IBM Security AppScan Source for Analysis

バージョン 9.0.3.7

ユーザー・ガイド

IBM Security AppScan Source for Analysis

バージョン 9.0.3.7

ユーザー・ガイド

IBM

目次

第 1 章 AppScan Source for Analysis の概要. 1

IBM Security AppScan Source の概要 1
米国政府の規制の準拠 2
AppScan Source の新機能. 4
AppScan Source バージョン 9.0.3.7 の新機能 . . 5
AppScan Source バージョン 9.0.3.6 の新機能 . . 5
AppScan Source バージョン 9.0.3.5 の新機能 . . 5
AppScan Source バージョン 9.0.3.4 の新機能 . . 6
AppScan Source バージョン 9.0.3.3 の新機能 . . 9
AppScan Source バージョン 9.0.3.2 の新機能. . 11
AppScan Source バージョン 9.0.3.1 の新機能. . 12
AppScan Source バージョン 9.0.3 の新機能 . . 12
現行バージョンの AppScan Source へのマイグレーション 16
バージョン 9.0.2 からのマイグレーション . . . 16
バージョン 9.0 からのマイグレーション. . . . 18
バージョン 8.7 からのマイグレーション. . . . 18
AppScan Source for Analysis の概要 20
ワークフロー. 20
重要な概念 21
分類. 22
AppScan Source 製品から AppScan Enterprise Server へのログイン 23
Common Access Card (CAC) 認証の有効化 . . 26
AppScan Source ユーザー・パスワードの変更 28
AppScan Enterprise Server の SSL 証明書 . . 28
AppScan Source とアクセシビリティー. 29
特記事項 30
著作権 33

第 2 章 アプリケーションおよびプロジェクトの構成 35

AppScan Source アプリケーションおよびプロジェクト・ファイル 35
アプリケーションの構成 39
新規アプリケーション・ウィザードによる新規アプリケーションの作成. 40
アプリケーション・ディスカバリー・アシスタントを使用したアプリケーションおよびプロジェクトの作成 41
既存のアプリケーションの追加. 45
複数のアプリケーションの追加. 47
Apache Tomcat および WebSphere Application Server Liberty プロファイル・アプリケーション・サーバーからの既存の Java アプリケーションのインポート 48
Eclipse または Eclipse ベースの製品ワークスペースの追加 51
Eclipse プロジェクトおよび Rational Application Developer for WebSphere Software (RAD) プロジェクトの開発環境の構成 52
Eclipse または Application Developer の更新 . 52
Eclipse ワークスペース・インポーター: Eclipse または Rational Application Developer for WebSphere Software (RAD) の設定構成 . . . 53
アプリケーションの新規プロジェクトの作成 . . . 54
既存のプロジェクトの追加 54
複数のプロジェクトの追加 56
新規 Arxan プロジェクトの追加 58
新規 ASP プロジェクトの追加. 59
新規 C/C++ プロジェクトの追加. 61
新規 COBOL プロジェクトの追加. 63
新規 ColdFusion プロジェクトの追加 64
新規 Java または JavaServer Page (JSP) プロジェクトの追加. 65
新規 JavaScript プロジェクトの追加. 73
新規 .NET アセンブリー・プロジェクトの追加 74
新規パターン・ベース・プロジェクトの追加 . . 75
新規 Perl プロジェクトの追加. 76
PHP プロジェクト構成 77
新規 PL/SQL プロジェクトの追加 90
新規 T-SQL プロジェクトの追加 91
新規 Visual Basic プロジェクトの追加 92
プロジェクトのコピー. 93
アプリケーションおよびプロジェクトのプロパティーの変更 93
グローバル属性 94
アプリケーション属性. 95
アプリケーションおよびプロジェクトの削除 . . . 95
「エクスプローラー」ビュー 96

第 3 章 設定. 103

全般設定. 103
AppScan Enterprise Console の設定 106
JavaServer Page コンパイル用のアプリケーション・サーバー設定. 107
Tomcat 108
WebLogic 11 および 12. 108
WebSphere Application Server 109
変数の定義 109
設定による障害追跡の有効化 110
Rational ClearQuest の設定 110
Quality Center の設定 111
Rational Team Concert の設定 113
Team Foundation Server の設定. 115
Eclipse ワークスペース・インポーター: Eclipse または Rational Application Developer for WebSphere Software (RAD) の設定構成 115
E メール. 116

Java および JavaServer Pages. 116
ナレッジベース・データベースの記事 117
プロジェクト・ファイル拡張子 117

第 4 章 ソース・コードのスキャンおよび評価の管理 119
ソース・コードのスキャン 119
すべてのアプリケーションのスキャン 120
1 つ以上のアプリケーションのスキャン . . . 120
1 つ以上のプロジェクトのスキャン 120
1 つ以上のファイルのスキャン 121
コードの再スキャン 121
スキャンに関する考慮事項 121
スキャン構成の管理 123
Java の増分分析 132
スキャンからのファイルの除外 135
スキャンのキャンセルまたは停止. 135
Linux での AppScan Source for Analysis および AppScan Source for Development (Eclipse プラグイン) コンポーネントの前提条件 . . . 136
「自分の評価」の管理 137
分析のためのクラウドへの AppScan Source 評価の送信. 138
評価の公開 143
AppScan Source に公開するためのアプリケーションおよびプロジェクトの登録. 144
AppScan Source への評価の公開. 145
AppScan Enterprise Console への評価の公開 146
評価の保存 152
評価の自動保存. 153
「自分の評価」からの評価の削除. 153
変数の定義 153
公開時および保存時の変数の定義. 154
例: 変数の定義 155

第 5 章 トリアージおよび分析 157
検出結果の表示. 158
AppScan Source トリアージ・プロセス 160
トリアージ例 161
フィルターを使用したトリアージ. 163
AppScan Source 事前定義フィルターの使用 169
フィルターの作成および管理 175
フィルターの適用 181
除外を使用したトリアージ 183
除外の有効範囲. 183
除外の指定 184
検出結果表で検出結果に除外としてマークを付ける 184
除外としてマークされた検出結果の再組み込み 185
例: フィルター除外の指定 185
「プロパティー」ビューからのバンドル除外の指定 186
バンドルを使用したトリアージ 187
バンドルの作成. 187
既存のバンドルへの検出結果の追加 188
バンドル内の検出結果の表示 189
バンドルのファイルへの保存 190
バンドルの障害追跡への送信および E メールによる送信 190
バンドルへの注釈の追加. 191
検出結果の変更. 191
検出結果表からの変更の実行 191
「検出結果の詳細」ビューでの検出結果の変更 193
検出結果の変更の取り消し 195
検出結果の比較. 197
「差分評価」ビューでの 2 つの評価の比較 . . 197
メインメニュー・バーからの 2 つの評価の比較 197
「自分の評価」ビューと「公開された評価」ビューでの評価の差分の検出. 197
カスタム検出結果 198
「プロパティー」ビューでのカスタム検出結果の作成 199
検出結果ビューでのカスタム検出結果の作成 . . 200
ソース・コード・エディターでのカスタム検出結果の作成 201
セキュリティー問題の解決と修復支援の表示 . . . 202
エディターでのソース・コードの分析 202
サポートされる注釈と属性 203

第 6 章 AppScan Source トレース 207
AppScan Source トレース スキャン結果 208
検証とエンコード 208
AppScan Source トレース の検索 209
入出力トレース. 209
「トレース」ビューの使用 210
「トレース」ビューの入出力スタック 211
エディターでのソース・コードの分析 214
検証とエンコードの有効範囲 215
AppScan Source トレース によるカスタム・ルールの作成 215
トレースのコード例 218
例 1: ソースからシンク 218
例 2: ソースからシンクへの変更 220
例 3: ソースとシンクのファイルが異なる場合 225
例 4: 詳細な検証 226

第 7 章 AppScan Source for Analysis および障害追跡 229
設定による障害追跡の有効化 229
Rational ClearQuest の設定 229
Quality Center の設定 230
Rational Team Concert の設定 232
Team Foundation Server の設定. 233
HP Quality Center と AppScan Source for Analysis の統合 234
Quality Center への検出結果の送信. 234
Quality Center に送信された検出結果の追跡 234
Quality Center における AppScan Source の検出結果情報 235
Rational ClearQuest と AppScan Source for Analysis の統合 235
Rational ClearQuest への検出結果の送信 . . . 236

Rational ClearQuest への障害の送信 236
Rational Team Concert と AppScan Source for Analysis の統合 236
Rational Team Concert への障害の送信 . . . 237
Rational Team Concert の SSL 証明書 . . . 238
Microsoft Team Foundation Server と AppScan Source for Analysis の統合 238
Microsoft Team Foundation Server への障害の送信 238
送信された障害の操作 239
バンドルの障害追跡への送信および E メールによる送信. 239
E メールによる障害の追跡 (E メールによる検出結果の送信). 240

第 8 章 検出結果レポートと監査レポート 243
検出結果レポートの作成. 243
AppScan Source レポート 245
AppScan Source カスタム・レポートの作成 246
CWE/SANS Top 25 2011 レポート. 248
DISA Application Security and Development STIG V3R10 レポート 248
Open Web Application Security Project (OWASP) Top 10 2013 レポート 249
Open Web Application Security Project (OWASP) Mobile Top 10 レポート. 249
Payment Card Industry Data Security Standard (PCI DSS) バージョン 3.2 レポート . 249
Software Security Profile レポート 249

第 9 章 カスタム・レポートの作成. . . 251
レポート・エディター 251
「レポート・レイアウト」タブ 252
「カテゴリー」タブ 254
「プレビュー」タブ 255
カスタム・レポートの生成 256
既存のカスタム・レポートからのレポートの設計 256
レポートへのカテゴリーの組み込み 256
レポートのプレビュー 258
レポート・テンプレートの保存 258

第 10 章 脆弱性データベースとパターン・ルールのカスタマイズ 259
AppScan Source セキュリティー・ナレッジ・データベース の拡張 259
カスタム・ルールの作成. 260
カスタム・ルール・ウィザードの使用 260
Likelihood ルール属性 266
AppScan Source トレースによる入出力トレースのカスタマイズ 267
パターン・ベースのルールによるカスタマイズ . . 268
パターン・ルール・セット 268
パターン・ルール 271
パターン・ルールおよびパターン・ルール・セットの適用 275

第 11 章 アプリケーション・サーバーのインポート・フレームワークの拡張. . 289

第 12 章 AppScan Source for Analysis サンプル 295

第 13 章 AppScan Source for Analysis の作業環境 297
AppScan Source for Analysis のワークベンチ . . 297
メインメニュー. 299
「ファイル」メニュー 299
「編集」メニュー 304
「スキャン」メニュー 306
「ツール」メニュー 306
「管理」メニュー 307
「表示」メニュー 307
「パースペクティブ」メニュー 308
「ヘルプ」メニュー 308
ツールバー 309
吹き出しヘルプ. 309
ステータス・バー 310

第 14 章 ビュー 311
構成ビュー 311
「カスタム・ルール」ビュー 311
「エクスプローラー」ビュー 311
「パターン・ルール・ライブラリー」ビュー . . 318
「プロパティー」ビュー. 319
「スキャン構成」ビュー. 329
レポート・エディター 332
スキャン出力に役立つビュー 336
「コンソール」ビュー 336
「メトリック」ビュー 337
「自分の評価」ビュー 337
「公開された評価」ビュー 338
トリアージに役立つビュー 339
「差分評価」ビュー 339
「カスタム検出結果」ビュー 340
検出結果を含むビュー 340
「ソースとシンク」ビュー 349
単一の検出結果の調査に使用できるビュー. . . . 350
「検出結果の詳細」ビュー 350
「修復支援」ビュー 353
「トレース」ビュー 353
評価の操作に使用できるビュー 355
「評価の概要」ビュー 355
「フィルター・エディター」ビュー 356
「脆弱性マトリックス」ビュー 357
「バンドル」ビュー 359
「バンドル」ビュー 359

第 15 章 CWE サポート. 365

用語集. 367
A 367

B 367
C 367
D 368
E 368
F 368
L 368
P 368
R 368
S 369
T 369
V 369
W 369
X 369

特記事項 371

索引 375

第 1 章 AppScan Source for Analysis の概要

このセクションでは、AppScan® Source for Analysis を総合的な AppScan Source ソリューションに合わせて調整する方法と、ソフトウェア・アシュアランス・ワークフローについて理解するための基礎知識について説明します。

IBM Security AppScan Source の概要

IBM® Security AppScan Source は、ソフトウェア・セキュリティーに携わる組織内のすべてのユーザーに対して、最大の価値を提供します。 AppScan Source 製品により、セキュリティー・アナリスト、品質保証専門家、開発者、経営幹部などの職種に関わらず、それぞれのユーザーが必要とする機能、柔軟性、処理能力がデスクトップにもたらされます。

この製品ファミリーの製品を以下に示します。

- **AppScan Source for Analysis**: アプリケーションとプロジェクトの構成、コードのスキャン、分析、トリアージ、優先度の高い脆弱性に対するアクションを実施するためのワークベンチです。
- **AppScan Source for Automation**: AppScan Source ワークフローの主要なステップを自動化し、ソフトウェア開発ライフサイクルを通じてビルド環境にセキュリティーを統合します。
- **AppScan Source for Development**: Developer Plug-in によって、AppScan Source for Analysis のさまざまな機能を Microsoft Visual Studio、Eclipse ワークベンチ、および Rational® Application Developer for WebSphere® Software (RAD) に統合します。これにより、ソフトウェア開発者は、開発プロセスで脆弱性を検出して対応することができます。Eclipse プラグインを使用することで、ソース・コードをスキャンしてセキュリティーの脆弱性を検出することができます。また、IBM MobileFirst Platform プロジェクトを Eclipse プラグインでスキャンできます。

これらの製品には、組織における AppScan Source の価値をさらに高めるための以下のコンポーネントが用意されています。

- **AppScan Source** セキュリティー・ナレッジ・データベース: 各脆弱性に関するコンテキストに沿った情報が格納されています。根本的な原因、リスクの重大度、実行可能な修復アドバイスに関する正確な説明を提供します。
- **AppScan Enterprise Server**: AppScan Source の大半の製品およびコンポーネントは、AppScan Enterprise Server と通信を行う必要があります。このコンポーネントがなくても、AppScan Source for Development をローカル・モードで使用できますが、カスタム・ルール、共有スキャン構成、および共有フィルターなどの機能は使用できません。

  このサーバーは、一元的ユーザー管理機能と、AppScan Source データベースを介して評価を共有するためのメカニズムを提供します。サーバーには、オプションとして Enterprise Console コンポーネントが含まれています。管理者がこのコンポーネントをインストールしている場合は、評価を AppScan Source for

Analysis、AppScan Source for Automation、および AppScan Source コマンド行インターフェース (CLI) からそのコンポーネントに公開できます。Enterprise Console は、レポート機能、問題管理、トレンド分析、ダッシュボードなど、評価に関する作業を行うためのさまざまなツールを備えています。

重要: AppScan Source および AppScan Enterprise の一部のバージョンでは、AppScan Source から AppScan Enterprise Server に接続するために 2 つの製品のバージョンとリリース・レベルが一致していなければなりません。http://www.ibm.com/support/docview.wss?uid=swg21975211 を参照して、どのバージョンの AppScan Source と AppScan Enterprise に互換性があるかを確認してください。

注:

- AppScan Enterprise Server は、macOS ではサポートされていません。
- 基本サーバー・ライセンスを保有している場合、AppScan 製品から同時にサーバーにアクセスできる接続数は最大で 10 までです。プレミアム・サーバー・ライセンスを保有している場合は、接続数に制限はありません。

重要: スキャン中は、AppScan Enterprise Server および AppScan Source のクライアント (AppScan Source for Development 以外) はいずれも、AppScan Source データベース (solidDB® または Oracle のいずれか) への直接接続が必要です。

このソフトウェア・オファリングは個人情報を収集するための Cookie またはその他の技術を使用していません。

翻訳済みの各国語

AppScan Source ユーザー・インターフェースは、次の言語で使用できます。

- 英語
- ブラジル・ポルトガル語
- 中国語 (簡体字)
- 中国語 (繁体字)
- ドイツ語
- スペイン語
- フランス語
- イタリア語
- 日本語
- 韓国語
- ロシア語

米国政府の規制の準拠

米国政府によるセキュリティーおよび情報技術の規制への準拠は、営業における障害を取り除く上で役立ちます。また、これにより IBM が自社製品を業界内で最もセキュアなものにするべく取り組んでいるというプルーフ・ポイント (証明) を、世界中の見込み客に提示することができます。このトピックでは、AppScan Source がサポートする規格とガイドラインについて示します。

- 『インターネット・プロトコル・バージョン 6 (IPv6)』
- 『連邦情報処理標準 (FIPS)』
- 『米国連邦情報・技術局 (NIST) Special Publication (SP) 800-131a』
- 4 ページの『United States Government Configuration Baseline (USGCB) を使用するように構成されている Windows 7 マシン』

インターネット・プロトコル・バージョン 6 (IPv6)

AppScan Source は IPv6 に対して有効ですが、以下の例外があります。

- IPv6 数値アドレスの入力がサポートされておらず、代わりにホスト名を入力する必要があります。IPv4 数値アドレスの入力はサポートされています。
- Rational Team Concert™ に接続する場合は、IPv6 はサポートされません。

連邦情報処理標準 (FIPS)

AppScan Source でサポートされている Windows プラットフォームおよび Linux プラットフォーム上では、AppScan Source は、FIPS 140-2 の認定済み暗号モジュールと承認されたアルゴリズムを使用することで、FIPS 出版物 140-2 に対応しています。AppScan Source でサポートされている macOS プラットフォームの場合、FIPS 140-2 モードで操作するには、手動ステップが必要です。

AppScan Source FIPS 準拠に関する背景情報を学習し、AppScan Source FIPS 140-2 モードを有効および無効にする方法を確認するには、以下の技術情報を参照してください。

- macOS 上での FIPS 140-2 モードでの AppScan Source バージョン 8.7 以降の操作
- AppScan Source で FIPS 140-2 モードを有効化/無効化/検証する方法 (Linux および Windows)
- AppScan Source バージョン 8.7 以降の FIPS 140-2 サポートに関する背景情報

米国連邦情報・技術局 (NIST) Special Publication (SP) 800-131a

NIST SP 800-131A ガイドラインは、暗号鍵管理に関する指示を提供します。次のようなガイドラインが含まれます。

- 鍵管理の手順
- 暗号アルゴリズムの使用方法
- 使用するアルゴリズムとそれらの最小強度
- セキュア通信のための鍵の長さ

政府機関および金融機関は、製品が指定のセキュリティー要件に準拠していることを保証するために NIST SP 800-131A ガイドラインを使用します。

NIST SP 800-131A は、AppScan Source が FIPS 140-2 モードで作動している場合のみサポートされます。AppScan Source FIPS 140-2 モードを有効および無効にする方法については、『連邦情報処理標準 (FIPS)』 を参照してください。

重要: 接続先の AppScan Enterprise Server で NIST 800-131a コンプライアンスが有効になっている場合、AppScan Source を設定してトランスポート層セキュリ

ティー V1.2 を強制する必要があります。トランスポート層セキュリティー V1.2 が強制されない場合、サーバーへの接続は失敗します。

- AppScan Source データベースをインストールしていない場合 (例えば、クライアント・コンポーネントのみをインストールしている場合など)、<data_dir>¥config¥ounce.ozsettings (<data_dir> は、ご使用の AppScan Source プログラム・データの場所です。詳しくは、 360 ページの『インストールとユーザー・データ・ファイルの場所』 を参照してください。) を変更してトランスポート層セキュリティー V1.2 を強制することができます。このファイルで、以下の設定を見つけます。

```
<Setting
   name="tls_protocol_version"
   read_only="false"
   default_value="0"
   value="0"
   description="Minor Version of the TLS Connection Protocol"
   type="text"
   display_name="TLS Protocol Version"
   display_name_id=""
   available_values="0:1:2"
   hidden="false"
   force_upgrade="false"
 />
```

  この設定で、value="0" を value="2" に変更し、ファイルを保存します。

- AppScan Source データベースをインストールしている場合は、AppScan Source と Enterprise Server の両方をインストールした後に、IBM Security AppScan Enterprise Server データベース構成ツールでトランスポート層セキュリティー V1.2 を強制します。

United States Government Configuration Baseline (USGCB) を使用するように構成されている Windows 7 マシン

AppScan Source は、USGCB 仕様で構成されている Windows 7 マシン上のアプリケーションのスキャンをサポートしています。

注: USGCB 仕様で構成されているマシンでは、AppScan Source は、障害追跡システムと、HP Quality Center または Rational ClearQuest® との統合をサポートしていません。

AppScan Source の新機能

以下の、AppScan Source に追加された新機能と、このリリースで非推奨になった機能についての注意について説明します。

- 5 ページの『AppScan Source バージョン 9.0.3.7 の新機能』
- 5 ページの『AppScan Source バージョン 9.0.3.6 の新機能』
- 5 ページの『AppScan Source バージョン 9.0.3.5 の新機能』
- 6 ページの『AppScan Source バージョン 9.0.3.4 の新機能』
- 9 ページの『AppScan Source バージョン 9.0.3.3 の新機能』
- 11 ページの『AppScan Source バージョン 9.0.3.2 の新機能』
- 12 ページの『AppScan Source バージョン 9.0.3.1 の新機能』

- 12 ページの『AppScan Source バージョン 9.0.3 の新機能』

AppScan Source バージョン 9.0.3.7 の新機能

- 『拡張および新規のスキャン・サポート』
- 『AppScan Source バージョン 9.0.3.7 でサポートされなくなった機能およびフィーチャー』

拡張および新規のスキャン・サポート

- Red Hat Enterprise Linux (RHEL) バージョン 7.3 がサポート対象のオペレーティング・システムになりました。
- Visual Studio 2015 への AppScan Source for Development Visual Studio プラグイン の適用がサポートされるようになりました。

AppScan Source バージョン 9.0.3.7 でサポートされなくなった機能およびフィーチャー

AppScan Source バージョン 9.0.3.7 では次のような変更が行われました。

- OS X バージョン 10.10 は、サポート対象のオペレーティング・システムではなくなりました。
- Xcode バージョン 6.3 はサポートされなくなりました。このバージョンの Xcode による Objective-C プロジェクトのスキャンはサポートされなくなりました。
- Tomcat バージョン 5 および 6 はサポートされなくなりました。

AppScan Source バージョン 9.0.3.6 の新機能

- 6 ページの『拡張および新規のスキャン・サポート』
- 『AppScan Source バージョン 9.0.3.6 でサポートされなくなった機能およびフィーチャー』

拡張および新規のスキャン・サポート

- Objective-C 用 Xcode 8.1 と 8.2 (iOS アプリケーションのみ) は macOS でサポートされるコンパイラーになりました。Xcode のこれらのバージョンのサポートは、AppScan Source バージョン 9.0.3.5 で遡ります。

AppScan Source バージョン 9.0.3.6 でサポートされなくなった機能およびフィーチャー

AppScan Source バージョン 9.0.3.6 では次のような変更が行われました。

- Red Hat Enterprise Linux バージョン 5 は、サポート対象のオペレーティング・システムではなくなりました。
- Oracle WebLogic サーバー バージョン 8、9、および 10 は、サポート対象のコンパイラーではなくなりました。

AppScan Source バージョン 9.0.3.5 の新機能

- 6 ページの『拡張および新規のスキャン・サポート』

- 『Java ソースおよびバイトコードのより効率的かつ短時間での再スキャンを可能にする増分スキャンのサポート』

拡張および新規のスキャン・サポート

- macOS バージョン 10.12 がサポート対象のオペレーティング・システムになりました。 macOS バージョン 10.12 のサポートは、AppScan Source バージョン 9.0.3.4 まで遡ります。
- Objective-C 用 Xcode 8.0、8.1、および 8.2 (iOS アプリケーションのみ) は、macOS でサポートされるコンパイラーになりました。

Java ソースおよびバイトコードのより効率的かつ短時間での再スキャンを可能にする増分スキャンのサポート

バージョン 9.0.3.5 では、Windows および Linux で Java 増分スキャンのサポートを有効にできるようになりました。増分分析が有効にされている場合、AppScan Source によって分析データがキャッシュに入れられます。その後、プロジェクトあるいはアプリケーションを再スキャンすると、AppScan Source は、このデータを使用してコードの変更を判別し、その変更によって影響を受けるコードの部分のみが再度分析されます。これにより、コードの分析は完全に行われますが、時間は短縮されます。

この機能は、IBM Security AppScan Source for Analysis、AppScan Source for Development Eclipse プラグイン、IBM Security AppScan Source for Automation、または IBM Security AppScan Source コマンド行インターフェース (CLI) の使用時にサポートされます。

AppScan Source バージョン 9.0.3.4 の新機能

- 『拡張および新規のスキャン・サポート』
- 7 ページの『共通アクセス・カード (CAC) によって認証される場合、AppScan Enterprise Console への評価の公開がサポートされるようになりました。』
- 7 ページの『Payment Card Industry Data Security Standard (PCI DSS) バージョン 3.2 レポートのサポート』
- 7 ページの『AppScan Source for Analysis 製品資料』
- 8 ページの『AppScan Source for Analysis でスキャン構成を使用してすべての除外フィルターの検出結果を削除する機能』
- 9 ページの『AppScan Source for Automation および AppScan Source コマンド行インターフェース (CLI)での WAR ファイルと EAR ファイルのスキャン時のライブラリー処理の改善』
- 9 ページの『分析のためのクラウドへの AppScan Source 評価の送信』
- 9 ページの『AppScan Source バージョン 9.0.3.4 でサポートされなくなった機能およびフィーチャー』

拡張および新規のスキャン・サポート

PHP バージョン 7.0 は、IBM Security AppScan Source for Analysis、IBM Security AppScan Source for Automation、および IBM Security AppScan

Source コマンド行インターフェース (CLI) の Windows と Linux 上でスキャンできるようになりました。

共通アクセス・カード (CAC) によって認証される場合、AppScan Enterprise Console への評価の公開がサポートされるようになりました。

CAC 認証を使用して AppScan Enterprise Server に接続する場合、AppScan Source ユーザー・インターフェース、AppScan Source コマンド行インターフェース (CLI)、および AppScan Source for Automation から AppScan Enterprise Console に評価を公開することができるようになりました。

Payment Card Industry Data Security Standard (PCI DSS) バージョン 3.2 レポートのサポート

AppScan Source は、Payment Card Industry Data Security Standard (PCI DSS) バージョン 3.2 レポートをサポートするようになりました。

AppScan Source for Analysis 製品資料

バージョン 9.0.3.4 では、AppScan Source for Analysis で「ヘルプ」 > 「ヘルプの目次」メニュー項目を使用すると、AppScan Source の IBM Knowledge Center でオンライン・ヘルプが開きます (バージョン 9.0.3.4 の場合、ヘルプは、IBM Security AppScan Source V9.0.3.4 資料を開きます)。同様に、AppScan Source for Analysis の「ようこそ」ビューのリンク先にアクセスすると、IBM Knowledge Center が開きます。

AppScan Source for Analysis は、さまざまなビュー、設定ページ、およびダイアログ・ボックスのコンテキスト・ヘルプも提供しています。コンテキスト・ヘルプへのキーボード・ショートカットは、Windows では F1、Linux では Shift+F1、macOS では command+F1 です。バージョン 9.0.3.4 では、このコンテキスト・ヘルプを使用して、AppScan Source の IBM Knowledge Center を開くこともできます。

インターネット接続なしで製品を使用している場合、以下のように、ヘルプをローカルで使用することができます。

- IBM Security AppScan Source の README およびリリース情報は、AppScan Source のインストール・ディレクトリーにある `readme.html` ファイルで参照することができます。
- 以下の PDF のユーザー・ガイドは、AppScan Source インストール・ディレクトリーの `doc/<lang>` ディレクトリーまたは `doc¥<lang>` ディレクトリー内にインストールされています。ここで、`<lang>` は、AppScan Source のインストール済み環境の各国語です。
  - Windows および Linux のみ: *IBM Security AppScan Source for Analysis* ユーザー・ガイド (`Security_AppScan_Source_Analysis.pdf`)
  - Windows および Linux のみ: *IBM Security AppScan Source Utilities* ユーザー・ガイド (`Security_AppScan_Source_Utilities.pdf`)
  - macOS のみ: *IBM Security AppScan Source for Analysis* ユーザー・ガイド - *macOS* (`Security_AppScan_Source_Analysis_OSX.pdf`)

- macOS のみ: *IBM Security AppScan Source Utilities* ユーザー・ガイド - *macOS* (Security_AppScan_Source_Utilities_OSX.pdf)
- *IBM Security AppScan Source* インストールと管理のガイド (Security_AppScan_Source_Installation_and_Administration.pdf)

それらのファイルを表示するには Adobe Acrobat Reader が必要です。Acrobat Reader のコピーを持っていない場合は、http://www.adobe.com/ からダウンロードすることができます。

- 一部の AppScan Source for Analysis 機能の Javadoc は、AppScan Source インストール・ディレクトリーの doc/Javadoc ディレクトリーまたは doc¥Javadoc ディレクトリーにあります。バージョン 9.0.3.4 では、以下の機能の Javadoc が使用可能です。
  - アプリケーション・サーバー・インポートのフレームワーク API クラスおよびメソッドの Javadoc は、doc/Javadoc/appserverimporter または doc¥Javadoc¥appserverimporter で入手可能です。
  - Framework for Frameworks API クラスおよびメソッドの Javadoc は、doc/Javadoc/frameworks または doc¥Javadoc¥frameworks で入手可能です。

  これらのフォルダーで、index.html ファイルを開きます。

AppScan Source for Analysis でスキャン構成を使用してすべての除外フィルターの検出結果を削除する機能

除外フィルターには、脆弱性タイプ、アプリケーション・プログラミング・インターフェース (API)、ファイル、ディレクトリー、プロジェクト、あるいはトレースを検出結果から除去する対象のルールが含まれます。スキャン構成に複数の除外フィルターを組み込んだ場合、相互に競合して、検出結果に影響する可能性があります。例えば、以下の 2 つのフィルターが提供されたとします。

- フィルター 1 は、脆弱性タイプ Validation.EncodingRequired のすべての検出結果を除去します。これは反転されないため、これらの検出結果は評価から除外されます。
- フィルター 2 は、脆弱性タイプ Validation.Required のすべての検出結果を除去します。これは反転されないため、これらの検出結果は評価から除外されます。

スキャン構成を使用してこれらのフィルターの両方が適用された場合、デフォルトでは、これらのフィルターはお互いを無視します。フィルター 1 は、Validation.EncodingRequired の検出結果を除外しますが、Validation.Required の検出結果は含めます。フィルター 2 は、Validation.Required の検出結果を除外しますが、Validation.EncodingRequired の検出結果は含めます。最終的な結果には、Validation.EncodingRequired の検出結果と Validation.Required の検出結果がすべて含まれます。

バージョン 9.0.3.4 では、スキャン構成の作成時に「任意の非反転除外フィルターを突き合わせます」を選択して指定されたすべての 除外フィルターによる検出結果を除外することができます。このチェック・ボックスは、「スキャン構成」ビューの「全般」タブの「フィルター情報」セクションにあります。上記の例の場合、このチェック・ボックスを選択すると、Validation.EncodingRequired の検出結果およ

び Validation.Required の検出結果は、すべて評価から除外されます。

AppScan Source for Automation および AppScan Source コマンド行インターフェース (CLI)での WAR ファイルと EAR ファイルのスキャン時のライブラリー処理の改善

WAR ファイルをスキャンする際、以下の設定が使用可能になりました。

- -include_all_lib_jars: この設定は、スキャン時に WAR ファイル内のすべてのライブラリーを組み込むために使用します。
- -include_lib_jars: この設定は、スキャン時に WAR ファイル内の組み込みたいライブラリーを指定する場合に使用します。

EAR ファイルをインポートする場合、共有ライブラリーを保管するためのプロジェクトが自動的に作成されます。 このプロジェクトは、共有ライブラリーが存在しない場合に作成されますが、内容は空になります。 -no_ear_project 設定が使用可能になったので、この設定を使用すると、EAR ファイルに対してプロジェクトが作成されません。

分析のためのクラウドへの AppScan Source 評価の送信

IBM Cloud Marketplace での IBM Application Security on Cloud に対するサブスクリプション、あるいは Application Security on Cloud for Bluemix に対するサブスクリプションがある場合は、AppScan Source 評価を分析のために送信することができます。AppScan Source バージョン 9.0 以上からの評価がサポートされます。送信できるスキャンの数は、Application Security on Cloud サブスクリプションによって異なります。詳しくは、http://www.ibm.com/support/knowledgecenter/SSYJJF_1.0.0/ApplicationSecurityonCloud/src_managing_assessments_cloud.htmlを参照してください。

AppScan Source バージョン 9.0.3.4 でサポートされなくなった機能およびフィーチャー

AppScan Source バージョン 9.0.3.4 時点:

- OS X バージョン 10.9 は、サポート対象のオペレーティング・システムではなくなりました。
- Xcode バージョン 5.x、6.0、および 6.2 はサポートされなくなりました。これらのバージョンの Xcode による Objective-C プロジェクトのスキャンはサポートされなくなりました。
- PHP バージョン 5.3 および 5.4 のスキャンのサポートは非推奨です。

AppScan Source バージョン 9.0.3.3 の新機能

- 10 ページの『新規プラットフォームと統合ソリューションのサポート』
- 11 ページの『拡張および新規のスキャン・サポート』
- 11 ページの『Windows の新規インストール・ファイル名』
- 11 ページの『Windows での Common Access Card (CAC) サポート』
- 11 ページの『「DISA Application Security and Development STIG V3R10」レポートのサポート』

新規プラットフォームと統合ソリューションのサポート

AppScan Source バージョン 9.0.3.3 時点:

- Microsoft Windows 10 がサポート対象のオペレーティング・システムになりました。 サポート対象には Windows 10 Education、 Enterprise、および Pro の各エディションが含まれます。

  注:

  – Windows 10 では、AppScan Source インストーラー (`AppScanSrc_Installer.exe` ファイル) を Windows 7 互換モードで実行する必要があります。Windows 10 では、AppScan Source をアンインストールする前に、`AppScan_Uninstaller.exe` ファイルもWindows 7 互換モードで実行するように設定する必要があります。このファイルは、`<install_dir>¥Uninstall_AppScan¥AppScan_Uninstaller.exe` (`<install_dir>` は、AppScan Source インストール済み環境がある場所です。詳しくは、360 ページの『インストールとユーザー・データ・ファイルの場所』を参照してください。) にあります。詳しくは、http://www.ibm.com/support/docview.wss?uid=swg21696098 を参照してください。

  – Windows 10 サポートは、http://www.ibm.com/support/docview.wss?uid=swg21689814 に記載されている問題に影響を受けます。

- AppScan Enterprise Server バージョン 9.0.3.1 以上に接続している場合、IBM Security AppScan Source データベース は Oracle 12c データベースにインストールできます。

  重要: Oracle 11g データベースを使用する AppScan Source の既存のインストール済み環境があり、Oracle 12c にアップグレードしたい場合、Oracle データベースをアップグレードする前に AppScan Source をアップグレードする必要があります。

- **Tomcat 8** は、AppScan Source のインストール済み環境に含まれるようになりました。

- Visual Studio 2015 ソリューションとプロジェクトのファイルは、AppScan Source for Analysis、 AppScan Source for Automation、および AppScan Source コマンド行インターフェース でスキャンできるようになりました。Visual Studio 2015 で作成された `.sln` ファイルまたは `.vcproj` ファイルがある場合、そのファイルは Windows で AppScan Source for Analysis、AppScan Source for Automation、または AppScan Source コマンド行インターフェースを使用する場合にインポートおよびスキャンできます。

  重要:

  – AppScan Source for Development Visual Studio プラグイン の Visual Studio 2015 への適用はサポートされていません。

  – Managed C++ プロジェクトがサポートされています。 Unmanaged C++ プロジェクトは、そのプロジェクトが Visual Studio 2013 以前の Platform Toolset (Platform Toolset V120 以前) を使用して作成された場合にはサポートされます。

- Xcode 7.3 for Objective-C (iOS アプリケーションのみ) は、macOS でサポートされるコンパイラーになりました (Xcode 7.3 のサポートは、AppScan Source バージョン 9.0.3.2 にまで遡ります)。

拡張および新規のスキャン・サポート

- PHP バージョン 5.5 および 5.6 は、IBM Security AppScan Source for Analysis、IBM Security AppScan Source for Automation、および IBM Security AppScan Source コマンド行インターフェース (CLI) の Windows と Linux 上でスキャンできるようになりました。
- Java™ をスキャンするために AppScan Source を使用する際は、`@ValidatorMethod`、`@CallbackMethod`、および `@SuppressSecurityTrace` のメソッド・レベル・アノテーションがサポートされるようになりました。

Windows の新規インストール・ファイル名

Windows でのインストール・ファイル名が `setup.exe` から `AppScanSrc_Installer.exe` に変更されました。

Windows での Common Access Card (CAC) サポート

Common Access Card (http://www.cac.mil) は、米国で、現役の軍および政府職員、SR、国防総省職員、有資格請負業者により使用されている標準 ID です。CAC を使用することにより、建物や管理スペースへの物理的アクセスが可能になり、DoD コンピューター・ネットワークおよびシステムにアクセスすることができます。CAC は、さまざまなスマート・カード・リーダーが装備されたコンピューターおよびネットワークへのアクセスに使用できます。 CAC をリーダーに挿入すると、デバイスはユーザーに PIN の入力を要求します。

AppScan Source を Windows で実行中であり、Common Access Card (CAC) 認証が有効になった AppScan Enterprise Server バージョン 9.0.3.1 iFix-001 以上に接続している場合、AppScan Source は CAC 認証をサポートするようになりました。

「DISA Application Security and Development STIG V3R10」レポートのサポート

AppScan Source は、Defense Information Systems Agency (DISA) Application Security and Development Security Technical Implementation Guide (STIG) V3R10 レポートをサポートするようになっています。

AppScan Source バージョン 9.0.3.2 の新機能

AppScan Source および AppScan Enterprise バージョン互換性

AppScan Source の一部のバージョンでは、AppScan Enterprise Server への接続時や AppScan Enterprise Console への公開時に、AppScan Source および AppScan Enterprise のバージョンとリリース・レベルを一致させる必要がなくなりました。 http://www.ibm.com/support/docview.wss?uid=swg21975211 を参照して、どのバージョンの AppScan Source と AppScan Enterprise に互換性があるかを確認してください。

この変更は、AppScan Source の一部の旧バージョンにそ及的です。 http://www.ibm.com/support/docview.wss?uid=swg21975211 で説明されています。

AppScan Source バージョン 9.0.3.1 の新機能

- 『新規統合ソリューションのサポート』
- 『AppScan Source for Automation および AppScan Source コマンド行インターフェース (CLI)での WAR ファイルと EAR ファイルのスキャン』

新規統合ソリューションのサポート

AppScan Source バージョン 9.0.3.1 では、次のようになっています。

- Tomcat 8 は、Java と JSP のコンパイルでサポートされるようになりました。

  注: オペレーティング・システムのサポートは、個々のコンパイラーがサポートするオペレーティング・システムによって異なります。
- Objective-C 用 Xcode 7.0、7.1、および 7.2 (iOS アプリケーションのみ) は、macOS でサポートされるコンパイラーになりました。

AppScan Source for Automation および AppScan Source コマンド行インターフェース (CLI)での WAR ファイルと EAR ファイルのスキャン

CLI の `openapplication (oa)` コマンドを使用して、`WAR` ファイルと `EAR` ファイルを開けるようになりました。 また、これらのファイルは `ScanApplication` コマンドを使用して AppScan Source for Automation でスキャン可能です。

AppScan Source バージョン 9.0.3 の新機能

- 『新規プラットフォームと統合ソリューションのサポート』
- 13 ページの『スキャン構成の機能拡張』
- 13 ページの『新規ルール属性によって、重大度が高の確定セキュリティー検出結果をより正確に特定』
- 14 ページの『自動逸失シンク解決によりスキャン結果が改善』
- 15 ページの『拡張および新規のスキャン・サポート』
- 15 ページの『AppScan Source バージョン 9.0.3 でサポートされなくなった機能およびフィーチャー』

新規プラットフォームと統合ソリューションのサポート

AppScan Source バージョン 9.0.3 では、以下のオペレーティング・システムがサポートされています。

- Red Hat Enterprise Linux バージョン 6 アップデート 6 および 7
- OS X バージョン 10.11。OS X バージョン 10.11 のサポートは、AppScan Source バージョン 9.0.2 にまで遡りますが、制限事項が http://www.ibm.com/support/docview.wss?uid=swg21968948 で説明されています (この制限事項は、AppScan Source バージョン 9.0.2 にのみ影響します)。

さらに、以下もサポートされるようになりました。

- Objective-C 用の Xcode 6.3 と 6.4 (iOS アプリケーションのみ) は、OS X でサポートされるコンパイラーになりました (Xcode 6.3 と 6.4 のサポートは、AppScan Source バージョン 9.0.2 にまで遡ります)。 Xcode 6.3 と 6.4 のサポートには、いくつかの制限事項があることにご注意ください。詳細については、http://www.ibm.com/support/docview.wss?uid=swg21962208を参照してください。これらの制限は、AppScan Source バージョン 9.0.3.1 以降には適用されません。
- AppScan Source for Development Eclipse プラグイン が IBM MobileFirst Platform Foundation バージョン 7.1 と統合されました。これで、IBM MobileFirst Platform バージョン 7.1 のプロジェクト、アプリケーション、環境、および HTML ファイルを AppScan Source 製品でスキャンできます。
- Rational Application Developer for WebSphere Software (RAD) バージョン 9.1.1 プロジェクト・ファイルとワークスペースをスキャンできます。また AppScan Source for Development (Eclipse プラグイン) を RAD バージョン 9.1.1 に適用できます。
- Eclipse バージョン 4.5 プロジェクト・ファイルとワークスペース (Java および IBM MobileFirst Platform のみ) をスキャンできます。また AppScan Source for Development (Eclipse プラグイン) を Eclipse バージョン 4.5 に適用できます。
- IBM WebSphere Application Server バージョン 8.5.5 は、Java と JSP のコンパイルでサポートされるようになりました。

  注**:** オペレーティング・システムのサポートは、個々のコンパイラーがサポートするオペレーティング・システムによって異なります。

スキャン構成の機能拡張

「スキャン構成」ビューが再設計され、以下のような重要機能が提供されるようになりました。

- フィルターを指定する機能。
- スキャンで実行する分析のタイプの設定。これには、汚染フロー分析とパターン・ベース分析が含まれます。

AppScan Source には、以下の標準装備のスキャン構成が含まれるようになりました: 「Web プレビュー・スキャン」、「Webクイック・スキャン」、「Web のバランス・スキャン」、および「詳しい Web のスキャン」

新規ルール属性によって、重大度が高の確定セキュリティー検出結果をより正確に特定

このリリースの AppScan Source では、`Attribute.Likelihood.High` 属性と `Attribute.Likelihood.Low` 属性が導入されます。これらの属性は、標準装備のルールに追加され、カスタム・ルールの作成時も使用することができます。

AppScan Source では、可能性 は、セキュリティー検出結果が悪用される可能性や機会を表します。 AppScan Source は、https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology#Step_2:_Factors_for_Estimating_Likelihood に示された可能性の定義を取り込み、トレース・プロパティーに基づいて可能性を決

定することで、その定義を詳細化します。提供された一連のトレース・プロパティー (例えば、ソース API 名、ソース API タイプ、ソース・テクノロジー、あるいはソース・メカニズムなど) によって、AppScan Source は、将来において特定の脆弱性を使用してトレースが悪用される可能性を判別します。

可能性は、トレースのソース・エレメントと結び付けられます。ソースはプログラムへの入力で、ファイル、サーブレット要求、コンソール入力、ソケットなどがあります。 多くの入力ソースの場合、内容と長さについて制限のないデータが返されます。チェックされていない入力については、汚染のソースと見なされます。

可能性の例には、以下のものがあります。

- HTTP ソースを持つトレース (`Request.getQueryString` など) とクロスサイト・スクリプティング・シンク (`Response.write` など) が提供されると、高可能性が決定されるため、検出結果の信頼性が上昇します。
- システム・プロパティー・ソースを持つトレース (`getProperty` など) とクロスサイト・スクリプティング・シンク (`Response.write` など) が提供されると、低可能性が決定されるため、検出結果の信頼性が低下します。

可能性は、即時にアクションを実行するか修正する必要がある、優先度が高い要アクションの検出結果を識別するために使用されます。これは、悪用される可能性が高い汚染のソースと結び付けられ、検出結果を分類するためにより微細化されたアプローチを提供することができます。可能性は、汚染のソースに結び付けられた属性として、AppScan Source 脆弱性データベースに保管されます。この機能は、すぐに使用可能です。

IBM は、ソースの可能性要因を判別するための大規模な研究を実施してきています。カスタム・ルール・ウィザードを使用して、ルール・ベースに追加する新規の汚染ソースに可能性情報を追加することができます。これにより、スキャンによって生成された検出結果の分類が改善され、それによってトリアージ・ワークフロー全体の効率性が向上します。

カスタム・ルール・ウィザードには、「可能性」プロパティーに設定可能な 2 つの値 (「高」および「低」) があります。値「高」は、汚染に対してソースが非常に影響を受けやすいことを意味します。つまり、システムに侵入する汚染に対する障壁が非常に低く、攻撃者が悪意のあるデータを手動あるいは自動のいずれの方法でも容易に送信することが可能になります。値「低」は、このソースを介した悪意のあるデータの侵入に対する障壁が非常に高くなります。これは、攻撃者がソースを汚染させるには、システムの内部知識と、攻撃対象のネットワーク上で操作するための権限が必要になることを意味します。

注**:** これらのルール属性のために、以前のバージョンの AppScan Source で評価を生成している場合、その同じソースをバージョン 9.0.3 でスキャンすると、検出結果の分類が変更されることがあります。 詳細について、またこれらのルール属性を無効にする方法については、これらの変更に関するマイグレーション考慮事項を参照してください。

自動逸失シンク解決によりスキャン結果が改善

AppScan Source は、getter や setter などの逸失シンク・メソッド、およびブール値を返すメソッドのマークアップを自動的に推測することで、トレース内の逸失シ

ンクを解決しようとします。これにより、ご使用のコードの分析がより厳密になり、逸失シンクの解決をより適切に行うことができます。

注**:** これらの機能のために、以前のバージョンの AppScan Source で評価を生成している場合、解決されなかった逸失シンクの検出結果に変更がある場合があります。 詳細について、また自動マークアップ生成を無効にする方法については、これらの変更に関するマイグレーション考慮事項を参照してください。

拡張および新規のスキャン・サポート

- PHP バージョン 5.4 は、IBM Security AppScan Source for Analysis、IBM Security AppScan Source for Automation、および IBM Security AppScan Source コマンド行インターフェース (CLI) の Windows と Linux 上でスキャンできるようになりました。
- AppScan Source では、Spring MVC 4 フレームワークが標準でサポートされるようになりました。
- **Java** スキャンの最適化**:**
  - JavaServer Pages をスキャンする場合、スキャン中にそれらをコンパイルするのではなく、プリコンパイル済みクラス・ファイルをスキャンすることができるようになりました。 AppScan Source for Development Eclipse プラグイン のプリコンパイル済みクラス・ファイルをスキャンするには、セキュリティー・スキャンのプロジェクトを構成 (「セキュリティー分析」 > 「スキャンの構成」 > 「セキュリティーのためのプロジェクトの構成」を選択) し、「プリコンパイル済みクラス」チェック・ボックスを選択します。 IBM Security AppScan Source for Analysis のプリコンパイル済みクラス・ファイルをスキャンするには、以下のいずれかの場所にある「プリコンパイル済みクラス」チェック・ボックスを選択します。
    - プロジェクト・プロパティーの「プロジェクト依存関係」タブ。
    - 新規プロジェクトまたはアプリケーションを作成する場合の「Java プロジェクト依存関係」ページ。
  - Java のスキャン中、AppScan Source は、依存関係の欠落やコンパイル・エラーについて Java ファイルや Java バイトコードをスキャンします。 従属関係の欠落やコンパイル・エラーが存在する場合、その情報はログ・ファイルに書き込まれます。 その情報を使用して、依存関係をプロジェクト・プロパティーに追加して、再スキャンし、スキャン結果の全範囲をカバーすることができます。
- AppScan Source バージョン 9.0.3 では、 Xcode プロジェクトのインポートおよびスキャン時に、ヘッダーの位置と構成オプションが、より正確に判別されます。この変更では、すべてのファイルのビルド構成を取得するために `xcodebuild -dry-run` を使用するようになったので、続行前に AppScan Source がファイルの構成を判別するとき、スキャンの開始時に一時停止する可能性があります。

AppScan Source バージョン 9.0.3 でサポートされなくなった機能およびフィーチャー

AppScan Source バージョン 9.0.3 では次のような変更が行われました。

- OS X バージョン 10.8 は、サポート対象のオペレーティング・システムではなくなりました。
- Xcode バージョン 4.6 はサポートされなくなりました。このバージョンの Xcode による Objective-C プロジェクトのスキャンはサポートされなくなりました。
- Eclipse バージョン 3.6 および 3.7 のプロジェクト・ファイルとワークスペースはサポートされなくなり、AppScan Source for Development (Eclipse プラグイン) は Eclipse バージョン 3.6 および 3.7 に適用できなくなりました。
- Rational Application Developer for WebSphere Software (RAD) バージョン 8.0.x プロジェクト・ファイルとワークスペースはサポートされなくなり、IBM Security AppScan Source for Development プラグイン for IBM Rational Application Developer for WebSphere Software (RAD) は RAD バージョン 8.0.x に適用できなくなりました。
- IBM Rational Team Concert バージョン 3.0 および 3.0.1 は、障害追跡システムとしてサポートされなくなりました。
- WebSphere Application Server バージョン 6.1 は、アプリケーション・サーバーとしてサポートされなくなりました。
- PHP バージョン 4.x から 5.2 のスキャンのサポートは非推奨です。

現行バージョンの AppScan Source へのマイグレーション

このトピックには、このバージョンの AppScan Source で行われた変更についてのマイグレーション情報が記載されています。旧バージョンの AppScan Source からアップグレードしている場合は、必ず、アップグレードしている AppScan Source のバージョンと、この現行バージョンまでのすべてのバージョンにおける変更内容に注意してください。

- 『バージョン 9.0.2 からのマイグレーション』
- 18 ページの『バージョン 9.0 からのマイグレーション』
- 18 ページの『バージョン 8.7 からのマイグレーション』

バージョン 9.0.2 からのマイグレーション

- 『新規ルール属性により、既存スキャンの検出結果の分類が変更される可能性があります。』
- 17 ページの『自動逸失シンク生成』

新規ルール属性により、既存スキャンの検出結果の分類が変更される可能性があります。

バージョン 9.0.2 より後のバージョンでは、`Attribute.Likelihood.High` と `Attribute.Likelihood.Low` のルール属性が導入されました。 これらの属性を使用すると、AppScan Source は、検出結果が「確定」または「要確認」 (あるいはその両方) かを正確に判別することができます。 そのため、AppScan Source バージョン 9.0.2 以前でソース・コードをスキャンした場合、その同じソース・コードを 9.0.2 より後の製品バージョンでスキャンすると、検出結果の分類が変更されることがあります。これは、悪用可能性が高い Web ソースに関連する検出結果、または悪用可能性が低いプロパティーや環境のソースにおいて最も顕著です。

これらのルール属性はデフォルトで使用可能です。以下のように無効にすることができます。

1. テキスト・エディターで <data_dir>¥config¥ipva.ozsettings を開きます (<data_dir> は、ご使用の AppScan Source プログラム・データの場所です。詳しくは、 360 ページの『インストールとユーザー・データ・ファイルの場所』 を参照してください。)。ファイル内の allow_likelihood 設定を見つけます。 この設定は、以下の例のようになります。

```
<Setting
  name="allow_likelihood"
  value="true"
  default_value="true"
  description="Allow the processing of the Likelihood
    attributes to help determine trace confidence based
    on the source API"
  display_name="Allow Likelihood"
  type="bool"
/>
```

   この設定では、value 属性を変更します。属性が true に設定されている場合、この設定はオンになります。 false に設定されている場合、AppScan Source はスキャン中にこれらの属性ルールを使用しません。

2. この設定の変更後、ファイルを保存してAppScan Source を始動または再始動します。

自動逸失シンク生成

9.0.2 より後のバージョンでは、getter/setter、およびブール値を返すメソッドで終わるトレースに、自動逸失シンク解決が導入されました。これは、これらのアプリケーション・プログラミング・インターフェース (API) のマークアップを自動的に推測することで実現します。そのため、AppScan Source バージョン 9.0.2 以前でソース・コードをスキャンした場合、同じソース・コードを 9.0.2 より後の製品バージョンでスキャンすると、未解決の逸失シンクが含まれる検出結果が変更される可能性があります。

自動マークアップ生成はデフォルトでオンにされています。カスタム・ルールなど、他の手段を使用して逸失シンクを解決する場合は、生成をオフにすることができます。

1. テキスト・エディターで <data_dir>¥config¥ipva.ozsettings を開きます (<data_dir> は、ご使用の AppScan Source プログラム・データの場所です。詳しくは、 360 ページの『インストールとユーザー・データ・ファイルの場所』 を参照してください。)。ファイル内の automatic_lost_sink_resolution 設定を見つけます。この設定は、以下の例のようになります。

```
<name="automatic_lost_sink_resolution"
  value="true"
  default_value="true"
  description="This setting tries to perform automatic
    lost sink resolution by assuming taint propagation
    for getters, setters and APIs which return boolean
    with no arguments."
  display_name="Auto Lost Sink Resolution"
  type="bool"
/>
```

この設定では、value 属性を変更します。属性が true に設定されている場合、この設定はオンになります。 false に設定されている場合、AppScan Source はこれらのメソッドのマークアップを自動生成しません。

2. この設定の変更後、ファイルを保存してAppScan Source を始動または再始動します。

バージョン 9.0 からのマイグレーション

AppScan Enterprise Server の認証: IBM Rational Jazz™ ユーザー認証コンポーネントの IBM WebSphere Liberty への置き換えに関するマイグレーションの考慮事項

- ローカル **Jazz** ユーザーのみが含まれる **Enterprise Server** からのマイグレーション: このアップグレード・シナリオでは、以前の Jazz ユーザーは、AppScan Source データベースにAppScan Enterprise Server ユーザーとして表示されますが、有効ではありません。これらのユーザーをデータベースから削除できます。あるいは、http://www.ibm.com/support/docview.wss?uid=swg21686347 の説明に従って変換を有効にする場合、これらのユーザーを AppScan Source ユーザーに変換できます。
- **LDAP** を使用して構成された **Enterprise Server** からのマイグレーション: Enterprise Server のアップグレード時に、LDAP を使用して Enterprise Server を再構成するオプションがあります。これを行う場合、既存のユーザーは引き続き AppScan Source で機能します。
- **Windows** 認証を使用して構成された **Enterprise Server** からのマイグレーション: Enterprise Server が Windows 認証を使用して構成されていた場合、新しい Enterprise Server Liberty が Windows 認証を使用するように構成されていると、既存のユーザーは AppScan Source で機能します。

バージョン 8.7 からのマイグレーション

- 『検出結果の分類の変更』
- 19 ページの『スキャン範囲を改善するデフォルト設定の変更』
- 20 ページの『以前のバージョンからの AppScan Source 事前定義フィルターの復元』

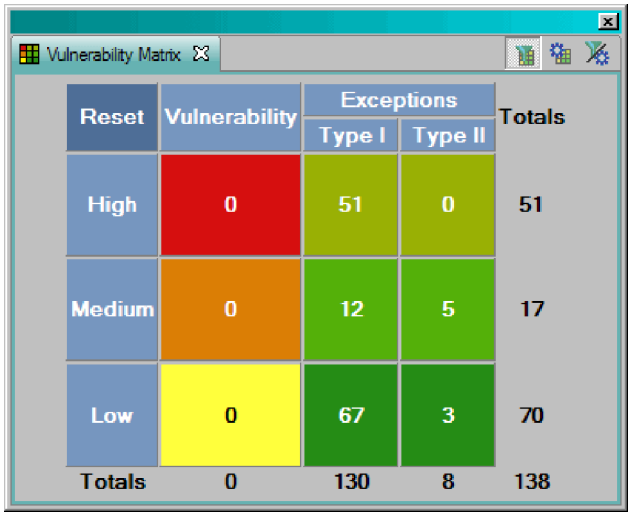
検出結果の分類の変更

バージョン 8.7 以降では、検出結果の分類が変更されました。この表では、従来の分類と新しい分類の対応を示します。

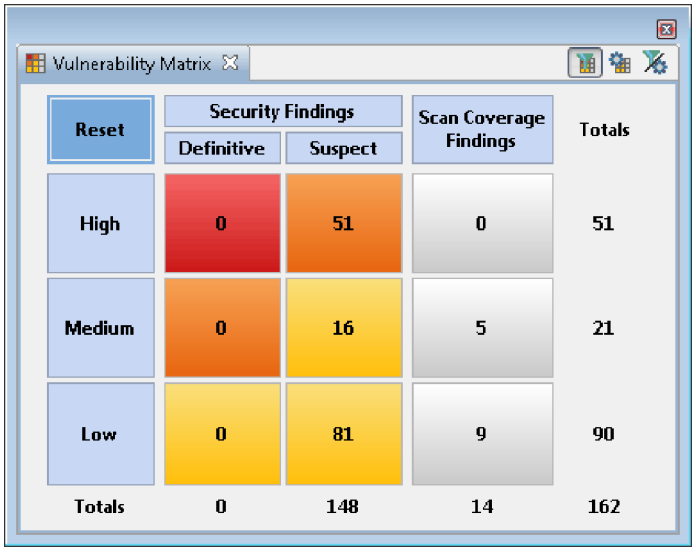
表 *1.* 検出結果分類の変更

バージョン **8.8** より前の **AppScan Source** の検出結果分類	**AppScan Source** バージョン **8.8** での検出結果の分類
脆弱性	「確定」セキュリティー検出結果
タイプ I 例外	「要確認」セキュリティー検出結果
タイプ II 例外	スキャン範囲検出結果

これらの変更の例は、「脆弱性マトリックス」ビューで確認できます。



バージョン 8.8 では、次のようになります。



スキャン範囲を改善するデフォルト設定の変更

AppScan Source バージョン 8.8 の場合:

- `scan.ozsettings` 内の `show_informational_findings` のデフォルト値が `true` から `false` に変更されました。
- `ipva.ozsettings` 内の `wafl_globals_tracking` のデフォルト値が `false` から `true` に変更されました。この設定により、AppScan Source がフレームワーク・ベースの各種コンポーネント間のデータ・フローを検出することが可能になります (例えば、コントローラーからビューへのデータ・フローなど)。

`show_informational_findings` に対する変更によって、重大度レベルが「情報」の検出結果は、デフォルトでは評価に組み込まれないようになります。

注**:** 8.8 より前のバージョンで作成されたスキャン構成があり、それらに対して値を明示的に設定していなかった場合、そのスキャン構成では新しいデフォルト値が使用されます。

以前のバージョンからの AppScan Source 事前定義フィルターの復元

AppScan Source バージョン 8.8 では、より有用なスキャン結果が得られるように定義済みフィルターが改善されました。 AppScan Source の旧バージョンからの定義済みフィルターを引き続き使用する必要がある場合は (アーカイブ・フィルターのリストは 173 ページの『AppScan Source 事前定義フィルター (バージョン 8.7.x 以前)』に記載されています)、 174 ページの『アーカイブ済みの事前定義フィルターの復元』の指示のとおりに行ってください。

AppScan Source for Analysis の概要

AppScan Source for Analysis は、コードを分析して基幹システムにおけるソース・コードの脆弱性に関する詳細情報を提供するツールです。 AppScan Source for Analysis を使用することにより、複数のアプリケーションにわたって、またはポートフォリオ全体にわたっても、ソフトウェアのリスクを集中管理することができます。 ソース・コードをスキャンしてトリアージを実施し、脆弱性を事前に排除することにより、脆弱性による組織の責任が発生するのを防ぐことができます。

AppScan Source for Analysis は、ソース・コードをスキャンし、結果についてトリアージを実行し、障害情報を障害追跡システムに送信するためのツールを、監査チームや品質管理チームに対して提供します。

AppScan Source セキュリティー・ナレッジ・データベースに基づくコンテキスト内情報を活用することにより、アナリスト、監査員、開発者は以下の操作を行うことができます。

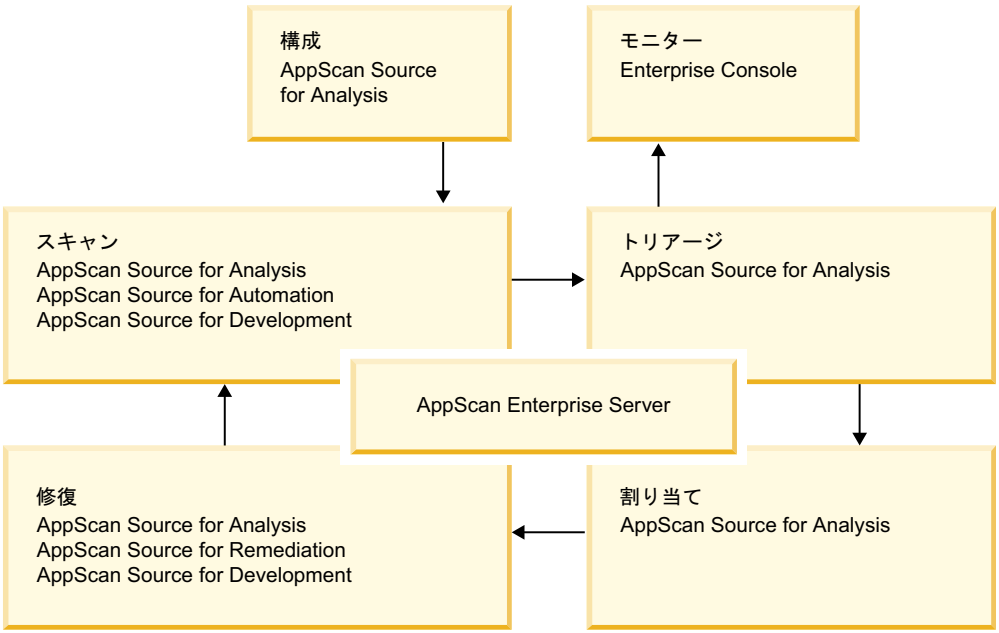
- 選択したソース・コードを必要に応じてスキャンし、重大な脆弱性を検出する。
- 的確な修復アドバイスを受け取り、任意の開発環境とコード・エディターを分析情報から直接起動する。
- 入力から出力までの正確な対話式呼び出しグラフを使用して、汚染されたデータをトレースする。
- エンコード・ポリシーを適用し、AppScan Source トレースを使用して、承認された入力検証ルーチンとエンコード・ルーチンを検証する。
- ソフトウェア開発時におけるセキュア・プログラミングのベスト・プラクティスについて学習し、実践する。

ワークフロー

インストール、デプロイメント、ユーザー管理が完了したら、AppScan Source ワークフローの基本ステップは以下のようになります。

1. セキュリティー要求の設定: 管理者またはセキュリティーの専門家が、脆弱性と重大度の判定方法を定義します。
2. アプリケーションの構成: アプリケーションとプロジェクトを編成します。

3. スキャン: ターゲット・アプリケーションに対して分析を実行し、脆弱性を特定します。
4. 結果のトリアージおよび分析: セキュリティーを担当するスタッフが、分析結果を調査して修復ワークフローの優先順位付けを行い、実際の脆弱性と潜在的な脆弱性を切り分けます。これにより、重大な問題に関するトリアージを即座に開始することができます。この状態で、最初に修正する必要がある問題を特定します。
5. ナレッジベース・データベースのカスタマイズ: AppScan Source セキュリティー・ナレッジ・データベースをカスタマイズして、内部ポリシーに対応します。
6. スキャン結果の公開: スキャン結果を AppScan Source データベースに追加するか、または AppScan Enterprise Console に公開します。
7. 修復タスクの割り当て: 脆弱性の解決のため、開発チームに障害問題を割り当てます。
8. 問題の解決: コードの書き換え、問題部分の削除、またはセキュリティー機能の追加により、脆弱性を解消します。
9. 修正の検証: コードを再スキャンして、脆弱性が排除されたことを確認します。



重要な概念

AppScan Source の使用または管理を開始する前に、AppScan Source の基本的な概念についてよく理解しておく必要があります。このセクションでは、AppScan Source の基本的な用語と概念について定義します。 これ以降の章では、こうした用語と概念の定義が繰り返して出てきます。この定義を参照することにより、これらの用語と概念が AppScan Source for Analysis ではどのような意味を持つのかを理解することができます。

AppScan Source for Analysis は、ソース・コードをスキャン して脆弱性を検出し、検出結果 を生成します。 検出結果とは、スキャンによって検出された脆弱性

のことです。スキャンの結果は、評価 と呼ばれます。 バンドル は、個別の検出結果の名前付きコレクションであり、アプリケーションと共に保管されます。

アプリケーション、アプリケーションの属性、およびプロジェクトは、AppScan Source for Analysis で作成および編成されます。

- アプリケーション: アプリケーションには、1 つ以上のプロジェクトと関連する属性が格納されます。
- プロジェクト: プロジェクトは、ソース・コードを含む一連のファイルと、構成データなどの関連情報から構成されます。 プロジェクトは、常にアプリケーションの一部になります。
- 属性: 属性とは、スキャンの結果を意味のあるグループ (部門別やプロジェクト・リーダー別など) に分けて整理するのに役立つアプリケーションの特性のことです。 属性は AppScan Source for Analysis で定義します。

