary
Depth Limit	Explore Options	10	Enables the handling of ASP.NET 2.0 postback links
Redundancy Tuning Defaults	Parameters and Cookies	1st and 3rd check boxes selected	
Additional parameter	Parameters and Cookies	All Redundancy Tuning check boxes deselected ("Don't Track")	The parameter is in the form of a regular expression: <code>.*(?i)(page redirect content target EVENTTARGET EVENTARGUMENT goto node action ctrl source.*</code> You may need to edit this regexp. if your site uses a navigational parameter that is not listed

See also:

“Sites that use parameter-based navigation” on page 293

“The challenge of parameter-based navigation sites” on page 293

User-Defined Scan Templates

When you set up a scan, you can save the configuration as a template for use in future scans.

About this task

You can save the template before running the scan; or you can save a finished scan as a template for future scans. Also, if you have a scan with interesting results, and you want to rerun the same scan throughout the development and QA process, you can save the scan as a scan template.

Procedure

1. Do one of the following:
 - Configure a scan using the **Scan Configuration Wizard**.
 - Configure a scan using the **Scan Configuration** dialog box.
 - Load a saved scan whose configuration you want to save as a template.
2. On the **File** menu, click **Save As**.
3. Choose the **.scant** file type in the **Save As** dialog box.
4. Type in an easily recognizable name for the template.
5. Click **Save**.

Loading scan templates

About this task

After you have saved a scan template, you can load it, to run a scan based on the predefined scan configuration of the template, or to change the template.

Procedure

1. Do one of the following:
 - Drag and drop the SCAN or SCANT file from its folder to the AppScan interface.

Limitation: This feature does not work on MS Windows 8 systems where the user is has Administrator permissions.

 - Click **File > New**, select the template that you want to use, deselect the **Launch Scan Configuration Wizard** check box, and then click **OK**.

The predefined configurations of the selected scan template are loaded.

2. To start the scan:
 - Click **Scan menu > Start Scan > [Full Scan / Explore / Test]**.
 - If Start Scan is not available, click **Re-scan > [Full Scan / Explore / Test]**.

Editing Scan Templates

About this task

You can edit a scan template, as a persistent modification for all scans that will be based on this template.

You cannot permanently edit the Default scan template. If you change the configuration of Default, the changes are used in the next scan, but they are not kept in the Default scan template.

Procedure

1. Open a scan template (see “Loading scan templates”).
2. Make changes to the configuration settings.
3. Do one of the following:

- **Scan Configuration Wizard:** on the final page of the wizard, select **I will start the scan later**. Then, on the **File** menu, click **Save** or **Save As** and choose the **scant** file type.
- **Scan Configuration** dialog box: click **Save as Template**.

Changing the configuration during a scan

If you change the configuration after you have started a scan, you will need to rerun the scan, or at least the Test stage, to see the effect of the changes. In general:

- If you changed Explore configurations, you should fully re-scan the application, by running Explore and Test (**Scan > Re-Scan > Re-Scan (Full)**).
- If you changed the Test configurations, you need not run the Explore stage again (if it is complete), and can simply re-test the application (**Scan > Re-Scan > Re-Test**).

Chapter 5. Manual Exploring

This section deals with ways in which you can manually explore your application or services before starting the Test stage of the scan, or before starting an automatic scan (that includes both automatic Explore and Test stages).

Manual exploring is when you explore your site to gather data that can be used by AppScan to ensure that when it tests the site it covers parts of the application or services that it might have missed with its automatic Explore stage. This may be because specific user input is required, or because the site responds only to a different type of tool or device. You can manually explore using AppScan, or using it as a recording proxy, or using Generic Service Client (GSC).

In general:

AppScan

is used for manual exploring when the site does not include services. Your actions and input are recorded and used by AppScan in creating tests for the Test stage of the scan.

AppScan as recording proxy

is used when you want to explore manually using your own remote device (such as a mobile phone) or external application (such as a mobile simulator or emulator), with AppScan configured as recording proxy. Use cases may include SOAP web services, or non-SOAP services that do not include a security envelope, or applications that require a different browser. Requests are sent to the site from an external device or application, with AppScan configured as recording proxy. This enables AppScan to record the responses and use them to create tests for the Test stage of the scan.

Generic Service Client (GSC)

is used for exploring web services that have a WSDL file. GSC creates a simple interface for sending requests to the service, and the responses are imported into Appscan and used in creating tests for the Test stage of the scan.

Using AppScan

Manual exploring enables you to explore specific parts of your application, filling in fields and forms as you go. This can be a way of ensuring that particular areas of the site are covered, and that AppScan has the information needed to complete forms correctly.

The AppScan browser opens and you record your actions, links and input data, while you browse through the application. When you stop the recording, AppScan shows you list of the links you crawled, with relevant form filler data that it can now use when exploring or testing the site automatically.

Manual exploring can be done before, instead of, or after the Automatic Explore stage of the scan.

Note: URLs discovered in a Manual Explore are tested individually, just like those found in the Automatic Explore. If you need AppScan to test URLs that it will only be able to access by clicking two or more links *in a specific order*, you must record a multi-step operation (see “Multi-Step Operations view” on page 84).

Before Automatic Explore

There are several reasons why you might choose to perform a Manual Explore before an Automatic Explore:

- You want to use the Manual Explore as a way of providing AppScan with data for filling forms, by manually exploring and completing the data as you go.
- You want to ensure that AppScan tests a specific, important part of the site.
- You want to scan a specific user process (the URLs, files, and parameters that a user will access given a certain scenario), you can create a manual explore for this process only. This manual explore could be done before you begin a scan.
- Your application uses JavaScripts or Java applets that reveal certain parts of the application only when states (such as Hover and Mouse Over) follow each other in a specific order. Note that this is not the same as a multi-step operation. For a multi-step operation, AppScan must visit links in a specific order; in this case, once AppScan has the link, it can test it in a single step, like any other link.

After performing the Manual Explore, you can continue with an automatic Explore stage (**Explore Only**, or **Full Scan**), so that the scan covers your entire application.

Instead of Automatic Explore

Sometimes you may choose to perform a Manual Explore *instead of* an Automatic Explore:

- You want to scan only a small part of the site, and prefer to define the parts to test by manually exploring.

After manually exploring you can complete your scan by clicking **Test Only**.

After Automatic Explore

There are two main reasons why you may choose to perform a Manual Explore *after* an Automatic Explore:

- Your scan resulted in some URLs being categorized as Interactive (see “User Interaction Needed” on page 181), meaning that AppScan was unable to automatically fill in the required data. You can manually explore these URLs to do this.

Note: After you do this the URLs are removed from the list of Interactive URLs.

- Your site includes SWF (Adobe Flash) files. AppScan tests these, if configured (see “Explore Options view” on page 62), but if you find it missed certain files, you can identify them for AppScan using Manual Explore. Note that you do not need to explore the movie itself, just click on the SWF file, close Manual Explore, and then rerun Automatic Explore.

Recording a Manual Explore Procedure

1. Click **Scan > Manual Explore > Using browser**

The AppScan embedded browser opens, with the **Record** button  selected (grayed out).

Note: The default browser is used. You can set the default browser to be either of the two built-in browsers (IE or Chromium), or a supported external browser, in Tools > Options > Preferences tab

2. Browse the site manually, completing data and clicking on links as you go.
3. When you have finished exploring, click **Pause**  or simply close the browser.

Note: You can create a Manual Explore recording that includes two or more unrelated parts of the application: Click **Pause**, browse to a different location, and then click **Record** to resume recording. The **Explored URLs** dialog box appears, displaying the URLs that you visited during your login.

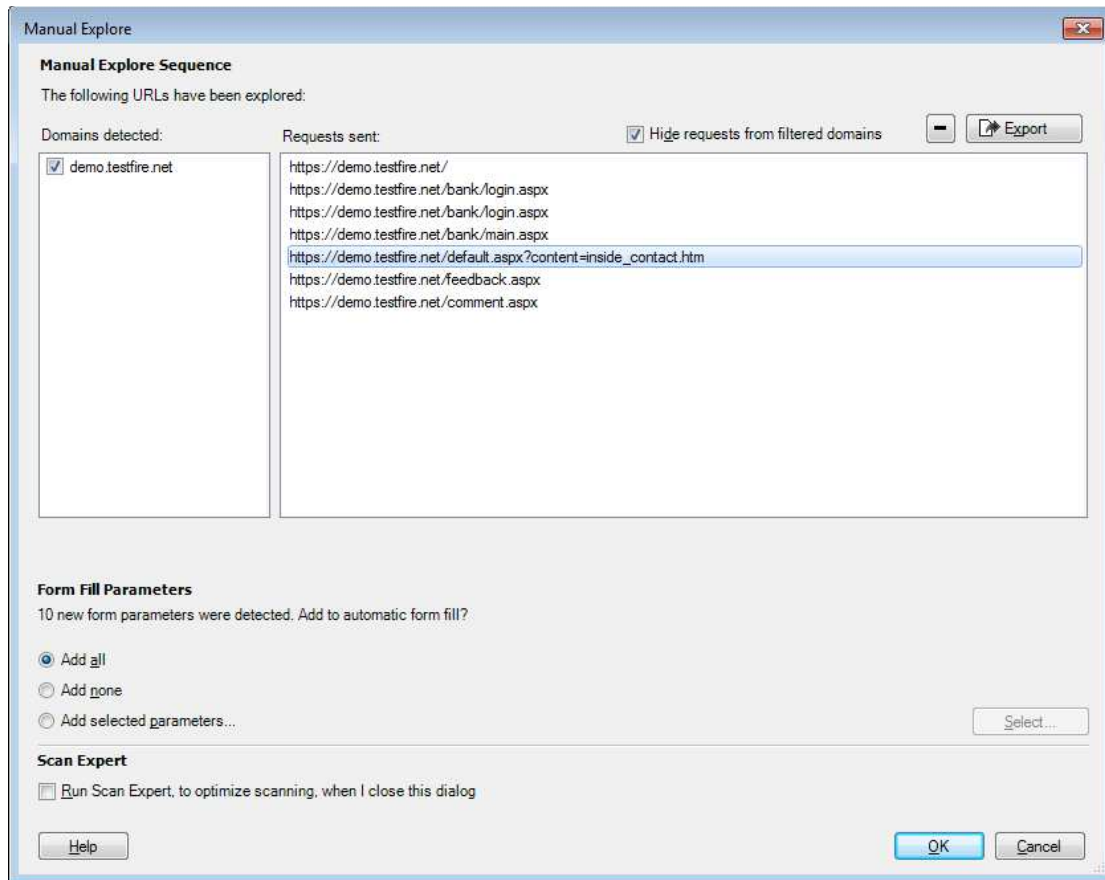
Tip: You can delete any unwanted steps (links) in the sequence by selecting the link and clicking



Working with the Manual Explore sequence

Reviewing and editing and exporting the sequence.

About this task



Procedure

1. You can delete any unwanted steps (links) in the sequence by selecting the link and clicking .
2. Review the list.

If the Manual Explore performed is a standard procedure for the site, you may want to save it for use in future scans. For details see “Exporting Manual Explore data” on page 128.

If you filled HTML forms during the Manual Explore, AppScan determines which information can be added to the Automatic Form Filler settings. If applicable input is found a message appears in the lower half of the dialog-box, and the radio buttons are activated:

[n] new form parameters were detected. Add to Automatic Form Fill?

- If you want *all* your recorded input to be added to the Automatic Form Filler, click **Add All**. Your input is added automatically to the Form Filler information.
 - If you do not want your recorded input to be saved, click **Add None**. Even if it is not permanently saved, your manual input has already been applied.
 - If you want to add only selected parameters and values, click **Add selected parameters**, and then click **Select**. The **Add Auto Detected Form Parameters** dialog box appears (see “Add automatically detected form parameters” on page 128).
3. Using the links that you have added, a Scan Expert evaluation may be able to reveal weaknesses in the configuration. To run Scan Expert when this dialog box closes, select the **Scan Expert** check box. (For more details see “Scan Expert” on page 143.)

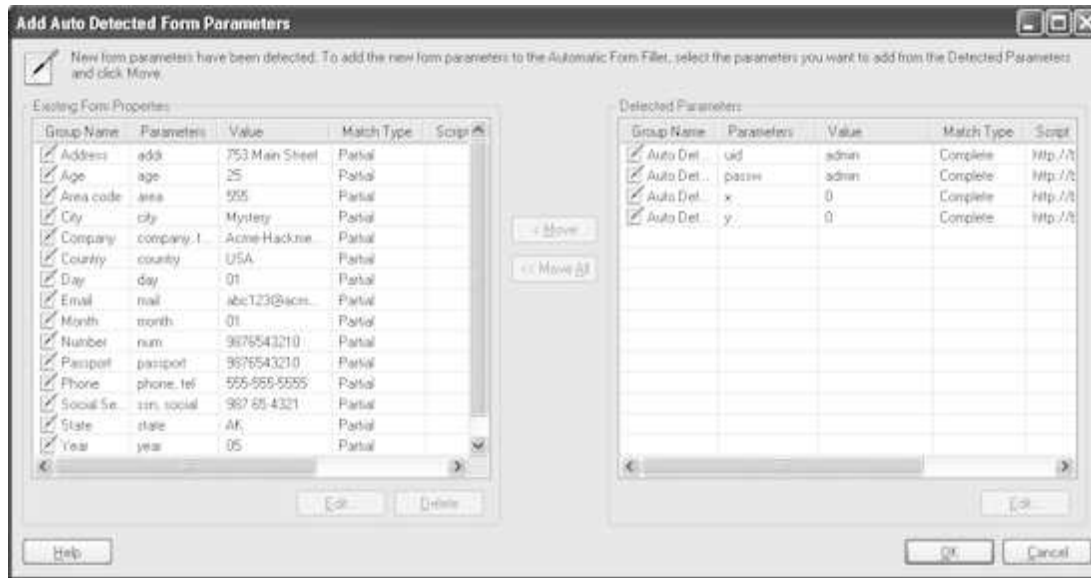
- Click **OK**.

AppScan creates the test requests of your Manual Explore. This may take a few moments.

Add automatically detected form parameters

About this task

The **Add Auto Detected Form Parameters** dialog box lets you select which of the newly detected form parameters you want to add to the Form Properties, to be used automatically in future scans.



Procedure

- In the **Detected Parameters** list (right pane), select the rows that you want to add to the Automatic Form Filler (left pane) and then click **Move**; or select any parameter in the list and click **Move All**.
Parameters that you move to **Existing Form Parameters** will be available to future scans of these URLs.
- Click **OK**.

AppScan analyzes the URLs that you crawled and creates tests based on this analysis.

Exporting Manual Explore data

Standard procedures can be "recorded" in the form of a Manual Explore, and then imported to future scans.

About this task

If the Manual Explore you performed is a standard procedure for the site, you can save the data (requests sent and responses received) for use in future scans. The data is saved as an Explore Data file (.EXD) in XML format, and can be loaded when required, saving you the trouble of recording the same procedure each time.

Procedure

- Record the Manual Explore as in the first steps of the previous procedure.
- When the **Explored URLs** dialog box appears, click **Export**.
A **Save As** dialog box appears.
- Enter a name for the file and then click **OK**.

Note: To view or change the default location to which the file is saved, see **Options > Preferences** (see “Preferences tab” on page 236).

What to do next

Related tasks:

“Recording a Manual Explore” on page 126

“Importing Manual Explore data”

Back to:

“Using AppScan” on page 125

Importing Manual Explore data

If you have saved a standard procedures in the form of a Manual Explore, you can import it to another scan to test it again, even on a different server.

About this task

You can import a Manual Explore procedure that you previously saved, to use as part of a new scan. This is useful in the case of standard procedures that need to be rescanned. This can be done even if scanning the same application on a different server (see note in Step 3 below).

Supported Manual Explore file formats are: EXD, HAR, DAST.CONFIG, and CONFIG.

Procedure

1. Click **File > Import > Explore Data**.
2. Browse to the saved Manual Explore file (EXD, HAR, DAST.CONFIG, or CONFIG), and click **Open**.
The Import dialog box opens.
3. The **Import with responses** check box is selected by default, and AppScan will analyze these responses when preparing to test the site. If your site has changed in a way that affects its responses, you can clear the checkbox, so that AppScan sends the requests again and gathers up-to-date responses. However, in this case it may be more effective to perform the Manual Explore again.

Note: If the file was saved in a version of AppScan that did not save the response data, the check box is cleared and grayed out. You can select **Continue anyway** to import the requests, send them to the site, and gather *new* responses to analyze for testing. However, in this case it may be more effective to perform the Manual Explore again.

4. To start scanning, click **Continue** or **Continue anyway**.
 - a. AppScan analyzes the Explore stage data, to check for host conflicts.
If the file covers hosts that are not included in the current configuration, the Conflicts in Explore Data File dialog box appears. Resolve any conflicts by selecting an appropriate **Action** for each host:
 - **Ignore:** Do not explore this host.
 - **Add to tested hosts:** Add the listed host to the servers of the scan.
 - **Replace with <hostname>:** For each host defined in the scan, there is an option to replace the conflicting host with host already in the scan configuration.

Note: This feature effectively lets you record a manual explore on one host and replay the process on a different server that hosts the same application. This can be used, for example, to record a process on a staging server and replay it on a production server.

- b. The scan begins:
 - If the imported data includes responses, the cached data is analyzed in preparation for the Test stage.
 - If the imported data does *not* include responses, a full Explore stage is run first, followed by analysis of the responses to formulate tests for the Test stage.
 - c. When the Explore stage and analysis are complete, the scan pauses.
5. At this point you can optionally continue Manual or Automatic Exploring.
6. To continue with the Test stage of the scan, click **Scan > Continue > Test**.

What to do next

Related tasks:

“Recording a Manual Explore” on page 126

“Exporting Manual Explore data” on page 128

Back to:

“Using AppScan” on page 125

Using AppScan as a proxy server

You can set AppScan to act as a proxy server and then manually explore your application using a third part browsing utility (browser, web services client, Automatic Explore script, mobile phone and so on) through AppScan. In this mode of operation, AppScan records the HTTP/HTTPS traffic going through it, analyzes it, and creates the appropriate tests.

About this task

Note: Exploring with AppScan as a proxy server can be used to explore SOAP and non-SOAP web services, or in cases where the application client is not compatible with Internet Explorer. By configuring your web services client to use AppScan as a proxy, AppScan will collect the requests to the web service as sent by the client. GSC (Generic Service Client), supplied with AppScan, is used for this purpose for SOAP web services. See “Exploring with the External Traffic Recorder” on page 131.

Note: In the unlikely event of the AppScan browser producing error messages while your external browser does not, you may use this facility to browse manually using an external browser, with AppScan as proxy. (If this happens, contact the AppScan support team so the issue can be resolved.)

Procedure

1. Make sure that AppScan is not configured to use Internet Explorer proxy settings. In **Scan Configuration > Connection View**, do one of the following:
 - Select **Don't use proxy**
 - Select **Use custom proxy settings**; type in proxy address, port, and authentication information.

Note: If you do not do this the setting is automatically changed to **Don't use proxy** when you start exploring.

2. Find the AppScan listening port. Open **Tools > Options > Recording Proxy tab** (see “Recording Proxy tab” on page 237).

The **Proxy Port** area shows the port that AppScan is using to listen to traffic going to the web application. It may be a port that AppScan assigned to itself (the port number you are looking for will be gray), or a port that you selected manually.

3. Configure your web browser to use AppScan as its proxy:

In your browser, find the section for configuring a proxy server. Change the hostname or address to the IP address (localhost is usually an acceptable entry) in use by the machine running AppScan, and change the port to the AppScan listening port.

4. Perform a Manual Explore of your application (see “Using AppScan” on page 125 for full details):
 - a. Click **Scan > Manual Explore**, to open the AppScan internal browser.
 - b. Without closing the internal browser, open your external browser.
 - c. Manually explore the application as necessary.
 - d. Close the external browser.
 - e. Close the AppScan internal browser.

Configuring your browser to use AppScan as a proxy

An example showing how to configure your browser to use AppScan as a proxy.

Procedure

1. In AppScan, click **Tools > Options > Scan Options tab**.
Let's say the port is set to **Automatic** and is **4744**.
2. Open **Microsoft Internet Explorer > Tools > Internet Options > Connections tab > LAN Settings**, and select the check box in the **Proxy Server** section.
The **Address** and **Port** text boxes are enabled.
3. In Internet Explorer, enter the configuration you found in AppScan:
Address: localhost and **Port:** 4744

Using AppScan as recording proxy

You can use AppScan's External Traffic Recorder as a recording proxy, and manually explore RESTful or other non-SOAP web services - or SOAP services that do not require security envelopes - using a mobile phone, simulator, or emulator. AppScan displays the domains and requests in its External Traffic Recorder, and create appropriate tests from the input.

Note: For web services that have a WSDL file, and utilize SOAP security envelopes, you should use GSC to explore the service instead (for details see “Using GSC” on page 134).

Note: If your application uses man-in-the-middle protection, it is not possible to scan it with AppScan as proxy.

Exploring with the External Traffic Recorder

This is an example of a simple workflow for a non-SOAP web services scan.

About this task

This sample workflow shows each conceptual step separately.

Note: You can configure and send requests from more than one mobile device through the same instance of the External Traffic Recorder. All domains and requests will be listed together.

Procedure

1. **Select a template**
Click **File > New** and select a template:
 - For IBM Worklight Developer:
 - a. Select the Worklight template

- b. If your Worklight application code uses app authentication (authenticity): In the Worklight server, open the Worklight console, and make sure that application authentication (authenticity) is disabled, or disable it in the application code
- For other environments: Use the Regular Scan template

Note: If Internet Explorer is configured on your machine to use AppScan as a proxy, you must verify that AppScan is not configured to use your Internet Explorer proxy settings, as this would result in a loop. To resolve this conflict, in Configuration > Communication and Proxy tab, select one of the other two options:

- **Don't use proxy**
- **Use custom proxy settings**

If you record a Manual Explore using the External Traffic Recorder without doing this, the setting will automatically be changed to **Don't use proxy**. The Test stage Redundancy Tuning is used whether or not the check box is selected.

2. In the wizard welcome dialog box, select External device/client (with AppScan as recording proxy), and then click **Next**.
3. Follow the wizard steps:
 - a. "Recording Proxy" on page 33
 - b. (Optional;) "Connection Settings" on page 34
 - c. "SSL Certificate" on page 34
 - d. "Login Management" on page 35
 - e. (Optional;) "Login Management Details" on page 35
 - f. "Test Policy" on page 36
 - g. "Complete" on page 36
4. When the External Traffic Recorder opens with status "Waiting for incoming connections, manually Explore the web service from your device/application:
 - a. Using your device or application, explore the web service.
As you explore, domains detected are listed in the left pane of the recorder, and URLs are listed in the right pane.
 - b. When finished, in AppScan click **Stop Recording**.
5. Review and edit the Manual Explore data:

Domains detected

All domains to which requests were sent are listed, and by default selected for adding to the list of Additional Servers and Domains (Configuration > URLs and Servers > Additional Servers and Domains) so they can be included in the scan. You can deselect any you do not want included in the scan.

Tip: You should deselect any domains that belong to other companies.

Requests sent

All requests sent by the device to domains *that are selected* (in the left pane), are listed. If you select/clear domains in the left pane, the requests list is updated. You can delete specific requests if they are not needed.

Tip: If the total number of filtered requests is more than 200, deleting some of them may produce a more efficient scan.

Note: At this stage you can click **Export** to save the Explore data for use on another machine.

6. Click OK to close the recorder.
AppScan takes a few moments to process and display the data.

7. To start the Test stage, click **Scan > Test Only**

The Test stage starts and when complete the scan results are displayed.

Related topics:

-
- “Recording Proxy tab” on page 237
- Chapter 8, “Results: Security Issues,” on page 185

External Login Recorder

When you manually explore using AppScan as recording proxy, this recorder displays traffic received during login, and enables you to edit and approve it for use logging in to the application during the scan.

The External Login Recorder opens when you click **Configuration > Login Management > Record > Use external device**, and is used for recording the login sequence. It shows the requests sent.

Item	Description
Proxy Connection Status	Shows whether incoming connections are being recorded, and other status messages.
Listening on port	Shows the current port assigned to the recorder.
Recording Proxy Configuration	Opens Tools > Options > Recording Proxy tab to change the port, or any other recording proxy configurations (for details, see “Recording Proxy tab” on page 237).
Login Requests sent	Shows all requests recorded during the Manual Explore. Requests from domains that are selected in the left pane are shown in black; others are gray.
Stop recording	Stops recording incoming requests, but leaves the dialog box open for review and editing the login sequence. To delete individual requests from the list that are not relevant to the scan, select the request and then click 
OK	Closes the recorder.

For the full workflow, see “Exploring with the External Traffic Recorder” on page 131.

See also:

- “Recording Proxy tab” on page 237
- “Redundancy tuning” on page 73

External Traffic Recorder

When you manually explore using AppScan as recording proxy, this recorder displays domains detected and traffic received, and enables you to control which of these will be tested. A limited version of the recorder is used to record the login sequence.

The External Traffic Recorder opens when you click **Manual Explore > Using external device**.

Item	Description
Proxy Connection Status	Shows whether incoming connections are being recorded, and other status messages.
Listening on port	Shows the current port assigned to the recorder. To change the port, or any other recording proxy configurations, click Recording Proxy Configuration (for details, see “Recording Proxy tab” on page 237).
Recorded Traffic	

Item	Description
Domains Detected (left pane)	List of all domains detected in the recorded traffic. Select the domains that should be included in the scan. When you close the recorder, all selected domains are added to the Additional Servers and Domains list (Configuration > URL and Servers > Additional Servers and Domains) and will be included in the scan.
Requests Sent (right pane)	Shows all requests recorded during the Manual Explore. Requests from domains that are selected in the left pane are shown in black; others are gray. • To see <i>only</i> the requests from the selected domains, click the Hide requests from filtered domains check box • To delete individual requests from the list that are not relevant to the scan, select the request and then click 
Export	Click to export the recording for use on another machine. This button is active only after the recording has been stopped.
Apply Explore stage Redundancy Tuning	(Selected by default) When selected, Explore stage Redundancy Tuning (Configuration > Parameters and Cookies tab > Redundancy Tuning Defaults > Explore) will be applied to the current recording when you close the dialog box, to help avoid duplicate requests. Clear the checkbox only if selecting it results in cookies from the Manual Explore being missed.
Stop Recording	Stops the recording, leaving the dialog box open for viewing and editing the lists. Note: Once you stop the recording you cannot start recording again without discarding the current data.
OK	Closes the dialog box and adds all currently selected domains to the list of additional servers and domains included in the scan (Configuration > URL and Servers > Additional Servers and Domains).

For the workflow, see “Exploring with the External Traffic Recorder” on page 131.

See also:

- “Recording Proxy tab” on page 237
- “Redundancy tuning” on page 73

Using GSC

Generic Service Client (GSC) uses your web service's WSDL file to display a simple interface showing the services available, and lets you enter parameters and view the results. Use the GSC interface to explore your web service manually, so that AppScan can use your input to create appropriate tests.

Note: GSC is used for exploring web services that are defined with a WSDL file. However, if your service does not have a WSDL file, or the service does not utilize SOAP security envelopes, or if you have an application of your own that you want to use for the Explore stage, you can instead use a mobile device, emulator or simulator, with AppScan's in-built External Traffic Recorder configured as a recording proxy (for details see “Using AppScan as recording proxy” on page 131).

After you have sent some parameters to the service and received responses, and you close GSC, click **Run Tests Only** to start an automatic scan based on the input. (This scan is actually a Test stage only, since the Explore stage is completed during your manual exploration using GSC.)

If SSL is used without a client-side certificate, no special configuration of GSC is needed (if you are using the version of GSC supplied with AppScan 8.0). For configuration in the case of CCS, refer to the GSC Information Center, available from the main GSC toolbar.

Note: AppScan remains open while GSC is open, but does not function until GSC is closed.

Note: Some of the functionality described in the GSC Help file may not apply when using it through AppScan.

Exploring with GSC

This is an example of a simple workflow for a SOAP web services scan.

Before you begin

In order to send tests to web services, GSC must be installed on your system. During AppScan installation you were asked if you want to install it. If you did not install GSC then, you can do so at any time by clicking on the GSC_Setup.exe file in your main AppScan folder.

Procedure

1. Open URL and Servers view of the Scan Configuration dialog box, and in the Starting URL field add one of the following:
 - The URL of the WSDL file
 - The path to the WSDL file on your local network, in the form:
`file:///c:/mywsdlfile.wsdl`

Restriction: When the WSDL file is supplied locally, GSC is unable to extract the domain name from the file. Therefore if you chose the second option, you must provide GSC with the domain name in the **Additional Servers and Domains** area. For example: `demo.testfire.net`

2. If relevant, select the Case Sensitive check box.
3. Click **OK** to close the Scan Configuration dialog box.
4. Click **Scan > Explore web Services**.

GSC opens and a tree of the web services appears in the left pane. (Click the  icons to expand the tree and see the individual web services.)

5. Explore the services:
 - a. Click on a service in the tree to select it. An interface appears in the right pane for sending requests to the service.
 - b. In the Message tab in the right pane, type in the value to send.
 - c. Click **Invoke** to send the request.

The result is displayed in the main pane, and the request is added to the Call History pane at the bottom left of the screen.
 - d. Repeat for additional services as required.
6. When you have sent enough requests, close GSC.

GSC closes and tests are created based on the data.
7. To start the scan, click **Scan > Test Only**.

Results

When the scan is complete the results are displayed.

Example

See Chapter 8, “Results: Security Issues,” on page 185

Scanning a site that includes a SOAP web service as *part* of the site

Use this procedure if your site includes both a web Service and other pages that require scanning.

About this task

If your site contains both web services and other pages that need to be scanned, you must explore the web services manually using GSC, but can then let AppScan explore the rest of the site automatically, and then test the *whole* site. In this case you need to provide AppScan both with the URL of the web service WSDL file, *and* a starting URL for exploring the site.

Procedure

1. Click **Scan > Scan Configuration > URL and Servers**, and in the **Starting URL** field, type in the URL of the WSDL file for the web service.
2. In the **Additional Servers and Domains** area, click  and type in the starting URL for scanning the application.
3. Click **Explore web Services**.
GSC opens.
4. Send requests to the service and then close GSC.
5. Click **Scan > Full Scan**.

AppScan explores the application, and then tests the whole site and presents the results.

See also:

Chapter 8, "Results: Security Issues," on page 185

Chapter 6. Scanning

Learn how to start a scan, and what happens during the scan; how to manually manipulate the Explore stage, and how to export the results of a scan.

Starting scans

Note: If configured, Scan Expert will run before the main scan, when you start the scan. It will evaluate your configuration and suggest changes to optimize the main scan. For details see “Scan Expert” on page 143.

Starting scans from the Scan Configuration Wizard

About this task

If you use the **Scan Configuration Wizard** (see “Scan configuration wizard” on page 27) to create your scan, the final step of the wizard offers options for starting a scan.

Procedure

Select an option (see table below), and click **Finish**.

Option	Click to:
Start a full automatic scan	Start a scan with the configuration you just created in the wizard; start with automatic Explore and proceed automatically to the Test stage.
Start with automatic Explore only	Start the automatic Explore stage of a scan, but do not proceed automatically with the Test stage.
Start with Manual Explore	Open the browser to let you manually explore the application (see “Using AppScan” on page 125).
I will start the scan later	Close the wizard without scanning (for example, if you want to further edit the scan configuration before starting the scan, or if you want to start the scan later).
Start Scan Expert when the Scan Configuration wizard is complete	Scan Expert analyzes your configuration and suggests configuration changes to increase scan efficiency. (For more details see “Scan Expert” on page 143) Select this option to run Scan Expert as soon as you close the wizard.

Starting scans from the Scan menu or the toolbar

About this task

When AppScan is open, you can start a scan with the current configuration from the Scan menu or the toolbar.

Procedure

On the **Scan** menu, or from the  button on the toolbar, select:

If the configuration does *not* include any Manual or Automatic Explore data:

Icon	Option	Description
	Full Scan	Run a full scan. Explore the application until there are no more unvisited URLs, and then continue automatically with the Test stage. (If multiphase scanning is configured, complete multiple phases as necessary.)
	Explore Only	Explore the site but do not test it. This lets you examine the Explore results, and if necessary explore the site manually, before proceeding to the Test Stage.
	Test Only	(This option is disabled when there is no Explore data to test.)

If the configuration includes any Manual or Automatic Explore data:

Icon	Option	Description
	Continue Full Scan	Explore and Test the site using this configuration, including the existing Automatic and Manual Explore data.
	Continue Explore Only	Explore the site automatically, and add the data to the existing Automatic and Manual Explore data. This lets you examine the Explore results, and if necessary explore the site manually, before proceeding to the Test Stage.
	Continue Test Only	Test the site using the existing Automatic and Manual Explore data only.

Note: If you have configured one or more Multi-Step Operations, and they constitute a significant subset of your site that you want to scan, you can run a scan on those sequences only (see “Scan Multi-Step Operations Only” on page 170).

Starting scans from the Welcome dialog box

When you start up AppScan, the **Welcome** dialog box appears.

Note: If the **Welcome** dialog box does not appear on startup, see “General tab” on page 240 to learn how to make the **Welcome** dialog box appear again.

You can:

- “Create a new scan” or
- “Load an existing scan” on page 139

Create a new scan

Procedure

1. Select **Create a New Scan**.
2. Do one of the following:
 - If you want to run the scan using one of the predefined configurations, select the **Launch Scan Configuration Wizard** check box. (See “Automatic scan” on page 142.)
 - If you want to make changes to the configuration before starting the scan, *clear* the **Launch Scan Configuration Wizard** check box. (See “Scan configuration wizard” on page 27.)
3. Click on a scan template. If you do not need a specific template, select Regular Scan.

Load an existing scan

Procedure

1. Select **Open an Existing Scan**.
2. Select a saved scan from the list; or click **Browse** to open an unlisted saved scan.
3. Click **OK**.

Starting scans from the New Scan dialog box

You can open the **New Scan** dialog box at any time (**File > New**). It allows you to choose to open the Scan Configuration Wizard or to choose a scan template, upon which you can run a full scan immediately.

Launching the Scan Configuration Wizard:

1. Make sure the **Launch Scan Configuration Wizard** check box is selected or cleared as needed.
2. In the Predefined Templates list, click on the required template. If you do not need a *specific* template, click **Regular Scan**.

Note: To use a saved template, click **Browse**, browse to the required SCANT file, and then click **Open**.

Scan progress

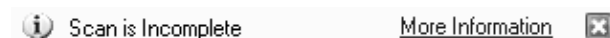
When a scan begins, the **Progress panel** appears below the Result list.



This shows:

- Current URL being explored or tested.
- Percentage completed of the Explore or Test stage.
- If this is a subsequent scan phase, the current phase number (see "Automatic multiphase scanning" on page 143).
- Amount of time (mm:ss or hh:mm:ss or dd:hh:mm:ss) since the scan began.

If the scan is stopped, either due to communication issues or by your request, the Scan Notice Panel appears in place of the Progress Panel



The **Scan Notice Panel** shows:

- Communication issues and timeout before automatic resume.
- Notification that scan is incomplete.
- "More Information" link, which opens a dialog box explaining what has been done so far and how to continue.

Pausing and continuing scans

You can pause a scan at any time and resume it later. If a scan was paused due to connection problems, you can continue the scan after the problem has been fixed.

Note: When you pause a scan it may continue for some time before actually stopping.

1. To pause a scan, click the **Pause** button  on the toolbar (or click **Scan > Pause**).
The **Notice Panel** appears, displaying: Scan is incomplete.
2. To continue a scan that was paused, click the **Scan** button  on the toolbar, and select one of the options:
 - **Full Scan:**  Resume full scan. (Continue with the Explore stage until there are no more unvisited URLs, and then continue automatically with the Test stage. If multiphase scanning is configured, complete multiple phases as necessary.)
 - **Explore Only:**  Resume and complete the Explore stage but do not continue with the Test stage.
 - **Test Only:**  Resume the Test stage, ignoring any unvisited URLs that had not been explored when the scan was paused.

Scans stopped due to connectivity issues

A scan may be stopped due to connection problems between AppScan and the server it is scanning, or between AppScan and the local web proxy server.

When a connection problem is detected, a 90 second countdown begins while AppScan waits for the problem to be fixed. If the problem is fixed during the countdown, the Scan Notice Panel disappears and the scan resumes.

If there is a communication problem, the **Notice panel** shows the message:

Trying to connect to: <IP> Explore/Test stops in <n>

- The IP address shown is the IP of the scanned server, the IP of the proxy (if the connection problem is with the web proxy), or a list of IPs, if the scan is connecting to multiple servers and more than one are down.
- The number <n> is the 90 second countdown.
- If the problem is not fixed by the time the countdown reaches zero, the scan stops.
- After the problem is fixed, you can continue the scan where it left off (**Scan > Continue**) or redo the scan (**Scan > Re-scan**).

Scans stopped due to application issues

About this task

Sometimes a scan stops and the **Notice panel** displays a link for **More Information**.

Procedure

1. Click **More Information**.

A message box appears, explaining what to do next. The instructions include how to find the list of Interactive URLs or the list of Broken Links due to missing NTLM authentication. If a scan finds either of these, and is unable to resolve them automatically, the **More Information** message box will summarize the issue.

2. Use the following procedures as appropriate:
 - “Dealing with interactive URLs” on page 141
 - “Dealing with failed requests” on page 141

Dealing with interactive URLs

How to include in the scan URLs that require user interaction.

About this task

If the scan is stopped because it could not automatically fill in all the input forms, you can complete the Explore manually.

Procedure

1. Select **Application Data** view.
2. Select the **User Interaction Needed** display.
3. Create a Manual Explore of these URLs (see “Manually exploring interactive URLs” on page 182) and insert the form-filler values.
4. Continue the scan, preferably including another automatic Explore stage.

Dealing with failed requests

How to deal with requests that failed due to missing NTLM authentication.

About this task

If the scan encounters broken links due to missing NTLM authentication, the scan will stop.

Procedure

1. Select **Application Data** view.
2. Select the **Failed Requests** display.
3. In the **Scan Configuration** dialog box > **Login/Logout**, enter your platform authentication details.
4. Continue the scan with a re-run of the Explore stage.

Saving and loading scans

When you save a scan, the configuration and all data that AppScan has collected so far (in both Explore and Test stages) are saved as a .SCAN file.

After you save a scan, you can load the .SCAN file to resume or rerun the scan.

Saving scans

About this task

You can save a scan any scan that has finished, was paused, or that stopped due to external causes.

Procedure

Click **File > Save** (or press [Ctrl] + S).

Note: To view or change the default folder where scans are saved, click **Tools > Options > Preferences**. See “Preferences tab” on page 236 for details.

Automatic scan save

About this task

You can set AppScan to save data to a .SCAN file automatically during scanning. When this option is selected, AppScan saves both periodically (at an interval you define) and at certain milestones during the scan.

Procedure

Click **Tools > Options > Scan Options** tab and select the **Automatically Save during Scan** check box. You can also adjust the interval between saves.

The running scan will be saved at the interval you selected. It will also be saved:

- When you start or continue any part of the scan
- When the Explore stage ends
- When the Test stage ends

When a new scan starts and auto-save is enabled, a message appears, explaining that the scan needs to be saved because auto-save is turned on.

- Click **Yes** to open a **Save As** dialog box to save the scans
- Click **No** to disable the "Automatically save during scan" function for the current scan only
- Click **Disable** to disable the "Automatically save during scan" function for this and future scans

Loading saved scans

Procedure

Click **File > Open** and select a scan.

The **Loading** progress bar opens for a few moments, and then the **Application Tree** of the scan, as explored or tested when you last saved it, appears in the main window.

Importing a legacy scan template

How to import the scan template from a scan saved in a version of AppScan earlier than 8.6.

In AppScan version 8.6 the scan file format was redesigned, and scans saved in earlier versions cannot be opened in the current version. If necessary you can import the template for use in a new scan.

To import the scan template:

1. Change the scan file's extension from SCAN to ZIP
2. Open the ZIP file, then locate and extract `templateconfig.xml`
3. Change its extension from XML to SCANT
4. Open with AppScan.

Automatic scan

A full automatic scan consists of one or more cycles of an Explore stage followed by a Test stage.

"Automatic multiphase scanning" on page 143

Explore stage

When you begin a full automatic scan, the Explore stage starts first. During this stage AppScan does the following:

- Crawls the application as a user would, from the starting URL you gave in the scan configuration (see "URL and Servers view" on page 40), to every URL in the application that has not been excluded.

Note: Scan Configuration lets you filter the Explore stage (see "Exclude Paths and Files view" on page 57). If you exclude specific paths from it, or set specific Explore limits, these filters are applied to the Explore stage.

- Builds the **Application Tree**, a hierarchical model of the URLs in the application.
- Analyzes the explored URLs and generates tests.

Test stage

During the Test stage, AppScan does the following:

- Logs in to the application.
- Performs preliminary tests on the URLs, which help interpret results.
- Tests URLs by sending requests designed to reveal vulnerabilities.
- Records the response to each request.
- Provides test results.

During the Test stage the Security **Issues** view of the **Result List** shows the results of the scan.

Automatic multiphase scanning

About this task

Some tests may reveal new areas of the web application which are not accessible through normal browsing of the website (for example, directory listings, or the contents of the robots.txt file). If Multiphase scanning is enabled, AppScan will add URLs that were discovered during the Test stage, to the list of URLs to be explored. When the Test stage is complete, AppScan then automatically explores the newly discovered URLs, creates new tests for them, and tests them.

By default, scans may contain up to four phases, if necessary. You can configure AppScan to run between 1 and 10 phases.

Note: The multiphase scanning setting only applies if you run a Full Scan. If you activate the **Explore Only** and **Test Only** functions, the result will be a single-phase scan.

Procedure

1. Open **Scan Configuration | Test Options** view.
2. Select the **Enable Multiphase Scanning** check box.
3. In the **Max. Phases** text box, enter the maximum number of phases you want to allow (a number between 1 and 10; default 4).

As AppScan tests the site it analyzes test responses for additional URLs that it could not access during the initial Explore stage. It can then run additional Explore and Test stages on these new links. The number you enter here determines the number of times AppScan will do this. (By default, multiphase scanning is enabled for four phases.)

Note: The Progress Panel (see “Scan progress” on page 139) shows which phase the scan is currently working on.

Note: If you re-scan your application (**Scan > Re-scan**), the phase number restarts at 1.

Note: If you save a scan, the current phase number is saved. If you later load that scan to run again, it begins from the saved phase number.

Scan Expert

Scan Expert is a feature that explores your application and network behavior to evaluate the efficiency of your scan configuration. Based on its findings it may recommend changes to the configuration to ensure a more rigorous scan.

Scan Expert connects with your application and runs a short Explore stage on it. During this time the Scan Panel indicates progress, and the Application Tree shows the parts of the application that have been explored, just as during a regular scan. The Scan Expert pane also opens (showing that this is not a regular scan).

Ways of using Scan Expert:

- Scan Expert can be configured to run automatically before a full scan. Depending on how you configure it, it can then:
 - prompt you with recommendations which you can then accept or reject manually, or
 - apply auto-enabled recommendations automatically (not all recommendations are auto-enabled) and proceed with the scan(See “Scan Expert view” on page 104 for details. To enable/disable Scan Expert from running automatically at the start of scans go to **Tools > Options... > Preferences.**)
- Scan Expert can be run on its own at any time, to evaluate your configuration:
 - To have Scan Expert run a short Explore stage on the application and then analyze the responses to evaluate the configuration, click **Scan > Run Scan Expert Evaluation.**
 - If you have already explored the application, you can save time by letting Scan Expert analyse the *existing* data, without running its own Explore stage. Click **Scan > Run Scan Expert Analysis Only.**
- If you try to start a scan and AppScan detects a serious problem with the configuration, it may run Scan Expert even if it is not configured to run automatically, in order to attempt to resolve the issue.

Scan Expert recommendations

When its Explore stage is complete, Scan Expert analyzes the responses received and recommends changes to the configuration. Depending on how you have configured Scan Expert (see “Scan Expert view” on page 104), the suggested changes may be applied automatically or displayed as an interactive list.

The table below summarizes the options available from the Scan Expert Recommendations list.

Option	Description
Recommendation	Recommendations which AppScan is able to implement without additional user input are automatically selected. Select/deselect the check boxes as required. Recommendations which require user input to implement, appear with a blue link. Click the link to open the Configuration dialog box at the relevant tab, and provide the necessary input (such as recording a login procedure).
More Info	Click to open the More Info window for the selected recommendation. This window includes the reasoning behind the recommendation and instructions on how to apply the recommendation manually.
Manual Edit	Lets you apply the currently selected recommendation manually by opening the Scan Configuration dialog box at the relevant view.
Apply Recommendations	Updates the scan configuration to comply with all recommendations whose check box is selected, and closes the Scan Expert pane.
Ignore All	Discards all recommendations and closes the Scan Expert pane.

Glass box scanning

Overview

Outlines the principles of glass box scanning and its setup.

While regular scanning looks on the application as a "black box", analyzing its *output* without "looking inside" it; glass box scanning uses an agent installed on the application server to inspect the code itself, during the scan. Hence the term "glass" box. To do this the AppScan glass box agent must be installed on the same server as the application you want to test, not on the local machine where AppScan itself is installed.

Glass box scanning has the following advantages:

- During the Explore stage, glass box scanning can reveal HTTP parameters that affect the server-side but which are not found in responses, and which would therefore not be discovered by black box scanning alone.
- During the Test stage, glass box scanning can verify the success or failure of certain tests, such as Blind SQL Injection, with greater accuracy, resulting in fewer "false positive" results. It can also reveal the existence of certain security issues that cannot be detected by black box techniques.
- Glass box scanning enables AppScan to show you the vulnerability in the actual source code, simplifying both reporting and remediation.

Including glass box scanning adds an extra dimension to the scan in terms of the kind and number of issues that can be found, and the issue information offered.

To set up and work with glass box scanning:

Task	Description
1. Install agent	Install the AppScan glass box agent on your application server. <i>Do this once only for a single server.</i> Note: Agents can be installed on more than one server, but only one server can be included in a glass box scan.
2. Define agent	Define the installed agents in AppScan, so it can communicate with them. <i>Do this once only for each AppScan machine.</i> Note: Multiple instances of AppScan (on different machines) can use the same glass box web server agent, but they cannot do so simultaneously.
3. Configure scan	Configure the <i>scan</i> to use the glass box agent you require. By default this is configured automatically, but it can be adjusted in Scan Configuration > Glass Box . <i>Do this for each scan.</i>
4. Run scan	Scan your application with glass box scanning enabled.
5. Update agent rules	Update the server agent rules when prompted to do so by the automatic update process, so that the version of the rules on the web server remains synchronized with the rules on your local AppScan version. Note: After running the update process you must restart the web application server.

For Java platforms

Installing and using the glass box agent on a Java server.

Installing the glass box agent

This section describes how to set up the server-side glass box agent, to enable glass box scanning.

The AppScan glass box agent must be installed on your application server (or servers), by copying certain files from the AppScan installation to the server.

Note: The glass box agent installs and uses a dedicated Java agent (`gbAgent.jar`) to function. If there are other Java agents defined on the web server, you can add the glass box agent to the application server command line (the exact path will depend on your installation):

```
java ... -javaagent:c:\otherAgent\otherAgent.jar  
-javaagent:c:\glassbox\gbAgent.jar ...
```

System requirements

The following platforms and technologies are supported.

Software	Details
JRE	Versions 6 and 7 are supported. JRE 8 is <i>not</i> supported.
Operating System	Supported Microsoft Windows systems (both 32-bit and 64-bit editions): • Microsoft Windows Server 2012 • Microsoft Windows Server 2012 R2 • Microsoft Windows Server 2008 R2 Supported Linux systems: • Linux RHEL 5, 6, 6.1, 6.2, 6.3, 6.4 Supported UNIX systems: • UNIX AIX 6.1, 7.1 • UNIX Solaris (SPARC) 10, 11
Java EE container	JBoss AS 6, 7; JBoss EAP 6.1; Tomcat 6.0, 7.0; WebLogic 10, 11, 12; WebSphere 7.0, 8.0, 8.5, 8.5.5

Before you start

It will save you time if you have the following information ready before starting to install. You may need to consult your web application server administrator to get the information.

- Your server's operating system (Windows, Linux, Unix)
- Your Java EE application server (WebSphere, WebLogic, Tomcat, JBoss), and the mode in which it is installed (standard or as an operating system service)
- The web applications deployment location of your Java EE application server (for example: D:\apache-tomcat-6.0.32\webapps)
- The location of the Java runtime used by your Java EE application server (for example: C:\Program Files (x86)\Java\jre6)
- The credentials for your Java EE application server administration (for the purpose of deploying a new web application)

Note: If the server will be running in secure mode, you must add special permissions. See “Permissions needed to work in secure mode” on page 155

Logs

Glass box logs are saved to:

[Installation folder]\instrumentation.log

Install automatically using the glass box agent installer:

This section describes how to install the glass box agent automatically through the user interface.

About this task

When AppScan is installed, the files needed for installing the server agent are saved in a dedicated folder on your machine. To perform this task you will need access to this folder, and to your application server.

Procedure

1. Open ...\\Program Files\\IBM\\AppScan Standard\\Glass box
The exact path will depend on where you installed AppScan.
2. Copy the relevant setup file to the web server:
 - **Linux server:** Copy the file GB_Java_Setup.bin
 - **Windows server:** Copy the file GB_Java_Setup.exe
3. Launch the GB_Java_Setup application, and follow the online instructions. During this process you will be asked to:
 - Select your web application server. If the server is not listed (for example JBoss Service, Tomcat Service, or WebLogic Service), select **Other**.
 - Define a username and password for the agent. Later, when the agent is defined in AppScan, these credentials will be needed to enable communication between it and the agent. Only English ASCII characters can be used.

Note: If you selected **Other**, continue as for relevant *manual* installation, starting from Step 5.

Note: You may also be asked to supply the path to the relevant JAVA_HOME (JDK) or JRE_HOME (JDK or JRE) folder for your framework.

Tip: The installer interface language options include only languages supported by your operating system. If you want to run the installer in a different language, you can start the installer through the command line, and add a flag for the required language. For example, to run the installer in Japanese on an English OS, run the command GB_Java_Setup.bin -l ja

- A starting script is created in the glass box agent installation folder
- If you have a desktop, a shortcut is created for starting the server with the glass box agent activated.

Important: In order to enable glass box scanning, the application server must be started using one of these, as this starts the server with the glass box agent activated.

Install automatically using the command line:

This section describes how to install the glass box agent through the command line.

About this task

When AppScan is installed, the files needed for installing the server agent are saved in a dedicated folder on your machine. To perform this task you will need access to this folder, and to your application server.

Procedure

1. Open ...\\Program Files\\IBM\\AppScan Standard\\Glass box
The exact path will depend on where you installed AppScan.
2. Copy the relevant setup file to the web server:
 - For a Linux server, copy the file GB_Java_Setup.bin
 - For a Windows server, copy the file GB_Java_Setup.exe
3. Run a command line containing *all* of the following arguments *that apply to your server*, separated by spaces.

Command	Description
GB_Java_Setup.bin (Linux) or GB_Java_Setup.exe (Windows)	The setup file.
-i console	

Command	Description
-l en	Sets the language for the installation. Use the code for your language ("en" is for English).
-DCHOSEN_INSTALL_SET=JBoss / WebSphe / Tomcat / WebLogic / Other	The type of server on which you are installing the agent. Note: If your server is not one of those listed (for example, if it's JBoss Service, Tomcat Service, or WebLogic Service), set Other .
-DUSER_INSTALL_DIR=value	Sets the path where the agent will be installed.
-DGLASS_USERNAME=value	Sets a username for accessing the agent. Only English characters and numbers can be used.
-DGLASS_PASSWORD=value	Sets a password for accessing the agent. Only English characters and numbers can be used.
-DWEBLOGIC_PATH / -DJBOS_PATH / -DTOMCAT_PATH / -DWEBSPPHERE_PATH=value	Sets the path of the web server installation directory. Examples (actual location depends on your system): WebLogic: C:\weblogic\user_projects\domains\base_domain\ JBoss: C:\jboss-6.0.0\ Tomcat: C:\apache-tomcat-6.0.32\ WebSphere: C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01\
-DJBOS_SERVER_NAME=value / -DWEBSPPHERE_SERVER_NAME=value	The server name.
-DWEBLOGIC_USERNAME / -DWEBSPPHERE_USERNAME=value -DWEBLOGIC_PASSWORD / -DWEBSPPHERE_PASSWORD=value	Username and password to access the application server
-DWEBLOGIC_LIB=value	The path to the LIB folder, for example: C:\weblogic\wlserver_10.3\server\lib\
-DWEBLOGIC_TARGET=value	The server/target name.
-DWEBLOGIC_PORT=value	The administration port.
-DSELECTED_JBOS_TYPE=value	Use one of the following values: JBoss AS JBoss EAP Standalone JBoss EAP Managed Domain

Example for WebSphere:

```
GB_Java_Setup.bin -i console -l en -DCHOSEN_INSTALL_SET=WebSphe
-DUSER_INSTALL_DIR=/opt/glass_box -DGLASS_USERNAME=jsmith
-DGLASS_PASSWORD=1234
-DWEBSPPHERE_PATH=/opt/IBM/WebSphere/AppServer/profiles/AppSrv01
-DWEBSPPHERE_SERVER_NAME=server1
-DWEBSPPHERE_USERNAME=admin -DWEBSPPHERE_PASSWORD=admin_pw
```

4. Verify that your web server stopped at the end of the procedure. If it did not, then stop it manually.
5. Start the web server again using the desktop shortcut, or the starting script in the glass box agent installation folder, so that the glass box agent is now active.

Install manually on a JBoss server or a JBoss Service server:

This section describes how to manually install the glass box agent on a JBoss or JBoss Service server.

About this task

When AppScan is installed, the files needed for installing the server agent are saved in a dedicated folder on your machine. To perform this task you will need access to this folder, and to your application server.

Procedure

1. Open ...\\Program Files\\IBM\\AppScan Standard\\Glass box
The absolute path will depend on where you installed AppScan.
2. Locate GB_Java_Manual_Setup.zip and copy it to your web server.
3. Extract the contents of this folder to your preferred location on the web server.
4. Define a username and password for the agent (use only English ASCII characters):
 - **Linux server:** Run AgentCredentials.sh <username> <password>

Note: AgentCredentials.sh must have execution permissions.

- **Windows server:** Run AgentCredentials.bat <username> <password>
5. Deploy GBootstrap web application:
 - a. Log in to the JBoss Administration Console. The default location is `http://<server_name>:<port_number>/admin-console/`
 - b. Click **Applications > web Application WARs**, then click **Add a new resource**.
 - c. Type in the path to GBootstrap.war (found in the glass box folder you extracted), then click **Continue**.

A message confirms that the resource was created successfully.

6. Shut down the JBoss server/service.
7. Supply the Java agent gbAgent.jar (found in the folder you extracted) as a JVM argument by doing *one* of the following:
 - **Through the environmental variable:** Configure JBoss's JVM through the JAVA_OPTS environmental variable (if it does not exist, create it), with the value:

```
-javaagent:"<path_to_gbAgent.jar>/gbAgent.jar"  
-Djava.net.preferIPv4Stack=true  
-Djboss.modules.system.pkgs=com.ibm.glassbox.asm.agent,  
  com.ibm.glassbox.agent,com.ibm.glassbox.logger,  
  com.ibm.glassbox.jsp,org.jboss.byteman
```

- **Through batch/sh scripts:** Pass arguments to the JVM running JBoss by editing the configuration files that are shipped with the server:

Linux server:

- a. In the JBoss folder locate and open the appropriate file in an editor (usually located in /bin):
 - **JBoss AS:** run.sh
 - **JBoss EAP Standalone:** standalone.sh
 - **JBoss EAP Managed Domain:** domain.sh
- b. Locate a line that begins with JAVA_OPTS and add the following code below it:

```
export JAVA_OPTS=$JAVA_OPTS "-javaagent:<path_to_gbAgent.jar>/gbAgent.jar"  
-Djava.net.preferIPv4Stack=true  
-Djboss.modules.system.pkgs=com.ibm.glassbox.asm.agent,  
  com.ibm.glassbox.agent,  
  com.ibm.glassbox.logger,  
  com.ibm.glassbox.jsp,org.jboss.byteman
```
- c. Save and close the file.

Windows server:

- a. In the JBoss folder locate and open the appropriate file in an editor (usually located in /bin):
 - **JBoss AS:** run.conf
 - **JBoss EAP Standalone:** standalone.conf
 - **JBoss EAP Managed Domain:** domain.conf
- b. Locate a line that begins with set JAVA_OPTS= and add the following line below it:

```
set JAVA_OPTS=$JAVA_OPTS "-javaagent:<path_to_gbAgent.jar>\gbAgent.jar"
-Djava.net.preferIPv4Stack=true
-Djboss.modules.system.pkgs=com.ibm.glassbox.asm.agent,
  com.ibm.glassbox.agent,com.ibm.glassbox.logger,
  com.ibm.glassbox.jsp,org.jboss.byteman"
```
- c. Save and close the file.

Important: Ensure that JAVA_OPTS is initialized only once, and prior to the lines above, so that other JAVA_OPTS assignments do not overwrite -javaagent.

Note: To add arguments to JAVA_OPTS, use the %JAVA_OPTS% / \$ JAVA_OPTS convention.

8. Restart the JBoss server/service.

Install manually on a Tomcat server:

This section describes how to manually install the glass box agent on a Tomcat server.

About this task

When AppScan is installed, the files needed for installing the server agent are saved in a dedicated folder on your machine. To perform this task you will need access to this folder, and to your application server.

Procedure

1. Open ...\\Program Files\\IBM\\AppScan Standard\\Glass box
The absolute path will depend on where you installed AppScan.
2. Locate GB_Java_Manual_Setup.zip and copy it to your web server.
3. Extract the contents of this folder to your preferred location on the web server.
4. Define a username and password for the agent (use only English ASCII characters):
 - **Linux server:** Run AgentCredentials.sh <username> <password>

Note: AgentCredentials.sh must have execution permissions.

 - **Windows server:** Run AgentCredentials.bat <username> <password>
5. Deploy GBootstrap web application.
 - Preferred method:
 - a. Log in to Tomcat Manager. The default location is `http://<server_name>:<port_number>/manager/html`
 - b. In **Deploy table > War file to deploy**, click **Choose file**
 - c. Locate GBootstrap.war (in the glass box folder you extracted), and click **Open**
 - d. Click **Deploy** and verify that GBootstrap has been added the applications list
 - Alternative method:
 - a. Copy the file GBootstrap.war from the glass box folder you extracted
 - b. Paste it into the \\webapps folder, default location:
C:\\apache-tomcat-[version]\\webapps
6. Shut down Tomcat.

7. Configure Tomcat to always use the glass box agent by doing *one* of the following:
 - **Through the environmental variable:** Configure Tomcat's JVM through the JAVA_OPTS environmental variables (if it does not exist, create it), with the value `-javaagent:<path_to_gbAgent.jar>/gbAgent.jar`.
 - **Through batch/sh scripts:** Pass arguments to the JVM running Tomcat by editing the configuration scripts that are shipped with it:

Linux server:

- a. In the Tomcat folder (usually located in `<path_to_Tomcat_folder>/bin`), open `startup.sh`
- b. Locate a line that begins with `CATALINA_OPTS` and add the following line below it:
`export CATALINA_OPTS = $CATALINA_OPTS -javaagent:<path_to_gbAgent.jar>/gbAgent.jar`
- c. Save and close the file.

Windows server:

- a. In the Tomcat folder (usually located in `<path_to_Tomcat_folder>\bin`), open `startup.bat`
- b. Locate a line that begins with `set CATALINA_OPTS=` and add the following line below it:
`set CATALINA_OPTS = %CATALINA_OPTS% -javaagent:<path_to_gbAgent.jar>/gbAgent.jar`
- c. Save and close the file.

Important: Ensure that `CATALINA_OPTS` is initialized only once, and prior to the lines above, so that other `CATALINA_OPTS` assignments do not overwrite `-javaagent`.

Note: To add arguments to `CATALINA_OPTS`, use the `%CATALINA_OPTS% / $ CATALINA_OPTS` convention.

8. Restart Tomcat.

Install manually on a Tomcat Service server:

This section describes how to manually install the glass box agent on a Tomcat Service/Daemon server.

About this task

When AppScan is installed, the files needed for installing the server agent are saved in a dedicated folder on your machine. To perform this task you will need access to this folder, and to your application server.

Procedure

1. Open `...\Program Files\IBM\AppScan Standard\Glass box`
The absolute path will depend on where you installed AppScan.
2. Locate `GB_Java_Manual_Setup.zip` and copy it to your web server.
3. Extract the contents of this folder to your preferred location on the web server.
4. Define a username and password for the agent (use only English ASCII characters):
 - **Linux server:** Run `AgentCredentials.sh <username> <password>`

Note: `AgentCredentials.sh` must have execution permissions.

 - **Windows server:** Run `AgentCredentials.bat <username> <password>`
5. Deploy GBootstrap web application:
 - Log in to Tomcat Manager. The default location is `http://<server_name>:<port_number>/manager/html`
 - In **Deploy table > War file to deploy**, click **Choose file**.
 - Locate `GBootstrap.war` (in the glass box folder you extracted), and click **Open**.
 - Click **Deploy** and verify that GBootstrap has been added the applications list.
6. Shut down Tomcat.

7. Configure Tomcat to use the glass box agent:
 - a. Locate Tomcat JVM at ...\\Tomcat 7.0\\bin\\tomcat7w.exe
 - b. Double-click on it and select **Properties > Java** tab.
 - c. In the Java Options area, add the following as a new line: -javaagent:<path_to_gbAgent.jar>/gbAgent.jar

Note: If there are other Java agents defined on the web server, you can *add* the glass box agent to the Java Options area (the exact path will depend on your installation):

... -javaagent:c:\\...\\otherAgent.jar -javaagent:c:\\glassbox\\gbAgent.jar ...

- d. Click **OK**.
8. Restart Tomcat.

Install manually on a WebLogic server:

This section describes how to manually install the glass box agent on a WebLogic server.

About this task

When AppScan is installed, the files needed for installing the server agent are saved in a dedicated folder on your machine. To perform this task you will need access to this folder, and to your application server.

Procedure

1. Open ...\\Program Files\\IBM\\AppScan Standard\\Glass box
The absolute path will depend on where you installed AppScan.
2. Locate GB_Java_Manual_Setup.zip and copy it to your web server.
3. Extract the contents of this folder to your preferred location on the web server.
4. Define a username and password for the agent (use only English ASCII characters):
 - **Linux server:** Run AgentCredentials.sh <username> <password>

Note: AgentCredentials.sh must have execution permissions.

 - **Windows server:** Run AgentCredentials.bat <username> <password>
5. Deploy GBootstrap web application:
 - a. Log in to the WebLogic Administration Console. The default location is http://<server_name>:<port_number>/console/
 - b. In the **Domain Structure** pane, click **Deployments**, then click **Install**
 - c. In the **Path** field, type in the path to GBootstrap.war, then click **Next**
 - d. Select the **Install this deployment as an application** radio button, then click **Next**
 - e. Verify that the text in the **Name** field is **GBootstrap**
 - f. Click **Finish**, then click **Save**
 - g. In the **Domain Structure** pane, click **Deployments**, and verify that **GBootstrap** has been added, and that it has a green checkmark icon in the **Health** column.
6. Shut down the WebLogic server.
7. Supply the Java agent gbAgent.jar (found in the folder you extracted) as a JVM argument by doing *one* of the following:
 - **Through the environmental variable:** Configure WebLogic's JVM through the JAVA_OPTS environmental variable (if it does not exist, create it), with the value:
 - **Linux server:** -javaagent:<path_to_gbAgent.jar>/gbAgent.jar
 - **Windows server:** -javaagent:<path_to_gbAgent.jar>\\gbAgent.jar

- **Through batch/sh scripts:** Pass arguments to the JVM running WebLogic by editing the configuration scripts that are shipped with the server:

Linux server:

- In the WebLogic folder (usually located in <path_to_Weblogic_folder>\bin), open the startup file in an editor:
 - **Admin server:** Open DOMAIN_NAME/bin/startWebLogic.sh
 - **Managed server:** Open DOMAIN_NAME/bin/startManagedWebLogic.sh
- Locate a line that begins with JAVA_OPTIONS and add the following line below it:
export JAVA_OPTIONS = \$JAVA_OPTIONS -javaagent:"<path_to_gbAgent.jar>/gbAgent.jar"
- Save and close the file.

Windows server:

- In the WebLogic folder (usually located in <path_to_WebLogic_folder>\bin), open startWebLogic.bat
- Locate a line that begins with set JAVA_OPTIONS and add the following line below it:
set JAVA_OPTIONS = %JAVA_OPTIONS% -javaagent:<path_to_gbAgent.jar>/gbAgent.jar
- Save and close the file.

Important: Ensure that JAVA_OPTIONS is initialized only once, and prior to the lines above, so that other JAVA_OPTIONS assignments do not overwrite -javaagent.

Note: To add arguments to JAVA_OPTIONS, use the %JAVA_OPTIONS% / \$ JAVA_OPTIONS convention.

- Restart the WebLogic server.

Install manually on a WebLogic Service server:

This section describes how to manually install the glass box agent on a WebLogic Service/Daemon server.

About this task

When AppScan is installed, the files needed for installing the server agent are saved in a dedicated folder on your machine. To perform this task you will need access to this folder, and to your application server.

Procedure

- Open ...\\Program Files\\IBM\\AppScan Standard\\Glass box
The absolute path will depend on where you installed AppScan.
- Locate GB_Java_Manual_Setup.zip and copy it to your web server.
- Extract the contents of this folder to your preferred location on the web server.
- Define a username and password for the agent (only English characters and numbers can be used):
 - **Linux server:** Run AgentCredentials.sh <username> <password>

Note: AgentCredentials.sh must have execution permissions.

 - **Windows server:** Run AgentCredentials.bat <username> <password>
- Deploy GBootstrap web application:
 - Log in to the WebLogic Administration Console. The default location is http://<server_name>:<port_number>/console/
 - In the **Domain Structure** pane, click **Deployments**, then click **Install**
 - In the **Path** field, type in the path to GBootstrap.war, then click **Next**
 - Select the **Install this deployment as an application** radio button, then click **Next**

Note: If the Install option is disabled (grayed out), you may need to change the Lock & Edit mode setting to enable it.

- e. Verify that the text in the **Name** field is **GBootStrap**
 - f. Click **Finish**, then click **Save**
 - g. In the **Domain Structure** pane, click **Deployments**, and verify that **GBootStrap** has been added, and that it has a green checkmark icon in the **Health** column.
6. Shut down the WebLogic service.
 7. Configure WebLogic to use the glass box agent by locating the registry key HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\<WEBLOGIC_SERVICE> and editing the value of CmdLine to include -javaagent:<path_to_gbAgent.jar>/gbAgent.jar
 8. Restart the service.

Install manually on a WebSphere server:

This section describes how to manually install the glass box agent on a WebSphere server.

About this task

When AppScan is installed, the files needed for installing the server agent are saved in a dedicated folder on your machine. To perform this task you will need access to this folder, and to your application server.

Procedure

1. Open ...\\Program Files\\IBM\\AppScan Standard\\Glass box
The absolute path will depend on where you installed AppScan.
2. Locate GB_Java_Manual_Setup.zip and copy it to your web server.
3. Extract the contents of this folder to your preferred location on the web server.
4. Define a username and password for the agent (use only English ASCII characters):
 - **Linux server:** Run AgentCredentials.sh <username> <password>

Note: AgentCredentials.sh must have execution permissions.

- **Windows server:** Run AgentCredentials.bat <username> <password>
5. Deploy GBootStrap web application:
 - a. Log in to the WebSphere's Integrated Solutions console. The default location is `http://<server_name>:<port_number>/ibm/console/`

Note: If more than one profile has been defined, log into the appropriate one.
 - b. Select **New Applications > New Enterprise Application**
 - c. In the **Path to the new application** area, add the full path to the GBootStrap.war file.
 - d. In the **Context root** field, type in GBootStrap, then click **Next**.
The Install New Application wizard opens.
 - e. Unless you want to change any default options, simply click **Next** until the wizard completes. If you are asked to save the configuration, do so. When you have completed the wizard, GBootStrap.war will appear in the list of applications. A red X or a green arrow icon in the Application Status column indicate whether or not it is active. If it is *not* active (red X), select it and click **Start**. A green arrow indicates that GBootStrap.war is active.
 6. Now select **Servers > Application Servers**
 7. In the right pane, click on your server's name.
 8. Click on the **Configuration** tab to bring it to the front.
 9. Under Server Infrastructure, click **Java and Process Management > Process Definition**.
 10. Under Additional Properties click **Java Virtual Machine**.

11. Add the following argument to the Generic JVM Arguments:
-javaagent:c:/path/to/gbAgent.jar (the example is for Windows operating systems; for other systems the path should be adjusted appropriately).

Note: The path should not contain any spaces.

Note: If there are other Java agents defined on the web server, you can *add* the glass box agent to the Generic JVM Arguments area (the exact path will depend on your installation):

... -javaagent:c:\otherAgent\otherAgent.jar -javaagent:c:\glassbox\gbAgent.jar ...

12. Click **Apply**, and save the configuration if asked.
13. Restart WebSphere server.

Permissions needed to work in secure mode

This section describes the special permissions you need to add if the web application server will be running in secure mode during the scan.

If the server will be running in secure mode (that is, with Java Security Manager enabled), you must add the following special permissions to the GBootstrap web application:

- Permission to access "getClassLoader" (java.lang.RuntimePermission)
- Permission to use "accessClassInPackage.sun.net.www.protocol.*" (java.lang.RuntimePermission)
- Read permission to the "java.io.tmpdir" property (java.util.PropertyPermission)
- Read/Write/Delete permission to "<<ALL FILES>>"

If necessary, consult your web server manual for instructions on how and where to add these Java Security permissions. The sections below provide some sources and examples that may help you in doing this. It should be stressed that the examples are examples only, and would almost certainly need adaptation to work in your particular case.

IBM WebSphere

Guide:

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.express.doc/info/exp/ae/tsec_waspolicyfile.html

Outline of lines to be added:

```
grant codeBase "file:${application}" {  
    permission java.lang.RuntimePermission "getClassLoader";  
    permission java.lang.RuntimePermission  
"accessClassInPackage.sun.net.www.protocol.*";  
    permission java.io.FilePermission "<>", "read, write, delete";  
    permission java.util.PropertyPermission "java.io.tmpdir", "read";  
};
```

Location:

<profile_root>/config/cells/<cell_name>/applications/<ear_file_name>
/deployments/<application_name>/META-INF/was.policy

Example of lines to be added:

```
"C:\Program Files (x86)\IBM\WebSphere\AppServer\profiles\AppSrv01  
\config\cells\R0IS-PSNode01Cell\applications\GBootstrap_war.ear  
\deployments\GBootstrap_war\META-INF\was.policy")
```

Apache Tomcat

Guide:

http://tomcat.apache.org/tomcat-6.0-doc/security-manager-howto.html#Configuring_Tomcat_With_A_SecurityManager

Outline of lines to be added:

```
grant codeBase "file:${catalina.base}/webapps/GBootStrap/-"{
    permission java.lang.RuntimePermission "getClassLoader";
    permission java.lang.RuntimePermission
"accessClassInPackage.sun.net.www.protocol.*";
    permission java.io.FilePermission "<<ALL FILES>>">>, "read, write, delete";
    permission java.util.PropertyPermission "java.io.tmpdir", "read";
};
```

Location:

<CATALINA_HOME>/conf/catalina.policy

Example of lines to be added:

"C:\Software\Tomcat\apache-tomcat-6.0.33\conf\catalina.policy"

Weblogic

Guide:

http://download.oracle.com/docs/cd/E13222_01/wls/docs81/security/server_prot.html

Specifically refer to the section entitled "Setting Application-Specific Security Policies"

Outline of lines to be added:

```
<security-permission>
  <description>
    Allow getting the J2EEJ2SETest4 property
  </description>
  <security-permission-spec>
    grant {
      permission java.util.PropertyPermission
"welcome.J2EEJ2SETest4","read";
    };
  </security-permission-spec>
</security-permission>
```

Line to be added:

weblogic.xml

JBoss

Guide:

<http://docs.jboss.org/jbossweb/latest/security-manager-howto.html>

Outline of lines to be added:

```
grant codeBase "file:${catalina.base}/webapps/GBootStrap/-"{
    permission java.lang.RuntimePermission "getClassLoader";
    permission java.lang.RuntimePermission
"accessClassInPackage.sun.net.www.protocol.*";
    permission java.io.FilePermission "ALL FILES", "read, write, delete";
    permission java.util.PropertyPermission "java.io.tmpdir", "read";
};
```

Line to be added:

CATALINA_HOME/conf/catalina.policy

Defining the glass box agent in AppScan

This section describes how to configure AppScan to access a server-side glass box agent.

About this task

Once you have installed the glass box agent on your application server, you must define it to AppScan so it can access the agent. After this is done AppScan will automatically use the agent for glass box scanning in any relevant scan (unless you deselect it in the scan configuration).

Restriction: You can define the same server agent on several AppScan machines, however an agent can be used for only one scan at a time.

Restriction: You can define multiple agents, but only one agent can be selected in a scan configuration.

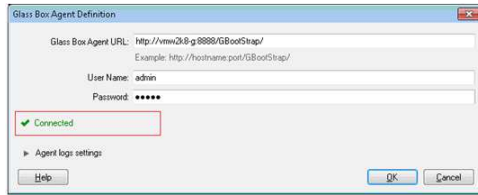
Procedure

1. In **Configuration > URL and Servers tab**, set the **Starting URL**. Note that in this example, the URL includes port 8080.
`https://altdoro.testfire.net:8080/`
2. Click **OK** to close the Configuration dialog box.
3. Click **Tools > Glass box agent management**.
The Glass Box Agents dialog box appears, listing any agents that have already been defined.
4. To add a new agent to the list, click .
The Glass Box Agents Definition dialog box appears. The Glass Box Agent URL is automatically filled, based on the Starting URL you entered previously.

Important: The Glass Box Agent URL automatically contains a port value extracted from the Starting URL, (in this case 8080). If you installed your agent on a different port, replace the port value with the correct one. In this example the port value has been changed to 8888.
`http://altdoro.testfire.net:8888/GBootstrap/`
5. Fill in the remaining fields and options as needed.

Option	Description
Username and password	Type in the username and password that were defined when the agent was installed on the server.
Agent log settings:	(Optional:) Click to open these settings. Note: Agent log settings are saved in the agent on the server-side.
Max. log rows	(Optional:) Use the slider to limit the size of the log.
Log content	(Optional:) Select the level of information to be include in the log: Errors: Include only error messages; Warnings: Include error and warning messages; Info: Include errors, warnings and informational messages; Debug: Include all messages. Glass box logs are saved to: [Installation folder]\instrumentation.log

AppScan tries to connect to the agent server. If successful, a green "Connected" icon appears. If there is any problem, a red icon will appear with a message such as "Unable to connect to agent" or "Credentials needed".



Note: If your site offers both HTTP and HTTPS, verify that the starting URL is HTTPS (and change it to HTTPS if it is not).

Note: If you get an error message, refer to

<http://www.ibm.com/support/docview.wss?uid=swg21567723>

6. Click **OK** to close the definition dialog box.

The agent is added to the list.

7. Click **OK** to close the list.

The agent is now defined on this machine.

Change user credentials:

This section describes how to change the username and password used to access the glass box agent.

About this task

When the glass box agent is installed, you are asked to define a username and password for accessing the agent. These credentials help insure that the agent is accessible only to AppScan. If you need to change the credentials after installation, follow these instructions.

Procedure

1. On your application server, open a command line to your installation directory.
2. Define the new credentials:
 - For a Linux server: Run `AgentCredentials.sh <newusername> <newpassword>`
 - For a Windows server: Run `AgentCredentials.bat <newusername> <newpassword>`
3. Undeploy GBootstrap from the server.
4. Deploy the new modified GBootstrap.war from the glass box installation folder to the server.

Configuring a glass box scan

Although the scan is configured automatically, this section describes options you may want to change.

Procedure

1. Configure your scan as usual.
2. In **Configuration > Glass Box tab**, select one of the defined glass box agents from the drop-down list, and optionally adjust the settings:

Setting	Details
Use this glass box agent	If the glass box agent has been installed on your application server, and defined in AppScan, you can select it for use in the scan. If you have entered a Starting URL, AppScan attempts to select the appropriate agent automatically. When an agent is selected, AppScan attempts to connect to it, and indicates whether this was successful. Note: If you select an agent and get the message "Credentials needed", check that the credentials supplied in Tools > Glass Box Management are correct. If the required server does not appear in the drop-down list, you can define it by clicking the Glass box agent management link. Restriction: Only one glass box agent can be selected for use in a scan. If the application being scanned has more than one server, you must scan using each server agent separately.
Use glass box in the Explore stage	(Selected by default.) This function can increase coverage of the site, by examining the server-side source code for the presence of parameters that affect the behavior of the server, but do not appear in the response. Example server-side code: <pre>String debugOn = request.getParameter("debug"); if (debugOn == "true"){ response.getWriter().println(SECRET_SERVER_DATA); }</pre> In this example the developer has left the parameter "debug" in the code. It does not appear in any link on the site, but if an attacker were to send a request containing it, SECRET_SERVER_DATA could be obtained.
Use glass box in the Test stage	(Selected by default.) Select this check box to send glass box tests during the Test stage of the scan. This function can verify the success or failure of certain tests, such as Blind SQL Injection, with greater accuracy, and also reveal the existence of certain security issues that cannot be detected by black box techniques.
Skip equivalent black box tests	(Cleared by default.) This means that both glass box tests and black box tests for the same vulnerability (WASC Threat Classification) are sent. This is because although the glass box tests are generally both more accurate and give more detailed results, occasionally a glass box test may fail while the equivalent black box test succeeds. If the results for your application are unchanged when black box tests are skipped, you can reduce scan time by selecting this check box.

The status bar indicates that glass box scanning is enabled, and you are ready to start the scan.



Scanning with glass box

This section describes glass box scanning.

About this task

Once you have defined the glass box agent to AppScan, glass box scanning is enabled by default. You can use the Scan Configuration dialog box to verify that the correct server agent is selected, and glass box scanning is configured to run as part of the scan.

Glass box scanning can discover hidden URLs in the Explore stage, and additional issues and information during the Test stage.

Procedure

1. Click **Configuration > Glass Box** view.
2. Select the agent you want to use from the drop-down list.

Note: If your agent does not appear in the list, click the **Glass box agent management** link, and define it.

3. Verify that one or both of the two main glass box scan options are selected:
 - Use glass box in the Explore stage
 - Use glass box in the Test stage

Note: The **Skip equivalent black box tests** check box is cleared by default. This means that both glass box tests and black box tests for the same vulnerability (WASC Threat Classification) are sent. This is because although the glass box tests are generally both more accurate and give more detailed results, occasionally a glass box test may fail while the equivalent black box test succeeds. If the results for your application are unchanged when black box tests are skipped, you can reduce scan time by selecting this check box.

4. Click **Scan > Full Scan** to start the scan.

The scan starts and the status bar messages indicate that glass box scanning is active.



The scan results will include glass box data in the Issue Information tab where available.

Uninstalling the glass box agent

This section describes how to uninstall the server-side glass box agent.

- “Uninstall automatically”
- “Uninstall manually on a JBoss server” on page 161
- “Uninstall manually on a JBoss Service server” on page 161
- “Uninstall manually on a Tomcat server” on page 162
- “Uninstall manually on a Tomcat Service server” on page 162
- “Uninstall manually on a WebLogic server” on page 163
- “Uninstall manually on a WebLogic Service server” on page 163
- “Uninstall manually on a WebSphere server” on page 163

Uninstall automatically:

This section describes how to uninstall the glass box agent automatically through the user interface.

About this task

The glass box agent can be uninstalled automatically *only* if it was installed automatically using the glass box agent installer. If it was installed manually, it must be uninstalled manually too, following the instructions in the next sections of this guide.

To perform this task you will need access to your application server.

Procedure

1. Open the glass box directory on your server.
2. Launch the uninstaller:
 - **Linux server:** Double-click on `Uninstall.bin`
 - **Windows server:** Double-click on `Uninstall.exe`

3. Follow the online instructions for completely uninstalling glass box.

Uninstall manually on a JBoss server:

This section describes how to manually uninstall the glass box agent on a JBoss server.

About this task

When glass box is installed on the application server, the installation includes a Java agent and a GBootstrap web application. To uninstall glass box both of these must be removed. This task requires access to your application server.

Procedure

1. Undeploy GBootstrap web application:
 - a. Log in to the JBoss Administration Console. The default location is `http://<server_name>:<port_number>/admin-console/`
 - b. Click **Applications > Web Application WARs**, then, in the GBootstrap.war entry, click **Delete**.
 - c. When asked to confirm, click **OK**.
2. Shut down the JBoss server.
3. Remove the Java agent from JAVA_OPTS. During manual installation, JAVA_OPTS is edited to include gbAgent.jar. Refer to the installation instructions to remove gbAgent.jar from JAVA_OPTS, and restore JAVA_OPTS to its original state.
4. Delete the glass box directory from the JBoss server, unless it is still to be used by other web application servers.
5. Restart the JBoss server.

Uninstall manually on a JBoss Service server:

This section describes how to manually uninstall the glass box agent on a JBoss Service server.

About this task

When glass box is installed on the application server, the installation includes a Java agent and a GBootstrap web application. To uninstall glass box both of these must be removed. This task requires access to your application server.

Procedure

1. Undeploy GBootstrap web application:
 - a. Log in to the JBoss Administration Console. The default location is `http://<server_name>:<port_number>/admin-console/`
 - b. Click **Applications > Web Application WARs**, then, in the GBootstrap.war entry, click **Delete**.
 - c. When asked to confirm, click **OK**.
2. Shut down the JBoss service.
3. Remove the Java agent from JAVA_OPTS. During manual installation, JAVA_OPTS is edited to include gbAgent.jar. Refer to the installation instructions to remove gbAgent.jar from JAVA_OPTS, and restore JAVA_OPTS to its original state.
4. Delete the glass box directory from the JBoss server, unless it is still to be used by other web application servers.
5. Restart the service.

Uninstall manually on a Tomcat server:

This section describes how to manually uninstall the glass box agent on a Tomcat server.

About this task

When glass box is installed on the application server, the installation includes a Java agent and a GBootstrap web application. To uninstall glass box both of these must be removed. This task requires access to your application server.

Procedure

1. Undeploy GBootstrap web application:
 - a. Log in to Tomcat Manager. The default location is `http://<server_name>:<port_number>/manager/html`
 - b. In the application list, select **GBootstrap.war**, then click **Commands > Undeploy**.
 - c. When asked to confirm, click **OK**.
2. Shut down Tomcat server.
3. Remove the Java agent from CATALINA_OPTS. During manual installation, CATALINA_OPTS is edited to include gbAgent.jar. Refer to the installation instructions to remove gbAgent.jar from CATALINA_OPTS, restoring CATALINA_OPTS to its original state.
4. Delete the glass box directory from the Tomcat server, unless it is still to be used by other web application servers.
5. Restart the Tomcat server.

Uninstall manually on a Tomcat Service server:

This section describes how to manually uninstall the glass box agent on a Tomcat Service server.

About this task

When glass box is installed on the application server, the installation includes a Java agent and a GBootstrap web application. To uninstall glass box both of these must be removed. This task requires access to your application server.

Procedure

1. Undeploy GBootstrap web application:
 - a. Log in to Tomcat Manager. The default location is `http://<server_name>:<port_number>/manager/html`
 - b. In the application list, select **GBootstrap.war**, then click **Commands > Undeploy**.
 - c. When asked to confirm, click **OK**.
2. Shut down Tomcat Service.
3. Remove the Java agent from JVM arguments:
 - a. Locate Tomcat JVM at `...\Tomcat 7.0\bin\tomcat7w.exe`
 - b. Double-click on it, and select **Properties > Java** tab.
 - c. In the Java Options area, remove the following line (if it exists): `-javaagent:c:/path/to/gbAgent.jar`
 - d. Click **OK**.
4. Restart Tomcat Service.

Uninstall manually on a WebLogic server:

This section describes how to manually uninstall the glass box agent on a WebLogic server.

About this task

When glass box is installed on the application server, the installation includes a Java agent and a GBootstrap web application. To uninstall glass box both of these must be removed. This task requires access to your application server.

Procedure

1. Undeploy GBootstrap web application:
 - a. Log in to WebLogic Administration Console. The default location is `http://<server_name>:<port_number>/console/`
 - b. In the Domain Structure pane, click **Deployments**, then select the **GBootstrap** checkbox, and click **Delete**.
 - c. In the Domain Structure pane, click **Deployments**, and verify that GBootstrap has been removed.
2. Shut down WebLogic server.
3. Remove the Java agent from JAVA_OPTIONS: During manual installation, JAVA_OPTIONS is edited to include gbAgent.jar. Refer to the installation instructions to remove gbAgent.jar from JAVA_OPTIONS, restoring JAVA_OPTIONS to its original state.
4. Delete the glass box directory from the WebLogic server, unless it is still to be used by other web application servers.
5. Restart the WebLogic server.

Uninstall manually on a WebLogic Service server:

This section describes how to manually uninstall the glass box agent on a WebLogic Service server.

About this task

When glass box is installed on the application server, the installation includes a Java agent and a GBootstrap web application. To uninstall glass box both of these must be removed. This task requires access to your application server.

Procedure

1. Undeploy GBootstrap web application:
 - a. Log in to WebLogic Administration Console. The default location is `http://<server_name>:<port_number>/console/`
 - b. In the Domain Structure pane, click **Deployments**, then select the **GBootstrap** checkbox, and click **Delete**.
 - c. In the Domain Structure pane, click **Deployments**, and verify that GBootstrap has been removed.
2. Shut down WebLogic Service.
3. Remove the Java agent from JAVA_OPTIONS: During manual installation, JAVA_OPTIONS is edited to include gbAgent.jar. Refer to the installation instructions to remove gbAgent.jar from JAVA_OPTIONS, restoring JAVA_OPTIONS to its original state.
4. Restart the service.

Uninstall manually on a WebSphere server:

This section describes how to manually uninstall the glass box agent on a WebSphere Service server.

About this task

When glass box is installed on the application server, the installation includes a Java agent and a GBootstrap web application. To uninstall glass box both of these must be removed. This task requires access to your application server.

Procedure

1. Undeploy GBootstrap web application:
 - a. Log in to WebSphere's Integrated Solutions console. The default location is `http://<server_name>:9043/console/`

Note: If more than one profile is defined, log in to the console of the appropriate one.
 - b. Select **Applications > Application Types > WebSphere Enterprise Application**.
 - c. In the Enterprise Applications area, select the **GBootstrap** box, and click **Uninstall**.
 - d. Follow the wizard's instructions to remove the GBootstrap application.
GBootstrap is removed from the list of applications.
 - e. Select **Servers > Application Servers**.
 - f. In the right pane, click on your server's name.
 - g. Click on the **Configuration** tab to bring it to the front.
 - h. In Server Infrastructure, click **Java and Process Management > Process Definition**.
 - i. In Additional Properties, click **Java Virtual Machine**.
2. In Generic JVM Arguments, remove the glass box agent by clearing `-javaagent:c:/path/to/gbAgent.jar`

Note: This example applies to Windows operating systems; for other systems adjust the path as appropriate.

3. Click **Apply**, and save the configuration if asked.
4. Restart WebSphere.

For .NET platforms

Installing and using the glass box agent on a .NET server.

Installing the glass box agent

This section describes how to install the glass box agent on a .NET server.

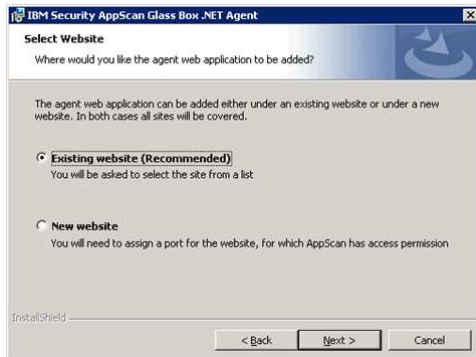
Before you begin

The AppScan glass box agent must be installed on your application server (or servers), by copying certain files from the AppScan installation to the server. To do this you will need access to your application server. The following systems and technologies are supported:

Item	Details
Operating System	Supported operating systems (both 32-bit and 64-bit editions): • Microsoft Windows Server 2012 • Microsoft Windows Server 2012 R2 • Microsoft Windows Server 2008 R2
Other	Microsoft IIS 7.0 or later Microsoft .NET Framework 4.0 or 4.5 must be installed, and IIS must be configured at the root level to work with this version of ASP.net

Procedure

1. Open ...\\Program Files\\IBM\\AppScan Standard\\Glass box
The absolute path will depend on where you installed AppScan.
2. Locate GB_DotNET_Setup.exe and copy it to your web server machine.
3. Double-click on it to start the wizard.
4. Follow the wizard steps.



5. When you are asked whether to install on a new or existing website, select the recommended option (unless you are sure the alternative is better).

Note: Which of the two options is recommended depends on your system. Where possible an "existing website" will be recommended, but this is not always possible, and "new website" may be recommended in some cases. In general "existing website" is preferred because it does not require a new port, which would then need to be configured in AppScan, and might also require additional firewall configuration to be accessible to AppScan.

6. This step depends on your choice in the previous step:
 - a. **Existing site:** If you chose "Existing site", you are now asked to select a site from a list of existing sites.

Note: When there is more than one site, the wizard will list them in order of preference. If the wizard identifies some sites as potentially problematic, it will list them lower down, and label them "not recommended". However, the order, and even the "not recommended" label are only suggestions, and in some cases you may decide to select a site that is labeled "not recommended".

Tip: Whichever selection you make will enable monitoring of all sites, but where possible try to select the site which you plan to configure as the starting URL of your *first* scan.

- b. **New site:** If you chose "New site" you are now asked to assign a port. You must define a port that will allow access to AppScan when scanning from a remote machine.

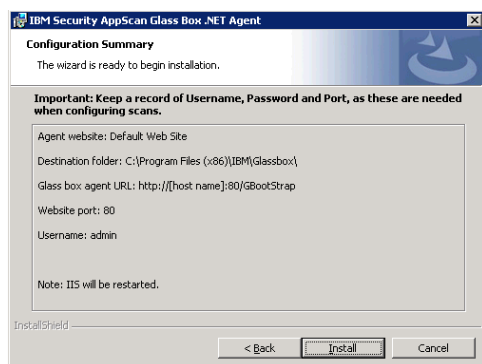
Important: You will need to supply the port number when configuring AppScan.

Note: You can use the **Test** button to verify that the selected port is currently available. However, this will automatically be done when you click **Next**.

7. Set the Agent Username and Password for the glass box web application agent.

Important: You will need to supply these credentials when configuring a scan with glass box.

8. The summary screen appears before the actual installation takes place.



Important: Before clicking **Install** make sure you have a record of the Username, Password and Port, as you will need these when configuring AppScan.

9. Click **Install**.

Note: The installation includes a restart of your IIS.
When the process completes a final message appears indicating its success or failure.

Note: If the installation fails for any reason, the final dialog will provide a link to the installation log for use in troubleshooting. It may be worth attempting to install on a different or new site on the server, even a different site that was labeled "not recommended". In all cases you must uninstall before attempting installation again. The location of the two installation logs is the %temp% folder. The names of the two installation logs are:

glassbox_setup_api.txt
glassbox_setup_msi.txt

Note that the uninstall process includes a restart of your IIS.

Note: Glass box logs are saved to:
C:\Program Files (x86)\IBM\Glassbox\GBootstrap\GlassBoxLog.log

Defining the glass box agent in AppScan

This section describes how to configure AppScan to access a server-side glass box agent.

About this task

Once you have installed the glass box agent on your application server, you must define it to AppScan so it can access the agent. After this is done AppScan will automatically use the agent for glass box scanning in any relevant scan (unless you deselect it in the scan configuration).

Restriction: You can define the same server agent on several AppScan machines, however an agent can be used for only one scan at a time.

Restriction: You can define multiple agents, but only one agent can be selected in a scan configuration.

Procedure

1. In **Configuration > URL and Servers tab**, set the **Starting URL**. Note that in this example, the URL includes port 8080.
`https://altdoro.testfire.net:8080/`
2. Click **OK** to close the Configuration dialog box.
3. Click **Tools > Glass box agent management**.
The Glass Box Agents dialog box appears, listing any agents that have already been defined.

4. To add a new agent to the list, click .

The Glass Box Agents Definition dialog box appears. The Glass Box Agent URL is automatically filled, based on the Starting URL you entered previously.

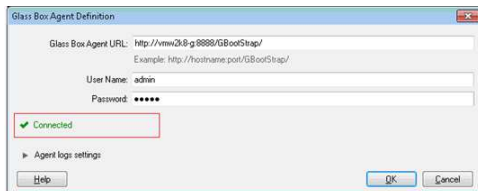
Important: The Glass Box Agent URL automatically contains a port value extracted from the Starting URL, (in this case 8080). If you installed your agent on a different port, replace the port value with the correct one. In this example the port value has been changed to 8888.

`http://altoro.testfire.net:8888/GBootstrap/`

5. Fill in the remaining fields and options as needed.

Option	Description
Username and password	Type in the username and password that were defined when the agent was installed on the server.
Agent log settings:	(Optional:) Click to open these settings. Note: Agent log settings are saved in the agent on the server-side.
Max. log rows	(Optional:) Use the slider to limit the size of the log.
Log content	(Optional:) Select the level of information to be include in the log: Errors: Include only error messages; Warnings: Include error and warning messages; Info: Include errors, warnings and informational messages; Debug: Include all messages. Glass box logs are saved to: [Installation folder]\instrumentation.log

AppScan tries to connect to the agent server. If successful, a green "Connected" icon appears. If there is any problem, a red icon will appear with a message such as "Unable to connect to agent" or "Credentials needed".



Note: If your site offers both HTTP and HTTPS, verify that the starting URL is HTTPS (and change it to HTTPS if it is not).

Note: If you get an error message, refer to

<http://www.ibm.com/support/docview.wss?uid=swg21567723>

6. Click **OK** to close the definition dialog box.
The agent is added to the list.
7. Click **OK** to close the list.
The agent is now defined on this machine.

Configuring a glass box scan

Although the scan is configured automatically, this section describes options you may want to change.

Procedure

1. Configure your scan as usual.

2. In **Configuration > Glass Box tab**, select one of the defined glass box agents from the drop-down list, and optionally adjust the settings:

Setting	Details
Use this glass box agent	If the glass box agent has been installed on your application server, and defined in AppScan, you can select it for use in the scan. If you have entered a Starting URL, AppScan attempts to select the appropriate agent automatically. When an agent is selected, AppScan attempts to connect to it, and indicates whether this was successful. Note: If you select an agent and get the message "Credentials needed", check that the credentials supplied in Tools > Glass Box Management are correct. If the required server does not appear in the drop-down list, you can define it by clicking the Glass box agent management link. Restriction: Only one glass box agent can be selected for use in a scan. If the application being scanned has more than one server, you must scan using each server agent separately.
Use glass box in the Explore stage	(Selected by default.) This function can increase coverage of the site, by examining the server-side source code for the presence of parameters that affect the behavior of the server, but do not appear in the response. Example server-side code: <pre>String debugOn = request.getParameter("debug"); if (debugOn == "true"){ response.getWriter().println(SECRET_SERVER_DATA); }</pre> In this example the developer has left the parameter "debug" in the code. It does not appear in any link on the site, but if an attacker were to send a request containing it, SECRET_SERVER_DATA could be obtained.
Use glass box in the Test stage	(Selected by default.) Select this check box to send glass box tests during the Test stage of the scan. This function can verify the success or failure of certain tests, such as Blind SQL Injection, with greater accuracy, and also reveal the existence of certain security issues that cannot be detected by black box techniques.
Skip equivalent black box tests	(Cleared by default.) This means that both glass box tests and black box tests for the same vulnerability (WASC Threat Classification) are sent. This is because although the glass box tests are generally both more accurate and give more detailed results, occasionally a glass box test may fail while the equivalent black box test succeeds. If the results for your application are unchanged when black box tests are skipped, you can reduce scan time by selecting this check box.

The status bar indicates that glass box scanning is enabled, and you are ready to start the scan.



Scanning with glass box

This section describes glass box scanning.

About this task

Once you have defined the glass box agent to AppScan, glass box scanning is enabled by default. You can use the Scan Configuration dialog box to verify that the correct server agent is selected, and glass box scanning is configured to run as part of the scan.

Glass box scanning can discover hidden URLs in the Explore stage, and additional issues and information during the Test stage.

Procedure

1. Click **Configuration > Glass Box** view.
2. Select the agent you want to use from the drop-down list.

Note: If your agent does not appear in the list, click the **Glass box agent management** link, and define it.

3. Verify that one or both of the two main glass box scan options are selected:
 - Use glass box in the Explore stage
 - Use glass box in the Test stage

Note: The **Skip equivalent black box tests** check box is cleared by default. This means that both glass box tests and black box tests for the same vulnerability (WASC Threat Classification) are sent. This is because although the glass box tests are generally both more accurate and give more detailed results, occasionally a glass box test may fail while the equivalent black box test succeeds. If the results for your application are unchanged when black box tests are skipped, you can reduce scan time by selecting this check box.

4. Click **Scan > Full Scan** to start the scan.

The scan starts and the status bar messages indicate that glass box scanning is active.



The scan results will include glass box data in the Issue Information tab where available.

Uninstalling the glass box agent

This section describes how to uninstall the glass box agent from your .NET server.

About this task

To perform this task you will need access to your application server.

Procedure

Do one of the following:

- Use the Microsoft Windows Add/Remove Programs feature.
- Run GB_.NET_Setup.exe again, and select the Uninstall option in the wizard.

Partial scans

Several options on the Scan menu let you perform part of a scan, or scan part of your site.

Task	Description
Explore Only	You can let AppScan Explore your site and build the site model, but not proceed with the Test stage of the scan. That way you can examine the site model and application data gathered to verify that it meets your requirements, and also explore manually if necessary, before proceeding with a Full Scan.
Test Only	If you previously ran Explore Only, or if you stopped a full scan before it was complete, you can let AppScan Test your site using existing Explore results. This can save scan time if, for example, the site's structure has not changed, but you want to see the effect of security changes that have been made to it. Note: In a Full Scan there are often parts of the site that are revealed on during the Test stage. When that happens AppScan runs an additional phase of Explore and Test stages. There may be several such phases in a scan. If you scan using the Explore Only and Test Only functions, the result is a single-phase scan, which may not include all parts of the site.

Task	Description
Re-Scan (Full)	Deletes current scan results, and runs a full scan using the current configuration.
Re-Explore	Deletes current scan results and runs an Explore stage only using the current configuration.
Re-Test	Deletes current Test results and runs a new Test stage using the current configuration and Explore results. Limitation: If the response during Re-Test is identical to the original response, the result is not updated, and shows the timestamp of the original response.

Scan Multi-Step Operations Only

You can define a multi-step operation and use it to test just that part of the site.

About this task

A multi-step operation is a sequence of requests that must be sent in a particular order to reach certain parts of the application. (For more details, see “Multi-Step Operations view” on page 84.) In some cases these operations may be the only part of the application you want to scan.

Procedure

1. Configure the scan, including at least one multi-step operation. (See “Multi-Step Operations view” on page 84.)
2. (Recommended:) Run Scan Expert to evaluate the configuration. (See “Scan Expert” on page 143.)
3. On the Scan menu, click **Scan Multi-Step Operations Only**.
AppScan scans the configured operations, and displays the results.

Note: Note that only tests directly related to the sequence are sent. (Site infrastructure tests, for example, are not sent.) This means that when the scan is over, the Status Bar (bottom left) may indicate that there are more tests to be sent. If you are using the Scan Multi-Step Operations function as a way of running a *full* scan on *part* of the site, you should continue with Test Only (next step), to send these tests.

4. (Optional:) Click **Scan > Test Only** to complete a full scan of the part of the site being tested.
Infrastructure and other remaining tests are sent to the site, and the results are added to the scan results.

Changing the configuration during a scan

If you change the configuration after you have started a scan, you will need to rerun the scan, or at least the Test stage, to see the effect of the changes. In general:

- If you changed Explore configurations, you should fully re-scan the application, by running Explore and Test (**Scan > Re-Scan > Re-Scan (Full)**).
- If you changed the Test configurations, you need not run the Explore stage again (if it is complete), and can simply re-test the application (**Scan > Re-Scan > Re-Test**).

Exporting scan results

When a scan is complete, the results are displayed on the main window. The different views (Issues, Remediations, Application Data) offer the scan results, filtered for usage.

You can export the scan results from AppScan in different ways:

- Configure and generate an AppScan report; export to PDF or other readable, portable format. See Chapter 10, “Reports,” on page 209.

- Select test variants from Issues and allow AppScan to attach a zip file of the variant information to a new e-mail. See Chapter 8, “Results: Security Issues,” on page 185.
- Generate a database or an XML file from the complete scan results. See “Generating scan result DB and XML files” below.

Generating scan result DB and XML files

Before you begin

You can export the complete scan results as an XML file, or as a relational database. (The database option exports the results into a Firebird database structure. This is open source, and follows ODBC and JDBC standards.)

The schema for the XML output is named **ScanExport.xsd** can be found in your AppScan \Docs folder, for example:

[AppScan Standard installation folder]\Docs\ScanExport.xsd

Procedure

1. Click **File > Export** and select **XML** or **DB**.
2. Browse to the location you want, and type in a name for the file.
3. Click **Save**.

What to do next

“Accessing the information in an exported database file”

Accessing the information in an exported database file

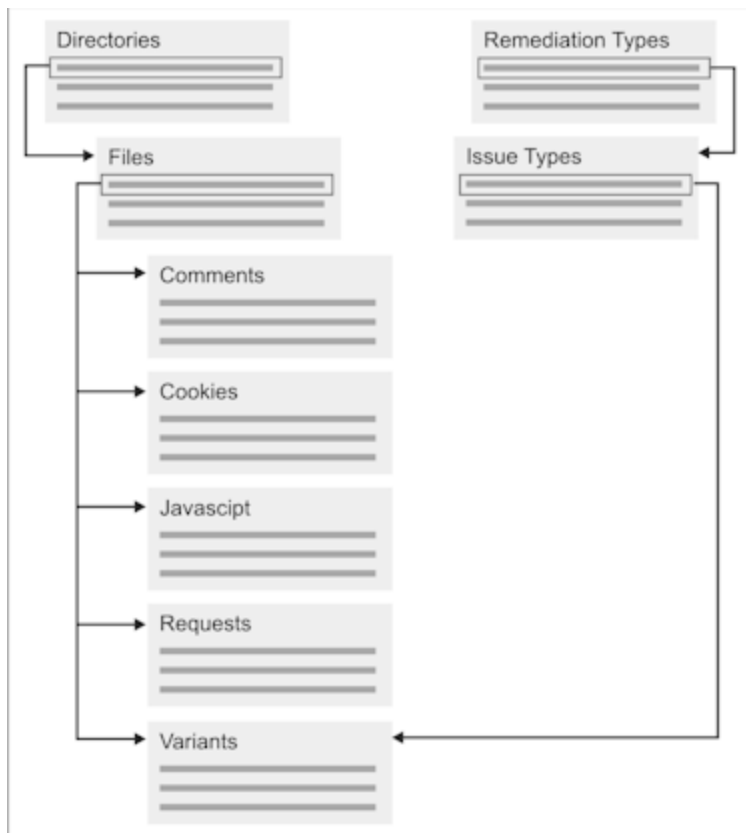
Procedure

1. Download the Firebird database engine: <http://firebird.sourceforge.net/index.php?op=files&id=engine>
2. Download a Firebird driver (choose one):
 - **Firebird ODBC driver:** <http://firebird.sourceforge.net/index.php?op=files&id=odbc>
 - **Firebird JDBC driver:** <http://firebird.sourceforge.net/index.php?op=files&id=jaybird>
3. Open Firebird with username SYSDBA and password masterkey, using any ODBC or JDBC compliant client.

Firebird database structure

When scan results are exported into a Firebird database structure, they can be viewed using one of many ODBC and JDBC database viewers. The structure of the relevant database components is illustrated in the figure below, and described in the following sub-sections.

Note: Some fields in the database are internal AppScan fields and not relevant to the user. These fields are marked "AppScan Internal" in the tables following.



Directory

The Directory section of the database contains a line for each directory or sub-directory from the scan.

Field Name	Explanation/Comments
ID	AppScan ID for the directory
NAME	Directory name
PARENTID	Directory in which this directory is contained (if this is a sub-directory)
PATH	Directory path
DIRTYPE	Directory type: Host/Application

Files

The Files section of the database contains a line for each file.

Field Name	Explanation/Comments
ID	AppScan ID for the file
FILENAME	Filename
PARENTID	ID of the directory that contains this file

Comments

The Comments section contains a line for each HTML comment found in the site page.

Field Name	Explanation/Comments
ENGINEID	AppScan ID for the comment
FILEID	ID of file where comment was found
SHORTTEXT	Text of comment (may be truncated)

Cookies

The Cookies section contains a line for each cookie found.

Field Name	Explanation/Comment
ID	AppScan ID for the cookie
REQCOOKIEID	AppScan Internal
RESPCOOKIEID	AppScan Internal
NAME	Name of cookie file
COOKIEVALUE	Value of cookie
SETINURL	Source URL for the cookie
FILEID	AppScan ID of the file where the cookie is saved
PATH	Path attribute of cookie
COOKIEDOMAIN	Domain attribute of cookie
EXPIRES	Date cookie expires
SECURE	Secure attribute of cookie

Issue types

The Issue Types section contains a line for each issue found.

Field Name	Explanation/Comment
ENGINEID	AppScan ID for this issue
NAME	Issues name as it appears in the GUI
INVASIVE	Whether issue is invasive: Y/N
SEVERITY	Severity code of this issue as shown in the AppScan GUI: 4 = All 3 = High 2 = Medium 1 = Low 0 = Informational
THREATCLASS	Threat classification as it appears in the GUI
THREATCLASSREFERENCE	URL to Internet reference on this classification (if relevant)
REMEDIATIONTYPEID	AppScan ID for the Remediation Type
ADVISORYID	AppScan Internal

Field Name	Explanation/Comment
ENTITYTYPE	AppScan Internal
INFRASTRUCTURE	AppScan Internal

Javascript

The Javascript section contains a line for each Javascript found in the site pages.

Field Name	Explanation/Comments
ENGINEID	AppScan ID for the Javascript
FILEID	ID of file where Javascript was found
SHORTTEXT	Text of Javascript (may be truncated)

Remediation types

The Remediation section contains a line for each Remediation.

Field Name	Explanation/Comments
ENGINEID	AppScan ID for the remediation type
NAME	Name of the remediation type as it appears in the GUI
REMEDIATIONPRIORITY	Remediation priority code (1=highest)

Requests

The Requests section contains a line for each test request sent.

Field Name	Explanation/Comments
ENGINEID	AppScan ID for the test request
FILEID	AppScan ID of the file to which the request was sent
URL	URL of the request
QUERY	Parameters sent in the request
STATUS	AppScan Internal
REASONID	AppScan Internal
XMLTYPE	AppScan Internal
LOGINTYPE	AppScan Internal

Script parameters

This section contains a line for each parameter that was sent in test request.

Field Name	Explanation/Comments
ENGINEID	AppScan ID for the parameter
FILEID	ID of file where parameter was sent
NAME	Actual parameter that was sent
PARAMETERTYPE	Parameter type: Get/Post

Variants

The Variants section contains a line for each variant.

Field Name	Explanation/Comments
ENGINEID	AppScan ID for the variant.
ENTITYNAME	Name of cookie to which variant sent (is applicable)
FILEID	ID of file to which variant was sent
ISSUETYPE	Issue type name as it appears in the GUI
REMEDIATIONTYPE	AppScan Internal
SEQUENCEINDEX	AppScan Internal

Chapter 7. Results: Application Data

AppScan offers three ways of viewing and working with scan results: Application Data, Security Issues, and Remediation Tasks. This section describes Application Data View.

This view is useful for verifying that all the parts of the site that you want covered by the scan are indeed explored, before you begin the Test stage of the scan. It shows results from the Explore stage only, without reference to the Test stage.

On the **View Selector**, click:



Application Data: Application Tree

The **Application Tree** shows explored folders, URLs, and files.

After the Explore stage, you can review the **Application Tree** to easily view your application and make sure that everything was explored.

The node that you select in the **Application Tree** acts as a filter for the data listed in the **Result List**. If there are no results for a specific type of data, select a higher node in the tree. If you select **My Application**, all data of a specific type are listed.

Application Data: Result List

The Result List shows lists of URLs, parameters, and scripts that were discovered during the Explore stage, *for the selected node in the Application tree*. The table below lists the categories of data.

Data Type	Description
"Requests" on page 178	URLs that AppScan accessed.
"Parameters" on page 178	Scripts and their associated parameters that AppScan discovered.
"Cookies" on page 179	Cookies that AppScan discovered.
Pages	All pages explored by AppScan.
"Failed Requests" on page 180	Links that did not respond to requests.
"Filtered URLs" on page 180	URLs that were not explored, either because of default AppScan settings or because of your defined Explore Filters (see "Exclude Paths and Files view" on page 57).
"User Interaction Needed" on page 181	URLs that require user input which AppScan could not provide automatically. To learn how to define form parameter input, see "Login tab" on page 44.
"Comments" on page 182	Comments on your web pages that are accessible to users.
"JavaScript" on page 182	JavaScripts that AppScan discovered.

In the Result list:

- Mouse-over each data category (Requests, Parameters, and so on) to see the number of items in that category (for the selected node in the application tree).
- Click on a category to display the items in that category (in the Result list)
- Click on an item in the Result list to display its details in the Detail Pane

The Result List and the Detail Pane for each data type are described in the following sections.

Requests

This view lists requests for which AppScan received a valid response. Based on these responses, AppScan generates tests, to reveal weaknesses in the site, that will be sent during the Test stage.

If the request or the response body contains XML (including XHTML or SOAP), the Visited URLs icon is replaced with the XML icon: 

Requests in the Result List

The **Result List** shows the URL of each page that AppScan visited, as well as method and parameters.

- You can view a visited URL by right-clicking on a request and clicking **Show in Browser**, or selecting the URL and clicking the **Show in Browser** link in the **Detail Pane**.
- You can create a manual test for the URL by right-clicking the **Visited URL** item and clicking **Manual Test**, or selecting the URL and clicking the **Manual Test** link in the **Detail Pane**. (See “Manual tests” on page 200 to learn more.)

Requests in the Detail Pane

The **Detail Pane** includes links to **Show in Browser** and **Manual Test**, which operate in the same manner as the right-click commands in the **Result List**.

The **Request/Response** tab is displayed and shows the request and immediate response of the URL selected in the **Result List**.

Parameters

These are requests that included one or more parameters.

The **Result List** shows all the parameters found during the Explore stage. The URLs in this list are the ones most likely to be vulnerable to malicious attacks. This list can be crucial in evaluating whether the scan has generated a useful set of test requests.

For each parameter in the Script Parameters list, AppScan shows the name, type, value, and URL (Results pane) and value (Detail Pane), and whether it is tracked. One parameter name may be listed more than once, if it is on different URLs, or if it has different values on the same URL.

The table below shows the options available when you right-click on an item in the list.

Table 8. Right-click options

Option	Function
Copy URL	To copy the selected URL to the clipboard.
Add to list in Parameters and Cookies tab	To add the selected parameter name (all values) to the list in the Configuration dialog box by right-clicking and selecting Add to Parameters and Cookies list. The Parameter Definition dialog box opens, to configure AppScan's treatment of this parameter.

Table 8. Right-click options (continued)

Option	Function
Exclude this path-parameter-value combination from the scan	To exclude a specific parameter value from the scan when it occurs in a particular URL. When you select the option, the Edit Exclusion or Exception dialog box opens, with the relevant data filled in. Example Consider a site with the following URL: <code>http://site/command</code> and a post parameter named 'action'. Each value triggers a different response from the server: • <code>action=login</code>, redirects to login page • <code>action=logout</code>, session expires • <code>action=clean</code>, server deletes user data For AppScan to be able to scan this site it must exclude <code>http://site/command</code> when <code>action=logout</code> or <code>clean</code> but not when <code>action=login</code> or other values. This can be done using this feature to exclude <code>http://site/command</code> with parameter name <code>action</code> and value <code>logout</code> or <code>clean</code>. For more details, see “Adding new exclusions or exceptions” on page 58
Do not test selected parameter(s)	To exclude one or more parameter names (all values) from the Test stage of the scan. This setting applies to all values of the specified parameter. It does not affect the Explore stage. The parameter name is added to the list in Parameters and Cookies view of the Configuration dialog box, with its Test Exclude value set as "Yes". For more details, see “Parameters and Cookies view” on page 67

Cookies

Cookies lists all cookies that AppScan found during the scan, whether set by responses, generated by Javascripts, or already on the host before the scan.

- The Result List shows all cookies found during the Explore stage. For each cookie the list shows whether tracked, name, set-cookie URL, values, whether excluded from testing, path, domain, expire date, whether secure, context.

Note: The URL listed is the one that leads to the response that set the cookie (even if there were others that included the Set Cookie command). If the cookie was not set by a response (for example, if it was generated by a Javascript, or already present on the host) N/A is displayed in the Response URL field.

- You can add any cookie to the list in the Configuration dialog box by right-clicking and selecting **Add this cookie to the Parameters and Cookies tab** list. The Parameter Definition dialog box opens, to configure AppScan's treatment of this cookie.
- When a particular cookie is selected the Detail pane shows:

Item	Description
Path	The specific folder or subfolder in your application to which this cookie is sent. The path attribute is used to specify the subset of URLs in a domain for which the cookie is valid. If a cookie has already passed domain matching (next item), then the pathname component of the URL is compared with the path attribute, and if there is a match, the cookie is considered valid and is sent along with the URL request.
Domain	To which domain or sub-domain this cookie will be sent. (If Domain is not set, the cookie is sent to the domain which issued the Set Cookie command, and all sub-domains.)
Expires	The date and time that the cookie will expire and be removed from the user's machine.

Item	Description
Secure	Yes (secured) or No. If a cookie is marked Secure , it will only be transmitted if the communications channel with the host is secure (currently only to HTTPS servers). If Secure is not specified, a cookie is considered safe to be sent over all channels.
Request URL	The first request that AppScan sent with the cookie.

Failed Requests

These are requests sent that did not return a valid response. This usually happens when the site is down or when other communication problems exist; or when a request returns an error response status instead of the requested page.

Note: If a URL in the application tree contains *only* error responses, the URL is displayed with strike through formatting (with a line through it).

- Where an error page is the expected response
- You can resend the request to a broken link, rather than repeat the entire scan. If the application has been changed or fixed since the Explore stage was done, perform the following procedure to re-explore the broken links.

Note: If your application requires a login, it is recommended that you manually log AppScan in (see “Login tab” on page 44) before doing the following procedure; otherwise the broken links will be explored outside of typical usage patterns.

- You can view all broken links by selecting **Failed Requests** in the Result pane.
- If you click **Retry All Failed Requests**, these links are removed from the Broken Links list and added to the Unvisited Links list. AppScan then continues the Explore stage, visiting the links in the Unvisited Links list whenever possible. This Explore stage will end when all the Unvisited Links have been explored.

Note: If there are communication problems between AppScan and your server during a scan, some links might be marked as Broken Links. When AppScan has communication problems, it attempts to resend its requests for 90 seconds. If the connection is not achieved within that time, the scan stops. The Notice panel on the main window notifies you of the problem and shows the timeout count down. If you see this notification, you should fix the connection between AppScan and your application before attempting to troubleshoot the links.

- The Detail Pane shows the response page of specific broken link, by clicking the **Show in Browser** button.

Filtered URLs

These are URLs that AppScan did not visit, because they were filtered out of the Explore: either by standard filters, or by filters you defined when you configured the scan (see “Exclude Paths and Files view” on page 57).

Filtered URLs in the Result List

The **Result List** shows the URLs that were not explored and provides the Filter Type (the reason why this page was filtered out).

You can view a filtered URL by right-clicking a Filtered URL and then clicking **Show in Browser**.

The following table lists the main filters that can be configured.

Filter name	What it means and how to configure it
Depth Limit	The URL was filtered due to the limit configured in Config > Explore Options > Scan Limits > Click Depth Limit .
File Extension Filter	The extension is one of those listed in Config > Exclude Paths and Files > Exclude File Types .
Likely Similar DOM	A page that was filtered from the scan because AppScan estimates that the response will have the same structure (DOM) as that of a previously explored page, and will contain no new elements to test. Controlled by Config > Explore Options > Main > Filter pages that are likely to be similar based on structure (DOM) .
Path Filter	The path is one of those listed in Config > Exclude Paths and Files > Exclude Paths .
Path Limit	The URL was filtered due to the limit configured in Config > Explore Options > Scan Limits > Redundant Path Limit .
Similar Body	A request (from a page that was not filtered due to Similar DOM) that was filtered from the scan because its response body content is similar to that of a request that was previously explored. Controlled by Config > Explore Options > Main > Filter similar pages based on structure (DOM) .
Similar DOM	A page that was filtered from the scan because its structure (DOM) is similar to that of a previously explored page, and probably contains no new elements to test. Controlled by Config > Explore Options > Main > Filter similar pages based on structure (DOM) .
Total Visited Links Limit	The URL was filtered due to the limit configured in Config > Explore Options > Scan Limits > Total Page Limit .
Untested Web Server	The domain is different to that of the Starting URL and is not one of the Additional Domain configured in Config > URL and Servers > Additional Servers and Domains .