AppScan Source for Analysis の主な機能は、ソース・コードをスキャンして脆弱性を分析することです。評価データによって提供される脆弱性についてのソース・コードの分析情報には、以下のようなものがあります。

- 重大度: リスクのレベルを示すための値 (高、中、低)。
- 脆弱性タイプ: SQL 注入やバッファー・オーバーフローなどの脆弱性カテゴリー。
- ファイル: 検出結果が存在するコード・ファイル
- **API/**ソース: API と API に渡される引数を表示する脆弱な呼び出し。
- メソッド: 脆弱な呼び出しが作成される関数またはメソッド。
- 位置: 脆弱な API が記述されているコード・ファイル内の行番号と列番号。
- 分類: セキュリティー検出結果またはスキャン範囲検出結果のいずれか。詳しくは、『分類』を参照してください。

分類

検出結果は、セキュリティー検出結果かスキャン範囲検出結果のどちらであるかを示すために AppScan Source によって分類されます。セキュリティー検出結果は、実際のセキュリティーの脆弱性または疑わしいセキュリティーの脆弱性を表し、一方でスキャン範囲検出結果は、構成を改善することでスキャンの範囲がより適切となる可能性があるエリアを表します。

各検出結果には、以下のいずれかの分類 が割り当てられます。

- 「確定」セキュリティー検出結果: 本来意図されていない動作を攻撃者がアプリケーションに実行させるおそれがあることを表す、明確な設計違反、実装違反、またはポリシー違反が含まれている検出結果。

  こうした攻撃は、データ、システム、リソースの無許可アクセス、盗難、破損を招く可能性があります。 「確定」セキュリティー検出結果はすべて完全に明確に示され、各脆弱性条件に固有の基本パターンは識別されて記述されます。
- 「要確認」セキュリティー検出結果: 追加の情報や調査が必要な、脆弱性が発生するおそれのある疑わしい条件が存在することを示す検出結果。不正に使用されると脆弱性が発生するおそれのあるコード・エレメントまたは構造。

「要確認」検出結果は、明確に脆弱性として定義できない不明な状況が存在するという点で、「確定」検出結果とは異なります。この不明条件の例としては、ソース・コードが提供されていない動的要素やライブラリー関数を使用していることなどがあります。したがって、「要確認」検出結果が「確定」であるかどうかを断定するには、もう一段階詳しい調査が必要となります。

- 「スキャン範囲」検出結果: 構成の改善によって、スキャンの範囲がより適切となる可能性があるエリアを表す検出結果 (逸失シンク検出結果など)。

注: 場合によっては、「なし」の分類を使用して、セキュリティー検出結果でもスキャン範囲検出結果でもない分類が示されることがあります。

AppScan Source 製品から AppScan Enterprise Server へのログイン

ほとんどの AppScan Source 製品とコンポーネントでは、AppScan Enterprise Server への接続が必要です。このサーバーは、一元的ユーザー管理機能と、AppScan Source データベース を介して評価を共有するためのメカニズムを提供します。

AppScan Source for Analysis を起動すると、AppScan Enterprise Server に認証を受けるように求めるプロンプトが表示されます。 サーバー・モードで AppScan Source for Development を実行している場合、サーバーにアクセスする必要があるアクション (スキャンの起動、またはスキャン構成の表示など) を初めて開始するときに、AppScan Enterprise Server に認証を受けるように求めるプロンプトが表示されます。

- 『AppScan Enterprise Server ユーザー ID とパスワードを使用した AppScan Source for Analysis および AppScan Source for Development からのログイン』
- 24 ページの『AppScan Source for Analysis および AppScan Source for Development からログインするための Common Access Card (CAC) 認証の使用』
- 25 ページの『AppScan Source for Automation および AppScan Source コマンド行インターフェース (CLI)からのログイン』
- 25 ページの『AppScan Enterprise Server の SSL 証明書』
- 25 ページの『AppScan Enterprise Server 証明書エラーの解決』

AppScan Enterprise Server ユーザー ID とパスワードを使用した AppScan Source for Analysis および AppScan Source for Development からのログイン

AppScan Source for Analysis では、ログイン時に以下の情報が要求されます。

- ユーザー **ID**: ユーザー ID を指定します (アカウントがどのようにセットアップされているかに応じて、これは、AppScan Enterprise Server 上と AppScan Source データベース 内の両方に存在するユーザー ID か、AppScan Source データベース 内のみに存在するユーザー ID のいずれかです)。
  - AppScan Enterprise Server が Windows 認証を使用するように構成されている場合、Enterprise Console への接続に使用するドメイン名とユーザー名を入力します。ドメイン名とユーザー名は ¥ で区切ります (例えば、`my_domain¥my_username`)。

  - AppScan Enterprise Server が LDAP を使用して構成されている場合、Enterprise Console への接続に使用するユーザー名を入力します。
- パスワード: ユーザー ID のパスワードを指定します。
- **AppScan Enterprise Server**: AppScan Enterprise Server インスタンスの URL を指定します。この URL の形式は `http(s)://<hostname>:<port>/ase` です。ここで、`<hostname>` は、AppScan Enterprise Server がインストールされているマシンの名前、`<port>` は、サーバーが稼働しているポートです。この URL の例は、`https://myhost.mydomain.ibm.com:9443/ase` のようになります。

AppScan Source for Development では、ログイン時に以下の情報が要求されます。

- サーバー **URL**: AppScan Enterprise Server インスタンスの URL を指定します。この URL の形式は `http(s)://<hostname>:<port>/ase` です。ここで、`<hostname>` は、AppScan Enterprise Server がインストールされているマシンの名前、`<port>` は、サーバーが稼働しているポートです。この URL の例は、`https://myhost.mydomain.ibm.com:9443/ase` のようになります。
- ユーザー **ID**: ユーザー ID を指定します (アカウントがどのようにセットアップされているかに応じて、これは、AppScan Enterprise Server 上と AppScan Source データベース 内の両方に存在するユーザー ID か、AppScan Source データベース 内のみに存在するユーザー ID のいずれかです)。
  - AppScan Enterprise Server が Windows 認証を使用するように構成されている場合、Enterprise Console への接続に使用するドメイン名とユーザー名を入力します。ドメイン名とユーザー名は `¥` で区切ります (例えば、`my_domain¥my_username`)。
  - AppScan Enterprise Server が LDAP を使用して構成されている場合、Enterprise Console への接続に使用するユーザー名を入力します。
- パスワード: ユーザー ID のパスワードを指定します。

AppScan Source for Analysis および AppScan Source for Development からログインするための Common Access Card (CAC) 認証の使用

Windows の場合、CAC 認証 (http://www.cac.mil) を使用して AppScan Enterprise Server に接続できます。 接続する前に、 Common Access Card (CAC) 認証用に AppScan Enterprise Server と AppScan Source をセットアップする必要があります。 Enterprise Server が CAC 認証用に設定されている場合、Enterprise Server のユーザー ID とパスワードを使用してログインすることはできません。

AppScan Source for Analysis では、ログイン時に以下の情報が要求されます。

- ユーザー: ご使用の CAC 共通名をリストから選択します。
- **AppScan Enterprise Server**: AppScan Enterprise Server インスタンスの URL を指定します。この URL の形式は `http(s)://<hostname>:<port>/ase` です。ここで、`<hostname>` は、AppScan Enterprise Server がインストールされているマシンの名前、`<port>` は、サーバーが稼働しているポートです。この URL の例は、`https://myhost.mydomain.ibm.com:9443/ase` のようになります。

AppScan Source for Development では、ログイン時に以下の情報が要求されます。

- サーバー **URL**: AppScan Enterprise Server インスタンスの URL を指定します。この URL の形式は `http(s)://<hostname>:<port>/ase` です。ここで、`<hostname>` は、AppScan Enterprise Server がインストールされているマシンの名前、`<port>` は、サーバーが稼働しているポートです。この URL の例は、`https://myhost.mydomain.ibm.com:9443/ase` のようになります。
- ユーザー: ご使用の CAC 共通名をリストから選択します。

「**OK**」のクリック後、「Windows セキュリティ」ダイアログ・ボックスに、ご使用の CAC カード PIN を入力するようにプロンプトが出されます。

ヒント:

- ログインが失敗した場合、AppScan Enterprise Server が正しくセットアップされていること、およびご使用の証明書が有効なことを確認してください。ブラウザーを経由して AppScan Enterprise Server にアクセスできるかどうかを確認してください。そうである場合は、証明書を選択してログインできるはずです。
- ログイン・ダイアログ・ボックスの「ユーザー」フィールドに使用可能な証明書がリストされていない場合、JRE の `java.security` ファイルを 26 ページの『Common Access Card (CAC) 認証の有効化』 の説明に従って変更したことを確認してください。
- 「Windows セキュリティ」ダイアログ・ボックスにより CAC カード・ピンに関するプロンプトが出されない場合、Microsoft Smart Card Resource Manager サービスが実行中であることを確認してください。一部のリモート・デスクトップ接続タイプでは、このサービスが実行されていない可能性があることにご注意ください。

AppScan Source for Automation および AppScan Source コマンド行インターフェース (CLI)からのログイン

ログイン・アクションは、AppScan Source for Automation または AppScan Source コマンド行インターフェース (CLI) を実行する際にも必要です。詳しくは、*IBM Security AppScan Source Utilities* ユーザー・ガイド を参照してください。

AppScan Enterprise Server の SSL 証明書

AppScan Enterprise Server の SSL 証明書について詳しくは、 28 ページの『AppScan Enterprise Server の SSL 証明書』を参照してください。

AppScan Enterprise Server 証明書エラーの解決

不明な認証局を使用して Enterprise Server にログインしようとすると、ログイン時に証明書例外またはエラーが表示される場合があります。AppScan Source には、この修正に役立つ小さなユーティリティーが含まれています。 このツールは、`<install_dir>¥bin¥certificatetool.bat` (`<install_dir>` は AppScan Source インストール済み環境がある場所です) または （Linux および macOS の場合は) `<install_dir>/bin/certificatetool.sh` です。

Common Access Card (CAC) 認証の有効化

このトピックでは、Common Access Card (CAC) 認証が有効になっている AppScan Enterprise Server への接続を許可するように AppScan Source を設定する方法を示します。

始める前に

CAC 認証は、Windows における AppScan Enterprise Server バージョン 9.0.3.1 iFix-001 以上への接続用にのみサポートされています。

手順

1. AppScan Enterprise Server が CAC 認証用にまだセットアップされていないことを確認してください。
2. AppScan Source for Analysis または AppScan Source コマンド行インターフェース (CLI) に AppScan Source 管理者としてログインします。
3. すべての AppScan Enterprise Server ユーザーがすべての許可を持つように設定するには、*IBM Security AppScan Source* インストールと管理のガイド の指示に従ってください。これにより、AppScan Enterprise Server ユーザーに完全管理アクセス権を与える初期のデフォルト許可が設定されますが、CAC セットアップの完了後は組織のニーズに合うように、そのデフォルト許可を変更できます。
4. すべての AppScan Source クライアント・アプリケーションを終了またはシャットダウンします。
5. CAC 認証を許可するために AppScan Enterprise Server をセットアップします。
6. 「*IBM Security AppScan Source* インストールと管理のガイド」の手順に従って、AppScan Source データベース を Common Access Card (CAC) 認証が有効にされている AppScan Enterprise Server に登録します。
7. `<data_dir>¥config¥ounce.ozsettings` (`<data_dir>` は、ご使用の AppScan Source プログラム・データの場所です。詳しくは、 360 ページの『インストールとユーザー・データ・ファイルの場所』 を参照してください。)) を開きます。 このファイルで、以下の設定を見つけます。

   ```
   <Setting
      name="client_cert_auth"
      value="false"
      default_value="false"
      description="Uses client certificate authentication"
      display_name="Uses client certificate authentication"
      type="boolean"
      read_only="true"
      hidden="true"
    />
   ```

8. この設定で、`value="false"` を `value="true"` に変更し、ファイルを保存します。
9. AppScan Source for Analysis または AppScan Source for Development Eclipse プラグイン から AppScan Enterprise Server にログインする場合には以下のようにします。
   a. Java インストール・ディレクトリーで、`jre/lib/security/java.security` を見つけます。 AppScan Source for Analysis の場合、`jre` フォルダーは

AppScan Source インストール・ディレクトリー内にあります。このファイルのバックアップ・コピーを作成します。

b. `java.security` を編集します。

c. プロバイダーとその優先順位のリストに、1 番目のセキュリティー・プロバイダーとして `com.ibm.security.capi.IBMCAC` を追加します。 例えば、AppScan Source for Analysis 使用のため `java.security` を編集する場合、次の部分が変更対象箇所です。

```
security.provider.1=com.ibm.crypto.fips.provider.IBMJCEFIPS
security.provider.2=com.ibm.jsse2.IBMJSSEProvider2
security.provider.3=com.ibm.crypto.provider.IBMJCE
security.provider.4=com.ibm.security.cert.IBMCertPath
security.provider.5=sun.security.provider.Sun
```

次のように変更します。

```
security.provider.1=com.ibm.security.capi.IBMCAC
security.provider.2=com.ibm.crypto.fips.provider.IBMJCEFIPS
security.provider.3=com.ibm.jsse2.IBMJSSEProvider2
security.provider.4=com.ibm.crypto.provider.IBMJCE
security.provider.5=com.ibm.security.cert.IBMCertPath
security.provider.6=sun.security.provider.Sun
```

d. `java.security` ファイルを保存して閉じます。

10. CAC 認証を使用して、AppScan Source 管理者として AppScan Source for Analysis または AppScan Source コマンド行インターフェース (CLI) にログインします。
11. AppScan Enterprise Server ユーザーのデフォルト許可を組織のニーズに合うように変更します。

次のタスク

連邦情報処理標準 (FIPS) モードを実施したい場合は、証明書を SHA-1 に設定することはできません。 SHA-2 証明書を使用し、`appscanserverdbmgr_cac_fips.bat` ツールを実行して FIPS モードを実施できます。その方法は *IBM Security AppScan Source* インストールと管理のガイド に説明されています。本書で、Common Access Card (CAC) 認証が有効になっている AppScan Enterprise Server に AppScan Source データベース を登録するためのヘルプを参照してください。

所有している Web 証明書を判別するには、以下のようにします。

1. Windows 証明書マネージャーを開きます。Windows の「スタート」メニューの検索ボックスで `certmgr.msc` と入力し、Enter を押します。管理者パスワードまたは確認のプロンプトが表示されたら、パスワードまたは確認の入力をします。
2. ダブルクリックするか、ユーザー・インターフェースの「開く」アクションで証明書を開きます。
3. 証明書の「詳細」タブを選択します。
4. 「署名ハッシュ アルゴリズム」フィールドを見つけます。このフィールドの値が証明書のタイプを示しています。

AppScan Source ユーザー・パスワードの変更

AppScan Sourceユーザー・パスワードを変更するには、「ユーザーの管理」権限が必要です。また、AppScan Source for Analysis で変更を行う必要があります。この権限がない場合は、代わりに管理者にこのトピックの説明に従ってパスワードを変更してもらってください。LDAP 認証または Windows 認証を使用するように AppScan Enterprise Server が構成されている場合、このトピックは適用されません。

手順

1. AppScan Source for Analysis で、メイン・ワークベンチ・メニューから「管理」 > 「ユーザーの管理」を選択します。
2. 「ユーザーの管理」ダイアログ・ボックスに、既存の AppScan Source ユーザーがリストされます。いずれかのユーザーのパスワードを変更するには、以下のいずれかのタスクを実行してユーザー情報を編集します。
   - ユーザーをダブルクリックします。
   - ユーザーを右クリックして、「ユーザーの編集」を選択します。
   - ユーザーを選択して、「ユーザーの編集」ボタンをクリックします。

   注**:** AppScan Enterprise Server ユーザーのパスワードを AppScan Source から変更することはできません。
3. 「ユーザーの編集」ダイアログ・ボックスで、新規パスワードを入力してから、「パスワードの確認」フィールドにパスワードを再び入力します。
4. 「**OK**」をクリックして、パスワードを変更します。

AppScan Enterprise Server の SSL 証明書

AppScan Enterprise Server をインストールしたら、有効な SSL 証明書を使用するように構成する必要があります。これを行わないと、AppScan Source for Analysis または AppScan Source コマンド行インターフェース (CLI) - あるいは Windows および Linux 上の AppScan Source for Development からサーバーにログインするときに、「信頼できない接続」というメッセージが表示されます。

SSL 証明書の保管場所

永続的に受け入れられた証明書は、<data_dir>¥config¥cacertspersonal と <data_dir>¥config¥cacertspersonal.pem (<data_dir> は、ご使用の AppScan Source プログラム・データの場所です。詳しくは、 360 ページの『インストールとユーザー・データ・ファイルの場所』 を参照してください。) に保管されます。証明書を永続的に保管する必要がなくなった場合は、これらの 2 つのファイルを削除してください。

AppScan Source for Automation と SSL 証明書の検証

デフォルトでは、AppScan Source for Automation を使用すると、証明書が自動的に受け入れられます。この動作は、Automation Server 構成ファイル (<data_dir>¥config¥ounceautod.ozsettings (<data_dir> は、ご使用の AppScan Source プログラム・データの場所です。詳しくは、 360 ページの『インストールとユーザー・データ・ファイルの場所』 を参照してください。)) の

ounceautod_accept_ssl 設定によって決まります。この設定を編集して value="true" を value="false" に設定すると、SSL 検証が試行され、無効な証明書が検出された場合は、AppScan Enterprise Console へのログインまたは公開が失敗します。

AppScan Source コマンド行インターフェース (CLI) と SSL 証明書の検証

デフォルトでは、CLI login コマンドを使用すると、SSL 検証が試行され、無効な証明書が検出された場合 (別の AppScan Source クライアント製品でログインしたときに、証明書を永続的に受け入れていない場合) は AppScan Enterprise Console へのログインまたは公開がエラーとなって失敗します。この動作は、login コマンドの実行時にオプションの -acceptssl パラメーターを使用することにより変更できます。このパラメーターを使用すると、SSL 証明書が自動的に受け入れられます。

AppScan Source とアクセシビリティー

アクセシビリティーは、動作や視界に制約があるなど、身体に障害を持つユーザーに影響します。 アクセシビリティーに問題があると、ソフトウェア製品の正常な使用に支障をきたすことがあります。 このトピックでは、AppScan Source のアクセシビリティーに関する既知の問題と、その回避策について概要を説明します。

AppScan Source インストーラーと JAWS 画面読み上げソフトウェアの併用

AppScan Source インストーラーを実行する際に Freedom Scientific JAWS (http://www.freedomscientific.com/products/fs/jaws-product-page.asp) を使用するには、Java Access Bridge を AppScan Source JVM にインストールする必要があります。これにより、インストーラー・パネルのラベルとコントロールを JAWS で正しく読み上げることができるようになります。

- Java Access Bridge の詳細 (ダウンロードのリンクやインストールの説明を含む) については、http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136191.html を参照してください。
- Java Access Bridge をインストールするための InstallAnywhere の要件について詳しくは、http://kb.flexerasoftware.com/selfservice/documentLink.do?externalID=Q200311 を参照してください。

説明テキスト付きのユーザー・インターフェース・パネルで JAWS 画面読み上げソフトウェアを使用する

AppScan Source ユーザー・インターフェースの多くの部分に説明テキストが表示されます。 この説明テキストを読み上げられるようにするには、ほとんどの場合、JAWS で Insert+B のキー・ストロークを使用する必要があります。

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。 本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。 これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。 ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒103-8510
東京都中央区日本橋箱崎町19番21号
日本アイ・ビー・エム株式会社
法務・知的財産
知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。

IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態で提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。

国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。 本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。 IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。 そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性がありますが、その測定値が、一般に利用可能なシステムのものと同じである保証はありません。 さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。 お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。 IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確証できません。 IBM 以外の製品の性能に関する質問は、それらの製品の供給者にお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があり、単に目標を示しているものです。

表示されている IBM の価格は IBM が小売り価格として提示しているもので、現行価格であり、通知なしに変更されるものです。卸価格は、異なる場合があります。

本書はプランニング目的としてのみ記述されています。記述内容は製品が使用可能になる前に変更になる場合があります。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。 より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。 このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。 従って IBM は、これらのサンプル・プログラムについて信頼性、

利便性もしくは機能性があることをほのめかしたり、保証することはできません。お客様は、IBM のアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。

© (お客様の会社名) (西暦年).このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。 © Copyright IBM Corp. _年を入れる_. All rights reserved.

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

商標

IBM、IBM ロゴおよび ibm.com® は、世界の多くの国で登録された International Business Machines Corporation の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、http://www.ibm.com/legal/copytrade.shtml をご覧ください。

Adobe、PostScript は、Adobe Systems Incorporated の米国およびその他の国における登録商標または商標です。

IT Infrastructure Library は英国 Office of Government Commerce の一部である the Central Computer and Telecommunications Agency の登録商標です。

インテル、Intel、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Intel Centrino、Intel Centrino ロゴ、Celeron、Intel Xeon、Intel SpeedStep、Itanium、および Pentium は、Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

ITIL は英国 Office of Government Commerce の登録商標および共同体登録商標であって、米国特許商標庁にて登録されています。

UNIX は The Open Group の米国およびその他の国における登録商標です。

Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。

Cell Broadband Engine は、Sony Computer Entertainment, Inc.の米国およびその他の国における商標であり、同社の許諾を受けて使用しています。

Linear Tape-Open、LTO、LTO ロゴ、Ultrium、および Ultrium ロゴは、HP、IBM Corp. および Quantum の米国およびその他の国における商標です。

著作権

第 2 章 アプリケーションおよびプロジェクトの構成

スキャンの前に、アプリケーションおよびプロジェクトを構成する必要があります。このセクションでは、アプリケーション・ディスカバリー・アシスタント、新規アプリケーション・ウィザード、および新規プロジェクト・ウィザードについて説明します。AppScan Source for Analysis の属性を構成する方法についても説明します。さらに、このセクションでは、スキャンのために既存のアプリケーションおよびプロジェクトを追加する方法と、ファイルをプロジェクトに追加する方法についても説明します。

AppScan Source for Analysis の構成には、アプリケーションの作成、ソース・コードの構成、および属性の構成が含まれます。 構成およびスキャンの後で、トリアージに進みます。ソース・コードは、「プロパティー」ビューで構成しても、新規プロジェクト・ウィザードを使用して構成しても構いません。この章では、ウィザードを使用する方法を説明します。アプリケーション・プロパティーおよびプロジェクト・プロパティーの概要については、 319 ページの『「プロパティー」ビュー』を参照してください。

AppScan Source for Analysis が使用するアプリケーション/プロジェクトのモデルは、Microsoft Visual Studio、Eclipse 、Rational Application Developer for WebSphere Software (RAD) の各プロジェクト、あるいは AppScan Source ユーティリティーを使用して以前に作成した AppScan Source プロジェクトを直接インポートするモデルです (詳しくは、「*IBM Security AppScan Source Utilities* ユーザー・ガイド」を参照してください)。

タイプも、含まれている言語もさまざまであるプロジェクトを追加および構成できます (ターゲット・コード・ベースおよびそのビルド・プロシージャーから収集した設定を指定します)。 構成中に、スキャンから除外するディレクトリーおよびファイルを指定できます。

スキャンの前に、アプリケーション およびプロジェクト を構成する必要があります。プロジェクトは、スキャン対象のファイル一式および使用される設定 (構成) です。

AppScan Source アプリケーションおよびプロジェクト・ファイル

AppScan Source のアプリケーションおよびプロジェクトには、スキャンおよびトリアージのカスタマイズに必要な構成情報を保守する対応ファイルがあります。プロジェクトをビルドするために必要な構成情報 (依存関係やコンパイラー・オプションなど) は、AppScan Source がプロジェクトを正常にスキャンするために必要な構成情報と非常に似ているため、これらのファイルはソース・コードと同じディレクトリーに配置することをお勧めします。 ベスト・プラクティスとしては、これらのファイルをソース・コントロール・システムで管理する方法が挙げられます。

AppScan Source for Analysis で作成されたアプリケーションおよびプロジェクトには、それぞれ `.paf` および `.ppf` という拡張子が付けられます。 これらのファイルは、AppScan Source for Analysis、AppScan Source for Automation、および

AppScan Source コマンド行インターフェース でアプリケーションまたはプロジェクトを手動で作成および構成するときに生成されます。

Windows では、Visual Studio ソリューションとプロジェクトを AppScan Source for Analysis、AppScan Source for Automation、および AppScan Source コマンド行インターフェース にインポートすると、そのファイル (拡張子は `.sln.gaf` および `.vcproj.gpf`) が作成されます。

macOS では、Xcode ディレクトリーおよびプロジェクトをインポートすると、そのファイル (拡張子は `.xcodeproj.gaf` および `.xcodeproj.gpf`) が作成されます。同様に、Xcode ワークスペースをインポートすると、`.xcworkspace.gaf` 拡張子を持つファイルが作成されます。

注**:** Eclipse Importer を Eclipse または Rational Application Developer for WebSphere Software (RAD) ワークスペースで実行すると、AppScan Source は、`.ewf` および `.epf` という拡張子の中間ファイルを作成します。これらのファイルは、AppScan Source for Analysis への初回インポート時、および後でスキャンを実行するときに必要です。

重要**:** 開発環境内に依存関係を持つ AppScan Source プロジェクト (例えば、IBM MobileFirst Platform プロジェクト) を処理する場合は、必ず、プロジェクトをインポートする前に開発環境でビルドしてください。プロジェクトをインポートした後、その中のファイルを変更した場合は、AppScan Source でスキャンする前に開発環境で再ビルドしてください (そうしないと、ファイルに対する変更は AppScan Source によって無視されます)。

表 *2. AppScan Source* ファイル

AppScan Source ファイル拡張子	説明
`ppf`	• AppScan Source プロジェクト・ファイル • AppScan Source for Analysis またはサポートされている AppScan Source ユーティリティーでプロジェクトを作成するときに生成されます。 • 名前はユーザー指定
`paf`	• AppScan Source アプリケーション・ファイル • AppScan Source for Analysis またはサポートされている AppScan Source ユーティリティーでアプリケーションを作成するときに生成されます。 • 名前はユーザー指定

表 2. *AppScan Source* ファイル (続き)

AppScan Source ファイル拡張子	説明
sln.gaf	• Visual Studio ソリューションをインポートする場合に生成される AppScan Source アプリケーション・ファイル • カスタム・アプリケーション情報 (除外設定やバンドルなど) を保持するために使用されます。 • インポートするワークスペースまたはソリューションの名前が採用されます。 例: d:¥my_apps¥myapp.sln d:¥my_apps¥myapp.sln.gaf
vcproj.gpf	• Visual Studio プロジェクトをインポートする場合に生成される AppScan Source プロジェクト・ファイル • カスタム・プロジェクト情報 (パターンや除外設定など) を保持するために使用されます。 • インポートするプロジェクトの名前が採用されます。例: d:¥my_projects¥myproject.vcproj d:¥my_projects¥myproject.vcproj.gpf
xcodeproj.gaf	• Xcode ディレクトリーをインポートする場合に生成される AppScan Source アプリケーション・ファイル • カスタム・アプリケーション情報 (除外設定やバンドルなど) を保持するために使用されます。 • インポートするワークスペースまたはソリューションの名前が採用されます。 例: /Users/myUser/myProject.xcodeproj /Users/myUser/myProject.xcodeproj.gaf

表 2. *AppScan Source* ファイル (続き)

AppScan Source ファイル拡張子	説明
xcodeproj.gpf	• Xcode プロジェクトをインポートする場合に生成される AppScan Source プロジェクト・ファイル • カスタム・プロジェクト情報 (パターンや除外設定など) を保持するために使用されます。 • インポートするプロジェクトの名前が採用されます。例: /Users/myUser/myProject.xcodeproj /Users/myUser/myProject.xcodeproj.gpf
xcworkspace.gaf	• Xcode ワークスペースをインポートする場合に生成される AppScan Source アプリケーション・ファイル • カスタム・アプリケーション情報 (除外設定やバンドルなど) を保持するために使用されます。 • インポートするワークスペースの名前が採用されます。例: /Users/myUser/myProj.xcworkspace.gaf
ewf	• Eclipse ワークスペース・ファイル • Eclipse ワークスペースを AppScan Source にインポートするときに生成されます。 • Eclipse エクスポーターが Eclipse ワークスペース内の情報に基づいてファイルを作成し、AppScan Source がそのファイルをインポートします。
epf	• Eclipse プロジェクト・ファイル • Eclipse プロジェクトを AppScan Source にインポートするときに生成されます。 • Eclipse エクスポーターが Eclipse プロジェクト内の情報に基づいてファイルを作成し、AppScan Source がそのファイルをインポートします。

ヒント: サポートされているビルド統合ツール (Ounce/Ant または Ounce/Maven など) を使用して AppScan Source アプリケーション・ファイルおよびプロジェクト・ファイルを生成する場合、これらを開発チーム間で共有できるように、ビルド自動化処理の一環としてソース・コントロール・システムでこれらのファイルを更新することをお勧めします。開発者が、ソース・コントロール・システムでファイルのローカル・ビューを更新すると、AppScan Source のアプリケーション・ファ

イルおよびプロジェクト・ファイルも更新されます。これにより、チーム全体が、一貫性のあるファイル・セットを使用して作業できます。

注**:** AppScan Source for Analysis、AppScan Source for Automation、および AppScan Source コマンド行インターフェース でサポートされているインポート・ファイルのバージョンを確認するには、http://www.ibm.com/support/docview.wss?uid=swg27027486を参照してください。 このページで、使用している AppScan Source のバージョンのタブを選択してから、使用している AppScan Source コンポーネントを選択します。AppScan Source が他の開発環境からのファイルのオープンおよびスキャンをサポートする場合、そのサポートは、「**Supported Software**」タブの「**Compilers and Languages**」セクションにリストされています。

アプリケーションの構成

新規アプリケーション・ウィザードまたはアプリケーション・ディスカバリー・アシスタントを使用して、アプリケーションを作成できます。アプリケーション・ディスカバリー・アシスタントでは、アプリケーションの設定が自動化されますが、新規アプリケーション・ウィザードでは、アプリケーションを追加して、構成処理を行うことができます。このウィザードを使用すれば、プロジェクトを手動で作成したり、既存のプロジェクトをアプリケーションに追加したりすることができます。 このセクションでは、アプリケーションを追加するためのこれら 2 つの方法と、基本的な構成作業について説明します。

注**:** アプリケーション・ディスカバリー・アシスタント は、Java ソース・コードおよび Microsoft Visual Studio ソリューション 用のアプリケーションおよびプロジェクトや、Java プロジェクトが含まれる Eclipse または IBM Rational Application Developer for WebSphere Software (RAD) ワークスペース用のアプリケーションおよびプロジェクトを迅速に作成して構成します。その他のサポート対象言語のアプリケーションを作成するには、新規アプリケーション・ウィザードを使用するか、AppScan Source for Analysis にサポート対象のアプリケーションをインポートします。

プロジェクトを追加する前に、新規アプリケーションを作成する (40 ページの『新規アプリケーション・ウィザードによる新規アプリケーションの作成』または 41 ページの『アプリケーション・ディスカバリー・アシスタントを使用したアプリケーションおよびプロジェクトの作成』を参照) か、既存のアプリケーションを追加する (45 ページの『既存のアプリケーションの追加』を参照) 必要があります。Microsoft Visual Studio を使用している場合は、既にソース・ファイルがプロジェクト内に配置されています。 AppScan Source for Analysis では、ソリューションをインポートし、それらを AppScan Source アプリケーションとして扱うことができます。

以下の表は、AppScan Source for Analysis で開いてスキャンできるアプリケーション・ファイル・タイプのリストです。

表 3. サポートされるアプリケーション・ファイル・タイプ

アプリケーション	ファイル・タイプ
Microsoft Visual Studio **注:** AppScan Source for Analysis、AppScan Source for Automation、および AppScan Source コマンド行インターフェース でサポートされているインポート・ファイルのバージョンを確認するには、http://www.ibm.com/support/docview.wss?uid=swg27027486を参照してください。 このページで、使用している AppScan Source のバージョンのタブを選択してから、使用している AppScan Source コンポーネントを選択します。AppScan Source が他の開発環境からのファイルのオープンおよびスキャンをサポートする場合、そのサポートは、「**Supported Software**」タブの「**Compilers and Languages**」セクションにリストされています。	`.sln` (ソリューション)
• Eclipse ワークスペース (Java のみ) • RAD ワークスペース (Java のみ) ワークスペース・スキャンでサポートされる Eclipse と RAD のバージョンを確認するには、AppScan Source のシステム要件を参照してください。	`<workspace directory>` または `.ewf` ワークスペース・ディレクトリーには、追加ディレクトリー `.metadata` が含まれます。
AppScan Source アプリケーション・ファイル	`.paf`

ヒント: 「エクスプローラー」ビュー内に、インポートされたアプリケーションを示すアイコンが表示されます(101 ページの『アプリケーションおよびプロジェクトのインディケーター』を参照)。

注: 新規アプリケーション・ウィザードおよび新規プロジェクト・ウィザードを使用してアプリケーションおよびプロジェクトを作成すると、ウィザードで入力された「名前」に従ってファイル名が自動的に割り当てられます (例えば、プロジェクトを作成している場合に「名前」フィールドに **MyProject** と入力すると、プロジェクトのファイル名は `MyProject.ppf` になります)。アプリケーション名およびプロジェクト名は、「プロパティー」ビューを使用して名前変更できます。

新規アプリケーション・ウィザードによる新規アプリケーションの作成

手順

1. 以下のアクションのいずれかを実行します。
   - メインメニュー・バーから「ファイル」 > 「アプリケーションの追加」 > 「新規アプリケーションの作成」を選択します。

- 「エクスプローラー」ビューのツールバーで、「アプリケーション・メニューの追加」下矢印ボタンをクリックして、メニューから「新規アプリケーションの作成」を選択します。
- 「エクスプローラー」ビューで、「すべてのアプリケーション」を右クリックし、メニューから「アプリケーションの追加」 > 「新規アプリケーションの作成」を選択します。

2. アプリケーションの「名前」を入力します。
3. アプリケーションの保存先の「作業ディレクトリー」を参照します。 新規アプリケーションのファイル名の拡張子は `.paf` になります。
4. 「次へ」をクリックして、アプリケーションの構成要素であるプロジェクトを構成するか、「終了」をクリックして、プロジェクトを構成しないでアプリケーションを追加します。 プロジェクトの構成および追加に関するヘルプについては、後でこのセクションで説明します。

アプリケーション・ディスカバリー・アシスタントを使用したアプリケーションおよびプロジェクトの作成

AppScan Source には、強力な アプリケーション・ディスカバリー・アシスタントが組み込まれています。これを使用すると、Java ソース・コードおよび Microsoft Visual Studio ソリューション用のアプリケーションおよびプロジェクトを迅速に作成および構成できます。アプリケーション・ディスカバリー・アシスタントを使用して、Java プロジェクトが含まれている Eclipse ワークスペースまたは Rational Application Developer for WebSphere Software (RAD) ワークスペースを見つけることもできます。アプリケーション・ディスカバリー・アシスタント では、ソース、ソリューション、またはワークスペース・ディレクトリーをポイントすることができます。その後の処理は AppScan Source によって行われます。

このタスクについて

アプリケーション・ディスカバリー・アシスタント を使用して、Java ソース、Microsoft Visual Studio ソリューション、または Eclipse の各ワークスペースの組み合わせが含まれているロケーションを検索できます。 アプリケーション・ディスカバリー・アシスタントの最終パネルでは、Java のみのアプリケーション/プロジェクト構造の設定を指定できます。 このパネルは、Microsoft Visual Studio ソリューションまたは Eclipse ワークスペースのアプリケーション・ファイルおよびプロジェクト・ファイルの配置とは関係がありません。アプリケーション・ファイルは、ソリューションまたはワークスペースのルートに自動的に置かれ、プロジェクト・ファイルは、個々のソリューションまたはワークスペースのプロジェクトのルートに自動的に置かれます。

手順

1. 以下のアクションのいずれかを実行して、アプリケーション・ディスカバリー・アシスタントを起動します。
   - メインメニュー・バーから「ファイル」 > 「アプリケーションの追加」 > 「アプリケーションのディスカバー」を選択します。
   - 「エクスプローラー」ビューの「クイック・スタート」セクションで、「アプリケーションのディスカバー」を選択します。

- 「エクスプローラー」ビューのツールバーで、「アプリケーション・メニューの追加」下矢印ボタンをクリックして、メニューから「アプリケーションのディスカバー」を選択します。
- 「エクスプローラー」ビューで、「すべてのアプリケーション」を右クリックし、メニューから「アプリケーションの追加」 > 「アプリケーションのディスカバー」を選択します。

2. 「ロケーションの検索」パネルで、スキャンするソース・コード、ソリューション、またはワークスペースが含まれているロケーションを指定します。 さらに、アプリケーションのディスカバリーの完了直後にスキャンが開始するように設定できます。

   このパネルで、「次へ」をクリックして、アプリケーション・ディスカバリー・アシスタントの追加オプション (外部依存関係の指定、除外ルール、Java アプリケーション/プロジェクト構造の設定など) を設定できます。または、「開始」をクリックしてアプリケーション・ディスカバリーを開始できます。 「開始」をクリックすると、以下のようになります。

   - 外部依存関係ロケーションは設定されません。 アプリケーションに外部依存関係があり、それらの依存関係が指定されていない場合は、スキャン結果に悪影響が及びます。
   - すぐに使用可能な除外ルールが使用されます (デフォルトのルールのリストについては、 45 ページの『デフォルトの アプリケーション・ディスカバリー・アシスタント 除外ルール』を参照してください)。
   - Java ソースを探している場合は、1 つのプロジェクトおよびアプリケーションが作成されます (検出されたすべてのソース・ルートが 1 つのプロジェクトに入れられます)。

   「次へ」をクリックすると、次のステップに進みます。

3. 「外部依存関係」パネルで、アプリケーションのそれぞれの外部依存関係のパス (JDK または Web サーバーへのパスなど) を設定します。 このパネルでの作業を完了するには、以下の手順に従います。

   a. 外部依存関係を追加するには、テーブル内をクリックするか「追加」をクリックしてから、外部依存関係パスを入力または参照します。 キーボードで入力したパスを受け入れるには、キーボードの Enter キーを押します。

      ヒント: 依存関係パス・フィールドの編集中に入力を行うと、選択可能なディレクトリーがリストされます。少なくともドライブ名を入力する必要があります。 パスを指定すると、そのパスに含まれているすべてのフォルダーがリストされます。

   b. 外部依存関係パスを削除するには、それを選択して「削除」をクリックします。

   c. 外部依存関係パスを変更するには、パス内をクリックしてから、外部依存関係パスを入力または参照します。

   このパネルで、「次へ」をクリックして、アプリケーション・ディスカバリー・アシスタントの追加オプションを設定できます。または、「開始」をクリックしてアプリケーション・ディスカバリーを開始できます。 「開始」をクリックすると、以下のようになります。

- すぐに使用可能な除外ルールが使用されます (デフォルトのルールのリストについては、 45 ページの『デフォルトの アプリケーション・ディスカバリー・アシスタント 除外ルール』を参照してください)。
- Java ソースを探している場合は、1 つのプロジェクトおよびアプリケーションが作成されます (検出されたすべてのソース・ルートが 1 つのプロジェクトに入れられます)。

「次へ」をクリックすると、次のステップに進みます。

4. 「除外ルール」パネルで、ファイルおよびディレクトリーをフィルターで除外するためのルールを指定します。 ルールは、PERL、Grep、EGrep、または完全一致の正規表現によって設定します。 例えば、`temp` という名前のディレクトリーを アプリケーション・ディスカバリー検索から除外する場合は、PERL `.*[¥¥/]temp` という除外ルールを追加できます。

   デフォルトで、PERL 正規表現のセットが、一部の共通ディレクトリーを除外するために提供されています (完全なリストについては、 45 ページの『デフォルトの アプリケーション・ディスカバリー・アシスタント 除外ルール』を参照してください)。 このリストを変更するか、新しいルールを作成するには、以下の手順に従います。

   a. 既存の除外ルールを変更するには、ルール内をクリックしてルール・エディターをアクティブにします。 ルールの編集が完了したら、クリックしてそのルールを終了するか、キーボードの **Enter** キーを押します。

      既存のルールの正規表現タイプを変更するには、ルールの「正規表現タイプ」セル内をクリックして、メニューから正規表現タイプを選択します。

   b. 除外ルールを追加するには、「追加」をクリックします。これによって、新しいルールがテーブルに追加されます。このルールは、ルールを変更するための上記の説明に従って変更することができます。

   c. 除外ルールを削除するには、そのルールを選択して「削除」をクリックします (現在パネルにリストされている除外ルールをすべて削除するには、「すべて削除」をクリックします)。

   重要**:** 有効な除外ルールは、テーブル内でチェック・マークで示されています。また、無効なルールは赤い `X` で示されています。すべてのルールが有効になるまで、アプリケーション・ディスカバリーを開始したり、アプリケーション・ディスカバリー・アシスタントで作業を続行したりすることはできません。

   ここで、以下のようにします。

   - Java ソースのみを検索している場合は、「次へ」をクリックしてアプリケーション・ディスカバリー・アシスタントアプリケーション/プロジェクト構造の設定を設定できます。または、「開始」をクリックしてアシスタントを実行できます。
   - Microsoft Visual Studio ソリューションまたは Eclipse ワークスペースのみを検索している場合は、「開始」をクリックしてアシスタントを実行します。 「次へ」をクリックすると、アシスタントによって、Java ソースのディスカバリーのみに適用されるパネルに進みます。

   「次へ」をクリックすると、次のステップに進みます。

5. 「アプリケーションおよびプロジェクトの作成」パネルは、Java ソースのディスカバリーにのみ適用されます。 このパネルで、作成するアプリケーションおよびプロジェクトの構造を、以下のように指定します。
   a. 検出されるすべてのソース・ルート用に単一のプロジェクトを作成するには、「プロジェクト」メニューで「単一のプロジェクトを作成」を選択します。 これを選択すると、単一のアプリケーションの作成のみを選択できます。
   b. 検出されるソース・ルートごとに別個のプロジェクトを作成するには、「プロジェクト」メニューで「検出されたソース・ルートごとにプロジェクトを作成」を選択します。 これを選択すると、1 つのアプリケーションを作成するのか、複数のアプリケーションを作成するのかを選択できます。 作成されるすべてのプロジェクトが含まれる単一のアプリケーションを作成するには、「アプリケーション」メニューで「単一のアプリケーションを作成」を選択します。 作成されるプロジェクトごとにアプリケーションを作成するには、「アプリケーション」メニューで「プロジェクトごとにアプリケーションを作成」を選択します。

   さらに、アプリケーションおよびプロジェクト定義ファイルを格納するロケーションを選択します。

   「ファイルを自動的に編成」を選択すると、以下のようになります。
   - 単一のプロジェクトを作成する場合、プロジェクトおよびアプリケーション・ファイルが検索ロケーションに作成されます。
   - 単一のアプリケーションのソース・ルートごとにプロジェクトを作成する場合、ソース・ルートごとのプロジェクト・ファイルがソース・ルートの上位ディレクトリーに作成され、アプリケーション・ファイルが検索ロケーションに作成されます。
   - ソール・ルートごとにプロジェクトを作成し、プロジェクトごとにアプリケーションを作成する場合、ソース・ルートごとのプロジェクト・ファイルおよびアプリケーション・ファイルがソース・ルートの上位ディレクトリーに作成されます。

   ディレクトリーを指定すると、すべてのアプリケーション・ファイルおよびプロジェクト・ファイルが、そのディレクトリーに作成されます。
6. 前のパネルで行った設定を変更する場合は、「戻る」をクリックします。 アプリケーション・ディスカバリーの設定が完了したら、「開始」をクリックして、ソース・ルートの検索ロケーションをスキャンします。

タスクの結果

アプリケーション・ディスカバリーでの作業が完了したら、アプリケーション・ディスカバリーの結果として作成された新しいアプリケーションおよびプロジェクトが、「エクスプローラー」ビューに表示され、スキャンの準備が整いました (アプリケーション・ディスカバリーの完了直後にスキャンが開始するように設定している場合は、スキャンが開始します)。

ディスカバリー中に問題が発生すると、アプリケーション・ディスカバリー・アシスタントは、完了時にディスカバリー・レポートを提供します。 例えば、「外部依

存関係」パネルに指定されていない外部依存関係がアプリケーションにある場合、このレポートには外部依存関係を解決できないことを示す警告が含まれます。 ディスカバリー・レポートで、以下のようにします。

- 「終了」をクリックしてアプリケーションとプロジェクトを作成します。 「警告を無視してスキャンを続行」を選択すると、アプリケーションとプロジェクトが直ちにスキャンされます。
- 「戻る」をクリックして アプリケーション・ディスカバリー・アシスタントの設定を変更するか、アプリケーション・ディスカバリーを再実行します。
- アプリケーションもプロジェクトも作成しないでディスカバリー・レポートを閉じるには、「キャンセル」をクリックします。

デフォルトの アプリケーション・ディスカバリー・アシスタント 除外ルール

アプリケーション・ディスカバリー・アシスタントの使用時に、「除外ルール」パネルを変更しない場合、または検索ディレクトリーの指定後にアプリケーション・ディスカバリーを開始する場合は、除外ルールが使用されます。 このトピックでは、デフォルトの アプリケーション・ディスカバリー除外ルールをリストします。

表 4. デフォルトのアプリケーション・ディスカバリー除外ルール

除外ルール	正規表現タイプ
.*[¥¥/]example	PERL
.*[¥¥/]test	PERL
.*[¥¥/]demo	PERL
.*[¥¥/]sample	PERL

既存のアプリケーションの追加

スキャンのために既存のアプリケーションを追加するには、「エクスプローラー」ビューにそれらをドラッグ・アンド・ドロップするか、または「アプリケーションの追加」アクションを使用します。さらに、WAR ファイルと EAR ファイルを追加するには、「エクスプローラー」ビューにドラッグ・アンド・ドロップします。

既存のアプリケーションを追加する方法については、以下のトピックを参照してください。

- 『ユーザー・インターフェース・アクションによる既存のアプリケーションの追加』
- 46 ページの『ドラッグ・アンド・ドロップによる既存のアプリケーションの追加』

ユーザー・インターフェース・アクションによる既存のアプリケーションの追加

手順

1. 以下のアクションのいずれかを実行します。
   - メイン・ワークベンチ・メニューで、「ファイル」 > 「アプリケーションの追加」 > 「既存のアプリケーションを開く」を選択します。

- 「エクスプローラー」ビューのツールバーで、「アプリケーション・メニューの追加」下矢印ボタンをクリックして、メニューから「既存のアプリケーションを開く」を選択します。
- 「エクスプローラー」ビューで、「すべてのアプリケーション」を右クリックし、メニューから「アプリケーションの追加」 > 「既存のアプリケーションを開く」を選択します。

2. 保存されたアプリケーション・プロファイル (.paf、.sln、.dsw、または .ewf) が格納されているディレクトリーを選択します。

   注: AppScan Source for Analysis、AppScan Source for Automation、および AppScan Source コマンド行インターフェース でサポートされているインポート・ファイルのバージョンを確認するには、http://www.ibm.com/support/docview.wss?uid=swg27027486を参照してください。 このページで、使用している AppScan Source のバージョンのタブを選択してから、使用している AppScan Source コンポーネントを選択します。AppScan Source が他の開発環境からのファイルのオープンおよびスキャンをサポートする場合、そのサポートは、「**Supported Software**」タブの「**Compilers and Languages**」セクションにリストされています。

3. アプリケーション・ファイルを開きます。

ドラッグ・アンド・ドロップによる既存のアプリケーションの追加

手順

1. ワークステーションで、スキャンのために追加するアプリケーション (.paf、.war、 .ear、.sln、.dsw、 または .ewf) を見つけます。 .war または .ear ファイルが含まれるディレクトリーを追加することもできます (一部のアプリケーション・サーバーでは、ドロップイン・フォルダー と呼ばれます)。

   注: Eclipse ワークスペース・ディレクトリーはドラッグ・アンド・ドロップできません。

   注: .war ファイルや .ear ファイル、または .war ファイルや .ear ファイルが含まれるディレクトリーを追加する場合、ファイルはローカル・ファイル・システム上またはマップされたドライブ内に配置する必要があります。

   注: AppScan Source for Analysis、AppScan Source for Automation、および AppScan Source コマンド行インターフェース でサポートされているインポート・ファイルのバージョンを確認するには、http://www.ibm.com/support/docview.wss?uid=swg27027486を参照してください。 このページで、使用している AppScan Source のバージョンのタブを選択してから、使用している AppScan Source コンポーネントを選択します。AppScan Source が他の開発環境からのファイルのオープンおよびスキャンをサポートする場合、そのサポートは、「**Supported Software**」タブの「**Compilers and Languages**」セクションにリストされています。

2. アプリケーションを選択して、「エクスプローラー」ビューにドラッグします。
3. 選択した項目を「すべてのアプリケーション」ノードまたはその下にドロップします。

4. .war ファイルや .ear ファイル、または .war ファイルや .ear ファイルが含まれるディレクトリーを追加する場合、ダイアログ・ボックスが開き、ファイル (複数の場合あり) をデプロイするアプリケーション・サーバーを指定することができます。 このダイアログ・ボックスに入力したら、「**OK**」をクリックします。

複数のアプリケーションの追加

AppScan Source for Analysis での作業を初めて開始するときに、アプリケーションを一度に 1 つのみ追加するのではなく、複数のアプリケーションをインポートすると便利です。「アプリケーションの選択」ダイアログ・ボックスを使用すると、AppScan Source アプリケーション・ファイル (.paf) または Visual Studio ソリューション・ファイル (.sln) の検索先にするルート・ディレクトリーを選択できます。複数のアプリケーションをスキャンの対象に追加するには、該当のアプリケーションを「エクスプローラー」ビューにドラッグ・アンド・ドロップします。

複数のアプリケーションを追加する方法については、以下のトピックを参照してください。

- 『ユーザー・インターフェース・アクションによる複数のアプリケーションの追加』
- 48 ページの『ドラッグ・アンド・ドロップによる複数のアプリケーションの追加』

注**:** 複数の WAR ファイルおよび EAR ファイルを追加する場合、ファイルが含まれるディレクトリーをドラッグ・アンド・ドロップすることで実行可能です。詳しくは、 46 ページの『ドラッグ・アンド・ドロップによる既存のアプリケーションの追加』を参照してください。

ユーザー・インターフェース・アクションによる複数のアプリケーションの追加

手順

1. ワークベンチのメインメニューで、「ファイル」 > 「アプリケーションの追加」 > 「複数のアプリケーション」を選択します。
2. 「アプリケーションの選択」ダイアログ・ボックスで、インポートするアプリケーションが格納されているルート・ディレクトリーを参照します。サブディレクトリー内を検索するには、「サブディレクトリーを再帰処理する」チェック・ボックスを選択します。
3. 以下のアクションのいずれかを実行します。
   - 「終了」をクリックして、アプリケーションをインポートし、それらを「エクスプローラー」ビューに追加します。
   - 「次へ」をクリックして検索結果を表示し、インポートするアプリケーションを選択します。 次に「終了」をクリックします。

   注**:** AppScan Source for Analysis、AppScan Source for Automation、および AppScan Source コマンド行インターフェース でサポートされているインポート・ファイルのバージョンを確認するには、http://www.ibm.com/support/docview.wss?uid=swg27027486を参照してください。 このページで、使用している AppScan Source のバージョンのタブを選択してから、使用している

AppScan Source コンポーネントを選択します。AppScan Source が他の開発環境からのファイルのオープンおよびスキャンをサポートする場合、そのサポートは、「**Supported Software**」タブの「**Compilers and Languages**」セクションにリストされています。

ドラッグ・アンド・ドロップによる複数のアプリケーションの追加

手順

1. ワークステーションで、スキャンのために追加するアプリケーション (`.paf`、`.sln`、`.dsw`、または `.ewf` の各ファイル) を見つけます。

   注**:** Eclipse ワークスペース・ディレクトリーはドラッグ・アンド・ドロップできません。

   注**:** AppScan Source for Analysis、AppScan Source for Automation、および AppScan Source コマンド行インターフェース でサポートされているインポート・ファイルのバージョンを確認するには、http://www.ibm.com/support/docview.wss?uid=swg27027486を参照してください。 このページで、使用している AppScan Source のバージョンのタブを選択してから、使用している AppScan Source コンポーネントを選択します。AppScan Source が他の開発環境からのファイルのオープンおよびスキャンをサポートする場合、そのサポートは、「**Supported Software**」タブの「**Compilers and Languages**」セクションにリストされています。
2. アプリケーションを選択または複数選択して、「エクスプローラー」ビューにドラッグします。
3. 選択した項目を「すべてのアプリケーション」ノードまたはその下にドロップします。

Apache Tomcat および WebSphere Application Server Liberty プロファイル・アプリケーション・サーバーからの既存の Java アプリケーションのインポート

サポートされているアプリケーション・サーバーにデプロイ済みの既存の Java アプリケーションがある場合、それらのアプリケーションを自動的に AppScan Source にインポートすることができます。

始める前に

サポートされている Apache Tomcat および WebSphere Application Server Liberty プロファイルのバージョンについては、AppScan Source のシステム要件を参照してください。このページで、使用している AppScan Source のバージョンのタブを選択してから、AppScan Source for Analysis コンポーネントを選択します。サポートされているアプリケーション・サーバーは、「**Supported Software**」セクションで見つかります。

手順

1. 以下のアクションのいずれかを実行します。
   - メイン・ワークベンチ・メニューから「ファイル」 > 「アプリケーションの追加」 > 「アプリケーション・サーバーからのインポート」を選択します。

- 「エクスプローラー」ビューのツールバーで、「アプリケーション・メニューの追加」下矢印ボタンをクリックして、メニューから「アプリケーション・サーバーからのインポート」を選択します。
- 「エクスプローラー」ビューで、「すべてのアプリケーション」を右クリックして、メニューから「アプリケーションの追加」 > 「アプリケーション・サーバーからのインポート」を選択します。

2. 「アプリケーション・サーバーからのインポート」ダイアログ・ボックスで、「参照」をクリックして、アプリケーション・サーバーがインストールされている場所を見つけて選択するか、フィールドにサーバー・パスとディレクトリーを入力して、「検索」をクリックし、入力した場所のアプリケーションを検索します。この場所がサポートされているアプリケーション・サーバーとして認識されている場合、選択可能なアプリケーションがダイアログ・ボックスの「インポートするアプリケーション」セクションにリストされます。このセクションで、インポートするアプリケーションを選択して、「**OK**」をクリックします。
3. アプリケーション・サーバーからインポートされるアプリケーションごとに AppScan Source アプリケーションが作成されます。

タスクの結果

WebSphere Application Server Liberty プロファイル・サーバー (WebSphere Application Server バージョン 8.5 以上) からインポートする場合、手動での JSP プリコンパイルが必要であることを示すメッセージを受け取ることがあります。これは、Liberty プロファイル・サーバーにスタンドアロンの JSP コンパイラーが含まれていないために起こります。このメッセージを受け取った場合、インポートの結果として作成されたアプリケーションをすべて削除してから、 50 ページの『WebSphere Application Server Liberty プロファイル用のプリコンパイル済み JavaServer Pages の生成』 の説明に従い、アプリケーション・サーバーからアプリケーションを再びインポートします。

アプリケーションがインポートされると、デフォルトでは、AppScan Source は、その JSP ファイルおよび `web-inf/classes` のコンテンツのみスキャンします。`web-inf/lib` のコンテンツはスキャンされません。他のファイルをスキャンしたい場合は、プロジェクト・プロパティーを使用してスキャン対象の追加ファイル拡張子を設定することができます (283 ページの『ファイル拡張子』 を参照)。例えば、`.jar` ファイルをスキャンしたい場合 (`web-inf/lib` 内の .jar ファイルを含む)、 93 ページの『アプリケーションおよびプロジェクトのプロパティーの変更』 のプロジェクト・プロパティーを変更するための指示に従ってください。 プロジェクトの「プロパティー」ビューでは、 283 ページの『ファイル拡張子』タブを選択します。ビューの「追加の拡張子」セクションでは、「拡張子の追加」をクリックします。「新しい拡張子」ダイアログ・ボックスで、「拡張子」フィールドに `jar` を入力し、「この拡張子のファイルをスキャン」を選択して「**OK**」をクリックします。ビューの右上にある「保存」をクリック (またはメインメニューの「ファイル」 > 「保存」を選択) して、その後プロジェクトを再スキャンします。スキャンしたくないファイルが存在する場合、「プロジェクト」ビューの 284 ページの『ソース』タブを使用して、それらをスキャンから除外できます。

サーバー上のアプリケーションが変更されて、変更された内容で AppScan Source アプリケーションを最新表示する場合は、上記のステップを再実行する必要があり

ます (作成された元のアプリケーションを最初に削除する必要はありません。AppScan Source が再インポート時に自動的に削除します。)。

注: サーバーから .war ファイルをインポートした後に、別のサーバーから同じ名前の別の .war ファイルをインポートすると、2 番目の .war ファイルが最初のファイルを上書きします。 これを防ぐには、2 番目の .war ファイルをインポートする前に名前変更します。

WebSphere Application Server Liberty プロファイル用のプリコンパイル済み JavaServer Pages の生成

WebSphere Application Server Liberty プロファイル (WebSphere Application Server バージョン 8.5 以上) からアプリケーションをインポートする場合、手動での JSP プリコンパイルが必要です (Liberty プロファイルにはスタンドアロンの JSP コンパイラーは含まれていません)。このトピックでは、手動での JSP プリコンパイルのセットアップに必要なステップについて説明します。

手順

1. WebSphere Application Server Network Deployment Knowledge Center に記載されている、Liberty プロファイル・サーバーを作成するための説明に従います。WebSphere Application Server バージョン 8.5.5 の場合は、トピック開発者ツールを使用した Liberty プロファイル・サーバーの作成を参照してください。
2. Liberty プロファイルの server.xml ファイルで、以下を server description セクションに追加します。

   ```
   <jspEngine prepareJSPs="0"/>
   <webContainer deferServletLoad="false"/>
   ```

   例:

   ```
   <server description="new server">

       <!-- Enable features -->
       <featureManager>
           <feature>jsp-2.2</feature>
           <feature>localConnector-1.0</feature>
          <feature>appSecurity-2.0</feature>
           <feature>restConnector-1.0</feature>
       </featureManager>

       <!-- To access this server from a remote client
            add a host attribute to the following element,
            e.g. host="*" -->
       <httpEndpoint httpPort="9080" httpsPort="9443" id="defaultHttpEndpoint"/>

   ...
       <jspEngine prepareJSPs="0"/>
       <webContainer deferServletLoad="false"/>
   ...
   </server>
   ```

   server.xml ファイルについては、WebSphere Application Server Center で説明されています。 WebSphere Application Server バージョン 8.5.5 の場合は、トピックLiberty プロファイル: server.xml ファイルの構成エレメントを参照してください。
3. 以下のいずれかの方法を使用して、サーバーをデバッグ・モードで始動します。

- `-Dwas.debug.mode=true` JVM 引数を追加します。Setting generic JVM arguments in the WebSphere Application Server V8.5 Liberty profile を参照してください。
- WebSphere Application Server Network Deployment Knowledge Center に記載されている、サーバーの始動と停止の手順に従います。WebSphere Application Server バージョン 8.5.5 の場合は、トピック開発者ツールを使用したサーバーの始動および停止を参照してください。

タスクの結果

上記のステップを完了した後、 48 ページの『Apache Tomcat および WebSphere Application Server Liberty プロファイル・アプリケーション・サーバーからの既存の Java アプリケーションのインポート』のステップに従い、Java アプリケーションを WebSphere Application Server Liberty プロファイルからインポートします。

Eclipse または Eclipse ベースの製品ワークスペースの追加

Java または IBM MobileFirst Platform(あるいはその両方) のプロジェクトを含む、Eclipse または Rational Application Developer for WebSphere Software (RAD) のワークスペースがある場合、それを AppScan Source for Analysis にインポートできます。

始める前に

ワークスペースを追加する前に、 52 ページの『Eclipse プロジェクトおよび Rational Application Developer for WebSphere Software (RAD) プロジェクトの開発環境の構成』で説明されているように開発環境がインストールおよび更新されていることを確認してください。

手順

1. 以下のアクションのいずれかを実行します。
   - メイン・ワークベンチ・メニューで、「ファイル」 > 「アプリケーションの追加」 > 「既存の **Eclipse** ベースのワークスペースのインポート」を選択します。
   - 「エクスプローラー」ビューのツールバーで、「アプリケーション・メニューの追加」下矢印ボタンをクリックして、メニューから「既存の **Eclipse** ベースのワークスペースのインポート」を選択します。
   - 「エクスプローラー」ビューで、「すべてのアプリケーション」を右クリックし、メニューから「アプリケーションの追加」 > 「既存の **Eclipse** ベースのワークスペースのインポート」を選択します。
2. 「ワークスペース・タイプ」を選択します。
3. ワークスペースを参照してディレクトリーを選択し、「**OK**」をクリックしてワークスペースを追加します。

Eclipse プロジェクトおよび Rational Application Developer for WebSphere Software (RAD) プロジェクトの開発環境の構成

Eclipse または Rational Application Developer for WebSphere Software (RAD) のプロジェクトをインポートする前に、開発環境を適切に構成する必要があります。それぞれのプロジェクト・タイプの基礎となるのは Eclipse ですが、AppScan Source では、バージョン間の違いが区別されます。

AppScan Source でサポートされる Eclipse および Rational Application Developer for WebSphere Software (RAD) のバージョンについては、http://www.ibm.com/support/docview.wss?uid=swg27027486 を参照してください。

開発環境を構成する方法について詳しくは、以下のヘルプ・トピックを参照してください。

- 『Eclipse または Application Developer の更新』
- 53 ページの『Eclipse ワークスペース・インポーター: Eclipse または Rational Application Developer for WebSphere Software (RAD) の設定構成』

Eclipse または Application Developer の更新

AppScan Source の外部にある Eclipse または Application Developer の環境では、適切なソフトウェア更新がインストールされていることを確認する必要があります。 以下では、更新を取得およびインストールする方法について説明します。手順は、バージョンごとに異なることがあります。

始める前に

重要**:** AppScan Source for Development には、バージョン 1.5 以降の Java ランタイム環境 (JRE) が必要です。 ご使用の環境がこの要件を満たさない JRE を指している場合は、Eclipse インストール・ディレクトリー内の `eclipse.ini` ファイルを編集して、この要件を満たす JRE を指すようにします。 `eclipse.ini` ファイルへのこの変更については、http://wiki.eclipse.org/Eclipse.ini の『*Specifying the JVM*』セクションを参照してください。

手順

1. Eclipse の「ヘルプ」メニューで、新規ソフトウェアをインストールするオプションを選択します (メニュー・ラベルは、使用している Eclipse のバージョンに応じて異なります)。
2. ローカル更新サイトを追加するオプションを選択します。
3. サイトの位置を指定するプロンプトが表示されたら、AppScan Source のインストール・ディレクトリーにナビゲートします。
4. この更新サイトを追加し、Eclipse の再始動を求めるプロンプトが出されるまで、表示されるステップを順番に実行します。
5. インストールが完了した後で、AppScan Source メニューが表示されます。

Eclipse ワークスペース・インポーター: Eclipse または Rational Application Developer for WebSphere Software (RAD) の設定構成

AppScan Source for Analysis のインストールには、デフォルトの Eclipse インポーターが用意されています。 このインポーターは、Eclipse と JRE の位置を識別します。 デフォルトの Eclipse インポーターでワークスペースをインポートできない場合は、新しい Eclipse インポーターの作成が必要になることがあります。

始める前に

各インポーター構成は、Eclipse または Rational Application Developer for WebSphere Software (RAD) のインストール済み環境を表します。これらの構成を使用して既存のワークスペースとプロジェクトを AppScan Source for Analysis にインポートするには、AppScan Source for Development のプラグインも Eclipse 環境にインストールしなければならない場合があります。

RAD ワークスペースを追加する前に、ワークスペース・タイプの構成を作成する必要があります。

手順

1. AppScan Source for Analysis のワークベンチのメインメニューで、「編集」 > 「設定」を選択します。
2. 「**Eclipse** ワークスペース・インポーター」を選択します。
3. 「新しい構成の作成」をクリックし、「新しいインポート構成」ダイアログ・ボックスの以下のフィールドに入力して新しい構成を作成します。
   - 製品: 該当する製品を選択します。

     注**:** ワークスペースの作成に使用した製品を選択できない場合は、 52 ページの『Eclipse または Application Developer の更新』に概要が説明されている構成手順が完了していることを確認してから、ワークスペース・インポーターを作成するようにしてください。
   - 名前: インポーターの名前。
   - 位置: Eclipse インストール済み環境の基本ディレクトリーへのパス。
   - **JRE** の位置: Java ランタイム環境 (JRE) のルート・ディレクトリーへのパス。 `<install_dir>¥JDKS` (`<install_dir>` は AppScan Source インストール済み環境がある場所です) にある JDK、またはその他の優先 JDK を使用します。
4. 「**OK**」をクリックします。
5. インポーターをデフォルトとして特定するには、そのインポーターを選択して「選択した構成をデフォルトにする」をクリックします。これにより、インポーターの「デフォルト」列にアイコンが表示されます。

アプリケーションの新規プロジェクトの作成

アプリケーションを追加した後で、そのアプリケーションにプロジェクトを追加します。 スキャンできるプロジェクト・タイプは、Java/JSP、ASP、C/C++、COBOL、ColdFusion、.NET Assembly、Pattern Based、Perl、PHP、PL/SQL、T-SQL、Visual Basic、および JavaScript です。

このタスクについて

`make` を使用してプロジェクトをコンパイルする場合は、Ounce/Make ユーティリティーを使用してプロジェクト・ファイルを作成してから、そのプロジェクト・ファイルを追加することをお勧めします。 `ant` を使用してプロジェクトをコンパイルする場合は、Ounce/Ant を使用してプロジェクト・ファイルを作成してから、そのプロジェクト・ファイルを追加します。 Ounce/Make およびOunce/Ant については、「*IBM Rational AppScan Source Edition Utilities* ユーザー・ガイド」を参照してください。

注**:** AppScan Source プロジェクトのデフォルトのファイル・エンコードは、**ISO-8859-1** です。 デフォルトのファイル・エンコードは、全般設定ページで変更できます。

注**:** 新規アプリケーション・ウィザードおよび新規プロジェクト・ウィザードを使用してアプリケーションおよびプロジェクトを作成すると、ウィザードで入力された「名前」に従ってファイル名が自動的に割り当てられます (例えば、プロジェクトを作成している場合に「名前」フィールドに **MyProject** と入力すると、プロジェクトのファイル名は `MyProject.ppf` になります)。アプリケーション名およびプロジェクト名は、「プロパティー」ビューを使用して名前変更できます。

手順

1. 「エクスプローラー」ビューで、プロジェクトを追加するアプリケーションを選択します (アプリケーションをまだ追加していない場合は、 39 ページの『アプリケーションの構成』を参照してください)。
2. 以下のいずれかのアクションを実行して、新規プロジェクト・ウィザードを開きます。
   a. ワークベンチのメインメニューで、「ファイル」 > 「プロジェクトの追加」 > 「新規プロジェクト」を選択します。
   b. 選択したアプリケーションを右クリックし、コンテキスト・メニューから「プロジェクトの追加」 > 「新規プロジェクト」を選択します。
3. 新規プロジェクト・ウィザードに従って設定します。

既存のプロジェクトの追加

AppScan Source for Analysis を使用して以前に作成した AppScan Sourceプロジェクト (`.ppf` ファイル) を AppScan Source アプリケーションに追加できます。 Eclipse プロジェクト・ファイル (`.epf`)、サポートされているビルド統合ツール (Ounce/Maven または Ounce/Ant など) のいずれかで作成されたプロジェクト、あるいは Microsoft Visual C/C++ (`.vcproj` または `.dsp`)、 VB.NET (`.vbproj`)、または C# (`.csproj`) で作成されたプロジェクト・ファイルを追加することもできます。

以下の表は、AppScan Source for Analysis で開いてスキャンできるプロジェクト・ファイル・タイプのリストです。

表 5. 開くことができるプロジェクト・ファイル・タイプ

プロジェクト・ファイル・タイプ	ファイル拡張子
Microsoft Visual Studio (バージョン 6)	.dsp
Microsoft Visual Studio C/C++	.vcproj
Microsoft Visual Studio C#	.csproj
Microsoft Visual Studio Visual Basic	.vbproj
AppScan Source プロジェクト・ファイル	.ppf
Eclipse プロジェクト・ファイル	.epf

既存のプロジェクトを追加する方法については、以下のトピックを参照してください。

- 『ユーザー・インターフェース・アクションによる既存のプロジェクトの追加』
- 56 ページの『ドラッグ・アンド・ドロップによる既存のプロジェクトの追加』

重要: 開発環境内に依存関係を持つ AppScan Source プロジェクト (例えば、IBM MobileFirst Platform プロジェクト) を処理する場合は、必ず、プロジェクトをインポートする前に開発環境でビルドしてください。プロジェクトをインポートした後、その中のファイルを変更した場合は、AppScan Source でスキャンする前に開発環境で再ビルドしてください (そうしないと、ファイルに対する変更は AppScan Source によって無視されます)。

注: 既存の .NET プロジェクトをインポートするときには、スキャン対象の追加アセンブリーを指定できます。 これらのアセンブリーは、プロジェクトの「プロパティー」ビューの「追加アセンブリー」タブで追加します。アセンブリーを追加するときに、ビルドする .NET プロジェクトをビルドしないアセンブリー (サード・パーティー・アセンブリーなど) と組み合わせて、単一のスキャンにまとめることができます。

注: WAR ファイルと EAR ファイルは、「エクスプローラー」ビューにドラッグ・アンド・ドロップしても追加できます。ただし、これらはプロジェクトではなくアプリケーションとして追加されます。 詳しくは、 46 ページの『ドラッグ・アンド・ドロップによる既存のアプリケーションの追加』を参照してください。

ユーザー・インターフェース・アクションによる既存のプロジェクトの追加

手順

1. 「エクスプローラー」ビューで、プロジェクトを追加するアプリケーションを選択します (アプリケーションをまだ追加していない場合は、 39 ページの『アプリケーションの構成』を参照してください)。
2. 以下のアクションのいずれかを実行します。
   - ワークベンチのメインメニューで、「ファイル」 > 「プロジェクトの追加」 > 「既存のプロジェクト」を選択します。

- 選択したアプリケーションを右クリックし、コンテキスト・メニューから「プロジェクトの追加」 > 「既存のプロジェクト」を選択します。

3. プロジェクト・ファイルを参照して、アプリケーションに追加します。

   重要: 開発環境内に依存関係を持つ AppScan Source プロジェクト (例えば、IBM MobileFirst Platform プロジェクト) を処理する場合は、必ず、プロジェクトをインポートする前に開発環境でビルドしてください。プロジェクトをインポートした後、その中のファイルを変更した場合は、AppScan Source でスキャンする前に開発環境で再ビルドしてください (そうしないと、ファイルに対する変更は AppScan Source によって無視されます)。

ドラッグ・アンド・ドロップによる既存のプロジェクトの追加

手順

1. ワークステーションでスキャンの対象として追加するプロジェクト (`.ppf`、`.vcproj`、`.dsp`、`.vbproj`、または `.csproj`) を見つけます。

   注: サポートされているビルド統合ツール (Ounce/Maven または Ounce/Ant など) のいずれかによって作成されたファイルはドラッグ・アンド・ドロップできません。

2. プロジェクトを選択して、「AppScan Source for Analysis エクスプローラー」ビューにドラッグします。
3. 以下のステップのいずれかを実行します。
   a. 選択した項目を既存のアプリケーションにドロップします。
   b. 選択した項目を「すべてのアプリケーション」ノードまたはその下にドロップします。 プロジェクトはアプリケーションに含まれていなければなりませんが、このアクションを実行してもプロジェクトは既存のアプリケーションに追加されません。そのため、プロジェクト用の新規アプリケーションを作成するように求めるプロンプトが、新規アプリケーション・ウィザードに表示されます。 アプリケーションの「名前」を入力し、アプリケーションの保存先の「作業ディレクトリー」を参照します。 「終了」をクリックして、新規アプリケーションを作成します (「エクスプローラー」ビューでは、追加したプロジェクトがこのアプリケーション内に含まれます)。

   重要: 開発環境内に依存関係を持つ AppScan Source プロジェクト (例えば、IBM MobileFirst Platform プロジェクト) を処理する場合は、必ず、プロジェクトをインポートする前に開発環境でビルドしてください。プロジェクトをインポートした後、その中のファイルを変更した場合は、AppScan Source でスキャンする前に開発環境で再ビルドしてください (そうしないと、ファイルに対する変更は AppScan Source によって無視されます)。

複数のプロジェクトの追加

アプリケーションに複数のプロジェクトを追加する場合は、プロジェクトを「エクスプローラー」ビューにドラッグ・アンド・ドロップできます。または、プロジェクトのディレクトリーを参照して、一部またはすべてのプロジェクトを現在のアプリケーションにインポートすることもできます。

複数のプロジェクトを追加する方法については、以下のトピックを参照してください。

- 『ユーザー・インターフェース・アクションによる複数のプロジェクトの追加』
- 58 ページの『ドラッグ・アンド・ドロップによる複数のプロジェクトの追加』

重要**:** 開発環境内に依存関係を持つ AppScan Source プロジェクト (例えば、IBM MobileFirst Platform プロジェクト) を処理する場合は、必ず、プロジェクトをインポートする前に開発環境でビルドしてください。プロジェクトをインポートした後、その中のファイルを変更した場合は、AppScan Source でスキャンする前に開発環境で再ビルドしてください (そうしないと、ファイルに対する変更は AppScan Source によって無視されます)。

ユーザー・インターフェース・アクションによる複数のプロジェクトの追加

ディレクトリー (サブディレクトリーを含む)、Eclipse ワークスペース、Rational Application Developer for WebSphere Software (RAD) ワークスペース、または Microsoft ソリューション・ファイルから複数のプロジェクトをアプリケーションに追加できます。

手順

1. 「エクスプローラー」ビューで、プロジェクトを追加するアプリケーションを選択します (アプリケーションをまだ追加していない場合は、 39 ページの『アプリケーションの構成』を参照してください)。
2. 以下のアクションのいずれかを実行します。
   - ワークベンチのメインメニューで、「ファイル」 > 「プロジェクトの追加」 > 「複数のプロジェクト」を選択します。
   - 選択したアプリケーションを右クリックし、コンテキスト・メニューから「プロジェクトの追加」 > 「複数のプロジェクト」を選択します。
3. 「複数のプロジェクトの追加」ダイアログ・ボックスで、以下のいずれかのアクションを実行します。
   - 「ディレクトリーからインポート」を選択して、追加するプロジェクトが格納されているルート・ディレクトリーを参照します。サブディレクトリー内を検索するには、「サブディレクトリーを再帰処理する」チェック・ボックスを選択します。
   - 「**Eclipse** ベースのワークスペースからのインポート」を選択します。「ワークスペース・タイプ」を選択し、ワークスペースを参照します。ワークスペースのディレクトリーを選択して、「**OK**」をクリックします。
   - 「**Microsoft** ソリューション・ファイルからインポート」を選択します。ファイルを参照して選択します。その後、「**OK**」をクリックします。
4. 以下のアクションのいずれかを実行します。
   - 「終了」をクリックして、プロジェクトをアプリケーションに追加します。
   - 「次へ」をクリックして検索結果を表示し、追加するプロジェクトを選択します。次に「終了」をクリックします。

   重要**:** 開発環境内に依存関係を持つ AppScan Source プロジェクト (例えば、IBM MobileFirst Platform プロジェクト) を処理する場合は、必ず、プロジェクトをインポートする前に開発環境でビルドしてください。プロジェクトをイン

ポートした後、その中のファイルを変更した場合は、AppScan Source でスキャンする前に開発環境で再ビルドしてください (そうしないと、ファイルに対する変更は AppScan Source によって無視されます)。

ドラッグ・アンド・ドロップによる複数のプロジェクトの追加

手順

1. ワークステーションでスキャンの対象として追加するプロジェクト (`.ppf`、`.vcproj`、`.dsp`、`.vbproj`、または `.csproj`) を見つけます。

   注: サポートされているビルド統合ツール (Ounce/Maven または Ounce/Ant など) のいずれかによって作成されたファイルはドラッグ・アンド・ドロップできません。
2. プロジェクトを選択または複数選択して、「エクスプローラー」ビューにドラッグします。
3. 選択した項目を既存のアプリケーションにドロップします。

   注: 選択した項目を「すべてのアプリケーション」ノードまたはその下にドロップすることもできますが、そのような操作はお勧めしません。 その代わりに、複数のプロジェクトを既存のアプリケーションにドロップするか、または新規アプリケーションが必要な場合はプロジェクトを個別にドロップすることをお勧めします。

   プロジェクトはアプリケーションに含まれていなければなりませんが、プロジェクトを「すべてのアプリケーション」ノードまたはその下にドロップしてもプロジェクトは既存のアプリケーションに追加されません。そのため、ビューに追加する各プロジェクト用の新規アプリケーションを作成するように求めるプロンプトが、新規アプリケーション・ウィザードに表示されます。

   まだ存在していない新規アプリケーションに複数のプロジェクトを追加する場合は、先にアプリケーションを作成してから、選択したプロジェクトをそのアプリケーションにドラッグ・アンド・ドロップします。

   重要: 開発環境内に依存関係を持つ AppScan Source プロジェクト (例えば、IBM MobileFirst Platform プロジェクト) を処理する場合は、必ず、プロジェクトをインポートする前に開発環境でビルドしてください。プロジェクトをインポートした後、その中のファイルを変更した場合は、AppScan Source でスキャンする前に開発環境で再ビルドしてください (そうしないと、ファイルに対する変更は AppScan Source によって無視されます)。

新規 Arxan プロジェクトの追加

プロジェクト構成ウィザードを使用すると、Arxan プロジェクトを手動で作成し、そのプロジェクトをアプリケーションに追加することができます。

このタスクについて

このトピックの手順では、新規プロジェクト・ウィザード (プロジェクトをアプリケーション内に作成する場合は、新規アプリケーション・ウィザード) のすべてのページで設定を完了するよう指示しています。ウィザードで行った設定は、選択し

たプロジェクトの「プロパティー」ビューでプロジェクトを作成した後に変更できます。

手順

1. 「エクスプローラー」ビューで、プロジェクトを追加するアプリケーションを選択します (アプリケーションをまだ追加していない場合は、 39 ページの『アプリケーションの構成』を参照してください)。
2. 以下のいずれかのアクションを実行して、新規プロジェクト・ウィザードを開きます。
   a. ワークベンチのメインメニューで、「ファイル」 > 「プロジェクトの追加」 > 「新規プロジェクト」を選択します。
   b. 選択したアプリケーションを右クリックし、コンテキスト・メニューから「プロジェクトの追加」 > 「新規プロジェクト」を選択します。
3. ウィザードの「プロジェクト・タイプの選択」ページで、プロジェクト・タイプとして「**Arxan Android**」または「**Arxan iOS**」を選択し、「次へ」をクリックして次のウィザード・ページに進みます。
4. 「プロジェクト・ソース」ウィザード・ページで、以下のようにします。
   a. プロジェクト・ソースを指定します。プロジェクト・ソースは、プロジェクト・ファイルが格納されているディレクトリー、およびプロジェクトに含める追加の個別ファイルから成り立っています。

      プロジェクトに名前を付けて、作業ディレクトリーを指定します。「作業ディレクトリー」は、AppScan Source プロジェクト・ファイル (`.ppf`) が置かれる場所です。これは、すべての相対パスの基準にもなります。
   b. 「ソース・ルートの追加」をクリックして、ソース・コード・ルートを指定し、スキャンに含めるまたはスキャンから除外するディレクトリーまたはファイルを指定します。 ソース・ルートを追加した後で、特定のディレクトリーまたはファイルを除外できます。 このためには、ソース・ルートでディレクトリーまたはファイルを選択 (するか、これらの項目を複数選択) し、選択項目を右クリックして、メニューから「除外」を選択します。ファイルを含めるかまたは除外すると、ファイル名の左側にあるアイコンが変更されます。
5. 「完了」をクリックします。

新規 ASP プロジェクトの追加

プロジェクト構成ウィザードを使用すると、ASP プロジェクトを手動で作成し、そのプロジェクトをアプリケーションに追加することができます。

このタスクについて

注**:** このプロジェクト・タイプは Windows でのみサポートされています。

このトピックの手順では、新規プロジェクト・ウィザード (プロジェクトをアプリケーション内に作成する場合は、新規アプリケーション・ウィザード) のすべてのページで設定を完了するよう指示しています。ただし、ウィザードの一部のページはオプションです (「終了」ボタンがアクティブ化されたら、必須の設定は完了しています)。ウィザードで行った設定は、選択したプロジェクトの「プロパティー」

ビューでプロジェクトを作成した後に変更できます。オプションのページを完了せずに新規プロジェクト・ウィザードを完了した場合は、「プロパティー」ビューで、該当ページの設定を後で変更できます。

注**:** PHP、VB6、および Classic ASP では、 ISO-8859-1 (西ヨーロッパ)、UTF-8、および UTF-16 の文字セットのみがサポートされます。

手順

1. 「エクスプローラー」ビューで、プロジェクトを追加するアプリケーションを選択します (アプリケーションをまだ追加していない場合は、 39 ページの『アプリケーションの構成』を参照してください)。
2. 以下のいずれかのアクションを実行して、新規プロジェクト・ウィザードを開きます。
   a. ワークベンチのメインメニューで、「ファイル」 > 「プロジェクトの追加」 > 「新規プロジェクト」を選択します。
   b. 選択したアプリケーションを右クリックし、コンテキスト・メニューから「プロジェクトの追加」 > 「新規プロジェクト」を選択します。
3. ウィザードの「プロジェクト・タイプの選択」ページで、プロジェクト・タイプとして「**ASP**」を選択し、「次へ」をクリックして次のウィザード・ページに進みます。
4. 「プロジェクト・ソース」ウィザード・ページで、以下のようにします。
   a. プロジェクト・ソースを指定します。プロジェクト・ソースは、プロジェクト・ファイルが格納されているディレクトリー、およびプロジェクトに含める追加の個別ファイルから成り立っています。

      プロジェクトに名前を付けて、作業ディレクトリーを指定します。「作業ディレクトリー」は、AppScan Source プロジェクト・ファイル (`.ppf`) が置かれる場所です。これは、すべての相対パスの基準にもなります。
   b. 「ソース・ルートの追加」をクリックして、ソース・コード・ルートを指定し、スキャンに含めるまたはスキャンから除外するディレクトリーまたはファイルを指定します。 ソース・ルートを追加した後で、特定のディレクトリーまたはファイルを除外できます。 このためには、ソース・ルートでディレクトリーまたはファイルを選択 (するか、これらの項目を複数選択) し、選択項目を右クリックして、メニューから「除外」を選択します。ファイルを含めるかまたは除外すると、ファイル名の左側にあるアイコンが変更されます。
5. 「次へ」をクリックして、次のウィザード・ページに進みます。
6. 「ASP プロジェクト構成」ページで、以下のようにします。
   a. ASP コンテンツ・ルートおよびデフォルト言語を指定して、ASP プロジェクトを構成します。

      ASP コンテンツ・ルート: メイン Web URL またはドメイン URL に対応するディレクトリー

      デフォルト言語: VB スクリプト (デフォルト) または JavaScript
   b. ASP プロジェクトがコンパイル時に依存するタイプ・ライブラリー (`dll`、`exe`、`ocx`、または `tlb`) を追加、削除、または移動します。

7. 「完了」をクリックします。

新規 C/C++ プロジェクトの追加

このタスクについて

アプリケーションに新規 C/C++ プロジェクトを追加するときには、スキャン対象のソース・ファイルのコレクションを指定します。

- include パス
- プリプロセッサー定義
- オプション

このトピックの手順では、新規プロジェクト・ウィザード (プロジェクトをアプリケーション内に作成する場合は、新規アプリケーション・ウィザード) のすべてのページで設定を完了するよう指示しています。ただし、ウィザードの一部のページはオプションです (「終了」ボタンがアクティブ化されたら、必須の設定は完了しています)。ウィザードで行った設定は、選択したプロジェクトの「プロパティー」ビューでプロジェクトを作成した後に変更できます。オプションのページを完了せずに新規プロジェクト・ウィザードを完了した場合は、「プロパティー」ビューで、該当ページの設定を後で変更できます。

重要: C++ プロジェクトをスキャンするには、プロジェクトがエラーなしでコンパイルおよびリンクされている必要があります。

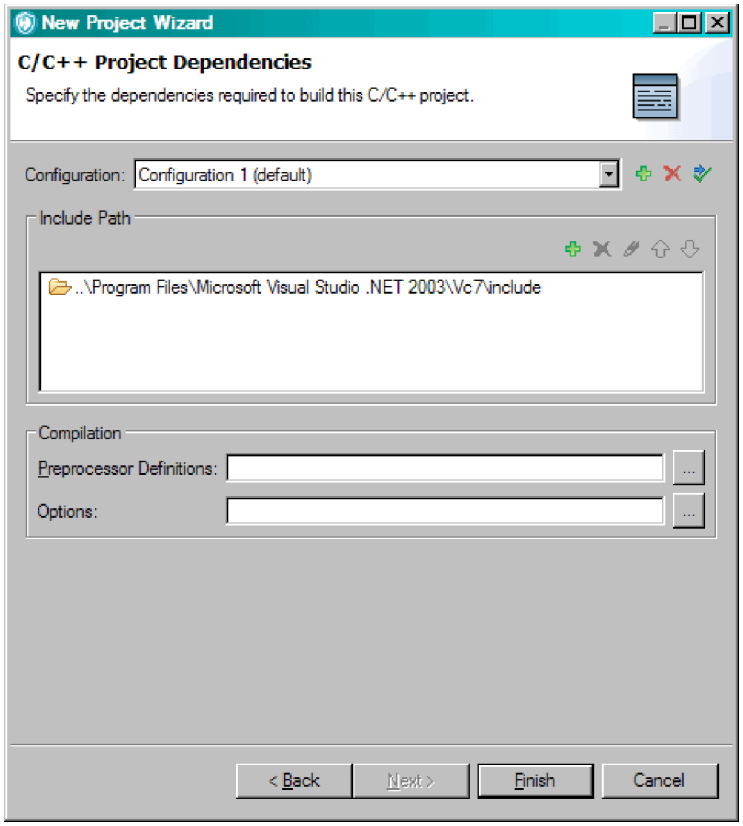
手順

1. 「エクスプローラー」ビューで、プロジェクトを追加するアプリケーションを選択します (アプリケーションをまだ追加していない場合は、39 ページの『アプリケーションの構成』を参照してください)。
2. 以下のいずれかのアクションを実行して、新規プロジェクト・ウィザードを開きます。
   a. ワークベンチのメインメニューで、「ファイル」 > 「プロジェクトの追加」 > 「新規プロジェクト」を選択します。
   b. 選択したアプリケーションを右クリックし、コンテキスト・メニューから「プロジェクトの追加」 > 「新規プロジェクト」を選択します。
3. ウィザードの「プロジェクト・タイプの選択」ページで、プロジェクト・タイプとして「**C/C++**」を選択し、「次へ」をクリックして次のウィザード・ページに進みます。
4. 「プロジェクト・ソース」ウィザード・ページで、以下のようにします。
   a. プロジェクト・ソースを指定します。プロジェクト・ソースは、プロジェクト・ファイルが格納されているディレクトリー、およびプロジェクトに含める追加の個別ファイルから成り立っています。

      プロジェクトに名前を付けて、作業ディレクトリーを指定します。「作業ディレクトリー」は、AppScan Source プロジェクト・ファイル (.ppf) が置かれる場所です。これは、すべての相対パスの基準にもなります。
   b. 「ソース・ルートの追加」をクリックして、ソース・コード・ルートを指定し、スキャンに含めるまたはスキャンから除外するディレクトリーまたはフ

ファイルを指定します。 ソース・ルートを追加した後で、特定のディレクトリーまたはファイルを除外できます。 このためには、ソース・ルートでディレクトリーまたはファイルを選択 (するか、これらの項目を複数選択) し、選択項目を右クリックして、メニューから「除外」を選択します。ファイルを含めるかまたは除外すると、ファイル名の左側にあるアイコンが変更されます。

5. 「次へ」をクリックして、次のウィザード・ページに進みます。
6. 「C/C++ プロジェクト依存関係」ページで、プロジェクト構成および `include` パスを指定して、プロジェクト依存関係を追加します。



- 構成: プロジェクトのすべての使用可能な構成のリスト。 新規の構成を追加するか、既存の構成を削除します。 各構成の残りの設定をすべて定義します。

  C/C++ プロジェクトの複数の構成 (`Debug` や `Release` など) を定義できます。 `Configuration 1` がデフォルトのプロジェクト構成名です。
- インクルード・パス: このセクションは、プロジェクトに必要な `#include` ファイルが格納されているディレクトリーへの完全修飾パス名を追加するときに使用します。
- プリプロセッサー定義: このフィールドは、プロジェクトに定義されているプリプロセッシング・シンボルを追加するときに使用します。プリプロセッサー定義は C/C++ コードに固有です。 プリプロセッサー定義を指定するときは、コンパイラーの `-D` オプションを含めないでください (例えば `-Da=definition1` の代わりに `a=definition1` を使用してください)。複数の定義を指定するときは、セミコロンで区切ったリストを使用します。

- オプション: プロジェクト構成に追加で必要なコンパイラー・パラメーター。

7. 「完了」をクリックします。

新規 COBOL プロジェクトの追加

プロジェクト構成ウィザードを使用すれば、COBOL プロジェクトを手動で作成し、そのプロジェクトをアプリケーションに追加することができます。

このタスクについて

このトピックの手順では、新規プロジェクト・ウィザード (プロジェクトをアプリケーション内に作成する場合は、新規アプリケーション・ウィザード) のすべてのページで設定を完了するよう指示しています。ウィザードで行った設定は、選択したプロジェクトの「プロパティー」ビューでプロジェクトを作成した後に変更できます。

手順

1. 「エクスプローラー」ビューで、プロジェクトを追加するアプリケーションを選択します (アプリケーションをまだ追加していない場合は、39 ページの『アプリケーションの構成』を参照してください)。
2. 以下のいずれかのアクションを実行して、新規プロジェクト・ウィザードを開きます。
   a. ワークベンチのメインメニューで、「ファイル」 > 「プロジェクトの追加」 > 「新規プロジェクト」を選択します。
   b. 選択したアプリケーションを右クリックし、コンテキスト・メニューから「プロジェクトの追加」 > 「新規プロジェクト」を選択します。
3. ウィザードの「プロジェクト・タイプの選択」ページで、プロジェクト・タイプとして「**COBOL**」を選択し、「次へ」をクリックして次のウィザード・ページに進みます。
4. 「プロジェクト・ソース」ウィザード・ページで、以下のようにします。
   a. プロジェクト・ソースを指定します。プロジェクト・ソースは、プロジェクト・ファイルが格納されているディレクトリー、およびプロジェクトに含める追加の個別ファイルから成り立っています。

      プロジェクトに名前を付けて、作業ディレクトリーを指定します。「作業ディレクトリー」は、AppScan Source プロジェクト・ファイル (`.ppf`) が置かれる場所です。これは、すべての相対パスの基準にもなります。
   b. 「ソース・ルートの追加」をクリックして、ソース・コード・ルートを指定し、スキャンに含めるまたはスキャンから除外するディレクトリーまたはファイルを指定します。 ソース・ルートを追加した後で、特定のディレクトリーまたはファイルを除外できます。 このためには、ソース・ルートでディレクトリーまたはファイルを選択 (するか、これらの項目を複数選択) し、選択項目を右クリックして、メニューから「除外」を選択します。ファイルを含めるかまたは除外すると、ファイル名の左側にあるアイコンが変更されます。
5. 「完了」をクリックします。

新規 ColdFusion プロジェクトの追加

プロジェクト構成ウィザードを使用すれば、ColdFusion プロジェクトを手動で作成し、そのプロジェクトをアプリケーションに追加することができます。

このタスクについて

このトピックの手順では、新規プロジェクト・ウィザード (プロジェクトをアプリケーション内に作成する場合は、新規アプリケーション・ウィザード) のすべてのページで設定を完了するよう指示しています。ウィザードで行った設定は、選択したプロジェクトの「プロパティー」ビューでプロジェクトを作成した後に変更できます。

手順

1. 「エクスプローラー」ビューで、プロジェクトを追加するアプリケーションを選択します (アプリケーションをまだ追加していない場合は、 39 ページの『アプリケーションの構成』を参照してください)。
2. 以下のいずれかのアクションを実行して、新規プロジェクト・ウィザードを開きます。
   a. ワークベンチのメインメニューで、「ファイル」 > 「プロジェクトの追加」 > 「新規プロジェクト」を選択します。
   b. 選択したアプリケーションを右クリックし、コンテキスト・メニューから「プロジェクトの追加」 > 「新規プロジェクト」を選択します。
3. ウィザードの「プロジェクト・タイプの選択」ページで、プロジェクト・タイプとして「**ColdFusion**」を選択し、「次へ」をクリックして次のウィザード・ページに進みます。
4. 「プロジェクト・ソース」ウィザード・ページで、以下のようにします。
   a. プロジェクト・ソースを指定します。プロジェクト・ソースは、プロジェクト・ファイルが格納されているディレクトリー、およびプロジェクトに含める追加の個別ファイルから成り立っています。

      プロジェクトに名前を付けて、作業ディレクトリーを指定します。「作業ディレクトリー」は、AppScan Source プロジェクト・ファイル (`.ppf`) が置かれる場所です。これは、すべての相対パスの基準にもなります。
   b. 「ソース・ルートの追加」をクリックして、ソース・コード・ルートを指定し、スキャンに含めるまたはスキャンから除外するディレクトリーまたはファイルを指定します。 ソース・ルートを追加した後で、特定のディレクトリーまたはファイルを除外できます。 このためには、ソース・ルートでディレクトリーまたはファイルを選択 (するか、これらの項目を複数選択) し、選択項目を右クリックして、メニューから「除外」を選択します。ファイルを含めるかまたは除外すると、ファイル名の左側にあるアイコンが変更されます。
5. 「完了」をクリックします。

新規 Java または JavaServer Page (JSP) プロジェクトの追加

アプリケーションに新規 Java プロジェクトを追加するときには、プロジェクト名を指定し、作業ディレクトリーを参照し、ソース・ルートおよびプロジェクト依存関係を指定します。

このタスクについて

このトピックの手順では、新規プロジェクト・ウィザード (プロジェクトをアプリケーション内に作成する場合は、新規アプリケーション・ウィザード) のすべてのページで設定を完了するよう指示しています。ただし、ウィザードの一部のページはオプションです (「終了」ボタンがアクティブ化されたら、必須の設定は完了しています)。ウィザードで行った設定は、選択したプロジェクトの「プロパティー」ビューでプロジェクトを作成した後に変更できます。オプションのページを完了せずに新規プロジェクト・ウィザードを完了した場合は、「プロパティー」ビューで、該当ページの設定を後で変更できます。

手順

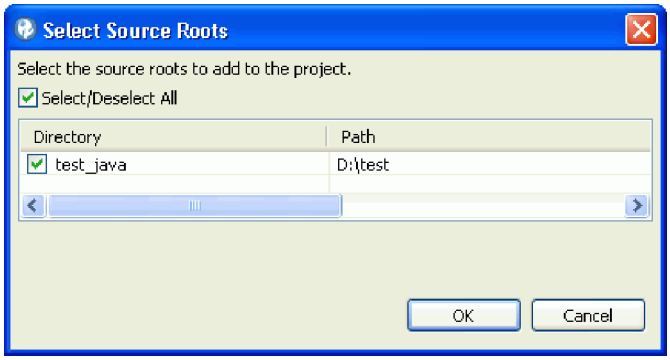
1. 「エクスプローラー」ビューで、プロジェクトを追加するアプリケーションを選択します (アプリケーションをまだ追加していない場合は、39 ページの『アプリケーションの構成』を参照してください)。
2. 以下のいずれかのアクションを実行して、新規プロジェクト・ウィザードを開きます。
   a. ワークベンチのメインメニューで、「ファイル」 > 「プロジェクトの追加」 > 「新規プロジェクト」を選択します。
   b. 選択したアプリケーションを右クリックし、コンテキスト・メニューから「プロジェクトの追加」 > 「新規プロジェクト」を選択します。
3. ウィザードの「プロジェクト・タイプの選択」ページで、プロジェクト・タイプとして「**Java/JSP**」を選択し、「次へ」をクリックして次のウィザード・ページに進みます。
4. 「プロジェクト・ソース」ウィザード・ページで、以下のようにします。
   a. プロジェクト・ソースを指定します。プロジェクト・ソースは、プロジェクト・ファイルが格納されているディレクトリー、およびプロジェクトに含める追加の個別ファイルから成り立っています。

      プロジェクトに名前を付けて、作業ディレクトリーを指定します。「作業ディレクトリー」は、AppScan Source プロジェクト・ファイル (`.ppf`) の格納場所であり、すべての相対パスの基準になります。
   b. ソース・ルートを手動で追加するか、AppScan Source for Analysis がすべての有効なソース・ルートを自動的に検出できるように設定します。

      重要**:**

      - Java クラス・ファイルを分析するには、`javac` を使用してクラス・ファイルをコンパイルするときに、`-g` オプションを指定する必要があります。 AppScan Source 分析は、このオプションによって生成されたデバッグ情報に依存します。

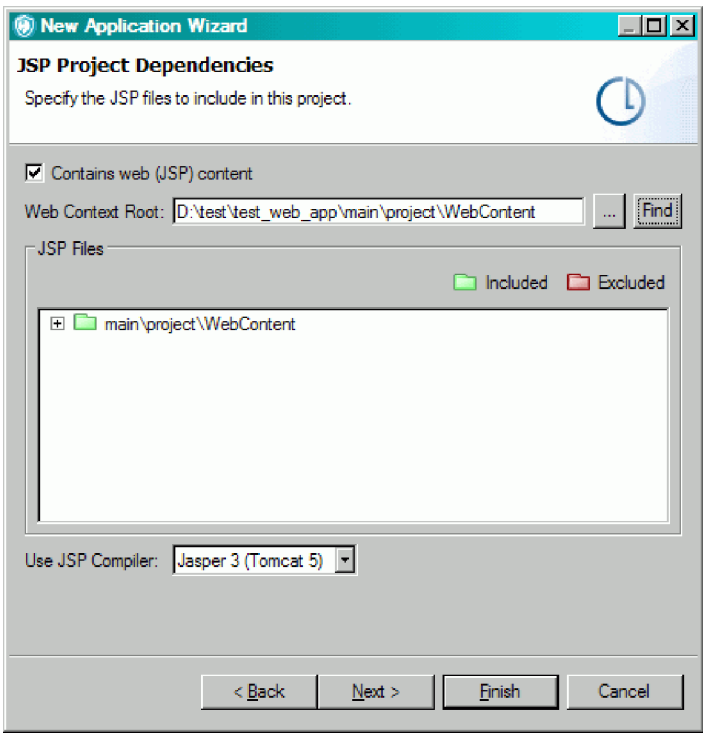
- 各国語文字を含んだ Java ソース・ファイルがプロジェクトに含まれている場合、ネイティブ・ロケール (例えば UTF-8) 以外のロケールで稼働していると、スキャンが失敗し、コンソールにエラーや警告が表示されます。
- ソース・ルートを自動的に検出するには、以下のようにします。
  1) 「ソース・ルートの検出」をクリックし、ソース・コードのルート・ディレクトリーを参照します。
  2) 検出されたすべてのソース・ルートのリストから、プロジェクトに追加するソース・ルートを選択します。



  3) 「**OK**」をクリックします。 スキャンに含めるソースが「プロジェクト・ソース」ダイアログ・ボックス内に表示されます。
- ソース・ルートを手動で検出するには、以下のようにします。
  1) 「ソース・ルートの追加」をクリックします。
  2) ソース・コードのルート・ディレクトリーまたはファイルを選択します。
  3) 「**OK**」をクリックします。 ソース・ルートを追加した後で、特定のディレクトリーまたはファイルを除外できます。 これを行うには、ディレクトリーまたはファイルを選択し (またはこれらの項目を複数選択し)、選択項目を右クリックし、メニューから「除外」を選択します。 ファイルを含めるかまたは除外すると、ファイル名の左側にあるアイコンが変更されます。

プロジェクト依存関係を設定しないでプロジェクトを追加するには、「終了」をクリックします。プロジェクト依存関係を指定するには、「次へ」をクリックします。

5. 「JSP プロジェクト依存関係」ページで、以下のようにします。
   a. JavaServer Page (JSP) プロジェクト依存関係の指定: JavaServer Pages を含む Java プロジェクトの場合、JSP プロジェクト依存関係を指定します。このプロジェクトが JavaServer Pages を含む Web アプリケーションである場合は、「**Web (JSP)** コンテンツを含む」チェック・ボックスを選択します。



b. 「**Web** コンテキスト・ルート」を手動で選択するか、「検索」をクリックして検索します。 「**Web** コンテキスト・ルート」は、1 つの WAR ファイルであるか、 WEB-INF ディレクトリーを含むディレクトリーです。 Web コンテキスト・ルートは、有効な Web アプリケーションのルートでなければなりません。

c. プロジェクトの「**JSP** コンパイラー」を選択します。 製品に付属の Tomcat 7 が、デフォルトの JSP コンパイラー設定です (デフォルト JSP コンパイラーは「Java および JSP」設定ページで変更できます)。 AppScan Source にサポートされるコンパイラーについて詳しくは、http://www.ibm.com/support/docview.wss?uid=swg27027486 を参照してください。

   Apache Tomcat バージョン 7 および 8 は、AppScan Source のインストール済み環境に含まれています。「**Tomcat 7**」および「**Tomcat 8**」設定ページが未構成の場合、AppScan Source は、提供されている Tomcat JSP コンパイラー (現在デフォルトとしてマーク) を使用して JSP ファイルをコンパイルします。外部でサポートされている Tomcat コンパイラーを使用したい場合は、Tomcat 設定ページを使用して、ローカルの Tomcat インストール済み環境を示します。

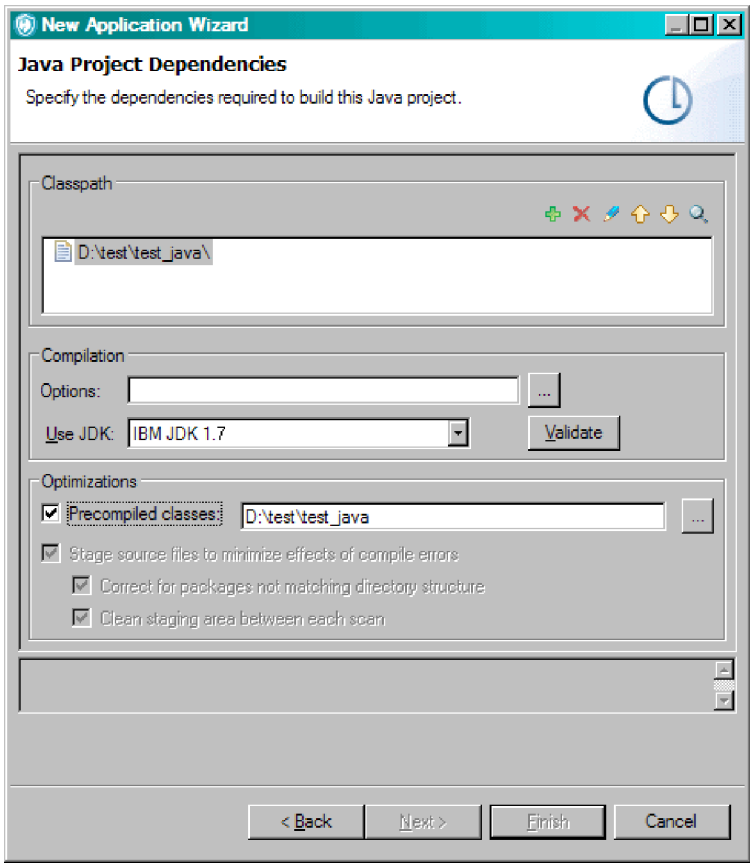
   Oracle WebLogic サーバー または WebSphere Application Server を使用する場合は、分析時にアプリケーション・サーバーを JSP コンパイルに使用できるようにするため、適切な設定ページを構成して、アプリケーション・サーバーのローカルのインストール済み環境を示す必要があります。この構成をまだ完了していない場合は、JSP コンパイラーを選択する際に構成を行うようにメッセージによって指示されます。メッセージ内の「はい」をクリックすると、該当する設定ページに進みます。「いいえ」をクリックす

ると、JSP コンパイラーの選択項目の隣に警告リンクが表示されます (リンクを選択すると、設定ページが開きます)。

JSP プロジェクト依存関係を設定してプロジェクトを追加するには、「終了」をクリックします。Java プロジェクト依存関係を指定するには、「次へ」をクリックします。

6. 「Java プロジェクト依存関係」ページで、この Java プロジェクトをビルドするために必要な依存関係を以下のようにして指定します。
   a. JAR ファイルを手動で追加するか、「検索」をクリックして、依存している JAR ファイルおよびクラス・ファイルが格納されているディレクトリーを AppScan Source for Analysis に検索させます。

      「クラスパス」リストに、プロジェクトへの相対パスが表示されます。クラスパスでは、必要な JAR ファイル、およびプロジェクトが必要とするクラス・ファイルが格納されているディレクトリーを指定する必要があります。



- 追加、削除、上へ移動、下へ移動: クラスパスのファイルを追加または削除するか、ファイルの順序を上または下へ移動します。
- 検索: プロジェクト内のソース・ファイルに基づいて、 JAR 項目およびクラスパス項目を検索します。

重要: Java プロジェクトに JavaServer Pages が含まれる場合は、JSP プロジェクト依存関係も追加する必要があります。

- プロジェクト依存関係を手動で検出するには、以下のようにします。

1) 「クラスパス」セクションのツールバーで「追加」をクリックし、Java プロジェクトをコンパイルするために必要な JAR ファイルおよびクラス・ファイルのディレクトリーを選択します。
2) 「**OK**」をクリックします。 JAR ファイルおよびディレクトリーがクラスパス内に表示されます。 必要に応じて順序を変更します。

- 依存関係を自動的に検出するには、以下のようにします。
  1) 「クラスパス」セクションのツールバーで「検索」をクリックします。
  2) Java プロジェクトをコンパイルするために必要な JAR ファイルおよびクラス・ファイルの検索先ディレクトリーを指定します。
  3) ソースに基づいて、指定の検索パスを使用して、必要なプロジェクト依存関係を AppScan Source for Analysis で検索する場合は、「ソースおよび **JAR** ファイル内を検索する」チェック・ボックスを選択します。
  4) 「次へ」をクリックして、プロジェクト依存関係を検索させ、競合を識別させます。
- 競合を解決するには、以下のようにします。
  1) 競合が存在する場合は、「競合の解決」ダイアログ・ボックスで、解決する項目を選択し、「解決」 をクリック (するか、「次へ」をクリックして競合を自動的に解決) します。競合は、AppScan Source for Analysis が、依存関係を満たす 1 つのディレクトリー内で複数の JAR またはクラスを検出した場合に発生します。

     未解決の競合の左側に、赤いアイコンが表示されます。競合が解決されると、赤いアイコンが緑に変更され、項目は「解決済み」になります。 競合を「削除」することもできます。
  2) 競合を解決または削除したら、クラスパス項目の検査、再配列、または削除が必要になる場合があります。検出できなかったインポートのリストに注意してください。 未解決のインポートは、AppScan Source for Analysis のスキャンの実行時に、コンパイル・エラーの原因になります。

b. オプション: プロジェクトに対して、追加の必須コンパイラー・パラメーターを指定します。

   コンパイル・オプションは、ソース・ファイルがコンパイルできるようにコンパイラーに渡されるオプションです。例えば、`-source 1.5` は、プロジェクトのソース・レベルを指定します。

c. **JDK** の使用: このコードをスキャンするときに使用する Java Development Kit (JDK) を指定します。 デフォルトでは、「**IBM JDK 1.8**」が使用されます。AppScan Source では、「**IBM JDK 1.7**」も選択できます。追加の JDK を定義する場合、または異なるデフォルトの JDK を設定する場合は、「**Java** および **JSP** の設定」を使用します。

   注**:** 製品に付属の JSP プロジェクトのデフォルト・コンパイラーは、Tomcat 7 です。これには、Java バージョン 1.6 以上が必要です。 **Tomcat 7** をデフォルトのまま使用している場合、古い JDK を選択すると、以下のスキャン中のコンパイル・エラーが発生します。

d. 「検証」アクションを行うと、プロジェクト依存関係が正しく構成されていることが保証されます。Java プロジェクト内でソースとクラスパスの間の構成競合をチェックすると共に、コンパイル・エラーもチェックします。クラスパス内のクラスがソース・ルート内のクラスと重複している場合、競合が存在します。

   競合が存在する場合、検証テキスト領域には、JAR またはクラスパス上でクラスが定義されている場所、およびソース内に重複が存在するかどうかが表示されます。 クラスパスから競合を削除し、チェックを再実行します。

   競合をチェックした後で、「検証」をクリックして、プロジェクトをコンパイルできるかどうか、およびコンパイル・エラーがレポートされるかどうかを判別します。

e. プリコンパイル済みクラス: このフィールドでは、スキャン中にコンパイルするのではなく、プリコンパイル済みの Java または JSP クラス・ファイルを使用できます。

f. コンパイル・エラーの影響を最小化するためにソース・ファイルをステージする: ソース・コードが正しくコンパイルされ、ディレクトリー内に正確に配置され、パッケージと一致する場合は、このチェック・ボックスをクリアします。

g. ディレクトリー構造と一致しないパッケージの修正: パッケージがディレクトリー構造と一致しない場合に選択します。

h. 各スキャン間のステージング領域のクリーンアップ: 最適化オプションです。

7. 「完了」をクリックします。

タスクの結果

ヒント**:** Java をスキャンしており、Java プロジェクトに欠落依存関係がある場合、AppScan Source は、依存関係が提供するはずだった部分を合成することで、トレースを作成します。 この合成には .jar ファイル内の情報が正確に反映されない場合があります。合成を制限することにより検出結果の精度を向上するために、欠落している依存関係を以下のように指定できます。

1. スキャン後に、<data_dir>¥logs¥scanner_exceptions.log (<data_dir> は、ご使用の AppScan Source プログラム・データの場所です。詳しくは、 360 ページの『インストールとユーザー・データ・ファイルの場所』 を参照してください。)を開いて、AppScan Source が欠落依存関係を報告しているかどうかを確認します。
2. 依存関係を組み込むようにプロジェクト・プロパティーを変更します。 そのためには、 93 ページの『アプリケーションおよびプロジェクトのプロパティーの変更』 の指示に従い、「**JSP** プロジェクト依存関係」または「プロジェクト依存関係」タブに依存関係を指定して保存します。
3. プロジェクトを再スキャンします。

注**:** デフォルトでは、AppScan Source は、依存関係の欠落やコンパイル・エラーについて Java ファイルや Java バイトコードをスキャンします。 これらの設定は、以下のように変更できます。

1. テキスト・エディターで <data_dir>¥config¥scan.ozsettings を開きます。

2. コンパイル・エラーの設定を変更するには、ファイル内で compile_java_sources_with_errors を見つけます。この設定は、以下の例のようになります。

```
<Setting
  name="compile_java_sources_with_errors"
  value="true"
  default_value="true"
  type="bool"
  hidden="true"
  display_name="compile_java_sources_with_errors"
  description="Attempt to scan java code with compilation errors."
/>
```

3. 欠落依存関係の設定を変更するには、ファイル内で scan_java_bytecode_without_dependencies を見つけます。この設定は、以下の例のようになります。

```
<Setting
  name="scan_java_bytecode_without_dependencies"
  value="true"
  default_value="true"
  type="bool"
  hidden="true"
  display_name="scan_java_bytecode_without_dependencies"
  description="Scans Java bytecode even when some of
    the dependencies are missing by artificially
    synthesizing the unresolved symbols."
/>
```

4. この設定では、value 属性を変更します。属性が true に設定されている場合、この設定はオンになります。 コンパイル・エラーの設定が false になっている場合、AppScan Source は、スキャン中にコンパイル・エラーのある Java コードをスキップします。 欠落依存関係の設定が false になっている場合、AppScan Source は、欠落依存関係が存在するかどうかについて、Java バイトコードをスキャンしません。
5. この設定の変更後、ファイルを保存してAppScan Source を始動または再始動します。

JSP プロジェクトへのコンテンツの追加

JavaServer Page (JSP) プロジェクトには、JavaServer Pages 上でビルドされた Web アプリケーションが含まれます。

このタスクについて

JSP プロジェクトを正常にスキャンするには、JavaServer Pages が有効な Web アプリケーション構造内にある必要があります。 このセクションでは、正常なスキャンのために必要な Web コンテキスト・ルートの下のファイル構造について説明します。 JSP プロジェクトを構成する前に、Web アプリケーション構造をよく理解する必要があります。

Web アプリケーション・サーバー内にデプロイされる Web アプリケーション (Tomcat など) では、標準ディレクトリー構造が必要です。 デプロイされるアプリケーションは、ディレクトリー構造内に配置された一連のファイル、または 1 つの WAR ファイルです。 1 つの WAR ファイルの場合、ディレクトリー構造は ZIP ファイル内に格納され、web context root がディレクトリー構造のルートです。

Web コンテキスト・ルートの下には、以下の標準ディレクトリーがあります。

表 *6. Web* コンテキスト・ルート・ディレクトリー

<web-context-root>¥	
WEB-INF¥	
classes¥	ディレクトリー (パッケージ) 内に配置された Java クラス・ファイル
lib¥	クラスパスに追加された JAR ファイル
web.xml	web.xml は、アプリケーションで使用可能なリソースを記述します

その他の必要なファイルが存在する場合は、他のディレクトリーに格納されています。 例えば、コンテンツ (JSP および HTML ファイル) 用のディレクトリーやタグ・ライブラリー用のディレクトリーが一般的です。

表 *7.* 他のディレクトリー

<web-context-root>¥	
jsp¥	アプリケーション内の JavaServer Pages が格納されています
WEB-INF¥	
tld¥	アプリケーション内で使用されるタグ・ライブラリーが格納されています

これらの標準 Web アプリケーション・ディレクトリーに加えて、Web アプリケーション・サーバーは、デプロイされるすべての Web アプリケーションによって共有されるクラス・ファイルおよび JAR ファイルを格納する特有のディレクトリーを持つことができます。 例えば、Tomcat 7 は、これらの JAR ファイルを common¥lib または common¥endorsed ディレクトリー内に格納します。これらの非標準ディレクトリーの場所は、各アプリケーション・サーバーに固有です。

重要**:** JavaServer Pages をスキャンする前に、Web コンテキスト・ルート内にすべての必要なファイルが存在することを確認してください。 AppScan Source for Analysis は、Web コンテキスト・ルート内の JavaServer Pages のみをスキャンします。

手順

1. 必要な場合には、Web コンテキスト・ルートの下の適切な場所にファイルをコピーします。
2. Web コンテキスト・ルートを、すべての JavaServer Pages を格納しているディレクトリーまたは 1 つの WAR ファイルとして指定します。
3. クラスパスに JAR ファイルまたはクラス・ファイルのディレクトリーが含まれていることを確認してください。
4. プロジェクト・プロパティーを構成します。

タスクの結果

AppScan Source for Analysis は、WEB-INF¥classes ディレクトリー、および WEB-INF¥lib 内のすべての JAR ファイルをクラスパスに追加します (JSP の場合のみ)。 Web-INF パスには含まれていないが、JSP のコンパイルには必要な項目を追加できます。これらの JAR ファイルは、アプリケーション・サーバーの共通ディレクトリー内に配置されている weblogic.jar またはベンダー JAR ファイルに類似しています。

JSP ソースは、スキャン対象の Web コンテキスト・ルートの下の JavaServer Pages です。 ソース・ファイルは、Web コンテキスト・ルートを基準として認識されます。 JSP ソースを指定するときには、Web コンテキスト・ルート内のファイルのセットのみに対象が制限されます。

JSP プロジェクト・ソースは、プロジェクト・ファイルが格納されているディレクトリー、およびプロジェクトに含める追加の個別ファイルから成り立っています。

- Web コンテキスト・ルート内の JavaServer Pages のサブセットを指定します。これを指定しない場合は、すべてのファイルがスキャンされます。
- JavaServer Pages が Java コードに依存する場合は、これらのソースを指定する必要があります。
- JSP ファイルには、jsp および jspx ファイルが含まれます。

新規 JavaScript プロジェクトの追加

プロジェクト構成ウィザードを使用すれば、JavaScript プロジェクトを手動で作成し、そのプロジェクトをアプリケーションに追加することができます。

このタスクについて

このトピックの手順では、新規プロジェクト・ウィザード (プロジェクトをアプリケーション内に作成する場合は、新規アプリケーション・ウィザード) のすべてのページで設定を完了するよう指示しています。ウィザードで行った設定は、選択したプロジェクトの「プロパティー」ビューでプロジェクトを作成した後に変更できます。

手順

1. 「エクスプローラー」ビューで、プロジェクトを追加するアプリケーションを選択します (アプリケーションをまだ追加していない場合は、 39 ページの『アプリケーションの構成』を参照してください)。
2. 以下のいずれかのアクションを実行して、新規プロジェクト・ウィザードを開きます。
   a. ワークベンチのメインメニューで、「ファイル」 > 「プロジェクトの追加」 > 「新規プロジェクト」を選択します。
   b. 選択したアプリケーションを右クリックし、コンテキスト・メニューから「プロジェクトの追加」 > 「新規プロジェクト」を選択します。
3. ウィザードの「プロジェクト・タイプの選択」ページで、プロジェクト・タイプとして「**JavaScript**」を選択し、「次へ」をクリックして次のウィザード・ページに進みます。

4. 「プロジェクト・ソース」ウィザード・ページで、以下のようにします。
   a. プロジェクト・ソースを指定します。プロジェクト・ソースは、プロジェクト・ファイルが格納されているディレクトリー、およびプロジェクトに含める追加の個別ファイルから成り立っています。

      プロジェクトに名前を付けて、作業ディレクトリーを指定します。「作業ディレクトリー」は、AppScan Source プロジェクト・ファイル (.ppf) が置かれる場所です。これは、すべての相対パスの基準にもなります。
   b. 「ソース・ルートの追加」をクリックして、ソース・コード・ルートを指定し、スキャンに含めるまたはスキャンから除外するディレクトリーまたはファイルを指定します。 ソース・ルートを追加した後で、特定のディレクトリーまたはファイルを除外できます。 このためには、ソース・ルートでディレクトリーまたはファイルを選択 (するか、これらの項目を複数選択) し、選択項目を右クリックして、メニューから「除外」を選択します。ファイルを含めるかまたは除外すると、ファイル名の左側にあるアイコンが変更されます。
5. 「完了」をクリックします。

新規 .NET アセンブリー・プロジェクトの追加

新規プロジェクト・ウィザードを使用すれば、.NET アセンブリー・プロジェクトを作成することができます。 .NET アセンブリー・プロジェクトは、ソース・ファイルを使用できないまたはビルドできない場合に、コンパイル済み .NET アセンブリー・ファイルをスキャンするために使用できます。 .NET アセンブリー・プロジェクトは、作業ディレクトリー、およびソース (ディレクトリーまたは個別のアセンブリー・ファイル) のリストから成り立っています。

このタスクについて

注: このプロジェクト・タイプは Windows でのみサポートされています。

このトピックの手順では、新規プロジェクト・ウィザード (プロジェクトをアプリケーション内に作成する場合は、新規アプリケーション・ウィザード) のすべてのページで設定を完了するよう指示しています。ウィザードで行った設定は、選択したプロジェクトの「プロパティー」ビューでプロジェクトを作成した後に変更できます。

手順

1. 「エクスプローラー」ビューで、プロジェクトを追加するアプリケーションを選択します (アプリケーションをまだ追加していない場合は、 39 ページの『アプリケーションの構成』を参照してください)。
2. 以下のいずれかのアクションを実行して、新規プロジェクト・ウィザードを開きます。
   a. ワークベンチのメインメニューで、「ファイル」 > 「プロジェクトの追加」 > 「新規プロジェクト」を選択します。
   b. 選択したアプリケーションを右クリックし、コンテキスト・メニューから「プロジェクトの追加」 > 「新規プロジェクト」を選択します。

3. ウィザードの「プロジェクト・タイプの選択」ページで、プロジェクト・タイプとして「**.NET** アセンブリー」を選択し、「次へ」をクリックして次のウィザード・ページに進みます。
4. 「プロジェクト・ソース」ウィザード・ページで、以下のようにします。
   a. プロジェクト・ソースを指定します。プロジェクト・ソースは、プロジェクト・ファイルが格納されているディレクトリー、およびプロジェクトに含める追加の個別ファイルから成り立っています。

      プロジェクトに名前を付けて、作業ディレクトリーを指定します。「作業ディレクトリー」は、AppScan Source プロジェクト・ファイル (`.ppf`) が置かれる場所です。これは、すべての相対パスの基準にもなります。
   b. 「ソース・ルートの追加」をクリックして、ソース・コード・ルートを指定し、スキャンに含めるまたはスキャンから除外するディレクトリーまたはファイルを指定します。 ソース・ルートを追加した後で、特定のディレクトリーまたはファイルを除外できます。 このためには、ソース・ルートでディレクトリーまたはファイルを選択 (するか、これらの項目を複数選択) し、選択項目を右クリックして、メニューから「除外」を選択します。ファイルを含めるかまたは除外すると、ファイル名の左側にあるアイコンが変更されます。
5. 「完了」をクリックします。

新規パターン・ベース・プロジェクトの追加

このタスクについて

新規プロジェクト・ウィザードを使用すれば、パターン・ベース・プロジェクトを手動で作成し、そのプロジェクトをアプリケーションに追加することができます。パターン・ベース・プロジェクトには、パターン・ベース分析およびスキャンのための任意の言語非依存ファイルのコレクションが含まれます。

例えば、`.xml` および `.config` ファイルを論理的にグループ化し、これらのファイルを対象に、パターン・ベースの特定の式の有無を検索できます。AppScan Source for Analysis は、ファイルをスキャンして、式の有無を検索します (詳しくは、268 ページの『パターン・ベースのルールによるカスタマイズ』を参照してください)。

このトピックの手順では、新規プロジェクト・ウィザード (プロジェクトをアプリケーション内に作成する場合は、新規アプリケーション・ウィザード) のすべてのページで設定を完了するよう指示しています。ウィザードで行った設定は、選択したプロジェクトの「プロパティー」ビューでプロジェクトを作成した後に変更できます。

手順

1. 「エクスプローラー」ビューで、プロジェクトを追加するアプリケーションを選択します (アプリケーションをまだ追加していない場合は、 39 ページの『アプリケーションの構成』を参照してください)。
2. 以下のいずれかのアクションを実行して、新規プロジェクト・ウィザードを開きます。

a. ワークベンチのメインメニューで、「ファイル」 > 「プロジェクトの追加」 > 「新規プロジェクト」を選択します。
b. 選択したアプリケーションを右クリックし、コンテキスト・メニューから「プロジェクトの追加」 > 「新規プロジェクト」を選択します。

3. ウィザードの「プロジェクト・タイプの選択」ページで、プロジェクト・タイプとして「パターン・ベース」を選択し、「次へ」をクリックして次のウィザード・ページに進みます。
4. 「プロジェクト・ソース」ウィザード・ページで、以下のようにします。
   a. プロジェクト・ソースを指定します。プロジェクト・ソースは、プロジェクト・ファイルが格納されているディレクトリー、およびプロジェクトに含める追加の個別ファイルから成り立っています。

      プロジェクトに名前を付けて、作業ディレクトリーを指定します。「作業ディレクトリー」は、AppScan Source プロジェクト・ファイル (`.ppf`) が置かれる場所です。これは、すべての相対パスの基準にもなります。
   b. 「ソース・ルートの追加」をクリックして、ソース・コード・ルートを指定し、スキャンに含めるまたはスキャンから除外するディレクトリーまたはファイルを指定します。 ソース・ルートを追加した後で、特定のディレクトリーまたはファイルを除外できます。 このためには、ソース・ルートでディレクトリーまたはファイルを選択 (するか、これらの項目を複数選択) し、選択項目を右クリックして、メニューから「除外」を選択します。ファイルを含めるかまたは除外すると、ファイル名の左側にあるアイコンが変更されます。
5. 「完了」をクリックします。

新規 Perl プロジェクトの追加

新規プロジェクト・ウィザードを使用すれば、Perl プロジェクトを手動で作成し、そのプロジェクトをアプリケーションに追加することができます。

手順

1. 「エクスプローラー」ビューで、プロジェクトを追加するアプリケーションを選択します (アプリケーションをまだ追加していない場合は、 39 ページの『アプリケーションの構成』を参照してください)。
2. 以下のいずれかのアクションを実行して、新規プロジェクト・ウィザードを開きます。
   a. ワークベンチのメインメニューで、「ファイル」 > 「プロジェクトの追加」 > 「新規プロジェクト」を選択します。
   b. 選択したアプリケーションを右クリックし、コンテキスト・メニューから「プロジェクトの追加」 > 「新規プロジェクト」を選択します。
3. ウィザードの「プロジェクト・タイプの選択」ページで、プロジェクト・タイプとして「**Perl**」を選択し、「次へ」をクリックして次のウィザード・ページに進みます。
4. 「プロジェクト・ソース」ウィザード・ページで、以下のようにします。

a. プロジェクト・ソースを指定します。プロジェクト・ソースは、プロジェクト・ファイルが格納されているディレクトリー、およびプロジェクトに含める追加の個別ファイルから成り立っています。

   プロジェクトに名前を付けて、作業ディレクトリーを指定します。「作業ディレクトリー」は、AppScan Source プロジェクト・ファイル (.ppf) が置かれる場所です。これは、すべての相対パスの基準にもなります。

b. 「ソース・ルートの追加」をクリックして、ソース・コード・ルートを指定し、スキャンに含めるまたはスキャンから除外するディレクトリーまたはファイルを指定します。 ソース・ルートを追加した後で、特定のディレクトリーまたはファイルを除外できます。 このためには、ソース・ルートでディレクトリーまたはファイルを選択 (するか、これらの項目を複数選択) し、選択項目を右クリックして、メニューから「除外」を選択します。ファイルを含めるかまたは除外すると、ファイル名の左側にあるアイコンが変更されます。

5. 「完了」をクリックします。

PHP プロジェクト構成

アプリケーションに新規 PHP (PHP: Hypertext Preprocessor) プロジェクトを追加するときには、プロジェクト名を指定し、作業ディレクトリーを参照し、ソース・ルートおよびプロジェクト依存関係を指定します。プロジェクト依存関係は、プロジェクトを作成した後で、「プロジェクト・プロパティー」の「プロジェクト依存関係」タブ内でも設定できます。

このタスクについて

このトピックの手順では、新規プロジェクト・ウィザード (プロジェクトをアプリケーション内に作成する場合は、新規アプリケーション・ウィザード) のすべてのページで設定を完了するよう指示しています。ただし、ウィザードの一部のページはオプションです (「終了」ボタンがアクティブ化されたら、必須の設定は完了しています)。ウィザードで行った設定は、選択したプロジェクトの「プロパティー」ビューでプロジェクトを作成した後に変更できます。オプションのページを完了せずに新規プロジェクト・ウィザードを完了した場合は、「プロパティー」ビューで、該当ページの設定を後で変更できます。

注: PHP、VB6、および Classic ASP では、 ISO-8859-1 (西ヨーロッパ)、UTF-8、および UTF-16 の文字セットのみがサポートされます。

手順

1. 「エクスプローラー」ビューで、プロジェクトを追加するアプリケーションを選択します (アプリケーションをまだ追加していない場合は、 39 ページの『アプリケーションの構成』を参照してください)。
2. 以下のいずれかのアクションを実行して、新規プロジェクト・ウィザードを開きます。
   a. ワークベンチのメインメニューで、「ファイル」 > 「プロジェクトの追加」 > 「新規プロジェクト」を選択します。
   b. 選択したアプリケーションを右クリックし、コンテキスト・メニューから「プロジェクトの追加」 > 「新規プロジェクト」を選択します。

3. ウィザードの「プロジェクト・タイプの選択」ページで、プロジェクト・タイプとして「**PHP**」を選択し、「次へ」をクリックして次のウィザード・ページに進みます。
4. 「プロジェクト・ソース」ウィザード・ページで、以下のようにします。
   a. プロジェクト・ソースを指定します。プロジェクト・ソースは、プロジェクト・ファイルが格納されているディレクトリー、およびプロジェクトに含める追加の個別ファイルから成り立っています。

      プロジェクトに名前を付けて、作業ディレクトリーを指定します。「作業ディレクトリー」は、AppScan Source プロジェクト・ファイル (`.ppf`) が置かれる場所です。これは、すべての相対パスの基準にもなります。
   b. 「ソース・ルートの追加」をクリックして、ソース・コード・ルートを指定し、スキャンに含めるまたはスキャンから除外するディレクトリーまたはファイルを指定します。 ソース・ルートを追加した後で、特定のディレクトリーまたはファイルを除外できます。 このためには、ソース・ルートでディレクトリーまたはファイルを選択 (するか、これらの項目を複数選択) し、選択項目を右クリックして、メニューから「除外」を選択します。ファイルを含めるかまたは除外すると、ファイル名の左側にあるアイコンが変更されます。
5. **PHP** プロジェクト構成: 「**PHP** 文書ルート」フィールドで、PHP アプリケーションのルートを表すディレクトリーを入力するか、参照して指定します。 これが、サイトの基本 URL にマップされるファイル・システム・ディレクトリーになります。 PHP 文書ルートを指定しなかった場合は、「プロジェクト・ソース」ページ内で指定されたソース・ルートが使用されます。
6. オプション: 「インクルード・パス」を設定します。インクルード・パス・ディレクトリーは、PHP `include` ステートメント (`include`、`include_once`、`require`、`require_once` など) 内で使用されるファイルへの相対パスを解決するときに使用します。
7. オプション: 「クラス・インクルード・パス」を設定します。クラス・インクルード・パス・ディレクトリーは、PHP クラス定義を含むファイルを検索するときに使用します。
8. 「完了」をクリックします。
9. オプション: 未解決の依存関係の構成:「プロジェクト・プロパティー」で、「プロジェクト依存関係」ページに移動し、 80 ページの『未解決の PHP `include` 式の構成』および 85 ページの『未解決の PHP クラス参照の構成』のステップを実行します。