Filtered URLs in the Detail Pane

The **Detail Pane** offers a link to show the URL in a browser: **Show in Browser**. This is equivalent to the right-click > **Show in Browser** in the **Result List**.

The **Request/Response** tab shows the request that would have been sent to the URL if it had not been filtered out.

Related reference:

“Exclude Paths and Files view” on page 57

Exclude Paths and Files view of the Configuration dialog box.

“Explore Options view” on page 62

Explore Options view of the Configuration dialog box.

User Interaction Needed

These are requests that were not sent because they require input from the user that AppScan could not supply. You can configure AppScan to provide input; see “Automatic Form Fill view” on page 80. If you missed some of your application parameters, or if you chose not to use the automatic form filler, AppScan will have a list of interactive URLs for you to review.

- You can examine the interactive URLs list. If you want these pages to be scanned, supply the required user information in a Manual Explore.
- It is recommended that you carefully examine the list of interactive URLs, fill in the required data, and send these requests. AppScan will then include these URLs during the Test stage.
- By enabling AppScan to send these requests, whole new parts of the site which were previously inaccessible may be reached. Therefore, after you visit interactive URLs, you should re-explore your application (**Scan > Re-scan > Explore**).

See “Manually exploring interactive URLs” on page 182).

Manually exploring interactive URLs

About this task

Interactive URLs are requests that were not sent because they require input from the user that AppScan could not supply. If you missed some of your application parameters, or if you chose not to use the automatic form filler, AppScan will have a list of interactive URLs at the end of a scan.

Procedure

1. In **Data view > Result List**, click **User Interaction Needed**.

The list of URLs with interactive input is displayed.

Note: A single URL may appear more than once, each instance being from a different form.

2. Right-click a URL in the list and click **Manually Explore this URL**.

The browser, with the **Manual Explore** buttons, appears, open to this URL. Complete the manual explore; see “Recording a Manual Explore” on page 126.

After you finish the manual explore, AppScan analyzes the new Explore.

If new URLs were found, a message appears, recommending that you continue the Explore (**Scan > Continue Scan > Explore**).

- If no new URLs were found but new tests were created, a message appears, recommending that you go on to the Test stage (**Scan > Continue Scan > Test**).
- If both new URLs were found and new tests were created, a message appears, recommending that you continue both the Explore and the Test (**Scan > Continue Scan > Full Test**).

Comments

Comments are HTML comments that AppScan found during the Explore stage. Comments hidden in an HTML page can contain information that hackers find useful: sometimes developers, intentionally or accidentally, leave comments for themselves or other developers in the final page. A hacker can harvest useful internal information from these comments, such as debug passwords.

Comments in the Result List

The list of comments shows the first line of the comment, and the first URL where it is located. If AppScan finds the same comment more than once, only the first instance is listed.

Comments in the Detail Pane

The **Detail Pane** shows the entire comment of the selected item in the **Result List**. Review the comments here to determine which should be removed from the final application.

JavaScript

JavaScript lists the JavaScript code that AppScan found during the Explore stage.

JavaScript in the Result List

The list shows the first line of the JavaScript, and the first URL where it was found. If the same script is to be found on more than one URL, only the first instance is listed.

JavaScript in the Detail Pane

The **Detail Pane** shows the entire script of the selected item in the **Result List**. Review the code here for comments that should be removed from the final application.

Application Data: Detail Pane

In Application Data view, the Detail Pane shows a variety of data and toolbar options, depending on the data type selected.

Data type	Toolbar options
Requests	Show in Browser, Set as Error Page, Manual Test, Search (for word/phrase), Search (for word/phrase).
Parameters	Requests (toggle through), Show in Browser, Set as Error Page, Manual Test, Search (for word/phrase).
Cookies	None.
Pages	Page information tab. Request/Response tab: Requests (toggle through), Show in Browser, Set as Error Page, Manual Test, Search (for word/phrase).
Failed Requests	Show in browser, Set as Error Page, Search (for word/phrase).
Filtered	Show in browser, Set as Error Page, Search (for word/phrase).
User interaction needed	Manually explore this URL, Search (for word/phrase).
Comments	None.
JavaScripts	Search (for word/phrase).

Chapter 8. Results: Security Issues

AppScan offers three ways of viewing and working with scan results: Security Issues, Remediation Tasks, and Application Data. This section deals with Security Issues View.

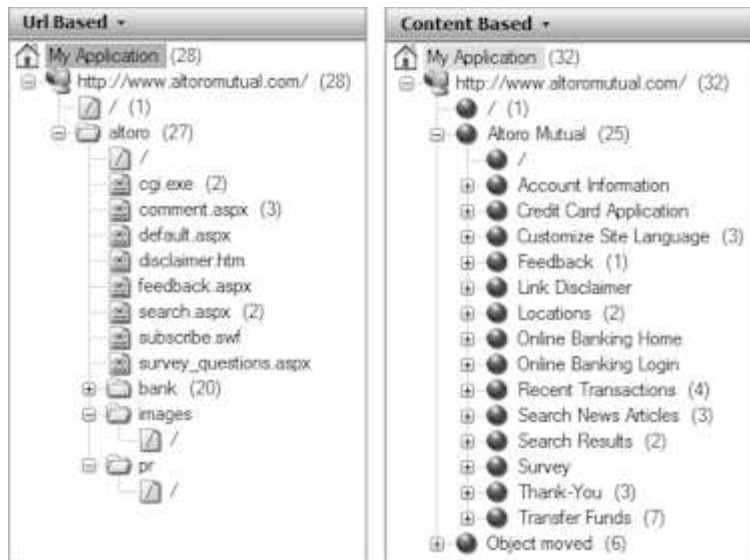
Security Issues View provides access to the results of a scan. You can view results at a high level or select specific tests or objects and access more details. These details include advisories, fix recommendations, requests/responses, and differences between the test variants that resulted in issues. You can manipulate the severity of issues, resend tests (with or without modifications), and create reports based on Issues.

In the **View Selector**, click:



Security Issues: Application Tree

The **Application Tree** displays the folders and files of the scanned application. Each node in the tree has a counter, which shows how many issues the node contains.



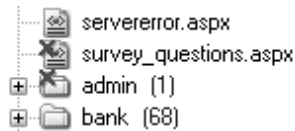
In the Application Tree you can:

- Select a node to filter the issues displayed in the **Result List**.
- Use the right-click menu to View in Browser, Manual Explore, Manual Test, Copy URL to Clipboard, and Exclude URL from Scan. (See “Application Tree right-click menu” on page 20.)
- If you have defined content-based rules, you can toggle between URL-Based view and Content-Based view by clicking the combo-box at the top of the pane. (See “Content-Based Results view” on page 90.)

Exclude URL from scan

Any URL or node in the Application Tree can be excluded from future scans by right-clicking on it and selecting **Exclude from Scan**. (To reinstate the URL or node, simply right-click again and select **Include in Scan**).

When a URL or node is excluded from scan a red X appears on its icon in the Application Tree.



When you exclude URLs in this way, "Exclude" items are added to the **Exclude or Include Paths** list (see "Exclude Paths and Files view" on page 57).

Security Issues: Result List

The Result List displays the issues relevant to the node selected in the Application Tree. If you selected the My Application node, the Result List shows all issues found in your web application.

Issues are grouped by Type. Under each Type, all URLs are listed. Under each URL, all issues are listed. (Individual variants of issues are not shown in the Result List, but can be seen from the Detail pane.)

Each node in the tree has a severity icon, indicating issue severity; and a counter, indicating how many issues of this type were found. In the case of Type and URL, the severity icon indicates the severity of the most severe issue included under the node.



You can change the way the issues are sorted, and you can manipulate their severity value (see "Severity levels").

You can designate issues you do not want to deal with currently as Noise, either removing them from the result display altogether, or having them appear with strikethrough (see "Issue state: Open or Noise" on page 187).

The total number of security issues (at the top of the list) is a measure of the vulnerable locations in the site, and depends in part on how site is structured. If you define a content-based structure (see "Content-Based Results view" on page 90), the total number of issues in the application tree may not be the same as it is for the URL-based application tree (for the same results). When site structure is content-based (rather than URL-based), and content-based view is configured correctly, the issue count in content-based view represents more accurately the number of "vulnerable locations" that exist in the site. The total number of *variants* (at the top of list in parentheses) is independent of site structure, and does not change between content-based and URL-based views.

Severity levels

In the case of issues, the severity icon indicates the severity level of the issue; in the case of Issue Types and URLs, it indicates the most severe of all issues under that node.

Icon	Indicates	Description	Examples
	High severity	Direct danger to your application, web server, or information	Executing commands on server, stealing customer information, denial of service
	Medium severity	Threat through unauthorized access to private areas, though the database and operating system are not at risk	Script source disclosure, forceful browsing
	Low severity	Allow for unauthorized reconnaissance	Server path disclosure, internal IP address disclosure
	Informational	Issues you should know about, not necessarily security issues	Insecure methods enabled

Related tasks:

“Changing severity levels”

Changing severity levels Procedure

You can change the severity assigned to any node by right-clicking on the node and selecting **Severity >**, followed by the new value.

Note: If you change the severity level of a Type or URL node, all issues under that node are changed to the new level.

Issue state: Open or Noise

Issues not relevant to your application can be designated as "Noise" and removed from the results.

About this task

If a specific issue that AppScan discovers is not relevant for your application (meaning that for your application it is effectively a "false positive" result), for example an issue that exists only in the development environment but will not exist in the deployment environment, you may decide to classify it as "Noise".

There are two display options for issues marked as noise: Included in the Result list but as gray text with a strikethrough, and Not included in the result list at all.

Procedure

To toggle between the two display options, click **View > Show Issues marked as Noise**.

Results

When a checkmark appears next to the menu item, issues marked as Noise will be included in the Result list but with gray text with a strikethrough.

Example

Noise definitions are applied in the following way:

- On the workstation where the scan is configured, noise designations are saved, and automatically applied to future scans, *when the scan is saved*. (The location of the Noise Classification file is defined in Tools > Options > General tab.)

- When you open a saved scan *even on a different workstation*, these issues will be designated as noise *for this scan*, even if the workstation has different definitions. (However, note that if you *save* the scan on the second workstation its definitions will be saved to that workstation, overwriting any previous Noise Classification file there.)

What to do next

See also:

“Changing Issue State”

“Test Options view” on page 100

Changing Issue State

You can change the state assigned to any node.

Procedure

Right-click on the node and select **State >**, followed by the new value ("Open" for relevant issues; "Noise" for issues not relevant to your application).

Note: If you change the state of a Type or URL node, all issues under that node are changed to the new state.

Exporting and importing Issue states

If you have designated certain Issues as "Noise" (not relevant to your application), you can export these designations for use at other workstations.

About this task

To export Noise designations for use at other workstations:

Procedure

Click **File > Export > Cross-Scan Data**, and save the data as an XML file.

What to do next

To import the states at another workstation, click **File > Import > Cross-Scan Data**.

Resending tests

About this task

You can resend a test without running a complete Full Scan or Test stage. For example, if the results of a test seem inconsistent with the results of a previous scan, you can resend that one test.

Procedure

1. On the **Result List**, right-click a node.
2. On the menu that appears, click **Re-test**.

AppScan sends all the test requests included in the selected node, and the new results are added to the **Result List**.

Right-click menu

The right-click menu in the Security Issues Results List includes the following options:

Item	Explanation
Severity	Change the Severity value for the selected item (select High, Medium, Low, Informational)
State	Issue state is always Open by default. If for some reason you do not need to relate to certain issues, you can define them as Noise. (Options: Open/Noise)
Re-test	Resend the selected test and add the result to the scan results
Manual Test	Create a manual test (see "Manual tests" on page 200)
Delete	Delete the selected item from the test results (cannot be reinstated)
Set as Non-vulnerable	Set the result as non-vulnerable, so it will not appear in the test results (but can be viewed, and reinstated, from the Non-Vulnerable Variants list, see "Non-Vulnerable Variants List" on page 203)
Report False Positive	Zip and email test information to AppScan Support or to members of your own organization, see ("Report false positive test results" on page 199)
Generate Issue Information	Populates the Issue Information tab for the selected result only. Tip: To update Issue Information for <i>all</i> results, select Tools > Generate All Issue Information instead of this option.

Filtering Security Issues in Result List

You can filter the **Result List** for types of issues, or you can search for a specific issue.

Procedure

1. On the **Edit** menu, click **Find** (or press **Ctrl + F**).
The **Find** bar appears in the main window, below the **Result List**.
2. On the **Application Tree**, select a node.
 - If you select the **My Application** node, Find searches through all the results.
 - If you select a node within the tree, Find searches through the selected node and its sub-nodes.
3. In the **Look for** text box of the **Find** bar, type a string or partial string.
4. In the **Search in** combo box, select where in the results to search for the string. Options are: ID number of test, Security Issue, URL, CVE ID, CWE ID, XFID (X-Force ID), Request/Response data, Variant description, Paramater/Cookie Name, or All of these.
5. Click **Find Now** or press **Enter**.
Results appear in the **Result List**, overwriting the list that was previously displayed.
If you enter another **Find** string and click **Find Now** again, the new search is done on the selected node in the **Application Tree**, not on the displayed results of the previous search.

To find:	Look for:	Search in:
The variant with ID: "5016" (see Properties sub-tab of Detail Pane)	"5016"	ID
Coss-Site Scripting issues	"cross"	Test Name
Login pages with issues	"login"	Test URL
Issues relevant to password manipulation	"passwd"	Request/Response

Sorting the Result List

About this task

You can re-arrange issues in the Result List. By default, issues are sorted by severity.

Procedure

1. Click the **Arranged By** column header.
2. Click a command on the menu to sort the issues:
 - **Severity** - test names are listed by severity order: High Severity first, down to Informational.
 - **Count** - number of issues in each test. Tests that found the most issues are listed first.
 - **Name** - alphabetical order by test name.

The issues are re-sorted in the **Result List**.

To reverse the order of the sort (ascending or descending), click the second column of the **Result List**. Click the second column again to reset the order.

Security Issues: Detail Pane

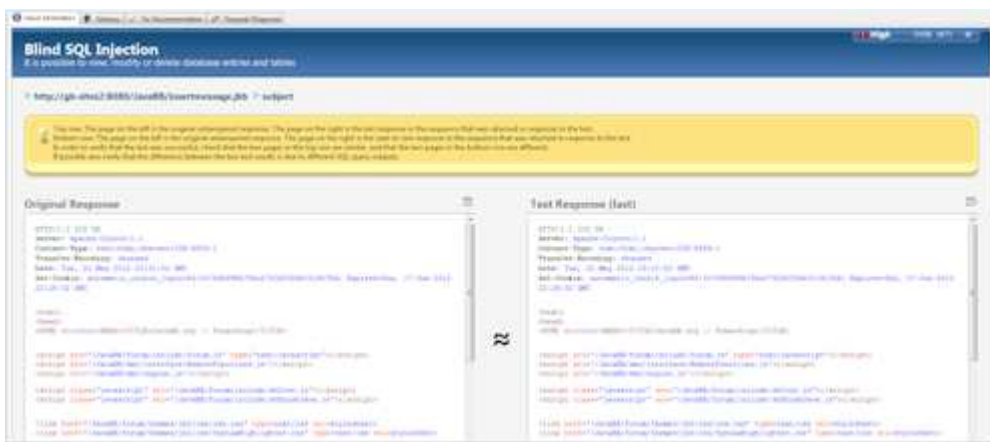
The Detail Pane shows relevant information for a selected test, and all its variants, selected in the **Result List**.

The Detail Pane has four tabs (clicking on a tab brings its contents to the front): “Issue Information tab,” “Advisory tab” on page 194, “Fix Recommendations tab” on page 196, “Request/Response tab” on page 197.

Issue Information tab

The first tab of the Detail pane summarizes the information available.

During scanning, as issues are discovered and added to the tree, the Issue Information tab provides a summary of the information available on the other Detail pane tabs, as well as valuable additional information such as CVSS Metric scoring for the issue, and relevant screen captures, that can be saved with the results and included in your reports.



Area/Icon	Description
Header	Issue header including URL, Entity and Security Risk (High, Medium, Low or Informational).
CVSS Metrics score	An average score based on the three CVSS metrics groups: Base, Temporal and Environmental. Click on the link to edit (see “CVSS settings” on page 192).

Area/Icon	Description
	Enables you to include screen captures, other relevant images of your choice, and your own comments with the scan results, and include them in reports (see “Editing Issue Information”).
Tip (yellow box)	This information refers to the Content area (below) and explains what to look out for in the image(s) or HTML that appear there.
Content (screen capture(s) or HTML code)	Depending on the issue, this area may include a screen capture, two screen captures for comparison, a screen capture with a simulated pop-up, or HTML code. In the case of HTML, you can toggle text wrap on and off by clicking  at the top right of the content area.
Reasoning (blue box)	Explains what AppScan did, and why it thinks this is an issue.
Technical summary (gray box)	The technical details of what AppScan did to test for this issue, and how it validated the response.

Issue Information toolbar

The toolbar above the Detail pane shows the current severity of the selected issue, and its state, and lets you toggle between issues.

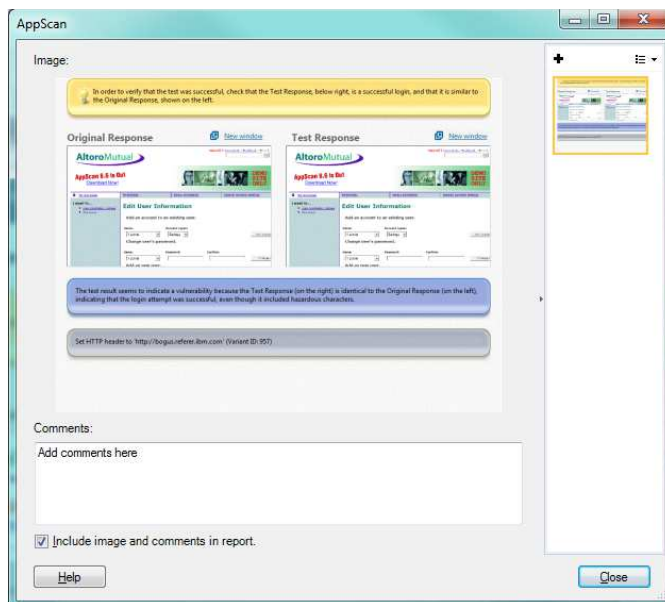
- **Severity:** Select one of the four standard severity settings, or manually adjust the CVSS settings for this issue.
- **State:** Options are "Open" or "Noise". Default is Open. Select Noise for issues that are unimportant to you. By default, results designated as Noise appear in the Result List with a line through them. To remove them from the display altogether, deselect **Tools > Show Issues Marked as Noise**.

Editing Issue Information

Add images and comments to the issue information, for inclusion in reports.

Issue Information tab screen captures, other related images of your choice, and your own comments can be saved with the scan results, and included in reports.

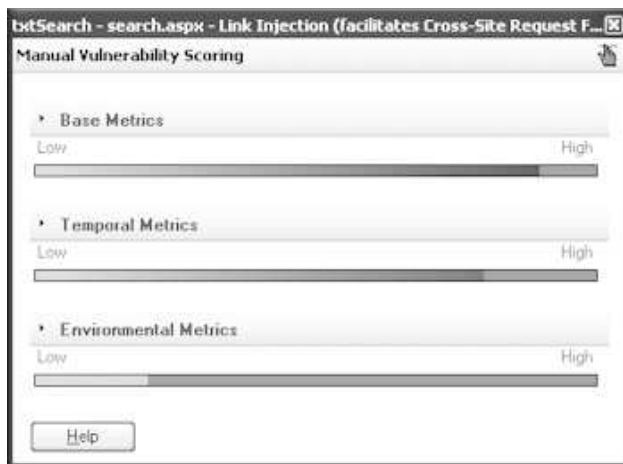
- Click  at the top right hand corner of the Issue Information tab to edit issue information.



Area/Icon	Description
Image	Shows the image currently selected from the available thumbnails in the right-hand pane.
	Click to add an image to this issue in the scan results.
	Click to save the current image to your computer.
	Click to delete the current image from the scan results.
Comments	Type in comments about the currently displayed image, to be saved with the scan.
Include image and comments in report	Select to include <i>the currently displayed image and comments</i> in reports. You can configure each image separately. All images are included by default.

CVSS settings

You can manually fine-tune the severity setting for a specific issue based on CVSS metrics. This is done from the Issue Information toolbar, by clicking **Severity > CVSS Settings**.



From the CVSS window, clicking on the name of one of the three sections opens that section for configuration. You can restore the default settings by clicking  , which becomes active when changes are made.

txtSearch - search.aspx - Link Injection (facilitates Cross-Site Request F... [X]

Manual Vulnerability Scoring

▸ Base Metrics

Low High

▸ Temporal Metrics

Low High

Exploitability [Proof of Concept]

Remediation Level [Workaround]

Report Confidence [Confirmed]

▸ Environmental Metrics

Low High

Help

Base metrics

These are metrics of the vulnerability that are constant over time and across user environments.

Metric	Explanation	Options
Access Vector	Whether the vulnerability can be exploited only locally, also from adjacent networks, or from any network connection ("remotely exploitable").	Local, Adjacent Network, Network
Access Complexity	The difficulty involved in exploiting this vulnerability.	High, Medium, Low
Authentication	The number of times an attacker must authenticate to exploit the vulnerability.	None, Single, Multiple
Confidentiality Impact	The impact on confidentiality if this vulnerability is successfully exploited.	None, Partial, Complete
Integrity Impact	The extent to which system integrity (the accuracy of information supplied by the application) is compromised if this vulnerability is successfully exploited.	None, Partial, Complete
Availability Impact	The impact on the availability of information resources if this vulnerability is successfully exploited.	None, Partial, Complete

Temporal metrics

These are metrics of the vulnerability that may change over time.

Metric	Explanation	Options
Exploitability	The current state of exploit techniques utilizing this vulnerability.	Unproven, Proof-of-Concept, Functional, High, Not Defined
Remediation Level	The level of remediation available to protect against the vulnerability.	Official Fix, Temporary Fix, Workaround, Unavailable, Not Defined
Report Confidence	The degree of confidence in the existence and technical details of the vulnerability.	Unconfirmed, Uncorroborated, Confirmed, Not Defined

Environmental metrics

These metrics reflect the application environment, and should be set globally using the Configuration dialog box > Environmental Metrics tab. Change them here only if this vulnerability is specific to a part of the application environment that has different characteristics.

Metric	Explanation	Options
Collateral Damage Potential	The potential for damage or theft if the application is vulnerable.	None, Low, Low-Medium, Medium, Medium-High, High, Not Defined
Target Distribution	The proportion of systems in the environment that are potential targets.	None, Low, Medium, High, Not Defined
Availability Requirement	The relative importance of availability (of information).	None, Low, Medium, High, Not Defined
Confidentiality Requirement	The relative importance of confidentiality (of user information).	None, Low, Medium, High, Not Defined
Integrity Requirement	The relative importance of integrity (accuracy) of information.	None, Low, Medium, High, Not Defined

Restoring default severity settings Procedure

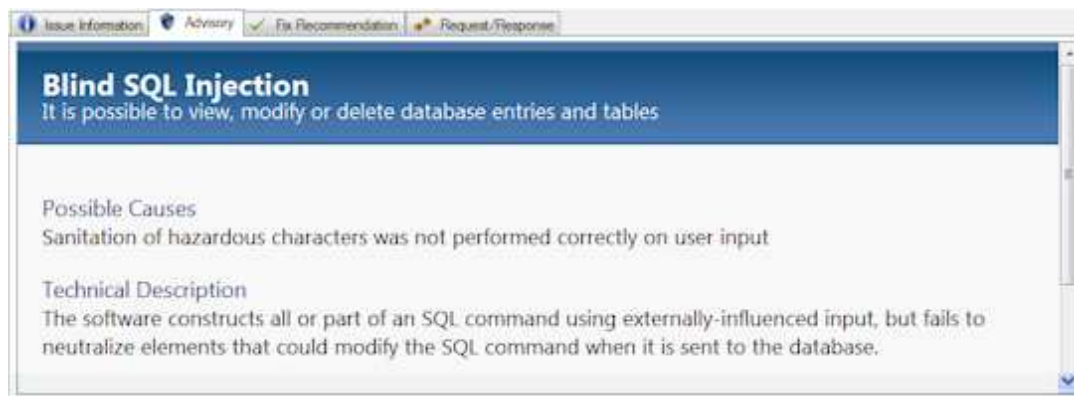
If you have manually changed the severity setting for a particular issue (by selecting a different severity or by adjusting the CVSS settings) you can restore the previous setting.

- To replace your manual setting (High/Medium/Low/Informational) with a setting based on the current CVSS settings for this issue: On the Issue Information toolbar, click **Severity** > **Use CVSS to calculate severity**.
- To restore the *default* CVSS settings: On the Issue Information toolbar, click **Severity** > **CVSS Settings**, and in the CVSS settings window that opens click 

Advisory tab

The second tab of the Detail pane is the advisory.

The information on the Advisory tab provides technical details on the selected issue and reference links for more information. This information is essential when you need to explain what has to be fixed and why.



The **Advisory** tab may include any of the following sections:

Test Name

The name of the test as it appears in the Result List.

Severity

The severity assigned to this vulnerability.

Type Whether this vulnerability is application-level or infrastructure-level.

WASC Threat Classification

An Internet link to the page of the web Application Security Consortium that describes this class of threat.

CVE ID(s)

The Industry Standard number(s) for this type of vulnerability (see “CVE support.”)

CWE ID(s)

The Industry Standard number(s) for this issue (see “CWE support” on page 196.)

XFID The X-Force ID for this issue (see “X-Force support” on page 196.)

Security Risk

An explanation of how this issue is a security risk to your application.

Training Module

An Adobe Flash presentation that explains the and demonstrates the issue.

Possible Causes

Suggests how the issue came to exist in your application

Technical Description

Detailed technical description of the issue

Affected Products

Third Party products that could be affected by the issue.

References and Relevant Links

Links to additional information.

CVE support

CVE (Common Vulnerabilities and Exposures) is an industry standard list that provides common names for publicly known information security vulnerabilities and exposures. This makes it easier to share data across separate databases and tools. (For more details see the CVE website at: <http://cve.mitre.org/>)

AppScan advisories for vulnerabilities that have been assigned a CVE ID, include the reference, with a link to the description on the CVE website. (CVE IDs begin with the letters CVE for accepted vulnerabilities or CAN for candidate ones.)

You can:

- View the CVE ID of a test result in the advisory for the test (see “Advisory tab” on page 194)
- Click on the CVE ID to link to its description on the CVE website
- Include CVE IDs (as part of the advisory) in reports
- See the CVE ID that is tested for by any particular test, in Test Policy view (Scan Configuration > Test Policy; see “Test Policy view” on page 95)
- List all tests that have CVE IDs by searching for CVE abd CAN strings in Test Policy view
- Search for a particular CVE by searching for its ID in Test Policy view

CWE support

CWE (Common Weakness Enumeration) is an industry standard list that provides common names for publicly known software weaknesses. This makes it easier to share data across separate databases and tools. (For more details see the CWE website at: <http://cwe.mitre.org/>)

AppScan advisories for vulnerabilities that have been assigned a CWE ID include the reference number, and a link to the description on the CWE website. Where specific vulnerabilities have their own CWE ID (in addition to the ID for the *issue*), this is shown in the Variant Details pane.

You can:

- View the CWE ID for an issue in the advisory for that issue (see “Advisory tab” on page 194)
- View the CWE ID for a variant, and its parent issue, in the Variant Details tab (see “Variant details” on page 198)
- Click on the CWE ID to link to its description on the CWE website
- Include CWE IDs (as part of the advisory) in reports
- Search for a particular CWE by searching for its reference number in Test Policy view

For the version of the CWE database used in the current version of AppScan, refer to the Release Notes located in [AppScan Standard installation directory]\Docs. If a daily update changes the database used, the change will be listed in the Update Log (**Help menu > Update Log**).

X-Force support

X-Force is one of the world's most comprehensive threats and vulnerabilities database. All AppScan advisories for vulnerabilities include their XFID, and a link to the X-Force site where you can find additional information about the security issue.

You can:

- View the XFID for an issue in the advisory for that issue (see “Advisory tab” on page 194)
- View the XFID for a variant, and its parent issue, in the Variant Details tab (see “Variant details” on page 198)
- Click on the XFID to link to its description on the X-Force website
- Include XFIDs (as part of the advisory) in reports
- Search for a particular XFID by searching for its reference number in Test Policy view

Fix Recommendations tab

The third tab of the Detail pane shows the fix recommendation.

The information on the Fix Recommendations tab is the exact tasks that should be done to make your web application secure against the specific selected issue.



The **Fix Recommendations** tab shows known recommendations for fixing the selected issue. These solutions may be quite involved, step-by-step instructions.

Fix Recommendations are sorted:

- **General** - always selected
- **.Net** - Microsoft© .NET
- **Java EE** - Sun© Java Platform, Enterprise Edition

Note: You can set AppScan to hide the Fix Recommendations that you do not find relevant. See “Preferences tab” on page 236 for details.

Request/Response tab

The fourth tab of the Detail pane is the Request/Response tab.

The **Request/Response** tab provides information about the tests, and their specific variants, that were sent to your web application to discover where it has weaknesses. A test may have multiple variants. A *variant* is a slight change to the original test request that AppScan sends to your web application server. (AppScan first sends a request that is meant to be legal and to follow the business logic of your application. Then it sends a similar request, modified for the purpose of discovering how your application handles illegal or mistaken requests. Each test request may have a number of variants; as many variants as needed to cover all the security rules in the extensive AppScan database.)

For example, consider a test sent to check that you have enforced user input rules for a specific parameter. One variant might checks that apostrophes are not valid input, another that quotation marks are not allowed.

The variant itself is shown in red text, and the validation (the part of the response that indicates the existence of a security issue) is highlighted in yellow.

Besides a large amount of explanatory information, the **Request/Response** tab provides advanced features for understanding and using the results of a scan.

The **Request/Response** tab has two panes and its own toolbar along the top. The toolbar and tabs are shown below, and summarized in the table following.

Tool	Function
Variant < >	Indicates the number of variants of the current test. Clicking the < and > icons toggles to the previous and next variant respectively.

Tool	Function
Test/Original	Toggles between the Original and Test information.
Next Highlight	(Available where validation text is highlighted). Moves cursor to the next highlighted text.
Show in Browser	Opens the built-in browser to show the current page, with the option of taking a screen capture from the browser. When the browser opens you can take a screen capture of the page by clicking the camera icon  on the browser toolbar. The screen capture is added to the Issue Information tab.
Options > Report False Positive	Use to email the current variant to the AppScan support team, or within your enterprise. (See “Report false positive test results” on page 199.)
Options > Manual Test	Modify test and save it as a manual test. (See “Manual tests” on page 200.)
Options > Delete Variant	Permanently deletes the selected variant from the test results (irreversible). This can also be done by right-clicking on the variant in the Result pane.
Options > Set as Non-vulnerable	Changes the definition of the selected variant to Non-vulnerable. Positive responses that were changed by the user to non-vulnerable are removed from the scan results, and do not appear in reports, but can be viewed (and reinstated) via the Non-Vulnerable Variants list. (See “Non-Vulnerable Variants List” on page 203.)
Options > Set as Error Page	Adds the current page to the list of error pages (Scan Configuration dialog-box > Error Pages) and updates the results to reflect the fact that this response is an error page.
Options > Add to Issue Information	Runs Result Review on the current issue, and adds any new information available to the Issue Information tab.
Find	Type in text to search for a specific string. (See “Filtering Security Issues in Result List” on page 189.)
Variant details	The right pane shows details for the current variant: ID, Description, Difference (the difference between this variant and the original request), Reasoning, and CWE ID.

Viewing variants

About this task

Each test can have multiple relevant variants; each variant changes a request slightly to check your application's security against numerous attack techniques.

Procedure

1. Click **Test**.
2. Click the right or left arrow buttons to view the variant requests.
For each variant test sent, the modified part of the request is highlighted in red. For a more descriptive explanation of the variant, see the **Variant Details** tab.

Variant details

Variant Details is a side tab in the Request Response tab of the Detail pane.

The **Variant Details** tab, within the **Request/Response** tab of the **Detail Pane**, describes the variant and explains its purpose.

Section	Description
ID	An ID number is assigned to each variant to simplify search and management.
Description	A brief description of the test.

Section	Description
Difference	Shows which changes were made to the original request for this test. The changes are highlighted in red. (Changes may include changing the value of a parameter, cookie or method; changing a path; deleting a parameter, deleting or adding an HTTP header; removing or adding a parameter to the body.)
Reasoning	Explains why this test result indicates a vulnerability.
CVE ID	The CVE ID of the vulnerability (see “CVE support” on page 195).
CWE ID	The CWE ID of the variant and its parent issue. (See “CWE support” on page 196).

Taking screen captures

You can take screen captures of your application that demonstrate the vulnerability, to include in reports.

Procedure

1. Select an item in the **Result List** and toggle through to the required variant.
2. In the **Detail Pane > Request/Response** tab, click **Show in Browser**.

3. In the browser, click the **Camera icon** 

A screen capture of the current page is added to the data in the Issue Information tab, and will be included in reports.

Report false positive test results

You can email test information to AppScan Support in order to:

- Report a result that AppScan classified as positive (indicates a vulnerability) but which you believe is negative (does *not* indicate a vulnerability).
- Ask AppScan Support why a result was classified as positive.

You can also use the feature to conveniently zip and email the result to the developers and auditors in your own organization.

Note: By default AppScan saves the data in an encrypted format, accessible only to Support personnel. If you are sending the file within your own organization, you must configure AppScan to save the information as a .zip file. In the **Tools | Options | General** tab, deselect the **Encrypt Attachments** check box.

Report a single false positive variant

Procedure

1. Select an item in the **Result List**.
2. In the **Detail Pane > Request/Response** tab, browse through the variants of the selected issue.
3. When the variant you want to send is displayed, click **Report False Positive** on the Detail pane toolbar.
The Report False Positive dialog box opens.
4. Click **Save File** and save the file to disk.
Depending on your Encryption Setting (see below), the file is saved in encrypted or .zip format.
5. To send the file to AppScan Support, click the **Browse to your support provider** link, log in and upload the file.

Report a false positive variant set

About this task

You can report all the variant information of any Issue, URL, or child item in one attachment, using the right-click menu in the Result List.



Procedure

1. In the **Result List**, right-click on an issue, URL or parameter, and select **Report False Positive**.
The Report False Positive dialog box opens.
2. Click Save File and save the file to disk.
Depending on your Encryption Setting (see below), the file is saved in encrypted or .zip format.

Note: By default the file is encrypted. To change this setting go to **Tools > Options > General > Report False Positive**, and deselect the **Encrypt attachments** check box.

3. To send the file to AppScan Support, click the **Browse to your support provider** link, log in and upload the file.

Encryption of false positive reports

About this task

By default the Report False Positive feature saves the data in an encrypted format that only AppScan Support personnel can open.

If you will be sending the attachment within your own organization, the recipient will not be able to decrypt it, so you need to disable the encryption functionality.

Procedure

1. Click **Tools > Options > General > Report False Positive**.
2. Select/deselect the **Encrypt attachments** check box as required.

Manual tests

About this task

The **Manual Test** feature lets you send your own tests and save them as security issues, to be included in your reports.

You can create a manual test as soon as you have Explore results. Manual tests are saved for the current scan only. Note that if you re-scan the test will be lost.

You can base your manual test on an existing test, or you can create a new one from scratch.

Procedure

1. To base the manual test on an existing variant:
 - On the **Result List**, click a test variant, or
 - On the **Result List**, click a test, then in the **Detail Pane** use the toolbar to browse to the required variant.

Alternatively, to create a new variant from scratch, simply open the **Manual Test** dialog box (next step) without selecting an existing variant.

2. Open the **Manual Test** dialog-box:

- On the **Tools** menu, click **Manual Test**, or
- Right-click on a node in the **Application Tree** or the **Result List**, and select **Manual Test** from the pop-up menu, or
- Click the **Manual Test** button in **Application Data > Detail Pane** for the selected variant.

The **Manual Test** dialog box appears, displaying the properties of the selected test variant.

3. In the **Hostname/IP address** field, enter the server to which the test will be sent.

4. In the **Port** field, enter the port to be used by AppScan to reach the server.

The default port is **80**; unless SSL is selected, in which case **443** is the default port.

5. If necessary, you can edit the **Request** itself.

6. In the **Options** list, select or clear the following options:

Option	Click to
SSL	Send request in SSL.
Login before sending request	Send login request to the application before sending the manual test.
Auto content length	Automatically update the Content-Length HTTP header in the request, to the value that equals the request content as you edit the request. If this option is selected, the user cannot edit the content length value. If the request header does not have the Content-Length parameter, this option has no effect.

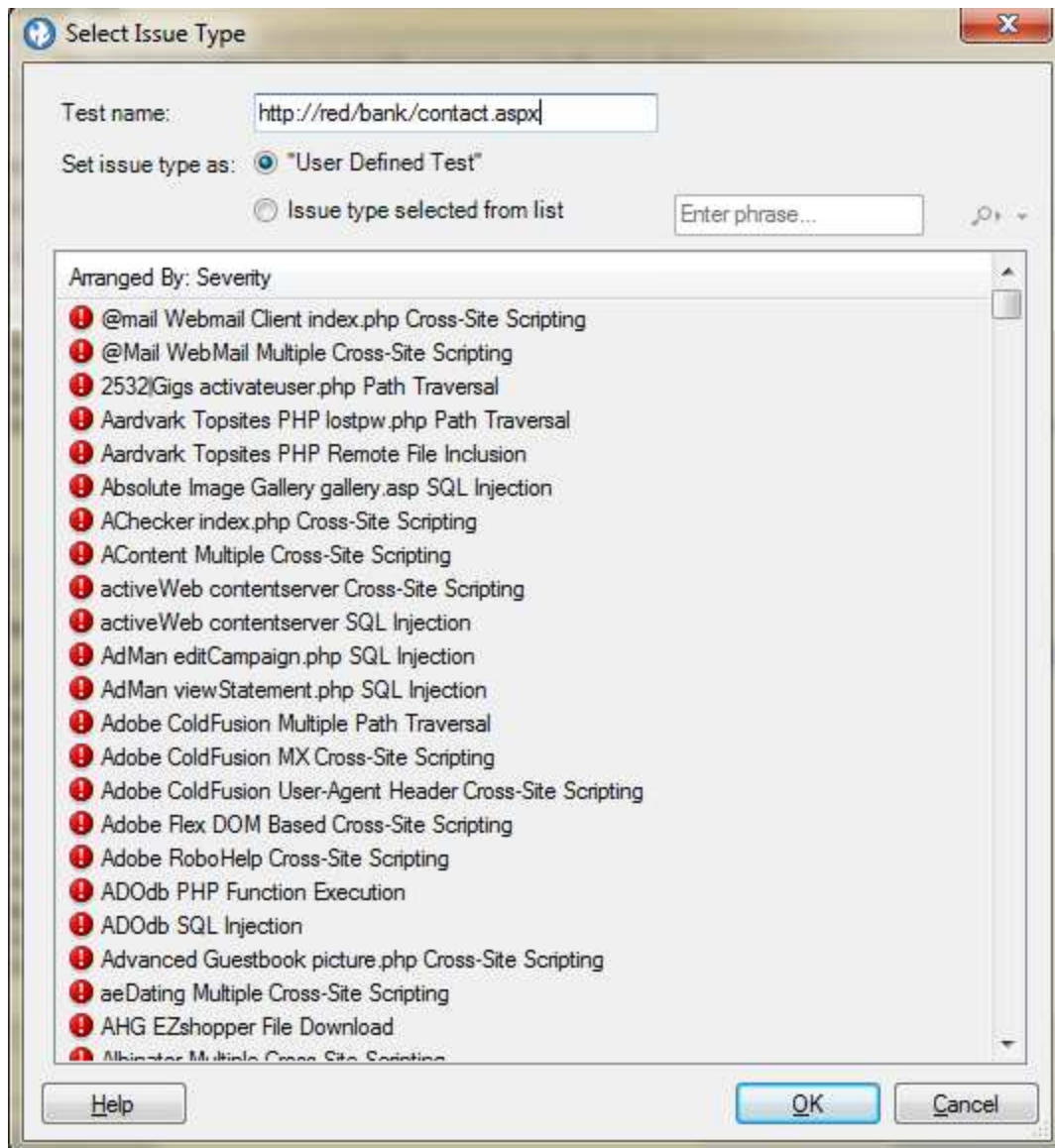
7. Click **Send**.

The request is sent, and the response is displayed in the **Response** text area (lower pane).

8. To view the response in the embedded browser, click **Show in Browser**.

9. To add this manual test to the current scan, click **Save**.

The **Select Issue Type** dialog box appears, with the **User-Defined Test** radio button selected by default.



10. In the **Test name** field, you can leave the default name (the URL) or define a new name.
11. To save the test under an existing test type (rather than the default User-Defined test type), select the second radio button, and then click on a test type in the list.

Note: To locate an issue in the list using the Search feature, type part of the issue name into the Search field. Click on the magnifying glass icon to move from one search result to the next.

12. Click **OK**.

The dialog boxes closes. The new test is added to the results, and will be included when you continue the current scan (**Scan > Continue > Scan/Test**).

Note: If you rescan the test is not included.

Non-vulnerable variants

During a scan, AppScan sends many thousands of test variants to the site it is testing. The responses to many of these indicate that they do not pose a security threat of any kind, and by default AppScan discards all these "non-vulnerable" results.

- If necessary, you can configure AppScan to save all non-vulnerable variants
- You can also change the status of individual results to "non-vulnerable"

"Saving all non-vulnerable variants"

"Defining variants as non-vulnerable"

"Non-Vulnerable Variants List"

"Deleting variants" on page 204

Saving all non-vulnerable variants

About this task

You can configure AppScan to save all "non-vulnerable" test variants, if you want to review them.

CAUTION:

Saving non-vulnerable test variant information may slow down AppScan performance and significantly increase the disk space required.

Procedure

In **Scan Configuration > Test Options** select the **Save Non-Vulnerable Test Variant Information** check box.

Defining variants as non-vulnerable

About this task

When you define a test variant as *non-vulnerable* it will not be displayed in the scan results or included in reports, but you maintain the option of viewing its details (via the "Non-Vulnerable Variants List") and reinstating it later if necessary.

Procedure

Do one of the following:

- On the Result List, right-click and select **Set as Non-vulnerable**
- Select the variable in the Result List, and on the Request/Response toolbar, click **Set as Non-vulnerable**.

The variant is removed from the scan result display, and will not be included in reports.

Non-Vulnerable Variants List

The Non-Vulnerable Variants List lets you view details of non-vulnerable variants that have been saved with the scan results, and reinstate them if necessary.

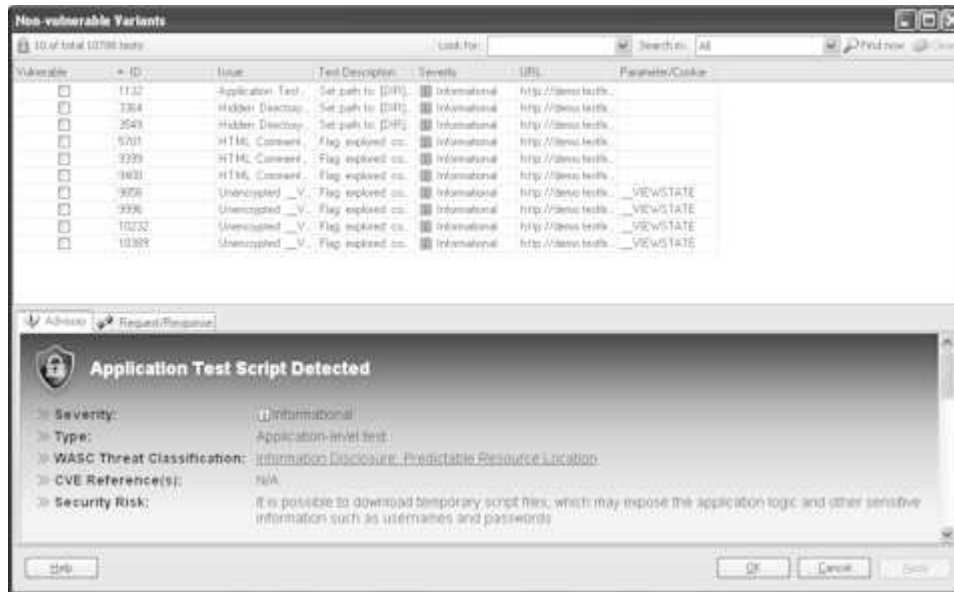
There are two ways that variants are added to this list:

- AppScan has been configured to save non-vulnerable test variant information (see "Test Options view" on page 100)

- You manually changed a result that AppScan categorized as "vulnerable" to "non-vulnerable" (as described in the previous section)

Viewing non-vulnerable variants Procedure

On the **View** menu, click **Non-vulnerable Variants**.



Reinstating non-vulnerable variables as vulnerable Procedure

1. Select the check box for one or more variants in the non-vulnerable Variants list, that you want to reinstate as vulnerable.
2. At the bottom of the dialog box, click **Apply**.
3. Click **OK** to confirm.

Deleting variants About this task

When you delete variants (rather than define them as non-vulnerable) they are deleted from the scan results altogether and cannot be reinstated later. (To access them again you would need to run a new scan.)

Procedure

Do one of the following:

- On the Result List, right-click and select **Delete**.
- Select the variant in the Result List, and on the Request/Response toolbar, click **Delete Variant**.

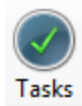
The variant is removed from the scan results, and will not be included in reports.

Chapter 9. Results: Remediation Tasks

AppScan offers three ways of viewing and working with scan results: Security Issues, Remediation Tasks, and Application Data. This section deals with Remediation Tasks view.

Remediation Tasks view provides solutions designed to address issues that were found in the scan. One remediation task usually addresses a number of security issues.

On the **View Selector**, click:



Remediation Tasks: Application Tree

The **Application Tree** displays the folders and files of the scanned application. Each node in the tree has a counter, which shows how many remediation tasks are held in the node. The count for each node will be the same as, or less than, the count for the **Issues** view, as multiple issues may be addressed by one remediation task.

The **Application Tree** displays remediation tasks in the following levels:

- **Task name**
- **URL**
- **Parameter or Cookie**

A single task designed for an issue found on several URLs is listed once, with the URLs under it.

Select a node in the **Application Tree** to filter the **Result List**, which will display only the results of the selected node.

Remediation Tasks: Result List

The **Result List** displays the Remediation tasks relevant to the node selected in the **Application Tree**. If you selected the **My Application** node, the **Result List** shows all remediation tasks that are relevant to your application.

Remediation tasks are consolidated by the type of remediation that can be done to address issues. Each remediation item has an icon, indicating priority for the task to be done; and a count, indicating how many files, parameters, or cookies will be affected by this remediation.

- ⊞ Ensure that accessed files reside in the virtual path and have certain extensions; remove special characters from user input (1)
- ⊞ Filter out hazardous characters from user input (3)
- ⊞ Disable HTTP TRACE support in your web server (1)
- ⊞ Encrypt all login requests (1)
- ⊞ Modify the server configuration to deny directory listing, and install the latest security patches available (6)
- ⊞ Cancel output of debugging error messages and exceptions and contact vendor for a security patch (1)
- ⊞ Contact the vendor of your product to see if a patch or a fix has been made available recently (1)
- ⊞ Disable WebDAV, or disallow unneeded HTTP methods (5)
- ⊞ Encrypt sensitive information sent to the server (1)
- ⊞ Install patch Q296576 (1)
- ⊞ Remove sensitive information from HTML comments (2)
- ⊞ Verify that parameter values are in their expected ranges and types. Do not output debugging error messages and exceptions (2)

Each task may hold URLs, which may hold files, parameters, and cookies. You can change the way the remediations are sorted, and you can manipulate their priority value.

Searching Remediation Tasks in Result List

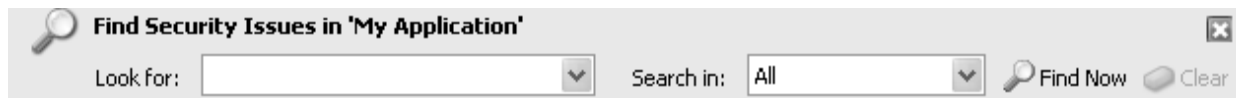
About this task

You can filter the **Result List** for types of remediation tasks, or you can search for a specific remediation.

Procedure

1. On the **Edit** menu, click **Find** (or press **Ctrl + F**).

The **Find** bar appears in the main window, above the **Result List**.



2. On the **Application Tree**, select a node.
 - If you select the **My Application** node, Find searches through all the results.
 - If you select a node within the tree, Find searches through the selected node and its sub-nodes.
3. In the **Look for** text box of the **Find** bar, enter a string or partial string.
4. In the **Search in** list, select the part of the remediation that you want to search in:

Search in	To find string within
Remediation	remediation name as it appears in the Result List
URL	pathname of URLs for which remediation results are relevant
Details	remediation task details
All	all of the above options

5. Click **Find Now** or press **Enter**.

Results appear in the **Result List**, overwriting the list that was previously displayed. If you enter another **Find** string and click **Find Now** again, the new search is done on the selected node in the **Application Tree**, not on the displayed results of the previous search.

Example

To find	Look for	Search in
Remediation tasks that concern virtual directories:	"virtual direc"	Remediation
Remediation tasks for login pages:	"login"	URL

Sorting Remediation Tasks

About this task

You can re-arrange remediation tasks in the **Result List**. By default, tasks are sorted by priority.

Procedure

1. Click the **Arranged By** column header.
A menu appears.
2. Click a command on the menu to sort the remediations:
 - **Priority** - task names are listed by priority order: High Priority first, down to Low.
 - **Count** - number of affected URLs, parameters, or cookies. Tasks that solve the most issues are listed first.
 - **Name** - alphabetical order by task name.

The remediation tasks are re-sorted in the **Result List**.

To reverse the order of the sort (ascending or descending), click the second column of the **Result List**. Click the second column again to revert to the previous order.

Manipulating priority levels

About this task

The icons of the remediation tasks indicate their priority levels. The following table explains the icons.

Icon	Indicates
	High priority task
	Medium priority task
	Low priority task

Priority is assigned to remediation tasks based on the issue severity. If the severity is High, the priority is High; Medium severity is Medium priority; and Low and Informational severity is Low priority.

You can view default priority settings as a starting point. You can change the priority assigned to remediation tasks. The new priority setting affects all remediations with the same Task name.

Procedure

On the **Result List**, right-click a remediation task and select **Priority > High/Medium/Low**. The remediation task icon changes to match the new priority.

Deleting Remediation Tasks from the Result List

About this task

You can delete a task from the **Result List**, which will delete the selected node and all the objects contained within it.

Procedure

1. On the **Result List**, right-click a node.
2. On the menu that appears, click **Delete**.

A message appears, asking you to confirm the delete and reminding you that the data of the tests will be permanently deleted.

3. Click **Yes**.

The remediation task is deleted, but this does not affect the issues; issues will still be available from the Issues view.

Remediation Tasks: Detail Pane

The **Detail Pane** for the Remediations view has one tab. It shows the remediation task currently selected in the **Result List**.

The information in the **Detail Pane** includes: Task Name, Issues (list of scan results addressed by this task), and Details (one or more possible solutions).

Chapter 10. Reports

This section describes how to generate reports from the scan results.

Related concepts:

“Exporting scan results” on page 170

Report overview

After AppScan has assessed your site's vulnerability, you can generate customized reports configured for the various personnel in your organization; from developers, internal auditors, and penetration testers, to managers and executives.

There are five basic types of report, as described below. The Security Report includes many options that can be included or excluded depending on who the report is intended for.

You can open and view the reports from within AppScan, and you can save a report as a file to be opened with a third-party application, such as Acrobat Reader.

Icon	Name	Short Description
	“Security reports” on page 211	Report of security issues found during the scan. Security information may be very extensive, and can be filtered depending on your requirements. Six standard templates are included, but each can easily be tailored to include or exclude categories of information, as necessary.
	“Industry Standard reports” on page 214	Report of the compliance (or non-compliance) of your application with a selected industry committee, or your own custom standards checklist.
	“Regulatory Compliance reports” on page 216	Report of the compliance (or non-compliance) of your application with a large choice of regulations or legal standards, or with your own custom Regulatory Compliance template).
	“Delta Analysis reports” on page 226	The Delta Analysis report compares two sets of scan results and shows the difference in URLs and/or security issues discovered.
	“Template-based reports” on page 228	Custom report containing user-defined data and user-defined document formatting, in Microsoft Word DOC and DOCX formats.

Configuring report layout

The layout tab of the Create Report dialog box enables you to customize the appearance of your reports. (This feature is optional, as you can simply generate reports using the default layouts.)

Procedure

1. In the **Create Report** dialog box, click the **Layout** tab.
2. Select the layout options that you want and enter appropriate values:

Layout Options	Description
Include Cover Page	Add a cover page to your report. If selected, cover page options are enabled.

Layout Options	Description
Company Logo	Include your company logo at the top left of the cover page. (Click  icon in the Company Logo area, and browse to the logo file on your computer). The AppScan logo is the default logo.
Additional Logo	Include an additional logo at the top right of the cover page. (Click  in the Company Logo area, and browse to the logo file on your computer).
Report Type	Include the Report Type (editable text) on the lower half of the cover page.
Report Title	Include the default title, or a title that you type in, as the main title in the center of the cover page.
Description	Include the default description, or a description that you type in, as the description on the cover page.
Report Date	Include the date in the footer to each inside page of the report.
Header/Footer	Add a header and/or footer to each inside page. Type the text you want to appear.
Table of Contents	Include a table of contents in the report.
Save as Default Layout	Save the layout settings and texts for future use.

Viewing and saving reports

Reports can be generated and viewed in AppScan's Report Viewer, and saved in various formats.

Procedure

1. Configure the report type, template, and filters as needed.
2. To see a preview in the Report Viewer, click **Preview**.
3. To save the report, click **Save Report**, type in a name for the report, and select a Format: PDF (Adobe Acrobat Reader), HTML (web browser), RTF (Microsoft Word), or TXT (text editor).

Note: If you have previously saved a report in one format and now want to save it in another, you must save it with a different filename. For example: If you have previously saved a report named Report458.pdf and now want to save it in RTF format, you cannot save it as Report458.rtf but could save it as Report_458.rtf

Creating partial reports

You can create a Security Report or a Template-Based Report for a subset of the scan results by right-clicking on the URL or folder for which you want to create the report.

Procedure

1. In the Application Tree, right-click on the URL or folder for which you want to create the report, then select:
 - **Report for this node > Security**
 - **Report for this node > Template-based**

The Create Report dialog box opens with report options for the selected node.
2. Continue as for a regular report.
 - "Security reports" on page 211
 - "Template-based reports" on page 228

Earlier versions of report templates

Earlier versions of some Industry Standard and Regulatory Compliance templates are saved in the "Old Versions" folder.

About this task

The Reports dialog box lists the latest versions of the report templates. However, in the case of some Industry Standard and Regulatory Compliance reports, earlier versions are saved in a special folder. The folder is located in [AppScan Standard installation folder]\Regulations\Old Versions

Procedure

1. In the Report dialog box, select **Industry Standard** or **Regulatory Compliance**.
2. In the Report Type tab, select **User Defined**.
3. Click **Browse...**, and go to [AppScan Standard installation folder]\Regulations\Old Versions
For Japanese, the location is [AppScan Standard installation folder]\ja-JP\Regulations\Old Versions
4. Select the required file, and continue as for regular report creation.

Security reports

The Security report provides information about security issues discovered, and you can choose from a variety of templates depending on the type of content you need.

About this task



You can create a security report that covers the whole scan, or for a particular URL or folder in the application tree.

Each report template is a set of content topics that are relevant to different audiences within your organization. The topics contain scan results from each of the views (Security Issues, Remediation Tasks, Application Data), formatted for easy printing, readability, and rapid comprehension of what the results mean, why they are relevant, and how to fix them.

Security Report Options

The table following summarizes the options in the Security Reports dialog box.

Option	Description
Template	Select one of several templates for the report, or define your own, by selecting/clearing check boxes in the right hand pane, as described in the table following. • Default: A medium-level report containing a high level summary and Issue Information, without details of variants. • Summary: A high level summary with highlights of security risks found in your web application, and statistics of scan results, formatted in tables and charts. • Detailed: A thorough report that includes the Summary, as well as security issues, advisories and fix recommendations, remediation tasks and application data. • Remediation Tasks: Remediation tasks: actions designed to address the issues discovered in the scan. • Developer: Security Issues, variants, advisories and fix recommendations, without the Summary or Remediation Tasks sections. • QA: Security Issues, advisories and fix recommendations, and application data, without detailed variant information, or the Summary or Remediation Tasks sections. • Site Inventory: Application data only. • Custom Template: This option lets you create a custom Security Report template by using the check boxes to define the report you want, and then clicking Create as Template. Once saved, the template can be used to generate reports both from the user interface and the command line interface. – Save as Template: Save the current Security Report configuration as a custom template. – Delete Template: Delete the current custom template.
Min. Severity	Select the lowest level of severity for issues to be included in the report.
Test Type	Select which types of test results to include in the report: All , Application , Infrastructure , or Third-Party Web Component tests.
Sort by	Select whether to sort issues by type or URL.
Limit number of variants per issue	You can reduce the length of the report by limiting the number of variants listed per issue, if this level of detail is unlikely to be useful to the recipient of the report.
Add page break after each issue	This setting applies only to PDF output. It can make the report clearer to read.
View when done	If you select this check box, the report will be opened in an appropriate viewer after it is generated. Note: This will only work if you have a program installed that can open the generated report.

After selecting any template as a basis, you can customize the individual report structure by selecting/deselecting the fields of information to be included. If you do this the template name changes to "Custom".

Security Report Sections

The table following summarizes the standard content of the various Security Reports. In all cases the actual content can be changed as required by selecting/clearing check boxes in the Report Content pane.

Note: A full detailed report could be hundreds of pages long, so be sure to include only the sections that are relevant to your audience.

Report Section	Description
Introduction	A short section that provides some general information about the scan, including such details as overall number of issues found (High, Medium, Low and Informational), and login settings. This section is included in all reports.