例: 新規 PHP プロジェクトの作成

このタスクについて

この例は、新規アプリケーション・ウィザードを使用して PHP プロジェクトを作成する方法を示します。

手順

1. 以下のアクションのいずれかを実行します。
   - メインメニュー・バーから「ファイル」 > 「アプリケーションの追加」 > 「新規アプリケーションの作成」を選択します。

- 「エクスプローラー」ビューのツールバーで、「アプリケーション・メニューの追加」下矢印ボタンをクリックして、メニューから「新規アプリケーションの作成」を選択します。
- 「エクスプローラー」ビューで、「すべてのアプリケーション」を右クリックし、メニューから「アプリケーションの追加」 > 「新規アプリケーションの作成」を選択します。

2. アプリケーションの「名前」を入力します。
3. アプリケーションの保存先の「作業ディレクトリー」を参照します。 新規アプリケーションのファイル名の拡張子は `.paf` になります。
4. 「次へ」をクリックして、プロジェクトを構成します。
5. ウィザードの「プロジェクト・タイプの選択」ページで、プロジェクト・タイプとして「**PHP**」を選択し、「次へ」をクリックして次のウィザード・ページに進みます。
6. 「プロジェクト・ソース」ページで、以下のようにします。
   a. 「名前」フィールドで、プロジェクトの名前 (ここでは `MyProject` など) を入力します。
   b. 「作業ディレクトリー」フィールドで、作成されるプロジェクト・ファイルの格納場所 (ここでは `C:¥Apps¥MyProject` など) を参照して指定します。
   c. 「ソース・ルートの追加」をクリックして、スキャン対象とする PHP ファイルが格納されているすべてのディレクトリーを追加します。 例えば、「ファイルまたはディレクトリーの選択」ダイアログ・ボックスで、`C:¥Apps¥MyProject¥root` を参照して指定し、「**OK**」をクリックしてダイアログ・ボックスを閉じます。

   「次へ」をクリックします。
7. 「PHP プロジェクト構成」ページで、以下のようにします。
   a. 「**PHP** 文書ルート」フィールドで、PHP アプリケーションのルートを表すディレクトリーを入力するか、参照します。これが、サイトの基本 URL にマップされるファイル・システム・ディレクトリーになります。 デフォルトでは、このフィールドには、「プロジェクト・ソース」ページ内で指定されたソース・ルートが事前に取り込まれます。
   b. オプション: インクルード・パス・ディレクトリーを追加します。これらのディレクトリーは、PHP インクルード・ステートメント (`include`、`include_once`、`require`、`require_once` など) 内で使用されるファイルへの相対パスを解決するために使用されます。
   c. オプション: クラスパス・ディレクトリーを追加します。 これらのディレクトリーは、PHP クラス定義を含むファイルを検出するために使用されます。
8. 「終了」をクリックします。 これで、PHP プロジェクトをスキャンする準備が整いました。

未解決の PHP include 式の構成

始める前に

「プロジェクト・プロパティー」で、「プロジェクト依存関係」ページに移動します。

手順

1. 「未解決のインクルード式の構成」をクリックして、「未解決のインクルード式の構成」ダイアログ・ボックスを開きます。
2. ダイアログ・ボックスの上部には、未解決の include 式 (解決されなかったか、または解決するために追加の処理を必要としたすべての include 式) が表示されます。式について提供される情報は、以下のとおりです。

オプション	説明
インクルード・テキスト / 更新済みテキスト	この列には、ソース・コード内の式テキストが、そのままの状態で表示されます。 「+」をクリックしてこの列を展開すると、最後のスキャン中に使用された更新済みテキストが表示されます。 最後のスキャン中に使用可能な更新済みテキストがなかった場合は、<empty> が表示されます。 列を展開すると、複数の更新済みテキスト行が表示されることがあります。各テキスト行は、ソース・コード内でこの式が使用されたそれぞれの場所に対応します。
状態	この列には、未解決の式を示す「X」または正常に解決された式を示すチェック・マークが表示されます。

オプション	説明
解決方法	この列は、更新済みテキストが生成された方法を示します。以下の値が含まれます。 • AutoResolver: アプリケーションは、内部ヒューリスティックを使用して、ファイルを検出しました。 • SearchReplace: 1 つ以上の検索および置換ルールをインクルード・テキストに適用して、更新済みテキストを生成しました。 • SearchReplace+AutoResolver: 1 つ以上の検索および置換ルールをインクルード・テキストに適用して、更新済みテキストを生成しました。その後、更新済みテキストに内部ヒューリスティックを適用して、ファイルを検出しました。 • SearchReplace+IncludePath: 1 つ以上の検索および置換ルールをインクルード・テキストに適用して、更新済みテキストを生成しました。その後、更新済みテキストを include パス上のディレクトリーと結合して、ファイルを検出しました。 • SearchReplace+RelativeDir: 1 つ以上の検索および置換ルールをインクルード・テキストに適用して、更新済みテキストを生成しました。その後、include 式が含まれるファイルのソース・ディレクトリーを基準として、ファイルを検出しました。
ソース・ファイル、行、列	これらの列は、ソース・コード内でこの式が使用された場所を示します。 エディター内でこれらの場所を表示して、どのように解決すべきかを確認できます。

注: 一部の列は空白になっていることがあります。 これは、インクルード・テキストを展開すると、複数の更新済みテキスト行が表示されることがあるためです。 これらの列では、更新済みテキスト行ごとに該当するテキストが表示されます。

3. ダイアログ・ボックスの下部にある「インクルード・パス」タブには、「PHP プロジェクト依存関係」ページに入力したのと同じ include パス情報が表示されます。 (未解決の include 式を表示しているときに) このダイアログ・ボックスにあるこの情報を更新できます。
4. ダイアログ・ボックスの下部にある「検索および置換」タブを使用して、include 式内の動的テキストを、include ファイルへの完全ファイル・パスまたは部分ファイル・パスを示す静的テキストに置き換えるためのルールを追加できます。 「検索および置換」表には、3 つの列があります。

オプション	説明
コマンド	この列の値は、「検索テキスト」および「置換テキスト」列の使用方法を決定します。 以下の選択項目があります。 • テキストの置換: このコマンドは、単純なテキスト検索および置換に使用されます。「検索テキスト」はそのまま使用されます。検索テキストは、インクルード・テキスト内のどこかで検出されると、「置換テキスト」に置き換えられます。 • 関数の置換: このコマンドは、関数呼び出しを置換するときに使用されます。 「検索テキスト」は、括弧なしの関数の名前である必要があります。 検索テキストは、指定された名前および後続の括弧を持つ関数呼び出しを検索できるように拡張されます。関数呼び出しが条件を満たす場合には、括弧内の値とは無関係に一致します。 • 正規表現の置換: これは、検索テキストとして正規表現を指定できる拡張機能です。
テキスト検索	これは、include 式内で検索するテキストです。 include 式のテキストの一部を選択し、クリップボードにコピーし、ここに貼り付けることができます。 検索テキストのさまざまな指定方法については、上記の「コマンド」列の説明を参照してください。

オプション	説明
置換テキスト	これは、検索テキストの置換に使用されるテキストです。 これは、include ファイルへの完全ファイル・パスまたは部分ファイル・パスを示す静的テキストです。置換テキストには、何種類かの変数を含むことができます。 変数は、「置換テキスト」セル内に直接入力するか、表の上の「置換テキスト変数」メニューから選択することができます (これにより、選択された変数がクリップボードにコピーされます)。「置換テキスト変数」メニュー・リストから選択できる変数は、以下のとおりです。 • %ROOT_DIR%: この変数は、プロジェクトに対して指定された PHP 文書ルート・ディレクトリーに置き換えられます。 • %SRC_DIR%: この変数は、include 式が含まれるファイルのディレクトリーに置き換えられます。 • %ARG_N%: この変数は、コマンドが「関数の置換」である場合にのみ適用されます。変数内の N は、整数に置き換える必要があります (%ARG_1% または %ARG_2% など)。これにより、この変数は、関数呼び出しの N 番目のパラメーターに渡されるテキストに置き換えられます。

ルールは順次的に適用されます。 検索および置換操作が正常に完了するごとに、ファイルを検出できるかどうかを確認するために、新規テキストがチェックされます。 ファイルが検出されない場合は、更新済みテキストに対して、その次のルールが試行されます。

すべての PHP プロジェクトは、include 式内で一般的に使用されるいくつかの標準 PHP 定数および関数の置換を試行する、3 つの検索および置換ルールで開始されます。

例: 未解決の **include** 式の構成:
始める前に

未解決の include 式は、「プロジェクト・プロパティー」の「プロジェクト依存関係」タブ付きページ内で構成されます。ページ内で、「未解決のインクルード式の構成」をクリックして、「未解決のインクルード式の構成」ダイアログ・ボックスを開きます。

この例では、include パスがプロジェクトに追加されていること (これは、プロジェクトの作成時に、または「プロジェクト依存関係」ページの表示中に実行できます)、およびスキャンが実行されていることを前提としています。 スキャンが完了した後で、「未解決のインクルード式の構成」ダイアログ・ボックスを開いて、「未解決のインクルード式」リストを表示します。この例では、

MYPROJECT_ROOT_PATH.'/a/b/filename.php'、MYPROJECT_ROOT_PATH.'/language/'.$configInfo['language'].'/mypage.php'、および configGet ('database_inc','./includes/database.inc') がそのリスト内の式です。

手順

1. 以下のステップを実行して、先頭の PHP 定数または変数をディレクトリーに置き換えます。
   a. MYPROJECT_ROOT_PATH.'/a/b/filename.php' を選択します。これにより、式の中のテキストが選択されます。その後、マウスまたはカーソル・キーを使用して、式の一部を選択できるようになります。 MYPROJECT_ROOT_PATH を選択してから、右クリックし、「コピー」を選択します。
   b. 「検索および置換」タブを選択します。
   c. 「選択された未解決の項目のルールを追加」ボタン (緑のプラス記号で装飾されています) をクリックします。これにより、新規の検索および置換ルールがリストに追加されます。
   d. 新規ルールで、NewSearchText を選択してから、右クリックし、メニューから「貼り付け」を選択します。 これにより、NewSearchText が MYPROJECT_ROOT_PATH に置き換えられます。
   e. 「置換テキスト変数」メニューから、%ROOT_DIR% を選択します。 これにより、%ROOT_DIR% テキスト文字列がクリップボードにコピーされます。
   f. ルールで、NewReplacementText を選択してから、右クリックし、メニューから「貼り付け」を選択します。 これにより、NewReplacementText が %ROOT_DIR% に置き換えられます。

   この新規ルールは、ある定数を、PHP 文書ルート・ディレクトリーへのパスに置き換えます。 PHP 連結演算子 (.) およびそれに続くテキスト文字列が、置換テキストと結合されて、単一のパス式が生成されます。 次にプロジェクトがスキャンされるときには、この定数を使用する include 式が正常に使用されます。
2. 動的式を単一値に置き換えるには、以下のようにします。
   a. MYPROJECT_ROOT_PATH.'/language/'.$configInfo['language'].'/mypage.php' を選択します。これにより、式の中のテキストが選択されます。その後、マウスまたはカーソル・キーを使用して、式の一部を選択できるようになります。 $configInfo['language'] を選択してから、右クリックし、「コピー」を選択します。
   b. 「検索および置換」タブを選択します。
   c. 「選択された未解決の項目のルールを追加」ボタン (緑のプラス記号で装飾されています) をクリックします。これにより、新規の検索および置換ルールがリストに追加されます。
   d. 新規ルールで、NewSearchText を選択してから、右クリックし、メニューから「貼り付け」を選択します。 これにより、NewSearchText が $configInfo['language'] に置き換えられます。
   e. ルールで、NewReplacementText を選択してから、これを置き換える新規テキストとして english を入力します。

この新規ルールは、式を、指定された値に置き換えます。 PHP 連結演算子 (.) が適用されて、単一のパス式が生成されます。 次にプロジェクトがスキャンされるときには、この式を使用する include 式が正常に使用されます。

3. PHP 関数呼び出しをその引数の 1 つに置き換えるには、以下のようにします。
   a. configGet('database_inc','./includes/database.inc') を選択します。これにより、式内のテキストが選択されます。 その後、マウスまたはカーソル・キーを使用して、式の一部を選択できるようになります。 configGet を選択してから、右クリックし、「コピー」を選択します。
   b. 「検索および置換」タブを選択します。
   c. 新規ルールで、最初の列内の「テキストの置換」を選択し、メニューから「関数の置換」を選択します。
   d. ルールで、NewSearchText を選択してから、右クリックし、メニューから「貼り付け」を選択します。 これにより、NewSearchText が configGet に置き換えられます。
   e. 「置換テキスト変数」メニューから、%ARG_1% を選択します。 これにより、変数がクリップボードにコピーされます。
   f. ルールで、NewReplacementText を選択してから、右クリックし、メニューから「貼り付け」を選択します。 貼り付けられたテキスト %ARG_1% を編集して %ARG_2% になるようにします。

   この新規ルールは、関数呼び出しを、その 2 番目のパラメーターの値に置き換えます。 次にプロジェクトがスキャンされるときには、この関数呼び出しを使用する include 式が正常に使用されます。

未解決の PHP クラス参照の構成

始める前に

「プロジェクト・プロパティー」で、「プロジェクト依存関係」ページに移動します。

手順

1. 「未解決のクラス参照の構成」をクリックして、「未解決のクラス参照の構成」ダイアログ・ボックスを開きます。
2. ダイアログ・ボックスの上部には、未解決のクラス参照 (最後のスキャン中に解決されなかったすべてのクラス参照) がリストされます。 クラス参照について提供される情報は、以下のとおりです。

オプション	説明
クラス名 / 生成済みファイル名	この列には、ソース・コード内で参照されたクラス名が表示されます。 「+」をクリックしてこの列を展開すると、最後のスキャン中にクラスの検出を試行するために使用された生成済みファイル名が表示されます。 列を展開すると、複数のファイル名が表示されることがあります。各ファイル名は、ソース・コード内でこのクラスが使用されたそれぞれの場所に対応します。

オプション	説明
状態	この列には、未解決のクラスを示す「X」または正常に解決されたクラスを示すチェック・マークが表示されます。
解決方法	この列は、生成済みファイル名が作成された方法を示します。以下の値が含まれます。 • AutoResolver: アプリケーションは、内部ヒューリスティックを使用して、ファイルを検出しました。 • SearchReplace: 1 つ以上の検索および置換ルールをクラス名に適用して、生成済みファイル名を作成しました。
ソース・ファイル、行、列	これらの列は、ソース・コード内でこのクラスが使用された場所を示します。 エディター内でこれらの場所を表示して、どのように解決すべきかを確認できます。

注: 一部の列は空白になっていることがあります。 これは、クラス名を展開すると、複数の生成済みファイル名の行が表示されることがあるためです。 これらの列では、生成済みファイル名の行ごとに該当するテキストが表示されます。

3. ダイアログ・ボックスの下部にある「クラス・インクルード・パス」タブには、「PHP プロジェクト依存関係」ページに入力したのと同じクラス include パス情報が表示されます。 未解決のクラス参照を表示しているときに、このダイアログ・ボックス内のこの情報を更新できます。
4. ダイアログ・ボックスの下部にある「検索および置換」タブを使用して、未解決のクラス名を、そのクラスの定義が含まれる完全ファイル・パスまたは部分ファイル・パスに変更するためのルールを追加できます。 「検索および置換」表には、3 つの列があります。

オプション	説明
コマンド	この列の値は、「検索テキスト」および「置換テキスト」列の使用方法を決定します。 以下の選択項目があります。 • テキストの一致: このコマンドは、テキスト検索および置換に使用されます。 検索テキストには * 文字を使用でき、0 個以上の任意の文字との一致を検索できます。 一致テキストの結果は、その他の検索および置換ルールには影響しません。 通常の場合、これは、ファイル名の生成時に、クラス名に拡張子を追加したり、クラス名から接頭部および接尾部を取り除いたりするために使用されます。 • テキストの置換: このコマンドは、単純なテキスト検索および置換に使用されます。 「検索テキスト」はそのまま使用され、クラス名の中でこのテキストが検出された場合には、「置換テキスト」に置き換えられます。 これは、後続のルールで使用するクラス名を変更するために使用されます。 • 正規表現の置換: これは、検索テキストとして正規表現を指定できる拡張機能です。
テキスト検索	これは、クラス参照内で検索するテキストです。 クラス名のテキストの一部を選択し、クリップボードにコピーし、ここに貼り付けることができます。 検索テキストのさまざまな指定方法については、上記の「コマンド」列の説明を参照してください。

オプション	説明
置換テキスト	これは、検索テキストの置換に使用されるテキストです。 これが、クラスの定義が含まれるファイルへの完全ファイル・パスまたは部分ファイル・パスを示す静的テキストになります。 置換テキストには、何種類かの変数を含むことができます。 変数は、「置換テキスト」セル内に直接入力するか、表の上の「置換テキスト変数」メニューから選択することができます (これにより、選択された変数がクリップボードにコピーされます)。「置換テキスト変数」メニュー・リストから選択できる変数は、以下のとおりです。 • %ROOT_DIR%: この変数は、プロジェクトに対して指定された PHP 文書ルート・ディレクトリーに置き換えられます。 • %SRC_DIR%: この変数は、クラスへの参照が含まれるファイルのディレクトリーに置き換えられます。 • %MATCH_N%: この変数は、コマンドが「テキストの一致」である場合にのみ適用されます。 変数内の N は、整数に置き換える必要があります (%MATCH_1% または %MATCH_2% など)。これにより、この変数は、検索テキスト内の N 番目の * に一致するテキストに置き換えられます。

ルールは順次的に適用されます。 検索および置換操作が正常に完了するごとに、ファイルを検出できるかどうかを確認するために、新規テキストがチェックされます。 ファイルが検出されない場合は、更新済みテキストに対して、その次のルールが試行されます (コマンドが「テキストの一致」である場合を除く)。

すべての PHP プロジェクトは、クラス名に対していくつかの共通ファイル拡張子の追加を試行する 2 つの単純な検索および置換ルールで開始されます。

5. ダイアログ・ボックスの下部にある「検出されたクラス」タブには、最後のスキャン中に検出されたすべてのクラスがリストされます。 これを使用して、クラス include パスと検索および置換ルールを更新できます。 ダイアログの「未解決のクラス参照」セクション内の未解決のクラス参照を選択し、「宣言の検出」をクリックすることができます。 宣言が検出された場合は、「検出されたクラス」タブ・リスト内に表示されます。

例: 未解決のクラス参照の構成:
始める前に

未解決のクラス参照は、「プロジェクト・プロパティー」の「プロジェクト依存関係」タブ付きページ内で構成されます。 ページ内で、「未解決のクラス参照の構成」をクリックして、「未解決のクラス参照の構成」ダイアログ・ボックスを開きます。

この例では、クラス include パスがプロジェクトに追加されていること (これは、プロジェクトの作成時に、または「プロジェクト依存関係」ページの表示中に実行できます)、およびスキャンが実行されていることを前提としています。 スキャンが完了した後で、「未解決のクラス参照の構成」ダイアログ・ボックスを開いて、「未解決のクラス参照」リストを表示します。

以下の例では、置換テキスト値を指定します。 これらの値のテキストには、複数の変数が含まれる場合があることに注意してください (例: %ROOT_DIR%/modules/%MATCH_1%/classes/%MATCH_1%.class.inc)。

手順

1. include ファイルで使用される別のファイル拡張子を追加するには、以下のようにします。
   a. 「検索および置換」タブを選択します。
   b. 「選択された未解決の項目のルールを追加」ボタン (緑のプラス記号で装飾されています) をクリックします。これにより、新規の検索および置換ルールがリストに追加されます。
   c. 新規ルールで、「置換テキスト」内の %MATCH_1%.php を選択します。この文字列で、.php を削除し、代わりに .class.inc を入力します。 「置換テキスト」は %MATCH_1%.class.inc になります。

   この新規ルールは、クラスを解決するときに、クラス名に接尾部 .class.inc を追加しようとします。
2. クラス名から接頭部を削除するには、以下のようにします。
   a. 「検索および置換」タブを選択します。
   b. 「選択された未解決の項目のルールを追加」ボタン (緑のプラス記号で装飾されています) をクリックします。これにより、新規の検索および置換ルールがリストに追加されます。
   c. 新規ルールで、「検索テキスト」列内の文字列 (*) を選択し、代わりに Abc* を入力します。
   d. %MATCH_1%.php 置換テキストを変更しないでください。

   この新規ルールは、AbcHello のようなクラス名を Hello.php にマップします。
3. クラス名から接尾部を削除するには、以下のようにします。
   a. 「検索および置換」タブを選択します。
   b. 「選択された未解決の項目のルールを追加」ボタン (緑のプラス記号で装飾されています) をクリックします。これにより、新規の検索および置換ルールがリストに追加されます。
   c. 新規ルールで、「検索テキスト」列内の文字列 (*) を選択し、代わりに *Xyz を入力します。
   d. %MATCH_1%.php 置換テキストを変更しないでください。

   この新規ルールは、ByeByeXyz のようなクラス名を ByeBye.php にマップします。

4. Abc_Def_Ghi_class のようなクラス名をマップして、それらの接頭部をファイル・システムへの相対パスとして使用できます (例: Abc/Def/Ghi/class.php)。他のルールで使用するためにクラス名テキストを変更するには、以下のようにします。
   a. 「検索および置換」タブを選択します。
   b. 「選択された未解決の項目のルールを追加」ボタン (緑のプラス記号で装飾されています) をクリックします。これにより、新規の検索および置換ルールがリストに追加されます。
   c. 新規ルールで、最初の列内の「テキストの一致」を選択し、メニューから「テキストの置換」を選択します。
   d. ルールで、「検索テキスト」列内の文字列 (*) を選択し、代わりに _ を入力します。
   e. 「置換テキスト」列内の文字列を選択し、代わりに / を入力します。
   f. ルールを選択した状態で「上へ移動」をクリックすると、このルールがリストの先頭に移動します。

   この新規ルールは、下線 (_) をスラッシュ (/) に置き換え、更新済みテキストがすべての後続のルールで使用されます。 このルールは、Abc_Def_Ghi_class を Abc/Def/Ghi/class に変更し、後続の「テキストの一致」ルールは、.php や .inc などの拡張子を追加しようとします。

新規 PL/SQL プロジェクトの追加

新規プロジェクト・ウィザードを使用すれば、PL/SQL プロジェクトを手動で作成し、そのプロジェクトをアプリケーションに追加することができます。

このタスクについて

このトピックの手順では、新規プロジェクト・ウィザード (プロジェクトをアプリケーション内に作成する場合は、新規アプリケーション・ウィザード) のすべてのページで設定を完了するよう指示しています。ウィザードで行った設定は、選択したプロジェクトの「プロパティー」ビューでプロジェクトを作成した後に変更できます。

手順

1. 「エクスプローラー」ビューで、プロジェクトを追加するアプリケーションを選択します (アプリケーションをまだ追加していない場合は、39 ページの『アプリケーションの構成』を参照してください)。
2. 以下のいずれかのアクションを実行して、新規プロジェクト・ウィザードを開きます。
   a. ワークベンチのメインメニューで、「ファイル」 > 「プロジェクトの追加」 > 「新規プロジェクト」を選択します。
   b. 選択したアプリケーションを右クリックし、コンテキスト・メニューから「プロジェクトの追加」 > 「新規プロジェクト」を選択します。
3. ウィザードの「プロジェクト・タイプの選択」ページで、プロジェクト・タイプとして「**PL/SQL**」を選択し、「次へ」をクリックして次のウィザード・ページに進みます。
4. 「プロジェクト・ソース」ウィザード・ページで、以下のようにします。

a. プロジェクト・ソースを指定します。プロジェクト・ソースは、プロジェクト・ファイルが格納されているディレクトリー、およびプロジェクトに含める追加の個別ファイルから成り立っています。

   プロジェクトに名前を付けて、作業ディレクトリーを指定します。「作業ディレクトリー」は、AppScan Source プロジェクト・ファイル (.ppf) が置かれる場所です。これは、すべての相対パスの基準にもなります。

b. 「ソース・ルートの追加」をクリックして、ソース・コード・ルートを指定し、スキャンに含めるまたはスキャンから除外するディレクトリーまたはファイルを指定します。 ソース・ルートを追加した後で、特定のディレクトリーまたはファイルを除外できます。 このためには、ソース・ルートでディレクトリーまたはファイルを選択 (するか、これらの項目を複数選択) し、選択項目を右クリックして、メニューから「除外」を選択します。ファイルを含めるかまたは除外すると、ファイル名の左側にあるアイコンが変更されます。

5. 「完了」をクリックします。

新規 T-SQL プロジェクトの追加

新規プロジェクト・ウィザードを使用すれば、T-SQL プロジェクトを手動で作成し、そのプロジェクトをアプリケーションに追加することができます。

このタスクについて

このトピックの手順では、新規プロジェクト・ウィザード (プロジェクトをアプリケーション内に作成する場合は、新規アプリケーション・ウィザード) のすべてのページで設定を完了するよう指示しています。ウィザードで行った設定は、選択したプロジェクトの「プロパティー」ビューでプロジェクトを作成した後に変更できます。

手順

1. 「エクスプローラー」ビューで、プロジェクトを追加するアプリケーションを選択します (アプリケーションをまだ追加していない場合は、 39 ページの『アプリケーションの構成』を参照してください)。
2. 以下のいずれかのアクションを実行して、新規プロジェクト・ウィザードを開きます。
   a. ワークベンチのメインメニューで、「ファイル」 > 「プロジェクトの追加」 > 「新規プロジェクト」を選択します。
   b. 選択したアプリケーションを右クリックし、コンテキスト・メニューから「プロジェクトの追加」 > 「新規プロジェクト」を選択します。
3. ウィザードの「プロジェクト・タイプの選択」ページで、プロジェクト・タイプとして「**T-SQL**」を選択し、「次へ」をクリックして次のウィザード・ページに進みます。
4. 「プロジェクト・ソース」ウィザード・ページで、以下のようにします。
   a. プロジェクト・ソースを指定します。プロジェクト・ソースは、プロジェクト・ファイルが格納されているディレクトリー、およびプロジェクトに含める追加の個別ファイルから成り立っています。

プロジェクトに名前を付けて、作業ディレクトリーを指定します。「作業ディレクトリー」は、AppScan Source プロジェクト・ファイル (.ppf) が置かれる場所です。これは、すべての相対パスの基準にもなります。

b. 「ソース・ルートの追加」をクリックして、ソース・コード・ルートを指定し、スキャンに含めるまたはスキャンから除外するディレクトリーまたはファイルを指定します。 ソース・ルートを追加した後で、特定のディレクトリーまたはファイルを除外できます。 このためには、ソース・ルートでディレクトリーまたはファイルを選択 (するか、これらの項目を複数選択) し、選択項目を右クリックして、メニューから「除外」を選択します。ファイルを含めるかまたは除外すると、ファイル名の左側にあるアイコンが変更されます。

5. 「完了」をクリックします。

新規 Visual Basic プロジェクトの追加

このタスクについて

注: このプロジェクト・タイプは Windows でのみサポートされています。

このトピックの手順では、新規プロジェクト・ウィザード (プロジェクトをアプリケーション内に作成する場合は、新規アプリケーション・ウィザード) のすべてのページで設定を完了するよう指示しています。ウィザードで行った設定は、選択したプロジェクトの「プロパティー」ビューでプロジェクトを作成した後に変更できます。

注: PHP、VB6、および Classic ASP では、 ISO-8859-1 (西ヨーロッパ)、UTF-8、および UTF-16 の文字セットのみがサポートされます。

手順

1. 「エクスプローラー」ビューで、プロジェクトを追加するアプリケーションを選択します (アプリケーションをまだ追加していない場合は、 39 ページの『アプリケーションの構成』を参照してください)。
2. 以下のいずれかのアクションを実行して、新規プロジェクト・ウィザードを開きます。
   a. ワークベンチのメインメニューで、「ファイル」 > 「プロジェクトの追加」 > 「新規プロジェクト」を選択します。
   b. 選択したアプリケーションを右クリックし、コンテキスト・メニューから「プロジェクトの追加」 > 「新規プロジェクト」を選択します。
3. ウィザードの「プロジェクト・タイプの選択」ページで、プロジェクト・タイプとして「**Visual Basic**」を選択し、「次へ」をクリックして次のウィザード・ページに進みます。
4. 「プロジェクト・ソース」ウィザード・ページで、以下のようにします。
   a. プロジェクト・ソースを指定します。プロジェクト・ソースは、プロジェクト・ファイルが格納されているディレクトリー、およびプロジェクトに含める追加の個別ファイルから成り立っています。

プロジェクトに名前を付けて、作業ディレクトリーを指定します。「作業ディレクトリー」は、AppScan Source プロジェクト・ファイル (.ppf) が置かれる場所です。これは、すべての相対パスの基準にもなります。

b. 「ソース・ルートの追加」をクリックして、ソース・コード・ルートを指定し、スキャンに含めるまたはスキャンから除外するディレクトリーまたはファイルを指定します。 ソース・ルートを追加した後で、特定のディレクトリーまたはファイルを除外できます。 このためには、ソース・ルートでディレクトリーまたはファイルを選択 (するか、これらの項目を複数選択) し、選択項目を右クリックして、メニューから「除外」を選択します。ファイルを含めるかまたは除外すると、ファイル名の左側にあるアイコンが変更されます。

5. 「完了」をクリックします。

プロジェクトのコピー

AppScan Source for Analysis を使用すると、.NET プロジェクト以外のすべてのプロジェクト・タイプをコピーできます。 プロジェクトへの変更は、複製されたプロジェクトには影響しません。プロジェクトをコピーした後では、元のプロジェクトとコピーされたプロジェクトの間の関連はなくなります。 インポートされたプロジェクトをコピーすると、すべての構成情報を含む AppScan Source プロジェクト・ファイル (.ppf) が作成されます。

手順

1. 「エクスプローラー」ビューで、コピーするプロジェクトを右クリックし、メニューから「プロジェクトのコピー」を選択します。
2. 「プロジェクトのコピー」ダイアログ・ボックスで、以下のようにします。
   a. 新規プロジェクトに名前を付けます。
   b. 複製したプロジェクトの出力先アプリケーションを指定します (出力先アプリケーションは、手動で作成した AppScan Source アプリケーションか、アプリケーション・ディスカバリー・アシスタントを使用して作成したアプリケーションでなければなりません)。
   c. 出力先ディレクトリー (新規プロジェクトの作業ディレクトリー) を指定します。

アプリケーションおよびプロジェクトのプロパティーの変更

「エクスプローラー」ビューでアプリケーションまたはプロジェクトを選択すると、「プロパティー」ビュー内に現在のプロパティーが表示されるので、ここで変更を行うことができます。

このタスクについて

279 ページの『「プロパティー」ビュー: 選択したアプリケーション』および 280 ページの『「プロパティー」ビュー: 選択したプロジェクト』では、アプリケーションまたはプロジェクトを選択したときに「プロパティー」ビューで変更できる設定について詳しく説明しています。

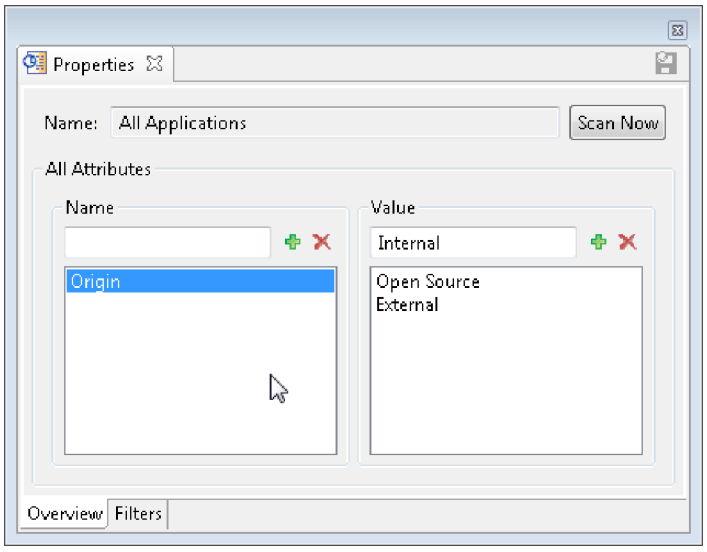
手順

1. アプリケーションまたはプロジェクトの「プロパティー」ビューを開くには、以下のいずれかの方法を使用します。
   a. 「エクスプローラー」ビューでアプリケーションまたはプロジェクトを選択すると、「プロパティー」ビューが開きそのプロパティーが表示されます。
   b. 「エクスプローラー」ビューでアプリケーションまたはプロジェクトを右クリックし、「プロパティー」を選択します。
2. 「プロパティー」ビューでプロパティーを確認します。
3. 該当するタブ・ページで変更を行います。 どのプロパティー・ページを使用できるかは、言語依存です。
4. 「保存」をクリックします。

グローバル属性

グローバル属性を個別のアプリケーションと関連付ける場合は、その前にグローバル属性を定義しておく必要があります。グローバル属性は、「エクスプローラー」ビューで「すべてのアプリケーション」を選択することにより、「プロパティー」ビューで定義されます。

このタスクについて



属性またはその値を削除するには、名前または値を選択し、「属性の削除」( ) をクリックします。 属性を削除しても、履歴結果には影響しません。

属性を作成し、その属性をすべてのアプリケーションで使用できるようにするには、以下のようにします。

手順

1. 「エクスプローラー」ビューで「すべてのアプリケーション」を選択します。
2. 「プロパティー」ビュー内の「概要」タブを開きます。

3. 属性の名前を入力し、「属性の追加」 ( ) をクリックします。または、先に名前を指定せずに「属性の追加」をクリックします (その後、ダイアログ・ボックスにより、属性の名前を入力するよう求められます)。
4. 属性の「値」を入力して、「属性値の追加」をクリックします。または、先に値を指定せずに「属性値の追加」をクリックします (その後、ダイアログ・ボックスにより、値を追加するよう求められます)。
5. 複数の属性値を追加するには、以上のステップを繰り返します。

アプリケーション属性

アプリケーション属性は、現在選択されているアプリケーションに適用され、以前に作成されたグローバル属性に依存します。

手順

1. 「エクスプローラー」ビューでアプリケーションを選択します。
2. 「プロパティー」ビュー内の「概要」タブを開きます。
3. 「属性の追加」をクリックします。 「グローバル属性」ダイアログ・ボックスが表示され、以前に作成した属性のリストが示されます (グローバル属性を作成する手順については、 94 ページの『グローバル属性』を参照してください)。
4. 追加する属性をダブルクリックします。または、追加する属性を選択して、「**OK**」をクリックします。属性は、「プロパティー」ビューの「アプリケーション属性」セクションに追加されます。
5. 「値」列をクリックし、このアプリケーションの値をリストから選択します (グローバル属性が複数の値で作成された場合は、複数の値が表示されます)。複数の属性をアプリケーションに関連付けることができます。

アプリケーションおよびプロジェクトの削除

登録されていないアプリケーションおよびプロジェクトを AppScan Source for Analysis から削除できます。

手順

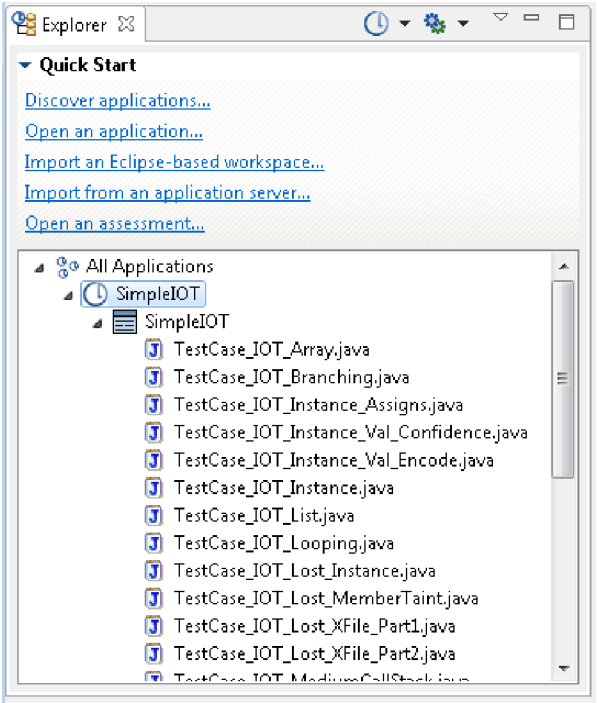
1. 削除するアプリケーションまたはプロジェクトを選択します。複数のアプリケーション、複数のプロジェクトを選択して削除することはできますが、アプリケーションとプロジェクトを組み合わせて選択して削除することはできません。
2. 以下のアクションのいずれかを実行します。
   - 選択項目を右クリックして、メニューから「アプリケーションの削除」または「プロジェクトの削除」を選択します。
   - キーボードの Delete キーを押します。
   - ワークベンチのメイン・メニューから「編集」 > 「削除」と選択します。

「エクスプローラー」ビュー

「エクスプローラー」ビューには、上部に「クイック・スタート」セクションがあり、下部に「エクスプローラー」セクションがあります。「エクスプローラー」セクションには、「すべてのアプリケーション」という 1 つのノードが含まれています。「クイック・スタート」セクションには、共通のアクションを起動するいくつかの便利なリンクが含まれています。 「エクスプローラー」セクションは、「すべてのアプリケーション」をルートとして、ご使用のリソース (アプリケーション、プロジェクト、ディレクトリー、およびプロジェクト・ファイル) を階層的に表示するツリー・ペインで構成されています。 ファイル・ブラウザーとほぼ同じようにして、これらのリソースをナビゲートします。 このビューをナビゲートするとき、このツリーの選択状態によって、「プロパティー」ビューで使用可能なタブが決まります。

- 『全般情報』
- 97 ページの『「クイック・スタート」セクション』
- 97 ページの『ツールバー・ボタン』
- 98 ページの『右クリックのメニュー・オプション』
- 101 ページの『アプリケーションおよびプロジェクトのインディケーター』

全般情報



「エクスプローラー」ビューでは、アプリケーションおよびプロジェクトを追加し、ツールバーのボタン、「クイック・スタート」セクションのリンク、および「エクスプローラー」セクションの右クリック・メニュー・コマンドを使用してコードをスキャンします。 アプリケーションを追加したら、「エクスプローラー」セクションに、アプリケーションおよびプロジェクトのビジュアル・インディケーターと、それぞれの状態が表示されます。

ヒント: 「エクスプローラー」ビューで、吹き出しヘルプに、アプリケーション、プロジェクト、およびファイルのファイル名とパスが表示されます。吹き出しヘルプには、アプリケーションまたはプロジェクトが登録されているかどうかも示されます。

「クイック・スタート」セクション

「クイック・スタート」セクションには、共通タスクを起動するための以下のリンクがあります。

- アプリケーションのディスカバー: これにより、アプリケーション・ディスカバリー・アシスタントが起動します。これを使用すると、Java および Microsoft Visual Studio ソース・コード用のアプリケーションおよびプロジェクトを迅速に作成および構成できます。
- アプリケーションを開く: これにより「オープン」ダイアログ・ボックスが起動します。このダイアログ・ボックスで、既存のアプリケーションを参照して、一連のアプリケーションに追加できます。追加できるファイルまたはディレクトリーのタイプとしては、`.paf`、`.sln`、`.dsw`、および `.ewf` があります。
- **Eclipse** ベースのワークスペースのインポート: これにより「ワークスペースの追加」ダイアログ・ボックスが起動します。このダイアログ・ボックスで、Java プロジェクトが含まれている既存の Eclipse ワークスペースまたは IBM Rational Application Developer for WebSphere Software (RAD) ワークスペースを追加できます。ワークスペースのインポートが完了したら、そのワークスペースに含まれているすべての Java プロジェクトをスキャンできます。

  注: ワークスペースをインポートする前に、 52 ページの『Eclipse プロジェクトおよび Rational Application Developer for WebSphere Software (RAD) プロジェクトの開発環境の構成』で説明されているように開発環境がインストールおよび更新されていることを確認してください。
- アプリケーション・サーバーからのインポート: Apache Tomcat または WebSphere Application Server Liberty アプリケーション・サーバーから既存の Java アプリケーションをインポートします。
- 評価を開く: これにより「オープン」ダイアログ・ボックスが起動します。そのダイアログ・ボックスで、AppScan Source 評価ファイルを参照できます。開くことができるファイルのタイプとしては、`.ozasmt` および `.xml` があります。

ツールバー・ボタン

表 *8.* ツールバー・ボタン

アクション	アイコン	説明
アプリケーション・メニューの追加		「アプリケーション・メニューの追加」ボタンの下矢印をクリックすると、新規アプリケーションの作成、既存のアプリケーションのオープン、ワークスペースのインポート、または アプリケーション・ディスカバリー・アシスタントの起動のためのアクションを選択できます。

表 8. ツールバー・ボタン (続き)

アクション	アイコン	説明
選択項目のスキャン		「選択項目のスキャン」ボタンを使用すると、「エクスプローラー」セクションで選択されるオブジェクトをスキャンできます。 スキャンには、デフォルトのスキャン構成が使用されます。別のスキャン構成を選択してスキャンに使用する場合は、「選択項目のスキャン」ボタンの下矢印をクリックします。使用するスキャン構成を選択します。あるいは、「構成の編集」アクションを選択して、別のスキャン構成をデフォルトとして設定します (「スキャン構成」ビューで、デフォルトとして設定する構成を選択し、「デフォルトとして選択」をクリックします)。
「表示」メニュー		「表示メニュー」ボタンは、「エクスプローラー」セクションを更新したり、登録済みの項目を非表示にしたりするためのメニューを開きます。

右クリックのメニュー・オプション

右クリックのメニュー・オプションが使用可能であるかどうかは、「エクスプローラー」セクションで選択されている項目によって決まります。

- 「エクスプローラー」セクションで「すべてのアプリケーション」が選択されている場合は、以下の右クリックのメニュー・オプションが使用可能になります。
  - すべてのアプリケーションのスキャン: すべてのアプリケーションをスキャンします。デフォルトのスキャン構成を使用してスキャンが実行されます。
  - すべてのアプリケーションのスキャン: 使用するスキャン構成を選択します。あるいは、「構成の編集」アクションを選択して、別のスキャン構成をデフォルトとして設定します (「スキャン構成」ビューで、デフォルトとして設定する構成を選択し、「デフォルトとして選択」をクリックします)。
  - アプリケーションの追加
    - 新規レポートの作成: 新規アプリケーションを一連のアプリケーションに追加します。このアクションによって、新規アプリケーション・ウィザードが起動します。
    - 既存のアプリケーションを開く: これにより「オープン」ダイアログ・ボックスが起動します。このダイアログ・ボックスで、既存のアプリケーシ

ョンを参照して、一連のアプリケーションに追加できます。追加できるファイルまたはディレクトリーのタイプとしては、.paf、.sln、.dsw、および .ewf があります。

- 既存の **Eclipse** ベースのワークスペースのインポート: これにより「ワークスペースの追加」ダイアログ・ボックスが起動します。このダイアログ・ボックスで、Java プロジェクトが含まれている既存の Eclipse ワークスペースまたは IBM Rational Application Developer for WebSphere Software (RAD) ワークスペースを追加できます。ワークスペースのインポートが完了したら、そのワークスペースに含まれているすべての Java プロジェクトをスキャンできます。

  注**:** ワークスペースをインポートする前に、 52 ページの『Eclipse プロジェクトおよび Rational Application Developer for WebSphere Software (RAD) プロジェクトの開発環境の構成』で説明されているように開発環境がインストールおよび更新されていることを確認してください。

- アプリケーションのディスカバー: これにより、アプリケーション・ディスカバリー・アシスタントが起動します。これを使用すると、Java および Microsoft Visual Studio ソース・コード用のアプリケーションおよびプロジェクトを迅速に作成および構成できます。

– すべて展開
– すべて縮小
– プロパティー: このオプションを選択すると、選択されている項目の「プロパティー」ビューが開きます。

• 「エクスプローラー」セクションでアプリケーションが選択されている場合は、以下の右クリックのメニュー・オプションが使用可能になります。
  – アプリケーションのスキャン: 選択されたアプリケーション、プロジェクト、またはファイルをスキャンします。デフォルトのスキャン構成を使用してスキャンが実行されます。
  – アプリケーションのスキャン: 使用するスキャン構成を選択します。あるいは、「構成の編集」アクションを選択して、別のスキャン構成をデフォルトとして設定します (「スキャン構成」ビューで、デフォルトとして設定する構成を選択し、「デフォルトとして選択」をクリックします)。
  – プロジェクトの追加
    - 新規プロジェクト: このアクションは、「エクスプローラー」ビューでアプリケーションを選択している場合に使用可能です。これを選択すると、アプリケーションに新規プロジェクトを追加できます。このアクションによって、新規プロジェクト・ウィザードが起動します。
    - 既存のプロジェクト: このアクションは、「エクスプローラー」ビューでアプリケーションを選択している場合に使用可能です。これを選択すると、アプリケーションに既存のプロジェクトを追加できます。このアクションによりダイアログ・ボックスが起動します。このダイアログ・ボックスで、.ppf、.vcproj、.vcxproj、.csproj、.vbproj、.dsp、または .epf ファイルを参照して開くことができます。
    - 複数のプロジェクト: 「エクスプローラー」ビューで選択したアプリケーションに複数のプロジェクトを追加します。このアクションは、以下のいずれかのタスクを実行するダイアログ・ボックスを起動します。

- プロジェクトを検索するディレクトリーを指定する。
- プロジェクトを検索するワークスペースを指定する。
- プロジェクトを検索する Microsoft ソリューション・ファイルを指定する。

検索した結果、1 つ以上のプロジェクトを選択して追加できます。

– アプリケーションの削除: このアクションは、「エクスプローラー」ビューでアプリケーションを選択している場合に使用可能です。これを選択すると、選択されているアプリケーションが削除されます。
– カスタム検出結果の追加: このアクションは、「カスタム検出結果の作成」ダイアログ・ボックスを起動します。このダイアログ・ボックスで、選択したアプリケーションのカスタム検出結果を作成することができます。
– 更新: 選択したアプリケーション、プロジェクト、またはビューのコンテンツを更新します。
– 登録/登録抹消:
  - アプリケーションの登録: 選択したアプリケーションまたはプロジェクトを AppScan Source に登録します。アプリケーションおよびプロジェクトを AppScan Source データベース に公開するには、事前に登録しておく必要があります。
  - アプリケーションに名前を付けて登録...: 新しい名前でアプリケーションを再登録する場合は、このオプションを選択します。
  - アプリケーションの登録抹消: 選択したアプリケーションまたはプロジェクトの登録を抹消します。
  - 位置指定: ローカルのアプリケーション/プロジェクトを、別の AppScan Source ユーザーが登録したアプリケーション/プロジェクトに関連付ける場合は、このオプションを選択します。
– すべて展開
– すべて縮小
– プロパティー: このオプションを選択すると、選択されている項目の「プロパティー」ビューが開きます。

- 「エクスプローラー」セクションでプロジェクトが選択されている場合は、以下の右クリックのメニュー・オプションが使用可能になります。
  – プロジェクトのスキャン: 選択されたアプリケーション、プロジェクト、またはファイルをスキャンします。デフォルトのスキャン構成を使用してスキャンが実行されます。
  – プロジェクトのスキャン: 使用するスキャン構成を選択します。あるいは、「構成の編集」アクションを選択して、別のスキャン構成をデフォルトとして設定します (「スキャン構成」ビューで、デフォルトとして設定する構成を選択し、「デフォルトとして選択」をクリックします)。
  – プロジェクトのコピー: このアクションは、「エクスプローラー」ビューでプロジェクトを選択している場合に使用可能です。これを選択すると、ダイアログ・ボックスが開き、プロジェクトを別のアプリケーションにコピーしたり、現在プロジェクトが含まれているアプリケーションにそのプロジェクトのコピーを作成したりすることができます。

- プロジェクトの削除: 選択したオブジェクトを除去します。
- 登録/登録抹消:
  - プロジェクトの登録: 選択したアプリケーションまたはプロジェクトを AppScan Source に登録します。アプリケーションおよびプロジェクトを AppScan Source データベース に公開するには、事前に登録しておく必要があります。
  - プロジェクトの登録抹消: 選択したアプリケーションまたはプロジェクトの登録を抹消します。
  - 位置指定: ローカルのアプリケーション/プロジェクトを、別の AppScan Source ユーザーが登録したアプリケーション/プロジェクトに関連付ける場合は、このオプションを選択します。
- すべて展開
- すべて縮小
- プロパティー: このオプションを選択すると、選択されている項目の「プロパティー」ビューが開きます。

- 「エクスプローラー」セクションでファイルが選択されている場合は、以下の右クリックのメニュー・オプションが使用可能になります。
  - ファイルのスキャン: 選択されたアプリケーション、プロジェクト、またはファイルをスキャンします。デフォルトのスキャン構成を使用してスキャンが実行されます。
  - ファイルのスキャン: 使用するスキャン構成を選択します。あるいは、「構成の編集」アクションを選択して、別のスキャン構成をデフォルトとして設定します (「スキャン構成」ビューで、デフォルトとして設定する構成を選択し、「デフォルトとして選択」をクリックします)。
  - スキャンから除外: 選択されているファイルをスキャンから除外します。
  - 内部エディターで開く: 選択したファイルを AppScan Source エディター (「分析」パースペクティブ内) で開きます。
  - 「外部エディターで開く」:選択したファイルを開く外部エディターを選択します。
  - プロパティー: このオプションを選択すると、選択されている項目の「プロパティー」ビューが開きます。

アプリケーションおよびプロジェクトのインディケーター

次の表に、「エクスプローラー」ビューでのアプリケーションおよびプロジェクトのアイコンを示します。

表 9. アプリケーションおよびプロジェクトのアイコン

アプリケーションまたはプロジェクトのタイプ	未登録	登録済み	存在しない/未検出
インポートされたアプリケーション			

表 9. アプリケーションおよびプロジェクトのアイコン (続き)

アプリケーションまたはプロジェクトのタイプ	未登録	登録済み	存在しない/未検出
手動で作成されたか、または アプリケーション・ディスカバリー・アシスタントを使用して作成されたアプリケーション			
インポートされたプロジェクト			
手動で作成されたか、または アプリケーション・ディスカバリー・アシスタントを使用して作成されたプロジェクト			

「エクスプローラー」ビューには、ローカル・アプリケーションおよびローカル・プロジェクトだけでなく、サーバーに登録されているアプリケーションおよびプロジェクトも表示されます (例えば、他のユーザーが登録したアプリケーションおよびプロジェクトなど、サーバーに登録されていてもローカルに保存されていないアプリケーションおよびプロジェクトはグレー表示されています)。 ツールバーの「表示メニュー」ボタンをクリックし、「サーバーに登録されている項目の非表示」メニュー項目を切り替えて選択解除すると、既存のサーバー・アプリケーションおよびプロジェクトを表示できます。プロジェクトがグレー表示されている場合は、右クリックして、メニューの「位置指定」を選択できます。

第 3 章 設定

設定は、AppScan Source for Analysis の外観および操作についての個人の選択項目です。

「設定」ページを開くには、ワークベンチのメインメニューから「編集」 > 「設定」を選択します。 「設定」ページをブラウズするときには、左側のペインにあるすべてのタイトルに目を通すか、左側のペインの上部にあるフィルター・フィールドを使用して検索を実行し、タイトルを絞り込みます。 このフィルターでは、「設定」ページのタイトルおよびキーワード (JSP や E メールなど) の両方に対して突き合わせが実行され、一致したものが結果として返されます。

右側のペインの右上にある矢印のコントロールを使用すると、以前に表示したページをナビゲートできます。 いくつかのページを表示した後で元のページに戻るには、下矢印をクリックして、最近表示した設定ページのリストを表示します。

全般設定

全般設定では、ユーザーの好みに合わせて、AppScan Source for Analysis のデフォルト設定の一部を調整できます。

言語の選択

AppScan Source for Analysis のユーザー・インターフェースは各国語で表示できます。 表示言語を変更するには、設定ダイアログ・ボックスで「言語の選択」リストから言語を選択して、「**OK**」をクリックします。 変更を有効にするにはワークベンチを手動で再始動する必要があります。

注**:** この機能を利用するには、インストール手順で少なくとも 1 つの言語パックをインストールする必要があります。 インストールした言語が英語のみの場合は、この設定を使用しワークベンチを再起動しても、製品は英語で表示されます。 その場合に英語以外の言語を表示するには、インストール・ウィザードを再実行して、1 つ以上の言語パックを追加することにより、インストールの修復を選択してください。

ファイル・エンコード

プロジェクト内のファイルの文字エンコードは、AppScan Source がファイルを適切に読み取る (そして、例えば、それらをソース・ビューに正しく表示する) ことができるように設定する必要があります。 このセクションでは、デフォルトの文字エンコードを選択します。

ロギング・レベル

エラー・ログに記録したい情報のレベルを指定するには、ロギング・レベルを変更します。「トレース」、「デバッグ」、「情報」、「警告」、「エラー」、「致命的」から選択します。 「トレース」は最も情報量が多いロギング・レベルで、以

降、順に上位レベルのロギングとなり、「致命的」は重大なイベントのみをログに記録します。

終了時にすべてのフィルターを保存

選択した場合、AppScan Source を終了するときに、プロンプトが表示されなくなり、新規作成または編集したすべてのフィルターが自動的に保存されます。

エラー時にスキャンをキャンセル

これを選択すると、不完全なスキャンを避けるため、エラーが発生した場合にスキャンがキャンセルされます。

検出結果ごとにマーカーを作成

これが選択されている場合、評価を開いて、スキャン対象ソースをエディターで開くと、そのソース内の検出結果がある場所にマーカーが表示されます。

デフォルトでは、マーカーの作成は有効になっています。

マーカーを作成すると、スキャン速度が低下する場合があります。 プロジェクトに多数のソース・ファイルや大きなソース・ファイルが含まれている場合は、マーカーの作成をオフにするとパフォーマンスを向上できる可能性があります。

スキャンの終了後

デフォルト設定では、スキャンの終了時に評価を自動的に開くかどうかを確認するプロンプトが出されます。 このプロンプトを表示したくない場合は、「常に新しい評価を開く」または「開かない」を選択します。

スキャンの完了後に同じターゲットの未保存のスキャンがある場合

デフォルトの場合、既存の保存されていないスキャンを保存するか破棄するかを選択するプロンプトが表示されます。この設定を変更して、重複するスキャンを常に自動的に削除するか、自動削除は実行しないかを選択することができます。この設定を変更する場合、重複するスキャンを削除するとメモリー使用量が減少することに注意してください。

絶対パスを使用して評価を公開またはエクスポートする場合

デフォルトの場合、評価の公開時に、絶対パスの変数を定義するためのプロンプトが表示されます。このセクションの設定を使用すると、このデフォルトのプロンプトを無効にすることも、絶対パスが存在する場合に変数を定義できるダイアログ・ボックスを自動的に表示させることもできます。

初期公開時にアプリケーションを自動登録する

デフォルトでは、未登録のアプリケーションまたはプロジェクトの評価を公開しようとすると、それらのアプリケーションまたはプロジェクトの登録を要求するプロンプトが表示されます。アプリケーションおよびプロジェクトを公開時に常に登録するか、あるいは登録しないかを選択することができます。

重要: アプリケーションおよびプロジェクトを登録するには、「登録」権限が必要です。

アプリケーション名が競合する場合

同じ名前の複数のアプリケーションが AppScan Source for Analysis に存在していてもかまいませんが、管理が困難になる可能性があります。デフォルトの場合、既存のアプリケーションと同じ名前を新しいアプリケーションに付けようとすると (または、既存のアプリケーションと同じ名前のアプリケーションをインポートしようとすると)、警告が表示されます。この警告メッセージを使用して、固有の名前をアプリケーションに付けることも、競合する名前をそのまま使用することも、操作をキャンセルすることもできます。

競合するアプリケーション名が検出された場合に、AppScan Source for Analysis によって固有の名前を自動的に生成するには、設定ページで 「固有の名前を生成する」を選択します。競合するアプリケーション名を自動的に受け入れる場合は、「既存の名前を保持する」を選択します。

注: アプリケーションは、`<application_name>.paf` というファイル名で保存されます。「既存の名前を保持する」を選択した場合、そのアプリケーションの作業ディレクトリーを、同じ名前を持つ既存のアプリケーションと同じ作業ディレクトリーとして設定することはできません。この場合、既存のファイル名を上書きするためのプロンプトが表示されますが、既存のアプリケーションは既に AppScan Source for Analysis で開かれているため、この上書き操作は失敗します。

プロジェクト名が競合する場合

この設定は、同じアプリケーション内に存在するプロジェクトと同じ名前のプロジェクトを作成またはインポートしようとする場合のみ適用されます。この場合、プロジェクト名の競合は許可されません。競合する名前のプロジェクトを作成またはインポートしようとすると、プロンプトがデフォルトで表示されます。このプロンプトでは、固有の名前を生成することも、操作をキャンセルすることもできます。競合するプロジェクト名が検出された場合に、AppScan Source for Analysis によって固有の名前を自動的に生成するには、設定ページで 「固有の名前を生成する」を選択します。

始動時に公開された評価およびマイ評価に表示される評価の数

「公開された評価」ビューまたは「自分の評価」ビューに表示する評価の最大数を設定します。

「ようこそ」ビューに表示される **RSS** フィード

デフォルトの場合、「ようこそ」ビューには、X-Force® RSS フィード・コンテンツが表示されます。代替のコンテンツを表示するには、該当する URL を「「ようこそ」ビューに表示される **RSS** フィード」フィールドに入力します。

高ネットワーク待ち時間の最適化

サーバー呼び出しを最小限にするために、クライアント上にキャッシュする情報を増やすには、このチェック・ボックスを選択します。

システム構成の再ロード

最新のシステム設定をロードします。製品の稼働中に製品の外部で設定を変更した場合 (例えば、<data_dir>¥config (<data_dir> は、ご使用の AppScan Source プログラム・データの場所です。詳しくは、 360 ページの『インストールとユーザー・データ・ファイルの場所』 を参照してください。) 内の .ozsettings ファイルを変更した場合) は、このボタンを選択して、製品の設定を更新します。

AppScan Enterprise Console の設定

ご使用の AppScan Enterprise Server が AppScan Enterprise Console オプションを指定してインストールされている場合は、Enterprise Console に評価を公開することができます。 Enterprise Console は、レポート機能、問題管理、トレンド分析、ダッシュボードなど、評価に関する作業を行うためのさまざまなツールを備えています。

この機能を有効にするには、AppScan Enterprise Console の設定ページで必要な設定を行います。 Enterprise Console の公開を有効にする前に、このページのすべてのフィールドに有効な値を入力する必要があります。

- 「ユーザー **ID**」フィールド: AppScan Enterprise Server ユーザー ID (ご使用の AppScan Source ユーザーの代わりに公開するために作成したユーザー ID) を入力します。
  - AppScan Enterprise Server が Windows 認証を使用するように構成されている場合、Enterprise Console への接続に使用するドメイン名とユーザー名を入力します。ドメイン名とユーザー名は ¥ で区切ります (例えば、my_domain¥my_username)。
  - AppScan Enterprise Server が LDAP を使用して構成されている場合、Enterprise Console への接続に使用するユーザー名を入力します。
  - Windows の場合、ご使用の AppScan Enterprise Server で Common Access Card (CAC) 認証が有効にされている場合は、管理者の CAC 共通名をリストから選択します。

  少なくとも、QuickScan ユーザーでなければなりません。バージョン 9.0.3 より前の AppScan Enterprise Server に接続されている場合、Enterprise Server 上に独自のユーザー・フォルダーがなければなりません。
- 「パスワード」フィールド: このフィールドは、ご使用の AppScan Enterprise Server 認証方式がユーザー ID とパスワードである場合にのみ使用可能です。Enterprise Console へのログインに使用するパスワードを入力します (入力されたユーザー名のパスワード)。
- 「**Enterprise Console** の **URL**」フィールド: Enterprise Console の Web アプリケーションへのアクセスに使用する URL を入力します。

  この URL の形式は次のとおりです。

  ```
  http(s)://<hostname>:<port>/ase
  ```

  ここで、<hostname> は、Enterprise Console がインストールされているマシンの名前、<port> は、コンソールが実行されているポートです (デフォルトの

<port> は 9443 です)。この URL の例は、https://myhost.mydomain.ibm.com:9443/ase のようになります。

注:

- 「**Enterprise Console** の **URL**」が既に設定されている場合は、このフィールドを変更する必要はありません。
- 「**Enterprise Console** の **URL**」フィールドを設定可能にするには、AppScan Source に「**AppScan Enterprise** 設定の管理」許可を使用してサインインする必要があります。 ユーザー・アカウントと許可については、製品のインフォメーション・センターの『管理』セクション、または「*IBM Security AppScan Source* インストールと管理のガイド」の『*AppScan Source* の管理』セクションを参照してください。
- 「ユーザー **ID**」と「パスワード」は AppScan Source クライアント (AppScan Source for Analysis など) が稼働しているマシンに格納されますが、「**Enterprise Console** の **URL**」は Enterprise Server (これはリモート・マシン上に存在している場合があります) に格納されます。 リモート・マシンから (例えば getaseinfo コマンドを発行して) ユーザー名とパスワードの情報にアクセスすることはできません。
- AppScan Source では、プロキシー設定を使用するように構成された AppScan Enterprise Console インスタンスへの公開はサポートされていません。 プロキシー設定を使用するインスタンスに公開しようとすると、エラーが発生します。

設定が完了した後で、「接続のテスト」をクリックして、Enterprise Console サーバーへの接続が有効であることを確認することを強くお勧めします。

ヒント: 接続テストが失敗した場合は、Enterprise Console サーバーが実行中かどうか、およびブラウザーを使用して製品のコントロール・センター URL にアクセスできるかどうかを確認してください (上記で指定したのと同じ「**Enterprise Console** の **URL**」を使用してください)。

JavaServer Page コンパイル用のアプリケーション・サーバー設定

JavaServer Pages (JSP) を含むアプリケーションをスキャンする場合、JSP コードを分析するためには、AppScan Source 分析エンジンが JSP コードをコンパイルできなければなりません。 JSP プロジェクトを作成するときに、AppScan Source が使用するべき JSP コンパイラーを指定する必要があります (またはデフォルト・コンパイラーを受け入れます。デフォルト・コンパイラーは「Java および JSP」設定ページで設定できます)。 AppScan Source が JSP ファイルをコンパイルできない場合、 アプリケーション・サーバー設定ページを使用して、アプリケーションが使用する JSP コンパイラーを構成してください。

Apache Tomcat バージョン 7 および 8 は、AppScan Source のインストール済み環境に含まれています。「**Tomcat 7**」および「**Tomcat 8**」設定ページが未構成の場合、AppScan Source は、提供されている Tomcat JSP コンパイラー (現在デフォルトとしてマーク) を使用して JSP ファイルをコンパイルします。外部でサポートされている Tomcat コンパイラーを使用したい場合は、Tomcat 設定ページを使用して、ローカルの Tomcat インストール済み環境を示します。

Oracle WebLogic サーバー または WebSphere Application Server を使用する場合は、分析時にアプリケーション・サーバーを JSP コンパイルに使用できるようにするため、適切な設定ページを構成して、アプリケーション・サーバーのローカルのインストール済み環境を示す必要があります (最初にアプリケーション・サーバーを構成しないで JSP プロジェクトを作成する場合は、この時点でアプリケーション・サーバーを構成するようにプロンプトで求められます)。

Tomcat

このトピックでは、AppScan Source が、AppScan Source に付属するアプリケーション・サーバー以外の Apache Tomcat アプリケーション・サーバーを参照するように構成するために必要な設定について説明します。

Apache Tomcat バージョン 7 および 8 は、AppScan Source のインストール済み環境に含まれています。「**Tomcat 7**」および「**Tomcat 8**」設定ページが未構成の場合、AppScan Source は、提供されている Tomcat JSP コンパイラー (現在デフォルトとしてマーク) を使用して JSP ファイルをコンパイルします。外部でサポートされている Tomcat コンパイラーを使用したい場合は、Tomcat 設定ページを使用して、ローカルの Tomcat インストール済み環境を示します。

外部のサポートされる Tomcat コンパイラーを使用する場合は、適切な設定ページに移動して、アプリケーション・サーバーのインストール・ディレクトリーを設定します。インストール・ディレクトリーを指定すると、AppScan Source は、プロジェクトを構成するときにすべてのアプリケーション・サーバー依存関係を自動的に検出できます。

WebLogic 11 および 12

このトピックでは、AppScan Source が Oracle WebLogic サーバー を参照するように構成するために必要な設定について説明します。

WebLogic 設定ページでは、サーバー・インストール・ディレクトリーを指定します。詳細構成オプションも設定できます。インストール・ディレクトリーを指定すると、AppScan Source は、プロジェクトを構成するときにすべてのアプリケーション・サーバー依存関係を自動的に検出できます。

AppScan Source が WebLogic インストール・ディレクトリー、WebLogic JAR ファイル、および JavaServer Page (JSP) コンパイラー・オプションを参照するように構成します。

デフォルトの WebLogic JSP コンパイラー・オプションを変更したり、weblogic.jar ファイルを見つけたりする必要がある場合にのみ、「詳細構成オプションの有効化」チェック・ボックスを選択します。デフォルトの WebLogic JSP コンパイラー・オプションは次のとおりです。

```
%JSP_JVM_OPTIONS%
 -Dcom.sun.xml.namespace.QName.useCompatibleSerialVersionUID=1.0
 -classpath
%JSP_COMPILER_CLASSPATH% weblogic.jspc
%JSP_OPTIONS% -verboseJspc -package
%PACKAGE_NAME% -linenumbers -g -debug -keepgenerated -compiler
%JAVAC_PATH% -webapp
%WEB_CONTEXT_ROOT_PATH% -d
%OUTPUT_PATH%
```

WebSphere Application Server

このトピックでは、AppScan Source が JSP コンパイルの目的で WebSphere Application Server を参照するように構成するために必要な設定について説明します。

AppScan Source でサポートされる WebSphere Application Server のバージョンについては、http://www.ibm.com/support/docview.wss?uid=swg27027486 を参照してください。

WebSphere Application Server 設定ページでは、サーバー・インストール・ディレクトリーを指定します。詳細構成オプションも設定できます。インストール・ディレクトリーを指定すると、AppScan Source が WebSphere Application Server JSP コンパイラーを検出して使用できるようになります。

AppScan Source が WebSphere Application Server インストール・ディレクトリーを参照するように構成します。さらに、詳細構成オプションでは、WebSphere Application Server JSP コンパイラーのコマンド行およびクラスパスを設定できます。

「詳細構成オプションの有効化」チェック・ボックスは、WebSphere Application Server JSP コマンド行をカスタマイズするか、デフォルトの WebSphere Application Server クラスパス以外のクラスパスを指定 (クラスパスに WebSphere Application Server バージョン 6.1 のすべてのアプリケーションによって使用される追加の JAR を含める場合にこの設定を変更) する場合にのみ選択します。

デフォルトの WebSphere Application Server JSP コンパイラーのコマンド行オプションは、以下のとおりです。

```
%CMD_EXE% %CMD_ARGS%
'%FILE(%%JSP_COMPILER_INSTALL_DIR%/bin/JspBatchCompiler%BAT%%)%'
-response.file
'%TMP_FILE(%-keepgenerated=true -recurse=true -useFullPackageNames=true
-verbose=false -createDebugClassfiles=true -jsp.file.extensions=%WEB_EXTS%
-javaEncoding=%ENCODING%
%JSP_OPTIONS% %QUOTE%-war.path=%WEB_CONTEXT_ROOT_PATH%%QUOTE%
%QUOTE%-filename=%RELATIVE_FILENAME_NO_QUOTE% %QUOTE%  %)%'
```

変数の定義

評価またはバンドルを保存するとき、または評価を公開するときに、絶対パスを置換する変数を作成するように AppScan Source for Analysis から提示されることがあります (変数がない場合、AppScan Source for Analysis は、ソース・ファイルなどの項目を参照するための絶対パスを評価ファイルに書き込みます)。絶対パスに代わる変数を構成すると、複数のコンピューターでの評価の共有が容易になります。 評価を共有する場合は、変数を使用することをお勧めします。

このタスクについて

保存アクションまたは公開アクションを開始する前に変数を作成することができます。その場合は、このトピックの以下の説明に従ってください。あるいは、 154 ページの『公開時および保存時の変数の定義』の手順に従うことにより、保存アクションまたは公開アクションの開始後に変数を作成することもできます。

評価を共有する際の変数の使用例については、 155 ページの『例: 変数の定義』を参照してください。

手順

1. メインメニューで、「編集」 > 「設定」を選択します。 「設定」ダイアログ・ボックスで、「変数の変更」を選択します。
2. 「変数の変更」設定ページで、「変数の追加」 ボタンをクリックします。
3. 変数の名前を入力し、変数で置換するファイルの場所を参照します (作成した変数には、AppScan Source for Analysis によって前後にパーセント記号 (%) が挿入されます)。
4. 評価内の他のすべての参照項目について、上記の手順を繰り返します (例えば、複数の場所のソースが評価内で参照されている場合は、それぞれの場所について変数を追加します)。
5. 設定ページでは、「変数の変更」ボタンを使用して変数を編集でき、「変数の削除」ボタンを使用して変数を削除できます。
6. 変数の定義が完了したら、「**OK**」をクリックします。

設定による障害追跡の有効化

「障害追跡システム」の設定で、障害追跡システムへの検出結果の送信を有効にし、障害の送信方法を指定することができます。

「障害追跡システム」設定ページの「全般」タブを使用して、AppScan Source における障害追跡システムの統合機能を有効または無効にします。「障害追跡システムの統合を有効にする」チェック・ボックスを選択すると、評価結果に対して「障害の送信」コンテキスト・メニュー・アクションを使用できるようになります。 また、「全般」タブを使用することで、障害の送信時にどの障害追跡システムを利用できるようにするかを個別に制御できます。

サポート対象の障害追跡システムに対して指定可能な設定については、以下のヘルプ・トピックを参照してください。

- 『Rational ClearQuest の設定』
- 111 ページの『Quality Center の設定』
- 113 ページの『Rational Team Concert の設定』
- 115 ページの『Team Foundation Server の設定』

Rational ClearQuest の設定

Rational ClearQuest の設定を行うには、必要な Rational ClearQuest の設定が Rational ClearQuest の管理者によって提供されている必要があります。設定は、それぞれの Rational ClearQuest 環境に固有です。

注**:** Rational ClearQuest バージョン 8.0 と統合する場合、Rational ClearQuest スキーマに、**DefectTracking** 事前定義スキーマで使用可能なフィールドが含まれている必要があります。

データベース・セット

1 つ以上の障害データベースの集合。

```
Linux default = Connection Name,
Windows default = Database Set
```

データベース名

障害の送信先データベースの名前。

データベース・ユーザー名

デフォルトの Rational ClearQuest データベース・ユーザー名。

CQPerl 実行可能プログラムの位置

ローカル・コンピューター上の Rational ClearQuest CQPerl 実行可能プログラムの位置。 指定されたデフォルトの位置は、デフォルトの Rational ClearQuest インストール位置にマップされます。

障害レコードのエンティティー

障害オブジェクト用に使用するために Rational ClearQuest インストール済み環境によって構成されたエンティティー (データベース・オブジェクト)。

デフォルトのエンティティーは「**Defect**」です。

レコードの「説明」フィールド

デフォルトの説明は「**Description**」です。

レコードの「ヘッドライン」フィールド

デフォルトのヘッドラインは「**Headline**」です。

検出結果ごとに単一の障害

複数の検出結果を単一の障害として送信するか、複数の障害として送信するかを選択します。 障害を作成するときに送信方式を変更できます。

Quality Center の設定

最初に全般設定で障害追跡システムとして HP Quality Center を有効にしてから、「Quality Center」タブで個別設定を指定する必要があります。

サーバー URL

Quality Center サーバーの URL (`http://<hostname>:<port>/qcbin/` や `https://<hostname>:<port>/qcbin/` など)。

ユーザー名 (オプション)

Quality Center にログインするユーザー名

パスワード (オプション)

ユーザー名を入力した場合は、対応するパスワードを入力してください。

ドメイン

接続先の Quality Center ドメイン。

プロジェクト

接続先の Quality Center プロジェクト

自動ログイン

true の場合、AppScan Source は、検出結果の送信時にログイン情報を要求するプロンプトを表示せず、「設定」で指定されたデフォルトの資格情報を使用してログインします。 false の場合、検出結果を Quality Center に送信するごとにログインする必要があります。

自動送信

true の場合、新規障害を送信するためのダイアログ・ボックスが検出結果の送信時に表示されません。 AppScan Source for Analysis は、「設定」で指定された「デフォルトの障害プロパティー」を使用します。 false の場合、検出結果の送信時に障害情報 (重大度、優先順位、障害タイプ、状態など) の入力を要求するプロンプトが表示されます。

以前に送信した検出結果の再送信

Quality Center に送信された検出結果には、Quality Center 障害情報 (障害 ID、送信ユーザー、および送信日付) のタグが付けられます。 デフォルトの場合、AppScan Source は、同じ検出結果を 2 回以上再送信することはありません。 これにより、複数の検出結果を Quality Center にディスパッチしても、新規の検出結果のみが Quality Center データベース内に入力されます。 選択した場合 (true の場合)、以前に送信した検出結果を Quality Center に再送信できます。

各検出結果を個別のバグとして送信

複数の検出結果を 1 回の操作で送信するときには、すべての検出結果を単一の Quality Center 障害として送信するか、個別の AppScan Source 検出結果ごとに別個の Quality Center 障害として送信するかを選択できます。このチェック・ボックスを選択すると、フラグは true に設定され、個別の検出結果ごとに別個の Quality Center 障害が作成されます。 フラグを false に設定すると、すべての検出結果を一括送信の一部として送信するための単一の Quality Center 障害が作成されます。

バグ概要の自動生成

true の場合、AppScan Source は、Quality Center に送信するための障害の概要を自動的に生成します。 この概要は、障害に含まれる検出結果の数および検出結果のタイプ (`Validation.Required` など) を示します。

false の場合、新規障害の作成時に開くダイアログ・ボックスで障害を送信するときに、「概要」フィールドが表示されて入力できるようになります。

バグ・フィールドの自動ロード

デフォルト設定は true です。 このチェック・ボックスを選択すると、AppScan Source は、Quality Center 内の現在のユーザーおよびグループ設定に基づいて、Quality Center データベースから障害フィールド定義を自動的にロードします。false の場合、 AppScan Source は、新規障害の作成時に開くダイアログ・ボックスに、Quality Center からの障害フィールドを表示しません。

デフォルトの障害プロパティー

さまざまな Quality Center 障害属性のデフォルト値を設定するには、Quality Center の設定タブの「デフォルトの障害プロパティー」をクリックします。デフォルト値は、送信時に「新規障害」ダイアログ・ボックスに事前に取り込まれるか、「自動送信」設定が選択されている場合には Quality Center に自動的に送信されます。

注**:** 「バグ・フィールドの自動ロード」が選択されている場合、「問題プロパティー」ダイアログ・ボックスが表示されるごとに、障害のプロパティーおよびその使用可能な値が Quality Center から動的に取得されます。したがって、Quality Center データベースに追加された新規フィールドおよび値は、AppScan Source for Analysis 内に自動的に表示されます。「問題プロパティー」ダイアログ・ボックスを開き、Quality Center 情報を取り込むには、サーバー、ログイン、および接続の有効な情報が必要です。

Quality Center の障害フィールドのカスタマイズ

構成ファイルを通じて、「新規障害」ダイアログ・ボックス内のフィールドおよびこれらのフィールドの間の相互作用をカスタマイズできます。カスタマイズのサンプルおよび追加説明が記載されているサンプル構成ファイルは、`<data_dir>¥config¥qc.dts` (`<data_dir>` は、ご使用の AppScan Source プログラム・データの場所です。詳しくは、 360 ページの『インストールとユーザー・データ・ファイルの場所』 を参照してください。) にあります。これらのカスタマイズを使用して、「新規障害」ダイアログ・ボックス内で Quality Center Workflow スクリプト・ロジックを直接モデル化できます。

以下のカスタマイズを選択できます。

- カスタム・フィールドまたは欠落フィールド (あるいはその両方) を表示します
- フィールドを常に強制的に表示します (Quality Center 設定に対し優先します)
- 他のフィールドの選択に基づいて、フィールドに必要な状態を更新します
- 別のフィールド内のリスト・ボックス選択に基づいて、フィールドのリスト・ボックス・オプションを動的に更新します

Rational Team Concert の設定

「Rational Team Concert」設定タブでは、Rational Team Concert サーバーへの接続と、ワークアイテム属性の値を構成できます。

接続情報を入力して正常にログインすると、1 つ以上のプロジェクト・エリアに接続できるようになります。プロジェクト・エリアごとに独自の属性事前設定値を構成できます。

注: Rational Team Concert に接続 (設定を構成するか、または障害を送信することにより接続) すると、SSL 証明書を受け入れるよう指示するプロンプトが出されます。詳しくは、『Rational Team Concert の SSL 証明書』を参照してください。

特定のプロジェクト・エリアの属性値を構成するには、そのプロジェクト・エリアを選択して「構成」を選択します。 「構成」ダイアログ・ボックスで、属性値をハードコーディング値に設定したり、選択した検出結果を参照する変数に設定したりすることができます。例えば、属性値で {Finding.fileName} を使用すると、障害の送信時に検出結果の実際のソース・コード・ファイル名に置き換えられます。 それらの変数をサポートしている属性値ではコンテンツ・アシスト (<Ctrl>+<Space>) が提供されます。 チームではこれらの構成を共有することをお勧めします。それには、「Rational Team Concert」メイン設定ページにある「インポート」ボタンと「エクスポート」ボタンを使用します。

Rational Team Concert の SSL 証明書

Rational Team Concert サーバーのインストール時に、有効な SSL 証明書を使用するように構成する必要があります。この構成を行わないと、(設定の構成時や障害の送信時に) サーバーにログインするときに、信頼できない接続を通知するメッセージを受け取ります。このトピックでは、Rational Team Concert SSL 証明書の考慮事項について簡単に説明します。

SSL 証明書の保管場所

永続的に受け入れられた証明書は、<user_home>/.jazzcerts (<user_home> は、ご使用のオペレーティング・システムのホーム・ディレクトリーです (例えば Windows では、ディレクトリーは C:¥Documents and Settings¥Administrator¥ などになります)。) に保管されます。 <user_home>/.jazzcerts を削除すると、AppScan Source および Rational Team Concert クライアント用に保管されているすべての証明書が削除されます。

Rational Team Concert クライアントと共有される SSL 証明書

AppScan Source は、その証明書ストアを Rational Team Concert クライアントと共有します。 Rational Team Concert クライアントを使用して証明書を永続的に受け入れると、その証明書は AppScan Source によって再使用されます (AppScan Source で証明書の受け入れを要求するプロンプトが表示されなくなります)。同様に、AppScan Source で証明書を永続的に受け入れると、その証明書が Rational Team Concert クライアントによって再使用されます。

AppScan Source for Analysis から障害追跡を行う場合の Rational Team Concert サーバー名変更の考慮事項

AppScan Source for Analysis で Rational Team Concert の障害追跡を有効に設定した状態で Rational Team Concert サーバーの名前を変更すると、そのサーバーのプロジェクト・エリアの既存の構成が AppScan Source for Analysis で使用できな

くなります。この場合、新しいリポジトリー URI からサーバーに接続して、障害追跡システムの設定で構成を再作成する必要があります。

Team Foundation Server の設定

「Team Foundation Server」設定タブでは、Microsoft Team Foundation Server への接続と、ワークアイテム・フィールドの値を構成できます。

接続情報を入力して正常にログインすると、1 つ以上のプロジェクトに接続できるようになります。

注: Team Foundation Server 2010 へのログインを構成するときに、接続先となるチーム・プロジェクトのコレクションを「サーバー URL」に含める必要があります。 例えば、`http://myserver:8080/tfs/DefaultCollection` のようにします。

プロジェクトごとに独自のフィールド事前設定値を構成できます。

特定のプロジェクトのフィールド値を構成するには、そのプロジェクトを選択して「構成」を選択します。 「構成」ダイアログ・ボックスで、フィールド値をハードコーディング値に設定したり、選択した検出結果を参照する変数に設定したりすることができます。 例えば、フィールド値で `{Finding.fileName}` を使用すると、障害の送信時に検出結果の実際のソース・コード・ファイル名に置き換えられます。 それらの変数をサポートしているフィールドではコンテンツ・アシスト (`<Ctrl>+<Space>`) が提供されます。

チームではこれらの構成を共有することをお勧めします。それには、「Team Foundation Server」メイン設定ページにある「インポート」ボタンと「エクスポート」ボタンを使用します。

Eclipse ワークスペース・インポーター: Eclipse または Rational Application Developer for WebSphere Software (RAD) の設定構成

AppScan Source for Analysis のインストールには、デフォルトの Eclipse インポーターが用意されています。 このインポーターは、Eclipse と JRE の位置を識別します。 デフォルトの Eclipse インポーターでワークスペースをインポートできない場合は、新しい Eclipse インポーターの作成が必要になることがあります。

始める前に

各インポーター構成は、Eclipse または Rational Application Developer for WebSphere Software (RAD) のインストール済み環境を表します。これらの構成を使用して既存のワークスペースとプロジェクトを AppScan Source for Analysis にインポートするには、AppScan Source for Development のプラグインも Eclipse 環境にインストールしなければならない場合があります。

RAD ワークスペースを追加する前に、ワークスペース・タイプの構成を作成する必要があります。

手順

1. AppScan Source for Analysis のワークベンチのメインメニューで、「編集」 > 「設定」を選択します。

2. 「**Eclipse** ワークスペース・インポーター」を選択します。
3. 「新しい構成の作成」をクリックし、「新しいインポート構成」ダイアログ・ボックスの以下のフィールドに入力して新しい構成を作成します。
   - 製品: 該当する製品を選択します。

     注**:** ワークスペースの作成に使用した製品を選択できない場合は、 52 ページの『Eclipse または Application Developer の更新』に概要が説明されている構成手順が完了していることを確認してから、ワークスペース・インポーターを作成するようにしてください。
   - 名前: インポーターの名前。
   - 位置: Eclipse インストール済み環境の基本ディレクトリーへのパス。
   - **JRE** の位置: Java ランタイム環境 (JRE) のルート・ディレクトリーへのパス。 `<install_dir>¥JDKS` (`<install_dir>` は AppScan Source インストール済み環境がある場所です) にある JDK、またはその他の優先 JDK を使用します。
4. 「**OK**」をクリックします。
5. インポーターをデフォルトとして特定するには、そのインポーターを選択して「選択した構成をデフォルトにする」をクリックします。これにより、インポーターの「デフォルト」列にアイコンが表示されます。

E メール

障害として検討中の検出結果を送信するために使用される E メール設定を構成します。

- 宛先アドレス: 受信者の E メール・アドレス。「検出結果の E メール送信」ダイアログ・ボックスの「宛先」フィールドには、デフォルトとして、この E メール・アドレスが指定されますが、これは、E メールの作成時に容易に変更できます。
- 差出人アドレス: 送信者の E メール・アドレス。

  注**:** 受信側のメール・クライアントが E メールをスパムとして処理することがないよう、正しい E メール・アドレスを指定することをお勧めします。
- メール・サーバー: `mail.myexample.com` として構成された SMTP メール・サーバー。

  重要**:** システム管理者に依頼して、正しいメール・サーバー情報が設定されていることを確認してください。

Java および JavaServer Pages

この設定ページを使用して、スキャンに使用される Java Development Kit (JDK) の追加、変更、または削除を行い、デフォルトの JDK を設定します。また、このページを使用して、デフォルトの JavaServer Page (JSP) コンパイラーを設定します。

デフォルト

スキャンで使用される JDK の位置を示します。 プロジェクトで明示的な JDK が指定されない場合、スキャンではデフォルトの JDK パスが使用されます。 JDK をデフォルトとして設定するには、示された JDK 名を右クリックし、「デフォルト **JDK** の設定」をクリックします。表内にデフォルトのアイコンが表示されて、現在のデフォルト JDK を示します。

注**:** 製品に付属の JSP プロジェクトのデフォルト・コンパイラーは、Tomcat 7 です。これには、Java バージョン 1.6 以上が必要です。 **Tomcat 7** をデフォルトのまま使用している場合、古い JDK を選択すると、以下のスキャン中のコンパイル・エラーが発生します。

JDK 名とパス

JDK の名前と位置を示します。

JSP プロジェクトのデフォルト・コンパイラー

製品に付属の Tomcat 7 がデフォルトの JSP コンパイラー設定です。 AppScan Source for Analysis にサポートされるコンパイラーについて詳しくは、http://www.ibm.com/support/docview.wss?uid=swg27027486 を参照してください。

ナレッジベース・データベースの記事

「ナレッジベース・データベースの記事」設定ページを使用して、AppScan Source セキュリティー・ナレッジ・データベースの記事が含まれているロケーションを設定できます。

ページには、記事が含まれているディレクトリーがリストされます。 ディレクトリーを追加するには、「コンテンツ・ディレクトリーの追加」をクリックして、記事の場所を参照します。 ディレクトリーを削除するには、削除するディレクトリーを選択して、「削除」をクリックします。

プロジェクト・ファイル拡張子

各プロジェクト・タイプの有効なグローバル・ファイル拡張子を構成または追加するか、スキャンに含める拡張子を変更するか、スキャンから拡張子を除外するか、Web ファイルとして拡張子を指定します。

使用可能な言語またはプロジェクト・タイプ
(Java、JavaScript、ASP、Perl、PHP、ColdFusion、PBSA (パターン・ベースのプロジェクト・タイプの場合)、COBOL、PL/SQL、T-SQL、VB.NET、.Net Assembly、VB、C/C++、ASP .NET 1.x、ASP .NET 2.x、WSDL、および C#) ごとにタブ・ページが表示されます。新規の拡張子を追加するときには、その新規の拡張子が付いたファイルに対して、スキャンできるか、Web ファイルとして指定できるか、または除外の操作を実行できるかどうかを確認します。

このページの設定値はグローバルです。個々のプロジェクトのファイル拡張子を設定するには、選択されたプロジェクトの「プロパティー」ビューの 283 ページの『ファイル拡張子』タブを使用します。

ファイル拡張子の設定

表 *10.* ファイル拡張子の設定

設定	説明	使用例
「スキャン」または「評価」	指定された拡張子を持つファイルを完全分析に含みます。	• Java プロジェクト用の `.xxx` 拡張子が作成され、「スキャン」または「評価」のマークが付けられると、その拡張子を持つファイルはコンパイルされ、スキャンされます。 • ファイルのコンパイルとスキャンを行わない場合 (C++ のヘッダー・ファイルなど)、そのファイルはプロジェクトの一部にすることができますが、「スキャン」または「評価」のマークはつきません。これらのファイルは、プロジェクトに含まれ、パターン・ベースの分析時に検索されます。
Web ファイル	JSP コンパイル用に指定の拡張子を持つファイルにマークを付けます。この設定により、AppScan Source は Web ソースを非 Web ソースと分離することができます。	Java プロジェクト用の `.yyy` 拡張子が作成され、「**Web** ファイル」のマークが付けられると、その拡張子を持つファイルは、プロジェクトで Web ソースとして調整されます。AppScan Source が分析の準備をすると、これらのファイルは分析のためにクラスにプリコンパイルされます。
除外	指定の拡張子を持つファイル用に、プロジェクトでソース・ファイルを作成しません。この拡張子を持つファイルはスキャンされません。	コンパイルのためにプロジェクトに必要であるものの、分析に組み込む必要がないファイルの `.zzz` 拡張子を作成します。

第 4 章 ソース・コードのスキャンおよび評価の管理

このセクションでは、ソース・コードのスキャンおよび評価の管理の方法について説明します。

アプリケーションおよびプロジェクトを構成するか、アプリケーション・ディスカバリー・アシスタントを使用してアプリケーションおよびプロジェクトを作成すると、ソース・コードをスキャンする準備が整いました。 スキャンの結果 (評価) を保存または公開できます。 保存済みの評価 は、ローカルに保存されたスキャン結果のファイルであり、公開したり、後から追加トリアージのために開いたり、AppScan Source for Development 内で開いたりすることができます。公開された評価 は、AppScan Enterprise Server に保存されたスキャン結果です。

以下の 2 つのビューで評価を管理します。

- マイ評価
- 公開された評価

注: 評価を保存する、公開する、または開く処理中には、ステータス・バーに進行状況が表示されます。

ソース・コードのスキャン

このタスクでは、スキャンを起動するためのさまざまな方法について説明します。

このタスクについて

さまざまなレベルでスキャンを実行できます (すべてのアプリケーション、1 つ以上のアプリケーション、1 つ以上のプロジェクト、1 つ以上のファイル)。スキャンを完了した後、スキャンの評価が開いている場合は、もう一度スキャンできます。

- 120 ページの『すべてのアプリケーションのスキャン』
- 120 ページの『1 つ以上のアプリケーションのスキャン』
- 120 ページの『1 つ以上のプロジェクトのスキャン』
- 121 ページの『1 つ以上のファイルのスキャン』
- 121 ページの『コードの再スキャン』

121 ページの『スキャンに関する考慮事項』を参照して、オペレーティング・システム固有の考慮事項、言語固有の考慮事項、またはその他、スキャンに影響する可能性のある制限事項を確認してください。

スキャン構成は、スキャン時に常に使用されます。デフォルト・スキャン構成を設定し、後でそのスキャン構成を削除した場合、スキャン時には、標準装備のスキャン構成である「通常のスキャン」が確認なしで使用されます。スキャン構成について詳しくは、 123 ページの『スキャン構成の管理』と、以下のスキャン・オプションの説明を参照してください。

すべてのアプリケーションのスキャン

手順

以下のアクションのいずれかを実行します。

1. メイン・ワークベンチ・メニューで、「スキャン」 > 「すべてスキャン」を選択します。 デフォルトのスキャン構成を使用してスキャンが実行されます。
2. 「エクスプローラー」ビューで、以下のような操作を行います。
   - 「すべてのアプリケーション」を右クリックし、メニューから「すべてのアプリケーションをスキャン」を選択します。 デフォルトのスキャン構成を使用してスキャンが実行されます。
   - スキャンに別のスキャン構成を使用するには、「すべてのアプリケーション」を右クリックし、メニューから「すべてのアプリケーションのスキャン」を選択します。 使用するスキャン構成を選択します。あるいは、別のデフォルトのスキャン構成を設定するには、「構成の編集」アクションを選択します (「スキャン構成」ビューで、デフォルトとして設定する構成を選択し、「デフォルトとして選択」をクリックします)。

1 つ以上のアプリケーションのスキャン

手順

1. 「エクスプローラー」ビューで、1 つ以上のアプリケーションを選択します。
2. 以下のアクションのいずれかを実行します。
   a. メイン・ワークベンチ・メニューで、「スキャン」 > 「選択項目のスキャン」を選択します。 デフォルトのスキャン構成を使用してスキャンが実行されます。
   b. 「エクスプローラー」ビューで、以下のような操作を行います。
      - 選択項目を右クリックして、メニューから「アプリケーションのスキャン」を選択します。 デフォルトのスキャン構成を使用してスキャンが実行されます。
      - スキャンに別のスキャン構成を使用するには、選択項目を右クリックして、メニューから「アプリケーションのスキャン」を選択します。 使用するスキャン構成を選択します。あるいは、別のデフォルトのスキャン構成を設定するには、「構成の編集」アクションを選択します (「スキャン構成」ビューで、デフォルトとして設定する構成を選択し、「デフォルトとして選択」をクリックします)。

1 つ以上のプロジェクトのスキャン

手順

1. 「エクスプローラー」ビューで、1 つ以上のプロジェクトを選択します。
2. 以下のアクションのいずれかを実行します。
   a. メイン・ワークベンチ・メニューで、「スキャン」 > 「選択項目のスキャン」を選択します。デフォルトのスキャン構成を使用してスキャンが実行されます。
   b. 「エクスプローラー」ビューで、以下のような操作を行います。

- 選択項目を右クリックして、メニューから「プロジェクトのスキャン」を選択します。 デフォルトのスキャン構成を使用してスキャンが実行されます。
- スキャンに別のスキャン構成を使用するには、選択項目を右クリックして、メニューから「プロジェクトのスキャン」を選択します。 使用するスキャン構成を選択します。あるいは、別のデフォルトのスキャン構成を設定するには、「構成の編集」アクションを選択します (「スキャン構成」ビューで、デフォルトとして設定する構成を選択し、「デフォルトとして選択」をクリックします)。

1 つ以上のファイルのスキャン

手順

1. 「エクスプローラー」ビューで、1 つ以上のファイルを選択します。
2. 以下のアクションのいずれかを実行します。
   a. メイン・ワークベンチ・メニューで、「スキャン」 > 「選択項目のスキャン」を選択します。デフォルトのスキャン構成を使用してスキャンが実行されます。
   b. 「エクスプローラー」ビューで、以下のような操作を行います。
      - 選択項目を右クリックして、メニューから「ファイルのスキャン」を選択します。 デフォルトのスキャン構成を使用してスキャンが実行されます。
      - スキャンに別のスキャン構成を使用するには、選択項目を右クリックして、メニューから「ファイルのスキャン」を選択します。 使用するスキャン構成を選択します。あるいは、別のデフォルトのスキャン構成を設定するには、「構成の編集」アクションを選択します (「スキャン構成」ビューで、デフォルトとして設定する構成を選択し、「デフォルトとして選択」をクリックします)。

コードの再スキャン

手順

現在のターゲットを再スキャンするには、メインメニューから「スキャン」 > 「再スキャン」を選択します。 項目 (選択した項目) のスキャンに使用した最後のスキャン構成が、再度スキャンに使用されます。

- 前のスキャンにデフォルトのスキャン構成が使用されていて、新しいデフォルトのスキャン構成が設定されている場合は、その新しいデフォルトのスキャン構成がスキャンに使用されます。
- 前のスキャンにデフォルト以外のスキャン構成が使用されていた場合は、そのスキャン構成がスキャンに使用されます。 デフォルト以外のスキャン構成が変更されていて、前のスキャン以後に保存されている場合は、変更されたスキャン構成が使用されます。

スキャンに関する考慮事項

このトピックでは、スキャンに影響を与える可能性のある制限事項と考慮事項について説明します。

- 『全般』
- 『Windows』
- 『Linux』
- 『Java』

全般

制約事項: 複数のアプリケーションまたはプロジェクトをスキャンする場合、各スキャン対象項目の評価を含む親ノードが「自分の評価」ビューに作成されます。この場合、個々の子評価を管理することはできません (例えば、子評価を個々に削除したり、公開したりすることはできません)。複数のアプリケーションまたはプロジェクトを同時にスキャンする場合は、評価をグループ (親ノード) としてのみ管理することができます。

重要: 開発環境内に依存関係を持つ AppScan Source プロジェクト (例えば、IBM MobileFirst Platform プロジェクト) を処理する場合は、必ず、プロジェクトをインポートする前に開発環境でビルドしてください。プロジェクトをインポートした後、その中のファイルを変更した場合は、AppScan Source でスキャンする前に開発環境で再ビルドしてください (そうしないと、ファイルに対する変更は AppScan Source によって無視されます)。

Windows

Microsoft .NET ファイル (`.cs` や `.vbnet` など) の場合、個々のファイルまたは複数選択したファイルをスキャンする機能は使用できません。

Linux

AppScan Source for Analysis クライアントは Eclipse 上に構築されています。Linux の場合、Eclipse は、ブラウザー・ベースのコンテンツをレンダリングするためにサード・パーティー・コンポーネントをインストールする必要があります。 このコンポーネントがないと、AppScan Source for Analysis は、ログイン後にハングしたり、製品使用中に障害が発生したりするなどの症状を示す可能性があります。詳しくは、 137 ページの『Linux 上の AppScan Source for Analysis でブラウザー・ベースのコンテンツを使用可能にする』を参照してください。

Java

ヒント: Java をスキャンしており、Java プロジェクトに欠落依存関係がある場合、AppScan Source は、依存関係が提供するはずだった部分を合成することで、トレースを作成します。 この合成には `.jar` ファイル内の情報が正確に反映されない場合があります。合成を制限することにより検出結果の精度を向上するために、欠落している依存関係を以下のように指定できます。

1. スキャン後に、`<data_dir>¥logs¥scanner_exceptions.log` (`<data_dir>` は、ご使用の AppScan Source プログラム・データの場所です。詳しくは、 360 ページの『インストールとユーザー・データ・ファイルの場所』 を参照してください。)を開いて、AppScan Source が欠落依存関係を報告しているかどうかを確認します。
2. 依存関係を組み込むようにプロジェクト・プロパティーを変更します。 そのためには、 93 ページの『アプリケーションおよびプロジェクトのプロパティーの

変更』 の指示に従い、「**JSP** プロジェクト依存関係」または「プロジェクト依存関係」タブに依存関係を指定して保存します。

3. プロジェクトを再スキャンします。

注**:** デフォルトでは、AppScan Source は、依存関係の欠落やコンパイル・エラーについて Java ファイルや Java バイトコードをスキャンします。 これらの設定は、以下のように変更できます。

1. テキスト・エディターで <data_dir>¥config¥scan.ozsettings を開きます。
2. コンパイル・エラーの設定を変更するには、ファイル内で compile_java_sources_with_errors を見つけます。この設定は、以下の例のようになります。

```
<Setting
  name="compile_java_sources_with_errors"
  value="true"
  default_value="true"
  type="bool"
  hidden="true"
  display_name="compile_java_sources_with_errors"
  description="Attempt to scan java code with compilation errors."
/>
```

3. 欠落依存関係の設定を変更するには、ファイル内で scan_java_bytecode_without_dependencies を見つけます。この設定は、以下の例のようになります。

```
<Setting
  name="scan_java_bytecode_without_dependencies"
  value="true"
  default_value="true"
  type="bool"
  hidden="true"
  display_name="scan_java_bytecode_without_dependencies"
  description="Scans Java bytecode even when some of
    the dependencies are missing by artificially
    synthesizing the unresolved symbols."
/>
```

4. この設定では、value 属性を変更します。属性が true に設定されている場合、この設定はオンになります。 コンパイル・エラーの設定が false になっている場合、AppScan Source は、スキャン中にコンパイル・エラーのある Java コードをスキップします。 欠落依存関係の設定が false になっている場合、AppScan Source は、欠落依存関係が存在するかどうかについて、Java バイトコードをスキャンしません。
5. この設定の変更後、ファイルを保存してAppScan Source を始動または再始動します。

スキャン構成の管理

スキャン構成は、スキャンの起動時に使用されます。 スキャン構成では、スキャン時に使用するソース・ルールを指定できます。スキャン構成で設定を行うと、良好なスキャン結果が得られることが多く、また、これらの設定を保存することができるため、スキャンを容易に、しかも短時間で行うことができます。

このタスクについて

このタスクでは、スキャン構成の管理に関係するステップについて説明します。

- 『スキャン構成の作成』
- 128 ページの『スキャン構成の変更』
- 128 ページの『スキャン構成の削除』
- 128 ページの『スキャン構成の共有と共有構成の操作』
- 129 ページの『スキャン構成をデフォルトとして設定する』
- 129 ページの『標準装備のスキャン構成』

スキャン構成は、 129 ページの『「スキャン構成」ビュー』で管理されます。 このビューは、メインメニュー・バーから「表示」 > 「スキャン構成」を選択するか、「エクスプローラー」ビューの「構成の編集」アクションを選択することで開くことができます。

スキャン構成を設定したら、AppScan Source for Analysis でのスキャンの起動時にスキャン構成を使用することができます (詳しくは、 119 ページの『ソース・コードのスキャン』を参照してください)。 また、AppScan Source for Automation、AppScan Source for Development、および AppScan Source コマンド行インターフェース (CLI) でのスキャンの起動時にスキャン構成を使用することもできます。

スキャン構成の作成

手順

1. 以下のアクションのいずれかを実行します。
   a. 「スキャン構成」ビューの「新規」ボタンをクリックします。
   b. リストから既存の構成を選択して「複写」をクリックします。 これにより、元のスキャン構成の設定に基づいてスキャン構成が作成されます。これを変更して、新しい構成として保存できます。
2. 「全般」タブの「基本情報」セクションで、以下を行います。
   a. 「名前」フィールドに、構成の固有の名前を入力します。スキャン構成では、固有の名前を指定することが唯一の必須設定であり、その他の設定はすべてオプションである点を覚えておいてください。
   b. オプション: スキャン構成の説明を入力します。
3. オプション: 「全般」タブの「フィルター情報」セクションを使用して、スキャンのフィルターを設定します。フィルターについては、 163 ページの『フィルターを使用したトリアージ』を参照してください。 このセクションでは、スキャン構成を使用すると必ずスキャンに適用されるフィルターを 1 つ以上選択できます。フィルターを選択するときは、AppScan Source 事前定義フィルターまたは 共有フィルター、あるいは自分で作成したフィルターを選択できます。 このセクションでは以下を行います。
   a. 「追加」をクリックしてから、「フィルターの選択」ダイアログ・ボックスで、追加するフィルターを 1 つ以上選択します。フィルターを選択すると、その特徴がダイアログ・ボックスの右側に読み取り専用で表示されます。「**OK**」をクリックして、その 1 つ以上のフィルターをスキャン構成に追加します。

注:

- フィルターを反転してスキャン構成に適用するには、「**OK**」をクリックする前に「反転フィルター」チェック・ボックスを選択します。
- 「フィルターの選択」ダイアログ・ボックスでは、追加するフィルターの複数選択が可能です。複数選択時に「反転フィルター」チェック・ボックスを選択すると、選択したすべてのフィルターが反転してスキャン構成に追加されます。

「フィルターの選択」ダイアログ・ボックスを終了すると、追加したフィルターがリストに表示され、そのフィルターが反転されているかどうかが「反転」列に示されます。

b. 追加したフィルターを削除するには、フィルターを選択または複数選択し、「削除」をクリックします。

c. 除外フィルターには、脆弱性タイプ、アプリケーション・プログラミング・インターフェース (API)、ファイル、ディレクトリー、プロジェクト、あるいはトレースを検出結果から除去する対象のルールが含まれます。スキャン構成に複数の除外フィルターを組み込んだ場合、相互に競合して、検出結果に影響する可能性があります。例えば、以下の 2 つのフィルターが提供されたとします。

- フィルター 1 は、脆弱性タイプ `Validation.EncodingRequired` のすべての検出結果を除去します。これは反転されないため、これらの検出結果は評価から除外されます。
- フィルター 2 は、脆弱性タイプ `Validation.Required` のすべての検出結果を除去します。これは反転されないため、これらの検出結果は評価から除外されます。

スキャン構成を使用してこれらのフィルターの両方が適用された場合、デフォルトでは、これらのフィルターはお互いを無視します。フィルター 1 は、`Validation.EncodingRequired` の検出結果を除外しますが、`Validation.Required` の検出結果は含めます。フィルター 2 は、`Validation.Required` の検出結果を除外しますが、`Validation.EncodingRequired` の検出結果は含めます。最終的な結果には、`Validation.EncodingRequired` の検出結果と `Validation.Required` の検出結果がすべて含まれます。

指定された任意の 除外フィルターを除去するには、「任意の非反転除外フィルターを突き合わせます」を選択します。上記の例の場合、このチェック・ボックスを選択すると、`Validation.EncodingRequired` の検出結果および `Validation.Required` の検出結果は、すべて評価から除外されます。

4. オプション: 「汚染フロー分析」タブの「汚染フロー分析」セクションを使用して、汚染フロー分析を有効にします。「汚染フロー分析」はデフォルトで選択されており、AppScan Source によって実行される基本の分析タイプです。スキャンを起動すると、汚染フロー分析によってデータ・フロー・トレースが行われ、このトレースによって脆弱性をより的確に特定できます。分析の「有効範囲」を次のように設定できます。

- 「アプリケーション」が有効範囲の場合: 汚染フロー分析は、アプリケーション内のプロジェクト、およびプロジェクト内のファイルを対象に実行されます。
- 「プロジェクト」が有効範囲の場合: 汚染フロー分析は、プロジェクト内のファイルを対象に実行されます。
- 「ファイル」が有効範囲の場合: 汚染フロー分析は各ファイルで個別に実行されます。

注**:** 「汚染フロー分析」タブの設定は、
JavaScript、ColdFusion、Perl、Cobol、PL/SQL、または T-SQL のスキャン時には適用されません。

5. オプション: 「汚染フロー分析」タブの「スキャン・ルール」セクションを使用して、スキャンで実施されるソース・ルールを指定することができます (詳しくは、 131 ページの『「汚染フロー分析」タブ』を参照してください)。このセクションでは、選択したソース・ルール・セットを使用してスキャンを実行することを選択するか、スキャンに使用する個々のルール・プロパティーを選択することができます。
   a. デフォルトで、このセクションでは、適用するルール・セットを選択できます。 1 つ以上の使用可能なルール・セットのチェック・ボックスを選択します。
   b. ルール・セットではなく、個々のルール・プロパティーを選択するには、「選択済みのルール・セットを破棄し、個々のルール・プロパティーを選択」をクリックします。 これにより、「ルール・プロパティーの選択」ダイアログ・ボックスが開き、個々のルール・プロパティーを選択できるようになります。 このダイアログ・ボックスでの作業が完了すると、選択されていたルール・セットがすべて破棄されます。 選択されたルール・プロパティーを持つスキャン・ルールがスキャンに使用されます。

   スキャン用に個々のルール・プロパティーを選択していて、代わりにルール・セットを選択したい場合は、「選択済みのルール・プロパティーを破棄してルール・セットごとに選択する」をクリックします。 これにより、「ルール・プロパティーの選択」ダイアログ・ボックスで選択されていたルール・プロパティーが破棄され、代わりにルール・セットを選択できるようになります。

   注**:**
   - 個々のルール・プロパティーを選択する場合、選択された項目は、シンクではなくソースのプロパティーに適用されます。 つまり、攻撃対象領域を、選択したプロパティーを持つソースのみに制限することになります。脆弱性タイプはソースではなく、シンクを基準にしているため、結果には選択されたプロパティーと一致しない脆弱性が表示されることがあります。
   - 「汚染フロー分析」タブの設定は、
     JavaScript、ColdFusion、Perl、Cobol、PL/SQL、または T-SQL のスキャン時には適用されません。
6. オプション: 「汚染フロー分析」タブの「詳細設定」セクションは上級者向けです。このセクションには、スキャン結果を向上させるための、さまざまな設定が含まれています。 吹き出しテキストは、このセクションの各設定について記述しています。

注**:** 「汚染フロー分析」タブの設定は、
JavaScript、ColdFusion、Perl、Cobol、PL/SQL、または T-SQL のスキャン時には適用されません。

7. オプション: 「パターン分析」タブの設定により、パターン・ベースのスキャンのルールを有効にしたり、設定したりすることができます。パターン・ベースのスキャンは、カスタマイズされた検索基準に基づいてソース・コードの分析を行う機能です。詳しくは、 268 ページの『パターン・ベースのルールによるカスタマイズ』を参照してください。 パターン・ベースのスキャンを有効にするには、「パターン分析」チェック・ボックスを選択します。これを選択すると、「パターン・ルール・セット」セクションと「パターン・ルール」セクションが有効になります。
   a. ルール・セットを追加するには、「パターン・ルール・セット」セクションで「追加」をクリックします。これによって「パターン・ルール・セットの追加」ダイアログ・ボックスが開き、ここで 1 つ以上のルール・セットを選択できます。ルール・セットを選択すると、そこに含まれるルールがダイアログ・ボックスの右側に表示され、そのルール・セットの適用対象のプロジェクト・タイプが「プロジェクト・タイプ」フィールドにリストされます。「**OK**」をクリックして、選択したルール・セットを追加します。
   b. ルールを追加するには、「パターン・ルール」セクションで「追加」をクリックします。これによって「パターン・ルールの追加」ダイアログ・ボックスが開き、ここで 1 つ以上のルールを選択できます。「新規ルールの作成」をクリックして新規ルールを作成することもできます (272 ページの『パターン・ルールの作成』を参照してください)。新規ルールを作成すると、そのルールはリストに追加され、選択対象になります。ルールを選択または作成したら、「**OK**」をクリックして、スキャン構成に追加します。

      ヒント**:** 「パターン・ルールの追加」ダイアログ・ボックスでは、ツールチップのヘルプによって各ルールに使用される式が示されます。

      注**:**
      - ルール・セットを追加すると、そのルール・セット内のルールは「パターン・ルールの追加」ダイアログ・ボックスから除外されます。
      - ルールを追加し、さらにそのルールを含むルール・セットも追加すると、「パターン・ルール」セクションにはそのルールがリストされ、そのルールがルール・セットに含まれるものであることも示されます。このルールは既にルール・セットに含まれているので、その特定のルールを削除しようとすると、そのルールは「パターン・ルール」セクションからのみ削除され、スキャン構成からは削除されません。スキャン構成からルールを削除するには、ルール・セットを削除するか、またはルール・セットを変更してルールを含まない状態にしてください。
   c. 追加したルール・セットまたはルールを削除するには、「削除」ボタンを使用するか、右クリックして「削除」を選択します。このアクションを使用する際に、ルールとルール・セットの複数選択も可能です。

      注**:** 後で脆弱性データベースから削除されるルールまたはルール・セットがスキャン構成に含まれている場合、次のスキャン構成を開くと、そのルールまたはルール・セットは、それらが存在しないことを示すメッセージと一緒

に表示されます。それらのルールまたはルール・セットに対して「削除」アクションを実行することはできませんが、次にスキャン構成を保存したときに自動的に削除されます。

8. スキャン構成ですべての設定を完了したら、「保存」をクリックします。

スキャン構成の変更

手順

1. 「スキャン構成」ビューで、変更したいスキャン構成を選択します。

   注: スキャン構成を共有する (あるいは共有スキャン構成を変更または削除する) には、「共有構成の管理」権限が必要です。権限の設定について詳しくは、「*IBM Security AppScan Source* インストールと管理のガイド」を参照してください。

   注: 129 ページの『標準装備のスキャン構成』は変更できません。

2. スキャン構成を変更したら、「保存」をクリックします。

スキャン構成の削除

手順

1. 「スキャン構成」ビューで、削除したいスキャン構成を選択します。

   注: 129 ページの『標準装備のスキャン構成』は削除できません。

2. 「削除」をクリックします。

スキャン構成の共有と共有構成の操作

このタスクについて

スキャン構成は、他のユーザーと共有する目的で、AppScan Source データベースに保存できます。 スキャン構成を他のユーザーと共有するには、「共有」をクリックします。

注: スキャン構成を共有する (あるいは共有スキャン構成を変更または削除する) には、「共有構成の管理」権限が必要です。権限の設定について詳しくは、「*IBM Security AppScan Source* インストールと管理のガイド」を参照してください。

他のユーザーによって共有されているスキャン構成が、スキャン構成のリストに表示されます。

注:

- いったんスキャン構成を共有した後に共有を解除することはできません。 代わりに、以下のいずれかのタスクを実行してください。
  - 共有されているスキャン構成を削除します。 これにより、そのスキャン構成がサーバーから削除されます。
  - 共有されているスキャン構成を複写してから削除します。 スキャン構成を複写すると、まったく同じローカル・コピーが作成されます。
- 既に共有されているフィルターを含むスキャン構成を共有する場合、共有アクションは、プロンプトなしで完了します。しかし、ローカルで作成したフィルターを含むスキャン構成を共有する場合は、そのフィルターも共有されることがプロ

ンプトに表示されます。ローカル・フィルターを共有したくない場合は、スキャン構成の共有アクションをキャンセルすることができます。

- ローカル・フィルターを追加することで、共有スキャン構成を変更して保存することはできません。共有スキャン構成にこれらのフィルターを追加するには、フィルターを共有して、それらのフィルターを共有スキャン構成に追加します。
- 「共有構成の管理」権限があるが、「共有フィルターの管理」権限がない場合、ローカル・フィルターを含むスキャン構成を共有することはできません。

スキャン構成をデフォルトとして設定する

このタスクについて

スキャン構成がローカルであるか、組み込みであるか、共有されているかにかかわらず、任意のスキャン構成をデフォルトとして設定することができます。 共有されているスキャン構成をデフォルトとして設定した場合、その設定はローカル上でのみ行われるため、他のユーザーには影響しません。スキャン構成は、スキャン時に常に使用されます。デフォルト・スキャン構成を設定し、後でそのスキャン構成を削除した場合、スキャン時には、標準装備のスキャン構成である「通常のスキャン」が確認なしで使用されます。

デフォルトのスキャン構成がどのように使用されるかについては、 119 ページの『ソース・コードのスキャン』を参照してください。

手順

1. 「スキャン構成」ビューで、デフォルトとして設定したいスキャン構成を選択します。
2. 「デフォルトとして選択」をクリックします。

標準装備のスキャン構成

このタスクについて

AppScan Source には、標準装備のスキャン構成が用意されています。 これらを変更または削除することはできません。これらのスキャン構成をリストで選択すると、複写したり、その設定を表示したりすることができます。

「スキャン構成」ビュー

「スキャン構成」ビューを使用して、スキャンの起動時に使用できる構成を作成することができます。このビューを使用すると、デフォルトのスキャン構成の設定も可能です。スキャン構成では、スキャン時に使用するソース・ルールを指定し、多数のスキャン設定を組み込むことができます。 スキャン構成で設定を行うと、良好なスキャン結果が得られることが多く、また、これらの設定を保存することができるため、スキャンを容易に、しかも短時間で行うことができます。

「スキャン構成」ビューには、以下の主なセクションがあります。

- 130 ページの『スキャン構成の管理』
- 130 ページの『「全般」タブ』
- 131 ページの『「汚染フロー分析」タブ』
- 132 ページの『「パターン分析」タブ』

スキャン構成の管理

このセクションは、スキャン構成を選択、追加、削除、保存、および共有する場合や、スキャン構成をデフォルトとして設定する場合に使用します。

- 新規スキャン構成を作成するには、「新規」をクリックします。スキャン構成の設定が完了したら、「保存」をクリックして変更内容を保存します。 スキャン構成をデフォルトとして設定するには、保存後に「デフォルトとして選択」をクリックします。 デフォルトのスキャン構成がどのように使用されるかについては、 119 ページの『ソース・コードのスキャン』を参照してください。
- 既存のスキャン構成を操作するには、既存のスキャン構成をリストから選択します。
  - スキャン構成の設定を変更する場合は、「保存」をクリックして変更内容を保存します (不要な変更内容は、別のスキャン構成に切り替えてから「破棄」をクリックすると、破棄することができます)。
  - 選択したスキャン構成を削除するには、「削除」をクリックします。
  - スキャン構成を複写するには、「複写」をクリックします。これにより、元のスキャン構成の設定に基づいて新しいスキャン構成が作成されます。
  - スキャン構成をデフォルトとして設定するには、「デフォルトとして選択」をクリックします。 デフォルトのスキャン構成がどのように使用されるかについては、 119 ページの『ソース・コードのスキャン』を参照してください。
  - スキャン構成を他のユーザーと共有するには、「共有」をクリックします。スキャン構成が AppScan Source データベース に保存されます。

    注: スキャン構成を共有する (あるいは共有スキャン構成を変更または削除する) には、「共有構成の管理」権限が必要です。権限の設定について詳しくは、「*IBM Security AppScan Source* インストールと管理のガイド」を参照してください。

注: AppScan Source には、標準装備のスキャン構成が用意されています。 これらを変更または削除することはできません。これらのスキャン構成をリストで選択すると、複写したり、その設定を表示したりすることができます。

「全般」タブ

基本情報

このセクションでは、スキャン構成に名前を付けて説明を提供することができます。

フィルター

このセクションでは、スキャン構成を使用すると必ずスキャンに適用されるフィルターを 1 つ以上選択できます。フィルターを選択するときは、AppScan Source 事前定義フィルターまたは 共有フィルター、あるいは自分で作成したフィルターを選択できます。詳しくは、 123 ページの『スキャン構成の管理』を参照してください。

「汚染フロー分析」タブ

汚染フロー分析

汚染フロー分析を有効にし、その有効範囲を設定します。

スキャン・ルール

このセクションは、スキャンで有効になるソース・ルールを判別するために使用します。

ソースはプログラムへの入力で、ファイル、サーブレット要求、コンソール入力、ソケットなどがあります。 一部のソース・ルールを除外することにより、スキャンの速度を上げたり、関係のない入力に起因する脆弱性の検出を避けることができます。

ルールが特定の脆弱性、メカニズム、属性、またはテクノロジーに関連していることを示すには、ルール・プロパティーでルールをタグ付けします。 これらのプロパティーはルール・セットにグループ化され、これらのルール・セットは、関連したルールの共通セットに対応します。 ルール・セットまたは個々のルール・プロパティーのいずれかを指定することにより、スキャンに含めるソース・ルールを制限できます。

- スキャンに組み込む 1 つ以上の脆弱性タイプ (ルール・セット内でタイプ別に編成される) を選択します。
  - すべて: これを選択すると、サポートされるすべての入力のソースに起因する脆弱性が検出されます。
  - ユーザーの入力: これを選択すると、エンド・ユーザーによる入力に起因する脆弱性が検出されます。
  - **Web** アプリケーション: これを選択すると、Web アプリケーションのリスクに起因する脆弱性が検出されます。
  - エラー処理およびロギング: これを選択すると、エラー処理とロギングのメカニズムに起因する脆弱性が検出されます。
  - 環境: これを選択すると、構成ファイル、システム環境ファイル、およびプロパティー・ファイルに起因する脆弱性が検出されます。
  - 外部システム: これを選択すると、外部エンティティーに起因する脆弱性が検出されます。
  - データ・ストア: これを選択すると、データ・ストア (データベースやキャッシュ処理など) に起因する脆弱性が検出されます。
  - 異常な項目: これを選択すると、通常は実動アプリケーションの一部ではないルーチンに起因する脆弱性が検出されます。
  - ファイル・システム: これを選択すると、ファイル・システムに起因する脆弱性が検出されます。
  - 機密データ: これを選択すると、機密データに起因する脆弱性が検出されます。

  吹き出しテキストは、このセクションの各ルール・セットについて記述しています。

- スキャンに組み込む個々のスキャン・ルール・プロパティーを選択します。「選択済みのルール・セットを破棄し、個々のルール・プロパティーを選択」をクリックします。 これにより、「ルール・プロパティーの選択」ダイアログ・ボックスが開き、個々のルール・プロパティーを選択できるようになります。このダイアログ・ボックスでの作業が完了すると、選択されていたルール・セットがすべて破棄されます。 選択されたルール・プロパティーを持つスキャン・ルールがスキャンに使用されます。

詳細設定

このセクションは、上級者向けです。 このセクションには、スキャン結果を向上させるための、さまざまな設定が含まれています。 吹き出しテキストは、このセクションの各設定について記述しています。

「パターン分析」タブ

パターン分析

このセクションを使用して、スキャン構成を使用する場合にパターン・ベースのスキャンを有効にします。パターン・ベースのスキャンは、カスタマイズされた検索基準に基づいてソース・コードの分析を行う機能です。

「パターン・ルール・セット」および「パターン・ルール」

これらのセクションを使用して、パターン分析時に使用するルールとルール・セットを追加します。詳しくは、 268 ページの『パターン・ベースのルールによるカスタマイズ』および 123 ページの『スキャン構成の管理』を参照してください。

Java の増分分析

増分分析が有効にされている場合、AppScan Source によって分析データがキャッシュに入れられます。その後、プロジェクトあるいはアプリケーションを再スキャンすると、AppScan Source は、このデータを使用してコードの変更を判別し、その変更によって影響を受けるコードの部分のみが再度分析されます。これにより、コードの分析は完全に行われますが、時間は短縮されます。

このタスクについて

増分分析は、Windows および Linux でサポートされます。増分分析は、有効にされている場合、AppScan Source プロジェクトまたはアプリケーション上、あるいは Eclipse プロジェクトまたはワークスペース上で実行されます。増分分析を有効にした後、プロジェクト、アプリケーション、またはワークスペース上で実行する最初のスキャンは、常に完全スキャンです (脆弱性分析キャッシュは、完全スキャンでのみ更新されます)。これにより、AppScan Source は、後続のスキャン用にデータをキャッシュに入れることができます。脆弱性分析キャッシュが消去されておらず、変更されたファイルの数がしきい値設定 (ユーザーが決定できます) を超えていない限りは、その後のプロジェクト、アプリケーション、またはワークスペースのスキャンは増分スキャンになります。

増分分析を有効にして使用するには、以下のステップを実行します。

手順

1. テキスト・エディターで <data_dir>¥config¥scan.ozsettings を開きます (<data_dir> は、ご使用の AppScan Source プログラム・データの場所です。詳しくは、 360 ページの『インストールとユーザー・データ・ファイルの場所』 を参照してください。)。ファイル内の incremental_analysis 設定を見つけます。この設定は、以下の例のようになります。

```
<Setting
  name="incremental_analysis"
  read_only="false"
  default_value="false"
  description="Attempt to scan only changed files,
    instead of re-scanning everything."
  type="bool"
  value="false"
  display_name="Incremental Analysis"
  hidden="true"
/>
```

   この設定では、value 属性を変更します。属性が true に設定されている場合、この設定はオンになります。 false に設定されている場合、AppScan Source は、スキャン時に増分分析を実行しません。

2. <data_dir>¥config¥scan.ozsettings で、percentage_of_files_changed 設定を見つけます。

```
<Setting
  name="percentage_of_files_changed"
  read_only="false"
  default_value="50"
  description="In incremental scanning, if percentage of files
  being changed since last scan exceeds the threshold, full
  scan will be initiated. The percentage ranges from 0 to 100.
  Default threshold is 50, which represents 50%."
  type="int"
  value="50"
  display_name="Percentage of files being changed"
  hidden="true"
/>
```

   この設定により、完全スキャンを開始する前に変更されている必要があるファイルのパーセンテージを指定することができます。デフォルトでは、このしきい値パーセンテージは 50% です。これは、プロジェクト、アプリケーション、またはワークスペース内の 50% 以上のファイルが変更された後に再スキャンを行うと、増分分析スキャンではなく完全スキャンが開始されることを意味します。必要に応じて、この設定で value 属性を目的のしきい値パーセンテージに変更します。

3. すべての関連設定を変更した後、<data_dir>¥config¥scan.ozsettings を保存し、増分分析をサポートする AppScan Source 製品を始動または再始動します。例えば、AppScan Source for Analysis、AppScan Source for Development Eclipse プラグイン、または AppScan Source コマンド行インターフェース (CLI) を再始動するか、AppScan Source for Automation サービスを再始動します。

4. 同じスキャン構成を使用して Java アプリケーションまたはプロジェクトを再スキャンする際に、変更されたファイルがしきい値を超えておらず、 脆弱性分析キャッシュが消去されていない場合は、増分分析が実行されるようになりました。

5. 脆弱性分析キャッシュの消去**:** 増分スキャンに問題がある場合、あるいは増分分析が有効にされている状態で完全分析スキャンを実行したい場合は、スキャンを再実行する前に脆弱性キャッシュを消去します。
   - AppScan Source for Analysis:
     a. AppScan Source プロジェクトの「プロパティー」ビューを開きます。アプリケーションをスキャンする場合は、子プロジェクトのプロパティー・ビューを開きます (プロジェクトのキャッシュを削除すると、そのアプリケーションのキャッシュの削除されます)。
     b. 「概要」タブで、「キャッシュの消去」をクリックします。
   - AppScan Source for Development Eclipse プラグイン: `<data_dir>¥temp¥<workspace>¥<project>` を削除します。ここで、
     – `<data_dir>` は、ご使用の AppScan Source プログラム・データの場所です。詳しくは、 360 ページの『インストールとユーザー・データ・ファイルの場所』 を参照してください。
     – `<workspace>` は、スキャンを行う Eclipse ワークスペースの名前です。ワークスペース全体のキャッシュを削除するには、`<data_dir>¥temp¥<workspace>` ディレクトリー全体を削除します。
     – `<project>` は、スキャンを行う Eclipse プロジェクトの名前です。プロジェクトのキャッシュを削除するには、`<data_dir>¥temp¥<workspace>¥<project>` ディレクトリーを削除します。
   - AppScan Source コマンド行インターフェース (CLI): 「*IBM Security AppScan Source Utilities* ユーザー・ガイド」の説明に従って、`clearcache` コマンドを使用します。
   - AppScan Source for Automation: 「*IBM Security AppScan Source Utilities* ユーザー・ガイド」の説明に従って、`ScanApplication` コマンドの `-clearcache` 引数を使用します。

タスクの結果

AppScan Source for Analysis でのスキャンが完了した後、**Assessment Diff** 機能を使用して、コード変更の前後の評価を比較することができます。

ヒント**:**

- 完全分析スキャンを強制的に実行したい場合は、増分分析を無効にするか、脆弱性分析キャッシュを消去してください。
- 増分分析を実行する場合、以下の変更のいずれかを行った後には完全分析スキャンを実行する必要があります。
  – プロジェクトまたはアプリケーションに適用可能なカスタム・ルールに対するセキュリティー・ルールの変更。
  – スキャン構成の変更。
  – スキャンに影響する `.ozsettings` ファイルへの変更。
  – アプリケーションまたはプロジェクトのプロパティーに対する変更。例えば、AppScan Source for Analysis の「プロパティー」ビューで、「すべてのアプリケーション」、「選択されたアプリケーション」、または「プロジェクト」に対して行った変更などです。

  - 新規プロジェクトのアプリケーションへの追加、または既存のプロジェクトの削除。
  - ファイルのスキャンからの除外。 例えば、AppScan Source for Analysis では、「エクスプローラー」ビューでファイルを右クリックするか、「スキャンから除外」を選択して、スキャンからファイルを除外することを選択できます。
- 増分分析についての現行情報は、http://www.ibm.com/support/docview.wss?uid=swg21994390を参照してください。

注:
- 増分スキャン後、エディターの検出結果マーカーは、正しい場所に配置されなくなります。
- 増分スキャン結果に、トレースを含まない、修復済みの検出結果が表示される場合があります。
- 増分スキャン中は同時に複数の AppScan Source 製品またはコンポーネントを開いたままにしておくことはできません。また、他のユーザーがスキャン中のアプリケーションやプロジェクトを、同じマシン上で同時にスキャンすることはできません。

スキャンからのファイルの除外

始める前に

注: 使用しているアプリケーションが Eclipse ワークスペースの場合、ファイルをスキャンから除外することはできません。

手順

1. 「エクスプローラー」ビューで、スキャンから除去する 1 つ以上のファイルを選択します。
2. 選択項目を右クリックし、メニューから「スキャンから除外」を選択します。

タスクの結果

そのファイルが含まれているプロジェクトの「プロパティー」ビューを開くと、ビューの「ソース」タブには、除外済みファイルを含むプロジェクト内のファイルがリストされています。

プロジェクト・ファイルは、「ソース・ルート」 アイコンの下に表示されます。スキャンから除外されたファイルには、赤いファイル・アイコンが付いています。(除外済みファイルを右クリックすると、そのメニューで「除外」は無効に、「含める」は有効になっています)。組み込みファイルを除外するには、ファイルを右クリックして、メニューで「除外」を選択します。 除外済みファイルを組み込むには、ファイルを右クリックして、メニューで「含める」を選択します。

スキャンのキャンセルまたは停止

スキャンの進行中にスキャンをキャンセルできますが、キャンセルした場合は、そのスキャンのすべてのデータが失われます。 別の方法として、スキャンを一時的に停止し、それまでに検出された結果を使用して評価を生成できます。

AppScan Source for Analysis でのスキャンのキャンセルまたは停止

現在進行中のスキャンをキャンセルするには、メインメニューから「スキャン」 > 「スキャンのキャンセル」または「スキャン」 > 「スキャンの停止」を選択します。

「スキャンのキャンセル」は、スキャンを強制終了し、結果をまったく生成しません。 「スキャンの停止」は、スキャンを一時停止し、それまでに検出された結果を使用して評価を生成します。

AppScan Source for Development (Eclipse プラグイン) でのスキャンのキャンセルまたは停止

スキャンの実行中に以下を行います。

- スキャンをキャンセルするには、メインメニューから「セキュリティー分析」 > 「スキャン」 > 「スキャンのキャンセル」を選択します。スキャンが強制終了します。結果は生成されません。キャンセル診断メッセージが Eclipse コンソールに表示されます。
- スキャンを停止するには、メインメニューから「セキュリティー分析」 > 「スキャン」 > 「スキャンの停止」を選択します。スキャンが強制終了し、停止アクションが開始されるまでに収集された結果の評価が生成されます。

注**:** AppScan Source for Development (Eclipse プラグイン) は、Windows および Linux でのみサポートされています。

AppScan Source for Development のスキャンのキャンセル (Microsoft Visual Studio プラグイン)

スキャンの実行中に、メインメニューから「**IBM Security AppScan Source**」 > 「スキャン」 > 「スキャンのキャンセル」を選択します。 スキャンが強制終了します。スキャンの結果は生成されません。

注**:** AppScan Source for Development Microsoft Visual Studio プラグインは Windows でのみサポートされています。

Linux での AppScan Source for Analysis および AppScan Source for Development (Eclipse プラグイン) コンポーネントの前提条件

Linux 上の Eclipse では、ブラウザー・ベースのコンテンツをレンダリングするために、サード・パーティー・コンポーネントをインストールする必要があります。このコンポーネントがないと、AppScan Source for Analysis および AppScan Source for Development Eclipse プラグインは、ログイン後にハングしたり、製品使用中に障害が発生したりするなどの症状を示す可能性があります。

この前提条件についての情報は、http://www.eclipse.org/swt/faq.php#browserwebkitgtk で参照できます。

- 137 ページの『Linux 上の AppScan Source for Analysis でブラウザー・ベースのコンテンツを使用可能にする』

- 『Linux で Eclipse バージョン 3.7 以降にインストールされている AppScan Source for Development でブラウザー・ベースのコンテンツを使用可能にする』

Linux 上の AppScan Source for Analysis でブラウザー・ベースのコンテンツを使用可能にする

AppScan Source for Analysis は Eclipse 上に構築されているため、この問題による影響を受けます。

この問題を修正するために推奨される方法は、32 ビット版または i686 版の WebKitGTK 1.2.0 以降をインストールすることです。パッケージを入手してインストールするために適切な方法をシステム管理者に相談してください。これは、システムによっては、`yum install webkitgtk.i686` を発行するだけで行えます。

WebKitGTK をインストールできない場合、Mozilla XULRunner 1.8 の 32 ビット・バージョンをインストールできます。 このオプションでは、環境変数に以下の更新を行う必要があるかもしれません。

- `MOZILLA_FIVE_HOME` を XULRunner のインストール場所に設定します。
- `LD_LIBRARY_PATH` の末尾または先頭に `$MOZILLA_FIVE_HOME` を追加して更新します。

Linux で Eclipse バージョン 3.7 以降にインストールされている AppScan Source for Development でブラウザー・ベースのコンテンツを使用可能にする