Report Section	Description
Summary	A series of tables summarizing the following information about the scan, or the part of the scan included in the report: • Issue types (includes number of issues found for each type, and their severity) • Vulnerable URLs (includes number and type of issues per URL) • Fix recommendations • Security risks • Causes • WASC threat classification
Security Issues	Issues found in your application: • Basic: If you select neither of the following two check boxes, basic information only is included • Additional: Includes more detailed information, including screen captures, similar to the Issue Information tab content • Variants: Includes specific variant information: – Request/Response – Difference: The difference between the original request and the test request, as shown in the Detail pane > Request/Response tab
Advisories and Fix Recommendations	Technical explanations of the issues found and recommendations for fixing them. Note: To include fix recommendations specific to .NET, Java EE and PHP environments, go to Tools > Options > Preferences and select the required options.
Remediation Tasks	Suggested tasks for improving site security based on the issues found. One task may solve more than one issues.
Application Data	List of data that AppScan found in your web application: Application URLs, Script Parameters, Broken Links, Comments, JavaScripts, Cookies, and Filtered URLs.

Procedure

1. Select the scan content on which to base the report:
 - To create a report for the whole scan, click **Tools > Report > Security Report**
 - To create a report for a particular URL or folder that was included in the scan, right-click on the node in the application tree, and then select **Report for this node > Security**
2. Select the relevant template, or define your own report content by selecting/clearing check boxes in the right pane.
3. Select the options required.
4. To save the configuration for future use, click **Save as Template** and give the template a unique name.
5. To customize the layout of the report, click the Layout tab. See “Configuring report layout” on page 209 for details.
6. Select the output format required: PDF, HTML, TXT, RTF, or XML.
7. Click **Save Report**.

Limiting the size of Security reports

Tips for reducing a large Security Report to a more manageable size.

About this task

Security reports can be very large. If, when you attempt to generate a Security Report, you receive a warning message that the file will be hundreds of pages long, or if the report creation process times out, you can try the following tips to reduce report size while including the information that is important to you.

Procedure

1. One way of reducing report size is to limit the report to a part of the application. You can do this by selecting the relevant node in the application tree, right-clicking it, and selecting **Report for this node > Security**. This will create a report for all parts of the application under the selected node.
2. If you see that your scan results contain thousands of issues, consider generating a **Summary** only, or the **Default Report** with the **Additional Issue Information** check box cleared.
3. By default *all* Test Types are included in reports (Min. Severity = Informational). Try increasing the **Min. Severity** setting so that only High, or High and Medium, severity issues are included.
4. Verify that the **Max. Variants** setting is 1, so that no more than a single variant is included for each issue.
5. Be aware that including **Variants > Request/Response**, and **Advisories and Fix Recommendations** can add a lot of volume to reports. Select these options only if needed.

Industry Standard and Compliance reports

Industry Standards reports let you know if your application complies with standards of a selected industry committee; Regulatory Compliance reports let you know if your application complies with specific regulations or legal standards.

Industry Standard reports

The Industry Standards report lets you know if your application complies with standards of a selected industry committee.

About this task



The list of templates to choose from is updated by IBM as new standards are created for different industries and your AppScan is updated automatically on a regular basis.

If you do not find the industry standard that you need in the list, you can create your own Industry Standard report template (see “User-defined reports” on page 219).

The Industry Standard reports include the following sections:

Section Title	Information Presented
Description	Explanation of the standard.
Compliance Summary	List and count of non-compliance issues. Note that a single issue may indicate non-compliance of more than one section, and therefore the number of issues per section, for all the sections, may be more than the number of unique issues.
Unique Compliance Issues	List of non-compliant URLs, relevant parameters or cookies, and test names. Each issue appears once only.
Compliance Issues by Section	Detailed explanations of how your application is non-compliant, with remediation to address issues.

The figure below shows a sample of part of an Industry Standard Report.

Compliance Issues by Section

1) Unvalidated input (A1)

3 Issues

Cross-Site Scripting

Security Risks

- It is possible to steal customer session and cookies, which may be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user

Causes:

- Sanitation of hazardous characters was not performed correctly on user input

Remediation:

Filter out hazardous characters from user input

Issues:

Issue ID	URL	Parameter/Cookie
1	http://bern/bank/search.aspx	searchterms

Procedure

1. Click **Tools > Report > Industry Standard**
2. Do one of the following:
 - Select one of the listed **Industry Standard Report Templates**.
 - Select the **User Defined** radio button and enter or browse to a custom Industry Standard template file (*.asreg). See “User-defined reports” on page 219 for details.
3. If you want to control the appearance of the report, open the **Layout** tab (see “Configuring report layout” on page 209).
4. Click **Preview** to generate and view the report within AppScan; or click **Save Report** to generate and save it to a file.

Supported versions

A list of supported Industry Standards and versions.

The following Industry Standard reports can be generated.

Industry Standard	Version
International Standard - ISO 27001	January 2013
International Standard - ISO 27002	January 2013
NERC CIPC Electricity Sector Security Guidelines	September 2013
NIST Special Publication 800-53	Revision 4
OWASP Top 10	2013, 2017
SANS/CWE Top 25 Most Dangerous Programming Errors	1.03

Industry Standard	Version
WASC Threat Classification	2.0

Regulatory Compliance reports

The Regulatory Compliance report lets you know if your application complies with regulations or legal standards.

About this task



There is a long list of templates from various countries to choose from, each template providing a compliance report for a different regulation.

If you do not find the regulation that you need in the list, you can create your own Regulatory Compliance report template (see “User-defined reports” on page 219 for details).

The Regulatory Compliance reports include the following sections:

Section Title	Information Presented
Description	Explanation of the regulation.
Compliance Summary	List and count of non-compliance issues. Note that a single issue may indicate non-compliance of more than one section, and therefore the number of issues per section, for all the sections, may be more than the number of unique issues.
Unique Compliance Issues	List of non-compliant URLs, relevant parameters or cookies, and test names. Each issue appears once only.
Compliance Issues by Section	Detailed explanations of how your application is non-compliant, with remediation to address issues.

The figure below shows a sample from a Regulatory Compliance Report.

Compliance Summary

34 unique issues across 44 sections of the regulation:

Section	No. of Issues
1. Implement Internet Protocol (IP) masquerading to prevent your internal address from being translated and revealed on the Internet. (Requirement 1.5)	3
2. Do not use vendor-supplied defaults for system passwords and other security parameters. (Requirement 2)	19
3. Always change the vendor-supplied defaults before you install a system on the network. (Requirement 2.1)	14
4. Develop configuration standards for all system components. Make sure these standards address all known security vulnerabilities and industry best practices. (Requirement 2.2)	15
5. Disable all unnecessary and insecure services and protocols. (Requirement 2.2.2)	14
6. Configure system security parameters to prevent misuse. (Requirement 2.2.3)	14
7. Remove all unnecessary functionality, such as scripts, drivers, features, subsystems, file systems. (Requirement 2.2.4)	15
8. Encrypt all non-console administrative access. Use technologies such as SSH, VPN, or	4

Procedure

1. Click **Tools > Report > Regulatory Compliance**
2. Do one of the following:
 - Select one of the listed **Regulatory Compliance Report Templates**:
 - Select the **User Defined** radio button and enter or browse to a custom Regulatory Compliance template file (*.asreg). See “User-defined reports” on page 219 for details.
3. If you want to control the appearance of the report, open the **Layout** tab (see “Configuring report layout” on page 209).
4. Click **Preview** to generate and view the report within AppScan; or click **Save Report** to generate and save it to a file.

Supported versions

A list of supported Regulations and versions.

The following Regulatory Compliance reports can be generated.

Regulation	Supported Version
[AUS] APRA PPG 234 - Management of Security Risk in Information and Information technology	January 2012
[CANADA] PIPED Act	January 2014

Regulation	Supported Version
[CANADA] Freedom of Information and Protection of Privacy Act (FIPPA)	September 2012
[CANADA] Management of Information Security Technology (MITS)	September 2012
[EU] European Directive 1995/46/EC	January 2012
[EU] European Directive 2002/58/EC	March 2012
[EU] Regulation 2016/679 of the European Parliament and of the Council (GDPR)	April 2016
[JAPAN] Japan's Personal Information Protection Act (JPIPA)	January 2012
[UK] Data Protection Act	December 2014
[US] California Assembly Bill No. 1950 and Senate Bill 1386	March 2012
[US] Children Online Privacy Protection Act (COPPA)	December 2014
[US] DCID 6/3 Availability Basic	January 2011
[US] DCID 6/3 Availability High	January 2011
[US] DCID 6/3 Availability Medium	January 2011
[US] DCID 6/3 Confidentiality Reqs Protection Level 1	January 2011
[US] DCID 6/3 Confidentiality Reqs Protection Level 2	January 2011
[US] DCID 6/3 Confidentiality Reqs Protection Level 3	January 2011
[US] DCID 6/3 Confidentiality Reqs Protection Level 4	January 2011
[US] DCID 6/3 Confidentiality Reqs Protection Level 5	January 2011
[US] DCID 6/3 Integrity Basic	January 2011
[US] DCID 6/3 Integrity High	January 2011
[US] DCID 6/3 Integrity Medium	January 2011
[US] DCID 6/3 Securing Advanced Technology IS	January 2011
[US] Electronic Funds and Transfer Act (EFTA)	December 2013
[US] Federal Financial Institutions Examination Council (FFIEC), Information Security handbook	February 2013
[US] Federal Information Security Management Act (FISMA)	September 2014
[US] The Federal Risk and Authorization Management Program (FedRAMP)	
[US] Financial Services (GLBA)	January 2013
[US] Healthcare Services (HIPAA)	December 2014
[US] NERC Cyber Security Standards	September 2014
[US] Privacy Act of 1974	January 2011
[US] Safe Harbor	November 2012
[US] Sarbanes-Oxley Act (SOX)	January 2013
[US] Title 21 Code of Federal Regulations	November 2011
[US] Family Educational Rights and Privacy Act (FERPA)	January 2013
[US] DISA's Application Security and Development STIG	V3 Release 9
[US] DoD Instruction 8500.1 - Cybersecurity	September 2014

Regulation	Supported Version
[US] DoD Instruction 8550.01 - Internet Services and Internet Based Capabilities	September 2014
[US] Massachusetts 201 CMR 17.00	January 2011
Basel II	October 2012
Payment Application Data Security Standard	3.0
The Payment Card Industry Data Security Standard (PCI DSS)	3.2

User-defined reports

You can create user-defined templates for either the Industry Standards or Regulatory Compliance reports.

About this task

AppScan report templates have the **.asreg** file extension. Supplied templates are stored in the \Regulations folder of your AppScan installation directory; templates you create should be stored in your AppScan User Files folder.

You can create a new template from scratch and save it with the **.asreg** extension, or copy an existing file and make changes as required. (The procedure below describes creating a template based on an existing one.)

Procedure

1. Open the [AppScan Standard installation folder]\AppScan\Regulations folder and copy an existing **.asreg** file.
2. Paste the file into your AppScan User Files folder, and give it a new name.

Note: By default the AppScan User Files folder is \My Documents\AppScan unless you specified a different location in **Tools > Options > Preferences tab > File Locations > User Files folder**.

3. The root tag is Regulation, with the attribute of format_version:


```
<Regulation format_version="2.0">
```
4. The next tag should be the title of your template:


```
<Title>Our Organization's web Application Requirement Compliance Report
</Title>
```
5. Enter a description of the regulation or standard using the Description tag:


```
<Description>
  <Subtitle>Sub Section</Subtitle>
  <p>This regulation addresses ...</p>
  <p>It is important because...</p>
  <Subtitle>Sub Section 2</Subtitle>
  <p>This section of the regulation addresses ...</p>
</Description>
```
6. By default, there is a <Disclaimer> tag, which ensures that you do not take legal responsibility for the contents of the report.
7. Create one or more requirement sections (using the <Section> tag) for your regulation template and define which AppScan issues are relevant for each section, by using the <Cause>, <Risk>, <ThreatClass>, and <CWE> tags.
 - Use the name attribute in the Section tag to define the section title for the report.
 - Between the opening and closing Section tags, add one or more of the following:

- <Cause> from the “List of causes.” A cause describes an incomplete or incorrect configuration, missing validation, or similar status.
- <Risk> from the “List of risks” on page 221. Each risk is a "worst case scenario".
- <ThreatClass> from the “List of threat classes” on page 223. A threat class is a category of tests.
- <CWE> by number.

Example:

```
<Section name="My Application login must be secured">
  <Cause>inputLengthNotChecked</Cause>
  <Risk>denialOfService</Risk>
  <Risk>siteDefacement</Risk>
  <CWE>79</CWE>
</Section>
```

8. Close the file with the </Regulation> closing tag.

List of causes

Cause	Description
hazardousCharactersNotSanitized	Sanitation of hazardous characters was not performed correctly on user input.
formatStringsVulnerability	User input is used directly as a formatting string input for C/C++'s sprintf and similar functions.
hiddenParameterUsed	Parameter values were 'hardcoded' in the HTML as a parameter of type 'hidden'.
boundsCheckingOnParamValues	Proper bounds checking were not performed on incoming parameter values.
incorrectDataType	No validation was done to ensure user input matches expected data type.
inputLengthNotChecked	User input length is not limited, thereby enabling buffer overflows.
errorMessagesReturned	Exceptions and error messages, which may contain sensitive debugging information, are presented to users.
debugInfoInHtmlSource	Debugging information was left by the programmer in web pages.
backDoorLeftBehind	A backdoor or a debugging option was left behind by programmers.
clientSideValidation	User input validation is done at the client-side and may be bypassed.
usOfClientSideLogic	The web application uses client-side logic to create web pages.
cookiesCreatedAtClientSide	Cookies are created at the client-side.
javaScriptPassWordMechanism	The web application uses a client-side password authentication.
sqlBuiltByJavaScript	The web application uses client-side logic to create SQL queries.
dotDotNotSanitized	User input is not checked for the '..' string.
weakTokenUsed	A weak token algorithm is used by the web application.
missingPatchesForThirdPartyProds	Latest patches or hotfixes for 3rd party products were not installed.
tempFilesLeftBehind	Temporary files were left in production environment.
improperFileDirPermissions	Improper permissions/ACLs were set to file/directory.
nimdaWormBackdoor	The Nimda worm was found on the system.
sampleScriptsFound	Default sample scripts or directories were installed on the website.
insecureThirdPartySoftware	A vulnerable third party software, which does not have a known patch, is installed on the website.
directoryBrowsingEnabled	Directory browsing is enabled.
managementConsoleAccess	Web management console is accessible from the web.
insecureWebServerConfiguration	The web server or application server is configured in an insecure way.
frontPageServerUnsecureInstall	FrontPage server extensions were installed with improper security settings.

Cause	Description
insecureWebAppConfiguration	Insecure web application programming or configuration.
vulnSOAPserializer	The SOAP serializer used by your web services server does not validate SOAP input properly.
sensitiveDataNotSSL	Sensitive input fields such as usernames, passwords, and credit card numbers are passed unencrypted.
nonSecureCookiesSentOverSSL	The web application sends non-secure cookies over SSL.
sessionCookieNotRAM	The web application stores sensitive session information in a permanent cookie (on disk).
redirectionFromWithinSite	The web application performs a redirection to an external site.
remoteFileInclusion	The web application allows remote file inclusion.
GETParamOverSSL	Query parameters were passed over SSL, and may contain sensitive information.
SensitiveCache	Sensitive information might have been cached by your browser.
InsufficientAuthentication	Insufficient authentication method was used by the application.
useOfGlobalFlashParamsInPDFNs	Global flash parameters used in potentially dangerous native functions.
causeNotAvailable	n/a
vulnActiveX	The ActiveX control used is categorized as vulnerable. The scanned Web site might have been hacked in order to serve malware.
compromisedDigiNotarSSLCert	The SSL certificate in use has been flagged as compromised due to DigiNotar's security breach.
paramValManipAllowed	Parameter value manipulation was permitted by the application logic.

List of risks

Risk Name	Description
tempScriptDownload	It is possible to download temporary script files, which may expose the application logic and other sensitive information such as usernames and passwords.
sourceCodeDisclosure	It is possible to retrieve the source code of server-side scripts, which may expose the application logic and other sensitive information such as usernames and passwords.
pathDisclosure	It is possible to retrieve the absolute path of the web server installation, which may help an attacker to develop further attacks and to gain information about the file system structure of the web application.
directoryListing	It is possible to view and download the contents of certain web application virtual directories, which may contain restricted files.
envVariablesExposure	It is possible to expose server environment variables, which may help an attacker to develop further attacks against the web application.
anyFileDownload	It is possible to view the contents of any file (e.g. databases, user information or configuration files) on the web server (under the permission restrictions of the web server user).
userImpersonation	It is possible to steal customer session and cookies, which may be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user.
remoteCommandExecution	It is possible to execute remote commands on the web server. This usually means complete compromise of the server and its contents.
cacheFilesDownload	It is possible to view the contents of cache files, which may contain sensitive information regarding the web application.
debugErrorInformation	It is possible to gather sensitive debugging information.

Risk Name	Description
eShoplifting	It is possible to steal goods or services (eShoplifting).
denialOfService	It is possible to prevent the web application from serving other users (denial of service).
privilegeEscalation	It is possible to escalate user privileges and gain administrative permissions over the web application.
genericWorstCase	It is possible to undermine application logic.
configurationFile Downloadable	It is possible to download or view the contents of a configuration file, which may contain vital information such as usernames and passwords.
sensitiveInformation	It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations.
genericWorstCaseJavaScript	It is possible to exploit JavaScript; the extent of the risk depends on the context of the page modified at the client side.
genericWorstCaseJSCookie	It is possible to exploit JSCookie code; the extent of the risk depends on the context and role of the cookies that are created at the client side.
emailSpoofing	It is possible to send emails through your web application, using spoofed email addresses.
siteDefacement	It is possible to upload, modify or delete web pages, scripts and files on the web server.
databaseManipulations	It is possible to view, modify or delete database entries and tables (SQL Injection).
authBypass	It is possible to bypass the web application's authentication mechanism.
siteStructureRevealed	It is possible to retrieve information about the site's file system structure, which may help the attacker to map the website.
publisherInformation Revealed	It is possible to retrieve sensitive FrontPage publishing information.
dataResourceDownload	It is possible to access information stored in a sensitive data resource.
sensitiveNotOverSSL	It is possible to steal sensitive data such as credit card numbers, social security numbers etc. that are sent unencrypted.
loginNotOverSSL	It is possible to steal user login information such as usernames and password that are sent unencrypted.
unsecureCookieInSSL	It is possible to steal user and session information (cookies) that was sent during an encrypted session.
sessionCookieNotRAM	It is possible to steal session information (cookies) that was kept on disk as permanent cookies.
phishing	It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.
cachePoisoning	It is possible to deface the site content through web-cache poisoning.
attackFacilitation	It is possible for an attacker to use the web server to attack other sites, which increases his or her anonymity.
maliciousContent	n/a
clientCodeExecution	It is possible to execute arbitrary code on the Web application's clients.
siteImpersonation	Using additional attack vectors, it is possible for a malicious attacker to impersonate this site.

List of threat classes

A summary of the WASC Threat Classification, a cooperative effort to classify the weaknesses and attacks that can lead to the compromise of a website, its data, or its users.

The tables below provides short descriptions of attacks and threats. More details about the WASC Treat Classification can be found at:

<http://projects.webappsec.org/w/page/13246978/Threat%20Classification>

Attacks

Name	Short description
Abuse of Functionality	An attack technique that uses a website's own features and functionality to consume, defraud, or circumvents access controls mechanisms.
Brute Force	An automated process of trial and error used to guess a person's username, password, credit-card number or cryptographic key.
Buffer Overflow	Attacks that alter the flow of an application by overwriting parts of memory with data that exceeds the allocated size of the buffer.
Content Spoofing	An attack technique used to trick a user into believing that certain content appearing on a website is legitimate and not from an external source.
Credential/Session Prediction	A method of hijacking or impersonating a website user, by deducing or guessing the unique value that identifies a particular session or user.
Cross-site Scripting	An attack technique that forces a website to echo attacker-supplied executable code, which loads in a user's browser.
Cross-site Request Forgery	An attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim.
Denial of Service	An attack technique with the intent of preventing a website from serving normal user activity.
Fingerprinting	The most common methodology for attackers is to first footprint the target's web presence and enumerate as much information as possible. With this information, the attacker may develop an accurate attack scenario, which will effectively exploit a vulnerability in the software type/version being utilized by the target host.
Format String	Attacks that alter the flow of an application by using string formatting library features to access other memory space.
HTTP Response Smuggling	A technique to "smuggle" 2 HTTP responses from a server to a client, through an intermediary HTTP device that expects (or allows) a single response from the server.
HTTP Response Splitting	The essence of HTTP Response Splitting is the attacker's ability to send a single HTTP request that forces the web server to form an output stream, which is then interpreted by the target as two HTTP responses instead of one.
HTTP Request Smuggling	An attack technique that abuses the discrepancy in parsing of non RFC compliant HTTP requests between two HTTP devices to smuggle a request to the second device "through" the first device.
HTTP Request Splitting	HTTP Request Splitting is an attack that enables forcing the browser to send arbitrary HTTP requests, inflicting XSS and poisoning the browser's cache.
Integer Overflows	The condition that occurs when the result of an arithmetic operation, such as multiplication or addition, exceeds the maximum size of the integer type used to store it.

Name	Short description
LDAP Injection	An attack technique used to exploit websites that construct LDAP statements from user-supplied input.
Mail Command Injection	An attack technique used to exploit mail servers and webmail applications that construct IMAP/SMTP statements from user-supplied input that is not properly sanitized.
Null Byte Injection	An active exploitation technique used to bypass sanity checking filters in web infrastructure by adding URL-encoded null byte characters to the user-supplied data.
OS Commanding	An attack technique used to exploit websites by executing Operating System commands through manipulation of application input.
Path Traversal	This is a technique that forces access to files, directories, and commands that potentially reside outside the web document root directory.
Predictable Resource Location	An attack technique used to uncover hidden website content and functionality, by making educated guesses.
Remote File Inclusion	An attack technique used to exploit "dynamic file include" mechanisms in web applications to trick the application into including remote files with malicious code.
Routing Detour	A type of "Man in the Middle" attack where Intermediaries can be injected or "hijacked" to route sensitive messages to an outside location.
Session Fixation	An attack technique that forces a user's session ID to an explicit value. After a user's session ID has been fixed, the attacker will wait for them to login. Once the user does so, the attacker uses the predefined session ID value to assume their online identity.
Weak Password Recovery Validation	When a website permits an attacker to illegally obtain, change or recover another user's password.
SOAP Array Abuse	A web-service that expects an array can be the target of a XML DoS attack by forcing the SOAP server to build a huge array in the machine's memory, thus inflicting a DoS condition on the machine due to the memory pre-allocation.
SSI Injection	A server-side exploit technique that allows an attacker to send code into a web application, which will later be executed locally by the web server.
SQL Injection	An attack technique used to exploit websites that construct SQL statements from user-supplied input.
URL Redirector Abuse	URL redirectors represent common functionality employed by web sites to forward an incoming request to an alternate resource, and can be used in phishing attacks.
XPath Injection	An attack technique used to exploit websites that construct XPath queries from user-supplied input.
XML Attribute Blowup	A denial of service attack against XML parsers.
XML External Entities	This technique takes advantage of a feature of XML to build documents dynamically at the time of processing. An XML message can either provide data explicitly or by pointing to an URI where the data exists. In the attack technique, external entities may replace the entity value with malicious data, alternate referrals or may compromise the security of the data the server/XML application has access to.
XML Entity Expansion	This exploits a capability in XML DTDs that allows the creation of custom macros, called entities, that can be used throughout a document. By recursively defining a set of custom entities at the top of a document, an attacker can overwhelm parsers that attempt to completely resolve the entities by forcing them to iterate almost indefinitely on these recursive definitions.

Name	Short description
XML Injection	An attack technique used to manipulate or compromise the logic of an XML application or service. The injection of unintended XML content and/or structures into an XML message can alter the intend logic of the application. Further, XML injection can cause the insertion of malicious content into the resulting message/document.
XQuery Injection	XQuery Injection is a variant of the classic SQL injection attack against the XML XQuery Language. XQuery Injection uses improperly validated data that is passed to XQuery commands.

Weaknesses

Name	Short description
Application Misconfiguration	These attacks exploit configuration weaknesses found in web applications.
Directory Indexing	Automatic directory listing/indexing is a web server function that lists all of the files within a requested directory if the normal base file (index.html/home.html/default.htm) is not present. Unintended directory listings may be possible due to software vulnerabilities combined with a specific web request.
Improper Filesystem Permissions	A threat to the confidentiality, integrity and availability of a web application. The problem arises when incorrect filesystem permissions are set on files, folders, and symbolic links.
Improper Input Handling	One of the most common weaknesses identified across applications today. Poorly handled input is a leading cause behind critical vulnerabilities that exist in systems and applications.
Improper Output Handling	If an application has improper output handling, the output data may be consumed leading to vulnerabilities and actions never intended by the application developer.
Information Leakage	An application weakness where an application reveals sensitive data, such as technical details of the web application, environment, or user-specific data.
Insecure Indexing	A threat to the data confidentiality of the web site. Indexing web site contents via a process that has access to files which are not supposed to be publicly accessible has the potential of leaking information about the existence of such files, and about their content. In the process of indexing, such information is collected and stored by the indexing process, which can later be retrieved by a determined attacker, typically through a series of queries to the search engine.
Insufficient Anti-automation	When a website permits an attacker to automate a process that should only be performed manually.
Insufficient Authentication	Website permits an attacker to access sensitive content or functionality without having to properly authenticate.
Insufficient Authorization	When a website permits access to sensitive content or functionality that should require increased access control restrictions.
Insufficient Password Recovery	When a web site permits an attacker to illegally obtain, change or recover another user's password.
Insufficient Process Validation	When a website permits an attacker to bypass or circumvent the intended flow control of an application.
Insufficient Session Expiration	When a website permits an attacker to reuse old session credentials or session IDs for authorization.
Insufficient Transport Layer Protection	Allows communication to be exposed to untrusted third-parties.
Server Misconfiguration	Exploits configuration weaknesses found in web servers and application servers.

Delta Analysis reports

The Delta Analysis report compares two sets of scan results and shows the difference in URLs and/or security issues that were discovered in them.

About this task



You select Base and Target scans, and AppScan compares the two sets of results to let you review how the security situation has improved or deteriorated between time of the two scans.

You can compare the currently loaded scan with a saved scan, or you can compare two saved scans.

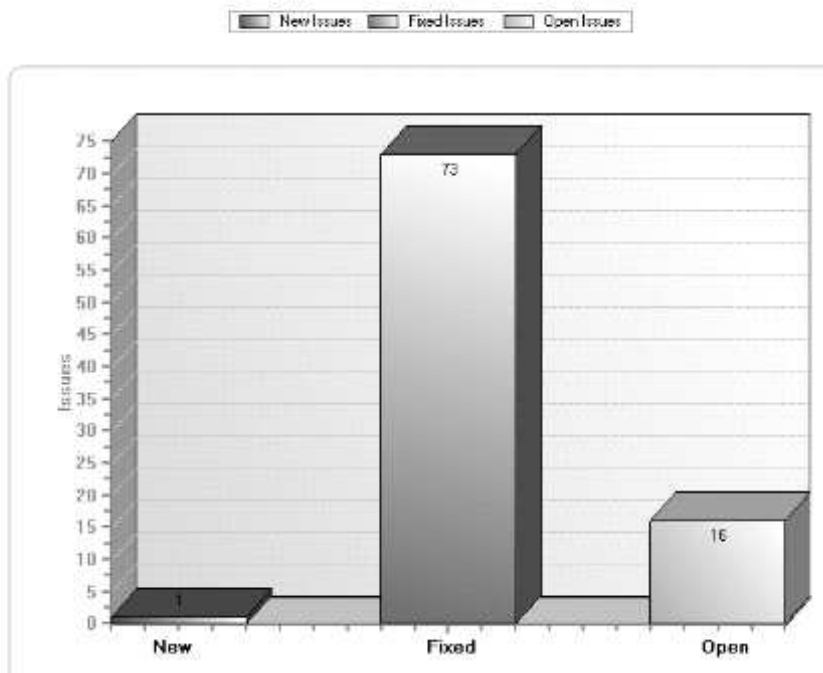
When comparing scans you will generally use the earlier scan as the Base scan. The Delta Analysis report will then indicate how the Target scan results differ from those of the Base scan.

Delta Analysis reports include the following sections:

Section Title	Information Presented
General Information	Lists the names and locations of the Base and Target scans.
Description	Description of the information included in the scan.
Issues per Host	Table showing the number of High, Medium, Low, Informational and Total issues found in each scan.
Application URLs	(If included) Bar chart showing number of New/Removed/Remaining URLs, followed by a full list of each type.
Security Issues	(If included) Bar chart of New/Fixed/Remaining Security Issues, followed by a bar chart showing the distribution of Severities in both scans (High/Medium/Low/Informational), followed by a full listing of all New/Fixed/Remaining issues.

The figure below shows a sample from a Delta Analysis Report.

Security Issues



Procedure

1. Click **Tools > Report > Delta Analysis**
2. Select the Base scan (usually the earlier of the two scans being compared). In the Base Scan area, do one of the following:
 - Click the **Current Scan** radio button
 - Click the **Saved Scan** radio button, and then browse to the location of the saved scan file.
3. Select the Target scan (usually the later of the two scans being compared). In the Target Scan area, do one of the following:
 - Click the **Current Scan** radio button
 - Click the **Saved Scan** radio button, and then browse to the location of the saved scan file.
4. In the Report Content area select one or both of the check boxes for the type of information you want the report to compare:
 - **Application URLs**, and/or
 - **Security Issues**
5. If you want to control the appearance of the report, open the **Layout** tab (see “Configuring report layout” on page 209).
6. Click **Preview** to generate and view the report within AppScan; or click **Save Report** to generate and save it to a file.

Template-based reports

The Template Based tab of the Create Report dialog box enables you to create reports in Microsoft Word DOC and DOCX formats, with exactly the data you want, and the document formatting you define.



Several sample templates are included with AppScan. These can be used:

- to generate reports (without editing)
- as a basis for creating your own custom report templates
- as a tool for understanding how to create your own report templates

The dialog box has two panes:

- The left pane lists the templates currently available. Sample templates supplied with AppScan appear in italicized text and are suffixed by the word "(sample)"; any user-created templates appear in regular (non-italic) text.
- The right pane shows an editable "Preview Image" to help you identify the template you require quickly (see table following).

Button or Link	Click to:
Import	Import a Word report template to the list in the left pane (e.g. one that you have previously customized). Imported templates are saved in: My Documents > AppScan > Report Templates > Word For more details, see "Importing a custom template" on page 233
Export	Export a template on the list to a different location (e.g. for editing).
Remove	Delete a template from the list. This does not delete the template, but moves it to the folder: My Documents > AppScan > Report Templates > Word > Deleted
Edit in MS Word	Open the selected template in MS Word for editing content and layout. For more details, see "Custom template tutorial" on page 229
View Field Reference	Open an HTML list of field references that can be used in creating Word Report Templates.
Select Report Preview Image	Browse-to and select an image that will represent the template in the right pane (when the template is selected in the left pane).
Revert to Default Image	Reinstate the default image (a blank page) for the selected template.
Preview	Generate and display a report on the current scan results, using the template selected in the template pane. MS Word will open and you can edit the content or layout if required. Note that AppScan creates only a temporary file; if you want to save it you must do so in Word.
Save Report	Generate and save a report on the current scan results (without opening MS Word to display the report contents), using the template selected in the template pane. AppScan asks you to specify a name and location for saving the file. Once the file has been saved it opens so you can view it.

See also:

"Creating a custom report template" on page 229

“Custom template tutorial”

“Importing a custom template” on page 233

“Creating partial reports” on page 210

Creating a custom report template

You can use field codes for defining your own custom reports to generate in MS Word.

Introduction to "merge fields"

AppScan Merge fields fall into two categories:

- **Repeaters (or loops):** These consist of a Start and an End tag. They do not produce any data themselves, but they "loop through" any plain fields placed between the Start and End tags, and draw the relevant data for these groups.

Example: <<AS:IssueTypeRepeaterStart>> <<AS:IssueTypeRepeaterEnd>>

This repeater loops through all IssueTypes in the scan results.

- **Plain fields:** These consist of a single field, and they produce the actual data. (Plain fields do not necessarily have to be placed within a repeater, but they cannot have other plain fields as "children".)

Example: <<AS:IssueTypeName>>

This field represents the names of all issues found in the scan.

When you insert a plain field between the Start and End tags of an appropriate loop, a report can be created. Using the two examples above, we can create a report of all the IssueTypes found by the scan:

<<AS:IssueTypeRepeaterStart>> <<AS:IssueTypeName>><<AS:IssueTypeRepeaterEnd>>

To better understand how this works in practice, see the “Custom template tutorial” following.

Note: A complete list of valid merge fields can be opened by clicking **Tools > Report > Template Based > View Field Reference**.

Custom template tutorial

This section describes how to make a simple custom template.

About this task

In this section we will create a simple custom template for generating a table of remediation tasks for all issues discovered during the scan. The template output will have the structure shown below, though obviously the actual data will depend on the scan you use when generating the report.

Index	Name	Count	Priority
1/2	RemediationTaskA	4	High
2/2	RemediationTaskB	2	High
1/3	RemediationTaskC	5	Medium
2/3	RemediationTaskD	2	Medium
3/3	RemediationTaskE	7	Medium
2/2	RemediationTaskF	3	Low

High priority tasks will be listed first, followed by Medium and then Low. For each task the table will indicate the task Index (e.g. 1/n), Name, Count (number of issues that the task applies to), and its priority.

Note: Basic familiarity with field codes is assumed in this tutorial. For information about field codes refer to your MS Word documentation.

Note: A complete list of valid merge fields can be opened by clicking **Tools > Report > Template Based > View Field Reference**.

Procedure

1. Open the list of valid merge fields by clicking **Tools > Report > Template based > View Field Reference**. You will need this as a reference from which to copy field names.
2. Open an MS Word document, type in the heading "Remediation Tasks", and save it.
3. **Create the table header:** Create a table with 4 columns and 1 row and type in the headers "Index", "Name", "Count", and "Priority".
4. **Add the fields that will generate the High priority entries in the table:**
 - a. From the Field Reference table, copy the code: AS:RemediationTypeRepeaterStart<Priority=High>
 - b. In the Word document, place your cursor after the header table.
 - c. Paste the merge field into the document *as a field* (MS Word 2003: **Insert > Field > MergeField > Field name field**; MS Word 2010: **Insert > Quick Parts > Field > MergeField > Field name**).

The merge field is created in the document in the form:

```
{MERGEFIELD AS:RemediationTypeRepeaterStart<Priority=High>\* MERGEFORMAT}
```

Note: By default Word displays the tags in a truncated form which, since they are long, may omit relevant information. To see the full tag names "show field codes" by clicking **[Alt] + [F9]** (toggles field codes on/off).

- d. Click **Enter** and create another table with 4 columns and 1 row. This table will later be populated with all the rows of High priority tasks.
- e. After the table add the following merge field: AS:RemediationTypeRepeaterEnd

This appears as:

```
{MERGEFIELD AS:RemediationTypeRepeaterEnd\* MERGEFORMAT}
```

You have now entered the opening and closing merge fields for the section of the document listing High Priority remediation tasks. These two fields form a "loop" that will create a listing based on the fields inserted between them. You can now enter the four fields that will create the *content* in the four columns in the table you added above.

- f. In the left hand column, add the merge field: AS:RemediationTypeRepeaterIndex. This will create a counter (in the form: 1/n, 2/n, etc.) for each of the n tasks in this section.
- g. In the second column, add the merge field for RemediationTypeName.
- h. In the third column, add the merge field for RemediationTypeCount.
- i. In the fourth column, type the word "High".

The High priority section of the table is now complete. It will produce multiple rows of data: one for each High priority remediation task in your scan results. Your document should now look like this:

Index	Name	Count	Priority
{ MERGEFIELD AS:RemediationTypeRepeaterStart<Priority=High> * MERGEFORMAT }			
{ MERGEFIELD AS:RemediationTypeRepeaterIndex * MERGEFORMAT }	{ MERGEFIELD AS:RemediationTypeName * MERGEFORMAT }	{ MERGEFIELD AS:RemediationTypeCount * MERGEFORMAT }	High
{ MERGEFIELD AS:RemediationTypeRepeaterEnd * MERGEFORMAT }			

5. Repeat Step 4 for Medium and Low priority tasks (in the Priority column, type "Medium" and "Low" respectively). Make sure that each table row (High, Medium and Low) has a Start merge field before it and an End merge field after it, as shown in the image below.

Note: The field content of the three rows is the same, so you can copy-and-paste the "High" priority row in between the Start and End tags for Medium and Low priority tasks, simply changing the text in the "Priority" column.

Note: Since the Name column will contain text strings, while the Index, Count and Priority columns will contain only numbers or a short word, you may want to adjust the column widths with this in mind.

Index	Name	Count	Priority
{ MERGEFIELD AS:RemediationTypeRepeaterStart<Priority=High> * MERGEFORMAT }			
{ MERGEFIELD AS:RemediationTypeRepeaterIndex * MERGEFORMAT }	{ MERGEFIELD AS:RemediationTypeName * MERGEFORMAT }	{ MERGEFIELD AS:RemediationTypeCount * MERGEFORMAT }	High
{ MERGEFIELD AS:RemediationTypeRepeaterEnd * MERGEFORMAT }			
{ MERGEFIELD AS:RemediationTypeRepeaterStart<Priority=Medium> * MERGEFORMAT }			
{ MERGEFIELD AS:RemediationTypeRepeaterIndex * MERGEFORMAT }	{ MERGEFIELD AS:RemediationTypeName * MERGEFORMAT }	{ MERGEFIELD AS:RemediationTypeCount * MERGEFORMAT }	Medium
{ MERGEFIELD AS:RemediationTypeRepeaterEnd * MERGEFORMAT }			
{ MERGEFIELD AS:RemediationTypeRepeaterStart<Priority=Low> * MERGEFORMAT }			
{ MERGEFIELD AS:RemediationTypeRepeaterIndex * MERGEFORMAT }	{ MERGEFIELD AS:RemediationTypeName * MERGEFORMAT }	{ MERGEFIELD AS:RemediationTypeCount * MERGEFORMAT }	Low
{ MERGEFIELD AS:RemediationTypeRepeaterEnd * MERGEFORMAT }			

6. Add explanatory text and apply any formatting to the rest of the document as required.
7. Save the file.
8. To create a report using this template, see "Importing a custom template" on page 233

Tag hierarchy

When creating reports using AppScan tags you may want to place one tag "within" another. For example, you may want a separate section for each Issue Type, and within each section a list of Issues, and for each issue a list of Variants. To do this you place the start and end tags for the "child" within the start and end tags for the "parent", as shown in the figure following.

```

«AS:IssueTypeRepeaterStart<Severity=High>»
«AS:IssueTypeRepeaterIndex»

  «AS:IssueRepeaterStart»
  «AS:IssueRepeaterIndex»

    «AS:VariantRepeaterStart<Limit=1>»
    «AS:VariantRepeaterIndex»
    «AS:VariantRepeaterEnd»

  «AS:IssueRepeaterEnd»
«AS:IssueTypeRepeaterEnd»

```

The following repeater (loop) hierarchies are allowed:

- Issue Type > Issue > Variant
- Remediation Type > Remediation

- Vulnerable URL > Issue Type > Issue > Variant
- Remediation > Issue Type > Issue > Variant

Repeaters should be used either alone, or in one of these hierarchies. Using the repeaters in hierarchies other than these may result in an error when the report is created.

See also:

“Repeaters and fields”

Repeaters and fields

Within each repeater (loop) there must be one or more plain fields, that will be the data input to the loop. (For example, within the Variant loop you could put the VariantDifference field, which will show the difference between the variant and the original for each variant in the loop.) By default, the Index field is added within each repeater, but you can delete it, or add additional fields (as in the tutorial).

Repeaters and Fields Example:

```
<<AS:IssueTypeRepeaterStart>>      [Start looping through issue types]

  <<AS:IssueTypeRepeaterIndex>>      [For each type list index (e.g. 3/24) and name]
  <<AS:IssueTypeName>>

  <<AS:IssueRepeaterStart>>          [Start looping through individual issues]
    <<AS:IssueRepeaterIndex>>        [For each issue list index (e.g. 3/24)]
    <<AS:VariantID>>                 [For each issue list ID and variants]
    <<AS:VariantTestRequest>>

  <<AS:IssueRepeaterEnd>>            [End issue repeater]
<<AS:IssueTypeRepeaterEnd>>        [End issue type repeater]
```

Field filters

About this task

Some of the merge fields include filters. For example, the VariantRepeater limit can be used to set how many variants will be included per Issue.

Note: In the VariantScreenShot field, the filter defines the scale ("Scale=50") by percentage (50%). You can change this to any integer value up to 100 (for full size).

Procedure

1. Select the field.
 <<AS:VariantRepeaterStart<Limit=1> >>
2. To see the full field names, click **[Alt]+F9** (or right-click on the field and select **Toggle Field Codes** from the menu).
 {MERGEFIELD AS:VariantRepeaterStart<Limit=1>*MERGEFORMAT}
3. Update the filter between the angled brackets.
 {MERGEFIELD AS:VariantRepeaterStart<Limit=4>*MERGEFORMAT}
4. To hide the full field names, click **[Alt]+F9** (or right-click on the field and select **Toggle Field Codes** from the menu).
 <<AS:VariantRepeaterStart<Limit=4> >>

Importing a custom template

About this task

If you have created and saved a custom template (see “Custom template tutorial” on page 229) you can add it to the list that appears in the Template Based Reports list.

Procedure

1. In *Template Based* view of the *Report* dialog-box, click **Import**.
2. Browse to your custom template file and click **Open**.

The template is added to the list in the left pane, and can be used for generating custom reports.

Chapter 11. Tools

This section explains how to use additional tools provided with IBM Security AppScan Standard.

Options dialog box

This section describes options you can control, to customize AppScan, from the **Options** dialog box (**Tools > Options**).

Changes made in this dialog box apply to AppScan across sessions and scans. The dialog box has five tabs.

Scan Options tab

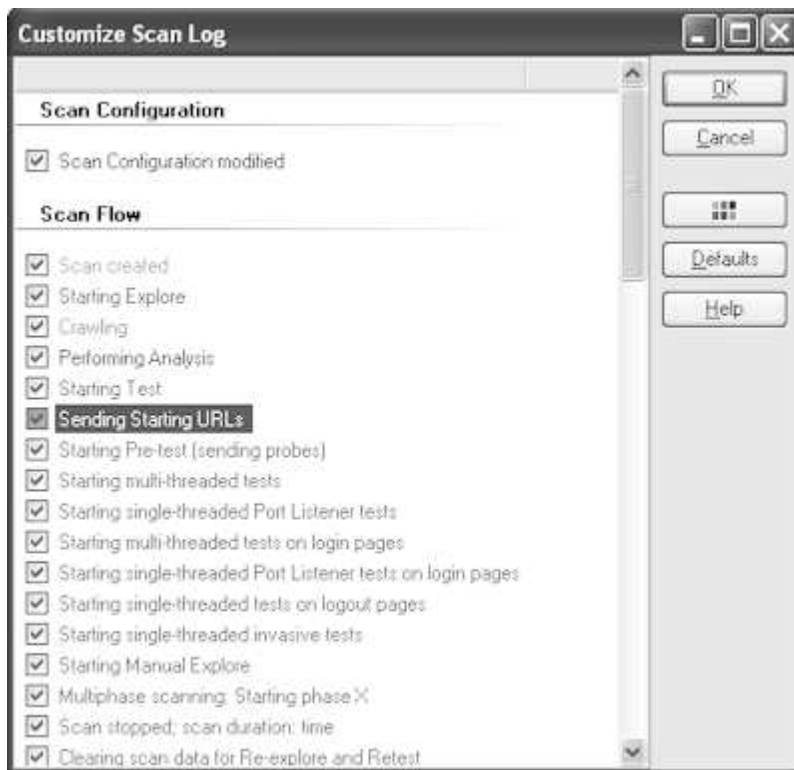
Tools > Options > Scan Options tab provides control over how AppScan operates during a scan.

Option	Description
<i>Scan Monitoring:</i>	
Enable Scan Log	You can enable/disable the scan log, and can use the Customize button to choose exactly which items appear in the log, and their color. See “Customizing the Scan Log.”
Enable request/response logging	You can enable the Request/Response Log , which can help technical support if you need assistance. However, turning on the log may affect performance. It should be disabled unless you need it.
Monitor memory consumption and disk space	AppScan checks available virtual and physical memory and disk space, and will send a warning if the available memory or disk space drops below the recommended amount. If you clear the Monitor memory consumption and disk space check box (disable memory and disk space monitoring), AppScan does not perform the checks.
Indicate if scan is incomplete	If selected, Scan Incomplete appears at the bottom left corner of the Result List, when a scan is stopped before completion, to remind you that the results are only partial.
Automatically save during scan	Save a file of the scan (.scan) as it runs. Interval in Minutes: Enter the number of minutes between saves; or use the slide bar to change the interval.
<i>Communications:</i>	
TLS/SSL support	If your application uses secure protocols to communicate with clients, configure AppScan to support the relevant protocols. Options are TLS 1, 1.1, 1.2 and SSL 2. Changing this option requires that you restart AppScan.
HTTP version	Choose the HTTP version of the site: 1.0 or 1.1

Customizing the Scan Log

You can control which items appear in the scan log, and their color.

The “Scan Log” on page 274 lists actions performed by AppScan during the current scan. The **Customize Scan Log** dialog box (**Tools > Options > Scan Options tab > Customize**) contains a scroll-down list showing all items that can be included in the log, a check box indicating whether they are currently included, and colored text indicating the color they appear in the log.



- To exclude/include an item from the scan log, simply deselect/select the check box next to it.
- To change the color an item appears in the scan log: Select the item, click , select a color from the palette that opens, then click **OK**.
- To restore the default settings, click **Default**.

Enabling Loopback for MS Edge with Windows 10

Special configuration may be necessary you use Edge as the external browser with Windows 10 OS.

MS Windows 10 uses an isolation technology ("AppContainer") that may prevent requests being sent from Edge via AppScan to your application. Although this does not always happen, we recommend enabling loopback before scanning with Edge configured as an external browser.

Instructions for enabling loopback can be found on this Microsoft troubleshooting page:<https://msdn.microsoft.com/en-us/library/windows/apps/Hh780593.aspx>

- To enable loopback for Edge, use this command:
`CheckNetIsolation LoopbackExempt -a -n="Microsoft.MicrosoftEdge_8wekyb3d8bbwe"`
- To disable loopback for Edge use this command:
`CheckNetIsolation.exe LoopbackExempt -d -n="Microsoft.MicrosoftEdge_8wekyb3d8bbwe"`

Note: For additional information see: <https://blogs.msdn.microsoft.com/fiddler/2011/12/10/revisiting-fiddler-and-win8-immersive-applications/>

Preferences tab

Tools > Options > Preferences tab provides various controls over AppScan files, fix recommendations that appear in the GUI, and the Welcome screen.

Option	Description
Display Welcome screen on startup	The Welcome screen appears when AppScan launches. It offers quick access to configuring and loading scans. If you find that you are closing the Welcome screen more often than using its features, you can stop it from opening by clearing this check box.
Check for updates on startup	AppScan will check for updated tests, security issues, and fixes when it launches. To disable this check, clear this check box; you can get updates whenever you want: Help > Check for Updates.
Automatically install security updates	If this check box is selected, the updates are installed while you work, without your intervention.
Report False Positive: Encrypt files	This setting applies to “Report false positive test results” on page 199. AppScan will encrypt selected variant information files that have been zipped, before attaching them to a new post in your default e-mail client. Clear this check box if the email you want to send is to anywhere other than theAppScan support team.
Run Scan Expert before Scan	(Selected by default) When selected, Scan Expert runs automatically whenever you run a full scan (Explore and Test), but not when you select Explore Only or Test Only.
Include advisory training videos	For some of the Advisories, short training videos are available that can be embedded in the Advisory tab for viewing when you view the advisory. Deselect if you need to reduce file size.
Fix Recommendation Settings	If your application does not use one or more of the environments listed, you do not need to see fix recommendations that are specific to those environments. Clear the check boxes of any environments (.NET / Java EE / PHP) that are not relevant to your application.
Recording and Viewing Browser	Select the default browser that will be used so show the Starting URL, and to browse when exploring manually. Choose one of the two radio buttons and then select a browser from its drop down list. • Use embedded browser: If you select this option you can choose between the built-in AppScan IE and AppScan Chromium browsers. • Use external browser: If you select this option you can choose from supported browsers installed on your machine.

Recording Proxy tab

Use this tab to configure AppScan to act as a proxy for external browsers, or for manually exploring non-SOAP web services using either a remote device (such as a mobile phone), or a local application (such as a simulator or emulator).

Option	Description
Proxy Port	Specifies which port AppScan uses. When using AppScan as a proxy server you need to configure the external browser or mobile device to use this port. Use the check box to select whether AppScan selects an available port automatically, or lets you choose the port. Note that if the port is chosen automatically it may change between sessions, and you will therefore need to re-configure your mobile device.

Option	Description
External Connections	This setting determines which connections to external domains are accepted. Reject all (Default) Connection attempts from all external IPs will be rejected. Use this setting only if you will be exploring using an application on the same machine as AppScan. Accept white list only Connections from external IPs that appear on the white list will be accepted; all others will be rejected. Accept white list and prompt for others Connections from external IPs that appear on the white list will be accepted automatically; for all others the AppScan user will be prompted, with the option of adding the new IP to the white list. Note that prompts are seen only if the External Traffic Recorder is open.
White List	Connections from IPs listed here will be accepted automatically. To add new IPs to the list, click , and select an option: - To add a single IP to the list, type in the IP and optionally a description. Tip: If you will be using a remote device but are not sure of its IP address, or if it changes frequently, select White List and prompt for others. The first time the device connects with a new IP, a pop-up appears giving you the option to add it to the white list. - To add a <i>range</i> of IP addresses, add an IPv4 address and subnet mask, or an IPv6 address and subnet prefix length, and optionally a description.

Option	Description
AppScan SSL Certificate	If the server uses HTTPS, since AppScan has to act as a proxy in order to record the traffic between the web service and the device you use to manually explore, it will be sending SSL certificates to the device instead of the web service's certificate. When a browser receives an unrecognized certificate it typically warns the user with a pop-up, but in the case of a mobile device the request is usually just ignored. It is therefore impossible to explore the application unless the AppScan certificate is accepted on the device sending the requests. Add Adds the AppScan SSL certificate to the root certificates on this machine. You must do this to allow sending requests to the web service. The AppScan certificate will be added to the root certificate, and requests from the web service to the simulator will not be rejected. Note: After you have added the certificate, the button changes to Remove, and can be used to remove the certificate from the AppScan machine. Export Saves the AppScan SSL certificate <i>that is currently installed on this machine</i>, as a ZIP file, so it can be added manually to the root certificates on a different device. Note that you do not usually need to do this, as you can import the certificate directly from the device in most cases. 1. In AppScan, click Scan > Manual Explore > Using External Device The External Traffic Recorder opens with status "Waiting for incoming connections". Important: Leave it open for the next sub-steps. 2. On the mobile device, browse to http://appscan 3. In AppScan, if you are prompted to allow an incoming connection from your device, click OK. When the device connects successfully to AppScan as its proxy, a message (on the device) confirms the connection, IP and port. If the certificate is installed on the AppScan machine, it also provides a button to install it on the device. Note: If the button is grayed out, the certificate is not installed on the AppScan machine. Note: The device's domain and request will appear in the External Traffic Recorder lists. 4. On the mobile device, tap Install AppScan SSL Certificate The certificate is installed. Note: If the device is unable to access the application you are testing after this procedure, you need to install the certificate (onto the remote device or application) manually: a. In AppScan, open Tools > Options > Recording Proxy b. Click Export and save the certificate as a ZIP file. c. Install the certificate as a root certificate on the device or application. 5. When finished, click Cancel on the External Traffic Recorder, to close it. Note: This option is active only if the certificate is already added to the root certificates on <i>this</i> machine. Attention: The AppScan certificate that is exported must be identical to the one installed locally. If you Remove the local certificate and then Add again, you must also reinstall it on the device, as the new certificate is not identical to the previous one.

For more information, see "Using AppScan as recording proxy" on page 131

General tab

Tools > Options > General tab provides options to control how AppScan prioritizes issues, gets updates, and displays dialog boxes.

Option	Description
File Locations	During use, AppScan writes various files to disk. You can change the locations to which these files are saved by clicking the browse button to the right of the file type, and browsing a different location.
	User files *.scan (Scan), *.exd (Explore Data), *.xml (Report, Export), *.scant (Scan Template), *.asreg (Report Template), *.aspol (Test Policy)\My Documents\AppDataScan
	Log files *.log (Log), *.lic (License), *.dmp (Memory Dump), *.css (Packed Support) [AppScan Standard installation folder]\Logs
	Custom advisories *.xml (User-created advisory file) ...\My Documents\AppDataScan\Advisories
	Noise classification file *.xml [AppScan Standard installation folder]\IssueManagement (For details see "Issue state: Open or Noise" on page 187.) Clicking the  button deletes the Noise Classification file, restoring default noise classifications.
Log File Size	Set the maximum log file size, in MBytes. When the limit is reached the file is saved as a backup and a new log file is opened. (The next time a backup is saved the earlier backup is deleted.) If you have limited space on your hard drive, you may want to limit the size of the AppScan log file.
Operating system priority level	By default, AppScan has a Normal priority among running applications. If other applications need more of your computer's resources, you can change the value of AppScan to Idle , to a lower priority, allowing it to run only if other applications do not take the resources. CAUTION: Changing the operating system priority level can dramatically affect the performance of other applications currently running on the same system.
Reset Dialog Suppress	If you have selected the "Don't Show this again" check box in any AppScan warning, you can reset them to display again by clicking the Reset Dialog Suppress button.
Clear History	Click to clear all history lists: URL History, Recent Scans list, Recent Templates list, Recent search strings, and recent user-defined policies.
Select Language	If AppScan can work on your workstation in more than one language, you can select the interface language here. After changing this setting you may be asked to close and reopen AppScan for the change to take effect.

Advanced tab

Tools > Options > Advanced tab lets you view and change the default values of advanced settings.

The Advanced tab lists the Type (String, DWord or Boolean) and current Value of many settings, and lets you change them. When you select a setting by clicking on it, a brief description of the setting and how to use it appears at the bottom of the list.

- To locate a particular Preference Name (or names), type a word or fragment into the filter field at the top of the dialog. Use the drill-down arrow to **Match case** and/or **Match whole word**
 - To change a setting, click on the Value field, and select or type in the new value.
- Settings that have been changed from their default values appear in bold.

Note: If you try to enter an invalid value a warning appears and the value is not accepted.

- To restore the default value of a *single* Preference, right-click on the Preference and select **Restore Default**.
- To restore *all* settings to their default values, click the **Restore Defaults** button at the bottom of the dialog.

Web Services Configuration wizard

This extension lets you scan using Open API description files. It is available from Tools > Extensions > Web Services Wizard (Open API), and the extension is enabled by default.

This AppScan extension supports web service scans based on Open API (v2 and v3) description files (JSON or YAML). The steps below show the wizard workflow. Click the step name to see details for that step.

Note: This extension explores web services only. Any other links are ignored.

Note: Using API keys as HTTP query parameters is not supported.

Table 9. Web Services Configuration workflow

Step	Step Name	Description
1	Description Files	Add one or more Open API description files that define the web service.
2	Domains	Domains found in the description files are added to the list of domains that can be scanned. In this step you can remove any that should not be scanned.
3	Login Management	Define the login procedure for the web service.
4	"Sequences" on page 243	Review the requests created from the description files, and their parameters, and create "sequences" of requests that must be sent in a specific order. Important: Correctly constructed sequences of requests are <i>essential</i> to enabling AppScan to create objects that depend on the previous creation of another object.
5	Parameters	Review all the parameters found in the requests. You can select which parameters are tracked and which are not tracked, and edit their values.
6	Complete	When configuration is complete decide whether to start the scan now or later.

Additional task:

After completing the wizard configuration, a possible additional task (depending on your service) may be configuring Custom Headers in the main AppScan Configuration dialog box. For details, see "Sequence variables" on page 89.

Description files

Add one or more Open API description files that define the web service.

Note that you cannot add local files, only URLs. If necessary upload your description files to your web server and provide the URL.

To create your list of description files:

1. In the URL field enter an Open API link and click Add. The URL is validated and added to the list.
2. Add more URLs as needed.
3. When the list is complete, click **Next**. The links are re-validated and the next step opens.

Next Step: “Domains”

Domains

All domains found in the description files are added to the list of domains that can be scanned. In this step you can remove any that should not be scanned.

Domains listed in the right ("Included domains") pane will be scanned. You must move any domains that should not be scanned to the left ("Excluded domains") pane.

To exclude a domain from the scan:

- Select the domain, and click the Left arrow. The selected domain is moved to the Excluded Domains pane.

Next Step: “Login Management”

Login Management

Configure the login procedure for the web service.

If a login is required you must configure it so that so that AppScan can log in to the service.

Limitation: Using API keys as HTTP query parameters is not supported.

Select one of the Login radio buttons:

Configure Login below

If you select this option the lower part of the dialog becomes active and lets you input the following:

1. **Login request:** Select a login request from the drop-down list of requests from the description files.

Note: If the web service implements authorization control using API keys, a login request is *not* needed, so select **None** from the drop-down list.

2. **Login credentials:** Review and if needed edit the values of Login credentials.
3. **Custom headers:** If the service uses custom headers (such as bearer authentication in the Authorization header), click **Edit** to open the Add Custom Header dialog box. For details see “Custom Header tab” on page 78.
4. **In-Session Detection request:** Select an in-session request from the drop-down list. This will be used by AppScan to verify that it is logged in when testing.

Use existing Login configuration

Select if your scan configuration already includes a valid login sequence you can use it.

Record Login sequence in AppScan Configuration > Login Management

Select if the description file does not include a login request. You can use the main AppScan Configuration dialog box to record the Login using the AppScan built-in browser or an external device. This is most likely to be the case when users log in through a user interface, or where JavaScript is involved in the login process. For details see “Login Management view” on page 44.

None Select if the service does not require logging in.

Next Step: "Sequences"

Sequences

Review the list of requests created from the description files (and their parameters), and define "Sequences" of requests that must be sent in a specific order (objects that depend on other objects being created first).

Requests

The **Left pane** of the Requests tab shows you a list of all requests created from the definition files; for each one its Method and Path. Select any request to see:

- **Upper-right pane:** A list of the parameters in the request, with their Name, Type, Location and Value. You can also click the **Edit** icon to see its Description, and edit its Value, Track the parameter, and Apply the Value change to all similar parameters.
- **Lower-right pane:** The raw HTTP request, including headers etc., exactly as it will be sent. You can click the **Send** button to send the request and see the response in the lower part of the same pane.
- Exclude a request from the scan by **Right-click > Exclude Request**. The request appears with a strikethrough. To re-include it right-click again use **Right-click > Include request**.

Sequences

Correctly constructed sequences, expressing dependencies between objects, are important tools in scanning a web service thoroughly. When a request to create a particular object depends on another object having been created previously, the right sequence of requests must be configured.

To configure a sequence:

1. In the **Requests tab**, click on the first request in the sequence to select it.
2. Click and hold down the **Control key**, and select the remaining requests in their correct order. Then release the Control key. All requests in the sequence now appear selected.
3. Enter a name for the sequence in the Name field, and then click **Create Sequence**.
4. Click on the **Sequences tab** to see:
 - **Left pane:** All configured sequences. Each sequence can be enabled for the scan or disabled to save for another scan, by selecting or clearing the **Enabled** check box.
 - **Upper-right pane:** The URLs recorded in the selected sequence. You can use the **Up/Down** arrows to change the order of the requests, and the **Minus** button to remove requests from the sequence.
 - **Lower-right pane:** The list of variables found in the selected sequence. You can right-click a variable to set a *dynamic value*. For more details See "Sequence variables" on page 89.

Next Step: "Parameters"

Parameters

Review all the parameters found in the requests. You can select which parameters are tracked and which are not tracked, and edit their values.

This stage displays all parameters found in the requests across the application as a single list, and is particularly useful when you want to track, or change the value of, multiple parameters. You can:

- Edit the Value and Tracked status of individual parameters by clicking on the parameter.
- Change the Default general value, Username and Password.

Note: If defaults are defined in the description files they are used unless changed here. If there are definitions in the Form Filler configuration they are used for parameters not defined in the description file.

- Restore default values after changes were made.

Next Step: “Complete”

Complete

When configuration is complete decide whether to start the scan now or later.

Now your configuration is complete you can:

- **Start a Full Automatic Scan**
- **Start with Automatic Explore Only** (so you can review the Explore results before continuing with the Test stage)
- **Start scan later**

By default the wizard applies the Web Services Test Policy, unless your configuration specifies a non-default test policy. You can change this by selecting or clearing the check box.

Scan Scheduler

You can set scans to start automatically on a schedule. Scan schedules may be one time in the future or on a routine basis. When the time of a scheduled scan arrives, AppScan is automatically opened and it runs the scan.

Schedule a new scan

About this task

You can schedule scans using the configuration of the current scan or of a saved scan. The scheduled scans are given the name of the source scan with the date and time appended, and saved in the same folder as the source scan.

Procedure

1. On the **Tools** menu, click **Scan Scheduler**.
The **Scan Scheduler** dialog box appears, listing any scans scheduled so far.
2. Click **New**.
The **Schedule Settings** dialog box appears.
3. Enter a name for the schedule.
4. Decide whether you want to schedule the current scan or schedule a loading of a saved scan (*.scan file).
If you choose the current scan, and the current scan is unsaved, AppScan asks you to save the scan as a scan file, to be loaded by the scheduler.
5. Select the recurrence schedule: daily, weekly, monthly, or once on a specified date and time.
6. Select the date and time that the first scan will start.
7. You can optionally set a time limit for the scan, by selecting the **Limit Scan Time** check box, and entering the maximum duration of the scan in minutes.
8. Enter your domain name, user name, and password.
These authentication values enable the task to run as if started by the given user; without this information, the task may not be able to run at all.
9. Click **OK**.

The schedule name appears in the **Scan Scheduler** dialog box.

Edit scheduled scan configuration

Procedure

1. In the **Scan Scheduler** dialog box (**Tools | Scan Scheduler**), select a scheduled scan.
2. Click **Edit**.
The **Schedule Settings** dialog box appears.
3. Edit the details as needed.
4. Click **OK**.

Delete a scheduled scan

Procedure

1. In the **Scan Scheduler** dialog box (**Tools > Scan Scheduler**), select a scheduled scan.
2. Click **Delete**.

Schedule a Test stage only

About this task

A full scan consists of two stages: Explore and Test. If it is not necessary to run the Explore stage each time, you can configure a scheduled scan to run the Test stage only. This is done using *Scheduled Tasks* on the *Windows Control Panel*.

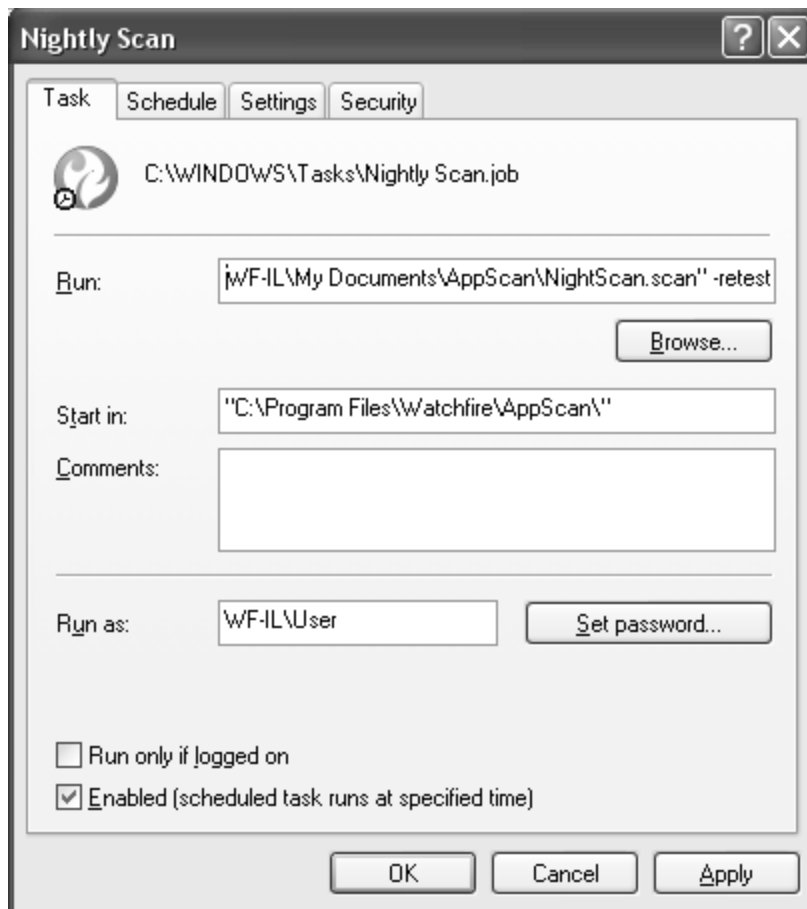
Procedure

1. In AppScan, “Schedule a new scan” on page 244 (**Tools | Scan Scheduler | New**).
2. On the Windows **Start** menu, click **Control Panel | Scheduled Tasks**.

Note: If the *Scheduled Task* option does not appear on your *Windows Control Panel*, you may be in *Category View* which does not display all options. If necessary switch to *Classic View*.

The Windows Scheduled Tasks dialog box opens, with the task that you created in AppScan on the list of tasks.

3. Right-click on the AppScan scheduled scan and select **Properties** from the pop-up menu.
The *Scheduled Task Properties* dialog box opens.



4. In the *Run* field, at the very end of the text string (after the final quotation marks), type in: [space][hyphen]retest
5. Click **OK**.

When the scheduled time arrives the Test stage only will run.

Schedule a scan in installments

About this task

If you have limited "windows of time" in which to run scans (for example, you have just three hours a day, between 1.00 and 4.00 am), and your scans need more than this amount of time to complete, you can configure a scheduled scan to run in installments. In order to do this you need to tell AppScan:

- To save the scan with the original filename (without the timestamp as part of its name)
- To continue the next scheduled scan from where the previous scan left off

This is done by adding command line parameters via *Scheduled Tasks* on the *Windows Control Panel*.

Procedure

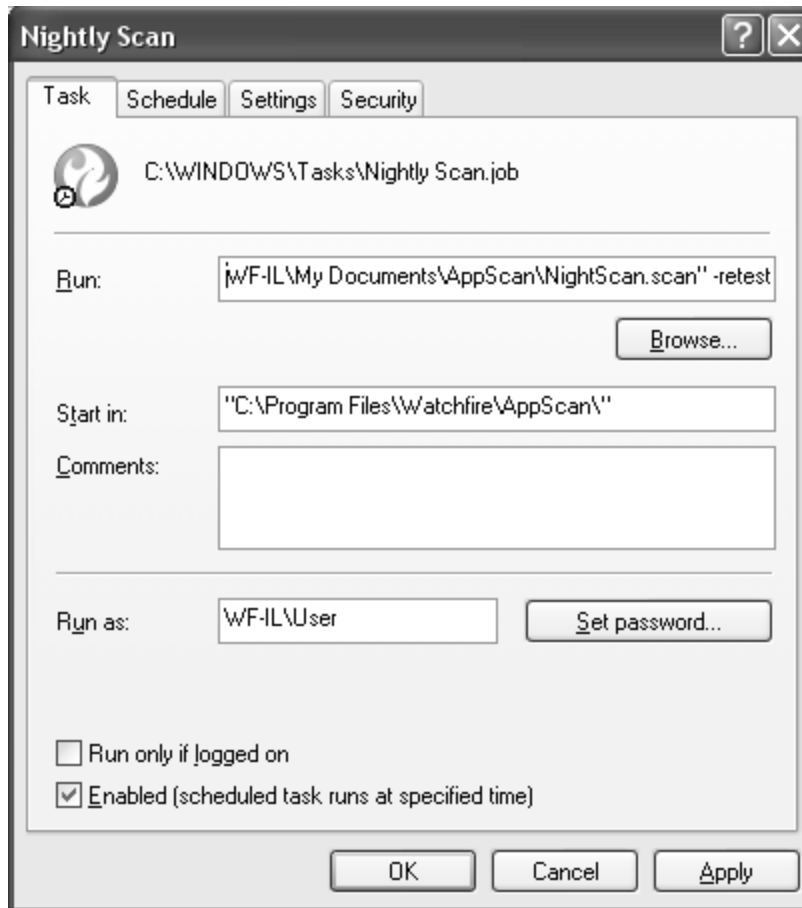
1. In AppScan, "Schedule a new scan" on page 244 (**Tools > Scan Scheduler > New**).
2. In the Windows Start menu **Run** field, type Task Scheduler
The Windows Scheduled Tasks dialog box opens, with the task that you created in AppScan in the list of tasks.
3. Right-click on the scheduled scan, and select **Properties**.
The Scheduled Task Properties dialog box opens.

4. In the Triggers tab, configure the task as needed. Refer to Microsoft Windows documentation for further information.

Scheduled task command line parameters

The table below shows the available command line parameters for use in the *Scheduled Tasks* dialog box of the *Windows Control Panel*.

Parameters are added after the main string in the Run field of the Task tab.



Command Line Parameter	Function
[space][number]	This number is the timeout, in minutes, for the scheduled scan.
[space][hyphen]retest	Configures the scheduled scan to start from the Test stage, omitting the Explore stage. This might be useful for re-testing a site at intervals, using a source scan that has already explored the site.

Command Line Parameter	Function
[space][hyphen]continue	Configures the scheduled scan to continue from where it left off rather than starting again. This might be useful for re-testing a site at intervals, using a source scan that has already explored the site. Note that when using this option to scan <i>in installments</i> you must also use the <i>saveName</i> switch, so that the scheduled scan overwrites the original one each time it runs. (See “Schedule a scan in installments” on page 246.)
[space][hyphen]saveName[filename]	Defines the name that the scheduled scan will be saved with. When no <i>saveName</i> is defined, AppScan saves the scheduled scan with the source filename followed by a timestamp.

User-Defined Tests

AppScan provides a database of thousands of tests. However, if your web application has issues that are specific to it, or if you want to write your own advisories for fixing issues, you can create your own tests. These tests are saved and included in your AppScan database of tests.

Each test looks for one specific issue. For example, one test will modify a path in a request; another test will modify user input to include a character that should be invalid. For each test you can define multiple conditions for the following three :

- **Filter:** What conditions must be met in order to execute the test.
- **Modification:** What are the changes will me made to the request.
- **Validation:** What conditions must be met for the test result to be considered positive.

To create and manage user-defined tests:

- Click **Tools > User-Defined Tests**.

The User-Defined Tests dialog box appears, listing tests that have been defined, and their type. The check box next to each test indicates whether or not it is currently enabled in scans.

Option	Description
Enable check box	Select/clear to include/exclude a test from the current scan.
	Click to Export All or Export Selected user-defined tests, and their enabled/cleared status, as a UDT file that can be imported to a different scan.
	Click to import a previously saved UDT file. The tests in the imported file are added to the current list of tests.
	Select a test <i>row</i> (not its check box), and click Edit to open the User-Defined Test wizard to edit the selected test.
	Click to delete the selected test.
	Click to open the User-Defined Test wizard to create a new test.

User-Defined Test wizard

The User-Defined Test wizard lets you create user-defined tests for AppScan to use when scanning, in addition to the tests it creates automatically.

In the welcome screen you define the basic test attributes.

Test Attributes

Name	The name you assign will appear in scan results and reports.
Description	The description appears only in the User-Defined Tests list within AppScan.
Created by	This attribute is used internally by AppScan to distinguish between user-defined tests of the same name and alert you to such conflicts.
Severity	Assign a High, Medium, Low or Informational severity level to the issue

When done, click **Next** to move to the next step.

Test Type

In this step you define the kind of test you are creating. This affects which wizard steps are displayed.

The wizard steps that follow depend on the type of test you choose.

Modify Path (Infrastructure)	Creates a test that attempts to reach a specific URL during the scan.
Modify Parameters and Cookies	Creates a test that changes the value of one or more cookies and/or parameters.
Modify Full Request	Creates a test that changes the path of the original request by: • Setting, prepending or appending a Filename section • Adding, removing or changing a cookie or parameter • Overwriting the request body
Pattern Search (no modifications)	Creates a test that looks for the Validation conditions in all filtered Explore-stage responses from the application (you can define a filter or include <i>all</i> responses).
Global Validation (for all AppScan tests)	Creates a test that looks for the Validation conditions in all Test-stage responses from the application. A positive result will be listed as vulnerable to this user-defined test.

When done, click **Next** to move to the next step.

Filter

In this step you define the conditions which must be met to run this test. Only requests that match the conditions will be tested.

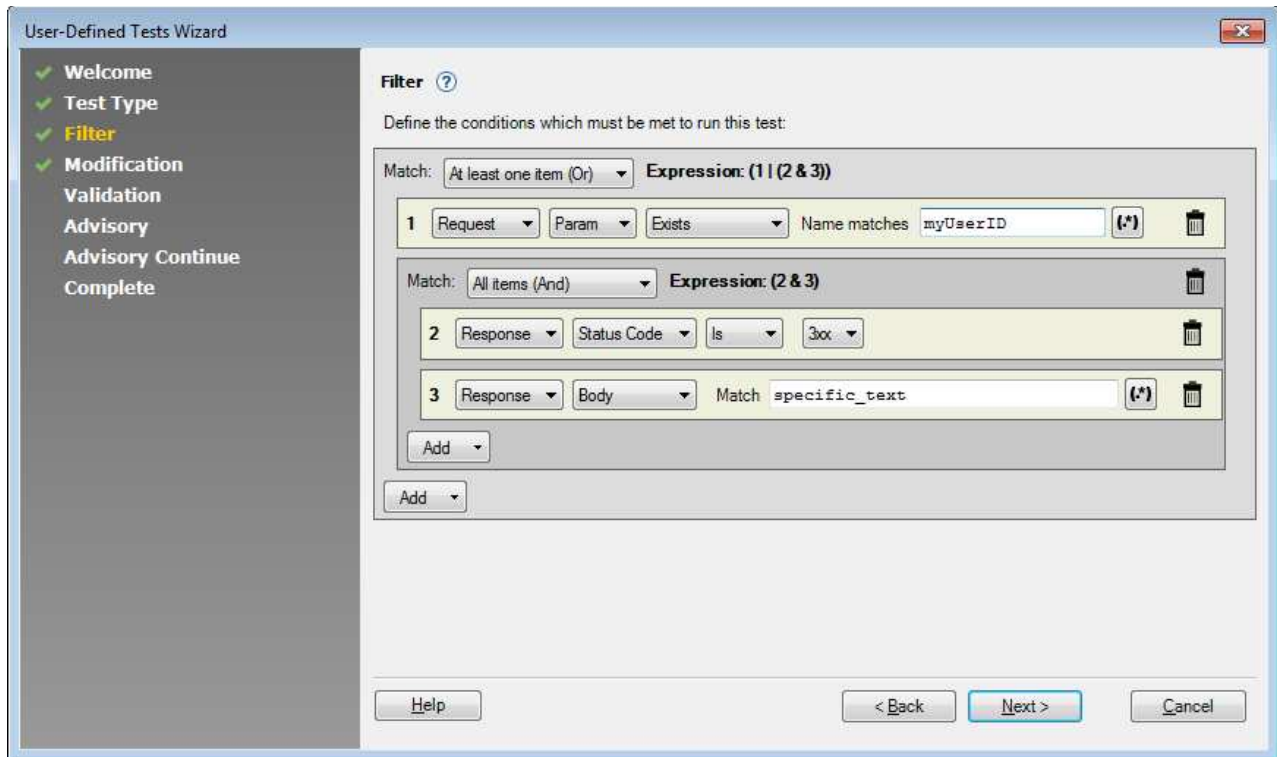
Defining filters that limit testing to relevant requests *only* can greatly improve efficiency.

- Click **Add** to add another **Single** or **Group** filter.
- When adding more than one item, you must specify the logical relation between them: **All** (And), **Any** (Or) or **None** (Not). See example below.

When done, click **Next** to move to the next step.

Example

In this example three conditions have been added. The new test will be run only on responses that meet either Condition 1 (Request includes my myUserID parameter), or *both* of Conditions 2 and 3 (Response Status Code is 3xx *and* Response Body contains specific_text).



As an expression this is:

(1 | (2 & 3))

Modification

In this step you define the changes that will be made to the original request, to create the test request.

Define the Modification by adding one or more changes to be applied to the original request, to create the Test request.

- Click **Add** to add a change.
- If you add more than one change, they will be applied together as a single modification. To apply them separately you would need to create separate tests.
- If you want to include a dynamic value in the change, click the Macro Editor icon next to the Value field.

When done, click **Next** to move to the next step.

Macro Editor

The Macro Editor is used to add dynamic values (macros) to your user-defined test modifications.

To add a Modification value that includes one or more macros:

1. Click the **Macro Editor** icon next to the Value field.
2. Type any plain text you need into the text field.
3. Position your cursor at the desired insertion point.
4. Select a macro from the list below.
5. Click **Insert**.

Validation

In this step you define the conditions that indicate the test succeeded.

- Click **Add** to add another **Single** or **Group** filter.
- When adding more than one item, you must specify the logical relation between them: **All**, **Any** or **None**.
- In most cases the validation is applied to the response to the test request, and this is the default behavior. However in the case of certain vulnerabilities, for example Stored Cross-Site Scripting, the validation must be done on the response to a *different* request. In that case select the second radio button (**Run this validation on the response to the following request**), and then define the request.

Tip: For more control over the URL definition, for example if you need to overwrite the body, or specify GET or POST, click **Edit Request**.

When done, click **Next** to move to the next step.

Example

For an example, see the “Filter” on page 249 step.

Advisory

In the Advisory and Advisory Continue steps, you input the text content of the Advisory that will appear in the scan results and reports. All fields are optional.

Technical Description	A technical description of this issue.
Affected Products	If this issue affects 3rd-Party products, list them here.
Remediation Task	Describe the general task needed to address the issue.
Fix Recommendation	Specify a solution or workaround for this issue.
References and Relevant Links	List any references or external links that provide more information about this issue.

When done, click **Next** to move to the next step.

Completing the wizard

When you have defined the new test, click **Finish** to add it to the list.

PowerTools

AppScan offers access to five utilities (PowerTools), each providing a specific feature to help you manage your application security or to help you use AppScan.

You can open the PowerTools from within AppScan (**Tools > PowerTools**) or independently from the Windows Start menu.

Authentication Tester

The Authentication Tester PowerTool is a testing utility that uses the "brute-force" technique to reveal weak username-password combinations that could be used to gain access to your web application. (A brute force attack is an automated process of trial and error used to guess authentication credentials, causing a server to acknowledge an imposter as a legitimate user.)

Using brute force, a malicious user will cycle through combinations until stumbling upon credentials that gains access to the authorized area. Using a brute force application, a malicious user can employ a

dictionary file, or even simply try all possible combinations of the accepted character set (depending on the username and password formats that the site accepts). Such an attack can generate thousands, even millions, of incorrect combinations before successfully gaining access, and typically takes hours, weeks, or more.

Ensuring that your web application enforces the use of strong passwords can greatly reduce the feasibility of brute force attacks.

- You can run Authentication Tester from AppScan, by clicking **Tools > PowerTools > Authentication Tester**
- You can run Authentication Tester on its own, from the Windows Start menu, by clicking **All Programs > [AppScan Standard installation folder] > PowerTools > Authentication Tester**

Authentication methods

In the Authentication Tester main window, you choose the authentication method used by your web application. Options are:

- **Form Authentication:** (authentication is performed by a custom web-page)
“Form authentication”
- **HTTP Authentication:** (authentication as defined in the protocol)
“HTTP authentication” on page 254

The scan options available change depending on which method you select.

Note: If the application uses *both* types of authentication (e.g. HTTP authentication in general and Form authentication for specific administrator pages), you must provide Authentication Tester with the *actual* username and password for the HTTP authentication, to enable it to test the Form authentication on these pages. (See “HTTP authentication over form authentication” on page 255 for details.)

Form authentication

About this task

If you select the **Form Authentication** radio button, perform the following procedures:

Procedure

1. “Provide a typical login.”
2. “Describe the application's login responses” on page 253.

Provide a typical login:

About this task

If you select **Form Authentication** in the **Authentication Method** section of the main window, the **Setup** button appears. This is used to configure Authentication Tester with the correct Login procedure.

Procedure

1. Click **Setup**.
The Authentication Tester browser opens.
2. Browse to the login page of your web application.
3. Perform the Login procedure using these credentials (you can cut-and-paste them from the upper part of the browser window):

username:	BruteUsername
password:	BrutePassword

Authentication Tester requires that you model the site login procedure using these values. During the testing stage, these strings will be replaced with possible username and password combinations as Authentication Tester attempts to gain access to the site by "Brute Force". When you complete the Login procedure now, however, Authentication Tester does not actually attempt to log in using these credentials, but simply examines the login request.

CAUTION:

If the strings "BruteUsername" and "BrutePassword" are not allowed by your client-side verification, no Login request will be created for Authentication Tester to examine! In such a case you must change the placeholder values for the username and password strings. See "Form Authentication tab" on page 256.

When you have completed the login process, Authentication Tester "captures" the login request, and a confirmation message appears.

4. On the confirmation message, click **OK**

The browser closes and the **Successful Login Detection** window opens. This is used to describe the login responses. See "Describe the application's login responses"

Describe the application's login responses:

About this task

In the Successful Login Detection window, you enable Authentication Tester to recognize login requests as either successful or failed. This information is necessary to know when the web application has accepted credentials as valid.

Some default Success responses appear in the list by default, but you should edit the list to include any and all responses that are specific to your application.

Procedure

1. Select the type of response that you want to describe:
 - **Success Response:** The response to a valid login attempt
 - **Error Response:** The response to an invalid login attempt
2. Enter a text string or regular expression (regex) that matches some content on the response page. (Be sure to match only static content, not variables.)

For example, if you know that invalid credentials often receive a response of: "Username and password do not match" you can use this to let Authentication Tester know the result of its tests.

The use of regular expressions, rather than strings, allows you to configure Authentication Tester once for multiple runs during the development stages of your web application.

For example, if the design of the successful-login page has not yet been finalized between a large, full page `Welcome!` note or a small `welcome` string at the top of the home page, you can enter `(?i)welcome` to indicate that the search word is case-insensitive.

Tip: To learn more about regexps, and the metacharacters that can be used to indicate more than literal strings, see "About metacharacters" on page 254. To test regular expressions before attempting to use them in Authentication Tester, try the Expression Test PowerTool.

3. Click **Add**.

The regexp is added to the list of responses.

You can add as many regexps as you want. Authentication Tester uses them with an OR operator: if one or more of the regular expressions matches content on a page of your site, that page is recognized as a result page (either successful login or error page, depending on your response type selection).

4. To remove unwanted regexps from the list, select the regexp and click **Remove**.
5. Click **OK**.

The **Successful Login Detection** window closes and you are returned to the main window. You can now run brute force tests using the current configuration (see “Running authentication tests” on page 255).

About metacharacters:

A metacharacter is one or more characters with a special, non-literal meaning in the context of a regular expression. For example, the circumflex character (^) is a metacharacter that means "search at the beginning". If you want to find the circumflex character, rather than the metacharacter pattern, you must protect (or "escape") it with a backslash: \^.

The table following lists examples of some common regular expression metacharacters.

	Metacharacter Description	Example
\	Protect the next character (interpret it literally and not as a metacharacter).	\! finds an exclamation mark (!) \. finds a period (.), rather than a character
^	Find at the beginning of a string.	^H finds H ome but not h ome or P HP
.	Find any character (letter, number, symbol, whitespace), except newline.	(.*) finds any paragraph
()	Find a pattern group.	(word) finds "In this w ord" ^(Word) finds " W ords in this line" Welcome ((back) (home)) finds " W elcome back" and " W elcome home"
[]	Find a pattern range.	[a-z] finds any lower-case alphabetic character
*	Find the pattern zero or more times.	<(.*> finds all HTML tags, with their content
+	Find the pattern one or more times.	<(.*>+ finds
?	Find the pattern zero or one time.	log(.*?)in finds l ogin and l og in
(?i)	Find the next characters with a case-insensitive search.	(?i)word finds w ord, W ord, w oRd, W ORD

HTTP authentication

About this task

If you select the **HTTP Authentication** radio button in the **Authentication Method** section, the **URL Address** and **Domain** fields appear.

Procedure

1. Enter the URL of the login page in the **URL Address** text-box.
2. To test that this is the correct URL, click **Show**.
The browser opens.
 - If a standard HTTP login window pops up in front of the browser, the URL is correct.
 - If the page appears without an HTTP login window, the URL is incorrect. Correct the URL.
3. Close the browser.
4. If the HTTP login window requires a domain, enter the correct domain name in the **Domain** field.
You can now run brute force tests using the current configuration (see “Running authentication tests” on page 255).

HTTP authentication over form authentication

About this task

In some instances an application may use *both* types of authentication. HTTP authentication may be used generally on all pages, while Form authentication is used to protect specific administrator areas. In such cases you will probably want to run both types of test.

HTTP Authentication: This is tested as described above (see “HTTP authentication” on page 254)

Form Authentication: To test this you need to provide Authentication Tester with the actual username and password for the HTTP authentication. This will enable it to “get past” the HTTP authentication and test the Form authentication on these pages.

Procedure

1. In the Authentication Tester main window, select the **Form Authentication** radio button.
2. Configure Form Authentication (see “Form authentication” on page 252).
3. Click **Advanced**.
The Advanced Configuration dialog box opens (with the General tab on top).
4. In the **HTTP authentication over Form authentication** area select the **Enable** checkbox.
The Username, Password and Domain fields become active.
5. Type in valid HTTP authentication credentials for Authentication Tester to use when testing Form Authentication pages.
6. If the HTTP login window requires a domain, enter the correct domain name in the **Domain** field.
7. Click **OK** to close the dialog box.
You can now run brute force tests using the current configuration (see “Running authentication tests”).

Running authentication tests

After you have chosen the authentication method and configured the basic tests, you can start the Authentication Tester scan by clicking **Start**.

The progress bar on the main window shows:

- Number of username-password pairs to be tested, against the number of pairs to try.
- Number of threads Authentication Tester is using in this scan.
- Status of the scan, if not in progress: paused, resumed, stopped, finished.

If you **Pause** a scan and leave Authentication Tester open (do not exit), you can resume the scan later by clicking **Resume**. If you exit Authentication Tester after pausing, the scan is deleted.

If you **Stop** a scan, the data collected so far will be deleted.

Scan results

When testing is complete, any username-password pairs that gained a successful login are listed in the table of Successful Logins.

You can save the results by exporting them in the form of an XML file. This feature is convenient for authentication testing during development, letting you compare the Authentication Tester results after implementing fixes.

Click **Export results** and give a name to the export file. An XML file is created, listing the credentials that successfully gained access.

Advanced configuration

The Advanced Configuration dialog box lets you customize the behavior of Authentication Tester for your local network and the application you are testing.

Tip: If you will be testing both Form and HTTP Authentication, you can enter all information at the same time. Only information relevant to the type of testing currently being done (Form or HTTP) is used.

Clicking **Advanced** in the main window opens the **Advanced Configuration** dialog box opens. It has four tabs:

- “General tab”
- “Form Authentication tab”
- “Proxy tab” on page 257
- “Credential Generation tab” on page 257

General tab:

The **General** tab of the **Advanced Configuration** window allows you to change network configurations.

Option	Description
Number of threads	Set the maximum number of connection threads Authentication Tester will test simultaneously. The default value is 12. More threads means a faster scan. If you find that Authentication Tester overloads your network you can reduce the number.
Request timeout (milliseconds)	Set the amount of time Authentication Tester should be given to reach the web application server before timing out the connection.
Internal proxy port	An available port is assigned automatically for the internal proxy. If necessary you can assign a different port. You will receive a warning if the port you select is unavailable.
HTTP Authentication over Form Authentication	If the web application uses HTTP authentication over form authentication, in order to test the <i>form</i> authentication you must select Enable, and enter valid values for the HTTP authentication username, password, and domain. This will allow Authentication Tester to access the Web application server to run the brute force tests on the Form Authentication mechanism. These values do not affect HTTP Authentication testing; they are applied only if you select the Form Authentication method.

Form Authentication tab:

This tab holds the Successful Login regular expressions that you enter to describe the page that is sent in response to an accepted username-password. (Information entered here applies only when Form Authentication is tested.)

Option	Description
<i>Successful Login Detection</i>	
Success Response	When selected, currently configured Success Responses are displayed in the pane below.
Error Response	When selected, currently configured Error Responses are displayed in the pane below.
Add/Remove	See “Describe the application's login responses” on page 253 for instructions on adding responses to the lists; or “About metacharacters” on page 254 to learn more about writing regular expressions.
<i>Default Counterfeit Credentials</i>	

Option	Description
Username/Password	These fields show the strings that you will be asked to enter when creating the login request to configure Form Authentication (see “Form authentication” on page 252). They do not need to be <i>valid</i> credentials as they are not used to actually log in to the site. They are simply used to identify to Authentication Tester the location of the credentials in the login request (for use in Brute Forcing the site). The default values are BruteUsername and BrutePassword. If client-side logic will not allow these values to be used in a login request to the site (for example, if the application requires an email as the username, and client-side logic enforces this rule when creating the login request), change these values to a valid format. When changing the default counterfeit credentials, make sure that neither value is a sub-string of the other. For example, if you enter user@email.com as the username, you may not use user as the password.

Proxy tab:

If your web application requires a web proxy connection, select the **Enable** checkbox on the **Proxy** tab and fill in the Server and Port. If the proxy requires authentication, fill in Username and Password, and Domain if required.

Credential Generation tab:

The options on this tab enable you to configure the usernames and passwords that Authentication Tester tries during the scan.

It has two areas:

- **Mode:** “Credential Generation: Mode area”
- **Configuration:** “Credential Generation: Configuration area”

Credential Generation: Mode area:

Procedure

Select which mode Authentication Tester will run:

- **Matrix:** Each username in the list is tried against every password in the list. Select this mode for a more comprehensive scan
- **Parallel:** Usernames and passwords are paired by index number. Select this mode for a faster scan

Credential Generation: Configuration area:

About this task

In this area you configure the options that Authentication Tester uses to generate the usernames and/or the passwords to be tested.

Procedure

1. Select one of the radio buttons to customize **Usernames** or **Passwords**.
All fields and data displayed in the Configuration area now apply to the selected item. For each item the drop-down list lets you configure the structure of the usernames/passwords that will be created and used in the attack.
2. From the **Element Type** drop-down list, select an element type that you want to be included in the attack, and in the field below it, fill in values for the element as described in “List types” on page 258.
3. Click **Add** to add the new definition to the list in the Configuration pane.
4. Repeat for additional elements as necessary.

5. Arrange the element types (using **Up** and **Down**) so that each username or password that is tried will be built of each type, in the order listed.

List types:

Element	Description	Sample Values
Dictionary	A "value-per-line" file used when generating usernames/passwords for the attack. Default files are provided for both Usernames and Passwords, but you can use the Browse button to browse to any other suitable file. The data folder also includes a much larger password dictionary file (passwords_long.txt) that can be used instead of the default file.	pathname to "value-per-line" dictionary file
Numeric	A range for numbers that will be included in the usernames/passwords generated	0 to 999999999
Constant	A string that you want included in all usernames/passwords that are generated	any string
Character Range	A range of characters and string length, for characters that will be included in the usernames/passwords that are generated	[space to tilde] ~ and a string length

Note: The **space to tilde** range includes a-z, A-Z, 0-9, and ASCII input symbols. If you enter " ~" (without the quotes) as a Character Range, Authentication Tester automatically inserts all included characters in the regular expression listing.

The valid value of the **length** field depends on the range. For example, if the range is 0-9 and the length is 10, the range is valid [0000000000, 0000000001, ...9999999999]; but if the range is a-z, Authentication Tester will not accept 10 as a valid length because the number of combinations would take an unreasonable amount of time and resources.

Note: To use a dash as a character, rather than to denote a range, precede it with a backslash: \-

Examples of Credential Generation configuration.:

Credential Generation	Resulting Credentials
Dictionary: C:\web tests\data\users.txt Constant value: _ Numbers range: 0 to 9	user_0 user_1 user_2 ... user_9
Constant value: passwd Character Range: a-zA-Z0-9; length: 3	passwdaaa passwdaab ... passwd999
Constant value: iamgod Character Range: ~; length: 1 Numbers range: 1900 to 3000	iamgod 1900 iamgod!1900 ... iamgod~3000

Deleting an element:

Procedure

Select the unwanted element in the lower pane, by clicking on it, and then click **Remove**.

Connection Test

The Connection Test PowerTool enables you to ping web sites without using the Ping protocol, which is blocked by many firewalls.

- You can run Connection Test from AppScan, by clicking **Tools > PowerTools > Connection Test**

- You can run Connection Test on its own, from the Windows Start menu, by clicking **All Programs > [AppScan Standard Installation Folder] > PowerTools > Connection Test**

Using Connection Test Procedure

1. In the **Web Site** text box, type or paste the URL.
2. Click **Ping Site**.

The Ping Results list shows whether Connection Test was able to successfully reach the URL and if so, how many milliseconds it needed to reach the URL.

What to do next

- To clear the Ping Results list, click **Clear**.

HTTP Method:

The **HTTP Method** drop-down menu offers two HTTP Request Methods.

- **HEAD:** Select to get only HTTP headers. This is the faster method.
- **GET:** Select to get entire page. Some sites do not allow HTTP requests using the Head method; in this case, try the Get method.

Port:

In the Port text box, type the listening port of the Web server.

The default for most non-secure URLs is **80**. You can change the port for specific URLs if required.

To use SSL (Secure Sockets Layer), check **Secure**. The default port changes to **443**.

No. of Intervals:

In the **No. of Intervals** text box, type the number of times Connection Test should ping the given URL.

- If you only want to test that you can reach the URL, use a small value (such as 3 to 5).
- If you need to monitor server crashes, the value should be much larger (such as 100 to 1000).
Connection Test works in the background while you carry on with other tasks, so you can make the number of intervals as large as you wish.

You can stop Connection Test at any time before the next ping by clicking **Stop**.

Interval Wait:

In the **Interval Wait** text box, type the number of milliseconds for Connection Test to wait between pings.

- If you want to test that you can reach the URL, the value could be 0 (zero).
- If you need to monitor server crashes, the value should be larger (such as 2000, which is two seconds).

Show Server Header:

Check the **Show Server Header** checkbox to see the type of Web server being pinged. The information is displayed after the final interval is reached.

Server header examples:

- www.ibm.com - Microsoft-IIS/6.0
- www.cnn.com - Apache
- www.sky-news.co.uk - Microsoft-IIS/5.0

- www.sourceforge.org - Apache/1.3.31 (Unix) PHP/4.3.11 mod_ssl/2.8.19 OpenSSL/0.9.7a

Encode/Decode

The Encode/Decode PowerTool encodes and decodes strings you put into it, to and from the format of your choice.

- You can run Encode/Decode from AppScan, by clicking **Tools > PowerTools > Encode/Decode**
- You can run Encode/Decode on its own, from the Windows Start menu, by clicking **All Programs > [AppScan Standard installation folder] > PowerTools > Encode/Decode**

Using Encode/Decode Procedure

1. In the Input text box, type or paste the text.
2. From the Method drop-down menu, select an encoding/decoding method. (See “Methods”)

If you choose 3DES for the method, type or paste a suitable encryption key in the Encryption Key text box.
3. Click Encode or Decode.

The encoded or decoded text appears in the Output box.

Layered Encoding/Decoding: You can send output back as input and encode or decode again, by clicking Send Back.

This is useful if you want to:

- Ensure encoding is 1-to-1 with decoding
- Perform multiple encoding on a string
- Decode a string that has been multiply encrypted

Methods:

Method	Capability	Translate between
URL	encode + decode	text <-> URL encoding
Base64	encode + decode	text <-> Base64 encoding
Overlong UTF-8	encode + decode	text <-> Overlong UTF-8 (2 bytes)
UU	encode + decode	text <-> UU encoding
HTML	encode + decode	text <-> HTML entities
MD5	encode	text <-> digital signature
SHA1	encode	text <-> digital signature
SHA256	encode	text <-> digital signature
SHA384	encode	text <-> digital signature
SHA512	encode	text <-> digital signature
3DES (192bit key)	encode + decode	readable text <-> encrypted text (using a given key)

Expression Test

Writing precise regular expressions can be a tedious trial-and-error process. You can use the Expression Test PowerTool to help accelerate the process.

- You can run Expression Test from AppScan, by clicking **Tools > PowerTools > Expression Test**
- You can run Expression Test on its own, from the Windows Start menu, by clicking **All Programs > [AppScan Standard installation folder] > PowerTools > Expression Test**

Using Expression Test Procedure

1. In the **Text** field, type or paste text that you want the regexp to find.
2. In the **Regular Expression** field, type the regular expression that you believe will find the text.
3. Click **Test**.

Results

Text in the Text box that matches the pattern is highlighted in red.

Text replacements

Replacing text that matches a specific pattern.

About this task

If you are using regular expressions to replace text that matches a specific pattern, you can use Expression Test to test that the replacement will be done correctly.

Procedure

1. In the **Regular Expression** field, type the regular expression.
2. In the **Replace** box, type the text that should replace the text that matches the pattern of the regular expression.
3. In the **Text** box, type or paste example text.
4. Click **Test** and check that the predicted text is matched by the regular expression.
5. Click **Replace**.

Results

The text that matched the regular expression pattern (highlighted in red) is replaced with the text in the **Replace** box.

Pattern groups

Applying a regular expression to a group of characters or metacharacters.

Often you need to apply a regular expression to a group of characters or metacharacters.

To group a set, put the characters in parentheses ().

The group is given a number in the Match Groups box and the matching text is displayed next to it.

If a match occurs, you can see which text matches which group in the Match Groups box.

For example, you have an HTML code with `<UL>first` in the text.

You type `(<.L>)([a-z]*)` as the regular expression.

In the Text field, first is highlighted in red.

In **Match Groups**, the groups are broken down:

- Group 1
- Group 2 first

Metacharacters

General information about metacharacters.

Any single character (letter, digit, or symbol) in a regular expression is matched to itself, literally; unless it is a metacharacter. A metacharacter is one or more characters that have a unique meaning and are not used literally in the regular expression match.

For example, the circumflex character (^) is a metacharacter that means "search at the beginning".

If you want to find the character, rather than the metacharacter pattern, put a backslash (\) before it.

For example, to find the circumflex as a text character, the regular expression must be: \^

Reg. Exp.	Description	Example
\	Find the next character as a character; do not use its metacharacter pattern.	\. finds a period (.) in the text . finds the first character (any character)
^	Find at the beginning of a string	^1 finds "1. Click Save." but not: "in the 210th line"
.	Find any character (except newline).	Finds a, A, 1, <, ., =, and so on; whatever the first character is.
()	Find a pattern group.	(word) finds "In this word " ^(word) finds " Words in this line"
[]	Find a pattern range.	[a-z] finds letters, but not numbers
*	Find the previous pattern zero or more times.	.* finds all characters <(.*)> finds all HTML tags
+	Finds the previous pattern one or more times.	(<.l>)+ finds
?	Finds the previous pattern zero or one time.	(<.l>)? finds
(?i)	Find the next characters with a case-insensitive search.	(?i)word finds word and Word

HTTP Request Editor

The HTTP Request Editor PowerTool enables you to send a fully-controlled HTTP request to your site, to test how your site responds to different kinds of HTTP request.

- You can run HTTP Request Editor from AppScan, by clicking **Tools > PowerTools > HTTP Request Editor**
- You can run HTTP Request Editor on its own, from the Windows Start menu, by clicking **All Programs > [AppScan Standard installation folder] > PowerTools > HTTP Request Editor**

Using HTTP Request Editor Procedure

1. In the **Request** tab, fill in the relevant fields.
2. Click **Send**.
3. Open the **Response** tab to see how the host dealt with the request.

Request tab

View options: The View options in the Request tab enable you to create an HTTP request in different ways:

- **Raw:** Create the request by typing or pasting text.
- **Parsed:** Create the request by filling in the form.

To update the raw HTTP request with details entered in the Parsed view, on the Raw tab, click **Update from Parsed**.

The Request that you set up by filling out the parsed form is displayed in raw form.

Request settings:

Setting	Description
Host	Type in the IP address or hostname of your website.
Port	Type in the TCP port on which your web server is listening. Default=80
Method	Select a method from the drop-down menu, or enter a value for the method request parameter. The default drop-down menu includes: GET: Retrieve all information identified by the Request-URI. POST: Request that origin server accept an entity enclosed in the request as a new subordinate of the resource identified by the Request-URI in the Request-Line. The actual function performed by this method is determined by the server and is usually dependent on the Request-URI. HEAD: Retrieve information identified by the Request-URI, but the server must not return a message-body in response. Meta-information in the HTTP headers received in response to a HEAD request are the same as those received in response to a GET request. Use to obtain information without transferring the entire body; often used to test hypertext links for validity, accessibility, and modifications.
Secure	Check to send request using HTTPS.
URL	Enter or paste a URL. You can view the list of links extracted from the last response in the Response Tab.
HTTP Version	Enter the version number of the HTTP protocol that the request should use. Default=1.1
Auto Content Length	When checked, HTTP Request Editor will automatically calculate and add the Content-Length header and its correct value.

Message details: Create or modify parameters, headers, cookies, and their values, to include in the request.

Parameters:

Adding parameters:

Procedure

1. Click **Add**.

The Add Parameter dialog box opens.

2. Enter a name for the parameter and then enter a value.
3. Select the type:
 - **Body:** Send the parameter in the body of the request.
 - **Query:** Send the parameter in the query part of the request.

4. Click **OK**.

The parameter is added to the request.

In the Parsed view, the Parameters table shows the name, value, and location that you entered.

In the Raw view, the parameter appears differently, depending on the location that you chose:

- **Body:** a name=value pair will appear in the body of the request.
- **Query:** a name=value pair will appear in the query part of the request (after the ? symbol)

In both cases, the name=value pairs will be concatenated with & if more than one pair exists.

Editing parameters:

Procedure

Do one of the following:

- In the Parsed view: select the parameter and click **Edit**.
- In the Raw view: edit the text.

Removing parameters:

About this task

You can remove parameters from the request without deleting them from HTTP Request Editor.

Procedure

In the Parsed view, uncheck the parameter name.

Deleting parameters:

Procedure

Do one of the following:

- In the Parsed view, select the parameter and click **Remove**.
- In the Raw view, delete the text.

Headers:

Adding headers:

Procedure

1. Click **Add**.

The Add Header dialog box opens.

2. Enter a name and a value for the new HTTP header.

3. Click **OK**.

The header is added to the request.

- In Parsed view, the Headers table shows the name/value of the new HTTP header.
- In Raw view, the header appears as: Header name: Header value

Editing headers:

Procedure

Do one of the following:

- In Parsed view, select the header name and click **Edit**.
- In Raw view, locate the header you want to edit and modify its text.

Removing headers:

About this task

You can remove headers from the request without deleting them from HTTP Request Editor.

Procedure

In Parsed view, clear the checkbox next to the header name.

The HTTP Request Editor generates a request with several default headers (Accept, Host, User Agent).

You can remove or modify these headers just like any other header.

Deleting headers:

Procedure

Do one of the following:

- In Parsed view, select the header name and click **Remove**.
- In Raw view, locate the HTTP header you want to delete and remove the text from the request.

Cookies:

Adding cookies:

Procedure

1. Click **Add**.

The Add Cookie dialog box opens.

2. Enter a name for the cookie and then enter a value.

3. Click **OK**.

The cookie is added to the request.

- In Parsed view, the Cookies table shows the name and value that you entered.
- In Raw view, the cookie appears in the header: `Cookie: name=value`

Editing cookies:

Procedure

Do one of the following:

- In Parsed view: select the cookie and click **Edit**.
- In Raw view: locate the cookie header you want to change and edit the text.

Removing cookies:

About this task

You can remove cookies from the request without deleting them from HTTP Request Editor.

Procedure

In Parsed view, clear the checkbox of the cookie name.

Deleting cookies:

Procedure

Do one of the following:

- In Parsed view: select the cookie and click **Remove**.
- In Raw view: locate the cookie header you want to delete and remove the text.

Response Tab

After you click **Send** in the Request tab, you will get a response from the web server.

There are three ways to view a response:

- **Raw:** Response is presented in raw text
- **Parsed:** List of links extracted from the HTML in the response, and SSL information
- **Browser:** Response is shown in a web browser

Note:

Only the actual response that was sent by the web server is presented in the browser. Redirections, images, and CSS are not requested automatically.

In raw view, you have the option to search for text in the response. In the text box at the bottom of the window, enter a regular expression (for example: "[a-z0-9]@" to find an email) and then click **Search**.

Check **Case Insensitive** to make the pattern match both upper- and lower-case letters.

Generic Service Client (GSC)

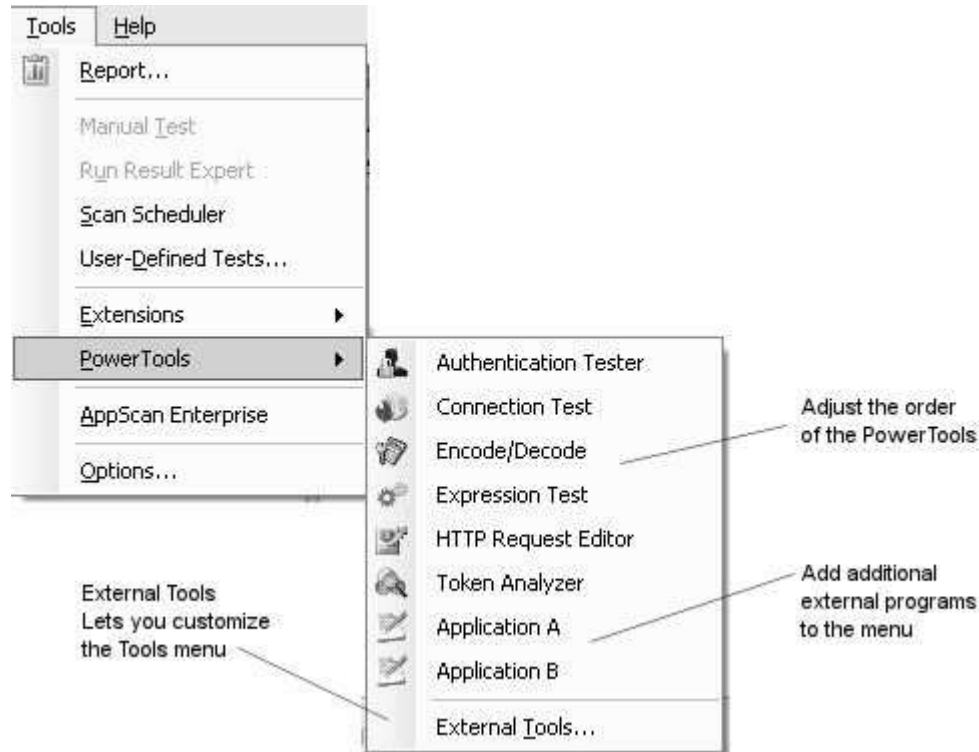
Generic Service Client (GSC) provides a simple interface that displays the services available, and lets you input parameters and view the results. Use it to explore your SOAP web service manually, so that AppScan can use your input to create appropriate tests.

See: "Using GSC" on page 134

Customizing the Tools menu

You can customize the Tools menu to:

- “Adjust the order of the PowerTools”
- “Add programs to the Tools menu”



Adjust the order of the PowerTools

Procedure

1. Click **Tools > PowerTools > External Tools**
The External Tools dialog box opens.
2. Select a menu item from the list, and click **Move Up** or **Move Down** as required.

Add programs to the Tools menu

About this task

You can add a link to an external program that you frequently use a particular program while running AppScan, you can add a link to the Tools menu so that you can open it from within AppScan.

Procedure

1. Click **Tools > PowerTools > External Tools**
The External Tools dialog box opens.
2. Click **Add**
The Create a New External Tool dialog box opens.
3. In the **Title** field, type in the name you want to appear on the **Tools** menu.
4. Click the **Browse** button (...), locate the EXE file for the program, and click **Open**.
5. Click **OK**.

The title and path of the EXE file are added to the External Tools list.

6. To adjust its position in the menu, select the new item and click **Move Up** or **Move Down** as required.

Extensions

With AppScan it is easy to write and integrate extensions. Extensions are add-ons that add functionality to AppScan. They can be anything from a small email notification utility, to a full blown vulnerability exploit tool. Extensions enable users to customize AppScan to their own needs. They are written using AppScan's SDK and extensions framework.

Further information on how extensions can be used with AppScan, and where to download/host extensions, is available at: http://www.ibm.com/developerworks/rational/downloads/08/appscan_ext_framework/

The SDK online help file **AppScanSDK.chm** can be found in the main AppScan documentation folder.

AppScan comes with the Pyscan extension, and some others, built-in. It is easy to integrate and manage additional extensions using the Extension Manager.

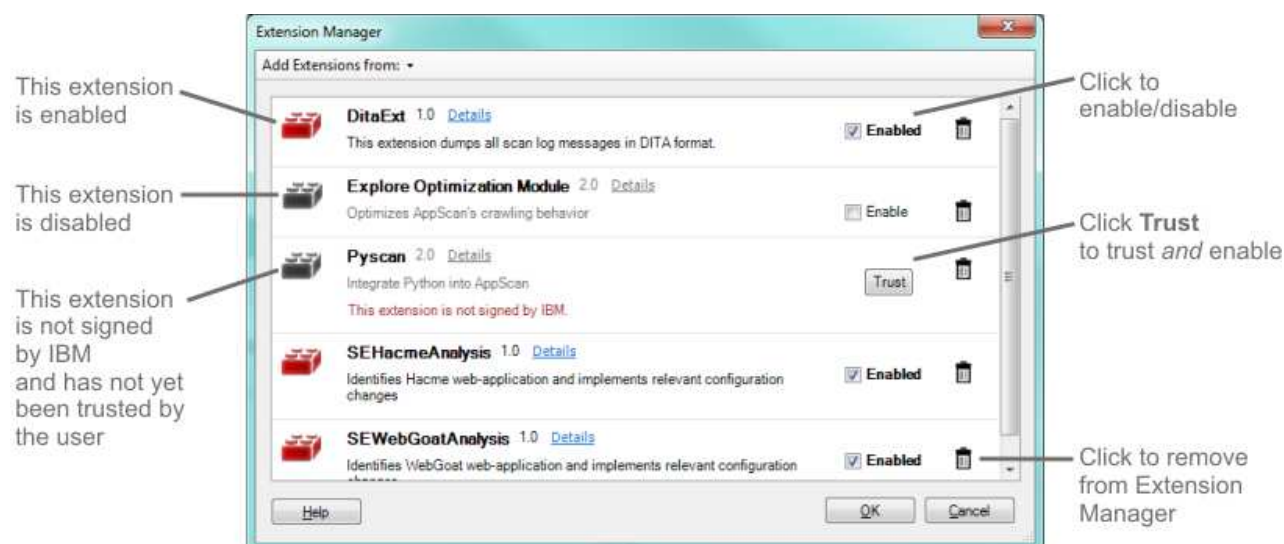
Extension Manager

Extension Manager lets you add/remove, enable/disable extensions for use with AppScan

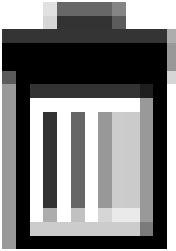
Extension Manager (**Tools > Extensions > Extension Manager**) lists all extensions that have been added.

- You can add extensions that are available from IBM Security, as well as your own extensions.
- You can enable and disable extensions that are added.
- You can elect to "Trust" unsigned extensions (for example, extensions you created).

If an extension is added but not trusted and enabled, the icon on the left is grayed out.



Option	Explanation
Add Extensions from:	To add a new extension, click and select the source: • This Computer, or • AppScan eXtensions Framework Page

Option	Explanation
Enabled check box	Select/deselect the check box to enable/disable the extension. Note that this change only takes effect the next time you open AppScan. When a new extension is added it is automatically enabled (and this check box selected), unless there is a reason AppScan was unable to do this (such as an incompatible version), or the extensions is not signed.
Trust button	If an unsigned extension is installed, AppScan does not load it, but displays a Trust button next to it. When you open AppScan a pop-up warns you that there are unsigned extensions that are not enabled. If you are sure the extension can be trusted (for example, if you wrote it yourself), click the Trust button to trust it from now on, and enable it. Tip: If you have an unsigned IBM extension that you used with a previous version of AppScan, you can either elect to trust it, or click Get More Extensions to see if a signed version is available to replace it (see “Replacing unsigned extensions” on page 314).
	Click to remove an extension.

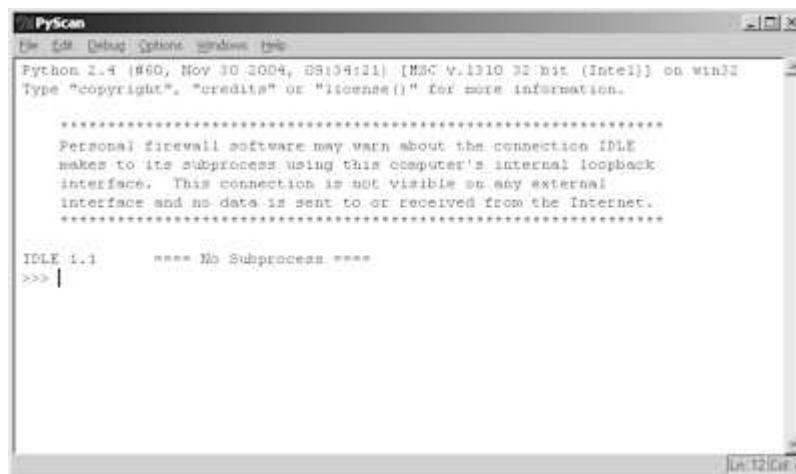
Pyscan

About this task

Pyscan is a Python extension that enables you to control AppScan using the Python interface. Pyscan is installed and activated during the standard AppScan installation.

Procedure

On the **Tools** toolbar, click **Extensions > Pyscan**.
The Pyscan interface opens.



Explore Optimization module

This extension helps improve scan efficiency by eliminating unnecessary URLs from the Explore results.

The extension can particularly improve performance in cases where URL rewriting is used by the site, inflating the Explore stage of the scan with duplicate URLs that appear different due only to built-in parameters.

The Explore Optimization module identifies URLs that are rewritten to include parameters. For example, if URLs such as those below occur a large number of times, AppScan would create a folder for each one where in fact these are actually parameters that have been rewritten into the URL for convenience.

```
http://mysite.com/2010/10/01/  
http://mysite.com/2010/10/02/  
http://mysite.com/2010/11/01/  
http://mysite.com/2010/11/02/  
http://mysite.com/2010/12/01/  
http://mysite.com/2010/12/02/
```

A case like this could result in thousands of unnecessary URLs, or a never-ending Explore stage.

When it encounters a large number of URLs that seem related, the module analyzes them. It then creates a custom parameter (in this case it would be `/[year]/[month]/[day]`), clears the Explore results, and runs a new Explore stage, which may have far fewer URLs to be tested in the Test stage. This can greatly improve performance.

Note: An additional functionality of this module, that can be useful in maximizing scan efficiency, is **Run navigational parameter detection** (see “Configuring Explore Optimization” on page 272 for details).

Running the module automatically

It is suggested to run the module during scans *whenever the number of links discovered in the Explore stage (including unvisited links) reaches the configured threshold*. This threshold (“Minimum links to start module”), is generally at least 1,000.

Note: If you start a scan from the command line interface (CLI), Explore Optimization does *not* run automatically even if configured.

Note: If your scan configuration includes Manual Explore data or a multi-step operation, Explore Optimization does *not* run automatically even if configured, though you can run it manually (Tools > Extensions > Explore Optimization Module > Run).

To change the configuration:

1. Click **Tools > Extensions > Explore Optimization: Configure**.
Explore Optimization Configuration dialog box opens.
2. Select the **Run Explore Optimizer automatically during scans** check box.
3. Click **OK**.

Running the module manually

You can run the module manually on any Explore results, even if the number of URLs discovered is *less* than the configured threshold (“Minimum links to start module”). However the module is most useful when the number of URLs is at least several hundred.

To run Explore Optimization manually:

- Click **Tools > Extensions > Explore Optimization: Run**.

The module begins to Explore, starting at the configured Starting URL. When the Explore stage is complete it analyzes the results. If optimizations are discovered, it will give you the option of saving the first set of results before clearing them and re-Exploring.

See also:

“Redundancy tuning” on page 73

“Redundancy tuning defaults” on page 74

Working with Explore Optimization

This section describes running a scan with Explore Optimization active.

About this task

If your site rewrites parameters into its URLs, or if initial attempts to explore the site, with Explore Optimization disabled, result in very large numbers of URLs or unending scans, the Explore Optimization module can help to reduce the scan to manageable proportions. In addition, activating the Run navigational parameter detection option may help.

Procedure

1. Configure the Starting URL and any other necessary settings as for a regular scan.
2. Open **Tools > Extensions > Explore Optimization: Configure**, and select the **Run Explore Optimizer automatically during scans** check box.

Note: If your scan configuration includes Manual Explore data or a multi-step operation, Explore Optimization does *not* run automatically even if configured, though you can run it manually (Tools > Extensions > Explore Optimization Module > Run).