この問題を修正するために推奨される方法は、32 ビット版または i686 版の WebKitGTK 1.2.0 以降をインストールすることです。パッケージを入手してインストールするために適切な方法をシステム管理者に相談してください。これは、システムによっては、`yum install webkitgtk.i686` を発行するだけで行えます。

WebKitGTK をインストールできない場合、Mozilla XULRunner 1.8 の 32 ビット・バージョンをインストールできます。 このオプションでは、環境変数に以下の更新を行う必要があるかもしれません。

- `MOZILLA_FIVE_HOME` を XULRunner のインストール場所に設定します。
- `LD_LIBRARY_PATH` の末尾または先頭に `$MOZILLA_FIVE_HOME` を追加して更新します。

「自分の評価」の管理

「自分の評価」ビューには、評価のリストが表示されます。リストには、現在開かれている評価と、自分が保存した評価が含まれています。このビューでは、評価を開く、削除する、保存する、名前変更する、または比較する操作を実行できます。スキャンが完了するか、保存済みの評価を開くと、「マイ評価」ビューに評価が表示されます。 「マイ評価」には、開いているまたは保存済みの評価の表が表示され、公開済みまたは変更済みの評価が識別されます。 評価の保存と公開を行わずにこのビューから評価を削除すると、その評価が完全に削除されます。

「自分の評価」ビューについて詳しくは、 337 ページの『「自分の評価」ビュー』を参照してください。

制約事項: 複数のアプリケーションまたはプロジェクトをスキャンする場合、各スキャン対象項目の評価を含む親ノードが「自分の評価」ビューに作成されます。この場合、個々の子評価を管理することはできません (例えば、子評価を個々に削除したり、公開したりすることはできません)。複数のアプリケーションまたはプロジェクトを同時にスキャンする場合は、評価をグループ (親ノード) としてのみ管理することができます。

ヒント: 一度に開くことができるのは、1 つのアプリケーションに関連するスキャン結果のみです。 複数のアプリケーションまたはプロジェクトのスキャン結果を表示するには、「自分の評価」ビュー内のツリーを展開し、開きたい評価をダブルクリックする必要があります。

分析のためのクラウドへの AppScan Source 評価の送信

IBM Cloud Marketplace での IBM Application Security on Cloud に対するサブスクリプション、あるいは Application Security on Cloud for Bluemix に対するサブスクリプションがある場合は、AppScan Source 評価を分析のために送信することができます。AppScan Source バージョン 9.0 以上からの評価がサポートされます。送信できるスキャンの数は、Application Security on Cloud サブスクリプションによって異なります。

このタスクについて

Application Security on Cloud サービスの 静的分析機能を使用する場合、Intelligent Finding Analytics (IFA) を使用するセキュリティー分析レポートを生成することができます。 IFA は強力な機械学習テクノロジーであり、特に誤検出をフィルタリングによって除去し、1 つのコード・ポイントの修正によって修復できる検出結果をグループ化することで、トリアージ作業の大部分を自動的に実行します。 IFA について詳しくは、この記事を参照してください。

AppScan Source バージョン 9.0 以上を使用しており、Application Security on Cloud サブスクリプションがある場合、AppScan Source 評価を Application Security on Cloud にアップロードすることで、このテクノロジーの恩恵を受けることができます。その返信として、このテクノロジーによって自動的にトリアージされた新規の評価を受け取ります。この評価は、HTML レポートの形式、またはご使用の AppScan Source 製品で開くことができる評価の形式で受け取ることができます。

Application Security on Cloud サブスクリプションがある場合、月ごとのスキャン数が制限される可能性があります。スキャンおよび同時スキャンのライセンスについて詳しくは、http://www.ibm.com/support/knowledgecenter/SSYJJF_1.0.0/ApplicationSecurityonCloud/src_managing_assessments_cloud.html を参照してください。

注: 無料試用版の Application Security on Cloud を使用して AppScan Source 評価をスキャンしている場合、IFA によってトリアージされた AppScan Source 評価ファイルに加えて、フル HTML レポートをダウンロードすることができます。その他のすべてのスキャン・タイプについては、無料試用版を使用している場合は、要約レポートのみをダウンロードすることができます。

手順

1. 既に **Application Security on Cloud** を使用して 静的分析 を行っている場合は、このステップをスキップしてください。
   a. Application Security on Cloud サブスクリプションがない場合、以下のようにして取得することができます。
      - **IBM Cloud Marketplace:** https://appscan.ibmcloud.com/serviceui/home にアクセスし、IBM ID を使用してサインインします。 IBM ID がない場合は、リンクを使用して作成してください。その後、無料試用版の登録を行うか、サービスにあるリンクを使用して有料のサブスクリプションを契約します。
      - **IBM Bluemix®:** https://console.ng.bluemix.net/ にアクセスし、「登録」ボタンを使用して、Bluemix のフォームに入力して登録します。その後、Application Security on Cloud for Bluemix サービス・インスタンスを作成します。
   b. **IBM Cloud Marketplace** のみ**:** Application Security on Cloud サービスで、アプリケーションを作成し (http://www.ibm.com/support/knowledgecenter/SSYJJF_1.0.0/ApplicationSecurityonCloud/ent_create_application.html を参照)、「スキャンの作成」をクリックします。
   c. 「今日はどのタイプのアプリをスキャンしますか**?**」画面で、「デスクトップ」または「**Web**」 > 「**Static**」を選択します。
   d. 以前に Static Analyzer クライアント・ユーティリティー をダウンロードおよびセットアップしていない場合は、ここで行います。詳しくは、http://www.ibm.com/support/knowledgecenter/SSYJJF_1.0.0/ApplicationSecurityonCloud/src_utility_install.htmlを参照してください。
2. AppScan Source 製品または任意のツールで評価 (`.ozasmt` ファイル) を生成します。バージョン 9.0 以上がサポートされます。
3. クライアント・ユーティリティー コマンド・ライン・インターフェース (CLI) を使用して、評価用 (`.ozasmt` ファイル) の 中間表現 (IRX または `.irx`) ファイルを生成します。
   a. クライアント・ユーティリティー をローカル・ドライブに抽出した後、その `¥bin` ディレクトリーのロケーションを `PATH` 環境変数に追加します。これを行わないと、コマンドを発行するたびに、`¥bin` ディレクトリーを使用してすべての クライアント・ユーティリティー CLI コマンドを修飾する必要があります。詳しくは、http://www.ibm.com/support/knowledgecenter/SSYJJF_1.0.0/ApplicationSecurityonCloud/src_irx_gen_cli.htmlを参照してください。
   b. Windows では、次のコマンドを発行します。

      ```
      appscan package -d <save_path> -f <assessment_file> -n <file_name>
      ```

      Linux では、次のコマンドを発行します。

      ```
      appscan.sh package -d <save_path> -f <assessment_file> -n <file_name>
      ```

      コマンド引数はオプションです。

      - `-d`: `-d <save_path>` と指定します。ここで、`<save_path>` は IRX ファイルを保存したいディレクトリーです。

- `-f`: `-f <assessment_file>` を指定します。ここで、`<assessment_file>` は、スキャンするためにパッケージする `.ozasmt` ファイルです。`<assessment_file>` ファイルが現行ディレクトリーにない場合、このオプションを使用してアセスメント・ファイルのパスおよびファイル名を指定します。

  注**:** このオプションが必要なのは、以下の記述のいずれかまたは両方が当てはまる場合のみです。

  – コマンドを発行するディレクトリーに複数のアセスメント・ファイルが含まれている。ディレクトリーに含まれているアセスメント・ファイルが 1 つのみであれば、`-f` オプションが使用されていない場合はそのファイルがパッケージされます。
  – コマンドを発行するディレクトリーにアセスメント・ファイルが 1 つも含まれていない。この場合、`-f` オプションを使用して、パッケージするアセスメント・ファイルのパスおよびファイル名を指定する必要があります。
- `-n`: `-n <file_name>` と指定します。ここで、`<file_name>` は IRX ファイル名です。`.irx` ファイル拡張子付きまたはファイル拡張子なしでファイル名を指定できます。ファイル拡張子なしで指定すると、ファイル生成時に自動的に拡張子が付けられます。

`package` コマンドに関する追加情報 (使用例を含む) は、構成コマンド (Windows) または構成コマンド (Linux) を参照してください。

4. CLI `queue_analysis` コマンドを使用して、IRX ファイルをアップロードします。
   a. CLI からサービスにログインします。これを行う方法は、IBM Cloud Marketplace と IBM Bluemix で異なります。CLI でのサービスに対する認証について詳しくは、認証コマンド (Windows) または認証コマンド (Linux) を参照してください。
      - **IBM Cloud Marketplace:**

        Windows では、次のコマンドを発行します。

        ```
        appscan scx_login -P <password> -u <user_name> -persist
        ```

        Linux では、次のコマンドを発行します。

        ```
        appscan.sh scx_login -P <password> -u <user_name> -persist
        ```

        これらの引数は必須です。

        – `-P`: `-P <password>` と指定します。ここで、`<password>` は、ユーザーが Application Security on Cloud サービスに登録した際に指定したパスワードです。
        – `-u`: `-u <user_name>` と指定します。ここで、`<user_name>` は、ユーザーが Application Security on Cloud サービスに登録した際に指定した E メール・アドレスです。

        この引数はオプションです。

        – `-persist`: ログイン・トークン・ファイルの有効期限が切れるときに、サービスに対する再認証を自動的に試みます。

- **IBM Bluemix:**

  Windows では、次のコマンドを発行します。

  ```
  appscan login -P <password> -u <user_name> -persist
  ```

  Linux では、次のコマンドを発行します。

  ```
  appscan.sh login -P <password> -u <user_name> -persist
  ```

  これらの引数は必須です。

  – -P: -P <password> と指定します。ここで、<password> は、サービス資格情報で指定されているパスワードです。
  – -u: -u <user_name> と指定します。ここで、<user_name> は、サービス資格情報で指定されているバインディング ID です。

  Bluemix サービス資格情報を判別するには、サービス・ダッシュボードの左側のナビゲーション・ペインで「サービス資格情報」を選択します。外部アプリが Bluemix サービスを使用できるようにするを参照してください。

  この引数はオプションです。

  – -persist: ログイン・トークン・ファイルの有効期限が切れるときに、サービスに対する再認証を自動的に試みます。

b. queue_analysis コマンドを使用して、IRX ファイルをアップロードします。

- Windows では、次のコマンドを発行します。

  ```
  appscan queue_analysis -a <app_id> -f <irx_file> -n <scan_name>
  ```

  Linux では、次のコマンドを発行します。

  ```
  appscan.sh queue_analysis -a <app_id> -f <irx_file> -n <scan_name>
  ```

  これらの引数は必須です。

  – -f: -f <irx_file> と指定します。ここで、<irx_file> は、スキャンするためにサブミットする IRX ファイルです。IRX ファイルが現行ディレクトリーにない場合、このオプションを使用して IRX ファイルのパスおよびファイル名を指定します。

    注: このオプションが必要なのは、以下の記述のいずれかまたは両方が当てはまる場合のみです。

    - コマンドを発行するディレクトリーに複数の IRX ファイルが含まれている。ディレクトリーに含まれている IRX ファイルが 1 つのみであれば、 -f オプションが使用されていない場合はそのファイルがサブミットされます。
    - コマンドを発行するディレクトリーに IRX ファイルが 1 つも含まれていない。この場合、 -f オプションを使用して、サブミットする IRX ファイルのパスおよびファイル名を指定する必要があります。

  – -n: -n <scan_name> と指定します。ここで、<scan_name> は、クラウドで実行されるスキャンの名前です。

– `-a` **(IBM Cloud Marketplace のみ)**: IBM Cloud Marketplace の Application Security on Cloud サービスに接続されている場合、クラウドに送信する IRX ファイルは、既存の Application Security on Cloud アプリケーションに関連付けられている必要があります。このオプションを使用する際は、`-a <app_id>` と指定します。ここで `<app_id>` は、関連付けるアプリケーションの ID です。 ID を判別するには、`list_apps` コマンドを使用します。

- `queue_analysis` コマンドが完了すると、分析ジョブの ID が表示されます。 CLI を使用して Application Security on Cloud 分析レポートを受け取りたい場合は、このジョブ ID を `get_result` コマンドに含める必要があります。ID をメモしてください。CLI を使用して分析レポートを受け取りたい場合は、`.ozasmt` ファイルが含まれるアーカイブ (`.zip`) ファイルを受け取るというオプションもあります。これにより、分析レポートを AppScan Source で開くことができます。HTML レポートを表示するだけで構わない場合は、CLI または Application Security on Cloud Web クライアントを使用して、レポートをダウンロードすることができます。

`queue_analysis` コマンドの使用について詳しくは、分析コマンド (Windows) または分析コマンド (Linux) を参照してください。

5. 分析が完了すると、CLI を使用して IRX をアップロードした場合、あるいは Application Security on Cloud Web クライアントで「スキャンの完了時に **E** メールで通知 **(Email me when the scan is complete)**」チェック・ボックスを選択した場合は、E メールを受信します。
6. 分析レポートを取得する方法を選択します。CLI `get_result` コマンドまたは Application Security on Cloud Web クライアントを使用することができます。CLI を使用して分析レポートを受け取りたい場合は、`.ozasmt` ファイルが含まれるアーカイブ (`.zip`) ファイルを受け取るというオプションもあります。これにより、分析レポートを AppScan Source で開くことができます。HTML レポートを表示するだけで構わない場合は、CLI または Application Security on Cloud Web クライアントを使用して、レポートをダウンロードすることができます。
7. **CLI `get_result`** コマンドを使用して分析レポートを取得する場合は、次のステップを実行します。
   a. CLI からサービスにログインしていることを確認します。
   b. Windows では、次のコマンドを発行します。

      ```
      appscan get_result -d <file_path> -i <job_id> -t <type>
      ```

      Linux では、次のコマンドを発行します。

      ```
      appscan.sh get_result -d <file_path> -i <job_id> -t <type>
      ```

      この引数は必須です。

      - `-i`: `-i <job_id>` と指定します。ここで、`<job_id>` は分析ジョブの ID です。

      注**:** `queue_analysis` コマンドの発行時に ID をメモしなかった場合、`appscan list` コマンドまたは `appscan.sh list` コマンドを使用して、すべ

ての分析ジョブのリストを表示することができます。詳しくは、分析コマンド (Windows) または分析コマンド (Linux) を参照してください。

これらの引数はオプションです。

- `-d`: `-d <file_path>` と指定します。ここで、`<file_path>` は、宛先ファイルの完全修飾パスまたは宛先ファイルのファイル名、あるいはその両方です。ファイル名が指定されていない場合、ファイル名はスキャン・ジョブ名に基づいた名前になります。パスが指定されていない場合、ファイルは現行ディレクトリーに保存されます。このオプションが組み込まれていない場合、ファイルは、スキャン・ジョブ名に基づいたファイル名で現行ディレクトリーに保存されます。
- `-t`: `-t <type>` と指定します。ここで、`<type>` は、 `html` または `zip` のいずれかです。結果は HTML ファイルとして、または、HTML 結果を含んでいる `.zip` ファイルとして保存されます。 このオプションが組み込まれていない場合、結果は HTML ファイルとして保存されます。

  スキャン結果が `package` コマンドにより生成された IRX ファイルに対するものである場合、`-t zip` を指定すると、AppScan Source バージョン 9.0 以降の製品にロード可能な新規の `.ozasmt` ファイルを含む結果が保存されます。

`get_result` コマンドの使用について詳しくは、結果コマンド (Windows) または結果コマンド (Linux) を参照してください。

8. **Web** クライアントを使用して分析レポートを取得する場合は、次のステップを実行します。HTML レポートを表示するだけで構わない場合は、Application Security on Cloud Web クライアントを使用して、レポートをダウンロードすることができます。

   サービスにログインすると、スキャンのリストが自動的に表示されます (サービスの別のセクションにナビゲートされた場合は、右上部にある **X** アイコンをクリックすると、スキャンのリストに戻ります)。スキャン・リストでスキャンを見つけ、「ダウンロード」アイコンを選択して、XML 形式または HTML 形式を選択します。

   IBM Cloud Marketplace での Application Security on Cloud スキャン結果について詳しくは、http://www.ibm.com/support/knowledgecenter/en/SSYJJF_1.0.0/ApplicationSecurityonCloud/appseccloud_results_dashboard_cm.html を参照してください。IBM Bluemix で、https://console.ng.bluemix.net/docs/services/ApplicationSecurityonCloud/appseccloud_results.html#results を参照してください。

評価の公開

AppScan Source には、2 つの公開オプションが用意されています。 評価を保管および共有するためには、評価を AppScan Source データベースに公開することができます。 また、ご使用の AppScan Enterprise Server が Enterprise Console オプションを指定してインストールされている場合は、Enterprise Console に評価を公

開することができます。 AppScan Enterprise Console は、レポート機能、問題管理、トレンド分析、ダッシュボードなど、評価に関する作業を行うためのさまざまなツールを備えています。

AppScan Source の公開機能について詳しくは、 145 ページの『AppScan Source への評価の公開』および 146 ページの『AppScan Enterprise Console への評価の公開』を参照してください。

注: AppScan Source および AppScan Enterprise の一部のバージョンでは、AppScan Source から AppScan Enterprise Console に評価を公開するために 2 つの製品のバージョンとリリース・レベルが一致していなければなりません。 評価を公開する場合に、互換性がある AppScan Source と AppScan Enterprise のバージョンを確認するには http://www.ibm.com/support/docview.wss?uid=swg21975211 を参照してください。

AppScan Source に公開するためのアプリケーションおよびプロジェクトの登録

AppScan Source データベースに評価を公開するには、その評価を作成するためにスキャンされたアプリケーションまたはプロジェクトをあらかじめ登録しておく必要があります。デフォルトでは、未登録のアプリケーションまたはプロジェクトの評価を公開しようとすると、そのアプリケーションまたはプロジェクトをその時点で登録するよう指示するプロンプトが出されます。「全般」設定の「初期公開時にアプリケーションを自動登録する」が「常に登録」に設定されている場合は、AppScan Source for Analysis がユーザーに代わって自動登録 します。

重要: アプリケーションおよびプロジェクトを登録するには、「登録」権限が必要です。

スキャンの前にアプリケーションとプロジェクトを登録するには、「エクスプローラー」ビューで対象のアプリケーションまたはプロジェクトを選択し、メイン・ワークベンチ・メニューから「ファイル」 > 「登録」を選択します。「エクスプローラー」ビューで選択した項目を右クリックして、「アプリケーションの登録」および「プロジェクトの登録」アクションを使用することもできます。

アプリケーションが既に登録されている場合は、そのアプリケーションを新規の名前で再び登録できます。 これを実行するには、そのアプリケーションを選択および右クリックし、メニューから「アプリケーションを別名で登録」を選択します。「名前変更」ダイアログ・ボックスに、登録済みのアプリケーションまたはプロジェクトの新規の名前を入力します。

アプリケーションおよびプロジェクトを登録抹消するには、「エクスプローラー」ビューでアプリケーションまたはプロジェクトを選択し、メイン・ワークベンチ・メニューから「ファイル」 > 「登録抹消」を選択します。 「エクスプローラー」ビューで選択した項目を右クリックして、「アプリケーションの登録抹消」および「プロジェクトの登録抹消」アクションを使用することもできます。

注: 項目を登録抹消しても、公開済みのデータは AppScan Source データベースから削除されません。

AppScan Source への評価の公開

評価を保管する場合および共有する場合は、評価を AppScan Source データベースに公開することができます。

このタスクについて

アプリケーションおよびプロジェクトの評価を公開するには、アプリケーションおよびプロジェクトを AppScan Source に事前に登録しなければなりません。 詳しくは、 144 ページの『AppScan Source に公開するためのアプリケーションおよびプロジェクトの登録』を参照してください。デフォルトでは、未登録のアプリケーションまたはプロジェクトの評価を公開しようとすると、そのアプリケーションまたはプロジェクトをその時点で登録するよう指示するプロンプトが出されます (登録するには「登録」権限が必要です)。

注**:** 個々のファイルのスキャン結果として作成される評価を公開することはできません。

制約事項**:** 複数のアプリケーションまたはプロジェクトをスキャンする場合、各スキャン対象項目の評価を含む親ノードが「自分の評価」ビューに作成されます。この場合、個々の子評価を管理することはできません (例えば、子評価を個々に削除したり、公開したりすることはできません)。複数のアプリケーションまたはプロジェクトを同時にスキャンする場合は、評価をグループ (親ノード) としてのみ管理することができます。

手順

1. 「トリアージ」パースペクティブで現在開いている評価を公開するには、メイン・ワークベンチ・メニューで、「ファイル」 > 「**AppScan Source** への評価の公開」を選択します。
2. 「自分の評価」ビュー内の評価を公開するには、評価を選択してビューの「**AppScan Source** への評価の公開」ボタンをクリックするか、評価を右クリックして「**AppScan Source** への評価の公開」を選択します。

タスクの結果

評価が保存されると、AppScan Source for Analysis は、ソース・ファイルなどの項目を参照するための絶対パスを評価ファイルに書き込みます。ディレクトリー構造が異なる別のコンピューターでファイルを共有するときに、これらの絶対パスが障害になることがあります。移植可能な評価ファイルを作成できるようにするには、変数を作成する必要があります (109 ページの『変数の定義』または 154 ページの『公開時および保存時の変数の定義』を参照してください)。

「自分の評価」ビューにリストされた評価は、公開されると、「公開済み」列にアイコンが表示されます。 また、その評価は「公開された評価」ビュー内に表示されます。これは、AppScan Source データベース に公開された評価を表示するフィルター駆動型のビューです。このビューは、フィルター基準と一致する評価のみが表示されるように設定することができます。 例えば、1,000 件の評価が公開されていて、自分が公開した評価のみを表示する場合には、フィルター基準として「公開者別」を指定し、その値として「現在のユーザー」 または自分のユーザー名を指定したフィルターを作成できます。

「公開された評価」ビューでのフィルターの設定

フィルターを使用すると、「公開された評価」ビューに表示される評価の数を制限することができます。

手順

1. 「公開された評価」ビューで、ツールバーの「フィルターの設定」ボタンをクリックします。
2. 目的のフィルター基準のチェック・ボックスを (1 つまたは複数) 選択します。
   - アプリケーション別: 評価を表示する対象のアプリケーションを選択します。 複数のアプリケーションに対して生成された評価も、指定されたアプリケーションがその評価対象の一部だった場合には表示されます。
   - 公開者別: 現在のユーザーによって公開された評価を表示するようビューを設定します。また、評価を公開したユーザーを指定して表示することもできます。
   - 日付からの期間別: 現在の日付を基準とした日付範囲 (単位は時間、日、週、月、または年) を指定します。「日付からの期間別」または「日付範囲別」を選択できますが、両方を選択することはできません。
   - 日付範囲別: ビュー内に表示する評価日付の範囲を指定します。 「日付からの期間別」または「日付範囲別」を選択できますが、両方を選択することはできません。
3. 「**OK**」をクリックして、フィルターを設定します。

タスクの結果

フィルター基準を適用した後で「フィルターの更新」をクリックすると、ビューが更新されて、フィルターを最後に適用した後に追加または削除された評価が反映されます。 「フィルターの消去」をクリックすると、既存のフィルターが削除され、すべての評価が表示されます。

公開された評価の AppScan Source からの削除

AppScan Source に評価を公開した場合に、「公開された評価」ビューのアクションを使用して、公開済みの評価を削除することができます。

手順

1. 「公開された評価」ビューで、削除する評価を選択します。キーボードの Ctrl キーまたは Shift キーを使用して、複数の評価を選択することもできます。
2. ビューのツールバーにある「評価の削除」ボタンを選択するか、または選択項目を右クリックしてメニューから「評価の削除」を選択します。

AppScan Enterprise Console への評価の公開

ご使用の AppScan Enterprise Server が Enterprise Console オプションを指定してインストールされている場合は、Enterprise Console に評価を公開することができます。 Enterprise Console は、レポート機能、問題管理、トレンド分析、ダッシュボードなど、評価に関する作業を行うためのさまざまなツールを備えています。

このタスクについて

Enterprise Console に評価を公開する前に、AppScan Enterprise Console の設定ページでサーバー設定を構成する必要があります。 設定については、 106 ページの『AppScan Enterprise Console の設定』を参照してください。

注**:** AppScan Source および AppScan Enterprise の一部のバージョンでは、AppScan Source から AppScan Enterprise Console に評価を公開するために 2 つの製品のバージョンとリリース・レベルが一致していなければなりません。 評価を公開する場合に、互換性がある AppScan Source と AppScan Enterprise のバージョンを確認するには http://www.ibm.com/support/docview.wss?uid=swg21975211 を参照してください。

制約事項**:** 複数のアプリケーションまたはプロジェクトをスキャンする場合、各スキャン対象項目の評価を含む親ノードが「自分の評価」ビューに作成されます。この場合、個々の子評価を管理することはできません (例えば、子評価を個々に削除したり、公開したりすることはできません)。複数のアプリケーションまたはプロジェクトを同時にスキャンする場合は、評価をグループ (親ノード) としてのみ管理することができます。

手順

1. Enterprise Console に 1 つ以上の評価を公開するには、以下のいずれかの方法を使用します。
   a. 「自分の評価」ビューで 1 つ以上の評価を選択し、「**AppScan Enterprise Console** に評価を公開」をクリックします。
   b. 「自分の評価」ビューで 1 つの評価を (または複数の評価を選択して) 右クリックし、「**AppScan Enterprise Console** に評価を公開」メニュー項目を選択します。
   c. 評価が表示されている場合は、メインメニューから「ファイル」 > 「**AppScan Enterprise Console** に評価を公開」を選択します。
2. 「AppScan Enterprise Console に公開」ダイアログ・ボックスでは、以下のようにします。
   a. 評価を関連付ける AppScan Enterprise Console アプリケーションを指定します。 これは、AppScan Enterprise Server バージョン 9.0.3 以上に接続する場合は必須です (こちらの説明に従って要件を無効にしている場合を除きます)。 AppScan Enterprise Server の旧バージョンに接続する場合、アプリケーションの関連付けは任意指定です。 旧バージョンの AppScan Enterprise Server に接続される場合、デフォルトでは、このアプリケーションは、最後に公開先として指定されたアプリケーションが設定されています。前の公開時に指定されたアプリケーションが存在しない場合は、デフォルトではアプリケーションが使用されません。 アプリケーションを指定するには、以下のようにします。
      1) 「アプリケーション」フィールドの「選択」ボタンをクリックします。
      2) 「アプリケーションの選択」ダイアログ・ボックスが開き、AppScan Enterprise Console に既に存在しているすべてのアプリケーションが表

示されます。 AppScan Enterprise Consoleにアプリケーションの属性を表示するには、そのアプリケーションの隣にある「プロファイルの表示」をクリックします。

3) スキャンを関連付けるアプリケーションを選択するか、「新規アプリケーションの作成」をクリックしてこの目的のために新規アプリケーションを作成します。 このリンクをクリックすると AppScan Enterprise Console が開き、新規アプリケーションを作成できます。 新規アプリケーションの属性をいったん保存すると、「アプリケーションの選択」ダイアログ・ボックスが自動的に更新され、選択肢に新規アプリケーションが含まれるようなります (自動的に新規アプリケーションが含まれない場合は「更新」をクリックしてください)。

ヒント**:** 「アプリケーションの選択」ダイアログ・ボックスでは、フィルター・フィールドを使用してアプリケーションのリストを絞り込むことができます。 このフィールドに入力すると、アプリケーションのリストにフィルターが自動的に適用されます。ワイルドカードとしてアスタリスク (*) および疑問符 (?) を使用できます。アスタリスクは、連続した複数の文字 (文字数がゼロの場合も含む) を表し、疑問符は単一の文字を表します。

4) アプリケーションを選択したら、「**OK**」をクリックします。

b. 必須: 「名前」フィールドでは、AppScan Enterprise Consoleに保存する際に使用する評価の名前を指定します。

c. オプション: バージョン **9.0.3** より前の **AppScan Enterprise Server** に接続された場合**:** 「フォルダー」フィールドを使用して、公開先のロケーションを設定します。デフォルトの場合、このロケーションは、最後に公開先として使用されたロケーションに設定されます。 まだ評価が公開されていない場合は、デフォルトの AppScan Enterprise Console フォルダーが選択されます (これは、AppScan Enterprise Console の設定ページで指定されたユーザー ID に対するデフォルトのフォルダーです)。 別のフォルダーを公開先として選択するには、「フォルダー」フィールドで「選択」ボタンをクリックしてから、任意のフォルダーを選択します (選択できるのは、自分が公開権限を持っているフォルダーだけです)。 公開先のフォルダーを選択できない場合は、「更新」をクリックして、サーバー上で行われた変更内容をフォルダー・ツリーに反映させてください。

3. 「公開」をクリックします。

タスクの結果

評価が保存されると、AppScan Source for Analysis は、ソース・ファイルなどの項目を参照するための絶対パスを評価ファイルに書き込みます。ディレクトリー構造が異なる別のコンピューターでファイルを共有するときに、これらの絶対パスが障害になることがあります。移植可能な評価ファイルを作成できるようにするには、変数を作成する必要があります (109 ページの『変数の定義』または 154 ページの『公開時および保存時の変数の定義』を参照してください)。

評価が公開された後で、AppScan Enterprise (Enterprise Console) へのリンクが情報メッセージによって提供されます。 このリンクをクリックすると、デフォルトの外部 Web ブラウザーでポータル・ページが開きます。

ヒント: 公開できない場合は、Enterprise Console サーバーが実行中かどうか、ブラウザーを使用してそのコントロール・センターの URL にアクセスできるかどうかを確認してください (AppScan Enterprise Console の設定ページで指定したものと同じ「**Enterprise Console の URL**」を使用してください)。

注:

- 大規模な評価は、ポータルに表示されるまでに時間がかかることがあります。公開の後でエラー・メッセージを受け取っていないのにレポートがポータルに表示されない場合は、管理者に問い合わせてください。
- Enterprise Console で現在処理されている評価と同じ名前を持つ評価を公開しようとすると、失敗します。 また、ある評価が処理された後でそれと同じ名前を持つ評価を公開した場合、2 番目の評価が最初の評価を上書きします (Enterprise Console を事前に構成しておくと、類似した名前を持つ複数のレポートに対する傾向分析を実行できます)。 評価の処理が終了したかどうかを判別するには、Web ブラウザーで Enterprise Console のコントロール・センターにアクセスし、該当するユーザー・フォルダーにナビゲートして、レポートの状態を確認します。
- AppScan Source では、プロキシー設定を使用するように構成された Enterprise Console インスタンスへの公開はサポートされていません。 プロキシー設定を使用するインスタンスに公開しようとすると、エラーが発生します。

重要:

AppScan Source バージョン 9.0.3.4 にアップグレードした場合、以下の変更点があります。

- 評価を AppScan Enterprise Console に公開する場合、その評価を AppScan Enterprise のアプリケーションと関連付けることが必要になりました (AppScan Enterprise Server バージョン 9.0.3 以上を実行中の場合)。 そのため、自動化スクリプトは、アプリケーションの関連付けが含まれていない場合に失敗する可能性があります。AppScan Enterprise Server では、AppScan Enterprise Server アプリケーション・セキュリティー・リスク管理機能を使用する場合、アプリケーションの関連付けが必須です。http://www.ibm.com/support/knowledgecenter/SSW2NF_9.0.3/com.ibm.ase.help.doc/topics/c_overview.html を参照してください。
- さらに、AppScan Enterprise URL からポートを除去する必要があります。
  1. AppScan Source for Analysis で、「編集」 > 「設定」をクリックします。
  2. AppScan Enterprise Console の設定で、「**Enterprise Console URL**」フィールドからポートを除去します。
- 評価の公開後、その評価を参照できるのは AppScan Enterprise モニター・ビューのみになります (旧リリースでは、AppScan Enterprise スキャン・ビューで評価を参照できました)。このビューへの移行は、http://www.ibm.com/support/knowledgecenter/SSW2NF_9.0.3/com.ibm.ase.help.doc/topics/t_workflow_for_applications.html に説明されています。

これは、Common Access Card (CAC) 認証を使用する場合に AppScan Enterprise Server へ公開するために必要な、AppScan Source と AppScan Enterprise Server の間の通信プロトコルを変更した結果、生じたものです。

CAC 認証が有効な場合に評価を AppScan Enterprise Server に公開したくない場合、または Enterprise Server アプリケーション・セキュリティー・リスク管理機能を利用したくない場合、以下の方法で前の通信プロトコルに戻すことができます。

1. <data_dir>¥config¥ounce.ozsettings (<data_dir> は、ご使用の AppScan Source プログラム・データの場所です。詳しくは、 360 ページの『インストールとユーザー・データ・ファイルの場所』 を参照してください。)) を開きます。
2. このファイルで、以下の設定を見つけます。

```
<Setting
  name="force_ase902_assessment_publish"
  value="false"
  default_value="false"
  description="Use ASE 9.0.2-style assessment publish"
  display_name="Use ASE 9.0.2-style assessment publish"
  type="boolean"
  read_only="true"
  hidden="true"
/>
```

3. この設定で、value="false" を value="true" に変更し、ファイルを保存します。
4. 評価の公開元の AppScan Source 製品を再始動します。

この設定が value="true" に設定されている場合、以下のようになります。

- 公開時に評価を AppScan Enterprise のアプリケーションに関連付けた場合、評価はモニター・ビューとスキャン・ビューで参照可能です。
- 公開時に評価をアプリケーションに関連付けていない場合、評価はスキャン・ビューで参照可能です。
- CAC 認証が有効なときは評価を AppScan Enterprise Server に公開できません。

詳しくは、http://www.ibm.com/support/docview.wss?uid=swg21993010を参照してください。

AppScan Enterprise Console の設定

ご使用の AppScan Enterprise Server が AppScan Enterprise Console オプションを指定してインストールされている場合は、Enterprise Console に評価を公開することができます。 Enterprise Console は、レポート機能、問題管理、トレンド分析、ダッシュボードなど、評価に関する作業を行うためのさまざまなツールを備えています。

この機能を有効にするには、AppScan Enterprise Console の設定ページで必要な設定を行います。 Enterprise Console の公開を有効にする前に、このページのすべてのフィールドに有効な値を入力する必要があります。

- 「ユーザー **ID**」フィールド: AppScan Enterprise Server ユーザー ID (ご使用の AppScan Source ユーザーの代わりに公開するために作成したユーザー ID) を入力します。
  - AppScan Enterprise Server が Windows 認証を使用するように構成されている場合、Enterprise Console への接続に使用するドメイン名とユーザー名を入力します。ドメイン名とユーザー名は ¥ で区切ります (例えば、`my_domain¥my_username`)。
  - AppScan Enterprise Server が LDAP を使用して構成されている場合、Enterprise Console への接続に使用するユーザー名を入力します。
  - Windows の場合、ご使用の AppScan Enterprise Server で Common Access Card (CAC) 認証が有効にされている場合は、管理者の CAC 共通名をリストから選択します。

  少なくとも、QuickScan ユーザーでなければなりません。バージョン 9.0.3 より前の AppScan Enterprise Server に接続されている場合、Enterprise Server 上に独自のユーザー・フォルダーがなければなりません。
- 「パスワード」フィールド: このフィールドは、ご使用の AppScan Enterprise Server 認証方式がユーザー ID とパスワードである場合にのみ使用可能です。Enterprise Console へのログインに使用するパスワードを入力します (入力されたユーザー名のパスワード)。
- 「**Enterprise Console** の **URL**」フィールド: Enterprise Console の Web アプリケーションへのアクセスに使用する URL を入力します。

  この URL の形式は次のとおりです。

  ```
  http(s)://<hostname>:<port>/ase
  ```

  ここで、`<hostname>` は、Enterprise Console がインストールされているマシンの名前、`<port>` は、コンソールが実行されているポートです (デフォルトの `<port>` は `9443` です)。この URL の例は、`https://myhost.mydomain.ibm.com:9443/ase` のようになります。

  注:
  - 「**Enterprise Console** の **URL**」が既に設定されている場合は、このフィールドを変更する必要はありません。
  - 「**Enterprise Console** の **URL**」フィールドを設定可能にするには、AppScan Source に「**AppScan Enterprise** 設定の管理」許可を使用してサインインする必要があります。 ユーザー・アカウントと許可については、製品のインフォメーション・センターの『管理』セクション、または「*IBM Security AppScan Source* インストールと管理のガイド」の『*AppScan Source* の管理』セクションを参照してください。
  - 「ユーザー **ID**」と「パスワード」は AppScan Source クライアント (AppScan Source for Analysis など) が稼働しているマシンに格納されますが、「**Enterprise Console** の **URL**」は Enterprise Server (これはリモート・マシン上に存在している場合があります) に格納されます。 リモート・マシンから (例えば `getaseinfo` コマンドを発行して) ユーザー名とパスワードの情報にアクセスすることはできません。

– AppScan Source では、プロキシー設定を使用するように構成された AppScan Enterprise Console インスタンスへの公開はサポートされていません。 プロキシー設定を使用するインスタンスに公開しようとすると、エラーが発生します。

設定が完了した後で、「接続のテスト」をクリックして、Enterprise Console サーバーへの接続が有効であることを確認することを強くお勧めします。

ヒント**:** 接続テストが失敗した場合は、Enterprise Console サーバーが実行中かどうか、およびブラウザーを使用して製品のコントロール・センター URL にアクセスできるかどうかを確認してください (上記で指定したのと同じ「**Enterprise Console** の **URL**」を使用してください)。

評価の保存

始める前に

重要**:** 評価を保存するには、「評価の保存」権限が必要です。権限の設定について詳しくは、「*IBM Security AppScan Source* インストールと管理のガイド」を参照してください。

このタスクについて

評価は、ローカルに保存して、いつでも開くことができます。デフォルトの場合、評価は `.ozasmt` というファイル拡張子で、オペレーティング・システムのホーム・ディレクトリーに保存されます (例えば Windows の場合、このディレクトリーは `C:¥Documents and Settings¥Administrator¥` などになります)。

手順

1. 「トリアージ」パースペクティブで現在開いている評価を保存するには、メイン・ワークベンチ・メニューで「ファイル」 > 「評価の保存」を選択するか、「ファイル」 > 「評価に名前を付けて保存」を選択します。「評価に名前を付けて保存」アクションを選択すると、保存する評価の場所とファイル名を指定することができます。
2. 「自分の評価」ビュー内の評価を保存するには、保存したい評価を選択し、ビューの「評価の保存」ボタンまたは「評価に名前を付けて保存」ボタンをクリックするか、保存したい評価を右クリックし、「評価の保存」または「評価に名前を付けて保存」をクリックします。

タスクの結果

評価が保存されると、AppScan Source for Analysis は、ソース・ファイルなどの項目を参照するための絶対パスを評価ファイルに書き込みます。ディレクトリー構造が異なる別のコンピューターでファイルを共有するときに、これらの絶対パスが障害になることがあります。移植可能な評価ファイルを作成できるようにするには、変数を作成する必要があります (109 ページの『変数の定義』または 154 ページの『公開時および保存時の変数の定義』を参照してください)。

評価の自動保存

デフォルトで、スキャンは自動的に <data_dir>¥scans (<data_dir> は、ご使用の AppScan Source プログラム・データの場所です。詳しくは、 360 ページの『インストールとユーザー・データ・ファイルの場所』 を参照してください。) に 3 日間保存されます。この動作は、<data_dir>¥config¥scanner.ozsettings 内の assessment_auto_save、assessment_auto_save_location、および assessment_auto_save_stale_period 設定により決定されます。

- assessment_auto_save 設定が true に設定されていると、評価の完了時に評価が自動的に保存されます (ユーザーには「評価の保存」権限が必要です)。
- assessment_auto_save_location 設定は、評価の保管場所を決定します。デフォルトで、評価は <data_dir>¥scans に保存されます。この場所を変更する場合は、value 属性を変更先のディレクトリーに設定します。例えば、場所を C:¥myFolder に設定する場合は、属性を value="C:¥myFolder" に設定します。
- assessment_auto_save_stale_period 設定は、評価を assessment_auto_save_location で保持する日数を決定します。この設定は、value 属性を使用して変更できます。例えば、この属性を value="10" に設定すると、保存された評価は 10 日間後に assessment_auto_save_location から除去されます。

「自分の評価」からの評価の削除

「自分の評価」ビューから評価を削除しても、ローカル・ファイル・システムからは削除されません。ビューから評価を削除した場合、その評価を「評価を開く」アクションで再度追加することができます。

このタスクについて

制約事項**:** 複数のアプリケーションまたはプロジェクトをスキャンする場合、各スキャン対象項目の評価を含む親ノードが「自分の評価」ビューに作成されます。この場合、個々の子評価を管理することはできません (例えば、子評価を個々に削除したり、公開したりすることはできません)。複数のアプリケーションまたはプロジェクトを同時にスキャンする場合は、評価をグループ (親ノード) としてのみ管理することができます。

手順

1. 「自分の評価」ビューで、削除する評価を選択します。キーボードの Ctrl キーまたは Shift キーを使用して、複数の評価を選択することもできます。
2. ビューのツールバーにある「自分の評価からの削除」ボタンを選択するか、または選択項目を右クリックしてメニューから「自分の評価からの削除」を選択します。

変数の定義

評価またはバンドルを保存するとき、または評価を公開するときに、絶対パスを置換する変数を作成するように AppScan Source for Analysis から提示されることがあります (変数がない場合、AppScan Source for Analysis は、ソース・ファイルなどの項目を参照するための絶対パスを評価ファイルに書き込みます)。絶対パスに

代わる変数を構成すると、複数のコンピューターでの評価の共有が容易になります。 評価を共有する場合は、変数を使用することをお勧めします。

このタスクについて

保存アクションまたは公開アクションを開始する前に変数を作成することができます。その場合は、このトピックの以下の説明に従ってください。あるいは、『公開時および保存時の変数の定義』の手順に従うことにより、保存アクションまたは公開アクションの開始後に変数を作成することもできます。

評価を共有する際の変数の使用例については、 155 ページの『例: 変数の定義』を参照してください。

手順

1. メインメニューで、「編集」 > 「設定」を選択します。 「設定」ダイアログ・ボックスで、「変数の変更」を選択します。
2. 「変数の変更」設定ページで、「変数の追加」 ボタンをクリックします。
3. 変数の名前を入力し、変数で置換するファイルの場所を参照します (作成した変数には、AppScan Source for Analysis によって前後にパーセント記号 (%) が挿入されます)。
4. 評価内の他のすべての参照項目について、上記の手順を繰り返します (例えば、複数の場所のソースが評価内で参照されている場合は、それぞれの場所について変数を追加します)。
5. 設定ページでは、「変数の変更」ボタンを使用して変数を編集でき、「変数の削除」ボタンを使用して変数を削除できます。
6. 変数の定義が完了したら、「**OK**」をクリックします。

公開時および保存時の変数の定義

評価を保存または公開しようとすると、その評価内のすべての絶対パスが AppScan Source for Analysis によって検出されます。これらの絶対パスに対応する変数が作成されていない場合、変数の作成プロンプトが表示されます。

このタスクについて

109 ページの『変数の定義』の説明に従い、保存アクションまたは公開アクションを開始する前に、変数を作成することができます。または、このトピックの手順に従い、保存アクションまたは公開アクションを開始してから変数を作成することもできます。

評価を共有する際の変数の使用例については、 155 ページの『例: 変数の定義』を参照してください。

手順

1. 保存アクションまたは公開アクションを開始してから、「絶対パスを検出しました」というメッセージで「はい」をクリックします。
2. AppScan Source for Analysis により、データを囲む一連のパスが「変数の定義」ダイアログ・ボックスに表示されます。
3. ディレクトリーを選択し、「変数の追加」をクリックします。

4. 評価内の他のすべての参照項目について、上記の手順を繰り返します (例えば、複数の場所のソースが評価内で参照されている場合は、それぞれの場所について変数を追加します)。
5. 「変数の定義」ダイアログ・ボックスの「変数の変更」ボタンと「変数の削除」ボタンを使用して、変数の編集や削除を行うこともできます。
6. 「**OK**」をクリックして、保存アクションまたは公開アクションを完了します。

例: 変数の定義

評価データを共有するには、適切な変数を定義する必要があります。 このトピックの例は、変数がなぜ必要かを示します。

ユーザー Joe は、コンピューター A 上でスキャンを実行します。すべてのソース・コードは、ディレクトリー C:¥dev¥my_code の下にあります。 Joe は、自分のスキャン結果をファイルに保存し、それを Bill と共有しようとします。 Bill は、コンピューター B を使用します。Joe がスキャンしたのと同じコードは、ディレクトリー C:¥code¥bills_code の下にあります。変数を使用しないと、評価ファイルは、C:¥dev¥my_code で開始される絶対パスを使用して、すべてのソース・ファイルを参照します。 Bill がこの評価ファイルをコンピューター B 上で開いた場合、コンピューター B 上ではソース・ファイルが C:¥code¥bills_code の下にあるため、AppScan Source for Analysis はソース・ファイルを見つけることができません。

解決方法

Joe と Bill の 2 人は、2 人ともソース・コードのルートを指す変数を作成する必要があります。 Joe は、AppScan Source for Analysis 内で SRC_ROOT という名前の変数を作成し、その変数に C:¥dev¥my_code の値を指定します。 この変数は、Joe の AppScan Source for Analysis インストール済み環境のローカル変数です。次に Joe は、変数名 (SRC_ROOT) と、その変数が指す場所を Bill に伝えます。次に、Bill は、自分の AppScan Source for Analysis 内で SRC_ROOT という名前の変数を作成し、その変数に C:¥code¥bills_code の値を指定します。 Joe が自分のスキャンを保存すると、変数 SRC_ROOT がパス C:¥dev¥my_code を置き換えます。Joe から受け取った評価ファイルを Bill が開くと、C:¥code¥bills_code が SRC_ROOT 変数に代入されます。

第 5 章 トリアージおよび分析

類似した検出結果をグループ化することによって、セキュリティー・アナリストまたは IT 監査員が、ソース・コードの問題を区分してトリアージを行うことができます。 このセクションでは、AppScan Source 評価のトリアージを行い、結果を分析する方法について説明します。

コードをスキャンすると、スキャン結果 (検出結果) が表示されます。 トリアージは、検出結果を評価し、それらの解決方法を決定するプロセスです。 ただし、この目的の達成に必要なステップは、検出結果の総数、特定のセキュリティー上の懸案事項、アプリケーションのリスク評価などの、複数の要因によって決まります。 検出結果が有意のセキュリティー問題を表しているかどうかを決定するだけでなく、トリアージでは、必要な場合に検出結果の属性 (重大度、タイプ、分類) を変更することも含まれます。

トリアージ方針は、望ましい順序で、望ましい期間のうちに目標を確実に達成するために重要です。 トリアージは、検出結果のサブセットを評価して、1 回の反復ごとに各サブセットの処理を決定するという反復的な方法で、最も適切に達成することができます。 トリアージの反復の定義方法を決定するには、多くの有効なアプローチがあります。 1 つのアプローチとしては、全体的な重大度に基づいて、高リスクの検出結果のサブセットを作成します。 潜在的リスクが最も高い検出結果から始めて、最も潜在的リスクが低い検出結果へという順序で解決することができます。 もう 1 つのアプローチは、「SQL 注入」や「検証が必要」などのセキュリティー上の懸案事項ごとにサブセットを定義する方法です。

通常、セキュリティー・アナリストまたは IT 監査員がトリアージを実行します。アナリストまたは監査員は、コード変更を必要とする検出結果を障害追跡システムに送信してから、開発者に修復を依頼します。開発者がトリアージを行って問題を解決するケースもあります。

トリアージ・フェーズでは、以下のことを行うことができます。

- 特に関心のある脆弱性タイプの検出結果を確認する
- 特定のカテゴリーの API を表示する
- 異なる評価での検出結果を比較する
- 特定の検出結果をフィルタリングまたは除外する
- 検出結果の重大度または脆弱性タイプを変更する
- 「要確認」およびスキャン範囲の検出結果を「確定」に昇格する
- 検出結果に注釈を付ける
- 障害追跡システムに障害を送信するか、検出結果を他のユーザーに E メールで送信する

AppScan Source には、さまざまなトリアージ方針を使用して結果を分析するために必要なツールのすべてが用意されています。フィルタリングは、特定の 1 回のトリアージの反復作業で処理される検出結果のみを表示するための手段を提供します。 反復方針が重大度と分類によるものである場合は、「脆弱性マトリックス」ビ

ューから検出結果をフィルタリングすることができます。 反復方針が脆弱性タイプによるものである場合は、「評価の概要」ビューからフィルタリングすることができます。AppScan Source for Analysis では、複雑な反復的アプローチをサポートするフィルター・エディターも用意されています。

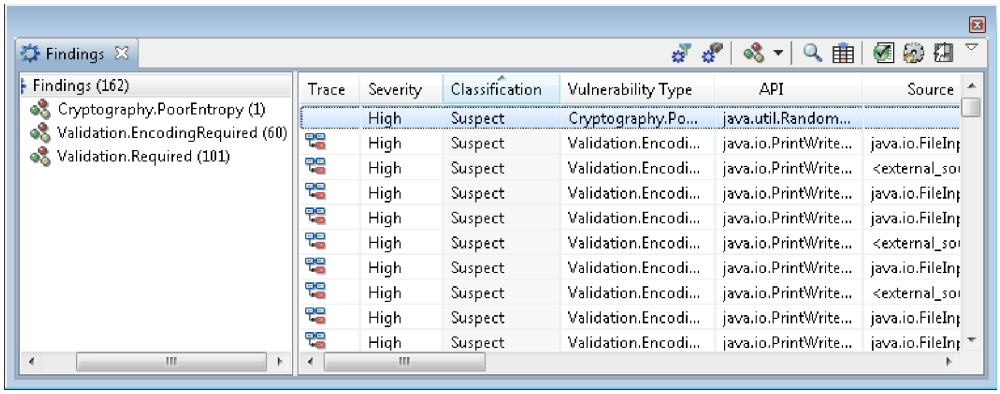
一度トリアージのアプローチを選択すれば、AppScan Source for Analysis が検出結果の処理をサポートします。

- 個別の検出結果または検出結果のコレクションを除外する
- 検出結果の詳細 (タイプ、重大度、分類) を変更する
- バンドル (検出結果のグループ化メカニズム) を作成する
- 「差分評価」ビューで評価を比較する

検出結果の表示

「検出結果」ビュー、および検出結果が含まれるすべてのビューで、スキャンごとに、「検出結果ツリー」(評価基準の階層グループ) および検出結果表が表示されます。 検出結果ツリーで選択されている項目により、表に表示される検出結果が決まります。

検出結果ツリーのルートを選択すると、すべての検出結果が表に表示されます。また、グループ化タイプを選択すると、検出結果のうち該当のタイプのみが表示されます。



AppScan Source for Analysis には、以下に示すようなさまざまなグループ分けによって検出結果が表示されます。

- 脆弱性タイプ
- 分類
- ファイル
- ソース
- シンク
- **API**
- バンドル
- **CWE**
- 表

注: 分類および重大度は、デフォルトでは降順にソートします。その他のすべての列では、昇順にソートします。

これらの列は、検出結果表に表示されます。

表 *11.* 検出結果表

列見出し	説明
トレース	この列のアイコンは、逸失シンクまたは既知のシンクのトレースが存在することを示します。
重大度	• 高: データの機密性や保全性、可用性、および/または処理リソースの保全性や可用性にリスクをもたらします。 重大度の高い状態は、即時に修復されるように優先順位付けする必要があります。 • 中: データ・セキュリティーおよびリソース保全性にリスクをもたらしますが、攻撃の影響は比較的受けにくい状態です。 重大度が中の状態は、可能であれば検討し修復します。 • 低: データ・セキュリティーおよびリソース保全性にもたらすリスクは最小です。 • 情報: 検出結果自体は、セキュリティーを侵害するわけではありません。これは、コードで使用されているテクノロジー、アーキテクチャー特性、またはセキュリティー・メカニズムについて説明するものです。
分類	検出結果のタイプ: 「確定」または「要確認」セキュリティー検出結果 - または「スキャン範囲」検出結果。 **注:** 場合によっては、「なし」の分類を使用して、セキュリティー検出結果でもスキャン範囲検出結果でもない分類が示されることがあります。
脆弱性タイプ	脆弱性カテゴリー (`Validation.Required` または `Injection.SQL` など)。
API	脆弱な呼び出し。API と、それに渡される引数の両方を表示します。
ソース	ソースはプログラムへの入力で、ファイル、サーブレット要求、コンソール入力、ソケットなどがあります。 多くの入力ソースの場合、コンテキストと長さについてはバインドされていないデータが返されます。 チェックされていない入力については、汚染されているものと見なされます。

表 *11.* 検出結果表 *(*続き*)*

列見出し	説明
シンク	シンクは、データの書き込み先となる任意の外部フォーマットです。 シンクの例としては、データベース、ファイル、コンソール出力、ソケットなどがあります。データをチェックせずにシンクに書き込むと、重大なセキュリティー脆弱性となる可能性があります。
ディレクトリー	スキャンされたファイルの絶対パス。
ファイル	セキュリティー検出結果またはスキャン範囲検出結果が発生するコード・ファイルの名前。検出結果内のファイル・パスは、スキャンされたプロジェクト作業ディレクトリーからの相対パスです。
呼び出し側メソッド	脆弱な呼び出しが行われている関数 (またはメソッド)。
行	脆弱な API を含むコード・ファイル内の行番号。
バンドル	この検出結果を含むバンドル。
CWE	コミュニティーが作成した、共通のソフトウェア脆弱性の辞書の ID およびトピック (共通脆弱性タイプ一覧 (CWE) のトピック)。

注**:** AppScan Source でソースを見つけることができない検出結果を選択すると、ソース・ファイルが見つからない場合にプロンプトを出すかどうかを尋ねるダイアログ・ボックスが提示されます。 「はい」を選択すると、ソース・ファイルが見つからない検出結果を選択するたびにプロンプトが出されます。 「いいえ」を選択すると、プロンプトが出されません。 現在の評価が開いている間は、この設定は存続します。 この設定は、評価を開くたび、または AppScan Source を終了した場合にリセットされます。

AppScan Source トリアージ・プロセス

トリアージ・プロセスには、バンドル、フィルター、および除外による検出結果の操作と、評価結果の比較が含まれます。

フィルター

フィルター は、特定の特徴を持つ検出結果を定義するルールのセットです。 フィルターを使用すると、これらの検出結果を動的に表示することができ、また、類似した検出結果をトリアージすることができます。

フィルターは、共有 またはローカル のいずれかです。

- 共有フィルターは AppScan サーバー上にあります。そのサーバーに接続していれば、誰でもそのフィルターを使用できます。
- ローカル・フィルターは、ローカル・コンピューター上にあります。

バンドル

バンドル は、アプリケーションと共に保管される、個別の検出結果の名前付きコレクションです。バンドルは、検出結果を選択し、その検出結果を新規または既存のバンドルに追加するだけで作成されます。

類似した検出結果をバンドルにまとめることによって、セキュリティー・アナリストが、ソース・コードの問題を区分してトリアージを行うことができます。 トリアージおよび分析のプロセスの一部として、バンドルを障害追跡システムに送信するか、レビューを行うために検出結果を開発者に E メールで送ることができます。

除外

除外 によって、スキャンから検出結果が除外されます。 AppScan Source には、除外されたバンドルが組み込まれており、これには、ユーザーが除外した検出結果が格納されます (例えば、これらの検出結果には解決策が必要ないという理由で除外)。

注: 評価結果から除外された検出結果は、アプリケーションまたはプロジェクトのメトリックの計算には寄与しません。

変更された検出結果

変更された 検出結果とは、脆弱性タイプ、重大度、または分類が変更された検出結果です。検出結果に注を追加した場合も、検出結果は変更済みと見なされます。

評価の比較

評価は、「差分評価」アクションを使用して AppScan Source for Analysis で比較されます。2 つの評価が比較されると、両者の差分は「差分評価」ビューに表示されます (これは、「自分の評価」ビューと「検出結果」ビューを組み合わせたものに似ています)。

注: 評価が比較されるとき、フィルターおよびバンドルは無視されます。

トリアージ例

この例では、セキュリティー・アナリストが使用する AppScan Source トリアージ・ワークフローについて説明します。 トリアージ・ワークフローは、ビジネス上のニーズによって変わる場合があります。

会社のセキュリティー・アナリストである Jones さんが、スキャン結果のトリアージを行いたいと考えています。 彼は類似した検出結果をグループ化して優先順位を付けてから、解決してもらうために該当する開発者に送信しようとしています。

Jones さんは、まず、アプリケーションのソース・コードをスキャンし、「トリアージ」パースペクティブで評価を開きます。スキャンにより、約 2,000 個の検出結果が生成されました。それらすべてを「検出結果」ビューで確認できます。ただし、Jones さんは、先に結果の概要を知りたいと考え、重大度タイプおよび検出結果タイプ (「セキュリティー」または「スキャン範囲」) ごとの内訳を示す「脆弱性マトリックス」ビューを開きます。「スキャン範囲」検出結果と「要確認」セキュリティー検出結果は、リスクを判断するためにさらに調査を必要とします。

Jones さんが「脆弱性マトリックス」を見ると、重大度の高い決定的セキュリティー検出結果が 8 件あります。8 件の決定的検出結果を示すマトリックス・ボックスをクリックすると、自動的にフィルターが作成され、「検出結果」ビューが最新表示されて、これら 8 件の重大な問題のみが表示されます。Jones さんは、これらの問題をバグとして扱うことに決めます。 8 個すべてを選択し、会社の障害追跡システムに送信します。 次に、「脆弱性マトリックス」からフィルターをリセットします。

Jones さんは、次に、「評価の概要」ビューに注意を向けます。 そして、2,000 個の検出結果が、6 種類を超える脆弱性タイプで構成されていることに気付きます。Jones さんは検証の問題に集中することに決めて、「評価の概要」ビューから、もう 1 つのフィルターを作成します。 グラフで `Validation.EncodingRequired` および `Validation.Required` をクリックし、「検出結果」ビュー内の検出結果の数を、約 500 個に減らします。

検出結果が減ったとはいえ、500 個ではまだトリアージを行うのは困難です。Jones さんは、さらに結果をフィルタリングすることにします。 「フィルター・エディター」ビューで、「評価の概要」から作成されたフィルターを、高い重大度を要件として補強します。 これで、検出結果表に表示される項目は 150 個になりました。

ファイル名でソートしたとき、検出結果の一部が、サード・パーティーのライブラリーのコード内で検出されていることに気付きます。 Jones さんは、このライブラリーの使用は他の部分から分離されていることが分かっており、そのセキュリティー問題に対処するつもりはありません。 そこで彼は、これらの検出結果を除外します。これにより、「検出結果」ビューおよびメトリックが直ちに更新されます。今後のスキャンでも、これらの検出結果は検出されますが、分離されて、メトリックには反映されません。

Jones さんは、タイプが `Validation.Required` である、重大度の高い要注意セキュリティー検出結果がいくつかあることに気付きます。彼は、検証が行われずにデータが取り込まれていることを認識します。 Jones さんは、これらの検出結果を、要注意から決定的に昇格させることに決めます。この変更を行っている間に、変更について説明する注を追加することに決め、次に、修復の優先順位を付けるため、または「変更された検出結果」ビューで検討するため、これらの検出結果を自分宛にリマインダーとして E メールで送信します。

次に、Jones さんは再度ファイル名によってソートし、検出結果の一部はバックエンド・サーバー内にあり、一部はユーザー・インターフェース内にあることに気付きます。 すべてのバックエンドの検出結果を選択し、`Backend Server - Validation Required` というラベルを付けた新規バンドルを作成します。 残りの検出結果を選択し、それらを `UI - Validation Required` というラベルを付けたバンドルに入れます。 トリアージは、対象を重大度の高い `Validation.EncodingRequired` タイプに絞り込んで続行されます。

その日の終わりまでに、Jones さんは 1 ダースのバンドルを作成しました。 1 日を通して、グラフ、フィルター、および脆弱性マトリックスを使用して、一度にビュー内で管理可能な数まで検出結果を絞り込みます。これらの個別の検出結果をバ

ンドルに入れる場合もあれば、重要ではない検出結果を除外することもあります。特定の検出結果用に新規バンドルを作成することも、既存のバンドルに検出結果を追加することもあります。

ここで Jones さんは、1 ダースのバンドルを検証します。 Backend Server - Validation Required バンドルおよび UI - Validation Required バンドルを、会社の障害追跡システムに送信して、これらの問題となる分野の開発者に通知します。

Jones さんは、「バンドル」ビューに移動し、Backend Server - Validation Required バンドルを開きます。 「**Backend Server - Validation Required**」というタイトルの新規ビューが開き、バンドルに入れた検出結果のリストが表示されます。 次に、このバンドルを障害追跡システムに送信します。その夜遅く、開発者が Rational ClearQuest にログインして、自分に割り当てられたバグがあることを確認した場合、AppScan Source for Development で検出結果を開くことができます。

Jones さんはその他のバンドルを検証します。 一部のバンドルを障害追跡システムに送信し、その他を同僚に E メールで送ります。 ただし、一部のバンドルには、さらに検討してみると、自分にとってそれほど重要ではないような検出結果が含まれています。 それらの重要性の低い検出結果を、By Design および Irrelevant という 2 つの新規バンドルに移動します。 Jones さんは、これらの検出結果が許容できると判断したため、コードを変更するつもりはありません。 Jones さんは、By Design および Irrelevant の検出結果だけでなく、Cryptography.PoorEntropy のすべての検出結果も自分にとって重要ではないと認識します。 それらの暗号化呼び出しではエントロピーが不十分であることが分かっており、高速なコンピューターであれば 1 週間以内に鍵を割り出す可能性がありますが、そのことは、暗号化してから数時間後にはデータが有用ではなくなってしまうアプリケーションの場合は、重要ではありません。 Jones さんは、これらも削除したいと考えます。

そこで Jones さんは、By Design バンドルおよび Irrelevant バンドルを「プロパティー」ビューの「除外されたバンドル」リストに追加します。また、フィルター・エディターを開いて、脆弱性タイプ Cryptography.PoorEntropy の別のフィルターを作成し、Crypto という名前でフィルターを保存して、Crypto フィルターの動作を「反転 **(Inverted)**」に設定します (「フィルターの選択」ダイアログ・ボックスで、「フィルターの反転 **(Invert filter)**」を選択します)。その上でスキャンを開始し、自宅に帰ります。メトリックでは、次にスキャンが行われるまで、これらの除外が反映されません。

フィルターを使用したトリアージ

AppScan Source for Analysis は、すべての潜在的なセキュリティーの脆弱性について報告するため、中規模から大規模のコード・ベースに適用した場合、膨大な量の検出結果を生成する可能性があります。 スキャンを実行したときに、あるユーザーにとっては重要でない項目が検出結果のリストに含まれることもあります。 「検出結果」ビューから特定の検出結果を削除するために、事前定義フィルターを選択するか、独自のフィルター を作成することができます。フィルターは、どの検出結果をビューから削除するかを決定する基準を指定します。

- 164 ページの『フィルターの概要』
- 164 ページの『フィルター・ルール』

- 167 ページの『フィルターの例』

フィルターの概要

フィルターは、フィルター・ルールによって決定された基準を満たす項目を削除または制限し、トリアージまたはレポート作成を行う際のスキャン結果の管理に役立ちます。フィルターは、ワークフローのガイドに役立ち、セキュリティー・アナリストが検出結果のサブセットの最も重要な領域に集中できるようにします。 例えばコードの調査中に、アナリストは、重大度の低い検出結果を表示しないようにするフィルターを作成することができます。あるいは、アナリストがシステム・ライブラリーの `include` ファイル内の脆弱性の除外を優先する場合も考えられます。フィルターは、これらの項目をビューから除去することができ、個別のファイルや、以前に調査されたファイルを除外することもできます。

フィルターは、スキャン前またはスキャン後に適用できます。

- スキャン前にフィルターを適用するには、プロジェクトまたはアプリケーションのプロパティーにグローバル・フィルターを設定するか、またはフィルターを含むスキャン構成を使用してスキャンを行います。スキャン前にフィルターを適用すると、フィルタリング対象でない検出結果を表示することも、再スキャンせずにフィルターを削除することもできません。
- 各種のビュー (特に、「フィルター・エディター」ビュー) を使用すると、スキャン後にフィルターを適用できます。これらのビューを使用してフィルタリングを行うと、フィルタリングされたすべての項目は、スキャン結果に残りますが、「検出結果」ビューには「フィルターに掛けた検出結果の表示」 ( ) トグルが選択されている場合にしか表示されません。

AppScan Source には、スキャン結果をフィルタリングするために選択できる事前定義フィルターがいくつか含まれています。

フィルターがあると、そのフィルターが除外 となるようにプロパティーを設定することができます。 除外はスキャンに効力を及ぼし、フィルターに一致するすべての検出結果、またはフィルターに一致しないすべての検出結果を除外します。

フィルター・ルール

各フィルターは、検出結果表内の結果から、どの検出結果を制限 (包含) または削除 (除外) するかを定義するルールからなります (トレース・ルールの場合は、トレース・プロパティーに基づいて制限と削除の両方を行うことができます)。

- 「制限 **(Restrict to)**」ルール (包含ルール) は、指定された基準を満たさない検出結果を除外し、検出結果表に表示される結果から、それらの検出結果を削除します。
- 「削除」ルール (除外ルール) は、基準を満たす検出結果をスキャン結果から削除します。 削除ルールは、指定された基準にあてはまる検出結果を除外し、表示される結果から、それらの検出結果を削除します。

フィルター・ルールには次のような特徴があります。

- 重大度: 個別の検出結果の潜在的な影響またはリスクを示します。重大度ルールは制限のみです。

- 高: データの機密性や保全性、可用性、および/または処理リソースの保全性や可用性にリスクをもたらします。 重大度の高い状態は、即時に修復されるように優先順位付けする必要があります。
- 中: データ・セキュリティーおよびリソース保全性にリスクをもたらしますが、攻撃の影響は比較的受けにくい状態です。重大度が中の状態は、可能であれば検討し修復します。
- 低: データ・セキュリティーおよびリソース保全性にもたらすリスクは最小です。
- 情報: 検出結果自体は、セキュリティーを侵害するわけではありません。これは、コードで使用されているテクノロジー、アーキテクチャー特性、またはセキュリティー・メカニズムについて説明するものです。

- 分類**:** このトピックで説明している分類に基づいて、検出結果をフィルターに掛けます。分類ルールは制限のみです。
- 脆弱性タイプ: `BufferOverflow` などの、特定の脆弱性カテゴリーによってフィルタリングします。脆弱性タイプを追加する場合、すべての可能な脆弱性タイプから選択するか、または現在の評価で検出された脆弱性タイプのみから選択することができます。現在の評価で検出された脆弱性タイプから選択する場合は、「値の選択」ダイアログ・ボックスで「開かれている評価の値のみを表示**(&S)**」を選択します。

  可能なすべての脆弱性タイプからの選択は、将来のスキャンのためのフィルターを作成する場合に便利です。すべての脆弱性タイプを表示するには、「開かれている評価の値のみを表示**(&S)**」を選択解除します (開かれている評価がない場合は、デフォルトですべての脆弱性タイプが表示され、「開かれている評価の値のみを表示**(&S)**」チェック・ボックスは使用できません)。
- **API**: 特定の API のすべての脆弱性をフィルタリングします。
- ファイル: 特定のファイルからのすべての脆弱性をフィルタリングします。
- ディレクトリー: 特定のディレクトリーからのすべての脆弱性をフィルタリングします。
- プロジェクト: 特定のプロジェクトからのすべての脆弱性をフィルタリングします。
- トレース: トレース・プロパティーに基づいて検出結果をフィルタリングすることができます (トレース・プロパティーについて詳しくは、 209 ページの『ソースとシンク』を参照してください)。 フィルターには、トレース・プロパティーに基づいて制限と削除の両方を行うトレース・ルールを含めることができます。どちらかのセクション (制限または削除) で「追加」をクリックすると、「トレース・ルール入力」ダイアログ・ボックスが開きます。 そこで、以下の内容を指定できます。
  - ソース:「ソース」セクションの「**API** 正規表現」フィールドで、トレース・ソースを指定するか、複数のソースを表す正規表現を指定します (デフォルトの入力は `.*` であり、これはすべてを返す正規表現またはワイルドカードです)。 正規表現を使用する場合は、「正規表現タイプ」フィールド・メニューでタイプを選択します (デフォルトの正規表現タイプは「**PERL**」です)。正規表現を使用しない場合は、「正規表現タイプ」フィールド・メニューで「完全一致」を選択します。

「**API** 正規表現」の入力が有効な表現である場合は、フィールドの横に緑のチェック・マークのアイコンが表示されます。 入力が有効な表現ではない場合は、フィールドの横に赤い「X」アイコンが表示され、ダイアログ・ボックスの「**OK**」ボタンは無効になります。いずれかのアイコンの上にマウスを移動すると、検証結果に関する詳細が表示されます。 有効ではない表現を入力したが、それを引き続き使用したい場合は、ダイアログ・ボックスの下部にある「上記の検証エラーを無視」チェック・ボックスを選択します。 これにより、表現が空でなければ、ダイアログ・ボックスの「**OK**」ボタンが有効になり、無効な表現の横のアイコンは、緑のチェック・マークに変わり、「検証は無効です」という吹き出しテキストが表示されます。

「ソース・プロパティー」セクションの「**VMAT** プロパティーを追加」ボタンを使用して、メカニズムまたはテクノロジーによりフィルターを詳細化することもできます (VMAT プロパティーについて詳しくは以下を参照)。ただし、脆弱性ごとに制限するためにこの機能を使用すると、脆弱性タイプはソースではなくシンクにより判別されるため、期待通りの効果は得られません。

– シンク:「シンク」セクションで、ソースを指定する場合と同じ方法で、シンクをフィルターとして追加することができます。

特定の脆弱性タイプに制限する (トレース・ルール入力の影響を特定のタイプの脆弱性、メカニズム、またはテクノロジーのみに制限する) ことでフィルターを詳細化することができます。 これを行うには、「シンク・プロパティー」セクションで「**VMAT** プロパティーを追加」 ボタンをクリックし、「プロパティーの選択」ダイアログ・ボックスでプロパティーを選択します。プロパティーのリストは、「フィルター」フィールドを使用してフィルタリングすることができます。

VMAT は、AppScan Source がアプリケーション・プログラミング・インターフェース (API) に適用するプロパティーを 4 つの主なタイプにカテゴリー化したものです。 VMAT プロパティー・カテゴリーを以下に示します。

- 脆弱性: セキュリティー違反につながる悪用または攻撃ベクトルのタイプ
- メカニズム: 脆弱性を避けるために使用されるセキュリティー管理
- 属性: これらのプロパティーは、現在「プロパティーの選択」ダイアログ・ボックスで使用可能ではありません。
- テクノロジー: API が提供する機能のタイプの全般的な説明

フィルター例**:** HTTP からの SQL 注入および XSS (最も危険度の高いソース) のすべてをフィルターに掛けるには、「ソース・プロパティー」セクションに `Technology.Communications.HTTP` フィルターを含み、「シンク・プロパティー」セクションに `Vulnerability.Injection.SQL` ルールと `Vulnerability.CrossSiteScripting` ルールを含む「制限」トレース・ルールを作成してください。

– 必須呼び出し:「必須呼び出し」セクションで、ソースからシンクへのパスに存在しなければならない特定の API 呼び出しを追加します。 必須呼び出しによって、検出結果は、指定された必須の呼び出しを通過するトレースを含むものに制限されます。 「中間呼び出しの追加」をクリックすると、「API

の構成」ダイアログ・ボックスが開きます。このダイアログ・ボックスで、ソースおよびシンクを指定するのと同様に、呼び出しを指定します。

– 禁止呼び出し:「禁止呼び出し」セクションで、ソースからシンクへのパスに存在してはならない特定の API 呼び出しを追加します。 禁止呼び出しによって、検出結果は、指定された禁止呼び出しを通過しないトレースを含むものに制限されます。 必須呼び出しを追加する場合と同様にして、禁止呼び出しを追加します。

ヒント:

– 「脆弱性タイプ」、「**API**」、「ファイル」、「ディレクトリー」、または「プロジェクト」によってフィルタリングする場合は、「値の選択」ダイアログ・ボックスの上部にあるフィルター・フィールドにパターンを入力して、ダイアログ・ボックスに表示されるリストをフィルタリングできます。
– 任意の検出結果表で、「ソース」列と「シンク」列を調べて、フィルタリングによって除外したいソースおよびシンクを把握します。
– フィルタリングしたいソース、シンク、および呼び出しのプロパティーを把握するには、任意の検出結果表の「脆弱性タイプ」列を調べます。
– フィルタリングしたい可能性のある呼び出しを確認するには、任意の検出結果表の「**API**」列の項目を調べてください。

フィルターの例

表 12. フィルターの例

検出結果表でのフィルターの動作	「フィルター・エディター」ビューでのフィルター設定
検出結果表に、重大度の高い要注意セキュリティー検出結果のみが含まれる。	• 「重大度」セクションで、「高」チェック・ボックスを選択し、他のすべてのチェック・ボックスの選択を解除します。 • 「分類」セクションで、「要注意 **(Suspect)**」チェック・ボックスを選択し、他のすべてのチェック・ボックスの選択を解除します。
検出結果表に、`ProjectA` という名前のプロジェクト内のすべての検出結果が含まれる。ただし「情報」脆弱性タイプは除きます。	• 「脆弱性タイプ」セクションで、「削除」ラジオ・ボタンを選択し、「追加」をクリックします。 「値の選択」ダイアログ・ボックスで、「**Vulnerability.Info**」を選択します。 • 「プロジェクト」セクションで、「制限 **(Restrict to)**」ラジオ・ボタンを選択し、「追加」をクリックします。 「値の選択」ダイアログ・ボックスで、`ProjectA` を選択します。

表 12. フィルターの例 (続き)

検出結果表でのフィルターの動作	「フィルター・エディター」ビューでのフィルター設定
トレースを含む検出結果のみが表示される。	「トレース」セクションで、「制限 **(Restrict to)**」セクションの「追加」をクリックします。「トレース・ルール入力」ダイアログ・ボックスでデフォルトの入力を受け入れ、「**OK**」をクリックします。ダイアログ・ボックス内のデフォルト値は、以下のとおりです。 • 「ソース」の「**API** 正規表現」フィールドは **.*** で、正規表現タイプは「**PERL**」です。 これにより、フィルタリングで、Perl 正規表現構文を使用するソースを含むすべての検出結果を見つけるように AppScan Source に指示します。 • 「シンク」の「**API** 正規表現」フィールドは **.*** で、正規表現タイプは「**PERL**」です。 これにより、フィルタリングで、Perl 正規表現構文を使用するシンクを含むすべての検出結果を見つけるように AppScan Source に指示します。

表 12. フィルターの例 (続き)

検出結果表でのフィルターの動作	「フィルター・エディター」ビューでのフィルター設定
検出結果表に、java.lang.Integer.parseInt を通過しない、HTTP 関連のソースから SQL 注入関連のシンクまでが表示される。	「トレース」セクションで、「制限 **(Restrict to)**」セクションの「追加」をクリックします。「トレース・ルール入力」ダイアログ・ボックスで、以下の手順を実行します。 • 「ソース」セクションで、「**VMAT** プロパティーを追加」をクリックします。「プロパティーの選択」ダイアログ・ボックスで、Technology.Communications.HTTP を選択します。「**OK**」をクリックして VMAT プロパティーを追加し、「トレース・ルール入力」ダイアログ・ボックスに戻ります。 • 「シンク」セクションで、「**VMAT** プロパティーを追加」をクリックします。「プロパティーの選択」ダイアログ・ボックスで、Vulnerability.Injection.SQL を選択します。「**OK**」をクリックして VMAT プロパティーを追加し、「トレース・ルール入力」ダイアログ・ボックスに戻ります。 • 「禁止呼び出し」セクションで、「中間呼び出しの追加」をクリックします。「API の構成」ダイアログ・ボックスで、「**API** 正規表現」フィールドに java.lang.Integer.parseInt.* と入力します。 「**OK**」をクリックして中間呼び出しを追加し、「トレース・ルール入力」ダイアログ・ボックスに戻り、次に「**OK**」をクリックしてトレース・ルール入力を追加します。

AppScan Source 事前定義フィルターの使用

AppScan Source には、スキャン結果をフィルタリングするために選択できる事前定義フィルター一式が含まれています。このヘルプ・トピックでは、すぐに使用可能なこれらのフィルターについて説明します。

注: AppScan Source バージョン 8.8 では、より有用なスキャン結果が得られるように定義済みフィルターが改善されました。 AppScan Source の旧バージョンからの定義済みフィルターを引き続き使用する必要がある場合は (アーカイブ・フィルターのリストは 173 ページの『AppScan Source 事前定義フィルター (バージョン 8.7.x 以前)』に記載されています)、 174 ページの『アーカイブ済みの事前定義フィルターの復元』の指示のとおりに行ってください。

注: AppScan Source for Development (Visual Studio プラグイン) では、このビューは「フィルターの編集」ウィンドウの一部です。

- 『! - AppScan 厳選テスト』
- 『! - 高リスク・ソース』
- 『! - 重要なタイプ』
- 171 ページの『CWE SANS Top 25 2010 脆弱性』
- 171 ページの『外部通信』
- 171 ページの『「低」重大度と「情報」』
- 171 ページの『ノイズ - 品質 (Noise - Quality)』
- 171 ページの『OWASP Mobile Top 10 の脆弱性』
- 172 ページの『OWASP Top 10 2010 脆弱性』
- 172 ページの『OWASP Top 10 2013 脆弱性』
- 172 ページの『PCI Data Security Standard の脆弱性』
- 172 ページの『対象となる脆弱性 - HTTP ソースの EncodingRequired』
- 172 ページの『対象となる脆弱性 - C/C++ シンクの Validation Required』
- 172 ページの『トラステッド・ソース』
- 173 ページの『トレースを含まない脆弱性 (Vulnerabilities with no trace)』

! - AppScan 厳選テスト

このフィルターは、最も危険な脆弱性カテゴリーの一部からの検出結果と一致します。結果は、「高」および「中」重大度の脆弱性に限定されます。特定のソースによる結果は、検出結果から削除されます。このフィルターに含まれる具体的な脆弱性カテゴリーは、以下のとおりです。

```
Vulnerability.CrossSiteScripting
Vulnerability.CrossSiteScripting.Reflected
Vulnerability.CrossSiteScripting.Stored
Vulnerability.Injection.OS
Vulnerability.Injection.LDAP
Vulnerability.Injection.SQL
Vulnerability.Injection.Mail
```

! - 高リスク・ソース

このフィルターは、以下のいずれかのプロパティーを持つ特定の脆弱性タイプおよびソースに検出結果を制限します。

```
Technology.Communications.HTTP
Technology.Communications.IP
Technology.Communications.RCP
Technology.Communications.TCP
Technology.Communications.UDP
Technology.Communications.WebService
```

! - 重要なタイプ

このフィルターは、より広範囲の重要な脆弱性カテゴリーから取得した検出結果を含みます。検出結果は、「決定的」または「要注意」に分類される重大度「高」と「中」のものに限定されます。このフィルターに含まれる具体的なカテゴリーは、以下のとおりです。

```
Vulnerability.AppDOS
Vulnerability.Authentication.Credentials.Unprotected
Vulnerability.BufferOverflow
```

```
Vulnerability.BufferOverflow.FormatString
Vulnerability.BufferOverflow.ArrayIndexOutOfBounds
Vulnerability.BufferOverflow.BufferSizeOutOfBounds
Vulnerability.BufferOverflow.IntegerOverflow
Vulnerability.BufferOverflow.Internal
Vulnerability.CrossSiteRequestForgery
Vulnerability.CrossSiteScripting
Vulnerability.CrossSiteScripting.Reflected
Vulnerability.CrossSiteScripting.Stored
Vulnerability.FileUpload
Vulnerability.Injection
Vulnerability.Injection.LDAP
Vulnerability.Injection.OS
Vulnerability.Injection.SQL
Vulnerability.Injection.XML
Vulnerability.Injection.XPath
Vulnerability.Malicious.EasterEgg
Vulnerability.Malicious.Trigger
Vulnerability.Malicious.Trojan
Vulnerability.PathTraversal
Vulnerability.Validation.EncodingRequired
Vulnerability.Validation.EncodingRequired.Struts
```

CWE SANS Top 25 2010 脆弱性

このフィルターは、2010 年の「*CWE/SANS* 最も危険なソフトウェア・エラー *TOP 25*」に関連する脆弱性タイプを対象としています。

「*2011 CWE/SANS* 最も危険なソフトウェア・エラー *TOP 25*」について詳しくは、http://cwe.mitre.org/top25/ を参照してください。

外部通信

このフィルターは、アプリケーション外部の、ネットワークから得られる検出結果を一致させます。 このフィルターは、`Technology.Communications` ソースに起因する検出結果を一致させます。

「低」重大度と「情報」

このフィルターには、重大度が「低」で「情報」の検出結果が含まれます。すべての分類 (確定、要確認、およびスキャン範囲) が含まれます。

ノイズ - 品質 (Noise - Quality)

このフィルターを適用すると、結果に、品質のコーディング手法に関連する脆弱性タイプのみが含まれます。

OWASP Mobile Top 10 の脆弱性

このフィルターは、「Open Web Application Security Project (OWASP) Mobile Top 10 Release Candidate v1.0」リストに関連する脆弱性タイプを対象としています。

OWASP については、https://www.owasp.org/index.php/Main_Pageを参照してください。さまざまな OWASP 文書およびセキュリティー・リスクへのリンクは、https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project を参照してください。

OWASP Top 10 2010 脆弱性

このフィルターは、「Open Web Application Security Project (OWASP) Top 10 2010」リストに関連する脆弱性タイプを対象としています。

OWASP については、https://www.owasp.org/index.php/Main_Pageを参照してください。さまざまな OWASP 文書およびセキュリティー・リスクへのリンクは、https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project を参照してください。

OWASP Top 10 2013 脆弱性

このフィルターは、「Open Web Application Security Project (OWASP) Top 10 2013」リストに関連する脆弱性タイプを対象としています。

OWASP については、https://www.owasp.org/index.php/Main_Pageを参照してください。さまざまな OWASP 文書およびセキュリティー・リスクへのリンクは、https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project を参照してください。

PCI Data Security Standard の脆弱性

このフィルターは、Payment Card Industry Data Security Standard (PCI DSS) バージョン 3.2 の規格に関連する脆弱性タイプを対象としています。

詳しくは、https://www.pcisecuritystandards.org/security_standards/index.phpを参照してください。

スキャン範囲検出結果

このフィルターを適用すると、結果に、スキャン範囲検出結果のみが含まれます (詳しくは、 22 ページの『分類』を参照してください)。

対象となる脆弱性 - HTTP ソースの EncodingRequired

このフィルターは、`Validation.EncodingRequired` および `Validation.EncodingRequired.Struts` の脆弱性カテゴリーの検出結果を対象としています。`Technology.Communications.HTTP` ソースから得られる検出結果のみが含まれます。検出結果は、「高」および「中」重大度で「確定」または「要確認」分類のものに限定されます。

対象となる脆弱性 - C/C++ シンクの Validation Required

このフィルターは、既知の C および C++ シンクのセットの `Validation.Required` 脆弱性を対象としています。検出結果は、「高」および「中」重大度で「確定」または「要確認」分類のものに限定されます。

トラステッド・ソース

このフィルターは、セッション・オブジェクトや要求属性など、特定のソースからのデータは安全であるとみなします。

トレースを含まない脆弱性 (Vulnerabilities with no trace)

このフィルターは、トレースを含まない脆弱性をリストします。

AppScan Source 事前定義フィルター (バージョン 8.7.x 以前)

このトピックでは、AppScan Source バージョン 8.7.x 以前に組み込まれていた事前定義フィルターをリストします。

以下のフィルターにアクセスする必要がある場合、 174 ページの『アーカイブ済みの事前定義フィルターの復元』 の説明に従ってください。

! - 厳選テスト

このフィルターは、最も危険な脆弱性カテゴリーの一部からの検出結果と一致します。 外部ネットワーク通信ソースに起因する検出結果のみが含まれます。このフィルターは、高リスクの検出結果を得るための開始点を正確に示します。 このフィルターに含まれる具体的なカテゴリーは、以下のとおりです。

```
Vulnerability.BufferOverflow
Vulnerability.BufferOverflow.FormatString
Vulnerability.PathTraversal
Vulnerability.CrossSiteScripting
Vulnerability.CrossSiteScripting.Reflected
Vulnerability.CrossSiteScripting.Stored
Vulnerability.Injection
Vulnerability.Injection.LDAP
Vulnerability.Injection.SQL
Vulnerability.Injection.OS
Vulnerability.Injection.XML
Vulnerability.Injection.XPath
```

高優先度 - 外部通信 (High Priority - External Communications)

このフィルターは、アプリケーション外部の、ネットワークから得られる検出結果を一致させます。 このフィルターは、Technology.Communications ソースに起因する検出結果を一致させます。

高優先度 - 重要なタイプ (High Priority - Important Types)

このフィルターには、最も危険な脆弱性カテゴリー (例えば、CrossSiteScripting や Injection.SQL など) の一部からの検出結果が含まれます。このフィルターに含まれる具体的なカテゴリーは、以下のとおりです。

```
Vulnerability.AppDOS
Vulnerability.Authentication.Credentials.Unprotected
Vulnerability.Authentication.Entity
Vulnerability.BufferOverflow
Vulnerability.BufferOverflow.FormatString
Vulnerability.CrossSiteScripting
Vulnerability.CrossSiteScripting.Reflected
Vulnerability.CrossSiteScripting.Stored
Vulnerability.Injection
Vulnerability.Injection.LDAP
Vulnerability.Injection.OS
Vulnerability.Injection.SQL
Vulnerability.Injection.XML
Vulnerability.Injection.XPath
Vulnerability.PathTraversal
```

低優先度 - テスト・コード (Low Priority - Test Code)

このフィルターには、テスト・コードからの検出結果が含まれます。 このフィルターの具体的なタイプは、以下のとおりです。

```
Vulnerability.Quality.TestCode
```

ノイズ - コピーに類似した操作 (Noise - Copy-like Operations)

このフィルターには、コピーに類似した操作に関連する検出結果が含まれます。 データの取得元のソースが信頼できるかどうかにかかわらず、ソースから取得されたデータに対して実行されるアクションが信頼できる場合に、コピーに類似した操作が発生します。

以下のパターンが検索されます。

```
Technology.Database --> Vulnerability.Injection.SQL
Mechanism.SessionManagement --> Mechanism.SessionManagement
Technology.XML, Technology.XML.DOM, Technology.XML.Schema,
Technology.XML.XPath --> Vulnerability.AppDOS.XML,
Vulnerability.Injection.XML
```

ノイズ - 問題のロギング (Noise - Logging Issues)

このフィルターには、エラー処理に関連した検出結果が含まれます。 検出結果は、エラー処理ルーチンからロギング・メカニズムにまでわたります。 以下のパターンが一致します。

```
Mechanism.ErrorHandling -->
Vulnerability.Logging, Vulnerability.Logging.Forge, Vulnerability.Logging.Required
```

ノイズ - 低重大度 (Noise - Low Severity)

このフィルターには、重大度が「低」の検出結果が含まれます。 すべての分類が含まれます。

ノイズ - 信頼できるソース (Noise - Trusted Source)

このフィルターには、信頼できるソースから得られる検出結果が含まれます。 `java.lang.System.getProperty.*` をソースとする検出結果だけがこのフィルターに組み込まれます。

アーカイブ済みの事前定義フィルターの復元

このタスクの手順に従うと、バージョン 8.8 より前の AppScan Source で提供されていた事前定義フィルターを製品に再び追加することができます。 それらの事前定義フィルターは、いったん 1 台のマシンに復元すると、ユーザーが作成するフィルターと同じ方法で管理することができます (例えば、フィルターを複数のクライアントで共有できます)。

このタスクについて

アーカイブ済みの事前定義フィルターは、`<data_dir>¥archive¥filters` (`<data_dir>` は、ご使用の AppScan Source プログラム・データの場所です。詳しくは、 360 ページの『インストールとユーザー・データ・ファイルの場所』 を参照してください。) にあります。

手順

1. <data_dir>¥archive¥filters で、復元するフィルターを見つけます (AppScan Source フィルターには .off ファイル拡張子が付いています)。
2. フィルターを <data_dir>¥scanner_filters にコピーします。
3. AppScan Source を再始動します。

次のタスク

フィルター (復元したアーカイブ済みフィルターを含む) の管理方法を確認するには、 176 ページの『「フィルター・エディター」ビューでのフィルターの作成および管理』 を参照してください。

フィルターの作成および管理

AppScan Source には、フィルターを作成および使用するための方法が複数用意されています。フィルター作成用のメイン・ビューである「フィルター・エディター」ビューは、幅広い対応力のあるルールのセットを提供しており、これらを手動で設定してからフィルターに保存することができます。 「フィルター・エディター」ビューでは、作成済みのフィルターを管理するメカニズムも用意され、それらのフィルターを容易に変更または削除できるようになっています。 また、検出結果をグラフィカルに表現するビューを使用して検出結果表をフィルタリングし、その後それらのフィルターを「フィルター・エディター」ビューで保存することもできます。 フィルターを作成すると、他のビューもフィルター・プロパティーを反映するように更新されます。

- 『「フィルター・エディター」ビューでのフィルターの作成、管理、および適用』
- 176 ページの『「評価の概要」ビューおよび「脆弱性マトリックス」ビューからのフィルタリング』
- 176 ページの『「ソースとシンク」ビューでのフィルターの作成』

「フィルター・エディター」ビューでのフィルターの作成、管理、および適用

「フィルター・エディター」ビューでは、フィルター・ルールを指定することによってフィルターを作成できます。「フィルター・エディター」ビューで作成されたフィルターは、保存、変更、および削除することができます。このビューでフィルターを作成すると、そのフィルターは、ビュー内のドロップダウン・メニューを使用して適用できるようになります。 176 ページの『「フィルター・エディター」ビューでのフィルターの作成および管理』を参照してください。

AppScan Source for Analysis では、AppScan Enterprise Server に対して作成したフィルターを共有することができます。そして、他のユーザーが共有したフィルターにアクセスすることができます。 AppScan Source for Development では、サーバー・モードで実行している場合、共有フィルターにアクセスできます。

注: AppScan Source for Development (Visual Studio プラグイン) では、このビューは「フィルターの編集」ウィンドウの一部です。

「評価の概要」ビューおよび「脆弱性マトリックス」ビューからのフィルタリング

注:

- 「評価の概要」ビューは、macOS では使用できません。
- AppScan Source for Development (Visual Studio プラグイン) では、これらのビューは「フィルターの編集」ウィンドウの一部です。

「評価の概要」ビューおよび「脆弱性マトリックス」ビューでは、検出結果をグラフィカルに表現します。 これらのビューでは、検出結果は異なる方法でグループ化されます。 それらのグループを選択して検出結果表をフィルタリングし、選択された 1 つまたは複数のグループ内にある検出結果のみを表示するようにすることができます。 この方法で実行したフィルタリングはすべて、「フィルター・エディター」ビューに自動的に反映されます。その後「フィルター・エディター」ビューからフィルター設定を保存できます。

「ソースとシンク」ビューでのフィルターの作成

注: 「ソースとシンク」ビューは AppScan Source for Development (Visual Studio プラグイン) では使用できません。

「ソースとシンク」ビューでは、入力および出力のトレースに基づいて検出結果を表示およびフィルタリングすることができます。 このビューで実行されるフィルタリングは、ビュー内で直接保存できます。 フィルターの作成中に、スキャン結果にフィルターを即時に適用するオプションが用意されています。

181 ページの『「ソースとシンク」ビューでのフィルターの作成』を参照してください。

「フィルター・エディター」ビューでのフィルターの作成および管理

このビューでは、フィルターの作成、編集、保存、削除、および管理を行えます。 AppScan Source for Analysis を使用している場合、フィルターを共有して、他のユーザーによって共有されているフィルターにアクセスすることができます。 AppScan Source for Development では、サーバー・モードを使用していて AppScan Enterprise Server にログインしている場合、共有フィルターにアクセスできます。

手順

1. 356 ページの『「フィルター・エディター」ビュー』のツールバーで、「新規」をクリックします。 新規フィルターにつく名前は `Untitled<-number>` (最初のタイトルのない新規フィルターは `Untitled`、次のタイトルのない新規フィルターは `Untitled-1`、以下同様) です。

   注: AppScan Source for Development (Visual Studio プラグイン) では、このビューは「フィルターの編集」ウィンドウの一部です。
2. カテゴリーを展開し、必要なフィルター基準を選択します。
3. 「保存」または「名前を付けて保存」をクリックします。
4. フィルターに名前を付け、「**OK**」をクリックします。 フィルターのリストで、`Untitled<-number>` が新規フィルター名に置き換えられます。

次のタスク

フィルターを適用するには、「フィルター・エディター」ビューのドロップダウン・メニューでフィルターを選択します。

注: 「脆弱性マトリックス」ビューの外部で適用されるフィルターは、「脆弱性マトリックス」ビューに作用しない可能性があります。フィルターが「脆弱性マトリックス」ビューに反映されるようにするには、「脆弱性マトリックス」ビューの「フィルターに掛けた検出結果の数を表示」ツールバー・ボタンを選択する必要があります。

フィルターは、リスト内でフィルターを選択してから操作することによって、「フィルター・エディター」ビュー内で直接管理できます。あるいは、「フィルターの管理」をクリックして「フィルターの管理」ダイアログ・ボックスを開くことができ、このダイアログ・ボックスに、保存されたフィルターのリストが表示されます。

- フィルターの変更:「フィルター・エディター」ビューまたは「フィルターの管理」ダイアログ・ボックスでフィルターを選択し、そのフィルター・ルールを変更して、変更内容を保存します。

  注: 標準装備フィルターは、変更も削除もできません。
- フィルターの削除:「フィルター・エディター」ビューまたは「フィルターの管理」ダイアログ・ボックスでフィルターを選択し、「削除」をクリックします。「フィルターの管理」ダイアログ・ボックスで、複数のフィルターを選択し、「削除」をクリックしてそれらを一度に削除することができます。
- 別のフィルターからのフィルターの作成: フィルターを変更し、「名前を付けて保存」をクリックして、新しい名前のフィルターとして保存することができます。 こうすることで、既存のフィルターの設定をベースとして新規フィルターを作成できます。 この操作は、「フィルター・エディター」ビューおよび「フィルターの管理」ダイアログ・ボックスの両方で実行できます。

  ヒント: フィルターを開き、「名前を付けて保存」アクションを使用して新規名で保存することでも、同じ内容を実行できます。これで、新規フィルターを開いて変更できます。この方法を選択すると、標準装備フィルターの 1 つから新規フィルターを作成できます。
- フィルター設定の復帰: フィルターのプロパティーを変更したが、それらの変更を元に戻したい場合は、「復帰」をクリックして、フィルターを最後に保存された設定に戻します。 このアクションは、「フィルター・エディター」ビューおよび「フィルターの管理」ダイアログ・ボックスの両方で実行できます。ダイアログ・ボックスで、変更が保存されていないフィルターが複数ある場合、「復帰」をクリックすると、変更が保存されていないすべての選択されたフィルターが、保存された設定に戻されます。
- フィルターの共有 (AppScan Source for Analysis のみ): 共有フィルターを作成するには、フィルター・エディターでフィルターを開き、「フィルター・エディター」ビューのツールバーで「フィルターの共有」をクリックします。

  注: 共有フィルターを変更、削除、または作成するには、「共有フィルターの管理」権限が必要です。権限の設定について詳しくは、「*IBM Security AppScan Source* インストールと管理のガイド」を参照してください。

「評価の概要」ビューからのフィルタリング

スキャンが完了すると、「評価の概要」ビューで検出結果を見ることができます (このビューはデフォルトで「トリアージ」パースペクティブで開きます)。 このビューでは、棒グラフからフィルターを作成できます。

このタスクについて

スキャンが完了すると、 355 ページの『「評価の概要」ビュー』には、検出結果が棒グラフのグラフィカル表現で表示されます。 このビューは、脆弱性タイプ、API、プロジェクト、またはファイルごとに検出結果を表示するように詳細化することができます。 「評価の概要」ビューで、グループ化された検出結果を選択すると、検出結果表は、「評価の概要」ビューで選択された検出結果のみを表示するように変わります。

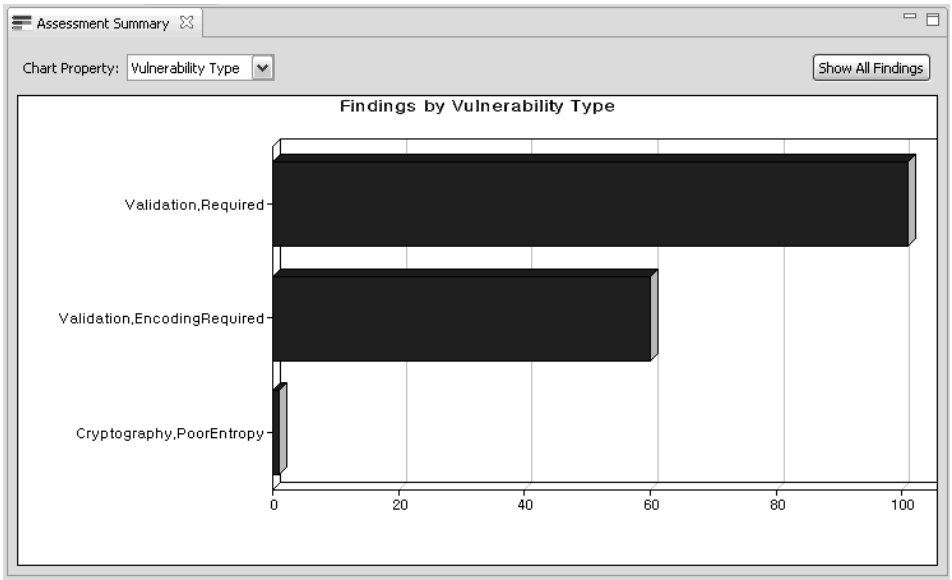
注: 「脆弱性マトリックス」ビューの外部で適用されるフィルターは、「脆弱性マトリックス」ビューに作用しない可能性があります。フィルターが「脆弱性マトリックス」ビューに反映されるようにするには、「脆弱性マトリックス」ビューの「フィルターに掛けた検出結果の数を表示」ツールバー・ボタンを選択する必要があります。

注:

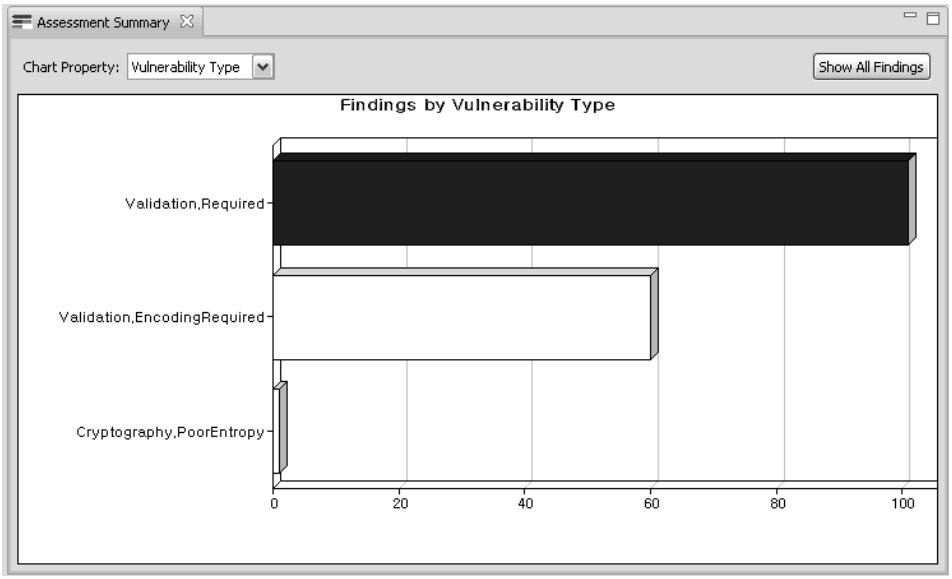
- 「評価の概要」ビューは、macOS では使用できません。
- AppScan Source for Development (Visual Studio プラグイン) では、このビューは「フィルターの編集」ウィンドウの一部です。

手順

1. 「評価の概要」ビューで、目的に合うようにグラフィカル表現を変更します。 例えば、`Validation.Required`、`Validation.EncodingRequired`、および `Cryptography.PoorEntropy` という脆弱性タイプを含む評価の場合は、「グラフのプロパティー」を「脆弱性タイプ」に設定します。 これにより、以下のように棒グラフ表現で脆弱性タイプごとに検出結果が表示されます。

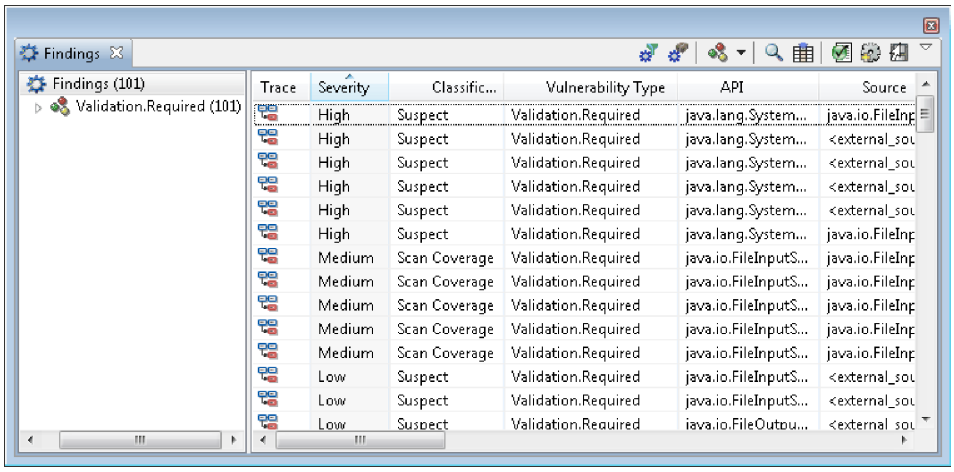


2. `Validation.Required` 脆弱性タイプのフィルターを作成するには、グラフ内の `Validation.Required` を示すグラフの棒をクリックします。

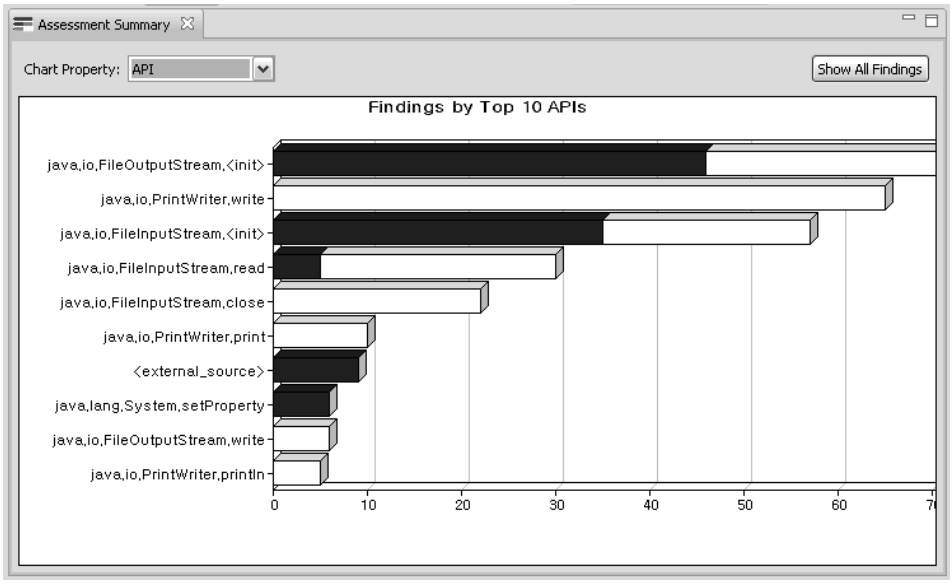


ヒント: 脆弱性の数を表示するには、マウスをバーの上に合わせた状態にします。

フィルタリングされた結果が検出結果表に表示されます。



3. また、このフィルタリング・アクションを行うと、「評価の概要」ビューでの選択内容のフィルター・ルール設定が「フィルター・エディター」ビューに取り込まれます。 このフィルターは、「フィルター・エディター」ビューで保存できます (フィルター・ルール設定と、フィルターの保存については、 176 ページの『「フィルター・エディター」ビューでのフィルターの作成および管理』を参照してください)。
4. API によって同じフィルター結果を表示するには、「グラフのプロパティー」を「**API**」に設定します。



「脆弱性マトリックス」ビューからのフィルタリング

「脆弱性マトリックス」ビューには、スキャンに含まれるすべてのアプリケーションの検出結果の総数が表示されます。これらの検出結果は、マトリックスで重大度レベルごとにグループ化されています。これらの検出結果グループを選択することによって、フィルターを作成できます。

このタスクについて

357 ページの『「脆弱性マトリックス」ビュー』で、グループ化された検出結果を選択すると、検出結果表は、「脆弱性マトリックス」ビューで選択された検出結果のみを表示するように変わります。

注: AppScan Source for Development (Visual Studio プラグイン) では、このビューは「フィルターの編集」ウィンドウの一部です。

注: 「品質」検出結果、および「情報」重大度レベルに分類される検出結果は、「脆弱性マトリックス」ビューに含まれません。

手順

1. 「脆弱性マトリックス」ビューで、検出結果表に表示したいマトリックスのセクションを選択します。 例えば、検出結果表に、「高」重大度の「要確認」セキュリティー検出結果のみを表示するには、マトリックスのそのセクションを選択します。これにより、フィルタリングされた結果が検出結果表に表示されます。
2. また、フィルタリング・アクションを行うと、「脆弱性マトリックス」ビューでの選択内容のフィルター・ルール設定が「フィルター・エディター」ビューに取り込まれます。 このフィルターは、「フィルター・エディター」ビューで保存できます (フィルター・ルール設定と、フィルターの保存については、176 ページの『「フィルター・エディター」ビューでのフィルターの作成および管理』を参照してください)。

「ソースとシンク」ビューでのフィルターの作成

手順

1. 「ソースとシンク」ビューを開くか、このビューにナビゲートします。
2. 「ソースとシンク」ビューには 3 つのセクションが含まれています。 ビューの検出結果表セクションに、他の 2 つのセクションで表示対象として選択したソース、シンク、および中間ノードの検出結果が表示されます。 これについては、 349 ページの『「ソースとシンク」ビュー』で説明しています。
3. 重要と考える検出結果を表示するように検出結果表を設定した後に、「選択したソース、シンク、および中間ノードに基づいて新規フィルターを作成」をクリックします。
4. 「フィルターの作成」ダイアログ・ボックスで、以下のようにします。
   - 「名前」フィールドで、フィルターの名前を指定します。
   - 「このフィルターを直ちに適用する」を選択して、評価作業においてこのフィルターがすべての検出結果表に適用されるようにします。このチェック・ボックスを選択する操作は、「フィルター・エディター」ビューでフィルターを選択する操作と同じ意味があります。 既定のメイン・フィルターが設定され、すべてのビュー (例えば、「脆弱性マトリックス」および「検出結果」ビュー) に効果が及びます。

     注**:** 「脆弱性マトリックス」ビューの外部で適用されるフィルターは、「脆弱性マトリックス」ビューに作用しない可能性があります。フィルターが「脆弱性マトリックス」ビューに反映されるようにするには、「脆弱性マトリックス」ビューの「フィルターに掛けた検出結果の数を表示」ツールバー・ボタンを選択する必要があります。
   - フィルタリングされた検出結果が現在の作業に関係しない場合は、「これらの検出結果を除外するアプリケーション・フィルターの作成」チェック・ボックスを選択することによって、評価からそれらの検出結果を削除することができます。このチェック・ボックスを選択すると、アプリケーション・プロパティーで、新規フィルターが除外フィルターとして追加されます (アプリケーションの「プロパティー」ビューで「除外」タブを選択すると、除外されたフィルターのリストが表示されます)。 今後アプリケーションをスキャンした場合、フィルターに一致する検出結果は「検出結果」ビューではなく「除外された検出結果」ビューで報告されます。
5. 「**OK**」をクリックして、検出結果をフィルタリングまたは除外します。

フィルターの適用

フィルターは、スキャン前またはスキャン後に適用できます。スキャン後にフィルターを適用するには、「フィルター・エディター」を使用するか、またはフィルターを適用できる別のビューを使用します。スキャン前にフィルターを適用するには、グローバル・フィルターを設定するか、スキャン構成を使用してフィルタリングを行います。スキャン前にフィルターを適用すると、フィルタリング対象でない検出結果を表示することも、再スキャンせずにフィルターを削除することもできません。

スキャン前のフィルターの適用

グローバル・フィルターの設定方法については『グローバル・フィルターの適用』を、スキャン構成でのフィルターの設定方法については 123 ページの『スキャン構成の管理』を参照してください。

スキャン後のフィルターの適用

「フィルター・エディター」ビューでフィルターを選択すると、そのフィルターが自動的に検出結果のリストに適用されます。フィルタリング操作が可能なその他のビューについては、 175 ページの『フィルターの作成および管理』に説明があります。

グローバル・フィルターの適用

既に作成済みのフィルターは、すべてのアプリケーション、個別のアプリケーション、および個別のプロジェクトに適用できます。グローバル・フィルターの適用は「プロパティー」ビューで行います。このビューでは、フィルターの適用方法を指定できます (フィルターは直接適用することも、反転して適用することもできます)。例えば、アプリケーションにグローバル・フィルターを設定する場合は、「エクスプローラー」ビューでアプリケーションを選択して、「プロパティー」ビューを開きます (「表示」メニューを使用するか、アプリケーションを右クリックして「プロパティー」をクリック)。

始める前に

すべてのアプリケーションまたは個別のプロジェクトに対してフィルターを設定する場合は、「プロパティー」ビューの「フィルター」タブを使用します。個別のアプリケーションに対してフィルターを設定する場合は、「プロパティー」ビューの「除外およびフィルター (Exclusions and Filters)」タブを使用します。

手順

1. タブの「フィルター」セクションで、「追加」をクリックします。
2. 「フィルターの選択」ダイアログ・ボックスで、グローバルに適用するフィルターを選択します。
3. オプション: フィルターを反転して適用する場合は (フィルターを直接適用する代わりに)、「フィルターの反転 **(Invert filter)**」を選択します。
4. 「**OK**」をクリックして「フィルターの選択」ダイアログ・ボックスを閉じます。
5. フィルターの追加を完了したら、「プロパティー」ビューで変更内容を保存します。

適用済みフィルターの判別

スキャンの前にアプリケーションとプロジェクトにフィルターをグローバルに適用することも、スキャンの後で評価にフィルターを適用することもできます。評価の検出結果にフィルターがどのように適用されたか素早く判別できるように、AppScan Source のメインワークベンチの下部にフィルター標識が表示されます。

フィルターが適用されていない場合は、ワークベンチの下部のフィルター標識に「検出結果はフィルタリングされていません **(Findings are not filtered)**」と表示されます。

フィルターが適用されている場合、標識は「検出結果はフィルタリングされています **(Findings are filtered)**」というリンクに変更されます。このリンクを選択すると次のようなメッセージが表示され、フィルターがどのように適用されたか確認できます。

- 「スキャン時フィルター **(Scan-time filters)**」は、アプリケーションとプロジェクトに次のように適用されるグローバル・フィルターです。
  - その評価が、フィルターが構成されていないアプリケーションまたはプロジェクトのスキャン結果である場合は、スキャン時フィルターが適用されていないことがメッセージに示されます。
  - スキャンされたアプリケーションまたはプロジェクトにフィルターが構成されている場合は、構成されたフィルターの名前がリストされます。
  - 場合によっては、スキャン時フィルターが適用されたことを AppScan Source が検出しても、これらのフィルターに関する情報が評価に含まれていないことがあります。例えば、古い評価を開くとこのようになる場合があります。
- 「現行フィルター **(Current filters)**」は、スキャン後に検出結果に適用されたフィルターです。メッセージは、現行フィルターが適用されていないか、またはフィルターが適用されているかを示しています。フィルターが適用されている場合は、リセット・リンクを選択できます。このリンクを選択すると、現行フィルターが検出結果から削除されます。

除外を使用したトリアージ

スキャン後に、一部の検出結果が現在の作業には関係しないと判断した場合、スキャン結果のトリアージを行うときに、検出結果表にその検出結果を表示しないようにします。 これらの除外 (または除外された検出結果) は、「検出結果」ビューに表示されなくなり、評価メトリックは、変更された結果によって直ちに更新されます。 構成に追加されたフィルターおよびバンドルの除外は、後続のスキャンが行われるときに初めて有効になります。

除外の有効範囲

除外は、すべてのアプリケーション (グローバル)、個別のアプリケーション、またはプロジェクトに適用されます。

- グローバル除外は、すべてのスキャンに適用されます。
- アプリケーション除外は、特定のアプリケーションと、それに対応するプロジェクトに対して実行されるスキャンにのみ適用されます。
- プロジェクト除外は、特定のプロジェクト内に存在する検出結果に適用されます。

注**:** 除外は、検出結果の合計などの評価メトリックに影響します (除外された検出結果は評価メトリックには含まれません)。

グローバル除外

グローバル除外は、任意の AppScan Source for Analysis アプリケーションから保管またはアクセスできます。また、これらの除外はすべてのスキャンに適用されます。共有フィルターのみがグローバル・フィルターとなることができます。

アプリケーション除外およびプロジェクト除外

バンドル除外は、アプリケーションに対してのみ適用されます。 フィルター除外は、アプリケーションまたはプロジェクトに適用できます。 アプリケーションおよびプロジェクトに適用された除外は、共有してもローカルにしても構いません。

除外の指定

検出結果表または「プロパティー」ビューから、検出結果に除外のマークを付けることができます。 除外は、個別の検出結果、フィルター、またはバンドルで構成されます。 通常、検出結果表から作成された除外は、直ちに有効になります。 「プロパティー」ビューで作成された除外は、追加でスキャンを実行しなければ有効になりません。

以下の手順では、除外はアプリケーションに対して直ちに適用されます。

- 1 つ以上の検出結果を選択し、選択した項目を右クリックして、次にメニューから「検出結果の除外」を選択する。
- 現在除外されているバンドル (「除外されたバンドル」を含む) に 1 つ以上の検出結果を追加する。
- 以前に除外されたバンドル (「除外されたバンドル」を含む) から 1 つ以上の検出結果を削除する。
- 除外されたバンドルを削除する。

以下の場合には、除外はアプリケーションに直ちに適用されません。

- バンドルを除外として追加したとき。
- フィルターを除外として追加したとき。
- 検出結果を、除外されたフィルターの基準に一致するように変更したとき。
- 検出結果を、除外されたフィルターの基準に一致しなくなるように変更したとき。

検出結果表で検出結果に除外としてマークを付ける

手順

1. 検出結果表で、自分にとって重要でない可能性があるか、表示不要な検出結果 (または検出結果のグループ) を選択します。
2. 選択項目を右クリックして、メニューから「検出結果の除外」を選択します。除外は直ちに適用されます。除外された検出結果は表に表示されなくなり、メトリックは直ちに更新されます。

タスクの結果

除外された検出結果を表示するには、「除外された検出結果」ビューを開きます。除外された検出結果は、「除外されたバンドル」という名前のバンドルにも表示されます。

除外された検出結果を再度組み込む場合は、『除外としてマークされた検出結果の再組み込み』の説明に従ってください。

除外としてマークされた検出結果の再組み込み

除外された検出結果は、「除外された検出結果」ビューに表示されます。このビューから、除外された検出結果を再度組み込むことができます。

手順

1. 「除外された検出結果」ビューで、再度組み込む検出結果 (または検出結果グループ) を選択します。
2. 選択項目を右クリックして、メニューから「検出結果の組み込み」を選択します。

タスクの結果

組み込まれた検出結果が評価に追加されます。検出結果表およびメトリックが直ちに更新されて、再度組み込まれた検出結果が反映されます。これらの検出結果は、「除外された検出結果」ビューには表示されなくなります。

注: AppScan Source for Analysis では、「除外されたバンドル」ビューでバンドルから検出結果を削除するか、除外対象ではない新規バンドルに検出結果を移動することで、除外された検出結果を再度組み込むこともできます。

例: フィルター除外の指定

フィルター基準は、フィルターが、フィルターに一致する検出結果を除外するか、一致しない検出結果を除外するかを決定します。

以下の例では、検出結果を除外するフィルターの作成方法を説明しています。

- 『例: ディレクトリーのフィルタリングおよび除外』
- 186 ページの『例: API のフィルタリングおよび除外』

例: ディレクトリーのフィルタリングおよび除外

この例では、Microsoft `include` ファイルを含む検出結果のみを示すフィルターが作成されます。このフィルターは、後で、検出結果を絞り込むために使用します (フィルターに一致するすべての検出結果を除外します)。

手順

1. 「フィルター・エディター」ビューの「ディレクトリー」セクションで、Microsoft インクルード・ファイル (例えば、`C:¥Program Files¥Microsoft Visual Studio 8¥VC¥include`) へのパスを追加します。
2. 「制限 **(Restrict to)**」を選択して、これを包含ルールにします。

3. 「検出結果」ビューのツールバーで、「フィルターに一致しない検出結果の表示」をクリックして、Microsoft ヘッダー・ファイルの検出結果のみを表示します。この機能により、フィルターの反転をグローバルに適用して再度スキャンした後に、スキャン結果がどのようになるか確認できます。
4. 「**MS Includes**」などの名前を付けてフィルターを保存します。
5. 「構成」パースペクティブに戻り、「エクスプローラー」ビューで、C/C++ アプリケーションまたはプロジェクトを選択します。
6. アプリケーションを選択した場合は、「プロパティー」ビューの「除外およびフィルター (Exclusions and Filters)」タブを開きます。プロジェクトを選択した場合は、「プロパティー」ビューの「フィルター」タブを開きます。「追加」をクリックします。「**MS Includes**」を選択し、「フィルターの反転 **(Invert filter)**」を選択します。
7. 「プロパティー」ビューで変更内容を保存し、アプリケーションまたはプロジェクトを再度スキャンします。
8. トリアージに戻ります。 除外に含まれる検出結果は、「除外された検出結果」ビューに表示されます。

例: API のフィルタリングおよび除外

一般的なトリアージのシナリオとして、検出結果の優先順位付けを行うときに、除外したい特定の検出結果が現れるということが、トリアージ・プロセスの早い段階で発生する可能性があります。 例えば、3 つの API が脅威ではないと判断し、それらの API を後続のスキャンから除外したい場合があります。

手順

1. フィルター・エディターの「API」セクションで、「追加」をクリックして 3 つの API を選択します。
2. 「制限 **(Restrict to)**」を選択します。
3. フィルターを保存して名前を付けます。
4. 「構成」パースペクティブに戻り、「エクスプローラー」ビューで、プロジェクト (またはアプリケーション) を選択します。
5. 「プロパティー」ビューで、フィルターの動作を「反転 **(Inverted)**」に設定します (「フィルターの選択」ダイアログ・ボックスで、「フィルターの反転 **(Invert filter)**」を選択します)。
6. 再度スキャンを行います。 フィルターに含まれる API は、検出結果に表示されなくなります。

タスクの結果

同様の例として、フィルターに含まれる検出結果のみを表示させたい場合もあります。 この例では、フィルターをリストに追加する際に、「フィルターの反転 **(Invert filter)**」は選択しないでください。再度スキャンを実行すると、このフィルターに含まれる検出結果のみが表示されます。

「プロパティー」ビューからのバンドル除外の指定

バンドル除外は、バンドル内の検出結果を除外します。除外できるのはアプリケーションからのバンドルのみです。

手順

1. 『バンドルの作成』での説明に従い、バンドルを作成します。
2. 「エクスプローラー」ビューで、バンドルに関連付けるアプリケーションを選択します。
3. 「プロパティー」ビューで、「除外」タブを選択します。
4. 「バンドルの追加」をクリックし、「バンドルの選択」ダイアログ・ボックスで、アプリケーションから除外する検出結果を含むバンドルを選択します。
5. 「**OK**」をクリックします。
6. 再度スキャンを行います。 バンドルされている検出結果は、検出結果表に表示されなくなります。

バンドルを使用したトリアージ

バンドルは、固有の特性を持ち、トリアージ・プロセスのために重要となる場合があります。

このタスクについて

- バンドルは、単一の障害として、またはバンドル内の各検出結果の障害として、障害追跡システムにエクスポートできます。
- バンドルをレポート生成の基礎とすることができます。
- バンドルはアプリケーションに付加されます。

重要**:** 検出結果は、一度に 1 つのバンドルにのみ存在できます。あるバンドル内の検出結果を別のバンドルに移動すると、その検出結果は最初のバンドルから削除されます。

以下の例で、バンドルを使用した簡単なトリアージの概略を示します。

手順

1. ソース・コードをスキャンします。
2. `Resolve ASAP` という名前のバンドルを作成します。
3. いくつかの重大な検出結果をバンドルに追加します。
4. バンドル内の検出結果にメモを追加します。
5. バンドルまたは検出結果を障害追跡システムに送信します。または、他の開発者に E メールで送信します。
6. 問題を修正します。

バンドルの作成

バンドルの作成は、「バンドル」ビューまたは検出結果表を含むビューで行われます。 既存のバンドルにも、新規バンドルにも、検出結果を追加できます。

以下のトピックでは、「バンドル」ビューおよび「検出結果」ビューでのバンドルの作成について説明しています。

- 188 ページの『「バンドル」ビューでの新規バンドルの作成』
- 188 ページの『「検出結果」ビューでの新規バンドルの作成』

注: 評価のバンドルを作成するには、評価を作成するためにスキャンされたアプリケーションが AppScan Source for Analysis にロードされている必要があります。ロードされていないアプリケーションの評価を開くと、バンドルの作成アクションは使用できません。

1 つ以上のバンドルを作成した後、「検索結果」ビューの「バンドルされている検出結果の非表示」アクション ( ) で、バンドルされた検出結果のビューへの表示を切り替えることができます。このアクションは、自分で作成した、含まれているすべてのバンドル内の検出結果を非表示にします。この設定は、除外されたバンドルの検出結果の表示には影響しません。これらの検出結果は「検出結果」ビューに表示されることはありません。

「バンドル」ビューでの新規バンドルの作成

手順

1. 「バンドル」ビューで、ツールバーの「新規バンドル」をクリックします。
2. バンドルに名前を付け、「**OK**」をクリックします。 バンドル名が「バンドル」ビューに表示されます。
3. バンドルに検出結果を追加するには、『既存のバンドルへの検出結果の追加』での説明に従ってください。

「検出結果」ビューでの新規バンドルの作成

手順

1. 「検出結果」ビューで、バンドルに追加する検出結果を選択します。
2. 選択したものを右クリックし、メニューで「バンドルに追加」 > 「新規」を選択します。
3. バンドルに名前を付け、「**OK**」をクリックします。

既存のバンドルへの検出結果の追加

このタスクについて

以下のように、複数のビューからバンドルに検出結果を追加できます。

- 「検出結果」ビュー
- 「除外された検出結果」ビュー
- 「修正された/変更された検出結果」ビュー
- 「存在しない検出結果」ビュー
- 「レポート」ビュー
- 検出結果の詳細

ヒント: ドラッグ・アンド・ドロップ操作を使用して、検出結果表から「バンドル」ビューへ検出結果を移動することができます。

バンドルに検出結果を追加するには、以下のようにします。

手順

1. バンドルに追加する検出結果を選択します。

2. 選択した項目を右クリックし、メニューから「バンドルに追加」 > 「<バンドル名>」 (このリストには最も新しく作成された 5 つのバンドルが含まれます) または「バンドルに追加」 > 「選択」を選択します。
3. 「バンドルに追加」 > 「選択」を選択した場合、「バンドルの選択」ダイアログ・ボックスで、検出結果を追加するバンドルを選択し、「**OK**」をクリックします。

バンドル間での検出結果の移動

手順

1. 「バンドル」ビューで、移動する 1 つ以上の検出結果が含まれるバンドルを開きます。
2. 移動する 1 つ以上の検出結果を選択して、以下のいずれかのアクションを実行します。
   - ビューのツールバーの「バンドルに移動」または「新規バンドルに移動」をクリックします。検出結果の移動先にするバンドルを選択するか、検出結果に対して新しいバンドルを作成します。
   - 選択項目を右クリックして、「バンドルに移動」をクリックします。これによってメニューが開き、リストまたはダイアログ・ボックスから既存のバンドルを選択するか、選択した検出結果の移動先にする新規バンドルを作成することができます。

タスクの結果

注**:** 除外されたバンドルに移動または追加された検出結果は、現在の評価では除外されません。検出結果を現在の評価で除外済みとしてマークするには、「検出結果の除外」アクションを使用してください。

バンドル内の検出結果の表示

バンドルに検出結果を追加すると、それらの検出結果はバンドル内の行として表示されます。 バンドルを開いた場合、バンドルに含まれるすべての検出結果が表示されます。

このタスクについて

バンドル内の複数のプロジェクトからの検出結果は、異なった表示になることがあります。 バンドル内の検出結果は、最後に実行したスキャンで検出されなかった場合、緑色のイタリック体で表示されます。

以下に、ある「アプリケーション X」の例について考えてみましょう。

手順

1. アプリケーション X には、プロジェクト A とプロジェクト B が含まれています。
2. アプリケーション X をスキャンします。
3. プロジェクト A およびプロジェクト B からの検出結果を含むバンドルを作成します。

4. プロジェクト B をスキャンします。「バンドル」ビューに、プロジェクト B からの検出結果が表示され、プロジェクト A からの検出結果は緑色のイタリック体で表示されます。

タスクの結果

緑色のイタリック体で強調表示された検出結果は、修正された/存在しない検出結果です。修正された/存在しない検出結果は、バンドル内にあるが、現在の評価には含まれない検出結果です。 検出結果が解決済み/存在しないとして識別されるのは、それが解決されたか、削除されたか、またはソース・ファイルがスキャンされなかったためです。「バンドル」ビューで、「除外済み」列にバンドルが除外されているかどうかが示されます。

バンドルのファイルへの保存

バンドルをファイルとして保存し、AppScan Source for Development で開くことができます。バンドルを使用すると、検出結果のスナップショットを AppScan Source for Analysis から AppScan Source for Remediation にインポートすることもできます。

手順

1. 以下のアクションのいずれかを実行します。
   a. 「バンドル」ビューで、バンドルを選択し、ツールバーの「バンドルをファイルに保存」をクリックします。
   b. バンドルを開き、ツールバーの「バンドルをファイルに保存」をクリックします。
2. バンドル・ファイルを保存するディレクトリーを選択します。
3. バンドル・ファイルに名前を付けます (`<file_name>.ozbdl`)。

タスクの結果

保存されたバンドルを開くには、以下のようにします。

- AppScan Source for Development(Eclipse プラグイン) で、「セキュリティー分析」 > 「開く」 > 「バンドルを開く」を選択します。
- AppScan Source for Development (Microsoft Visual Studio プラグイン) で、「**IBM Security AppScan Source**」 > 「バンドルを開く」を選択します。
- AppScan Source for Analysis で、「バンドル」ビュー・ツールバーの「バンドルを開く」をクリックします。

ヒント**:** Windows システムの場合は、「バンドル」ビューでバンドル・ファイルをダブルクリックすると、AppScan Source for Analysis または AppScan Source for Development で開きます。

バンドルの障害追跡への送信および E メールによる送信

バンドルでの検出結果は、自社で使用している障害追跡システムに送信できます。あるいは、E メールで送信することもできます。バンドルに検出結果を入れると、開発者による修復のため、これらの検出結果をバグとして送信することができます。

手順

1. バンドルを開きます。
2. 「バンドルを障害追跡に送信」ツールバー・ボタンの下矢印をクリックして、障害追跡システムを選択します。

   注**:** 障害追跡システムによっては、「障害追跡システム」設定を変更してからバンドルを送信することが必要な場合があります。

   あるいは、「バンドル」ツールバーの「バンドルを **E** メールで送信」をクリックして、バンドルを他のユーザーに送信します (E メールの設定は事前に構成しておく必要があります)。
3. 開いている構成ダイアログ・ボックスの設定を完了します。これらのダイアログ・ボックスは、選択した障害追跡システムにより異なります。詳しくは、ヘルプの『*AppScan Source for Analysis* および障害追跡』セクションで説明しています。

バンドルへの注釈の追加

手順

1. 「バンドル」ビューで、注を付けるバンドルを選択します。
2. 「バンドル」ツールバーの「注釈の追加」をクリックするか、選択した注釈を右クリックしてメニューの「注釈の追加」を選択します。
3. 注を入力し、「**OK**」をクリックします。

検出結果の変更

変更された検出結果とは、脆弱性タイプ、分類、または重大度が変更された検出結果、あるいは注釈が付けられた検出結果です。「変更された検出結果」ビューには、現在のアプリケーション (アプリケーションの評価を開いた結果、アクティブになっているアプリケーション) について、これらの検出結果が表示されます。「自分の評価」ビュー (AppScan Source for Analysis でのみ使用可能) では、「変更済み」列に、検出結果が現在の評価で変更されたかどうかが示されます。

検出結果への変更は即時に適用され、メトリックが更新されます。変更は、アプリケーションと共に保管され、それ以降のスキャンに適用されます。

検出結果は、「検出結果の詳細」ビューで、または検出結果表を含む任意のビューから変更できます。 「検出結果の詳細」ビューでは個別の検出結果の変更が可能です。または、検出結果表内の複数の検出結果を変更できます。

注**:** 評価を変更した後に変更を保存するには、「評価の保存」権限が必要です。

検出結果表からの変更の実行

同じ変更を複数のファイルに対して加える場合、検出結果表から検出結果を変更すると便利です。個々の検出結果を変更する場合、検出結果表または「検出結果の詳細」ビューを使用します。

- 192 ページの『脆弱性タイプの変更』
- 192 ページの『検出結果分類の昇格』

- 『重大度の変更』
- 203 ページの『サポートされる注釈と属性』

脆弱性タイプの変更

個別の検出結果について、または検出結果のグループについて、脆弱性タイプを変更できます。

手順

1. 検出結果表から、変更対象の 1 つの検出結果または検出結果のグループを選択します。
2. 選択したものを右クリックし、メニューから「脆弱性タイプの設定」を選択します。
3. 「脆弱性タイプの選択」ダイアログ・ボックスで、必要な脆弱性タイプを選択し、「**OK**」をクリックします。

検出結果分類の昇格

「要確認」セキュリティー検出結果またはスキャン範囲検出結果に分類された検出結果を、「確定」検出結果に昇格することができます。

手順

1. 検出結果表から、変更対象の 1 つの検出結果または検出結果のグループを選択します。
2. 選択したものを右クリックし、メニューから「確定に昇格」を選択します。