3. In the Scan configuration area, optionally change the **Run navigational parameter detection** setting to **True**.
4. Click **OK**.
5. Start a scan (**Scan > Full Scan**).

If the number of URLs found in the Explore stage (including unexplored URLs) reaches the threshold defined in **Minimum links to start module** (default: 1000), the Explore stage is paused and the Explore Optimization module starts a *phase*, consisting of two (main) stages:

Identify navigational parameters (if configured)

The module looks for navigational parameters, using the defined Names and Values (configured in Tools > Extensions > Explore Optimization: Configure). If it succeeds in identifying navigational parameters, it then:

- a. Defines them in the list of parameters (Scan Configuration > Parameters and Cookies > main tab)
- b. Sets their Redundancy Tuning to the strictest level
- c. Decreases the *default* Redundancy Tuning (for non-navigational parameters) to a lower level (see “Redundancy tuning defaults” on page 74)

Identify URL rewriting

The module looks for parameters written into the URL. If it finds such parameters, it defines them in the list of custom parameters (Scan Configuration > Parameters and Cookies > Advanced: Custom Parameters tab).

6. At the end of this *phase* of Explore Optimization:
 - If any configuration changes have been made, the existing Explore data is cleared and a new Explore stage is run. (If you started the module manually, you are given the option of saving the current data before the new Explore stage is run.)

- If no configuration changes have been made, a new *phase* of Explore Optimization is run, using a higher threshold (not user-configurable), in an attempt to gather enough data to identify parameters and reduce the Explore data to a reasonable size.
7. After the module has successfully run (with one or more phases, and one or more Re-Explore stages), the scan resumes and finishes.
 8. When the scan is complete, review the results, looking for the following indications that it has been successful:
 - The status bar, at the bottom left of the screen, indicates how many tests were created and how many sent. All tests should have been sent.
 - The application tree should be complete, indicating that all the important parts of the site were visited.
 - Review the navigational parameters that were added to see that all important parameters were fully tracked.
 - Review the custom parameters that were added to see that they correctly represent the way the site writes its parameters into its URLs.

Configuring Explore Optimization

This dialog box is used to activate and configure the Explore Optimization module.

Apart from activating the module, in most cases you do not need to change any other configurations. Some should not be changed at all without guidance from Support, as indicated below.

This dialog box is opened from **Tools > Extensions > Explore Optimization: Configure**.

Note: If you make changes to these settings and then create a new scan, all settings will revert to their defaults, except the check box setting.

Name	Description
<i>Checkbox</i>	
Run Explore Optimizer automatically during scans	When selected the module runs automatically when you run either Explore Only or a Full Scan, whenever the "Minimum links to start" limit (below) is reached. Important: This setting applies to <i>all</i> scans. Other settings in this dialog box apply to the current scan only. Default: Cleared Note: If you start a scan from the command line interface (CLI), Explore Optimization does <i>not</i> run automatically even if configured. Note: If your scan configuration includes Manual Explore data or a multi-step operation, Explore Optimization does <i>not</i> run automatically even if configured, though you can run it manually (Tools > Extensions > Explore Optimization Module > Run).
<i>Scan Configuration</i>	

Name	Description
Anchor rewrite rules to the right	Consider URLs such as these: <pre>http://...php/1/index http://...php/2/index http://...php/3/index</pre> When the module creates the custom parameter: If this is set to False, the parameter created will be: <pre>php/([^\./]+)</pre> If this is set to True, the parameter created will be: <pre>php/([^\./]+)/index</pre> Default: False
Confidence margin percentage	Change this only if instructed by Support.
Delimiters used in URL rewriting	Add any custom delimiters that your application uses.
Maximum depth for start of line pairs	Change this only if instructed by Support.
Maximum phase time (mins)	Maximum time for a phase of Explore Optimization to run.
Merge rewrite rules to the left	Change this only if instructed by Support.
Minimum links to start module	Minimum number of links in Explore stage data (including unvisited URLs) needed for the Explore Optimization module to start automatically, if the Run Explore Optimizer automatically during scans check box is selected. Note: Even if this setting is changed, the notice suggesting that you activate the module appears when 1,000 links are reached in the Explore stage. Default: 1,000
Name/Value pair delimiters	Specify any custom delimiters used by your application.
Navigational parameter names	Partial matches for navigational parameter names, separated by single spaces.
Navigational parameter values	Partial matches for navigational parameter value patterns, separated by single spaces.
Remove unused delimiters	Change this only if instructed by Support. Default: True
Run navigational parameter detection	When True, the module attempts to identify navigational parameters by name or value. It sets the Redundancy Tuning configuration for these parameters to the strictest level, <i>and reduces the default setting that is applied to all other parameters</i>. This enables AppScan to test navigational parameters <i>more</i> thoroughly than before, while safely treating all non-navigational parameters <i>less</i> thoroughly. Setting this value to True can greatly improve both accuracy and performance, however you should carefully review the results to make sure that scan coverage was not affected. Default: False

Name	Description
Switch complexity limit	If a specific folder contains more subfolders than this limit, AppScan will assume that the subfolders are dynamic parameter values and do not require individual scanning. Note: The value entered here is also used as a threshold value for running Explore Optimization manually. If less links have been discovered than the amount entered here, the module will not run. Default: 20

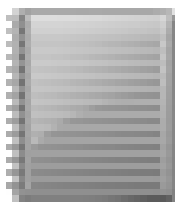
Logs

Logs can help you troubleshooting.

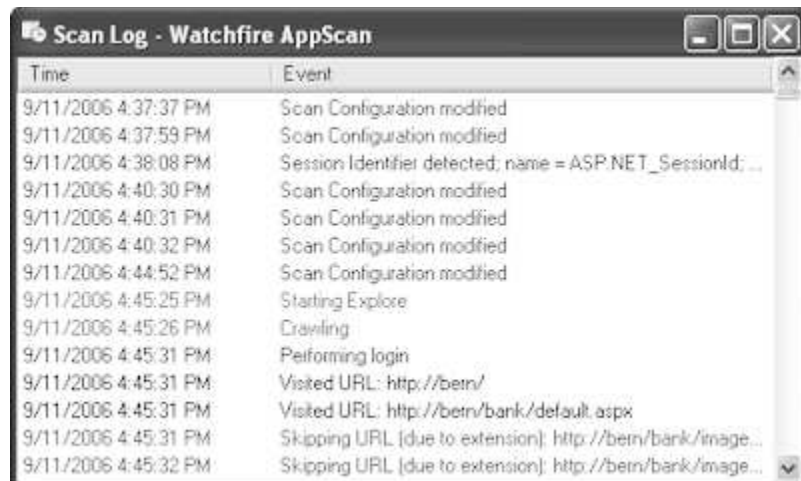
Scan Log

This log lists the actions performed by AppScan during the current scan.

When a scan runs the Scan Log is updated live, so you can see what AppScan is doing at any moment.



- On the toolbar, click the **Scan Log** icon



The scan log is saved as part of the scan. When a saved scan is loaded, the existing scan log loads too, and data is added to it as scanning continues.

Note: The Scan Log window does not show previous data saved in the scan log. This data can be found only in the actual log file. To view it, open the scan file (.SCAN) using a program that opens compressed files, look for ScanLog.log, and open it using a text viewer.

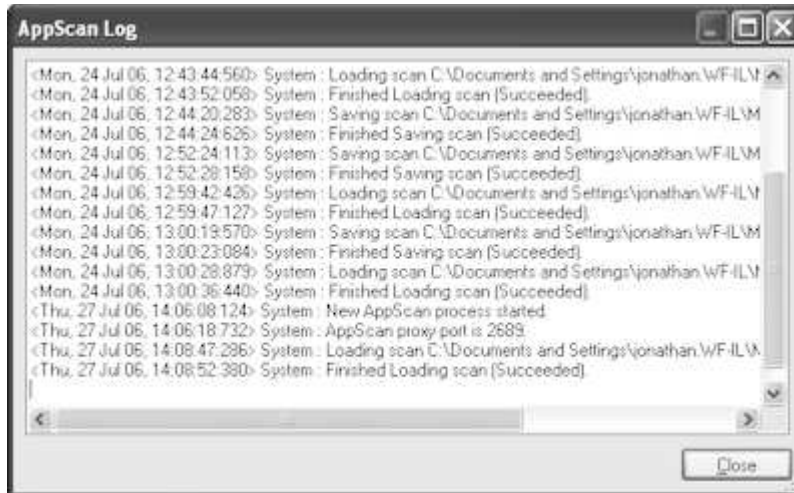
You can customize the scan log, defining exactly which items are included in it, and control their color in the Scan Log window (see “Scan Options tab” on page 235).

Where relevant, explanations of the messages, and suggested user responses are described in “Scan Log messages” on page 314

AppScan Log

This log lists errors, connections and AppScan system messages (general application events and warnings).

- To view the AppScan Log within AppScan, click **Help > AppScan Log**.



- To see where this log is saved to, click **Tools > Options > General tab > Log files folder**.

In Windows 7 the default location is:

...\AppData\Roaming\IBM\AppScan Standard\Logs

Where relevant, explanations of the messages, and suggested user responses are described in “AppScan Log messages” on page 323

Update Log

This log lists all updates to the program since installation.

- To view the Update log click **Help > Update Log**.



- To see where this log is saved to, click **Tools > Options > General tab > Log files folder**.

In Windows 7 the default location is:

...\\AppData\\Roaming\\IBM\\AppScan Standard\\Logs

See also:

“Updates” on page 11

Traffic Log

This log lists the requests and responses between AppScan and the site during scanning, and can be useful when troubleshooting.

The Traffic Log lists all requests responses from the application during the scan. By default AppScan does not save a traffic log, but you can enable it if needed by Support to help give you assistance.

Turning on the log may affect performance, so it should only be enabled if you need it.

- To enable the Traffic Log, click **Tools > Options**, and select **Enable request/response logging**.
- To see where this log is saved to, click **Tools > Options > General tab > Log files folder**.

In Windows 7 the default location is:

...\\AppData\\Roaming\\IBM\\AppScan Standard\\Logs

Searching Results

You can filter the Result List in any of the views, for specific data.

Procedure

1. On the toolbar click **Find**, or press **Ctrl + F**.
The **Find** bar appears at the bottom of the screen.
2. On the **Application Tree**, select a node.
 - If you select the **My Application** node, Find searches through all the results.
 - If you select a node within the tree, Find searches through the selected node and its sub-nodes.
3. In the **Look for** text box of the **Find** bar, enter a string or partial string.
4. In the **Show** list, click a data type.
5. Click **Find Now** or press **Enter**.

Results appear in the **Result List**, overwriting the list that was previously displayed, and the counters update to show the new number of items in each category. Tabs without results are grayed out.

If you enter another **Find** string and click **Find Now** again, the new search is done on the selected node in the **Application Tree**, not on the displayed results of the previous search.

Note: To revert to the full, unfiltered Result List, click **Clear**.

Chapter 12. Integrations

This section describes integrations of other applications with AppScan Standard.

AppScan Enterprise

This section describes ways AppScan Standard and Enterprise editions can interact.

AppScan Enterprise edition provides centralized scanning and user-access controls, remediation capabilities, executive dashboards, compliance reporting and seamless integration with AppScan Standard. If your organization uses AppScan Enterprise, you can:

- Import AppScan Enterprise user permissions to use *in addition* to those allowed by the local AppScan Standard license.
- Export ("Publish") AppScan Standard scan results to work with in AppScan Enterprise.
- Create an AppScan Enterprise job.

Importing AppScan Enterprise license permissions

Configure AppScan Standard to scan sites allowed by your AppScan Enterprise license.

About this task

If your organization has an AppScan Enterprise license that allows scanning additional sites to those allowed by your local AppScan Standard license, you can import these permissions to use on your local machine *in addition* to your existing license. You will then be able to scan any URL that is allowed by one or both of the licenses.

Note: This option is available only when a full AppScan Standard license (not a demo license) is loaded.

Procedure

1. Click **Help > License**.
The License dialog-box opens.
2. Click **Add AppScan Enterprise License**.
The AppScan Enterprise dialog-box opens.
3. Select the **AppScan Enterprise** check box.
The fields become active.
4. Select the check box, and complete the User Name, Password, Domain and URL for the AppScan Enterprise server.
5. (Optional:) Verify that you are connected to the network, and click the **Test these settings** button.
AppScan establishes contact with the AppScan Enterprise Server and the organization's license is displayed in the lower pane.

Note: In the event that the local client fails to make contact with the Enterprise server, you are notified that scanning is restricted to IPs allowed by the local license.

6. Click **OK**.
The AppScan Enterprise permissions are loaded to the AppScan License, in addition to its own permissions.

Publishing to AppScan Enterprise

You can publish AppScan Standard results to AppScan Enterprise, to work with them there.

Procedure

1. Open the scan whose results you want to export.
2. Click **File > Export > Publish to AppScan Enterprise**.

The Login dialog-box opens.

3. Configure your AppScan Enterprise sign-in information:

To sign in with a User ID and Password:

- a. Select **Log in with User ID and Password**.
- b. In the URL field:
 - **AppScan Enterprise 9.0.3.1 and higher:** Enter the AppScan Enterprise server's REST service URL.
Format: `https://[AppScan Enterprise Server]:[Server's REST port]/ase`
 - **AppScan Enterprise 9.0.3 and lower:** Enter the AppScan Enterprise server's SOAP service URL.
Format: `http://[AppScan Enterprise Server]/ase/desktop/permissions.asmx`
- c. Enter a valid User ID (with the format [domain name]\[username]) and Password.
- d. Click **Login**.

To sign in using a client-side certificate or smart card:

- a. Select **Log in using Client-Side Certificate / Smart Card**.
- b. In the URL field:
 - **AppScan Enterprise 9.0.3.1 and higher:** Enter the AppScan Enterprise server's REST service URL.
Format: `https://[AppScan Enterprise Server]:[Server's REST port]/ase`
 - **AppScan Enterprise 9.0.3 and lower:** Enter the AppScan Enterprise server's SOAP service URL.
Format: `http://[AppScan Enterprise Server]/ase/desktop/permissions.asmx`
- c. Select the check box for the certificate needed.
- d. Click **Login**.

Note: If a Smart Card PIN code is needed to log in, a dialog box opens for you to enter it.

The Publish Results dialog-box opens.

4. Define the Job Name and optionally AppScan Enterprise Folder and Application.

Note:

- Folder selection applies to SOAP only and is supported from AppScan Enterprise version 8.7. If you do not select a folder the default AppScan Enterprise folder is used. For REST services folder selection does not apply, and results are saved to the application you select.
- Application selection is supported from AppScan Enterprise version 9.0.
- The Select Application dialog box includes a **Create a new application on server** option if your permissions allow this.

5. Click **Publish**.

When the process is complete a green success message appears in the dialog box. The job can now be opened and worked within AppScan Enterprise.

Creating a job in AppScan Enterprise

You can create an AppScan Enterprise job using your AppScan Standard configuration.

About this task

Scan configurations can be exported to AppScan Enterprise as new jobs to work with them there.

Note: Security AppScan Enterprise Version 9.0 or later is required.

Procedure

1. Open the scan whose configuration you want to use.
2. Click **File > Export > Create Job in AppScan Enterprise**.
The Login dialog-box opens.
3. Configure your AppScan Enterprise sign-in information:
To sign in with a User ID and Password:
 - a. Select **Log in with User ID and Password**.
 - b. In the URL field:
 - **AppScan Enterprise 9.0.3.1 and higher:** Enter the AppScan Enterprise server's REST service URL.
Format: `https://[AppScan Enterprise Server]:[Server's REST port]/ase`
 - **AppScan Enterprise 9.0.3 and lower:** Enter the AppScan Enterprise server's SOAP service URL.
Format: `http://[AppScan Enterprise Server]/ase/desktop/permissions.asmx`
 - c. Enter a valid User ID (with the format [domain name]\[username]) and Password.
 - d. Click **Login**.**To sign in using a client-side certificate or smart card:**
 - a. Select **Log in using Client-Side Certificate / Smart Card**.
 - b. In the URL field:
 - **AppScan Enterprise 9.0.3.1 and higher:** Enter the AppScan Enterprise server's REST service URL.
Format: `https://[AppScan Enterprise Server]:[Server's REST port]/ase`
 - **AppScan Enterprise 9.0.3 and lower:** Enter the AppScan Enterprise server's SOAP service URL.
Format: `http://[AppScan Enterprise Server]/ase/desktop/permissions.asmx`
 - c. Select the check box for the certificate needed.
 - d. Click **Login**.

Note: If a Smart Card PIN code is needed to log in, a dialog box opens for you to enter it.
The Create Job dialog-box opens.

4. Define the Job Name and optionally AppScan Enterprise Folder and Application.

Note: Folder selection is supported from AppScan Enterprise version 8.7, and Application selection from AppScan Enterprise version 9.0. Scan template folders are not shown. If you do not select a folder the default AppScan Enterprise folder is used.

Note: The Select Application dialog box includes a **Create a new application on server** option if your permissions allow this.

5. If your configuration includes Manual Explore data, you are given the option to:
 - **Continue Full Scan:** AppScan will run an Automatic Explore stage and then test all Explore data (both Manual and Automatic)
 - **Test Only:** Only the existing Manual Explore data will be tested
6. Click **Create**.

When the process is complete a green success message appears in the dialog box. The results can now be opened and worked with in AppScan Enterprise.

Creating a scan template in AppScan Enterprise

You can create an AppScan Enterprise template using your AppScan Standard configuration.

About this task

Scan configurations can be exported to AppScan Enterprise as templates to work with them there.

Note: Security AppScan Enterprise Version 9.0 or later is required.

Procedure

1. Open the scan whose configuration you want to use.
2. Click **File > Export > Create Scan Template in AppScan Enterprise**.
The Create Template dialog-box opens.
3. Configure your AppScan Enterprise sign-in information:
 - a. Enter a valid User ID (with the format [domain name]\[username]) and Password.
 - b. In the URL field, enter the AppScan Enterprise server's URL with format:
`http://[AppScan Enterprise Server]/ase/desktop/permissions.asmx`
 - c. Click **Test these settings**.
The Create Job dialog-box opens.
4. Define the Template Name and select AppScan Enterprise Folder and Application.

Note: Folder selection is supported from AppScan Enterprise version 8.7, and Application selection from AppScan Enterprise version 9.0. Only template folders are shown. If you do not select a folder the default AppScan Enterprise folder is used.

Note: The Select Application dialog box includes a **Create a new application on server** option if your permissions allow this.

5. Click **Create**.
When the process is complete a green success message appears in the dialog box. The template can now be opened and worked with in AppScan Enterprise.

Automation Frameworks

You can use scripts written for your QA automation framework (such as Selenium) to create Manual Explore recordings for an AppScan scan.

If your QA personnel use an automation framework to run functional tests on your web app, you can take advantage of the scripts that are already written to create tailor-made scans. The requests from the automation framework to the app are sent with AppScan as proxy, enabling AppScan to record the actions as an Explore stage for its own scan. AppScan then tests the site based on that Explore stage. This is done using the AppScan CLI.

In principle you create and run a batch command that does the following:

1. Open AppScan and configure:
 - a. Starting URL
 - b. User credentials
 - c. Test Only
 - d. Open AppScan's proxy with a specific listening port
2. Run the automation framework script through the same port.
3. When the script ends, close AppScan's proxy and the Test stage starts.

4. Save the scan results, and optionally create and save a report.

The section following describes how to test the AppScan demo test site using Selenium, but the process can easily be adapted for any site and any automation framework.

Creating the batch command

This example runs a scan using a Selenium script for the Explore stage. The same principles can be applied to any other automation framework.

About this task

A Selenium script is assumed, and port 56232 is used for communication between Selenium and AppScan Standard. You could of course change these as needed.

Note: In each code sample, the code added in that step is highlighted in bold.

If you want to work through this example yourself, a folder containing the file used here can be found in: [AppScan Standard installation folder]\Docs\Selenium Example

Note: If the path to your AppScan folder is different, you will need to change the path in the JAR file.

Procedure

1. Create a TXT file and open it in a text editor.
2. Type in the AppScan command to open AppScan and define the Starting URL, login credentials, Test Only option, open proxy, and proxy listening port:

```
START "" "[AppScan Standard installation folder]\AppScanCMD.exe"  
/starting_url "https://demo.testfire.net"  
/credentials "jsmith:demo1234"  
/test_only  
/opr  
/lp "53262"
```

Tip: You can add additional commands as needed, such as to define a specific Test Policy, or create a report (for details, see Chapter 15, “CLI,” on page 331).

3. Add a fifteen second timeout batch, to allow AppScan time to open:

```
START "" "[AppScan Standard installation folder]\AppScanCMD.exe"  
/starting_url "https://demo.testfire.net"  
/credentials "jsmith:demo1234" /test_only /opr /lp "53262"
```

```
timeout /t 15
```

4. Add your Selenium script:

```
START "" "[AppScan Standard installation folder]\AppScanCMD.exe"  
/starting_url "https://demo.testfire.net"  
/credentials "jsmith:demo1234" /test_only /opr /lp "53262"
```

```
timeout /t 15
```

```
java -jar selenium-server-standalone-2.52.0.jar -trustAllSSLCertificates -htmlSuite  
"*firefox" "https://demo.testfire.net" "mytestsuite.html" "results.html"
```

Important: The Starting URL for Selenium must be identical to the Starting URL in the AppScan template.

5. Insert into the Selenium script the AppScan proxy port and host to which Selenium will send its requests:

```
START "" "[AppScan Standard installation folder]\AppScanCMD.exe"  
/starting_url "https://demo.testfire.net"  
/credentials "jsmith:demo1234" /test_only /opr /lp "53262"
```

```
timeout /t 15
```

```
java -Dhttp.proxyHost=localhost -Dhttp.proxyPort=56232 -Dhttps.proxyHost=localhost  
-Dhttps.proxyPort=56232 -jar selenium-server-standalone-2.52.0.jar -trustAllSSLCertificates  
-htmlSuite "*firefox" "https://demo.testfire.net" "mytestsuite.html" "results.html"
```

6. Add a command at the end to close the AppScan proxy and start the Test stage:

```
START "" "[AppScan Standard installation folder]\AppScanCMD.exe" /scan_template  
" C:\Users\<username>\Documents\AppScan\QA Automation Demo Template.scant " /test_only /opr /lp  
"56232"
```

```
timeout /t 15
```

```
java -Dhttp.proxyHost=localhost -Dhttp.proxyPort=64345 -Dhttps.proxyHost=localhost  
-Dhttps.proxyPort=56232 -jar selenium-server-standalone-2.52.0.jar -trustAllSSLCertificates  
-htmlSuite "*firefox" "https://demo.testfire.net" "mytestsuite.html" "results.html"
```

```
"[AppScan Standard installation folder]\AppScanCMD.exe" cpr
```

7. Save the file, and run it using the command line.

Results

Selenium opens and explores the site, then AppScan starts testing. When the scan is complete, AppScan saves it in the same folder as the batch file.

Important: If the process does not complete for any reason, or if you stop it before it completes, you may find that your Internet Explorer and Chrome browser proxy settings are changed, and you will need to undo the change manually.

Application Security on Cloud

This section describes ways AppScan Standard can interact with IBM Application Security on Cloud, to scan apps on the cloud.

You must have an Application Security on Cloud account to use this feature, and have created at least one application in it.

If your site is not accessible from the Internet, you must have already created an AppScan Presence, with access to the site and to the Internet, for Application Security on Cloud to connect with.

For details see the Application Security on Cloud documentation.

Uploading to Application Security on Cloud

You can upload an AppScan Standard scan or template file (SCAN or SCANT) to IBM Application Security on Cloud to run a new scan on the cloud.

Procedure

1. Open the scan or template you want to upload.
2. Click **File > Export > Upload Scan to Application Security on Cloud**.
3. Log in with your Key ID and Key Secret.
4. Click **Select Application**, select an existing application from the list, and then click **Select**.
The Run Scan dialog box opens.
5. **Scan Settings** area: You can optionally change the scan name, and select the check box to be notified when the scan is complete.
6. **Private Site Scan** area: Only if your site is *not* available on the Internet, select the check box and then select your AppScan Presence from the list.

7. **Test Options** area: Select one of the two radio buttons:

Test Only

Test the parts of the site that were already explored in AppScan Standard. Use this option if you want to test only the Multi-Step Operations or Manual Explore that you recorded in AppScan Standard. A Test stage is run on the existing Explore data in the file you upload.

Full Scan

Continue the Explore stage (adds data to any existing Explore data saved in the file), and then runs a Test stage. Note that if the uploaded file contains any Explore data that data is used and added to; to ignore it you must select Test Only.

8. Click **Upload and Run**.

A notice confirms that the upload is completed successfully. The scan should start immediately, but scan status can be seen only from Application Security on Cloud.

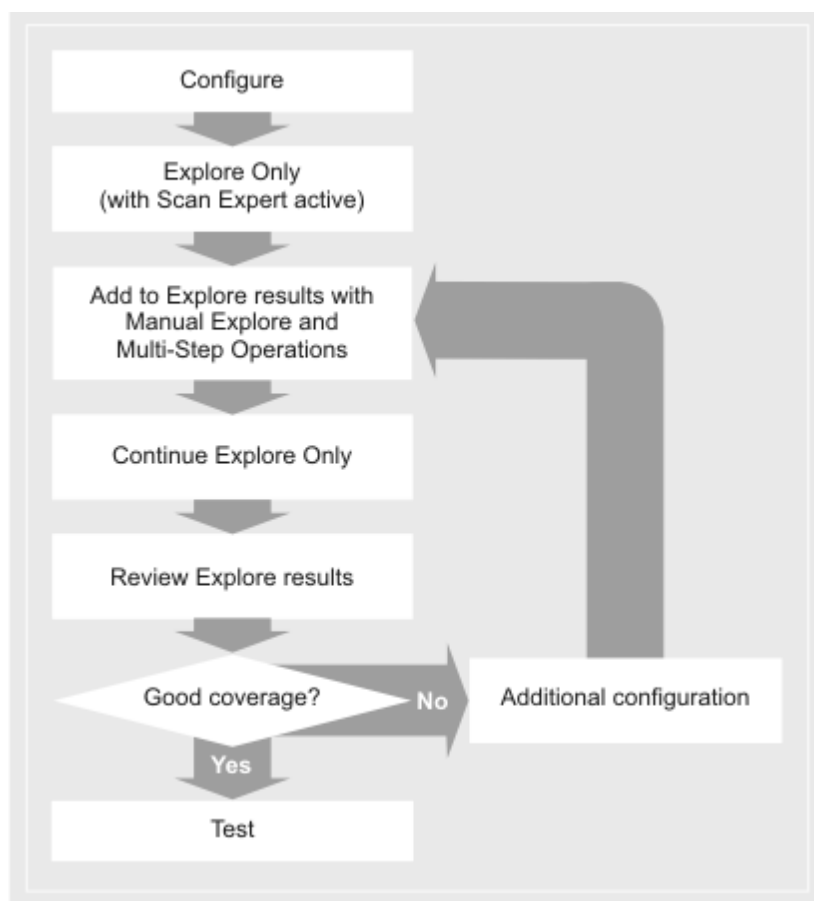
Chapter 13. Best practices and FAQ

This section contains some best practices and use cases for advanced users, and some frequently asked questions.

Workflow for advanced users

This workflow can help users with experience in the field of web security achieve a more thorough scan.

The success of the Test stage, and therefore of the scan itself, depends on the coverage achieved during the Explore stage. If the Explore stage misses important parts of the application logic, the Test stage will not be able to reveal important vulnerabilities that might exist. Following this workflow can help you improve Explore stage coverage.



Task	Description
1. Initial Configuration	Using the wizard or the Scan Configuration dialog box: 1. Define the starting URL 2. Record the login procedure 3. Validate the in-session pattern and if needed choose a new pattern 4. If your site has an account lockout feature, either disable the feature, or configure AppScan not to test login pages - otherwise AppScan will get locked out of the site during the Test stage and be unable to proceed. For a fuller description of this step, see “Initial Configuration” on page 287
2. Explore Only	Run an initial Automatic Explore: 1. On the toolbar, click  > Explore Only, and let the new Explore stage complete. AppScan will explore the site but not test it yet. At the start of the Explore stage Scan Expert will run, and may suggest some changes to the configuration. The default setting lets Scan Expert make only those changes that can be applied automatically. Note: If your site uses URL rewriting, run Explore Optimization (Tools > Extensions > Explore Optimization Module: Run), and if recommended by the extension, run the Automatic Explore stage again (Scan > Re-Explore). 2. If the Explore terminates early due to AppScan being out-of-session, re-record and reconfigure the login procedure, paying special attention to In-Session Detection and Session ID Tracking. For a fuller description of this step, see “Initial Automatic Explore” on page 288
3. Improve site coverage manually using the browser	Add URLs which Automatic Explore missed: 1. Manual Explore: Use Manual Explore to add individual pages, such as those that require specific input. Note: In rare cases where the built-in browser is unable to browse the application, you can configure AppScan to use a different browser. 2. Multi-Step Operations: If parts of the site can be reached only by clicking links <i>in a specific order</i>, record one or more Multi-Step Operations. For a fuller description of this step, see “Improve site coverage manually” on page 288
4. Continue Explore Only	With the new data you have provided through the Manual Explore, Automatic Explore will probably be able to explore the application more thoroughly. Note: Click  > Continue Automatic Explore (or Scan > Explore Only), to preserve the initial Explore results and the Manual Explore data. Do not click Re-Scan > Re-Explore, as this will delete the existing data.
5. Evaluate Explore results	Review the results so far, to see if the application logic has been well covered by the exploring done so far. Note: If you make any configuration changes, you should run Automatic Explore again (Scan > Re-Explore). For a fuller description of this step, see “Evaluate Explore results” on page 290
6. (If needed) Additional configuration	There are some additional configuration options that you should consider if application coverage so far is not sufficient. For a fuller description of this step, see “Additional configuration” on page 292
7. Test stage	Click Test Only to proceed with the Test stage, completing the scan.

Initial Configuration

About this task

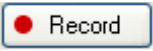
Perform these basic configuration steps before attempting to start exploring the site, whether manually or automatically.

Procedure

1. Define and verify the starting URL for the scan.

- Click **File > New** and create a new web application scan using the wizard (or configure using the configuration dialog box, **Scan Configuration > URL and Servers view**).
- Type in the Starting URL for the scan.
- If your application is case-sensitive, make sure that the Case Sensitive Path check box is selected.
In general, applications that run on Linux based operating systems tend to be case-sensitive, while those that run on Microsoft Windows are not case-sensitive. Java based applications are an exception, and tend to be case-sensitive on any operating system.
- Click the **View in Browser** icon, next to the URL field, and verify that the expected page appears in the AppScan browser.

2. Record the login procedure. This enables AppScan to log in to the application both to start the scan and whenever it gets logged out during the scan.

- In Step 2 of the wizard, (or in **Scan Configuration > Login Management view**), click  to start recording your actions. The browser opens at the starting URL you defined previously.
- Perform all the steps a user needs to do in order to log in to the application.
- Look for some indication on the page that confirms you are logged in, such as "Welcome [Username]", or a "Log Out" link, *that would only be seen by a logged-in user*.
- Close the browser and look for the green key icon  to confirm that an in-session pattern has been identified.

If the icon is red , an in-session pattern has *not* been detected, and you must define it manually (see "Select Detection Pattern dialog box" on page 53).

Note: Although, in general, the first URL whose response includes the in-session pattern should be the "In-Session URL", and this is the URL that is selected automatically, sometimes you can improve performance by selecting a later URL (see "Optimizing In-Session Detection" on page 54).

3. Validate the in-session pattern. The in-session pattern is a regular expression that matches a pattern or string on the page that appears to the user after successful login, such as "Welcome [Username]", or a "Log Out" link. Even though the icon is green, you should verify this pattern.

- In Step 2 of the wizard, select **I want to configure In-session detection options**, and then click **Next** (or go to **Scan Configuration > Login Management > Details view**).
The Login sequence is shown.
- Double-click on the page marked "In Session" to open it in the browser.
- In the browser, click on the Request/Response tab to see the source code, and verify that the selected pattern does indeed indicate in-session status.

Note: If the page content is JavaScript or CSS, then in all cases it is not suitable as the in-session page, and you should choose another page.

If the key icon is green, but the selected pattern is *not* an in-session pattern, refer to "Request-based login troubleshooting" on page 307.

4. **Lockout configuration.** During the Test stage, AppScan makes many invalid login attempts. If your site has an account lockout feature, that locks users out when invalid passwords are entered a certain number of times, AppScan will get locked out and be unable to complete the scan.
 - Disable account lockout, or (if this is not practical)
 - Configure AppScan not to test login and logout pages (**Scan Configuration > Test Options**, deselect **Send tests on login and logout pages**).

Related concepts:

“Workflow for advanced users” on page 285

This workflow can help users with experience in the field of web security achieve a more thorough scan.

Initial Automatic Explore

After your basic configuration, you can run an initial Automatic Explore to see how well AppScan covers the site at this stage.

About this task

A full scan consists of both Explore and a Test stages, but for the moment run only an Explore stage.

Site coverage based on your initial configuration may be incomplete, but seeing which parts of the site are discovered and which are not can help you improve the configuration.

Procedure

1. Click **Scan > Explore Only**.

Scan Expert is configured to run automatically before the Explore stage.

2. If Scan Expert suggests configuration changes, follow its recommendations.

Note: Some changes can be implemented automatically, others require input from you.

3. Let AppScan explore the site until it stops. During the scan you can see the Application tree and data fill in as the site is crawled.
4. Verify that the Explore stage finished successfully and did not terminate early due to AppScan being out-of-session.

Note: If it ended early because AppScan was out-of-session, re-record and reconfigure the login procedure, paying special attention to In-session detection and session ID tracking.

5. (Sites that use URL rewriting:) If your site uses URL rewriting, the Explore Optimization module can improve scan efficiency by eliminating unnecessary URLs from the Explore results at this stage.
 - a. Click **Tools > Extensions > Explore Optimization: Run**
 - b. If recommended by the module at the end of the process, click **Scan > Re-Explore**.

Attention: Do not perform this step unless you are sure your site uses URL rewriting.

Related concepts:

“Workflow for advanced users” on page 285

This workflow can help users with experience in the field of web security achieve a more thorough scan.

Improve site coverage manually

You can add URLs that the initial Automatic Explore stage missed, both individual URLs (such as those accessed by forms that require specific input) and ordered sequences of URLs (such as shopping carts).

About this task

There are various reasons why AppScan may be unable to crawl certain parts of the site automatically without first getting input from you.

- If the application has forms that require specific input, you can navigate to the page using the Manual Explore feature and fill in the required data. This will be recorded in the automatic form filler for use during scanning.
- If your application uses JavaScript, Java applets, or Flash - and has links that appear only in certain states, following a specific sequence of states - such links may be missed by Automatic Explore. Using Manual Explore to access them will ensure that AppScan tests them and any additional links they lead to.
- If your application uses Java applets that reveal links, these will not be tested by AppScan unless you explore them manually.
- If certain parts of the site can only be reached by clicking links in a *specific order* (such as a shopping cart), you should record a Multi-Step Operation (**Scan Configuration > Multi-Step Operations view**).

Procedure

1. **Manual Explore.** Use this feature to add URLs that were not discovered during the automatic Explore, and which can be accessed without a specific context.

- a. On the toolbar, click **Manual Explore**.

The AppScan built-in browser opens.

Note: If the application does not appear in the browser as expected, it may not be optimized for the AppScan built-in browser, in which case you can configure AppScan to use a different browser. See “Changing the default browser” on page 304.

- b. Browse the application, clicking the links you want tested, completing data, and trying to cover as much functionality as possible.
- c. When finished, close the window (by clicking the X button on the titlebar).
The manual Explore window opens.
- d. Save the data for use with future scans by clicking **Export**, and saving.
- e. To add the data to the current scan, select **Add All**, then click **OK**.
- f. After the new pages load, on the toolbar click **Scan > Continue Explore Only**.
AppScan will now explore new links revealed by your Manual Explore.
- g. After the Explore is finished, review the Application data to verify that the Explore achieved the coverage you expected.

2. **Multi-Step Operations.** Use this feature to record sequences of URLs that must be accessed in a specific order.

- a. Open **Scan Configuration > Multi-Step Operations view**.
- b. Click the red **Start Recording** button, and choose **Login and then record**.
- c. Log in to the application and perform the process you want to record (such as adding an item to the shopping cart and checking out).

Note: You can click **Pause** if you need to click links that you do not want included in the recording, and then click **Pause** again to resume recording.

- d. Close the browser window.
- e. If some of the parameters in the sequence require a unique value, see “Sequence variables” on page 89, or refer to tech note :

“Using Variables in Multi-Step operations”

Related concepts:

"Workflow for advanced users" on page 285

This workflow can help users with experience in the field of web security achieve a more thorough scan.

"Using AppScan" on page 125

Manual exploring enables you to explore specific parts of your application, filling in fields and forms as you go. This can be a way of ensuring that particular areas of the site are covered, and that AppScan has the information needed to complete forms correctly.

Related reference:

"Multi-Step Operations view" on page 84

Multi-Step Operations view of the Configuration dialog box is for testing parts of the site that can only be reached by clicking links in a specific order.

Evaluate Explore results

Before proceeding to the Test stage, review the Explore results, because if important areas of the site were missed during the Explore stage, they will not be tested during the Test stage.

About this task

The results of the Explore stage are shown in the three Data view panes. The following are some tips for evaluating whether the Explore stage worked well and achieved good coverage of the application.

Note: If you make any configuration changes during this stage, you should re-Explore the application before starting the Test stage.

Procedure

1. **Scan Log.** Use this to check that AppScan wasn't out-of-session too often.
 - a. Click **View > Scan Log**.
 - b. Scroll down the log entries to check that AppScan wasn't out-of-session too often.If AppScan was out-of-session more than a couple of times in five minutes, it may help to re-record and reconfigure the recorded login, paying special attention to the in-session detection configuration.
2. **Application Tree.** This is a graphical representation of all the areas of the site that were discovered and explored. Use it to see how well the site was covered.
 - a. Does the application tree accurately show the hierarchical structure and main pages of your application?
 - b. Are there Login URLs in the tree? (If there are not, the login was never sent.)
 - c. Does the total number of visited URLs (bottom left-hand corner) match your understanding of the actual site size?
 - d. Were a reasonable number of tests created, for sending during the Test stage (should be at least five times the number of URLs)?
3. **Requests sent.** Review and validate the requests sent during the Explore stage.
 - a. In the Data pane, select **Requests** view, to display all requests sent.
 - b. Verify that the login URLs appear in this list, particularly the in-session request and the login request containing the user credentials.
 - c. Look over some of the requests that appear after the login request in the Login procedure. Verify that the response does not contain errors. To do this, type the word "error" into the Detail pane search field, and then select URLs one by one in the upper panel. If a particular response contains the word "error", the search field color will change from red ("not found"), to green ("found"), and the word "error" will be highlighted in the response body.
 - d. If these requests contain error strings, this indicates that the user was out-of-session, and therefore that the Login procedure was not correctly recorded. Record it again.
4. **Application Data view.** This is the default view during the Test stage, and it offers various views by clicking the filters along the top of the pane.

- a. Click F2, or the data icon on the right hand side of the toolbar to open this view.
 - b. At the top of the Data pane, select a filter to view information.
 - c. Click on an item in the data pane to view its details in the Detail pane.
5. **Custom error pages.** 4xx responses are automatically identified as error pages. If your site returns 2xx responses with custom error pages, you must configure AppScan to recognize them. This information is essential in deciding whether tests succeed. Not configuring custom error pages will lead to inaccurate results, both false positive and false negative. Therefore, if you found pages in the previous step containing the word "error" in their response but were not classified as error pages, configure them now.
- a. In the Detail pane, click **Show in Browser** to verify that it is indeed an error page.
 - b. Click **Set as Error Page**.
- Note:** You can also define error pages in **Scan Configuration > Error Pages**, by clicking the plus (+) icon, and defining a string, regular expression, URL or page.
6. **Filtered URLs.** Review the list of requests that were *not* sent, to verify that it does not contain requests that *should* have been sent.
- a. In the Data pane, select **Filtered URLs** view, and verify that the filtered URLs should indeed be filtered, and that they are categorized correctly.
 - b. If URLs were incorrectly filtered because of their domain, ("Untested web Server"), add the domain to the scan (**Configuration > URLs and Servers > Additional Servers and Domains > +**).
 - c. If URLs were incorrectly filtered because the Path Limit was reached, consider making one of the following configuration changes:
 - Increase Redundant Path Limit (**Configuration > Explore Options > Redundant Path Limit**)
 - Adjust default Redundancy Tuning (**Parameters and Cookies > Redundancy Tuning Defaults**)
 - Adjust Redundancy Tuning for individual parameters
7. **Parameter-based navigation.** If all or part of the site sends a single URL, while different parameters control content and structure, refer to "Sites that use parameter-based navigation" on page 293.
8. **Parameters.** In the Data pane, review the parameters discovered during the Explore stage.
- a. In the Data pane, select **Parameters** view, to see all parameters discovered during the Explore stage.
 - b. If necessary, update the definitions (**Configuration > Parameters and Cookies**).
9. **Failed Requests.** These are requests whose response status was 4xx ("Error"). Review this list to check whether legitimate requests unexpectedly got error responses.
- a. In the Data pane, select **Failed Requests** view.
 - b. **404 Not Found:** Click **Show in Browser** to verify that the URL doesn't exist.
 - c. **Timeout or Connection Failed:** See if the scan needs a higher timeout (**Configuration > Communication and Proxy > Timeout**), the site's server or environment needs improving, or whether the connection problems are due to requests being sent simultaneously (**Configuration > Communication and Proxy > Number of Threads**, reduce the setting to "1"), or by too many requests being sent in a given time (**Configuration > Communication and Proxy > RequestRate Limit**).
 - d. **401 or 407 Authentication Required:** This means there are areas of the application that require HTTP authentication (set these in **Configuration > Platform Authentication**).
 - e. **Other 4xx statuses:** Check whether the site returned an error because the user was not logged in. If necessary, record the login procedure again (**Configuration > Login Management**).
10. If, after reviewing the initial Explore results, you feel that coverage is insufficient, refer to the next section for possible configuration changes. See "Additional configuration" on page 292.

Related concepts:

“Workflow for advanced users” on page 285

This workflow can help users with experience in the field of web security achieve a more thorough scan.

“Sites that use parameter-based navigation” on page 293

Sites in which all pages are reached using a single URL, need a specific scan configuration.

Related tasks:

“Additional configuration”

Here are some configuration changes to consider if site coverage in the initial Explore stage was insufficient.

Additional configuration

Here are some configuration changes to consider if site coverage in the initial Explore stage was insufficient.

About this task

If site coverage in the initial Explore stage was insufficient, or took more than about 30 minutes, consider making some or all of the following possible configuration changes to parameter and cookie definitions.

Procedure

1. Click **Configuration > Parameters and Cookies** view.
2. **Tracking.** If your site tracks cookies (such as session identifiers that are updated each time a user logs in), you should make sure they are correctly defined.
 - a. Verify that your site's cookies and parameters have been identified and are listed in the main Parameters and Cookies tab, and set as "Tracked".
 - b. If necessary, define additional parameters and cookies by clicking the  icon. For more details see “Parameter definition” on page 69.
3. **Redundancy Tuning.** Careful redundancy tuning can significantly reduce scan time without any reduction in scan coverage and accuracy. Review the redundancy tuning to ensure that unnecessary duplicate requests are not sent, but necessary ones are. For more details, see “Redundancy tuning” on page 73.
 - a. If the cookie or parameter you want to configure is not listed in click the  icon and define it.
 - b. Verify that settings are appropriate for each cookie or parameter.
4. **Session ID verification.** Accurate session ID definition is important to enable AppScan to create legitimate requests during the Explore stage.
 - a. Verify that the tracking setting for all session IDs are correctly defined as Login Value.
 - b. Verify that all parameters set as Login Value are session IDs.
5. If you made any changes to the Parameters and Cookies definitions, re-Explore the application. It may also be necessary to re-record the manual Explore and multi-step operations.

“Parameters and Cookies view” on page 67

Parameters and Cookies view of the Configuration dialog box.

“Parameter definition” on page 69

“Session IDs” on page 71

“Redundancy tuning” on page 73

Careful redundancy tuning can significantly reduce scan time.

“Workflow for advanced users” on page 285

This workflow can help users with experience in the field of web security achieve a more thorough scan.

Sites that use parameter-based navigation

Sites in which all pages are reached using a single URL, need a specific scan configuration.

A site with "parameter-based" navigation is a site where only one URL (controller) is sent, while different parameters in the URL result in different content and structure being returned. (Sites of this kind are sometimes referred to by our support team as "megascript" sites.)

- In some sites the URL does actually remain the same for all "pages", as in the following example:
`http://site.com/content.aspx?PageName=page1`
`http://site.com/content.aspx?PageName=page2`
- In other sites, all links are directed through a single, proxy page, using a GET parameter. For example:
`http://site.com?default.aspx/redirect=page1`
redirects to:
`http://site.com/page1.aspx`
This poses the same problem for AppScan as the previous case, since the same URL is sent each time.
- In the case of ASP.NET 2.0 postback links, each link generates a POST request to the page it is located on.
Once again, the problem posed to AppScan is the same.

In all cases the requests for all "pages" are sent to the same URL. This requires special treatment from AppScan, as the default configuration will result in an incomplete scan.

To scan a site with parameter-based navigation

1. When creating the scan, select the Parameter-Based Navigation template instead of the Regular Scan template.
2. Verify that the navigational parameter(s) of your site are correctly defined: Go to **Scan > Scan Configuration > Parameters and Cookies**, and check that the regular expression defining the last parameter in the list includes the navigational parameter(s) of your site. If necessary edit the regular expression. (See "Parameters and Cookies view" on page 67.)
3. (Optional and Advanced:) Configure the Content-Based Results tab of the Scan Configuration dialog box, so that AppScan can present a meaningful application tree. (See "Content-Based Results view" on page 90.)
4. Make any other configuration changes you need, and continue as for a regular scan.
5. (Optional:) When viewing scan results in the application tree, select Content-based view instead of the default URL-based view. (See "Application Tree" on page 18.)

See also:

"The Parameter-Based Navigation template" on page 122

"The challenge of parameter-based navigation sites"

The challenge of parameter-based navigation sites

An explanation of the configuration changes needed to scan a site in which the navigation is parameter-based.

By default, AppScan has a Redundant Path Limit of 5 (the maximum number of time a request can be sent to the same URL, see "Explore Options view" on page 62). In regular sites this prevents repeating tests unnecessarily. However, when site navigation is parameter-based, this low limit will effectively prevent AppScan from scanning the site thoroughly, and a scan run using the Regular Scan template will discover and test hardly any of the site.

Increasing the Redundant Path Limit, or disabling it altogether, does not alone solve the problem. In fact it may throw AppScan into an infinite loop, or at least create a scan with so many tests that AppScan runs out of memory. There are two reasons for this:

1. During the Explore stage, when hashing a request, AppScan will include all the parameters and cookies it finds in the request. With the redundant path limit gone, all combinations of these values will be taken into consideration.

Let's say for example that each page from a section of the site contains hundreds of links to a script that retrieves information from the database about an item available for sale. These links contain a parameter named `item_id` which is of no importance in generating new pages, and is used only to retrieve information about the item. AppScan will end up requesting thousand of instances of this item info page unless `item_id` can be excluded from the hash.

2. In the Test stage the problem becomes more severe. Let's say that a request contains two parameters `par1` and `par2`, and AppScan encounters four links that include these parameters:

```
http: // site.com/content.aspx?par1=a&par2=c
http: // site.com/content.aspx?par1=a&par2=d
http: // site.com/content.aspx?par1=b&par2=c
http: // site.com/content.aspx?par1=b&par2=d
```

If there are 400 tests applicable to each parameter, AppScan will send 1600 tests in total (800 on `par1` when `par2=c` and `par2=d` and 800 on `par2` when `par1=a` and `par1=b`)! Therefore, in addition to excluding these parameters from the Explore hash, we must tell AppScan to test each parameter only once: 400 tests on `par1` and 400 tests on `par2`.

The principles derived from the above for scanning a site with parameter-based navigation are therefore:

1. **Explore stage:** Ignore the values of all parameters except navigational parameters.
2. **Test stage:** Do not create new tests when the value of a parameter changes, except for navigational parameters.

See also:

"Sites that use parameter-based navigation" on page 293

"The Parameter-Based Navigation template" on page 122

Scanning live production environments

The following risks and suggestions should be considered before scanning a live site with AppScan.

When scanning a live site, you can use the predefined Production Site template. This template includes a specially selected Production Site test policy, as well as configuration settings designed to minimize the risk of damaging a live site, or causing Denial of Service to real users.

If you choose to use your own configuration or test policy, the following sections can help you configure your scan effectively.

Database may get filled with artificial information sent during scanning

You can reduce the impact of this by taking the following precautions:

- Disable Automatic Form Fill (Scan Configuration > Automatic Form Fill > first check box).

This will ensure that AppScan does not fill forms automatically, submitting data that might flood a database, bulletin board or online forum system, or send unwanted email to an administrator or moderator account. However, you should be aware that doing this will limit AppScan Standard's ability to reach areas of the site that are accessed by submitting forms. In this mode of operation, AppScan will only scan areas of the site that can be accessed by following links (with or without parameters).

- Create a test account for AppScan to use.

Using a test account makes it easier to track database changes (for example, to make sure that services are not actually ordered), and helps site administrators clean up site after scanning.

Consider the following suggestions when creating the account:

- Limit database access to test records only, so that modified records can be restored.
- Ensure that new records created by the test account will be deleted.
- Ensure that purchase orders (or other transactions) from the test account will be ignored.
- If transactions have an impact (such as when dealing with stocks), allow the account access to test records only.
- If the site has forums, allow the test account access to test forums only, so that real customers will not see the tests created during the Test stage.
- If the site has different privileges for different accounts, set up more than one test account, with different privileges. This will ensure a more comprehensive scan of the site.
- Do not create a test account with administrator-level access.

Risk of email flooding

When testing pages that use email notification, AppScan generates many requests and may overload the site's email server.

One or more of the following suggestions can help dealing with this:

- Temporarily change the email addresses on the pages being tested, so that email is sent to an invalid email address.
- If practical, configure AppScan to exclude those pages from the production scan.
- Scan only one web server at a time, and prevent it from connecting to the SMTP server during the scan.
- If you decide to leave Automatic Form Fill enabled, configure it to insert a unique value in the email field, so recipients can easily identify email generated by AppScan.

Scanning through a proxy

If possible, avoid scanning through a proxy. While this is supported, the proxy sometimes obscures results.

Risk of scan getting locked out of the application

Some applications are configured to lock users out after a certain amount of incorrect login attempts. If this happens during the scan, obviously AppScan will be unable to complete the scan.

To avoid this:

- Disable **Send tests on login and logout pages** (Scan Configuration > Test Options).

Risk of causing application failure

To avoid the risk of AppScan causing your live application to fail, you may want to deactivate invasive tests in the test policy. This will ensure that Denial of Service, Buffer Overflow, or other tests that might cause the application or web server to fail, are not sent.

Important: web applications often contain vulnerabilities that can only be discovered by invasive tests. It is not recommended that you omit invasive tests altogether. Instead, test your application for these kinds of vulnerabilities in coordination with your website owner or administrator, perhaps scheduling scans during off-peak hours when the application is likely to be idle.

To disable invasive tests in the current test policy:

1. Open **Configuration > Test Policy**.
2. Click on the **Invasive** column, to group all invasive tests together.
3. Scroll down the invasive tests (tests for which the Invasive value is "Yes"), and deselect any that are currently selected, to exclude them from the scan.

Flash content

This section contains suggestions for scanning Adobe Flash content.

AppScan explores and tests Adobe ActionScript 1.0, 2.0 and 3.0, and Adobe Flex frameworks 2.0 and 3.0. Flash parsing and execution are activated from Explore Options view of the Scan Configuration dialog box.

System requirements

In order for AppScan to be able to execute Adobe Flash content during scanning, you must have a supported version of the Adobe Flash Player for Internet Explorer installed. Versions 9.0.124.0 up to 14.0.0.125 are supported.

Note: The Flash Player comes in the form of an ActiveX plugin that is browser-specific. AppScan requires the Adobe Flash Player for Internet Explorer.

- If your version is too old, you can download the latest Flash Player from <http://get.adobe.com/flashplayer/>
- If you have Adobe Flash Player Version 10.1 or higher, you may get a message that it requires configuration to work with AppScan. See "Flash Player configuration" on page 7

Note: In both cases ("Supported Flash Player not installed", or "Flash Player not configured") an alert will appear in the Configuration dialog box *and Flash Execution will not run* during scans.

Limitations to Flash scanning

The following limitations to Flash scanning should be noted:

- Since AppScan scans Flash content through a proxy, localhost URLs are not scanned.
- Issues found are specific to the version of the Adobe Flash Browser for Internet Explorer that is installed on your machine. It is possible that:
 - Your player is vulnerable to the issue reported, but a player for a different browser, or a later version of the player, is not
 - There are issues that your Internet Explorer player is *not* vulnerable to, and therefore that AppScan does not report, which *do* affect players for other browsers, or earlier versions of this player

Incomplete URL coverage

The following are suggestions for when you have run a scan and looked at the results and it seems that AppScan has not identified URLs from your Flash content.

Why does AppScan identify some URLs from the Flash content but miss others?

There are a few possibilities:

- Verify that the Flash movie version is supported. (Unsupported versions are listed in Application Data view under "Filtered URLs.")
- Play the movie in the Internet Explorer browser, on the machine that ran the scan, to verify that it plays correctly.

- Check that JavaScript Execution (enabled by default) has not been disabled. (**Scan Configuration > Explore Options > Execute JavaScript to discover URLs and dynamic content**)

The Flash content coverage seems to be incomplete

There are a few possibilities:

1. Check that the form filler information is complete (**Scan Configuration > Automatic Form Fill**).
2. Try increasing the **Scan Configuration > Explore Options > Flash > Click limit**.
3. Try increasing the **Scan Configuration > Advanced > Flash: Max time between samples** setting above its default value of 160 ms.
4. Try increasing the **Scan Configuration > Advanced > Flash: Coverage** setting from 1 to 2.
5. Try playing the movie in Internet Explorer to verify that it plays as expected.
6. If the movie does not play, set the Flash Browser **Debug Level** to Trace 3, rescan, and send the browser log ([AppScan Standard installation folder]\Logs\AppScanFlashBrowser.log) to your Support provider.

Vulnerabilities not discovered

The following suggestions are for when AppScan has discovered the URLs and added them to the application tree, but has not discovered vulnerabilities in them.

Why doesn't AppScan discover vulnerabilities in the Flash URLs it discovered?

Possible reasons are:

- There may simply be no suspicious parameters in the movie. Look in **Application Data > Script Parameters** to see the Flash parameters that were discovered.
- Check that all Flash tests (ActionScript 2 and 3) are enabled. (Open **Scan Configuration > Test Policy**, search for "ActionScript", and verify that all tests are selected).
- The movie may simply not be vulnerable.

What else can I do?

If you still suspect that Flash vulnerabilities are being missed, enable Extended Support Mode, repeat the scan, and send to your Support provider. See "Extended Support Mode" on page 303.

Frequently Asked Questions

This topic addresses general application questions.

What are the different ways available to scan web services?

A site is scanned by first exploring it, and then, based on the data gathered, testing it. "Explore data" can be gathered using one or more different Explore methods. In all cases, once the Explore data is gathered AppScan is used to create and send tests to the site during the Test stage.

Exploring web applications (sites with a user interface)

- In the case of applications (sites) without web services it is often sufficient to supply AppScan with the start URL and login authentication credentials for it to be able to test the site.
- If necessary you can manually explore the site *through AppScan*, in order to get access to areas that can only be reached through specific user input.
- For pages that can be reached only by accessing pages in a specific order, you can record a multi-step operation for AppScan to use.

- While the Configuration Wizard lets you configure and start your scan in a few steps, for complex sites the Configuration Dialog Box lets you fine-tune and customize many more settings.

Exploring web services

- You can set up AppScan as a recording proxy for the device (such as a mobile phone or simulator) you use to explore the service. That way AppScan can analyze the Explore data collected, and send appropriate tests. You can also use AppScan to record traffic using external tool, such as a web services functional tester. See “Using AppScan as recording proxy” on page 131.
- If you have Open API description files (JSON or YAML) for your web service, you can use the Web Services Wizard extension to configure a scan, and the multi-step sequences needed to use the service. AppScan will then automatically scan the service.
- If you cannot use the first two methods, and have a WSDL file for your web service (such as a SOAP web service), the AppScan installation optionally includes a separate tool that lets users view the various methods incorporated in the web service, manipulate input data, and examine feedback from the service. You first need to give AppScan the URL of the service. The integrated "Generic Service Client" (GSC) uses the WSDL file to display the individual methods available in a tree format, and create a user-friendly GUI for sending requests to the service. You can use this interface to enter parameters and view the results. The process is "recorded" by AppScan and used to create tests for the service when AppScan scans the site. GSC can also be used as client for REST requests, without parsing a WSDL file, as a simple HTTP client. See “Using GSC” on page 134.

What is the difference between a manual exploring and a multi-step operation?

Manual Exploring

Manual exploring is when you explore your site to gather data that can be used by AppScan to ensure that when it tests the site it covers parts of the application or services that it might have missed with its automatic Explore stage. This may be because specific user input is required, or because the site responds only to a different type of tool or device. You can manually explore using AppScan, or using it as a recording proxy, or using Generic Service Client (GSC).

See Chapter 5, “Manual Exploring,” on page 125

Multi-Step Operation

A multi-step operation is needed to explore parts of the site that can only be reached by clicking links *in a specific order*, such as an online shop where the user adds items to a cart before paying for them. Consider the following three pages:

1. User adds one or more items to a shopping cart
2. User fills in payment and shipping details
3. User receives confirmation that the order is complete

Page 2 can be reached only via Page 1. Page 3 can be reached only via Page 1 followed by Page 2. This is a sequence. In order to be able to test Pages 2 and 3, AppScan must send the correct sequence of HTTP requests before each test.

See “Multi-Step Operations view” on page 84

What is the difference between action-based playback and request-based playback?

When a procedure is recorded for use as the Login or a multi-step operation, two possible playback methods are available:

Request-based playback

Sends the raw HTTP requests from the recording. This method is usually faster.

Action-based playback

Replays the clicks and keystrokes of the user. Reasons for selecting this method could be that the site includes a lot of JavaScript, or that some of the requests in the request-based playback were marked with a red X when you attempted to validate them. This method can increase scan time.

See Configure > Explore > “Review & Validate tab” on page 47, and Configure > “Multi-Step Operations view” on page 84

Chapter 14. Troubleshooting

This section provides suggestions for do-it-yourself troubleshooting.

Troubleshooting features

AppScan provides various information that you can use to more efficiently locate and solve problems.

- **AppScan Log** - Displays information about AppScan functionality, including how it connects to your proxy and the port that the proxy listens to after each restart.
- **Traffic Log** - When enabled, AppScan writes to this log, all traffic sent to and received from your website.

Note: Enabling the traffic log will slow the application and increase disk space consumption.

- **Dump File** - In the event of it crashing, AppScan creates a memory dump file and notifies you of its path and filename. This file contains all available information about the use case that caused the crash, and should be sent to AppScan Standard Support for further investigation.
- **Support Mode** - Enables you to reproduce the problem and then to pack data files, to send to technical support.
- **Packed Files** - AppScan creates an archive of data files for you to send to technical support. These files include: scan file (*.scan), AppScanDbg.log, AppScanSys.log, and AppScanTraffic.log.
- **Report False Positive**- Enables you to send feedback to IBM Security AppScan Standard about specific tests, especially those with positive test results that you suspect should not be positive in your application. AppScan zips information from selected variants and attaches the zip file to a new e-mail message, opened in the default e-mail client.

License troubleshooting

About this task

In order to use IBM Security AppScan Standard for scanning your site you must have a valid license. If you are sure your license is valid, but AppScan does not accept it, try the following steps.

Procedure

1. Verify that the server on which AppScan is used has a network connection with the license server.

Note: AppScan Standard Edition licenses are installed onto the IBM Rational License Key Server, which may not be the same as the server AppScan is run on. In order for AppScan to function, the server on which it is installed must have a network connection with the license key server. Each time the user opens AppScan a licence is checked out, and when AppScan is closed the license is checked back in.

2. Open the license with a text editor (such as MS Notepad).

Important: Do not alter the license file!

3. Verify that the license has not expired.
4. Verify that the site you want to scan is within the license restrictions (both the IP and the hostname).
5. Verify that the MAC address and Disk Serial Number on the license are the same as on the machine.

What to do next

If this procedure does not solve the problem, refer to the document: How to obtain and apply licenses for AppScan Standard, or contact AppScan Support.

Insufficient disk space

About this task

Temporary files created when working with an AppScan scan can reach a size up to the size of the scan itself, so all temporary folders must have this capacity. The size of a scan depends on the site being scanned, the template, the configuration, and the issues found.

- AppScan stores temp files in C:\ProgramData by default.
- Third-party components of AppScan store data in the paths defined for TEMP and TMP Windows User Environment Variables.

Procedure

1. See if you can free disk space for the three paths.
2. If necessary, change the paths:
 - To change the AppScan Temp path, define the new path in:
Tools > Options > Advanced > TempFilesDir

Note: The path must be local and use ASCII characters only.

- To change the third-party temp paths, change the TEMP and TMP values in Windows User Environment Variables.

Digital signature troubleshooting

About this task

Each time you open AppScan, it verifies the digital signature of its security rules (DLL file). Verification can fail for several reasons:

- DLL file is corrupted
- DLL file was tampered with (possibly by a hacker)
- The following certificate, which is installed by default as part of the Windows installation, is missing from the list of "Trusted Root Certification Authorities" certificates in the Windows store:

VeriSign Class 3 Public Primary Certification Authority - G5

Procedure

1. Make sure you have the following root certificate installed:
VeriSign Class 3 Public Primary Certification Authority - G5
2. Uninstall and re-install AppScan.
3. If these steps do not solve the problem, contact Support.

Note: The warning message gives you the option of continuing if you are sure your system has not been compromised.

Importing a legacy scan template

How to import the scan template from a scan saved in a version of AppScan earlier than 8.6.

In AppScan version 8.6 the scan file format was redesigned, and scans saved in earlier versions cannot be opened in the current version. If necessary you can import the template for use in a new scan.

To import the scan template:

1. Change the scan file's extension from SCAN to ZIP
2. Open the ZIP file, then locate and extract templateconfig.xml
3. Change its extension from XML to SCANT
4. Open with AppScan.

Reporting false positive results

If you feel that a particular AppScan result is wrong (false positive) you can email the request that AppScan sent, and your web application's response, to IBM Security AppScan Standard Support. See "Report false positive test results" on page 199.

Troubleshooting the Report False Positive feature

The following tips are provided to help you if you experience any difficulty using the **Report False Positive** feature to send the AppScan support team feedback with attached variant information.

Problem	Cause	Solution
Recipient cannot read attachment	E-mail was sent encrypted. No other recipient will have the private key needed to read the e-mail.	To turn off encryption: From Tools menu > Options command > General tab > Report False Positive option, clear Encrypt Attachments check box.

Extended Support Mode

Extended Support Mode logs all AppScan activity, for packing and sending to your support provider to help troubleshooting a problematic procedure.

About this task

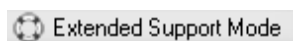
If you need help troubleshooting a problematic procedure, your support provider may ask you to run AppScan in Extended Support Mode, to log all activity and pack the data into a single file.

Important: Do not turn on Extended Support Mode unless directed to do so by authorized support personnel. This mode affects AppScan performance.

Procedure

1. Activate Extended Support Mode: **Help > Support > Extended Support Mode**
A message appears to say that Extended Support Mode is enabled, and will log all your actions.
2. Click **OK**.

An indicator in the Status bar shows that AppScan is running in Extended Support Mode:



3. Reproduce the problematic procedure.
4. When you have finished, disable Extended Support Mode: **Help > Support > Extended Support Mode**

OR

If AppScan stopped working as a result of the procedure, reopen AppScan, and when asked if you want to keep Extended Support Mode enabled, select **Disable**.

5. Select the options you require:

Option	Description
Encrypt	All support data that is saved, including the scan file if included, will be encrypted. Important: Do not select encryption unless you are sending the package to AppScan Support, as only they can open the encrypted file.
Include a scan file	You can include the current scan, or another relevant saved scan, in the support file.

6. Click:

Option	Description
Save	Saves the existing support data as a .SUPPORT file (encrypting and including a scan file if so configured). The support data is cleared from the AppScan interface, and AppScan reverts to its regular mode. The default path is: ...\\My Documents\\AppScan\\Support\\
Don't Save	The support data is cleared from the AppScan interface, and AppScan reverts to its regular mode.
Cancel	The support data is <i>not</i> cleared from the AppScan interface, and AppScan remains in Extended Support Mode.

Changing the default browser

You can configure AppScan to use a browser other than its built-in browser.

About this task

By default, AppScan opens its built-in browser when you record the login procedure, or a multi-step operation, or when you click the **Show in Browser** button.

If the built-in browser is unable to access parts of the site, or if your site is not optimized for it, you can configure AppScan to use a different browser that you have installed on your machine instead.

Procedure

1. Click **Tools > Options**, and select **Use External Browser**.
2. Select a browser from the drop-down list of supported browsers *that are installed on your machine*. Supported browsers are:
 - MS Internet Explorer
 - Mozilla Firefox
 - Google Chrome
 - MS Edge
3. If either of the following two configuration options has been changed, the defined header values will be used, and not the header values of the newly selected browser. If necessary, you can restore the default values
 - **Configuration > Advanced Configuration > Communication: Accept-Language request header value**

- **Configuration > Explore Options > User-Agent header value**

The ways in which these settings may have been changed *by the user* are:

- You opened the browser to record the login procedure or a multi-step operation, or clicked "Show in Browser"
- You manually changed them
- You loaded a saved scan

In order to set these two settings to those of the newly defined browser, simply delete the values, and (if relevant) record the login or multi-step operation again. From now on AppScan will automatically use the header value of the defined browser.

Note: The following limitations apply to the newly defined browser:

- If the selected browser is configured to use a proxy configuration file, this selection will be ignored and the built-in browser will open.
- If you record a multi-step operation using the "Login and then Record" option, the built-in browser will open.
- If another instance of AppScan is open, and you try to open an external browser, the built-in browser will open.
- In **Login Management > Details tab**, if you select a URL and click either **Show in browser** or **Select**, the built-in browser will open.
- If you select Firefox, AppScan automatically creates a Firefox profile named "AppScan" (if one does not already exist). If Firefox was open when this was done, you must close it for the change to take effect.

Related concepts:

"Workflow for advanced users" on page 285

This workflow can help users with experience in the field of web security achieve a more thorough scan.

Related tasks:

"Improve site coverage manually" on page 288

You can add URLs that the initial Automatic Explore stage missed, both individual URLs (such as those accessed by forms that require specific input) and ordered sequences of URLs (such as shopping carts).

Related reference:

"Advanced Configuration view" on page 106

The **Advanced** tab of the **Scan Configuration** dialog box is used to change advanced registry settings that affect specific scans (Scan > Scan Configuration > Advanced tab), and it should only be used by experienced AppScan users, or when instructed to do so by the support team to troubleshoot a problem.

"Explore Options view" on page 62

Explore Options view of the Configuration dialog box.

Login troubleshooting

Tips for troubleshooting session detection problems in Scan Configuration > Login Management view.

When you close the browser after recording the login procedure, the green key icon  confirms that AppScan has detected an in-session pattern that can be used during scanning to verify its in-session status. If one of the other icons appears instead, AppScan may not have enough information to log in to the site during scanning.

Scan Configuration > Login Management records the login sequence in two ways: as actions and as requests. As long as *one* of these two methods succeeds, AppScan will be able to log in to the site. The table below can help you troubleshoot in cases where both methods have failed.

The following table summarizes messages and possible user troubleshooting actions.

Icon	Message	Possible user actions
	Using action-based login Action-based login: Succeeded Request-based login: Succeeded	No action necessary. Action-based login will be used, and request-based login will be available as a fallback method.
	Using action-based login Action-based login: Succeeded Request-based login: Failed	No action necessary. Action-based login will be used. To troubleshoot the request-based sequence, see “Request-based login troubleshooting” on page 307
	Using request-based login Action-based login: Failed Request-based login: Succeeded	No action necessary. Although action-based login is the preferred method, since request-based login succeeded, that version will be used. To troubleshoot the action-based sequence, see “Action-based login troubleshooting” on page 307 Note: If one of the login pages is very slow, it may be more practical to use request-based login as many logins are typically required during a scan.
	Login not yet recorded	Either click  and record a login or, if login is not required, in the Login/Logout tab > Login method, select None to disable session detection.
	Login not yet validated	If changes have been made to one of the sequences, you must click the Validate button to validate the new login sequence.
	In-Session Detection Pattern not defined	First try recording the login again, but this time after you are logged in, click an additional link, before closing the login recorder. The extra link should be to a page whose response will include data or links that are available only when users are in-session. This may enable AppScan to automatically identify a valid pattern. If this does not work, define an in-session pattern yourself. For details see “Select Detection Pattern dialog box” on page 53
	Session request same as login request	Generally, the login sequence should end immediately when AppScan is logged in to the application. However, in rare cases, the in-session request also contains the login request (with username and password). In such cases, whenever AppScan replays the in-session request (to verify that it is logged in) it will actually log itself in, and therefore be unable to detect when it is logged out. The solution is to record the login sequence and when logged in, to click another link on the page. The login sequence will now have an extra step. As long as this new request does not include the credentials, AppScan will be able to use the sequence to verify when it is logged out, and the key icon will change to green.

Icon	Message	Possible user actions
	Session page redirects	If the page selected as the first in-session page redirects to another page, it is likely that the in-session pattern selected by AppScan is incorrect. • Verify that the current In-Session Detection Pattern does indicate in-session status • If you are unsure, try adding the redirect page as an extra step in the request-based login sequence
	Session page not identified	In the Request tab, open the final page of the login sequence, look for a pattern (either in the Browser tab or the Request/Response tab) that is <i>unique</i> to logged-in users (such as a "log out" link), and select that as the in-session pattern. • To troubleshoot the action-based sequence, see "Action-based login troubleshooting" • To troubleshoot the request-based sequence, see "Request-based login troubleshooting" • For an advanced troubleshooting workflow, see "Advanced login troubleshooting workflow" on page 308
	Session detection disabled	No action necessary. Session detection can be enabled by selecting one of the three Login methods: Recorded, Prompt, or Automatic.

Action-based login troubleshooting

Use these steps to troubleshoot action-based login.

Procedure

1. In **Login Management > Details > Actions**, verify that the username (uid) and password (passw) are correct (double-click on the password value to see the value entered), and correct it manually if needed
2. Click  to play the sequence in the player. This may help you see where the sequence fails.

Note: If one of the login pages is very slow, it may be more practical to use request-based login as many logins are typically required during a scan.

3. Click  and record the sequence again, but this time:
 - a. Type the username and password in a text editor and copy-and-paste them into the web page
 - b. Instead of clicking the Submit button on your site, click **Enter** on your keyboard

If none of these steps solve the problem, try using the request-based sequence.

Request-based login troubleshooting

If the In-Session Detection Pattern shown in the Details tab does *not* correctly identify in-session status, you can select a different pattern using the Requests sequence.

Procedure

1. In **Login Management > Details > Requests**, select the URL marked In-Session (highlighted in green), then click the **Select** button at the bottom of the dialog box, to choose a different pattern.

The browser opens and you can select a new pattern either in the browser or the response body tab. Then close the browser and click **Validate**.

2. If you cannot identify an in-session pattern on the final page, do the following:
 - a. Select the request *above* the request you just looked at
 - b. Double-click on it and check that it does not contain the login credentials
 - c. If it does not, click **Select**, and try identify a different pattern
3. If you do not find an in-session pattern, repeat the previous step for the next request up. You can repeat this as necessary, until you reach a request that contains the login credentials.
4. If you are unable to identify an in-session pattern in any of these pages, and there are one or more URLs listed *after* the In-Session page, use the same procedure to look for an in-session pattern on *that* page.
5. If there are *no* extra URLs, try recording the login sequence again, but click one extra link after you are logged in, preferably a personalized setting, and look for an in-session pattern on that page.
6. If this fails, try selecting an out-of-session pattern:
 - a. Select the URL that was originally marked as the In-Session request
 - b. Open a browser (outside AppScan) and send this request on its own (not preceded by the rest of the login sequence)
 - c. Compare the two responses, and try to identify an expression in the body of the response from Step B that does not exist in the in-session page (such as "You are not logged in")

Note: If the request redirects a different page, you cannot use the response you see in the browser, but need to use the response to the actual request, which can be done using a sniffer

- d. At the bottom of the Details tab click the In-Session drop-down button and select **Out-of-session**, and then paste the pattern you identified into the **Detection Pattern** field

What to do next

If this procedure does not solve the problem, refer to the following online technote:

<http://www.ibm.com/support/docview.wss?rs=3378&uid=swg21283302#Overview%20of%20In-Session%20Detection>

Advanced login troubleshooting workflow

Steps to help advanced users troubleshoot login issues.

About this task

The following steps are designed to help you identify and solve login problems systematically. After each suggested step in the workflow, try running the scan again to see if the problem is resolved.

Procedure

1. **Set the action-based browser to be visible**

In **Tools > Options > Advanced > SessionManagement.ShowActionBasedPlayerWindow** change the setting to **True**, so you can see the browser actions.
2. **Verify that the site is up and the credentials are correct**

Open the Starting URL in a regular browser and verify that it is possible to log in manually with the credentials you provided.
3. **Advanced Configuration adjustments** In **Config > URL and Servers** view, click **View in browser** and verify that the AppScan browser is able to log in.
 - If a script error pop-up appears, try the following (individually or all together):

- In **Config. > Advanced Configuration > Communication: Remove 'Accept-Encoding' header** change the setting to **False** and click **Apply**.
- In **Config. > Advanced Configuration > General: Proxy file extension filter** delete all content in the Value field and click **Apply**.
- In **Config. > Advanced Configuration > Session management: Sequence Content Type Filter** delete all content in the Value field and click **Apply**.
- If the site behaves differently in the AppScan browser to the way it behaved in your regular browser, try the following:
 - In **Config. > Explore Options > User Agent**, click the **Edit** icon, delete all content, and click **Apply**.

4. Site that use HTTP authentication

If the site uses HTTP authentication (if you get a pop-up requiring authentication), do the following:

- a. In **Config. > Login Management** view, set **Login Method** to **None**.
- b. In **Config. > HTTP Authentication** view, supply the username and password, and if needed also the domain.

Note: If the username contains a forward slash (/), the content before it is the domain, and after it is the username. Otherwise leave the Domain field empty.

5. Automatic Login

If you are using Automatic Login, try the following:

- a. In **Config. > Login Management > Login/Logout** tab, verify that the Login method is set to **Automatic**.
- b. Fill in the Username and Password.
- c. In **Config. > Login Management > Details** tab, click **Auto Detect In-Session configuration**.

When AppScan tries to log in to the site automatically, three types of issue can occur:

- If AppScan fails to fill the login fields, it may not be able to identify them:
 - 1) Open the Starting URL in a regular browser.
 - 2) Right-click on the Username field, and select **Inspect**.
 - 3) In the HTML source code pane that opens, locate the ID value of the Username field, and copy it to the clipboard.
 - 4) In AppScan, go to **Config. > Automatic Form Fill**, and paste the ID value into the Username Parameters field.
 - 5) Repeat steps ii-iv for the Password ID value.
- If AppScan clicks the wrong buttons, switch to Recorded Login.
- If AppScan fails to identify an In-Session Pattern, refer to “Select Detection Pattern dialog box” on page 53

6. Recorded Login

If you are using Recorded Login, try the following:

- a. In **Config. > Login Management > Login/Logout** tab, verify that the Login method is set to **Recorded**.
- b. Record the Login sequence.
- c. Open the **Details tab > Actions** list, and click the **Play** button.

AppScan attempts to log in to the site. The following issues may occur:

- If AppScan fills the login and password parameters too fast, go to **Login Management > Details > Actions** list, and increase the **Wait** period between actions.
- If AppScan misses out some actions, try changing mouse-click for Tab/Enter, or the reverse.
- d. In **Details tab > Actions** list, click **Validate**.

AppScan plays the sequence and attempts to identify an In-Session pattern. If an in-session pattern is not found, try adding an additional step to the sequence, after the login step, to reach a page with more information (such as "Welcome [username]" or "[userID]", that AppScan can use as the In-Session pattern.

7. Switch to Request-based Login

If none of the above work, try using Request-based Login with an external browser:

- a. In **Tools > Options > Scan Options** click the **Use External Browser** check box, and select a browser.
- b. In **Config > Login Management** set the Login Method to **Recorded**.
- c. Click **Record > Use External Browser**.
- d. Log in to the site and close the browser.

Long or never-ending Explore stage

For some types of site the Explore stage may take a long time or never end.

The cause of this problem may be that the site rewrites parameters into the URL, creating tens - or even hundreds - of "dynamic" URLs for what are essentially the same page.

Example:

```
http://...php/1/index
http://...php/2/index
...
http://...php/100/index
```

Although these URLs will appear in the application tree as 100 separate nodes, and tests will be created for each of them, there is probably no need to test them all separately.

The Explore Optimization module can identify the pattern and create a custom parameter that will enable AppScan to treat all these URLs as one.

See "Explore Optimization module" on page 270

Flash movie troubleshooting

This section offers suggestions for troubleshooting the Flash scanning feature.

Problem	Action
After installing a new version of AppScan, I get the following persistent error message each time I open it: FlashBrowser has encountered a problem and needs to close.	1. Close AppScan 2. Delete all files named user.config in folders of type (Windows 7): C:\Users\<USER_NAME>\AppData\Local\IBM_Corporation\FlashBrowser.exe_Url_<CODE> 3. Restart AppScan
AppScan didn't detect any URLs in the movie.	1. Verify that your system meets the system requirements. 2. Verify that "Execute Flash files to discover potential vulnerabilities" (Scan Configuration > Explore Options) is selected.

Problem	Action
AppScan identified URLs in some Flash movies but not in others.	1. Make sure the Flash movie version is supported. (Unsupported movies are listed in "Application Data" view under "Filtered URLs".) 2. Play the problematic movie in Internet Explorer, on the machine that ran the scan. Verify that the Internet Explorer can play it properly, and the IE status bar displays 'Done' at the end of the movie. 3. Verify that JavaScript Execution is enabled in Scan Configuration > Explore Options.
Scan coverage is incomplete.	1. Make sure that Scan Configuration > Automatic Form Fill information is complete. 2. In Advanced configuration, increase Flash: Time between samples. 3. In Advanced configuration, set 'Flash: Coverage' to 2. 4. Set Advanced Options > ShowDebugFlashExecution to True, restart AppScan, and verify that the movie plays properly in the "Player container" that opens.
AppScan doesn't find any vulnerabilities in the Flash movies.	1. Make sure that in the Test Policy, all ActionScript 2 and 3 Tests are enabled. 2. Check in Application Data view > Script parameters that no Flash parameter is missing. 3. This particular movie may simply not be vulnerable.
Why do I notice a change in standalone Flash movie performance after exploring Flash movies with AppScan?	AppScan makes a reversible configuration change to Flash Player versions 10.1 and later. For details, and instructions for undoing the change, see "Restore Adobe Flash Player settings" on page 313.
Where can I find the Flash Log?	See "Flash Log messages" on page 329.
If I contact Support, what information should I send?	1. Activate Extended Support Mode. 2. Rescan the movie. 3. Send the AppScanFlashBrowser.log file from the AppScan log directory.

Some Flash movies are not scanned

Flash execution is enabled, but AppScan fails to load certain Flash movies during the scan.

Cause

There is a difference in the initialization sequence that Adobe Flash Player uses when executing a Flash movie versus that of a embedded Flash SWF file in a Flash movie.

Behavior 1: Flash movie

In the case of a Flash movie, the Adobe Flash Player executes the following actions:

1. Initializes the Flash Stage object
2. Calls the constructor of the Flash movie itself (Sprite object or Movie Clip object)

Behavior 2: Embedded SWF file

In the case of an SWF file embedded in a Flash movie, the Adobe Flash Player executes the following actions:

1. Calls the constructor of the Flash movie itself (Sprite object or Movie Clip object)
2. Initializes the Flash Stage object

It follows that any embedded Flash movie that refers to the Stage object in its constructor will run into a “null pointer exception”, because the Stage is not initialized at that point.

In order to crawl SWF files during a scan, AppScan Standard loads those files in its own Flash container. While this should not influence the way the Flash files behave, due to the inconsistency mentioned above, when the movie is loaded into the AppScan Standard container, the Adobe Flash Player will default to Behavior 2 (instead of the expected Behavior 1). If the movie includes any reference in its constructor to the Stage object, AppScan Standard will find a null pointer and will be unable to load the movie.

Workaround

Given the current Adobe Flash Player functionality, the only workaround for this issue is to make a small change to the SWF files of the site being scanned. This change will not affect the functionality of the Flash movie, and will not pose any security risk for the SWF file in question.

Code example:

Typical structure of the problematic SWF file:

```
package {

    import flash.display.*;
    import flash.events.*;

    public class TestSample extends MovieClip {

        public function TestSample(){

            // Begin initialization tasks
            // There may be one or more references to the Stage object here

            // For example: stage.addEventListener(MouseEvent.CLICK,MouseClicked);

            // End of initialization tasks

        }
        // other functions – no change required

    }

}
```

The solution is to copy the initialization tasks:

```
package {

    import flash.display.*;
    import flash.events.*;

    public class TestSample extends MovieClip {

        public function TestSample(){
            this.addEventListener(Event.ADDED_TO_STAGE, solutionToFlashProblem);
        }

        private function solutionToFlashProblem(e:Event):void
        {

            // Begin initialization tasks
```

```

// There may be one or more references to the Stage object here
// For example: stage.addEventListener(MouseEvent.CLICK,MouseClicked);

// End of initialization tasks

}
// other functions – no change required
}
}

```

All that has been done is to copy the content of the constructor into a callback function that will be invoked when the current class is added to the Stage object. When that happens, the Stage object will be initialized and the Flash Player will therefore follow Behavior 1.

Restore Adobe Flash Player settings

About this task

After scanning with Flash Execution active, Flash Player 10.1 (and later) functionality is slightly impaired and behaves like Flash 10.0 in one respect. It is a feature of version 10.1 (unlike version 10.0) that the Flash player stops running when its window is not the active window. This feature is disabled when you scan.

When you have finished scanning you can undo the change that was made. However, be aware that next time you scan with Flash Execution active the setting will be changed back again.

Procedure

1. Open the Flash Player folder. By default this is usually:
C:\WINDOWS\system32\Macromed\Flash
2. Locate the file `mms.cfg` and open it with a text editor (such as Microsoft Notepad).
3. Locate the line:
FullFramerateWhenInvisible=True
4. Delete this line (or set its value to False).
5. Save the file.
6. Restart your browser for the change to take effect.

Multi-step operation troubleshooting

Some suggestions for troubleshooting action-based multi-step operations.

If AppScan fails to scan your action-based multi-step recording, try the following troubleshooting steps:

1. In **Configuration > Multi-step Operations**, select the sequence and click **Validate**. The browser opens and replays the sequence. Verify that the browser performs all the expected steps. If the recorded actions do not match the current version of the site, re-record the sequence.
2. When the browser closes, verify that a green check mark appears next to each action in the sequence. If one of more of the actions has a red X, verify that the action is needed to perform the next step in the sequence (such as clicking on a pop-up that doesn't appear in the play-back). If it is not needed, delete it by selecting the action and clicking .
3. After each step in the sequence, AppScan inserts a WAIT action of duration 0 seconds. Try increasing the duration of the WAIT action before a failed action.
4. Check that the Redundancy Tuning and Tracking Options for all parameters and cookies in the test request are correct. If they are not, either correct them, or delete them from the list in **Configuration > Parameters and Cookies > main tab**.
5. Increase the no-interaction timeout in **Configuration > Advanced Configuration > Action-based: Multi-step no-interaction timeout**.

Replacing unsigned extensions

If you want to use an unsigned extension that you used with a previous version of AppScan, you can either elect to trust it, or see if a signed version is available to replace it with.

About this task

When AppScan opens, it verifies that enabled extensions are either signed, or have been manually "trusted" by the user. Unsigned extensions that have not been manually "trusted" are not loaded.

If you have an unsigned IBM extension that you used with an earlier version of AppScan, you can elect to trust it by clicking the **Trust** button next to it, or see if a signed version is available to replace it with, as described below.

Procedure

1. Open Extension Manager (**Tools > Extensions > Extension Manager**).

Extensions that have been enabled by the user (checkbox selected) but which are not signed, have a Trust button next to them.

Note: To trust the extension without replacing it, simply click the Trust button. From now on the extension will be trusted.

2. Click the **Get More Extensions** link at the top of the dialog box.
3. In the page that opens, click the eXtensions tab and see if a signed version of the extension is available.
4. Save the extension to your machine.
5. In Extension Manager, select the unsigned version (highlight it) and click **Remove**.
6. Click **Install**, select the new extension, and click **Open**.

The new extension is installed and enabled. You need to restart AppScan for the change to take effect.

Related reference:

"Extension Manager" on page 268

Extension Manager lets you add/remove, enable/disable extensions for use with AppScan

Scan Log messages

The following sections include explanations of Scan Log messages (View > Scan Log).

All Scan Log messages are documented in this section. When relevant, explanations and suggested user actions are included.

CRWAD0201I Scan created

Explanation: A new scan has been created. Any previous open scan data has been discarded.

User response: n/a

Explanation: Crawling the application as part of the Explore stage.

User response: n/a

CRWAD0202I Starting Explore

Explanation: Starting the Explore stage of the scan.

User response: n/a

CRWAD0204I Analyzing Explore results

Explanation: Analyzing Explore results to create tests.

User response: n/a

CRWAD0203I Crawling

CRWAD0205I Starting Test

Explanation: Starting the Test stage of the scan.

User response: n/a

CRWAD0206I Sending Starting URLs

Explanation: Sending the Starting URL, and any multi-step operations that were recorded.

User response: n/a

CRWAD0214I Starting Manual Explore

Explanation: n/a

User response: n/a

CRWAD0215I Multiphase scanning: Starting phase <#>

Explanation: n/a

User response: n/a

CRWAD0216I Scan stopped; scan duration: <time>

Explanation: n/a

User response: n/a

CRWAD0217I Explore and Test data cleared

Explanation: n/a

User response: n/a

CRWAD0218I Test data cleared

Explanation: n/a

User response: n/a

CRWAD0219I Time limit reached. Stopping scan...

Explanation: The time limit set for Scan Expert's Explore stage was reached, so it was stopped and Scan Expert proceeded to the evaluation stage. (If AppScan is configured to apply changes automatically and then proceed to the main scan, it will do so. "Stopping scan" refers only to Scan Expert's Explore stage.)

User response: To change the time limit go to Scan Configuration > Scan Expert.

CRWAD0220I Starting Flash Execution

Explanation: n/a

User response: n/a

CRWAD0221I Supported Flash Player (Version 9.0.124.0 or later, for Internet Explorer) not installed. Flash Execution will not run.

Explanation: The scan is configured to include Flash file execution, but a supported version of the Adobe Flash Player is not installed on this machine, therefore

Flash files will not be executed.

User response: If you want AppScan to execute Flash files during scanning, you must install a supported version.

You can download the latest Adobe Flash Player from <http://get.adobe.com/flashplayer/>

CRWAD0222I Flash Player not configured. Flash Execution will not run.

Explanation: The scan is configured to include Flash file execution, but the Adobe Flash Player is not configured to work with AppScan, therefore Flash files will not be executed during the scan.

User response: If you want AppScan to execute Flash files during scanning, an administrator must configure the Flash Browser. See "Flash Player configuration" on page 7.

CRWAD0301I Visited URL: <URL>

Explanation: n/a

User response: n/a

CRWAD0302I Link <URL> extracted from dynamic content in URL <URL>

Explanation: A URL has been extracted for testing from JavaScript or Flash content.

User response: n/a

CRWAD0303I Skipping URL (due to extension): <URL>

Explanation: The scan is configured to exclude files with this extension, so this URL is being skipped.

User response: This can be changed in Configuration | Exclude Paths and Files.

CRWAD0304I Skipping URL (due to exclusion): <URL>

Explanation: The scan is configured to exclude this URL.

User response: This can be changed in Configuration | Exclude Paths and Files.

CRWAD0305I Skipping URL (host not included in scan): <URL>

Explanation: The scan is not configured to include this host.

User response: This can be changed in Configuration | URL and Servers.

CRWAD0306I Skipping URL (path limit exceeded):
<URL>

Explanation: n/a

User response: Path limit can be changed in
Configuration | Explore Options.

CRWAD0307I Skipping URL (depth limit exceeded):
<URL>

Explanation: n/a

User response: Depth limit can be changed in
Configuration | Explore Options.

CRWAD0308I Skipping URL (link limit exceeded):
<URL>

Explanation: n/a

User response: Link limit can be changed in
Configuration | Explore Options.

CRWAD0309I Created test <ID> <Issue Type Name>
for URL: <URL> <param>

Explanation: n/a

User response: n/a

CRWAD0310I Visited page: [URL]

Explanation: n/a

User response: n/a

CRWAD0311I Visited page: <URL>

Explanation: n/a

User response: n/a

CRWAD0312I No decoding service was found for the
domain: <Host>

Explanation: n/a

User response: n/a

CRWAD0313I A WebSphere Portal decoding service
URL <URL> was located.

Explanation: n/a

User response: n/a

CRWAD0401I Login request detected: <URL>

Explanation: n/a

User response: n/a

CRWAD0402I Logout request detected: <URL>

Explanation: n/a

User response: n/a

CRWAD0403I Session Identifier detected; name =
<name>; value = <value>

Explanation: n/a

User response: n/a

CRWAD0404I Session Identifier value refreshed;
name = <name>; value = <value>

Explanation: n/a

User response: n/a

CRWAD0405I Session expired

Explanation: n/a

User response: n/a

CRWAD0406I Performing login

Explanation: n/a

User response: n/a

CRWAD0407I In-session pattern not detected

Explanation: n/a

User response: The in-session pattern is defined in
Configuration | Login Management | Details.

CRWAD0408I Operation timed out

Explanation: n/a

User response: n/a

CRWAD0409I Communication error

Explanation: n/a

User response: n/a

CRWAD0410I Unnecessary request removed: [URL]

Explanation: n/a

User response: n/a

CRWAD0411I JavaScript execution in recorded login
enabled

Explanation: n/a

User response: n/a

CRWAD0412I Parameter [name] should be tracked in login, but is predefined as untracked.

Explanation: The specified parameter occurs in the login sequence, and AppScan wanted to add it to the list of parameters and cookies (**Scan Configuration > Parameters and Cookies**) and set it to "Tracked" (to track it during the scan). However it is already listed, and is configured as "Do not track", so AppScan has not changed its configuration.

User response: You may want to manually change the status of this parameter to "Tracked".

CRWAD0413I Cookie [name] should be tracked in login, but is predefined as untracked.

Explanation: The specified cookie occurs in the login sequence, and AppScan wanted to add it to the list of parameters and cookies (**Scan Configuration > Parameters and Cookies**) and set it to "Tracked" (to track it during the scan). However it is already listed, and is configured as "Do not track", so AppScan has not changed its configuration.

User response: You may want to manually change the status of this cookie to "Tracked".

CRWAD0414I New tracked parameter [name] added:

Explanation: The specified parameter has been added to the list of parameters and cookies (**Scan Configuration > Parameters and Cookies**), and defined as "Tracked", so it will be tracked during scanning.

User response: n/a

CRWAD0415I New tracked cookie [name] added:

Explanation: The specified cookie has been added to the list of parameters and cookies (**Scan Configuration > Parameters and Cookies**), and defined as "Tracked", so it will be tracked during scanning.

User response: n/a

CRWAD0416I In-Session Detection format set to: [pattern format]

Explanation: The format for the pattern which AppScan searches for to verify that it is in-session (see **Scan Configuration > Login Management > Details**), has been changed.

User response: n/a

CRWAD0417I In-Session Detection pattern set to: [pattern]

Explanation: n/a

User response: n/a

CRWAD0418I In-Session URL set to: [URL]

Explanation: The URL in the login sequence which AppScan uses to verify that it is logged-in, has been set to this URL (**Scan Configuration > Login Management > Details**).

User response: n/a

CRWAD0419I Login username set to: '<name>' = '<value>'

Explanation: n/a

User response: n/a

CRWAD0420I Login password set to: 'name' = 'value'

Explanation: n/a

User response: n/a

CRWAD0421I Logout page detection pattern set to: 'pattern'

Explanation: n/a

User response: n/a

CRWAD0501I Test <ID> (<name>) is POSITIVE on: <URL> <param>

Explanation: n/a

User response: n/a

CRWAD0502I Test <ID> (<name>) is negative on: <URL> <param>

Explanation: n/a

User response: n/a

CRWAD0503I Test request <URL> failed due to communication error: <Error description>

Explanation: Possible reasons for the five possible error messages:

- Unable to connect
 - The remote host is actively refusing the connection
 - The remote host is down
 - The network is unavailable
- Connection timed out
 - No response was received from the server within the time limit
- Connection closed (remotely)
 - The application tried to set KeepAlive on a connection that had already timed out
 - The connection was reset by the remote peer
- Connection closed (locally)

- The connection was aborted by the underlying socket provider
- The overlapped operation was aborted due to the closure of the socket
- There are too many open sockets in the underlying socket provider
- Unknown
 - All other causes

User response: n/a

CRWAD0504I URL discovered in test response: <URL>; Adding to unvisited URLs

Explanation: The new URL will be explored in the next phase of the scan, if there is one. If the phase limit has been reached, it will appear in the results as unvisited.

User response: n/a

CRWAD0505I Test <ID> (<name>) on <URL> was filtered by Adaptive test.

Explanation: n/a

User response: Adaptive test filtering can be changed in Configuration | Test Options.

CRWAD0506I Sending test <ID> (<name>) on: <URL> <param>

Explanation: n/a

User response: n/a

CRWAD0507I Page analysis started on [URL]

Explanation: n/a

User response: n/a

CRWAD0508I Page analysis completed on [URL]

Explanation: n/a

User response: n/a

CRWAD0509I Page analysis failed on [URL] with error: (error)

Explanation: n/a

User response: n/a

CRWAD0510I Malware test <URL> result: Not Vulnerable (<short description>)

Explanation: n/a

User response: n/a

CRWAD0601I Cannot connect to host: <server name>

Explanation: n/a

User response: n/a

CRWAD0602I Connection established with host: <server name>

Explanation: n/a

User response: n/a

CRWAD0603I Stopping scan because required host(s) stopped responding

Explanation: n/a

User response: n/a

CRWAD0604I Stopping scan due to out-of-session detection

Explanation: AppScan has detected that it is out-of-session, based on the pattern defined in Configuration | Login Management | Details, and was unable to log in. The scan is therefore being stopped.

User response: n/a

CRWAD0605I AppScan has detected that it is out-of-session.

Explanation: AppScan has detected that it is out-of-session, based on the pattern defined in Configuration | Login Management | Details.

User response: n/a

CRWAD0606I Cannot connect to AppScan Enterprise.

Explanation: n/a

User response: n/a

CRWAD0607I SSL configuration mismatch on host: {1}, defaulting to SSL protocol: {0}.

Explanation: When AppScan is unable to connect to the host using the most secure of the SSL protocols selected by the user, it attempts to use the other selected protocols, starting with the next most secure option. This message indicates that a successful match was found. This issue can arise when the operating system doesn't support an SSL protocol that is supported by AppScan and the host.

User response: Installing service packs or using a different operating system might resolve this mismatch issue.

CRWAD0701I Starting Manual Test**Explanation:** n/a**User response:** n/a

CRWAD0702I Security issue added from Manual Test**Explanation:** n/a**User response:** n/a

CRWAD0801I Scan saved: <full path>**Explanation:** n/a**User response:** n/a

CRWAD0802I Security issue <Issue Type Name> deleted by user**Explanation:** This issue was deleted by the user selecting it, right-clicking on it, and choosing Delete.**User response:** n/a

CRWAD0803I Retesting security issue <Issue Type Name>**Explanation:** n/a**User response:** n/a

CRWAD0804I Re-test not supported for Test <ID> [<name>]**Explanation:** The Re-test feature supports only black-box tests.**User response:** In order to re-send this test you must run the appropriate module (such as JSA or Test for Malware) again.

CRWAD0805I Loading scan from file [full path]. Scan created in [version], build [build number]**Explanation:** n/a**User response:** n/a

CRWAD0806I Security issue <Issue Type Name> set as vulnerable by user**Explanation:** n/a**User response:** n/a

CRWAD0807I Security issue <Issue Type Name> set as not vulnerable by user**Explanation:** n/a**User response:** n/a

CRWAD1001I Starting Scan Expert**Explanation:** n/a**User response:** n/a

CRWAD1002I Scan Expert finished**Explanation:** n/a**User response:** n/a

CRWAD1003I Starting Scan Expert Evaluation**Explanation:** n/a**User response:** n/a

CRWAD1004I Scan Expert Evaluation finished**Explanation:** n/a**User response:** n/a

CRWAD1005I Automatically applying Scan Expert test results**Explanation:** Automatically updating the scan configuration based on Scan Expert test results.**User response:** n/a

CRWAD1006I Finished applying Scan Expert test results**Explanation:** Finished automatically updating the scan configuration based on Scan Expert test results.**User response:** n/a

CRWAD1007I Scan Expert Analysis evaluated: <name>**Explanation:** n/a**User response:** n/a

CRWAD1008I Scan Expert Analysis failed to evaluate: <name>**Explanation:** n/a**User response:** n/a

CRWAD1009I Applying Scan Expert
Recommendation: <name>

Explanation: n/a

User response: n/a

CRWAD1010I Scan Expert: Explore Optimization:
Pausing scan to analyze...

Explanation: n/a

User response: n/a

CRWAD1011I Scan Expert: Explore Optimization:
Automatically started

Explanation: The Explore Optimization module is configured to start automatically when you run a scan

User response: This setting can be changed in **Tools > Explore Optimization: Configure**.

CRWAD1012I Scan Expert: Explore Optimization:
Manually started

Explanation: n/a

User response: n/a

CRWAD1013I Scan Expert: Explore Optimization:
Maximum repeats per iteration reached.
Resuming scan to gather more URLs

Explanation: n/a

User response: n/a

CRWAD1014I Scan Expert: Explore Optimization:
Activating High Performance Mode for
this phase

Explanation: n/a

User response: n/a

CRWAD1015I Scan Expert: Explore Optimization:
Phase complete. For next phase setting
URL limit to {0}

Explanation: n/a

User response: n/a

CRWAD1016I Scan Expert: Explore Optimization:
Restarting scan...

Explanation: Following configuration changes made by the Explore Optimization module, it was necessary to delete the existing Explore data and start the Explore stage again.

User response: n/a

CRWAD1017I Scan Expert: Explore Optimization:
Resuming scan...

Explanation: No configuration changes were made that required clearing the existing data before continuing. The scan is being resumed from where it left off.

User response: n/a

CRWAD1018I Scan Expert: Explore Optimization:
Not enough URLs (should be above {0})

Explanation: The Switch Complexity setting defines how many URLs must contain the same rewritten segment before the Explore Optimization module will attempt to define that segment as a parameter and rerun the Explore stage. This threshold was not reached.

User response: You may want to reduce the Switch Complexity setting in **Tools | Extensions | Explore Optimization: Configure**.

CRWAD1019I Scan Expert: Explore Optimization:
Aborting...

Explanation: Explore Optimization was aborted by the user.

User response: n/a

CRWAD1020I Scan Expert: Explore Optimization:
Step 1, identifying navigational
parameters...

Explanation: The Explore Optimization module is attempting to identify navigational parameters. If it does so it will set their Redundancy Tuning to the strictest setting.

User response: n/a

CRWAD1021I Scan Expert: Explore Optimization:
Analyzing {0} unique URLs

Explanation: n/a

User response: n/a

CRWAD1022I Scan Expert: Explore Optimization:
Time allowed for analysis exceeded

Explanation: The time allowed for Explore Optimization to run was exceeded.

User response: You can change this setting in **Tools | Extensions | Explore Optimization: Configuration**.

**CRWAD1023I Scan Expert: Explore Optimization:
Analysis complete****Explanation:** n/a**User response:** n/a

**CRWAD1024I Scan Expert: Explore Optimization:
Analysis failed****Explanation:** n/a**User response:** n/a

**CRWAD1025I Scan Expert: Explore Optimization:
Applying configuration changes...****Explanation:** n/a**User response:** n/a

**CRWAD1026I Scan Expert: Explore Optimization: No
optimizations found****Explanation:** n/a**User response:** n/a

**CRWAD1027I Scan Expert: Explore Optimization:
Redundancy Tuning Default settings
changed****Explanation:** n/a**User response:** n/a

**CRWAD1028I Scan Expert: Explore Optimization:
Found rewrite rule: {0}****Explanation:** n/a**User response:** n/a

**CRWAD1029I Scan Expert: Explore Optimization:
Found navigational parameter '{0}'****Explanation:** The Explore Optimization module found a navigational parameter. It will add it to the list of parameters and cookies, and increase its Redundancy Tuning to the strictest setting.**User response:** n/a

**CRWAD1030I Scan Expert: Explore Optimization:
Step 2, identifying URL rewriting...****Explanation:** The Explore Optimization module is attempting to identify URL rewriting. If it does so, it will create appropriate custom parameters (**Scan Configuration** | **Parameters and Cookies** | **Advanced tab**) and run a new Explore stage.**User response:** n/a

CRWAD1101I <Extension log messages>**Explanation:** This message was written to the log by an eXtension.**User response:** eXtensions are managed in **Tools** | **Extensions** | **Extension Manager**.

**CRWAD1201I Starting Explore of Multi-Step
Operation Sequences****Explanation:** n/a**User response:** Multi-Step Operations are managed in **Configuration** | **Multi-Step Operations**.

**CRWAD1202I Finished Explore of Multi-Step
Operation Sequences****Explanation:** n/a**User response:** n/a

**CRWAD1203I Exploring Multi-Step Operation
Sequence: <name>****Explanation:** n/a**User response:** n/a

**CRWAD1204I Starting Test of Multi-Step Operation
Sequences****Explanation:** n/a**User response:** n/a

**CRWAD1205I Finished Test of Multi-Step Operation
Sequences****Explanation:** n/a**User response:** n/a

**CRWAD1206I Testing Multi-Step Operation
Sequence: <name>****Explanation:** n/a**User response:** n/a

**CRWAD1207I Optimization found for sequence
[name]: Must play requests [#1] to [#2]
before testing request [n]****Explanation:** AppScan has determined that the specified steps in the sequence will be played in order to test request n. Any earlier steps in the sequence (before step [#1]), will be omitted when testing this request.**User response:** If necessary you can review and edit the sequence yourself in **Scan Configuration** > **Multi-Step Operations**.

CRWAD1208I Optimization found for sequence [name]: No playback needed before testing request [n]

Explanation: AppScan has determined that no multi-step playback is needed before testing request [n]. Any earlier steps in the sequence will therefore be omitted when testing this request.

User response: If necessary you can review and edit the sequence yourself in **Scan Configuration > Multi-Step Operations**.

CRWAD1209I Unable to optimize request [n] in sequence: [name]

Explanation: All attempts to send request [n] failed (with or without the earlier requests in the multi-step sequence). As the optimization attempt failed, this request will not be tested.

User response: If AppScan is wrong to ignore this request, disable the **Allow Play Optimization** check box in **Configuration > Multi-Step Operations**.

CRWAD1301I Generating Issue Information

Explanation: n/a

User response: You can control which Issue Information modules run automatically in **Configuration > Issue Information** view.

CRWAD1302I Finished generating Issue Information

Explanation: n/a

User response: n/a

CRWAD1303I Starting Issue Information module: <name>

Explanation: n/a

User response: n/a

CRWAD1304I Issue Information module finished: <name>

Explanation: n/a

User response: n/a

CRWAD1305I Issue Information module failed: <name>

Explanation: n/a

User response: n/a

CRWAD1401I <#> web service request(s) were explored

Explanation: n/a

User response: n/a

CRWAD1601I Glass box detected the parameter [parameter name] in [URL]

Explanation: n/a

User response: n/a

CRWAD1602I Glass box added the new URL [URL]

Explanation: n/a

User response: n/a

CRWAD1603I Cannot connect to glass box server: [server name]

Explanation: n/a

User response: n/a

CRWAD1604I Glass box test <ID> (<name>) is POSITIVE on: <URL> <param>

Explanation: n/a

User response: n/a

CRWAD1605I Glass box test <ID> (<name>) is negative on: <URL> <param>

Explanation: n/a

User response: n/a

CRWAD1606I Glass box test request <URL> failed due to communication error: <Error description>

Explanation: Possible reasons for the five possible error messages:

- Unable to connect
 - The remote host is actively refusing the connection
 - The remote host is down
 - The network is unavailable
- Connection timed out
 - No response was received from the server within the time limit
- Connection closed (remotely)
 - The application tried to set KeepAlive on a connection that had already timed out
 - The connection was reset by the remote peer
- Connection closed (locally)

- The connection was aborted by the underlying socket provider
- The overlapped operation was aborted due to the closure of the socket
- There are too many open sockets in the underlying socket provider
- Unknown
 - All other causes

User response: n/a

CRWAD1607I Sending glass box test <ID> (<name>) on: <URL> <param>

Explanation: n/a

User response: n/a

CRWAD1608I Fetch glass box sink results

Explanation: n/a

User response: n/a

CRWAD1609I Fetch glass box unreferenced parameter results

Explanation: n/a

User response: n/a

CRWAD1610I Glass box: Started unreferenced parameter monitoring.

Explanation: n/a

User response: n/a

CRWAD1611I Glass box: Stopped unreferenced parameter monitoring.

Explanation: n/a

User response: n/a

CRWAD1612I Glass box: Started sink monitoring.

Explanation: n/a

User response: n/a

CRWAD1613I Glass box: Stopped sink monitoring.

Explanation: n/a

User response: n/a

CRWAD1614I Detected glass box agent host operating system, or operating system version, or web server.

Explanation: An informational message indicating that AppScan detected this information about the glass box agent host.

User response: n/a

CRWAD1615I Action-based player started.

Explanation: n/a

User response: n/a

CRWAD1616I Action-based player finished [SUCCESS]

Explanation: n/a

User response: n/a

CRWAD1617I Action-based player finished [FAILED]

Explanation: n/a

User response: n/a

CRWAD1618I Action-based player failed to log in, rolling back to request-based login.

Explanation: AppScan failed to log in using the action-based log in recording, and from now on will use the request-based version.

User response: Try reviewing or re-recording the action-based login sequence to resolve the issue.

AppScan Log messages

The following sections include explanations of AppScan Log messages. (Help > AppScan Log)

Currently only the more important AppScan Log messages are documented in this section. When relevant, explanations and suggested user actions are included.

CRWAD3000E Bad AppScan registry entry <key name>.

Explanation: n/a

User response: n/a

CRWAD3002I Advanced Configuration option set. ID: <ID> = value <value>.

Explanation: n/a

User response: n/a

CRWAD3003E Advanced Configuration option not set due to error. ID: <#>, value <value>.

Explanation: AppScan was unable to apply an Advanced Configuration change made by the user due to incorrect format (ID and value specified).

User response: Locate the Advanced Configuration setting by its ID, and correct the value.

CRWAD3010I AppScan version: <#> (Build number: <#>, Security rules version: <#>)

Explanation: n/a

User response: n/a

CRWAD3100I New AppScan process started.

Explanation: n/a

User response: n/a

CRWAD3101I AppScan exited.

Explanation: n/a

User response: n/a

CRWAD3102I AppScan proxy port is <port number>.

Explanation: n/a

User response: n/a

CRWAD3103I AppScan Flash proxy port is <port number>.

Explanation: n/a

User response: n/a

CRWAD3104I Starting Explore stage on host(s): <host(s)>.

Explanation: n/a

User response: n/a

CRWAD3105I Resuming Explore stage.

Explanation: n/a

User response: n/a

CRWAD3106I Explore stage stopped by user.

Explanation: n/a

User response: n/a

CRWAD3107I Explore stage completed.

Explanation: n/a

User response: n/a

CRWAD3108I Clearing Explore and Test stage data.

Explanation: n/a

User response: n/a

CRWAD3109I Creating a new scan.

Explanation: n/a

User response: n/a

CRWAD3200I Starting Test stage on host(s): <host(s)>.

Explanation: n/a

User response: n/a

CRWAD3201I Resuming Test stage.

Explanation: n/a

User response: n/a

CRWAD3202I Test stage stopped by user.

Explanation: n/a

User response: n/a

CRWAD3203I Test stage completed.

Explanation: n/a

User response: n/a

CRWAD3204I Clearing Test stage data.

Explanation: n/a

User response: n/a

CRWAD3205I Importing Explore data from: <path>.

Explanation: n/a

User response: n/a

CRWAD3300W Import Explore Data failed, from file: <file path>.

Explanation: n/a

User response: n/a

CRWAD3407E License verification failed! RCL error message (if any): <message>

Explanation: Possible causes: 1) License expired, 2) No floating license available, 3) Unable to connect to license key server for more than 3 days. The RCL (Rational Common Licensing) message may contain more details of the cause of the problem.

Note: In some cases the RCL message may not be available.

User response: n/a

CRWAD3409W No license for host: <host>.

Explanation: n/a

User response: Update your license (Help > License) to include the specified host, or remove the host from your list of additional hosts (Scan Configuration > URL and Servers > Additional Servers and Domains).

CRWAD3410W License server unavailable. Running in Disconnected Mode allowed for up to 3 days. RCL message: <message>

Explanation: When AppScan is unable to connect to the license key server to verify the validity of a floating license, it can continue to run in Disconnected Mode for up to 3 days. After that time, if it is still unable to connect to verify the license, scanning will be disabled for all sites except demo.testfire.com. The RCL (Rational Common Licensing) message may contain more details of the cause of the problem.

User response: n/a

CRWAD3411I License successfully checked out.

Explanation: n/a

User response: n/a

CRWAD3500E An AppScan internal error occurred. A dump file was created in: <file path>. Contact Support.

Explanation: A critical internal error has occurred, which caused AppScan to terminate. The memory dump file may help to determine the cause of the problem.

User response: Contact Support and provide the dump file any other relevant information.

CRWAD3501E An AppScan internal error occurred. Attempting to save the scan.

Explanation: A critical internal error occurred. AppScan attempted to save the scan before closing. If the save was successful, the path of the saved scan will appear later in the log.

User response: n/a

CRWAD3502E Auto-Recover save succeeded. Recovered scan saved at: <file path>

Explanation: AppScan succeeded in saving a scan after a critical error. The scan is located in the path indicated.

User response: Try to load the saved scan file and continue the scan.

CRWAD3503E Attempt to save the scan before terminating failed.

Explanation: AppScan was unable to save an Auto-Recover scan file before terminating.

User response: Contact Support.

CRWAD3600E Rules file is corrupted.

Explanation: The security rules file, which defines AppScan tests, is corrupted.

User response: Contact Support.

CRWAD3601E User-Defined rules file is corrupted.

Explanation: The security rules file, which defines user-defined tests, is corrupted.

User response: Contact Support.

CRWAD3602W Some security issues in the loaded scan were deleted due to missing rules.

Explanation: n/a

User response: n/a

CRWAD3707I Server <name> is not responding.

Explanation: n/a

User response: n/a

CRWAD3708I Server <name> is responding again.

Explanation: n/a

User response: n/a

CRWAD3709I Proxy <name> is not responding.

Explanation: n/a

User response: n/a

CRWAD3703E Cannot connect to server {0} due to:
SSL certificate is not valid.

Explanation: AppScan has been configured by the user to reject connections to servers whose SSL certificate is not valid. The SSL certificate of the application server is not valid, and therefore a connection cannot be made.

User response: If you are not concerned about possible man-in-the-middle or other attacks, you can undo this configuration change. Go to **Tools > Options > Advanced tab**, locate the entry for **HttpsIgnoreCertErrors**, and change its Value to False.

CRWAD3710I Proxy <name> is responding again.

Explanation: n/a

User response: n/a

CRWAD3800E Fatal error: scan file <file path> is corrupt.

Explanation: Unable to load scan file.

User response: Contact Support.

CRWAD3801E AppScan was unable to load the session. The session file may be corrupt or invalid.

Explanation: The scan file was successfully opened, but AppScan failed to load a session from it.

User response: Contact Support.

CRWAD3806I Saving scan to file: <full path>

Explanation: n/a

User response: n/a

CRWAD3807I Finished Saving scan to file: <full path>

Explanation: n/a

User response: n/a

CRWAD3808I Loading scan from file: <full path>

Explanation: n/a

User response: n/a

CRWAD3809I Finished Loading scan from file: <full path>

Explanation: n/a

User response: n/a

CRWAD3810I Loading scan with version <original version>, updating to current version <current version>.

Explanation: n/a

User response: n/a

CRWAD3811E Loading scan with version {0}, later than current version {1}!

Explanation: The loaded scan was created in a version (specified), that is later than the current version (specified). The scan may fail to load.

User response: n/a

CRWAD3813E Failed to save scan <file path> due to insufficient disk space.

Explanation: Temporary files created when working with an AppScan scan can reach a size up to the size of the scan itself, so all temporary folders must have this capacity. The size of a scan depends on the site being scanned, the template, the configuration, and the issues found.

User response: Refer to: "Insufficient disk space" on page 302

CRWAD3814E Unable to save scan <file path>, since the scan file is a read only file.

Explanation: n/a

User response: n/a

CRWAD3816E AppScan was unable to load the scan due to insufficient disk space. Please free <#> MB on drive <drive letter> and try again.

Explanation: Temporary files created when working with an AppScan scan can reach a size up to the size of the scan itself, so all temporary folders must have this capacity. The size of a scan depends on the site being scanned, the template, the configuration, and the issues found.

User response: Refer to: "Insufficient disk space" on page 302

CRWAD3817W Rebuilding SDK result database from engine database.

Explanation: The SDK result database was not found. AppScan is rebuilding it from the engine database.

User response: n/a

CRWAD3818E Loading scan with DB version {n}, this version is unsupported.

Explanation: The scan was saved using an unsupported AppScan database version, and cannot be loaded in the current AppScan version.

User response: n/a

CRWAD4105E Critical thread (<thread name>, tid: <thread id>) died unnaturally.

Explanation: A critical internal error occurred.

User response: Contact Support.

CRWAD4106E Non-critical thread (<thread name>, tid: <thread id>) died unnaturally.

Explanation: A critical internal error occurred.

User response: Contact Support.

CRWAD4300I Starting AppScan Extended Support mode.

Explanation: n/a

User response: n/a

CRWAD4301I AppScan Extended Support mode is already <On/Off>.

Explanation: n/a

User response: n/a

CRWAD4302I AppScan Extended Support mode stopped.

Explanation: n/a

User response: n/a

CRWAD4303I Failed to pack Extended Support information into file: <full path>

Explanation: n/a

User response: n/a

CRWAD4500E AppScan stopped due to insufficient disk space. [Free: <#>M , Required: <#>M]

Explanation: Temporary files created when working with an AppScan scan can reach a size up to the size of the scan itself, so all temporary folders must have this capacity. The size of a scan depends on the site being scanned, the template, the configuration, and the issues found.

User response: Refer to: "Insufficient disk space" on page 302

CRWAD4501E AppScan stopped because memory usage reached the predefined limit. [MemUsage: <#>K, MaxMemUsage: <#>K]

Explanation: n/a

User response: Try increasing the memory limit (Tools > Advanced Options), or contact Support.

CRWAD4502E AppScan stopped because the system is running low on virtual memory. [VM: <#>K, AvailableVM: <#>K]

Explanation: n/a

User response: Try closing other applications to free memory.

CRWAD4503E Low system memory - AppScan may not continue to run properly. [MemUsage: <#>K , TotalPhysicalMem: <#>K]

Explanation: n/a

User response: Try closing other applications to free memory.

CRWAD5002E Importing XML Configuration failed. Parse error (if available): %s

Explanation: Failed to send the configuration to the engine. Unable to proceed with scan.

User response: Contact Support.

CRWAD5003I Applying updated scan configuration to engine.

Explanation: n/a

User response: n/a

CRWAD5004E Parsing XML configuration failed.

Explanation: Failed to send the configuration to the engine. Unable to proceed with scan.

User response: Contact Support.

CRWAD5100E AppScan Severe Error: %s

Explanation: n/a

User response: n/a

CRWAD5101E Software exception occurred: <message> Call Stack: <stack>

Explanation: n/a

User response: n/a

CRWAD5400I Glass box server <URL> versions:
 Agent version: <#> ; GBootstrap
 version: <#> ; Agent rules version: <#>

Explanation: n/a

User response: n/a

CRWAD5401E Cannot connect to glass box server {0}
 due to: Either server on which agent is
 installed, or agent application URL, is
 unreachable.

Explanation: AppScan failed to connect with the glass box server for one of the listed reasons. The agent URL may be incorrect, or GBootstrap may not be running, or the server is down.

User response: Verify that the server is running and the agent URL you entered is correct. Using your web browser to open the agent URL may help you verify that GBootstrap is running correctly. If this does not help you identify the problem, contact Support.

CRWAD5402E Cannot connect to glass box server {0}
 due to: Credentials for connecting to
 glass box agent are either not supplied
 or incorrect.

Explanation: The credentials given for accessing the glass box agent are not given or incorrect.

User response: Supply the correct username and password, as defined during the glass box agent installation.

CRWAD5403E Cannot connect to glass box server {0}
 due to: No glass box agent is installed
 on the glass box agent URL specified.

Explanation: The URL leads to a site that is not the glass box GBootstrap root.

User response: Verify the URL. By default it is of the form:

http://<server_name>:<port_number>/GBootstrap/

CRWAD5404E Cannot connect to glass box server {0}
 due to: The glass box agent
 instrumentation is not running.

Explanation: The glass box agent is not instrumented on the specified server.

User response:

1. Stop the server and restart it, using the desktop shortcut (if one was created when you installed the agent).
 2. Reinstall the glass box agent manually (see "Installing the glass box agent" on page 145).
-

CRWAD5405E Cannot connect to glass box server {0}
 due to: Client version {1} is older than
 either glass box agent file version {2} or
 glass box agent application version {3}.

Explanation: The version of AppScan on the local machine is not up to date.

User response: Install the latest version of AppScan.

CRWAD5406E Cannot connect to glass box server {0}
 due to: Both glass box agent file version
 {1} and glass box agent application
 version {2} are older than glass box
 client version {3}

Explanation: The glass box agent version is not up to date.

User response: Install the latest version of the glass box agent on the server machine.

CRWAD5407E Cannot connect to glass box server {0}
 due to: Glass box agent application
 version {1} older than glass box client
 version {2}

Explanation: You may have installed the glass box agent manually and omitted to deploy GBootstrap.war.

User response: Follow exactly the instructions for manually installing the glass box agent on your specific server. See "Installing the glass box agent" on page 145.

CRWAD5408E Cannot connect to glass box server {0}
 due to: Glass box agent file version {1}
 older than glass box client version {2}

Explanation: Glass box agent JAR file version is older than AppScan version.

User response: You may have simply omitted to restart the server after installing the agent. Try restarting the server. If this does not help, reinstall the glass box agent.

CRWAD5409E Cannot connect to glass box server {0}
 due to: Glass box agent rules version {1}
 older than glass box client rules version
 {2}

Explanation: The glass box rules are not up to date.

User response:

1. Go to **AppScan > Help > Check for Updates**.

- If a download is available, it will be automatically downloaded to your machine and installed.
- If a download is not available, check Fix Central for the latest version and install it.

2. To update the agent rules on the server, open **Tools > GB Agent Management**, double-click on the server you need, and in the dialog box that opens, click **Update Rules**.
3. Restart the application server, for the change to take effect.

CRWAD5410E Cannot connect to glass box server {0}
due to: Glass box agent rules were updated but restart is needed to complete the update process.

Explanation: The glass box agent rules were updated, and now the server must be restarted to load the new instrumentation rules.

User response: Restart the server.

CRWAD5411E Cannot connect to glass box server {0}
due to: Glass box client rules version {1} older than glass box agent rules version {2}

Explanation: The glass box agent rules were updated, and now the server must be restarted to load the new instrumentation rules.

User response:

1. Go to **AppScan > Help > Check for Updates**.
 - If a download is available, it will be automatically downloaded to your machine and installed.
 - If an automatic download is not available, check Fix Central (<http://www.ibm.com/support/fixcentral/>) for the latest version and install it.
2. Restart the application server, for the change to take effect.

CRWAD5412E Cannot connect to glass box server {0}
due to: SSL certificate is not valid.

Explanation: AppScan has been configured by the user to reject connections to servers whose SSL certificate is not valid. The SSL certificate of the glass box server is not valid, and therefore a connection cannot be made.

User response: If you are not concerned about possible man-in-the-middle or other attacks, you can undo this configuration change. Go to **Tools > Options > Advanced tab**, locate the entry for **HttpsIgnoreCertErrors**, and change its Value to False.

Flash Log messages

The following table describes using the Flash Log for troubleshooting.

The Flash Log contains messages that may help you with troubleshooting. It is located at:

...[AppScan Standard installation folder]\Logs\AppScanFlashBrowser.log

Error messages

Error message	Possible user response
Explore Failed: Couldn't reach initial state!	Try increasing Configuration > Advanced Configuration > Load movie wait time
Movie (NAME) load failed! (TIME) miliseconds timeout, (COUNT) try(s).	Try increasing Configuration > Advanced Configuration > Load movie wait time
current state failed, since Movie is not ready!	Try increasing Configuration > Advanced Configuration > Load movie wait time
Exception while sending status! (MESSAGE). Microsoft .NET framework Hotfix required! Contact IBM Support. Hotfix reference http://support.microsoft.com/kb/971521	Contact Support to get this hotfix.
Movie(NAME) version (NUMBER) not supported!	Recompile movie to version 7 or later.
Exception in Document Completed at URL(URL)! (MESSAGE)	Contact Support for the correct version of mshtml.dll
Flash Player not Found!	Refer to "Flash Player upgrade" on page 7
Flash Player version not Supported!	Refer to "Flash Player upgrade" on page 7
Flash Player not configured!	Refer to "Flash Player configuration" on page 7
Exception while trying to load SWF file!	Refer to "Some Flash movies are not scanned" on page 311

Warning messages

Warning message	Possible user response
Explore movie (NAME) stopped! Click limit (VALUE) reached.	Try increasing Scan Configuration > Explore Options > Click Limit
Explore movie (NAME) Stopped! Screen limit (VALUE) reached.	Try increasing Scan Configuration > Explore Options > Screen Limit
Exception while loading unsupported Flash Movie! (MESSAGE)	Republish the movie in Flash version 7 or later.
Parsing data from bridge failed due to unsupported movie(NAME)!	Republish the movie in Flash version 7 or later.
Exception while parsing data from bridge in unsupported movie(NAME)!	Republish the movie in Flash version 7 or later.
Movie (NAME) not supported!	Republish the movie in Flash version 7 or later.
Document Complete event not received from Browser!	Play the movie in a stand-alone version of Internet Explorer. If you don't see a "Done" message on the left side of the status bar, the link in the HTML page of the SWF file may be broken.
Movie (NAME) failed to load in (TIME) milliseconds for (NUMBER) try(s)! Reloading...	Try increasing Configuration > Advanced Configuration > Load movie wait time
No Browse Dialog found!	Recheck the specified strings in Advanced Configuration > Flash: File upload string or Flash: File download string . Delete strings which do not identify buttons for upload or download.
Exception while trying to create file for upload! (MESSAGE)	No file for uploading has being configured. Configure one in Advanced Configuration > Flash: File upload path .

Glass box troubleshooting

Open a web page listing glass box agent messages and other troubleshooting tips.

<http://www.ibm.com/support/docview.wss?uid=swg21567723>

Chapter 15. CLI

Certain AppScan functions may be performed using the Command Line Interface (i.e. by typing in a command at the command prompt, rather than using the Graphic User Interface). This may be particularly useful when you need to control AppScan automatically from within a script or batch file.

This section describes the syntax and options available using the CLI.

Command structure

AppScan CLI commands consist of three parts:

1. The utility command: AppScanCMD
2. The specific command to be executed, for example: `exec`
3. The required option for the selected command, for example:
`/base_scan <full path>/d <full path>`

In the above example, the full command is:

```
AppScanCMD exec /base_scan <full path> /d <full path>
```

(This command will run a new scan using the configuration of the selected base scan, and save the results in the specified location.)

Note: Commands can be entered with no prefix (e.g. `exec`), or with a minus sign (e.g. `-exec`), or with a forward slash (e.g. `/exec`). For simplicity, examples in this section will all be given without any prefix.

Note: Command options must be prefixed by *either* a minus sign (e.g. `-base_scan`) *or* a forward slash (e.g. `/base_scan`). For simplicity, the examples following will all be prefixed with a forward slash.

Commands

This section describes the application specific commands that can be run using the CLI.

Exec command

The `exec` command creates a new scan with the starting URL specified (in `/starting_url`, `/base_scan`, or `/scan_template`), runs it, and saves it. It can also optionally be used to generate and save a report of the scan.

To run the `exec` command, type `exec`, `ex`, or `e` at the command prompt, followed by the required command options, as described below.

Note: If no command is specified, the `exec` command runs by default.

Parameters

The following parameters can be included in the exec command.

Parameter	Explanation
/starting_url /url /su <full_path>	Sets the starting URL for the scan. If the starting URL is defined in the scan template or base scan you do not need to define it here.
/credentials /cred /cr <username:password>	Specifies Automatic Login and sets the username and password. This overrides any login configured in the SCANT file (if used).
/base_scan /base /b <full_path>	Specifies a source scan (must include full path), whose configuration will be used for the new scan.
/dest_scan /dest /d <full_path>	Specifies the destination where the new scan will be saved (must include full path). If no path is specified, the scan is saved to a Temp folder, and AppScanCMD will notify you of its exact location and filename.
/scan_template /stemplate /st <full_path>	Specifies the scan template file.
/old_host /ohost /oh <full_path> /new_host /nhost /nh <full_path>	These two parameters enable you to "search and replace" one host with another, in a base scan.
/login_file /lfile /lf <full_path>	Imports a saved login sequence.
/multi_step_file /mstepfile /mf <full_path>	Imports a multi-step operations file.
/manual_explore_file /mexplorefile /mef <full_path>	Imports a Manual Explore file (in format EXD, HAR, DAST.CONFIG or CONFIG). Note: As of AppScan Standard Version 9.0.1, EXD files include response data. To import this data, add flag /ir. If the flag is <i>not</i> added, the response data will not be imported, and instead an Explore stage will be run (the saved requests will be sent) to gather <i>new</i> responses to analyze for testing.
/policy_file /pfile /pf <full_path>	Imports a test policy file.
/additional_domains /adomains /ad <domain>	Defines domains other than that of the starting URL to be included in the scan. If there are more than one additional domains, separate them with a comma or add multiple instances of the parameter.
/report_file /rf <full_path>	Specifies the destination and name for the generated report (must include full path). This field is optional. If not set, no report is generated. Note that if /rt is defined as rc_ase no report file is needed, as the output is published to AppScan Enterprise.
/report_template /rtemplate /rtm <CliDefault GuiDefault Summary DetailedReport Developer QA SiteInventory>	Specifies the types of information that will be included in the report (see "Security reports" on page 211) Default template (if none is specified): CliDefault. This is <i>not</i> identical to the Default template in the Reports dialog box. To use that template, specify GuiDefault.
/report_type /rt <xml pdf rtf txt html rc_ase>	Specifies the report format. Default is XML. rc_ase refers to AppScan Enterprise reports and the output will be published to AppScan Enterprise using the existing settings (to see these click File > Export > Publish to AppScan Enterprise > Connection Settings).

Parameter	Explanation
/ase_application_name /aan <AppScan Enterprise application name>	Specifies the AppScan Enterprise application to which the report will be published. Used only with /report_type rc_ase
/min_severity /ms <low medium high informational>	Specifies the minimum result severity to include in reports. (non-xml reports only). Default is "low".
/test_type /tt <All Application Infrastructure ThirdParty>	Specifies which type of tests to include in the report. Default is "All".

Flags

The following flags can be included in the exec command. Including them is the equivalent of setting them from False to True.

Flag	Explanation
/continue /c	Continue the scan.
/explore_only /eo	Run an Explore stage only.
/include_responses /ir	When importing Manual Explore data (EXD file) that includes response data, include the responses (use with /mef). Note: As of AppScan Standard Version 9.0.1, EXD files include response data. If you add this flag, and the file includes response data, it is used when analyzing for testing. If the file does <i>not</i> include response data, an Explore stage will be run (the saved requests will be sent) to gather <i>new</i> responses to analyze for testing.
/merge_manual_explore_requests /mmer	Apply Explore-stage Redundancy Tuning settings to Manual Explore data, to help avoid duplicate requests (use with /mef).
/multi-step /mstep	Test multi-step operations only.
/open_proxy /oprxy /opr /listening_port /lport /lp <port number> /save_only /saveo /so	Open AppScan recording proxy. By default the port set in Tools > Options > Recording Proxy tab is used. To set a different port use /listening_port <port number>. To save as a SCAN file without running a scan, use /save_only /saveo /so SCAN files are ZIP files containing several component files, including Manual Explore sequences that are saved as individual Manual_Explore_#.exd files (where "#" is a sequential number). EXD files can be imported into another scan.
/scan_log /sl	Display the scan log during the scan.
/test_only /to	Run a Test stage only.
/verbose /v	Include progress lines in the output.

Below are some examples of complete commands.

Example 1

This command will start a scan with the specified starting URL, using the Regular Scan template.

```
appscancmd e /su http://demo.testfire.net.scan
```

Example 2

This will start an Explore stage only with the specified starting URL, using the Regular Scan template.

```
appscancmd e /su http://demo.testfire.net.scan /eo
```

Example 3

This scan includes a manual explore, multi-step operations, a recorded login and a test policy.

```
appscancmd e
/st D:\demo.testfire.net.scant
/d D:\demo.testfire.net.scan
/mef D:\ManualExplore.exd
/mf D:\MyMultistepOperation.seq
/lf D:\LoginSequence.login
/pf D:\MyTestPolicy.policy
```

Example 4: Additional domains

Two or more additional domains can be defined in a single `-additional_domains` parameter, separated by commas, or in separate parameters.

```
appscancmd e
/st D:\demo.testfire.net.scant
/d D:\demo.testfire.net.scan
/mef D:\ManualExplore.exd
/ad demo.testfire.net1,demo.testfire.net2,demo.testfire.net3
/sl
```

OR

```
appscancmd e
/st D:\demo.testfire.net.scant
/d D:\demo.testfire.net.scan
/mef D:\ManualExplore.exd
/ad demo.testfire.net1
/ad demo.testfire.net2
/ad demo.testfire.net3
/sl
```

Example 5: Change hosts and ports

You can change both the host and the port of the scan template or base scan.

```
appscancmd e
/st D:\demo.testfire.net.scant
/d D:\demo.testfire.net.scan
/mef D:\ManualExplore.exd
/oh http://demo.testfire.net:80
/nh http://demo.testfire.net2:8090
```

Report command

The report command loads a specified scan and generates a Security report (for more details, see “Security reports” on page 211).

To run this command, type `report`, `rep`, or `r` at the command prompt, followed by the required command options, as described below. (Base scan and destination parameters are compulsory; other parameters are optional.)

Commands	Explanation
/base_scan /base /b <full_path>	Specifies the source scan from which to create the report (must include full path).

Commands	Explanation
/report_file /rf <full_path>	Specifies the destination and name for the generated report (must include full path). Tip: If /rt is defined as rc_ase no report file is needed, as the output is published to AppScan Enterprise.
/report_template /rtemplate /rtm <CliDefault GuiDefault Summary DetailedReport Developer QA SiteInventory CustomTemplateName>	Specifies the types of information that will be included in the report. Default template (if none is specified): CliDefault. This is <i>not</i> identical to the Default template in the Reports dialog box. To use that template, specify GuiDefault. If you have saved any custom templates, you can specify them too. For details, see “Security reports” on page 211
/report_type /rt <xml xml_report pdf rtf txt html rc_ase>	Specifies the format for saving the results or report: xml The complete scan results saved as an XML file (this is <i>not</i> a report) xml_report A fully structured report, saved as an XML file pdf rtf txt html A fully structured report, saved as PDF, RTE, TXT, or HTML rc_ase Publishes the output to AppScan Enterprise using the settings defined in File > Export > Publish to AppScan Enterprise > Connection Settings
/ase_application_name /aan <AppScan Enterprise application name>	Specifies the AppScan Enterprise application to which the report will be published. Used only with /report_type rc_ase
/min_severity /ms <low medium high informational>	Specifies the minimum result severity to include in reports (non-xml reports only). Default is "Informational", meaning that all severities are included in the report.
/test_type /tt <All Application Infrastructure ThirdParty>	Specifies which type of tests to include in the report. Default is "All".

Flags

The following flags can be included in the report command.

Flag	Explanation
/verbose /v	Include progress lines in the output.

Example 1

This command will publish the results of the specified base scan to AppScan Enterprise.

```
report
-base_scan "D:\demo.testfire.net.scan"
-report_type rc_ase
```

Example 2

This command will start a scan with the specified starting URL, using the Regular Scan template.

```
report
-base_scan "D:\demo.testfire.net.scan"
-report_file D:\SecurityReport.pdf
-report_type pdf
-Scan_Log
-min_severity "informational"
```

Delta analysis report command

The delta analysis report command compares two sets of scan results (for details, see “Delta Analysis reports” on page 226).

To run this command, type `delta_analysis_report`, `delta_report`, or `dar` at the command prompt, followed by the required command options, as described below.

Commands	Explanation
<code>/base_scan /base /b <full_path></code>	Specifies the source scan from which to create the report (must include full path).
<code>/target_scan /target /ts <full_path></code>	Specifies the target scan with which the base scan will be compared (must include full path).
<code>/report_file /rf <full_path></code>	Specifies the destination and name for the generated report (must include full path).
<code>/report_type /rt <xml pdf rtf txt html></code>	Specifies the format for saving the report.

Flags

The following flags can be included in the report command.

Flag	Explanation
<code>/verbose /v</code>	Include progress lines in the output.

This command will produce an XML report comparing the results of the two specified scans.

```
das
-b "D:\demo.testfire.net_1.scan"
-ts "D:\demo.testfire.net_2.scan"
-rf D:\DeltaAnalysisReport.xml
-rt xml_report
```

Other commands

Close Proxy command

The `close_proxy` command closes the AppScan recording proxy if previously open.

To run the `close_proxy` command, type `close_proxy`, `cprxy`, or `cpr` at the command prompt.

Help command

The `help` command prints the command usage described in this section.

To run the `help` command, type `help`, or `h` at the command prompt.

Exit Status codes

When AppScan is run within scripts or batch files, the Exit Status code indicates whether or not operation was successful.

Code	Meaning
0	Finished Successfully
1	AppScan Startup Failed
2	Command Line Error
3	License Invalid
4	Load Failed
5	Scan Failed
6	Report Failed
7	Save Failed
8	General Error

Launching AppScan from the command line

Procedure

From the command line, type: `AppScan.exe [<filename>]`

- This command runs the AppScan executable, to launch the GUI, and you may add an optional parameter to call up a **.scan** or **.scant** file.
- If you include a filename in the command, the given scan or scan template is loaded as soon as the application starts.

Chapter 16. Menus, toolbars and keyboard shortcuts

This section summarizes the main menus and toolbars.

File menu

Used for creating, opening, and saving scans.

Command	Click to:
New	Create a new scan.
Open	Open a saved scan (.scan) or scan template (.scant). Tip: You can also open these files by dragging and dropping from the folder to AppScan.
Save	Save the current scan or scan template.
Save As	Save the current scan or scan template with a new name.
Export > Scan Results as XML	Export the complete scan results as an XML file for use with AppScan Enterprise. There are two options, depending on your version of AppScan Enterprise: • For ASE 9.0.3.1 and later • For earlier versions (legacy)
Export > Scan Results as DB	Export the complete scan results as a relational database. The database option exports the results into a Firebird database structure. This is open source, and follows ODBC and JDBC standards.
Export > Cross Scan Data	If you have defined specific issues as Noise (False Positive), you can export this information for use in scans run on other workstations. (You do not need to do this on your own workstation as the information is automatically saved and applied to future scans.)
Export > Publish results to AppScan Enterprise	Exports the results of an AppScan Standard scan to AppScan Enterprise, runs a Test stage on them, and compiles the results into a report.
Export > Create job in AppScan Enterprise	Exports an AppScan Standard scan to AppScan Enterprise. Requires AppScan Enterprise version 9.0 or later.
Export > Create scan template in AppScan Enterprise	Exports an AppScan Standard configuration to AppScan Enterprise as a template. Requires AppScan Enterprise version 9.0 or later.
Export > Upload Scan to Application Security on Cloud	Uploads an AppScan Standard configuration file (SCAN or SCANT) to Application Security on Cloud. You can use the configuration to run a full scan, or use existing Explore stage results saved in the file and run Test Only.
Import > Explore Data	Load a manual explore file. Supported formats are EXD, HAR, DAST.CONFIG and CONFIG. Note: Externally generated HAR (HTTP Archive) files v1 and v2 are supported.
Import > Cross Scan Data	If specific issues have been defined as Noise (False Positives) on a different workstation and exported, you can import this information for use in scans run on your workstation, so that these issues will not be included in the scan results.
Page Setup	Define paper size, source, orientation and margins for the Print command.
Print Preview	Open a preview window showing the Application Tree or Result List (depending on current cursor location), as they will appear when printed using the Print command.
Print	Print the current contents of the Application tree <i>and</i> the Result list. (All nodes that can be seen by scrolling up or down the Application tree and Result list are included, but the nodes that are closed on the screen will appear closed in the printout.)
Filenames	Recently used files.

Command	Click to:
Exit	Exit AppScan.

Edit menu

Used for customizing scan results.

Command	Click to:
Delete	Delete selected issue or remediation.
Severity	(Issues View only) Customize the severity level of the selected issue.
State	(Issues View only) Designate the selected issue as "Noise" (meaning that although AppScan classified it as an issue, in the context of your application it is not.) Issues designated as Noise can either be removed from the results altogether, or displayed with a strikethrough (View menu > Show Issues Marked as Noise).
Priority	(Remediation View only) Change the priority level of a remediation.
Find	Find strings, IDs, HTTP code, etc., in the current scan results. (Options depend on which of the three Views is currently selected.)

Note: Severity and Priority are mutually exclusive; only one is visible at any time, depending on the selected view.

View menu

Used to determine how the main window appears and what data is displayed.

Command	Click to:
Security Issues	Display Security Issues view.
Remediation Tasks	Display Remediation Tasks view.
Application Data	Display Application Data view (failed requests, visited URLs, script parameters, and so on).
Arrange By >	Select a sort method for the Result List: Severity/Result/Name. Select Reverse Order to list results in descending rather than ascending order.
Layout >	Select Horizontal or Vertical layout of the main window panes.
Welcome Screen	Open the AppScan Welcome Screen.
Show Issues Marked as Noise	Issues AppScan finds, that are irrelevant in the context of your application, can be marked as Noise. This menu item toggles between displaying such issues with strikethrough text, or not at all. When selected, Noise appears with strikethrough; when deselected, Noise is not shown. See "Issue state: Open or Noise" on page 187
Non-vulnerable Variants	Opens the list of variants that are defined as non-vulnerable. See "Non-vulnerable variants" on page 203
Scan Log	Open the log of all actions performed by AppScan during the current scan. See "AppScan Log" on page 275
Customize Toolbar >	Select Large Icons to display larger toolbar icons Select whether the toolbar shows the name of every icon, of selected icons (default), or of none of the icons. (In any case, when you roll your mouse over an icon the tool tip shows its name.)

Scan menu

Used to control the scan.

Command	Click to:
Full Scan	Start a full scan (Explore and Test stages) or continue a paused scan.
Pause	Pause current scan (whether Full Scan, Explore Only or Test Only). You can resume the scan later. You can also save a paused scan to continue at another time.
Re-Scan >	Rerun the current scan or scan stage. Select one of the sub-menu items: Re-Scan (Full): Delete all scan results, and run a full scan using the current configuration. Re-Explore: Delete all scan results and run an Explore stage only using the current configuration. Re-Test: Delete Test results and run a new Test stage using the current configuration and Explore results.
Explore Only	Run an Explore stage only, without following it with the Test stage.
Manual Explore	Explore your site manually. See “Using AppScan” on page 125
Explore Web Services (WSDL)	Opens Generic Service Client, so you can manually explore the application. The requests and responses are then used during the Test stage.
Test Only	Run a Test stage only (or continue a Test that was paused), without first running an Explore stage. This option is active only when there are already some Explore results.
Test Multi-Step Operations Only	If you have configured one or more Multi-Step Operations (see “Multi-Step Operations view” on page 84), and they constitute a significant subset of your site that you want to scan, you can test those sequences only. For details see “Scan Multi-Step Operations Only” on page 170 Note that Scan Expert does not run automatically before this function, even if configured to run before scans. If required, run it separately before Scan Multi-Step Operations Only (Tools > Run Scan Expert Evaluation).
Re-Test Issues Found	This option sends only the tests that revealed issues. This is a quick way of seeing whether issues found in the last scan have been fixed.
Clear All Scan Data	Delete all Explore and Test results, keeping only the Scan Configuration.
Change Host/Scheme/Port	If you have already recorded a login, multi-step operations, and/or a Manual Explore - and then the host, scheme or port of your Starting URL changes - requests and responses in these recordings must be updated and verified. Click Scan > Change Host/Scheme/Port to open a dialog box from which you can change the URL, and get AppScan to automatically update, verify and confirm the necessary changes. The dialog box shows the steps being performed, and indicates when each step is successful. If the update process does not complete successfully, the dialog box indicates which step failed, and gives you the option to save the changes and proceed manually, or undo all changes. Important: In some cases AppScan may update responses incorrectly, and part or all of the scan will fail. If that happens you will need to re-record the problematic procedures. Note: Although <i>Manual</i> Explore data is updated, <i>Automatic</i> Explore data, and scan results, are deleted when you change the Starting URL. Note: This option can be used to change <i>only</i> the host, scheme, or port of <i>only</i> the Starting URL. If you need to make <i>other</i> changes to the Starting URL, or change the host, scheme or port for one of the <i>Additional</i> Domains in the scan, you cannot use this option. Instead, save the scan as a template, and use that to create a new scan.

Command	Click to:
Run Scan Expert Evaluation	Scan Expert evaluates whether the current configuration is optimal for the application being scanned. (See “Scan Expert” on page 143) This option runs a full evaluation: Scan Expert will briefly explore the application, analyze its responses, and suggest changes to the configuration to get the best results.
Run Scan Expert Analysis Only	This option runs only the Analysis stage, and is active only if there are already some scan results on which to base the analysis. Scan Expert will analyze the current results to determine whether the configuration is optimal.
Scan Configuration	Define the properties of a scan. See “Scan configuration dialog box” on page 39

Tools menu

Provides various reporting and customization tools, including the IBM Security PowerTools.

Command	Click to:
Create Report	Create a report of the current scan. See Chapter 10, “Reports,” on page 209
Manual Test	Create and send a test variant manually to a selected URL. See “Manual tests” on page 200
Generate All Issue Information	(This option is active only if there are scan results:) Generates/updates Issue Information in the Issue Information tab for <i>all</i> issues. See “Issue Information tab” on page 190. Tip: To generate Issue Information <i>for a specific result only</i> , right-click the result in the Result pane, and select Generate Issue Information .
Scan Scheduler	Set time and frequency for automatic run of a scan.
User-Defined Tests	Define new tests for scans. See “User-Defined Tests” on page 248
Glass Box Agents	Define your server-side glass box agents so scans can include glass box scanning. See “Glass box scanning” on page 144
Extensions >	
Web Services Wizard (Open API)	Open the wizard to configure a web services scan based on one or more Open API (v2 and v3) description files (JSON or YAML). See Web Services Wizard
Start Pyscan	Open Pyscan to control AppScan using the Python interface. See “Pyscan” on page 269
Extension Manager	Open Extension Manager to manage add-on applications. See “Extension Manager” on page 268
PowerTools >	
Authentication Tester	Run Authentication Tester PowerTool to brute-force authentication tests.
Connection Test	Run Connection Test PowerTool to ping websites without using the ping protocol (which is blocked by many firewalls).
EncodeDecode	Run EncodeDecode PowerTool for string encoding/decoding between various formats.
ExpressionTest	Run ExpressionTest PowerTool to check regular expressions.
HTTP Request Editor	Run HTTP Request Editor PowerTool to edit and send HTTP requests, and view site responses.
External Tools...	Click to open a dialog box from which you can: • Adjust the order of the PowerTools on the Tools menu • Add additional external programs that can be opened from the Tools menu
Options	Customize AppScan operations. See “Options dialog box” on page 235

Help menu

Used for accessing documentation, getting support help, and getting a new license.

Command	Click to:
AppScan Help	Open Online Help; same as F1.
AppScan Getting Started	Open the PDF format Getting Started Guide (requires Adobe Acrobat Reader).
AppScan Training and Support Videos	Open a web page listing training and support videos.
Accessibility Controls	Adds the Accessibility menu as the first menu (before the File menu) on the menu bar.
Accessibility Documentation	Opens the online Help at the Accessibility Controls page.
AppScan Publications Library	Opens the IBM Security AppScan Publications Library page on the IBM website, containing a link to the AppScan User Guide. This is a printable, PDF version of the Help file (requires Adobe Acrobat Reader).
AppScan website	Open the IBM Security AppScan Standard website in a browser.
License	Install or request a new license. Import AppScan Enterprise license permissions.
Support >	Go into Extended Support Mode, Encrypt Support Files, or the Support Resources web page. Extended Support Mode creates detailed logs of a actions performed to enable troubleshooting by IBM Support team. This can slow down AppScan, it should only be enabled when needed.
AppScan Log	Open the system log file.
Check for Updates	Send a request for new security assessment knowledge to be integrated into your version of AppScan.
Update Log	Open a log of live update installations.
About IBM Security AppScan	View general product information.

Main toolbar

The icons on the toolbar offer quick access to frequently used features (that are also available from the menus).

Icon	Name	Click to:
	Scan >	(Available only if a scan is loaded and configured.) Opens a short Scan menu, with the following options:  Full Scan: Start a full scan (Explore and Test stages) or continue a paused scan.  Explore Only: Run an Explore stage only (or continue an Explore that was paused), without following it with the Test stage.  Test Only: Run a Test stage only (or continue a Test that was paused), without first running an Explore stage. Active only if there are already some Explore results.

Icon	Name	Click to:
	Pause	(Active only when a scan is running.) Pause current scan (whether Full Scan, Explore Only or Test Only). You can resume the scan later. You can also save a paused scan to continue at another time.
	Manual Explore	Open the browser to the application's URL and manually browse the site, completing required parameters as you go. AppScan will then add this Explore data to its own, automatically collected Explore data, when creating tests for the site. For details see "Using AppScan" on page 125.
	Configuration	Open the Scan Configuration dialog box to configure the scan. For details see "Scan configuration dialog box" on page 39.
	Report	Create a report with the current scan data. For details see "Report overview" on page 209.
	Scan on Cloud	Uploads an AppScan Standard configuration file (SCAN or SCANT) to Application Security on Cloud. You can use the configuration to run a full scan, or use existing Explore stage results saved in the file and run Test Only.
	Find	Search the results. For details see "Searching Results" on page 276
	Scan Log	Display the Scan Log during or after a scan. (Lists all actions performed by AppScan during the scan, as they occur.)
	PowerTools	Open one of the PowerTools, applications supplied with AppScan to help you with various tasks. For details, see "PowerTools" on page 251.

View Selector

The three icons on the right of the toolbar toggle between the three views: Application Data, Security Issues, and Remediation Tasks.

Icon	Name	Click to display:
	Data view	Application Data view. For details see Chapter 7, "Results: Application Data," on page 177
	Issues view	Security Issues view. For details see Chapter 8, "Results: Security Issues," on page 185
	Tasks view	Remediation Tasks view. For details see Chapter 9, "Results: Remediation Tasks," on page 205

Browser toolbar

The icons on the toolbar of the embedded AppScan browser, used to display and save screenshots of application responses.

Button	Description
	Back
	Next

Button	Description
	Stop
	Refresh
	Go to the default starting URL.
	Make the current page the starting URL.
	Capture this page (available only when the browser is opened from the Screenshot tab in the “Detail Pane” on page 21).

Keyboard shortcuts

AppScan utilizes the following keyboard shortcuts:

Shortcut	Function
F1	Open AppScan Online Help
F2	Open Data view
F3	Open Issues view
F4	Open Tasks view
F5	Start a full scan
[Shift] + F5	Pause scan
F10	Open the Configuration dialog box
[Ctrl] + N	Create a new scan
[Ctrl] + O	Open an existing scan
[Ctrl] + S	Save current scan
[Ctrl] + P	Print the current contents of the Application tree and Result list. (All nodes that can be seen by scrolling up or down the Application tree and Result list are included, but the nodes that are closed on the screen will appear closed in the printout.)
[Ctrl] + W	Open the welcome screen

Accessibility controls

Describes all keyboard shortcuts and controls.

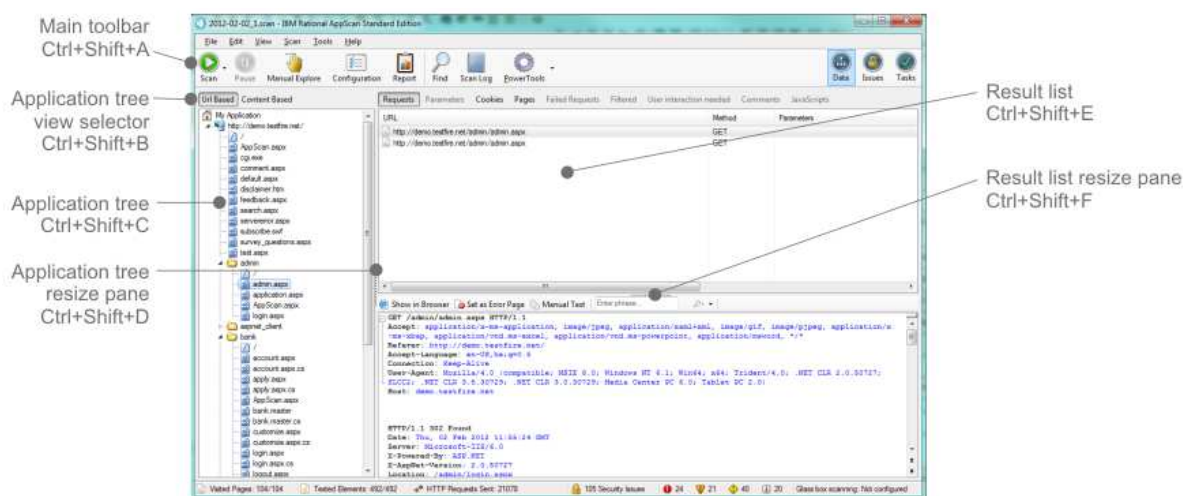
Full keyboard accessibility is activated by clicking **Help > Accessibility controls** so that it becomes checked. The Accessibility menu then appears to the left of the File menu, and additional keyboard control (described below) is enabled.

Accessibility menu

The Accessibility menu appears only when **Help > Accessibility controls** is checked, but the shortcuts listed in it work even when it is not checked. The first three sections control the user interface views. The remaining sections are keyboard shortcuts from other menus.

All views

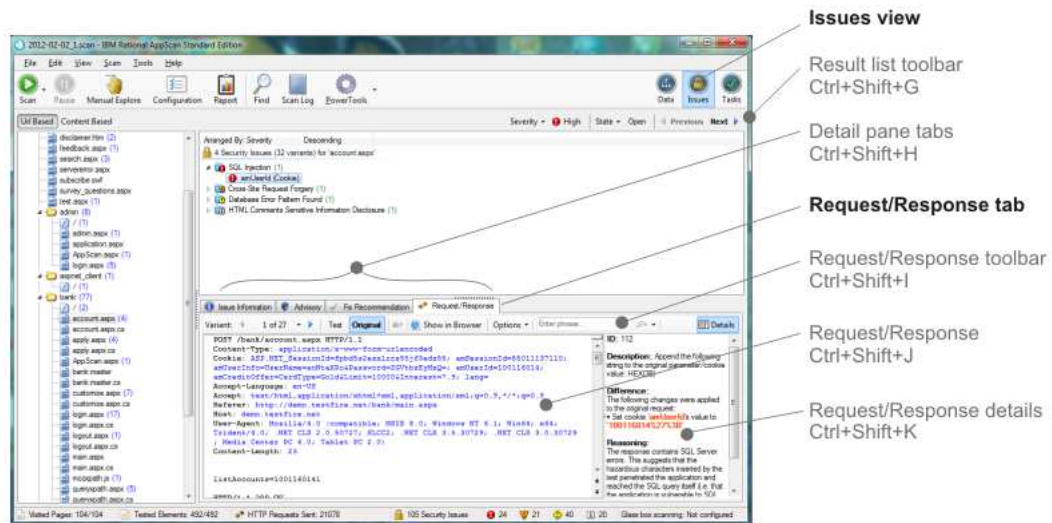
The following shortcuts apply to all views.



Command	Shortcut	Description
Main toolbar	Ctrl+Shift+A	Moves focus to the main toolbar. Use left and right arrows to select a toolbar icon. Note: The focus includes the three View Selector icons at the right end of the toolbar as well.
Application tree view selector	Ctrl+Shift+B	Moves focus to the Application tree view selector. Use left and right arrows to toggle between URL-based and Content-based views.
Application tree	Ctrl+Shift+C	Moves focus to the Application tree. Use up and down arrows to navigate. Use left and right arrows to open and close nodes.
Application tree resize	Ctrl+Shift+D	Moves focus to the border between the Application tree pane and the Result list. Use the arrow keys to adjust the panes.
Result list	Ctrl+Shift+E	Moves focus to the Result list. Use up and down arrows to navigate. Use left and right arrows to open and close nodes.
Result list resize pane	Ctrl+Shift+F	Moves focus to the border between the Result list and the Detail pane. Use the arrow keys to adjust the panes.

Issues view

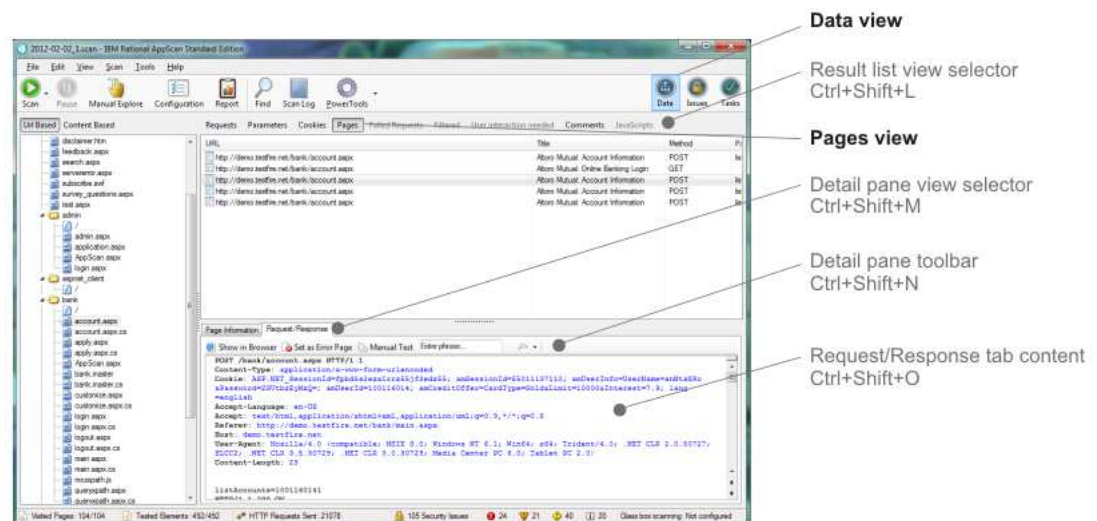
The following shortcuts apply to Issues view only.



Command	Shortcut	Description
Result list toolbar	Ctrl+Shift+G	In Issues view: Moves focus to the Result list. Use up and down arrows to navigate. Use left and right arrows to open and close nodes.
Detail pane	Ctrl+Shift+H	Moves focus to the Detail pane tabs.
Detail pane > Request/Response toolbar	Ctrl+Shift+I	In Issues view: Opens the Request/Response tab toolbar in the Detail pane, and moves focus to its toolbar.
Detail pane > Request/Response	Ctrl+Shift+J	In Issues view: Opens and moves focus to the Request/Response tab of the Detail pane.
Detail pane > Request/Response details	Ctrl+Shift+K	In Issues view: Opens and moves focus to the Detail section (right-hand pane) of the Request/Response tab in the Detail pane.

Data view

The following shortcuts apply to Data view only.



Command	Shortcut	Description
Result list view selector	Ctrl+Shift+L	In Data view: Moves focus to the Result list toolbar. Use left and right arrows to select Requests, Parameters, Cookies and so on.
Detail pane view selector	Ctrl+Shift+M	In Data view > Pages: Lets you select Page Information or Request/Response. Use left and right arrows to toggle.
Detail pane > Request/Response toolbar	Ctrl+Shift+N	In Data view > Pages: Moves focus to the Request/Response toolbar. Use left and right arrows to navigate.
Detail pane > Request/Response tab	Ctrl+Shift+O	In Data view > Pages: Moves focus to the content of the Request/Response tab.

Other keyboard shortcuts

The remaining shortcuts also exist in other menus, and are listed in “Keyboard shortcuts” on page 345

Additional keyboard control

When **Help > Accessibility controls** is enabled:

- When you use an Accessibility shortcut to move the focus to a different part of the user interface, the new focus is outlined briefly in red.



- At any time you can see where the current focus is, both in the main user interface and in the Configuration dialog box, by clicking F7. The current focus will be outlined briefly in blue.



Chapter 17. Glossary

This glossary explains terms and acronyms used in the AppScan Standard user interface and documentation.

For other terms and definitions, see the IBM Terminology website ([opens in new window](#)).

A

access control

In computer security, the process of ensuring that users can access only those resources of a computer system for which they are authorized.

action-based login

This type of login replay reproduces the actions performed when you recorded the login sequence, and is usually the preferred login method.

action-based login player

A browser with two panes in which action-based login is replayed for verification and troubleshooting. The left pane shows the list of actions and highlights which is currently being performed; the right pane shows the result of the current action.

advisory

A document that contains information and analysis about a threat or vulnerability.

application lifecycle

The succession of stages a product goes through, from development to production.

application server

A server program in a distributed network that provides the execution environment for an application program.

application test

A type of test which focuses on application logic and issues resulting from insecure software development.

application tree

A tree-view display of a web application's structure, including directories and files.

attack Any attempt by an unauthorized person to compromise the operation of a software program or networked system. See also [attacker](#).

attacker

A user (human or computer program), that attempts to cause harm to an information system or to access information not intended for general access. See also [hacker](#), [attack](#).

authentication

The process of validating the identity of a user or server.

Authentication Tester

A brute-force-like testing utility. One of the PowerTools. It detects weak username-password combinations that could be used to gain access to a user's web application.

authorization

The right granted to a user to communicate with or make use of a computer system.

B

back end

The set of support components of a computer system, such as the database management system.

black box

When the output of an application is examined without reference to its internal code, the application can be described as a "black box", and the testing as "black box testing", because it treats the application as a "black box" the contents of which cannot be seen. Compare with "white box" and "glass box".

broken link

A link that returns an invalid response when selected.

brute force

An attack by a program that tries every possible credential to compromise the security of a system.

buffer A reserved segment of memory used to hold data while it is being processed.

buffer overflow

An exploitation technique that alters the flow of an application by overwriting parts of memory. Buffer overflows are a common cause of malfunctioning software.

C**case-sensitive**

Pertaining to the ability to distinguish between uppercase and lowercase letters.

CGI See Common Gateway Interface.

character encoding

A character set consisting of a code that pairs a sequence of characters from a given set with something else, such as a sequence of natural numbers, octets or electrical pulses. Encoding facilitates the storage and transmission of text through telecommunication networks.

child node

A node within the scope of another node.

client The user's workstation that is connected to a network. See also host.

client-side

Pertaining to an operation that is performed on the client application and not on the server.

code injection

A technique that introduces new code into an application. Code injection can be used by an attacker to introduce code into a computer program to change the course of execution.

Common Gateway Interface (CGI)

An Internet standard for defining scripts that pass information from a web server to an application program, through an HTTP request, and vice versa.

communication timeout

The intentional ending of an incomplete task after waiting a specified amount of time.

concurrent login

A login that occurs simultaneously with other logins.

condition pattern

In regular expressions, a pattern that the regular expression defines. The regular expression can be used to find items that match the pattern.

cookie Information that a server stores on a client machine and accesses during subsequent sessions. Cookies allow servers to retrieve specific information about clients.

crawl To search for information across various web pages on the Internet or on an intranet.

cross-site scripting (XSS)

An attack technique that forces a website to echo client-supplied data, which execute in a user's web browser.

custom error page

A feature of most web server software that allows the user to replace default error messages with messages that are custom designed for the application.

CVE Common Vulnerabilities and Exposures. An industry standard list that provides common names for publicly known information security vulnerabilities and exposures.

CVSS Common Vulnerability Scoring System. An open framework for scoring the risk associated with vulnerabilities.

CWE Common Weakness Enumeration. An industry standard list that provides common names for publicly known software weaknesses.

D**database management system (DBMS)**

A software system that controls the creation, organization, and modification of a database and the access to the data that is stored within it.

database service

A service that provides the storage and retrieval of data in a database.

DBMS

See database management system.

debug command

A feature or command that assists in identifying programming errors during the software development process.

delta A difference, or an incremental value, between two instances.

denial-of-service attack (DoS)

In computer security, an assault on a network that brings down one or more hosts on a network such that the host is unable to perform its functions properly. Network service is interrupted for some period.

depth The number of clicks required for a user, or an automatic crawler, to get from a source page to a target page.

directory indexing

A web server feature that exposes contents of a directory when no index page is present.

directory traversal

A technique used to exploit websites by accessing files and commands beyond the document root directory.

domain

A subnetwork of clients and servers under the control of one security database.

DoS See denial-of-service attack.

dump file

The contents of memory without any report formatting.

E**embedded browser**

The web browser that is embedded in AppScan and opens with a special toolbar for working with scans.

encoding attack

An exploitation technique that aids an attack by changing the format of user-supplied data to bypass sanity checking filters.

encryption

The process of transforming data into an unintelligible form in such a way that the original data either cannot be obtained or can be obtained only by using a decryption process.

exclusion

A parameter or process whose values are excepted during tests.

executable

A program file that is ready to run in a particular environment.

Explore setting

A setting that configures the parameters governing how an application will be explored by AppScan.

Explore stage

The stage of an AppScan scan during which the logic and objects of an application are identified, prior to testing.

export To save a copy of the current document, database or image into the file format required by a different application.

extended support mode

A mode that allows the user to record usage options and behavior, and to save the data in a file to send to technical support.

F**false positive**

A test result classed as positive (indicating that the site is vulnerable to attack), that the user decides is in fact negative (not a vulnerability).

fix recommendation

The specific and technical details on fixing a web application to secure it against the issue that was discovered.

Flash A programming technique that enables movies and animation to display seamlessly in a web browser.

form property

A value used when forms are filled out automatically.

full path name

The name of any directory or file expressed as a string of directories and files beginning with the root directory.

G**glass box**

Unlike "black box testing", which treats the application as a "black box" the contents of which cannot be seen, "glass box" testing "looks inside the box" by examining application's code. This means that glass box tests may pinpoint security vulnerabilities that black box tests could not.

graphical user interface (GUI)

A type of computer interface that presents a visual metaphor of a real-world scene, often of a desktop, by combining high-resolution graphics, pointing devices, menu bars and other menus, overlapping windows, icons and the object-action relationship.

GUI See graphical user interface.

H

hard-coding

The software development practice of embedding output or configuration data directly into the source code of a program or other executable object.

hazardous character

A character which is used for performing web application attacks, such as XSS or SQL injection.

hidden parameter

An HTML form parameter that is not rendered in the web page.

host A computer that is connected to a network and that provides an access point to that network. The host can be a client, a server, or both a client and server simultaneously. See also client

HTML form element

An element that allows the user to enter information, such as text fields, text area fields, drop-down menus, radio buttons, or check boxes, in a form.

HTTP request

A request sent to the site either during the Explore or Test stage of the scan.

HTTP response

A response sent by the server.

I

ID See identifier.

identifier (ID)

One or more characters used to identify or name a data element and possibly to indicate certain properties of that data element.

import

To read a file in a format that is not native to the application in use.

Industry Standards report

A report of issues found on the user's web application and relevant information according to a selected industry standard. AppScan Industry Standard reports include SANS Top 20, OWASP Top 10, and WASC Threat Classification.

in-session detection

The detection of the in-session pattern in the responses AppScan receives, to verify that it is still logged in.

in-session pattern

A pattern identified in the login page, such as a logout link, that AppScan can use to verify that it is still logged in.

insufficient anti-automation

The result when a website permits an attacker to automate a process that should only be performed manually.

interactive URL

A URL that includes forms to be filled out manually by the user.

invasive test

An optional test which, if run on the application, may cause a denial-of-service situation.

issue A security risk to which a web application is vulnerable, or possibly sensitive information that is visible to unauthorized users.

J

Java applet

An applet that is written in Java, and that can run in a web browser using a Java virtual machine (JVM).

Java virtual machine (JVM)

A software implementation of a processor that runs compiled Java code (applets and applications).

L**link extraction**

The parsing or executing of code for discovery and collection of links from a web application.

login sequence

The sequence of user inputs that enables AppScan to log into your web application to scan it. It is recommended to record the login manually. AppScan then replays this sequence whenever it needs to log in during the scan. When you record a login sequence, AppScan analyzes both the actions and the requests. When replaying the login it attempts (by default) to reproduce the action-based login; if this is unsuccessful it will revert to the request-based login.

M**malware**

Malicious software or executable code, often downloaded or received in the form of an innocent-looking file.

manipulation

A modification by an attacker of a data element, group of elements, action, or group of actions based on one or more properties. For example, modification of input by removing a required argument, or performing steps out of order.

manual explore

The process of manually crawling a web application to access and test parts of the site that are dependent on input from a real user.

metacharacter

ASCII character with special meaning during pattern processing. Such characters are used to represent single-byte or multibyte character patterns that can be matched during processing.

multipart request

A request that contains more than one content type. To reduce unnecessary memory consumption, some content types are automatically filtered out of multipart requests during the scan. You can configure which types will not be filtered in **Configuration > Advanced Configuration > Multipart Content Type Filter**.

multiphase scan

A scan that consists of two or more phases.

multi-step operation

A sequence of two or more requests that must be sent in a specific order to access certain parts of the application. (Example: Add item to shopping cart > Enter payment details > Receive order confirmation.) Recording such multi-step operations as part of the scan configuration ensures that these parts of the site are scanned.

N**network service**

A service that transmits data or provides conversion of data in a network.

NTLM

See Windows NT LAN Manager.

numeric overflow

The result from an arithmetic calculation that exceeds the space designated to hold it.

P**parent node**

The node that contains the current node.

parse To break down a string of information, such as a command or file, into its constituent parts.

path The part of the URL which points to the location of an Internet resource.

path filtering

The process of filtering out or including pages according to set criteria.

path traversal

An attack technique that alters a document or resource location requested in a URL and forces access to files, directories, and commands that reside outside the web document root directory.

pattern

A method of describing text to be identified, using one or more regular expressions.

PCI See Peripheral Component Interconnect.

penetration test

A method of evaluating the security of a web application by simulating an attack by a hacker.

Peripheral Component Interconnect (PCI)

A local bus that provides a high-speed data path between the processor and attached devices.

permission

Authorization to perform activities, such as reading and writing local files, creating network connections, and loading native code.

personal identification number (PIN)

In Cryptographic Support, a unique number assigned by an organization to an individual and used as proof of identity. PINs are commonly assigned by financial institutions to their customers.

phase A process that includes the Explore stage followed by the Test stage of a scan.

phase limit

The maximum number of phases allowed in a scan. The limit is configurable.

PIN Personal Identification Number.

platform

The combination of an operating system and hardware that makes up the operating environment in which a program runs.

port An end point for communication between applications, generally referring to a logical connection. A port provides queues for sending and receiving data. Each port has a port number for identification.

port listener

A mechanism that allows the product to validate certain tests by listening to out-of-bound connections.

Predictable Resource Location

An attack technique used to uncover hidden website content and functionality. The attack searches for content in standard locations that is not intended for public viewing, such as temporary files, backup files, configuration files or sample files.

privilege escalation

The process of referring to scans that were run using different user privileges, in order to test whether privileged resources are accessible to users with insufficient access permissions.

prompt

A message or a displayed symbol that requests information or user action. The user must respond to allow the program to proceed.

proxy

An application gateway from one network to another for a specific network application such as Telnet or FTP, for example, where a firewall's proxy Telnet server performs authentication of the user and then lets the traffic flow through the proxy as if it were not there. Function is performed in the firewall and not in the client workstation, causing more load in the firewall.

R**redundant path limit**

The maximum number of times identical paths may be scanned in a scan, in order to reduce scan time and eliminate duplicate results.

regular expression

A set of characters, meta characters, and operators that define a string or group of strings in a search pattern.

regulatory compliance report

A report of issues found on a web application that do not comply with a selected regulation or legal standard. The regulations include legal Acts, Bills, and Laws of Canada, EU, Japan, UK, USA, and regulations of MasterCard and Visa. Custom regulatory compliance report templates can also be created.

relative path

A path that begins with the current working directory.

remediation

A suggestion for how to fix an issue.

request-based login

This type of login replay reproduces the requests that were sent when you recorded the login sequence.

restriction

A type of filter that limits a scan to listed URLs only.

Result Expert

An optional function that can be run after scanning to add CVSS settings, screen captures, and other information to the Issue Information tab of scan results.

reverse engineer

To analyze a device or system in order to learn details of its design, construction, and operation.

risk analysis

An analysis of the security issues found in a web application.

risk assessment

An evaluation of the benefits and consequences of an action or scenario.

risk management

The optimal allocation of resources to arrive at a cost-effective investment in defensive measures within an organization.

role

A set of permissions.

S**sanitize**

In web application security, to clean user input from harmful or hazardous characters, before using it.

scan

The process of AppScan exploring and testing an application and providing the results.

scan configuration

A collection of AppScan settings that define the user's application/service, environment, and chosen scan methods.

Scan Expert

An optional function that explores application and network behavior, and recommends configuration changes to optimize scanning.

Scan Expert analysis module

A single check done by Scan Expert during its analysis.

Scan Expert evaluation

Scan Expert's evaluation of the user's configuration.

scan template

A scan configuration that can be loaded to use for a scan.

scheduler

A multithread, multiprocess background server designed to handle the scheduling and launching of jobs, based on a simple timing scheme.

security audit

A manual or systematic measurable technical assessment of a system or application.

security risk

The potential success of a threat and the damage that could ensue.

sequence

A list of recorded URLs.

session

A logical or virtual connection between two stations, software programs, or devices on a network that allows the two elements to communicate and exchange data. See also transaction

session credential

A string of data provided by the web server, stored within a cookie or URL, which identifies a user and authorizes that user to perform various actions.

session fixation

An attack technique that allows an attacker to fixate a user's session identifier and assume their online identity.

session hi-jacking

The compromise of a user's session by an attacker. The attacker could reuse this stolen session to masquerade as the user.

session ID

See session identifier

session identifier (session ID)

The compromise of a user's session by an attacker. The attacker could reuse this stolen session to masquerade as the user.

session token

An identifier that is sent by the browser as a parameter or a cookie, in order to correlate between a user and their current session on the web application. See also session identifier, transient token.

severity rating

The level assigned to an issue by the scan, indicating the security risk it represents.

shell

A software interface between users and an operating system. Shells generally fall into one of two categories: a command line shell, which provides a command line interface to the operating system; and a graphical shell, which provides a graphical user interface (GUI).

source code

A computer program in a format that is readable by people. Source code is converted into binary code that can be used by a computer.

spoofing

The technique of faking the sending address of a transmission in order to gain illegal entry into a secure system.

SQL See Structured Query Language.

SQL injection

See Structured Query Language injection.

stage Part of a scan phase in which AppScan either explores or tests the site.

stateless protocol

A protocol that does not maintain a relationship between commands. HTTP is an example of a stateless protocol.

Structured Query Language (SQL)

A standardized language for defining and manipulating data in a relational database.

Structured Query Language injection (SQL injection)

An attack technique used to exploit websites by altering back-end SQL statements through manipulating application input.

syntax The rules for the construction of a command or statement.

T**test fix**

A temporary fix that is supplied to specific customers for testing in response to a reported problem.

test policy

A policy that limits the scan to certain categories and types of tests.

Test request

A request sent to the application during the Test stage of the scan. Test requests are designed to reveal security vulnerabilities.

Test stage

The stage of the scan during which the objects and logic of the scanned application are submitted to a comprehensive barrage of typical, erroneous, and simulated-malicious usage techniques, resulting in a complete inventory of security vulnerabilities.

thread A stream of computer instructions that is in control of a process. In some operating systems, a thread is the smallest unit of operation in a process. Several threads can run concurrently, performing different jobs.

threat A security issue, or a harmful act, such as the deployment of a virus or illegal network penetration.

threat class

A group of security issues, classed together by WASC-TC categories. For each threat class, there are numerous specific tests; and for each test, numerous variants.

transaction

A request (to an application) and the response (from the application) that it generated.

transient token

A token whose value changes (usually a session token). Sending an expired transient token could result in AppScan getting logged out of the application it is testing, so it must keep them up to date. See also session token.

U

Uniform Resource Locator (URL)

The unique address of an information resource that is accessible in a network such as the Internet. The URL includes the abbreviated name of the protocol used to access the information resource and the information used by the protocol to locate the information resource.

UNIX A highly portable operating system that features multiprogramming in a multiuser environment. The UNIX operating system was originally developed for use on minicomputers, but was adapted for mainframes and microcomputers. The AIX operating system is IBM's implementation of the UNIX operating system.

URL See Uniform Resource Locator.

user-defined test

A test that is created by a user in addition to the tests that are automatically created and run.

V

validation

The process of verifying whether a certain test succeeded or failed to achieve its goal.

vulnerability

A security exposure in an operating system, system software, or application software component.

W

web application

An application that is accessible by a web browser and that provides some function beyond static display of information, for instance by allowing the user to query a database. Common components of a web application include HTML pages, JSP pages, and servlets.

web browser

A client program that initiates requests to a web server and displays the information that the server returns.

web content

Files and other resources that compose a website. web content may consist of image files, audio files, HTML files, JSP files, style sheets, database entries, or anything you can see on a website.

web security

The theory and practice of information security relating to the World Wide web, HTTP and web application software.

web server

A software program that is capable of servicing Hypertext Transfer Protocol (HTTP) requests.

web service

An application that performs specific tasks and is accessible through open protocols such as HTTP and SOAP.

Web Services Description Language (WSDL)

An XML-based specification for describing networked services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information.

white box

White box scanning analyzes actual code, such as JavaScript code in the case of Static Analysis. Compare with "black box" and "glass box".

Windows NT LAN Manager (NTLM)

A protocol used in a variety of Microsoft network protocols for authentication purposes.

WSDL

See Web Services Description Language.

X

XSS See cross-site scripting.

Chapter 18. Notices

© Copyright IBM Corporation 2000, 2016. © Copyright HCL Limited 2017, 2018. All rights reserved. U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

This information was developed for products and services offered in the US.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan*

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (your company name) (year).

Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. 2000, 2017.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at <http://www.ibm.com/privacy> and IBM's Online Privacy Statement at <http://www.ibm.com/privacy/details> in the section entitled "Cookies, Web Beacons and Other Technologies," and the "IBM Software Products and Software-as-a-Service Privacy Statement" at <http://www.ibm.com/software/info/product-privacy>.



Printed in USA