重大度の変更

新しい重大度レベルを選択すると、選択した各検出結果の重大度が変わります。 例えば、AppScan Source が、ある API の重大度が中であると報告していても、会社のポリシーでは、より重大度が高いと見なされる場合があります。 要件に合わせて、重大度の変更が可能です。ただし、AppScan Source「修復支援」では、この変更は含まれていないことに注意してください。

手順

1. 検出結果表から、変更対象の 1 つの検出結果または検出結果のグループを選択します。
2. 選択したものを右クリックし、メニューから「重大度の設定」を選択します。
3. 新しい重大度レベルとして、「高」、「中」、「低」、または「情報」を選択します。

検出結果への注釈付け

注釈は、検出結果に対してさらにアクションを実行したり、他のユーザーに検出結果に関する情報を伝達したりするための注意を喚起するために使用できます。注釈は、単一の検出結果または検出結果のグループに追加できます。

手順

1. 検出結果表から、変更対象の 1 つの検出結果または検出結果のグループを選択します。
2. 選択項目を右クリックして、メニューから「注釈の追加」を選択します。

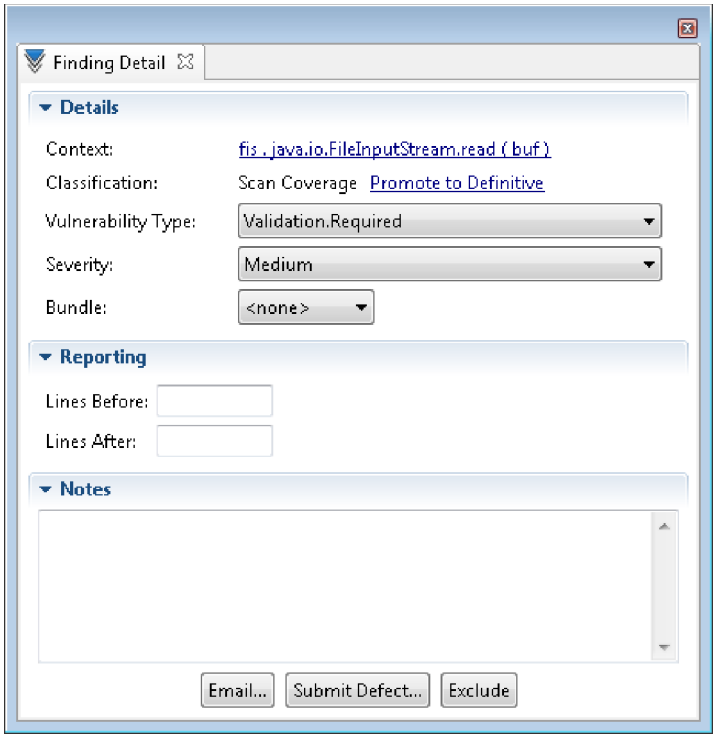
3. 注を入力し、「**OK**」をクリックします。

「検出結果の詳細」ビューでの検出結果の変更

個別の検出結果は、「検出結果の詳細」ビューで変更できます。表で検出結果を選択し、「検出結果の詳細」ビューを開いた場合、選択された検出結果と、その特性が表示されます。

「検出結果の詳細」ビュー

検出結果を選択すると、「検出結果の詳細」ビューが表示され、そのプロパティーを変更できます。このビューでは、個々の検出結果を変更できます。



- 『詳細セクション』
- 194 ページの『レポート作成セクション (AppScan Source for Analysis および AppScan Source for Development (Eclipse プラグイン) でのみ使用可能)』
- 194 ページの『注釈セクション』
- 194 ページの『「検出結果の詳細」ビューのアクション』
- 195 ページの『カスタム検出結果の「検出結果の詳細」ビュー (AppScan Source for Analysis でのみ使用可能)』

詳細セクション

- コンテキスト: 脆弱性のある部分を囲むコードのスニペット
- 分類: 「確定」または「要確認」のセキュリティー検出結果あるいは「スキャン範囲」検出結果 (分類が変更された場合に検出結果を 「確定」に昇格するか、元の値に戻すためのリンク付き)
- 脆弱性タイプ

- 重大度: 高、中、低、または情報
- バンドル: 検出結果を含むバンドルの名前 (AppScan Source for Development (Visual Studio プラグイン) では使用不可)

レポート作成セクション (AppScan Source for Analysis および AppScan Source for Development (Eclipse プラグイン) でのみ使用可能)

レポート内の検出結果の前または後 (またはその両方) に組み込むコードの行数を指定します。

注釈セクション

検出結果に注釈を付けます。

「検出結果の詳細」ビューのアクション

- 除外: 検出結果を検出結果表から除外 (削除) するには、「除外」をクリックします。 除外された検出結果を表示するには、「除外された検出結果」ビューを開きます。
- AppScan Source for Analysis でのみ使用可能:
  - **E** メール: E メール設定を構成した場合、検出結果バンドルを直接開発者に E メールで送信し、スキャン後に検出された潜在的な障害について通知することができます。 この E メールには、検出結果を含むバンドル添付ファイルが含まれ、E メール・テキストでは検出結果を説明しています。
    1. 「検出結果の詳細」ビューで現在の検出結果を E メールで送るには、「**E** メール」をクリックします。
    2. 「添付ファイル名」ダイアログ・ボックスで、E メールに添付される検出結果バンドルの名前を指定します。 例えば、「添付ファイル名」フィールドで `my_finding` と指定すると、ファイル名が `my_finding.ozbdl` のバンドルが E メールに添付されます。
    3. 「**OK**」をクリックすると、「検出結果の E メール送信」ダイアログ・ボックスが開きます。 デフォルトで、「検出結果の E メール送信」ダイアログ・ボックスの「宛先」フィールドには、E メール設定で指定されている「宛先アドレス」が入力されていますが、これは E メールの作成時に容易に変更できます。このダイアログ・ボックスで、E メールの内容を確認してから「**OK**」をクリックして E メールを送信します。
  - 障害の送信: 検出結果を障害として送信するには、「障害の送信」をクリックします。 これにより、「障害追跡システムの選択」ダイアログ・ボックスが開きます。
    - 「**ClearQuest**」を選択し、「**OK**」をクリックすると、「添付ファイル名」ダイアログ・ボックスが開きます。そこで、障害に添付される検出結果バンドルの名前を指定してから、「**OK**」をクリックします。 Rational ClearQuest にログインして、検出結果を送信します。
    - **Quality Center** を選択して「**OK**」をクリックすると、ログイン・ダイアログ・ボックスが開き、Quality Center にログインして検出結果を送信できます。

- いずれかの「**Team Foundation Server**」オプションを選択すると、ダイアログ・ボックスが開き、障害追跡システムにログインしてその他の構成詳細を入力するように求めるプロンプトが出されます。

注**:** Rational Team Concert は、macOS でサポートされている唯一の障害追跡システムです。

カスタム検出結果の「検出結果の詳細」ビュー **(AppScan Source for Analysis** でのみ使用可能**)**

カスタム検出結果の「検出結果の詳細」ビューには、以下の編集可能な追加情報が表示されます。

- ファイル
- 行
- 列
- API

また、 193 ページの『詳細セクション』を編集する方法は、一部のフィールドについては標準的な検出結果と異なります (例えば、カスタム検出結果の分類はリスト形式で表示されます)。

検出結果の変更の取り消し

検出結果を変更した場合に、このトピックで説明されている方法で、変更を取り消す (元の値に戻す) ことができます。

このタスクについて

検出結果の変更を取り消すには、さまざまな方法があります。

- 『「変更された検出結果」ビューでの変更の取り消し』: この方法では、取り消したい変更が含まれているアプリケーションの評価が開かれている必要があります。この方法は、複数の変更済み検出結果を元に戻す場合に便利です。
- 196 ページの『検出結果が表示されているその他のビューでの変更の取り消し』: この方法では、評価が開かれている必要があります。この方法は、ある検出結果に複数の変更を行っており、それらの変更の一部を元に戻したい場合に特に便利です。例えば、ある検出結果の重大度と分類を変更しており、分類に対する変更はそのままにして、重大度を元に戻したい場合には、この方法が最適です。
- 196 ページの『「プロパティー」ビューの「変更された検出結果」タブでの変更の取り消し (AppScan Source for Analysis のみ)』: この方法は、開かれている評価がないアプリケーションの変更を取り消す場合に便利です。また、複数の変更済み検出結果の変更を元に戻す場合にも使用できます。

「変更された検出結果」ビューでの変更の取り消し

手順

1. 「変更された検出結果」ビューで、元に戻したい変更済み検出結果を選択します。 Windows ではキーボードの Ctrl キーとシフト・キーを、macOS では command キーと shift キーを使用して、複数の検出結果を選択することができます。

2. 「変更の削除」をクリックするか、または選択項目を右クリックしてメニューから「変更の削除」を選択します。

タスクの結果

このアクションにより、検出結果に行われたすべての変更が取り消されます。ある検出結果に複数の変更を行っており、それらの変更の一部を元に戻したい場合は、『検出結果が表示されているその他のビューでの変更の取り消し』で説明されている方法を使用してください。

検出結果が表示されているその他のビューでの変更の取り消し

このタスクについて

検出結果表が含まれている任意のビューで、「列の選択と順序付け」アクションを使用して、表示する列を選択できます。この機能を使用して、「重大度 (オリジナル)」、「重大度 (カスタム)、「分類 (オリジナル)、 および「分類 (カスタム)」列を表示することができます。これらの列で、(検出結果表でのアクション、または「検出結果の詳細」ビューの使用により) 変更を元の値に戻すことができます。例えば、「重大度」または「重大度 (カスタム)」の値が 「高」で「重大度 (オリジナル)の値が「中」の検出結果の場合、以下のようなさまざまな方法で重大度レベルを「中」に戻すことができます。

- 検出結果表で、検出結果を右クリックして、メニューで「重大度の設定」 > 「中」と選択します。
- 検出結果を選択してから、「検出結果の詳細」ビューで「重大度」フィールドを「中」に設定します。

「プロパティー」ビューの「変更された検出結果」タブでの変更の取り消し (AppScan Source for Analysis のみ)

手順

1. 「エクスプローラー」ビューで、元に戻したい変更が含まれているアプリケーションを選択します。
2. 「変更された検出結果」ビューで、元に戻したい変更済み検出結果を選択します。キーボードの Ctrl キーとシフト・キーを使用して、複数の検出結果を選択することができます。
3. 「変更の削除」をクリックするか、または選択項目を右クリックしてメニューから「変更の削除」を選択します。

タスクの結果

このアクションにより、検出結果に行われたすべての変更が取り消されます。ある検出結果に複数の変更を行っており、それらの変更の一部を元に戻したい場合は、『検出結果が表示されているその他のビューでの変更の取り消し』で説明されている方法を使用してください。

検出結果の比較

評価は、「差分評価」アクションを使用して比較されます。 2 つの評価が比較されると、両者の差は「差分評価」ビューに表示されます。このビューには、新規の検出結果、修正された/存在しない検出結果、および共通の検出結果が表示されます。

「差分評価」ビューでは、以下のコントロールを使用できます。

- 差分評価: 選択された 2 つの評価の間の差分を表示します。
- 新規検出結果 (青): このツールバー・ボタンは、新規検出結果 (緑のラベルの評価ではなく、青いラベルの評価の検出結果) の表示を切り替えるときに使用します。
- 修正された/存在しない検出結果 (緑): このツールバー・ボタンは、修正された/存在しない検出結果 (青いラベルの評価ではなく、緑のラベルの評価の検出結果) の表示を切り替えるときに使用します。
- 共通 (白): このツールバー・ボタンは、2 つの評価に共通する検出結果の表示を切り替えるときに使用します。
- 次へ: 次の新規検出結果または修正された/存在しない検出結果のブロックに移動します。
- 前へ: 前の新規検出結果または修正された/存在しない検出結果のブロックに移動します。

「差分評価」ビューでの 2 つの評価の比較

手順

1. 左側のペインで、比較する 2 つの評価を選択します。
2. 「差分評価」ツールバー・ボタンをクリックするか、選択したものを右クリックし、メニューから「差分評価」を選択します。

メインメニュー・バーからの 2 つの評価の比較

手順

1. メインメニュー・バーで、「ツール」 > 「差分評価」を選択します。
2. 「差分評価」ダイアログ・ボックスで、2 つの評価を選択します。
3. 「**OK**」をクリックして、「差分評価」ビューで 2 つの評価の比較結果を開きます。

「自分の評価」ビューと「公開された評価」ビューでの評価の差分の検出

手順

1. いずれかのビューで、2 つの評価を選択します。
2. 「差分評価」ツールバー・ボタンをクリックするか、選択したものを右クリックし、メニューから「差分評価」を選択します。 これにより、「差分評価」ビューで 2 つの評価の比較結果が開きます。

カスタム検出結果

分析結果を補強するために、カスタム検出結果 を作成することができます。 これはユーザーが作成する検出結果であり、AppScan Source for Analysis が、現在開いている評価または選択されたアプリケーションにこの検出結果を追加します。 カスタム検出結果は、評価のメトリックに影響します。また、レポートに含めることができます。 一度作成されたカスタム検出結果は、自動的にアプリケーションの今後のスキャンに組み込まれます。

カスタム検出結果の動作は、それがどのビューから作成されたかによって異なります。

「検出結果」ビューから作成された場合、カスタム検出結果は以下のようになります。

- 現在開いている評価に適用されます。
- アプリケーションの一部として保存され、アプリケーション・プロパティーに表示されます。
- 同じアプリケーションの現在のスキャンと今後のスキャンに影響します。
- 直ちに評価のメトリックに影響します。

「プロパティー」ビューから作成された場合、または選択されたアプリケーションに対して「カスタム検出結果の追加」アクションが選択された場合、カスタム検出結果は以下のようになります。

- 選択されたアプリケーションに適用されます。
- アプリケーションが、スキャンされたアプリケーションである場合、現在の評価に追加されます。
- そのアプリケーションの今後のスキャンに含まれます。

コード・エディターから作成された場合は、以下のようになります。

- 評価が開いている場合、カスタム検出結果は「検出結果」ビューで作成された場合と同様に動作します。
- 開いている評価がない場合、カスタム検出結果は「プロパティー」ビューで作成された場合と同様に動作します。

ユーザーがカスタム検出結果を作成した後に、AppScan Source for Analysis が自動的にアプリケーションを保存します。 アプリケーションを変更せずに評価を変更することはできません。 ただし、評価がアプリケーションに関連付けられていない場合は、アプリケーションが変更されることはありません。

カスタム検出結果をアプリケーションに追加した場合、それらはそのアプリケーションの後続のスキャンに組み込まれ、除外することはできません。 カスタム検出結果を削除するには、評価から除外するかアプリケーションから削除する必要があります。

注: カスタム検出結果は、解決済み/存在しない となることはできません。

カスタム検出結果は、以下の属性で構成されます。

- 脆弱性タイプ (必須)

- 重大度 (必須)
- 分類 (必須)
- ファイル (必須)
- コンテキスト
- 行番号
- 列番号
- **API**
- 注
- バンドル

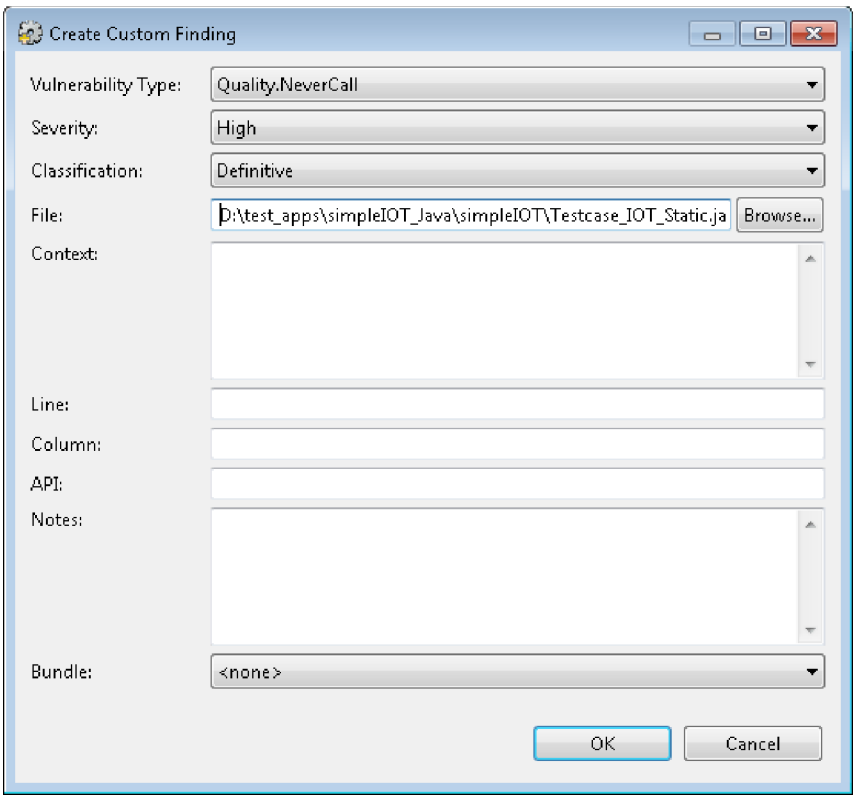
「プロパティー」ビューでのカスタム検出結果の作成

アプリケーションの「プロパティー」ビューからカスタム検出結果を作成または編集した場合、それは現在の評価の結果と今後のスキャンに影響します。

手順

1. 「エクスプローラー」ビューでアプリケーションを選択します。
2. 「プロパティー」ビューで、「カスタム検出結果」タブを選択します。
3. ツールバーで「カスタム検出結果の作成」をクリックします。
4. 「カスタム検出結果の作成」ダイアログ・ボックスで、以下の必須項目を追加します。
   - 脆弱性タイプ
   - 重大度
   - 分類
   - ファイル

   オプションで、コンテキスト、行番号、列、API、注釈、およびバンドルの指定を追加します。



5. 「**OK**」をクリックして、カスタム検出結果をアプリケーションに保存します。

「プロパティー」ビューでのカスタム検出結果の変更または削除

アプリケーションの「プロパティー」ビューからカスタム検出結果を作成または編集した場合、それは現在の評価の結果と今後のスキャンに影響します。

手順

1. 検出結果を選択します。 カスタム検出結果を削除する場合は、削除する検出結果のグループを選択できます。
2. カスタム検出結果を変更するには、ツールバーの「選択した検出結果の編集」をクリックし、以前に定義した検出結果情報を変更します。
3. 1 つ以上のカスタム検出結果を削除するには、ツールバーの「選択した検出結果の削除」をクリックします。

検出結果ビューでのカスタム検出結果の作成

カスタム検出結果の作成または管理は、複数の検出結果ビュー (例えば、「検出結果」ビューや「カスタム検出結果」ビュー) で行うことができます。

ビューからカスタム検出結果を作成すると、新規検出結果が現在の評価に追加され、評価メトリックが更新されます。

検出結果ビューにカスタム検出結果を追加する場合は、ビューの「カスタム検出結果の作成」ツールバー・ボタンをクリックします。これにより、「カスタム検出結

果の作成」ダイアログ・ボックスが開きます。このダイアログ・ボックスでは、199 ページの『「プロパティー」ビューでのカスタム検出結果の作成』で説明しているのと同じ要領で設定します。

カスタム検出結果を削除するには、評価から除外するかアプリケーションから削除する必要があります。または 200 ページの『「プロパティー」ビューでのカスタム検出結果の変更または削除』の指示に従ってください。これらのアクションは、他の検出結果ビューでは使用できません。

ソース・コード・エディターでのカスタム検出結果の作成

このタスクについて

ソース・コード・エディターを使用してカスタム検出結果を追加するときは、以下の条件が適用されます。

- ソース・コード・エディター内で表示されているソース・ファイルが、現在開いている評価に属する場合、カスタム検出結果は評価と関連アプリケーションに追加されます。
- カスタム検出結果が、現在開いている評価に属していない場合、そのカスタム検出結果は、該当するソース・ファイルを含むアプリケーションのみに追加されます。
- ソース・ファイルが複数のアプリケーションに属する場合、または AppScan Source for Analysis がアプリケーションを判別できない場合、適切なアプリケーションを選択する必要があります。

ソース・コード・エディターでカスタム検出結果を作成すると、「カスタム検出結果の作成」ダイアログ・ボックスにはエディターからの情報が事前に取り込まれます。

- ファイル: 現在開いているファイルの名前。
- コンテキスト: エディター内で選択されている任意のテキスト。テキストが選択されていない場合、現在カーソルが置かれている行がコンテキストになります。複数の行が選択されている場合は、選択されたすべての行がコンテキストになります。
- 行番号および列番号: 現在の行番号および列番号

エディターからカスタム検出結果を作成するには、以下のようにします。

手順

1. カスタム検出結果として追加するコード行を選択します。
2. 選択したものを右クリックし、メニューから「カスタム検出結果の作成」を選択します。 「カスタム検出結果の作成」ダイアログ・ボックスには、ファイル、コンテキスト、列番号、および行番号が取り込まれます。
3. 「脆弱性タイプ」、「重大度」、および「分類」を選択します。 オプションで、API、注釈、およびバンドルの指定を追加します。
4. 「**OK**」をクリックします。

セキュリティー問題の解決と修復支援の表示

AppScan Source は、セキュリティー・エラーまたは一般的な設計上の障害について警告し、解決のプロセスを支援します。 AppScan Source セキュリティー・ナレッジ・データベースや、内部または外部のコード・エディターが、このプロセスで役立ちます。

このタスクについて

AppScan Source セキュリティー・ナレッジ・データベース は、検出結果を修正する方法を提示します。それぞれの脆弱性についてのこのコンテキスト内情報は、根本原因とリスクの重大度についての正確な説明、および実施可能な修復のアドバイスを提供します。例えば `strcpy()` (バッファー・オーバーフロー・タイプ) について、重大度が高いものとして説明し、以下のような修復を支援する情報を提供します。

> `strcpy` では、宛先バッファーでのオーバーフローが起こりやすくなります。宛先バッファーの長さがわからないため、宛先バッファーを上書きしてしまわないように確認する作業を実行できないためです。長さのパラメーターをとる `strncpy` の使用を検討してください。`strncpy` もセキュリティー上のリスクになりますが、程度はかなり低くなります。

AppScan Source セキュリティー・ナレッジ・データベースを参照するには、以下のようにします。

手順

- AppScan Source for Analysis で、「修復支援」ビューを開き、検出結果表内の 1 つの検出結果を選択します。その特定の検出結果の修復支援が表示されます。または、メインメニュー・バーから「ヘルプ」 > 「セキュリティー・ナレッジ・データベース」を選択して、ブラウザーで AppScan Source セキュリティー・ナレッジ・データベース全体を開きます。
- AppScan Source for Development (Eclipse プラグイン) で、「修復支援」ビューを開き、検出結果表内の 1 つの検出結果を選択します。その特定の検出結果の修復支援が表示されます。
- AppScan Source for Development (Visual Studio プラグイン) で、検出結果表内の 1 つの検出結果を選択します。メインメニュー・バーから「**IBM Security AppScan Source**」 > 「ナレッジ・データベース・ヘルプ」を選択するか、検出結果を右クリックし、メニューから「ナレッジ・データベース・ヘルプ」を選択します。これにより、選択した検出結果に対する修復支援が開きます。

エディターでのソース・コードの分析

AppScan Source では、内部エディターでソース・コードを分析したり、変更したりできます。あるいは、さまざまな外部エディターから選択することもできます。

外部エディターを使用すると、AppScan Source for Analysis での結果を確認し、任意の開発環境でコードを変更することができます。使用できる外部エディターには、以下のものがあります。

表 13. サポートされる外部エディター

エディター	プラットフォーム
Eclipse (サポートされる Eclipse のバージョンについては、AppScan Source のシステム要件を参照してください)	Windows および Linux
ノートパッド	Windows
vi	Linux
システム・デフォルト	Windows および Linux

注: WAR ファイル内のソース・ファイルを編集することはできません。

ソース・コードをエディターで表示または変更するには、以下のいずれかのオプションを選択します。

- 検出結果表で検出結果をダブルクリックします。 内部エディターでコードの該当行が開きます。
- 検出結果表で検出結果を右クリックし、「内部エディターで開く」または「外部エディターで開く」 > **<editor>** を選択します (**<editor>** は上記の表に記載されているサポートされる外部エディターのいずれかです)。
- トレース・ノードを選択して、「内部エディターで開く」または「外部エディターで開く」 > **<editor>** ツールバー・ボタンを選択します。または、選択したトレース・ノードを右クリックし、メニューから「内部エディターで開く」または「外部エディターで開く」 > **<editor>** を選択します。

ファイルをエディターで開いた場合、マーカーは、検出結果を表すファイル内の場所を示します。これらをたどって検出結果表に戻るには、エディターでコードの行を右クリックし、次にメニューから「検出結果ビューで表示」を選択します。

サポートされる注釈と属性

コードの装飾 に使用される一部の注釈および属性は、スキャン中に処理されます。スキャン中に、サポートされる注釈または属性がコード内で検出されると、装飾されたメソッドに汚染されたコールバックとしてのマークを付けるために、その情報が使用されます。 汚染されたコールバックというマークが付けられたメソッドは、そのすべての引数に汚染されたデータが入っているものとして扱われます。これにより、トレースでの検出結果が増加します。 サポートされる注釈および属性のリストを、このヘルプ・トピックで示します。

- 『サポートされる Java 注釈』
- 204 ページの『サポートされる AppScan Source Java 注釈』
- 205 ページの『サポートされる Microsoft .NET 属性』

サポートされる Java 注釈

表 14. サポートされる *Java* 注釈

注釈	省略形
javax.xml.ws.WebServiceProvider	@WebServiceProvider
javax.jws.WebService	@WebService

表 14. サポートされる *Java* 注釈 (続き)

注釈	省略形
javax.jws.WebMethod	@WebMethod

サポートされる AppScan Source Java 注釈

- 『AppScan Source アノテーションの使用』
- 205 ページの『@ValidatorMethod』
- 205 ページの『@SuppressSecurityTrace』
- 205 ページの『@CallbackMethod』

AppScan Source を使用して Java をスキャンする場合、@ValidatorMethod、@CallbackMethod、および @SuppressSecurityTrace の各メソッド・レベルのアノテーションがサポートされます。

AppScan Source アノテーションの使用

以下のステップを実行することで、アノテーションを使用することができます。

1. アノテーションのサポートは、デフォルトで有効にされています。アノテーション .jar ファイルは、<install_dir>¥lib¥SecurityAnnotations.jar (<install_dir> は AppScan Source インストール済み環境がある場所です) です。
2. プリコンパイル済みクラスの .war ファイルまたは .jar ファイルをスキャンしている場合、アノテーション付きソースが含まれる Java プロジェクトを検索してください。
3. SecurityAnnotations.jar をプロジェクトのクラスパスに追加します。
4. プロジェクトを再ビルドします。

アノテーションは、スキャンの前にソース・コードに追加することも、スキャンの後で、誤検出を特定して除去するためのトリアージ時に追加することもできます。

アノテーションは、ユーザーの知識をソース・コードに直接、セキュリティー・アノテーションの形式で挿入できるようにするために提供されています。アノテーションは、コードの部分が安全であることを宣言するために使用されるものであるため、非常に慎重に使用する必要があります。セキュリティー脆弱性についてスキャンする必要があるコードには、アノテーションを使用しないでください。アノテーションを使用した場合、セキュリティー・アナリストは <data_dir>¥config¥scanner.ozsettings (<data_dir> は、ご使用の AppScan Source プログラム・データの場所です。詳しくは、 360 ページの『インストールとユーザー・データ・ファイルの場所』 を参照してください。) の機能を無効にすることで、アノテーションを無視するように選択できます。このファイルで、以下の設定を見つけます。

```
<Setting
name="process_security_annotations"
value="true"
default_value="true"
description="When turned on, security annotations in the
```

```
     source code will be processed by AppScan Source."
display_name="Process Security Annotations"
type="bool"
/>
```

この機能を無効にするには、value="true" を value="false" に変更します。

@ValidatorMethod

バリデーターは、入力データのチェックを行うメソッドであり、多くの場合、入力が有効か無効かを示すブール値を返します。バリデーターを使用して入力を受け入れたり拒否したりするのではなく、ユーザー入力を受け入れ可能なフォーマットに変更することができます。これらのメソッドは、サニタイザーと呼ばれます。

@ValidatorMethod アノテーションを使用して、アプリケーション・ソース・コード内のすべてのバリデーター・メソッドおよびサニタイザー・メソッドを識別することができます。 AppScan Source スキャン中に、この情報を使用して、これらのメソッドを通過するデータ・フローを除去するので、データは安全と見なされるようになります。

注: 現在、アノテーション付きのメソッドのどのパラメーターを検証する必要があるかを指定するために提供されているものはありません。 AppScan Source スキャン中は、すべての入力パラメーターが検証対象と見なされます。

@SuppressSecurityTrace

このアノテーションが付けられたメソッドを経由するすべてのトレースは削除されます。これは、特定のグループのトレースが、誤検出として識別されたり、他のトレースに比べて重要度や関心度が低いとして識別されている場合に便利です。このアノテーションを使用して、これらのトレースをフィルタリングによって除去したり、見やすくするために非表示にしたりすることができます。

@CallbackMethod

このアノテーションを使用して、アプリケーションへのコールバックやエントリー・ポイントを識別します。すべての引数が、汚染を広めていると見なされます。

サポートされる Microsoft .NET 属性

表 15. サポートされる *Microsoft .NET* 属性

属性	省略形
System.Web.Services.WebServiceAttribute	WebService
System.Web.Services.WebMethodAttribute	WebMethod

第 6 章 AppScan Source トレース

AppScan Source トレースを使用して、入力データの検証およびエンコードが自社のソフトウェア・セキュリティー・ポリシーに沿うものであることを確認できます。 入出力トレース・データの基になる検出結果を検索したり、メソッドを検証ルーチンおよびエンコード・ルーチン、ソースまたはシンク、コールバック、あるいは汚染伝播元としてマークすることができます。

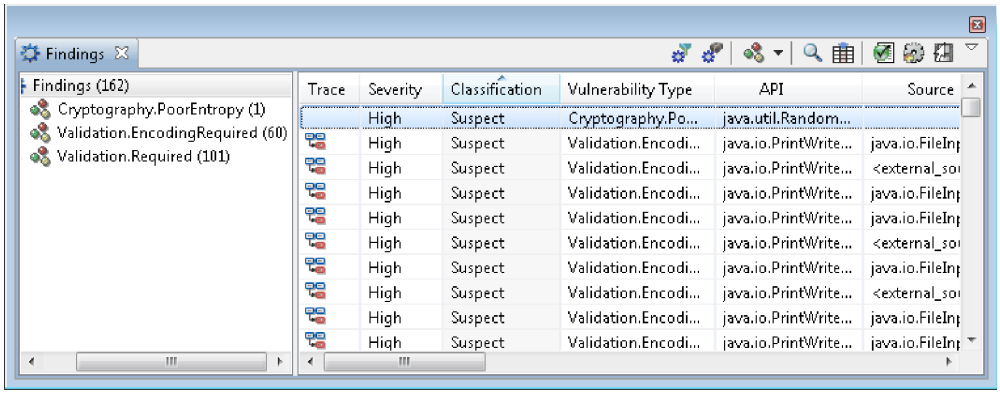
AppScan Source は、複数のモジュールおよび言語にまたがって、アプリケーション全体のデータの流れをトレースします。これは、潜在的に危険なデータのパスを呼び出しグラフに表示し、アプリケーションのどの部分が脆弱になりやすいかを示します。

トレースを実行して、承認された入力検証ルーチンと入力エンコード・ルーチンがアプリケーション内に存在しないことを検出することにより、SQL 注入やクロスサイト・スクリプティングなどの入力検証攻撃を防ぐことができます。 対話式に呼び出しグラフ全体のトレースを実行し、「トレース」ビューで直接ソースをクリックして、開発環境または任意のコード・エディターでソースを表示します。 また、トレース機能を使用してポリシーを適用することもできます。これにより、正しい入力データの検証とエンコードに必要な承認済みのルーチン、汚染伝播元、シンクとソースを特定して、次回からのスキャンで使用できるようになります。

スキャンの結果としてトレースが実行された場合は、それぞれの検出結果の入力検証ルーチンと入力エンコード・ルーチン、脆弱性、シンク、ソース、または汚染伝播元を「トレース」ビューから作成することができます。 例えば、AppScan Source for Analysis の任意のルーチンに検証ルーチンのマークを付けて AppScan Source セキュリティー・ナレッジ・データベース に追加すると、このルーチンが呼び出されるデータ・パスの `Validation.Required` または `Validation.Encoding.Required` の検出結果は、次回からのスキャンでは報告されなくなります。「トレース」ビューでは、脆弱性をソースまたはシンク (あるいはその両方) として定義したり、メソッドを汚染伝播元、汚染されたコールバック、または汚染の可能性がないメソッドのいずれかとして識別することもできます。

AppScan Source トレース スキャン結果

スキャンの結果には、AppScan Source トレース によって検出されたトレース情報が含まれる場合があります。「トレース」列内のアイコンは、呼び出しグラフのトレース情報が存在することを示します。



スキャンの結果、`Validation.Required` というタイプの検出結果と `Validation.EncodingRequired` というタイプの検出結果が生成されることがあります。これらの検出結果は、外部ソースからデータを読み取る場合のソース・コード内の位置、またはデータを外部シンクに保存する場合のソース・コード内の位置を示します。 悪質なデータやエラー・データによる被害を防ぐためにデータの検証やエンコードが必要になるため、こうしたコードがスキャンによって検出されると、フラグが立てられます。

検証とエンコード

検証 とは、入力データの形式が正しいことを確認するためのチェック・プロセスです。 検出結果 `Validation.Required` は、指定されたソースからシンクまでのデータ・パスで検証が発生しなかったことを示します。 検証には、データの長さを最大長までに制限するような単純なものから、名前とアドレスの形式が正しいかどうかなどの複雑なものまであります。 また、検証により、SQL 注入などの攻撃を可能にする不正な文字シーケンスを検出して、こうした攻撃をチェックすることもできます。

エンコード とは、データを正しい形式に変換するプロセスのことです。 検出結果 `Validation.EncodingRequired` は、指定されたソースからシンクまでのデータ・パスでエンコードが発生しなかったことを示します。 エンコードには、文字のエスケープのような単純なものから、データの暗号化のような複雑なものまであります。また、エンコードにより、クロスサイト・スクリプティングなどの攻撃の原因となる文字をエスケープして、これらの攻撃を防ぐこともできます。

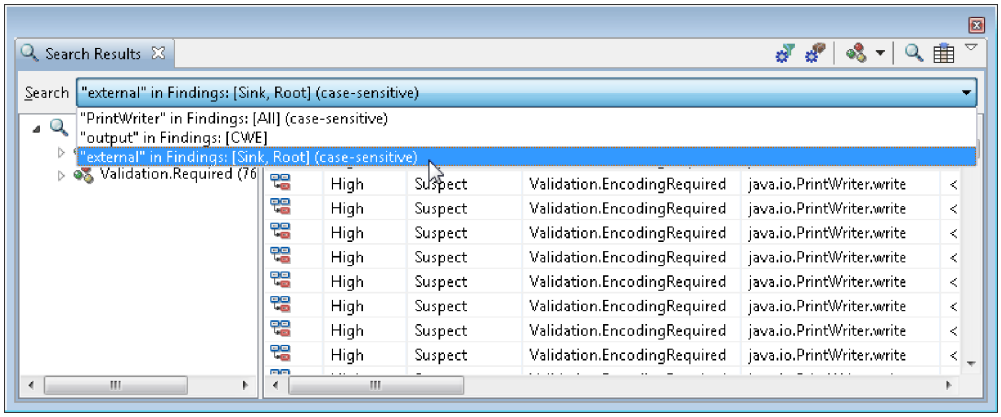
最初にスキャンを実行すると、AppScan Sourceにより、検出結果が要注意セキュリティー検出結果として示されることがあります。特定のソースに適用される検証ルーチンまたはエンコード・ルーチンを作成した場合、ソースからデータを受信した後に、指定された検証ルーチンまたはエンコード・ルーチンを呼び出さないと、AppScan Source for Analysis は、検出結果を確定 (要確認ではなく) として報告します。

評価データにより、既知のソースからのデータがプロジェクト全体を通じて追跡されます。既知のソースから既知のシンクまでのデータを追跡できる場合、指定された検証ルーチンとエンコード・ルーチンを使用して、バインドされていない入力データに対して悪質な攻撃が発生する可能性のないことを確認することができます。

AppScan Source トレース の検索

トレースの検出結果をグループ化したい場合は、ソースまたはシンクを検索することができます。 これにより、トレースの検出結果が「検索結果」ビューに表示されます。

「トレース」ビューで、「同じタイプのルーチンを持つトレースの検索」をクリックします。その後、「検出結果の検索」ダイアログ・ボックスで、ソース、シンク、逸失シンク (仮想逸失シンクを含む)、仮想逸失シンク、またはトレース呼び出しを選択して、対象の文字列を含むトレース情報を検出結果から抽出します。検索結果は、「検索結果」ビューに累積情報として表示されます。 このビューで、さらに詳細な検索を実行することができます。



入出力トレース

入出力トレース 情報は、既知のソースからシンク または逸失シンク へのデータを AppScan Source for Analysis で追跡できる場合に生成されます。

入出力トレース

汚染されたソースからシンクまたは逸失シンクへのデータをコード分析によって追跡できる場合は、入出力トレース情報が生成されます。 トレースの起点は、汚染の原因となるソースからデータを受け取って一連の呼び出しに渡すメソッドです。汚染されたデータは、これらの一連の呼び出しを経由して、最終的に保護されていないシンクに書き込まれます。

ソースとシンク

- ソース: ソースは、プログラムに対する入力情報です。ソースには、ファイル、サーブレット要求、コンソール入力、ソケットなどがあります。 多くの入力ソースの場合、コンテキストと長さについてはバインドされていないデータが返されます。 チェックされていない入力については、汚染されているものと見なされます。ソースは、すべての検出結果表の「ソース」列に表示されます。

- シンク: シンクは、データの書き込み先となる任意の外部フォーマットです。 シンクの例としては、データベース、ファイル、コンソール出力、ソケットなどがあります。 データをチェックせずにシンクに書き込むと、重大なセキュリティー脆弱性となる可能性があります。
- 逸失シンク: 逸失シンクとは、トレースできなくなった API メソッドのことです。

  注: 逸失シンクは JavaScript の検出結果には適用されません。

「トレース」ビューの使用

このタスクについて

「トレース」ビューでは、検出結果についての単一の入出力トレースを確認することができます。 このビューのペインは、以下の 3 つのパネルに分かれています。

- 入力スタックと出力スタック
- データ・フロー
- 呼び出しグラフ図

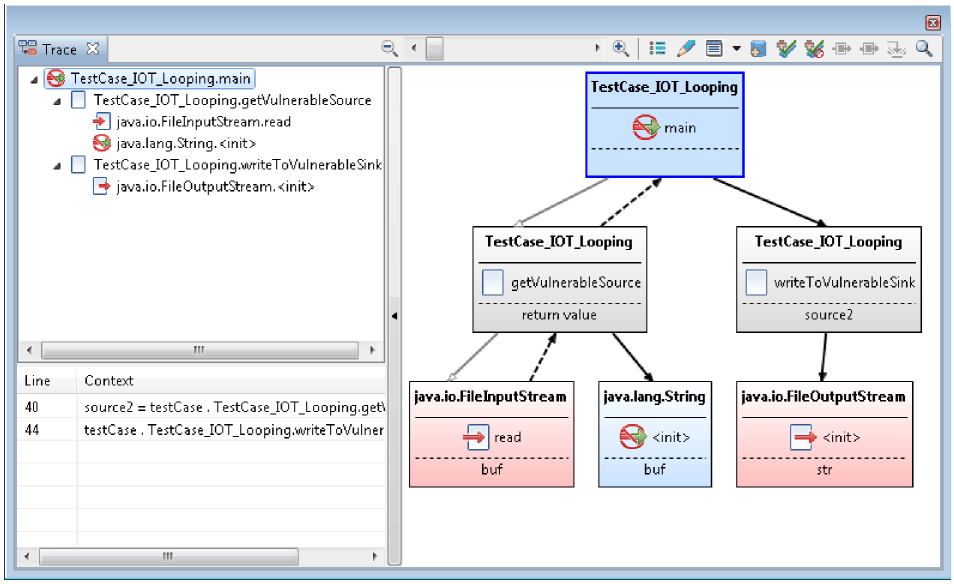
これらのパネルについては、 211 ページの『「トレース」ビューの入出力スタック』で詳しく説明しています。

注: JavaScript トレースでは、呼び出しグラフ図ではなく 213 ページの『JavaScript ステートメント・グラフ』 が表示されます。

AppScan Source トレースを参照するには、以下のようにします。

手順

1. スキャンを実行して、「検出結果」ビューでトレース結果を確認します。
2. 「表示」メニューから「トレース」ビューを開きます。
3. 検出結果表で、「トレース」アイコンが表示されている行を選択します。「トレース」ビューに、トレースの詳細が表示されます。



「トレース」ビューの入出力スタック

左上のパネルに入力スタックと出力スタックが表示されます。スタックとは、ソース (入力スタック) とシンク (出力スタック) のいずれかで終了する一連の呼び出しのことです。

データ・フロー

左下のパネルには、選択したメソッドのデータ・フローが表示されます。 このパネルで、メソッド呼び出しまたは割り当て内のデータ・フローを確認できます。 データ・フロー・セクションには、ソース・コード内の項目とコンテキストの位置を示す行番号が表示されます。

呼び出しグラフ

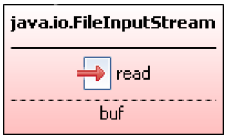
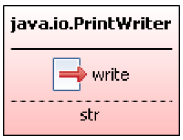
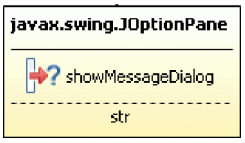
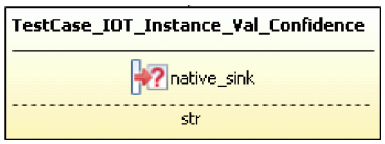
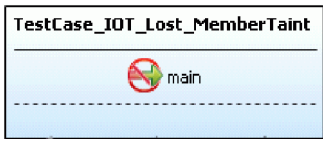
注: JavaScript トレースでは、呼び出しグラフ図ではなく 213 ページの『JavaScript ステートメント・グラフ』 が表示されます。

呼び出しグラフは、チャートがグラフィック表示されたものです。 各メソッド呼び出しは、グラフ内に長方形で表示され、その中にクラス名とメソッド名が表示されます。

- 赤で表示されたメソッドは、そのメソッド呼び出しがソースまたはシンク (あるいはその両方) であることを示しています。
- 逸失シンク とは、トレースできなくなった API メソッドのことです。仮想逸失シンク とは、仮想関数 (複数の実装を持つことができる関数) でもある逸失シンクのことです。 黄色で表示されたメソッドは、そのメソッド呼び出しが逸失シンクまたは仮想逸失シンクであることを示しています。
- 青で表示されたメソッドは、そのメソッド呼び出しが検証/エンコード・ルーチンではないことを示しています。
- グレーは、その他のすべてのトレース・ノード・タイプを表します。

各メソッド呼び出しは、クラス名、メソッド名、汚染された引数名の 3 つのセクションに分けられています。 メソッド呼び出しの吹き出しテキストに詳細が表示されます。

矢印付きの線は、メソッド間の呼び出しを示しています。 塗りつぶしなしの矢印は、既知の汚染されたデータが呼び出し内に存在しないことを示しています。塗りつぶしの矢印は、汚染されたデータのフローを示しています。 破線の矢印は、戻りステートメントを示しています。

記号	説明
	既知の汚染されたデータが存在しないメソッド呼び出し
	汚染されたデータが存在するメソッド呼び出し
	汚染されたデータが存在する戻りステートメント
	ソース (赤): 信頼できない可能性があるデータの発生元であるメソッド、関数、またはパラメーター。
	シンク (赤): 汚染されたデータに対して脆弱だと考えられるメソッドまたは関数、あるいは、使用するのは危険だと考えられるメソッドまたは関数。
	逸失シンク (黄色): 汚染されたデータに対して脆弱だと考えられるメソッドまたは関数、あるいは、使用するのは危険だと考えられるメソッドまたは関数。
	仮想逸失シンク (黄色): 複数の具体的な実装に解決されるタイプの逸失シンク。
	検証/エンコード・ルーチンなし (青)。 API を「検証/エンコード・ルーチンなし」としてマークすると、この API がデータ検証を実行しないことが示されます。
	汚染伝播元: 汚染を 1 つ以上のパラメーター、戻り値、または this ポインターに伝播する関数またはメソッド。

ヒント:

- 「トレース」ビューで、グラフのトレース・ノードの上にマウスを移動すると、ノードに関する情報が表示されます。
- ビューの左側にある 2 つパネル (「入出力スタック」パネルと「データ・フロー」パネル) は、呼び出し図グラフを見やすくするために省略表示することがで

きます。これらのパネルを省略表示するには、「ツリー・ビューの非表示」矢印ボタンを選択してください。非表示になったこれらのパネルを表示するには、「ツリー・ビューの表示」矢印ボタンを選択します。

- スクロール・バーを移動してズームインまたはズームアウトすることにより、詳細情報を拡大表示したり、表示範囲を広げたりすることができます。ポインターをズーム・スクロール・バー上に移動すると、現在のズーム・レベルが示されます。最大レベルまでズームインするには、「ズーム率 **200%**」をクリックします。可能な限りズームアウトするには、「適合ズーム」をクリックします。

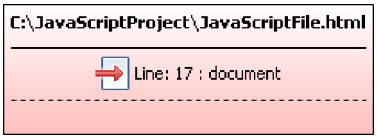
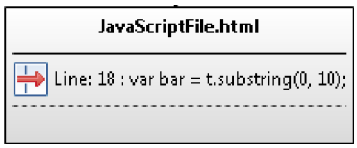
JavaScript ステートメント・グラフ

JavaScript トレースのステートメント・グラフ・セクションには、ステートメント間のデータ・フローが表示されます。

グラフ内で、各ステートメントは、以下の情報を提供する長方形です。

- 影響を受けるファイルのパスとファイル名。次のステートメントが同じファイルにある場合は、ファイル名のみがリストされます。
- ステートメントが含まれている行番号。
- 選択可能な場合は、対象となるコードのセクション。
- 長方形が赤い場合、ステートメントはソースまたはシンク (あるいはその両方) です。
- 長方形がグレーである場合、ステートメントは汚染伝播元です。
- ステートメントの吹き出しテキストに詳細が表示されます。

矢印付きの線は、ステートメント間のデータ・フローを示しています。

記号	説明
	汚染されたデータのフロー
	ソース (赤): 信頼できない可能性があるデータの発生元であるステートメント。
	シンク (赤): 汚染されたデータに対して脆弱だと考えられるステートメント、あるいは、使用するのは危険だと考えられるステートメント。
	汚染伝播元: 汚染を 1 つ以上のパラメーター、戻り値、またはこのポインターに伝播するステートメント。

ヒント:

- 「トレース」ビューで、グラフのトレース・ノードの上にマウスを移動すると、ノードに関する情報が表示されます。
- ビューの左側にある 2 つパネル (「入出力スタック」パネルと「データ・フロー」パネル) は、呼び出し図グラフを見やすくするために省略表示することがで

きます。これらのパネルを省略表示するには、「ツリー・ビューの非表示」矢印ボタンを選択してください。非表示になったこれらのパネルを表示するには、「ツリー・ビューの表示」矢印ボタンを選択します。

- スクロール・バーを移動してズームインまたはズームアウトすることにより、詳細情報を拡大表示したり、表示範囲を広げたりすることができます。ポインターをズーム・スクロール・バー上に移動すると、現在のズーム・レベルが示されます。最大レベルまでズームインするには、「ズーム率 **200%**」をクリックします。可能な限りズームアウトするには、「適合ズーム」をクリックします。

エディターでのソース・コードの分析

AppScan Source では、内部エディターでソース・コードを分析したり、変更したりできます。あるいは、さまざまな外部エディターから選択することもできます。

外部エディターを使用すると、AppScan Source for Analysis での結果を確認し、任意の開発環境でコードを変更することができます。使用できる外部エディターには、以下のものがあります。

表 *16.* サポートされる外部エディター

エディター	プラットフォーム
Eclipse (サポートされる Eclipse のバージョンについては、AppScan Source のシステム要件を参照してください)	Windows および Linux
ノートパッド	Windows
vi	Linux
システム・デフォルト	Windows および Linux

注**:** WAR ファイル内のソース・ファイルを編集することはできません。

ソース・コードをエディターで表示または変更するには、以下のいずれかのオプションを選択します。

- 検出結果表で検出結果をダブルクリックします。 内部エディターでコードの該当行が開きます。
- 検出結果表で検出結果を右クリックし、「内部エディターで開く」または「外部エディターで開く」 > **<editor>** を選択します (**<editor>** は上記の表に記載されているサポートされる外部エディターのいずれかです)。
- トレース・ノードを選択して、「内部エディターで開く」または「外部エディターで開く」 > **<editor>** ツールバー・ボタンを選択します。または、選択したトレース・ノードを右クリックし、メニューから「内部エディターで開く」または「外部エディターで開く」 > **<editor>** を選択します。

ファイルをエディターで開いた場合、マーカーは、検出結果を表すファイル内の場所を示します。これらをたどって検出結果表に戻るには、エディターでコードの行を右クリックし、次にメニューから「検出結果ビューで表示」を選択します。

検証とエンコードの有効範囲

「トレース」ビューから、カスタムの検証ルーチンとエンコード・ルーチン (AppScan Source セキュリティー・ナレッジ・データベースに格納されているルーチン) を指定することにより、データを汚染されたデータとしてではなくチェック済みデータとしてマークを付けることができます。これらのルーチンは、カスタム・ルール・ウィザードを使用して、有効範囲に基づいて定義します。

検証ルーチンとエンコード・ルーチンを作成する手順については、 226 ページの『例 4: 詳細な検証』を参照してください。

検証ルーチンまたはエンコード・ルーチンは、有効範囲に基づいて以下のように定義されます。

- 『API 単位』
- 『呼び出しサイト単位』

API 単位

API 単位の検証ルーチンとエンコード・ルーチンは、1 つのプロジェクトだけに関連付けることも、複数のプロジェクトに関連付けることもできます。

API 単位のルーチンは、特定のソース API のすべてのインスタンスからのすべてのデータの汚染を除去します。 例えば、API からのすべての入力に対する検証ルーチンを次のように指定することができます。

```
javax.servlet.ServletRequest.getParameter
(java.lang.string):java.lang.string
```

API 単位のルーチンはサーバー上に格納されます。 プロジェクトを対象とする API 単位のルーチンは、プロジェクト内に格納されます。

呼び出しサイト単位

呼び出しサイト単位のルーチンは、常に 1 つのプロジェクトだけに関連付けられます。

呼び出しサイト単位のルーチンは、コード内の特定の場所からのデータの汚染を除去します。 呼び出しサイト単位の検証ルーチンまたはエンコード・ルーチンを作成する場合は、そのルーチンが特定の入力呼び出しサイトに対して適用されるように指定します。 呼び出しサイト単位のルーチンは、常にプロジェクト内に格納されます。

注: 呼び出しサイト単位のルーチンは、同じメソッド内の検証ルーチンに対するすべての呼び出しに適用されます。

AppScan Source トレース によるカスタム・ルールの作成

「トレース」ビューからカスタム・ルールを作成できます。「トレース」ビューでは、汚染伝播元のトレース、汚染の可能性のないトレース、またはシンクであるトレースを持つ検出結果をフィルタリングで除外することができます。 また、トレース内のメソッドを検証/エンコード・ルーチンとしてマークする (または、それらが検証/エンコード・ルーチンでないことを示す) こともできます。

このタスクについて

サンプルのソース・コードと出力内容や、検証ルーチンとエンコード・ルーチンの作成手順については、 221 ページの『例 2:「トレース」ビューでの検証ルーチンとエンコード・ルーチンの作成』を参照してください。

表 17. 「トレース」ビューのノードの有効なマーキング

選択されたメソッド	有効なマーキング
中間ノード	• 検証/エンコード・ルーチン • 汚染の可能性なし • 検証/エンコード・ルーチンなし
逸失シンク	• 汚染伝播元 • 汚染の可能性なし • シンク

手順

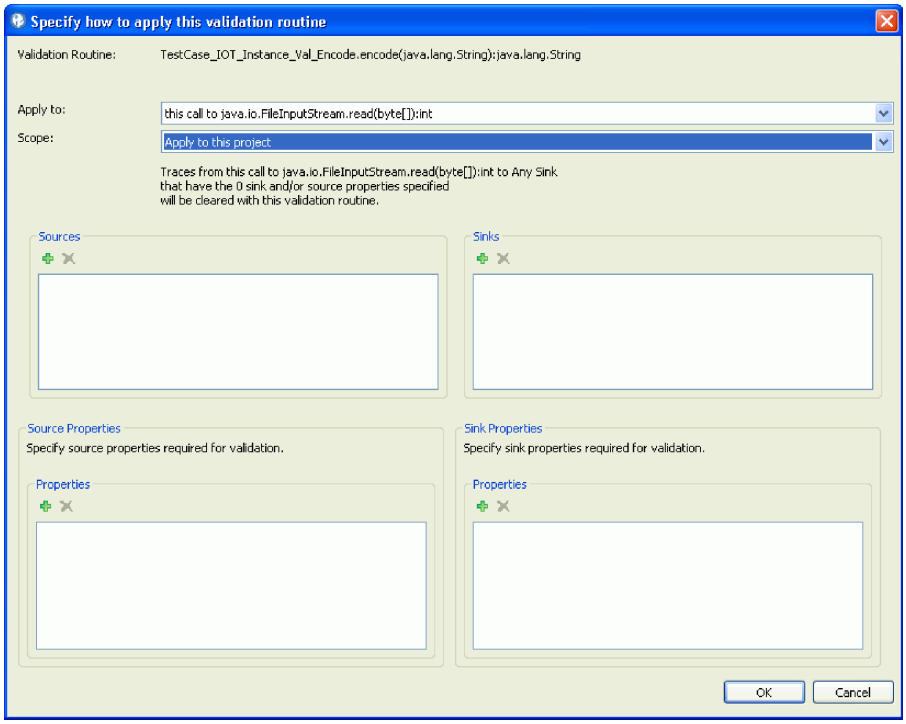
1. 「トレース」ビューで、カスタム・ルールを作成する対象のメソッドまたはノードを右クリックして、作成するカスタム・ルールを選択します。あるいは、メソッドまたはノードを選択してから、該当のカスタム・ルール・ツールバー・ボタンをクリックします。 ルーチンとメソッドのマーキング・オプションは、以下のとおりです。

オプション	説明
検証/エンコード・ルーチンとしてマークする	
検証/エンコード・ルーチンなしとしてマークする	
汚染伝播元としてマークする	
汚染の可能性なしとしてマークする	
シンクとしてマークする	

   注: 「トレース」ビューで、カスタム・ルールを作成する対象のメソッドの項目がない場合は、「カスタム・ルール・ウィザードを起動し、トレース・グラフにない検証ルーチンを追加します」をクリックします。 カスタム・ルール・ウィザードで、「検証/エンコード・ルーチンの選択」ページに進みます。 検証ルーチンを選択し、次のステップの指示に従って、場所、有効範囲、任意のソースまたはシンク、任意のプロパティーを指定します。このウィザードを使用して検証ルーチンを作成する方法について詳しくは、 224 ページの『例 2: カスタム・ルール・ウィザードでの検証ルーチンとエンコード・ルーチンの作成』を参照してください。
2. メソッドをシンクまたは検証/エンコード・ルーチンとしてマークするカスタム・ルールを作成している場合、詳細な設定が必要になる場合があります。

a. メソッドをシンクとしてマークする場合は、以下のシンク属性を指定します。
   - 脆弱性タイプ
   - 重大度

b. 検証ルーチンの場合は、検証ルーチンの適用対象にする場所と有効範囲、任意のソースまたはシンク、またはそれらのプロパティーを指定します。



- 適用対象:
  - <メソッド名> に対するこの呼び出し (呼び出しサイト単位): この呼び出しの入力に対してのみ適用します。
  - <メソッド名> に対するすべての呼び出し (API 単位): このメソッドに対するすべての呼び出しの検証ルーチンまたはエンコード・ルーチンに適用します。
  - <メソッド名> は考慮されません。すべての制約は下で指定されています): すべてのソースがルールの影響を受けるようにすることができます。
- 有効範囲:
  - このプロジェクトに適用: 選択すると、ルールはプロジェクト (.ppf) ファイルに格納されます。
  - すべてのプロジェクトに適用: この設定で作成された検証ルールはデータベースに格納されます。
- ソース: 検証ルーチンの適用対象にする入力ソースを選択します。 ソースを追加するには、「追加」をクリックし、「シグニチャーの選択」ダイアログ・ボックスからソースを選択します。 複数のソースを追加する場合は、「シグニチャーの選択」ダイアログ・ボックスからソースを複数選択できます。

- シンク: 検証ルーチンの適用対象にするシンクを選択します。 シンクを追加するには、「追加」をクリックし、「シグニチャーの選択」ダイアログ・ボックスから「シンク」を選択します。 複数のシンクを追加する場合は、「シグニチャーの選択」ダイアログ・ボックスからシンクを複数選択できます。
- ソース・プロパティー: ルールによって、特定のプロパティーを持つソース内で始まるトレースを消去する場合は、「**VMAT** プロパティーを追加」をクリックし、「プロパティーの選択」ダイアログ・ボックスからプロパティーを選択します。 複数のプロパティーを追加する場合は、「プロパティーの選択」ダイアログ・ボックスからプロパティーを複数選択できます。
- シンク・プロパティー: ルールによって、特定のプロパティーを持つシンク内で終了するトレースをフィルタリングで除外する場合は、「**VMAT** プロパティーを追加」をクリックし、「プロパティーの選択」ダイアログ・ボックスからプロパティーを選択します。 複数のプロパティーを追加する場合は、「プロパティーの選択」ダイアログ・ボックスからプロパティーを複数選択できます。

3. 「トレース」ビューでカスタム・ルールを作成した後で、コードを再度スキャンして、検出結果リストおよびトレースでルールが反映されていることを確認する必要があります。 「トレース」ビューで作成したカスタム・ルールは、「カスタム・ルール」ビューで表示および削除できます。 「カスタム・ルール」ビューでルールの詳細を表示するには、ルールを選択し、「カスタム・ルール情報」をクリックします。

トレースのコード例

このセクションでは、汚染されたデータをソースからシンクまで追跡する部分と、検証ルーチンとエンコード・ルーチンを作成する部分を示すコード例を示します。

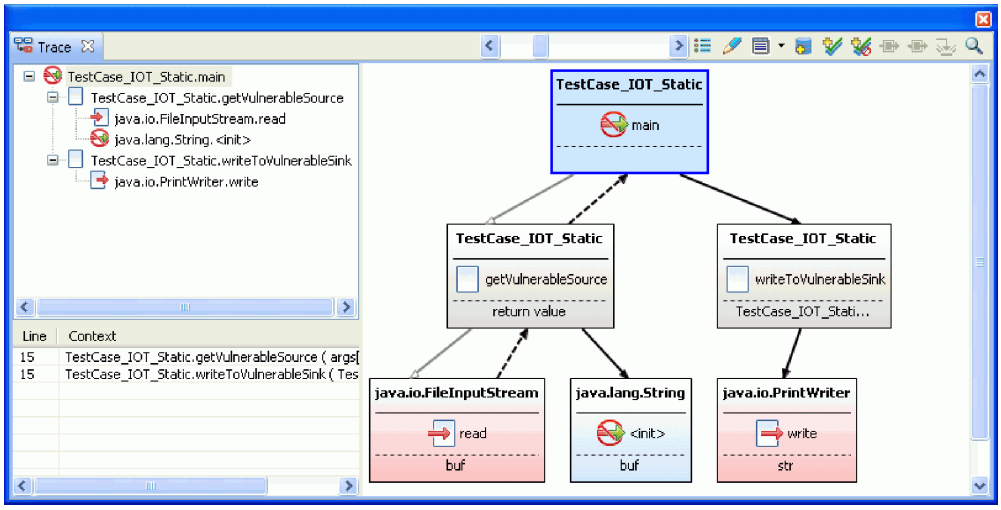
- 『例 1: ソースからシンク』
- 220 ページの『例 2: ソースからシンクへの変更』
  - 221 ページの『例 2:「トレース」ビューでの検証ルーチンとエンコード・ルーチンの作成』
  - 224 ページの『例 2: カスタム・ルール・ウィザードでの検証ルーチンとエンコード・ルーチンの作成』
- 225 ページの『例 3: ソースとシンクのファイルが異なる場合』
- 226 ページの『例 4: 詳細な検証』

例 1: ソースからシンク

以下のサンプル・コードでは、文字列を返す `getVulnerableSource` というメソッドが main メソッドから呼び出されます。 ただし、このメソッドではまったく未知のファイルからデータが読み込まれますが、返されるデータの妥当性は検査されません。 次に、この汚染されたデータが main メソッドから `writeToVulnerableSink` に渡されます。 `writeToVulnerableSink` メソッドは、受け取ったデータをファイルに書き出します。その際、データの妥当性は検査されません。

```
import java.io.*;

public class TestCase_IOT_Static {
  public static void main(String[] args) {
    try {
      writeToVulnerableSink(getVulnerableSource(args[0]));
    } catch (Exception e) {
    }
  }

  public static String getVulnerableSource(String file)
    throws java.io.IOException, java.io.FileNotFoundException {
    FileInputStream fis = new FileInputStream(file);
    byte[] buf = new byte[100];
    fis.read(buf);
    String ret = new String(buf);
    fis.close();
    return ret;
  }

  public static void writeToVulnerableSink(String str)
    throws java.io.FileNotFoundException {
    FileOutputStream fos = new FileOutputStream(str);
    PrintWriter writer = new PrintWriter(fos);
    writer.write(str);
  }
}
```

このコード例を実行すると、以下のトレース情報が作成されます。



このペインには、入力スタックと出力スタックが表示されています。入力スタックでは、main から getVulnerableSource が呼び出され、そこからさらに FileInputStream.read が呼び出されています。出力スタックでは、main から writeToVulnerableSink が呼び出され、そこからさらに PrintWriter.write が呼び出されています。グラフには、read メソッドから write メソッドまでのデータ・フローと、2 つの呼び出しスタックをつなぐ main が表示されています。データ・フローのセクションには、main メソッドにおいて汚染されたデータが渡される操作の行番号が表示されています。 この例では、どちらのメソッド呼び出しもメソッド内の同じ行 (行番号 15) に記述されています (上のコード例では、これは行番号 7 に変換されています。画面キャプチャーでは、ファイルに 8 行のコメントが含まれています)。

例 2: ソースからシンクへの変更

例 2 は、例 1 のコードを変更したものです。 getVulnerableSource メソッド内で呼び出される検証ルーチンと、writeToVulnerableSink メソッド内で呼び出されるエンコード・ルーチンを追加することにより、例 1 を改善しています。

```
import java.io.*;

public class TestCase_IOT_Instance_Val_Encode {
  public static void main(String[] args) {
    try {
      TestCase_IOT_Instance_Val_Encode testCase = new
        TestCase_IOT_Instance_Val_Encode();
      String file = args[0];
      String source = testCase.getVulnerableSource(file);
      source = testCase.validate(source);
      String encodedStr = testCase.encode(source);
      testCase.writeToVulnerableSink(file, encodedStr);
    } catch (Exception e) {
    }
  }

  public String getVulnerableSource(String file) throws Exception {
    FileInputStream fis = new FileInputStream(file);
    byte[] buf = new byte[100];
    fis.read(buf);
    fis.close();

    String ret = new String(buf);
    return ret;
  }

  public void writeToVulnerableSink(String file, String str)
      throws FileNotFoundException {
    FileOutputStream fos = new FileOutputStream(file);
    PrintWriter writer = new PrintWriter(fos);
    writer.write(str);
  }

  private String validate(String source) throws Exception {
    if (source.length() > 100) {
      throw new Exception("Length too long: " + source.length());
    }
  return source;
  }

  private String encode(String source) {
    return source.trim();
  }
}
```

最初のスキャンにより、例 1 と同じようなスタック・トレースが作成されます。

ナレッジベース・データベース を拡張して検証ルーチンとエンコード・ルーチンを組み込むことにより、検出結果内の不要な情報を削除し、すべての呼び出しグラフについて検証ルーチンとエンコード・ルーチンが呼び出されているかどうかを確認することができます。 例えば、例 1 で java.io.FileInputStream.read(byte[]) :int に対するすべての呼び出しからのデータを指定した場合、スキャンは read からのすべての呼び出しを除去します。これは、read もこの検証ルーチンを呼び出すためです。 また、カスタム検証メソッドを呼び出さなかった read からの呼び出し

の状況は、「確定」セキュリティー検出結果に引き上げられます。これは、コード内の既知の検証メソッドを呼び出さなかった場合、悪質な攻撃を受ける可能性があるためです。

検証ルーチンは、FileInputStream の read メソッドの他のバリエーションを検証することもできます。これらは追加ソースとして指定できます。 さらに、特定のシンク (または特定のプロパティーを持つシンク) のみがこのメソッドによって検証されることを認識することもできます。 例えば、このルーチンを、Technology.IO プロパティーを持つシンク (このサンプル・データを取り込むのに使用される PrintWriter.write シンクなど) に制限できます。

例 2:「トレース」ビューでの検証ルーチンとエンコード・ルーチンの作成

このタスクについて

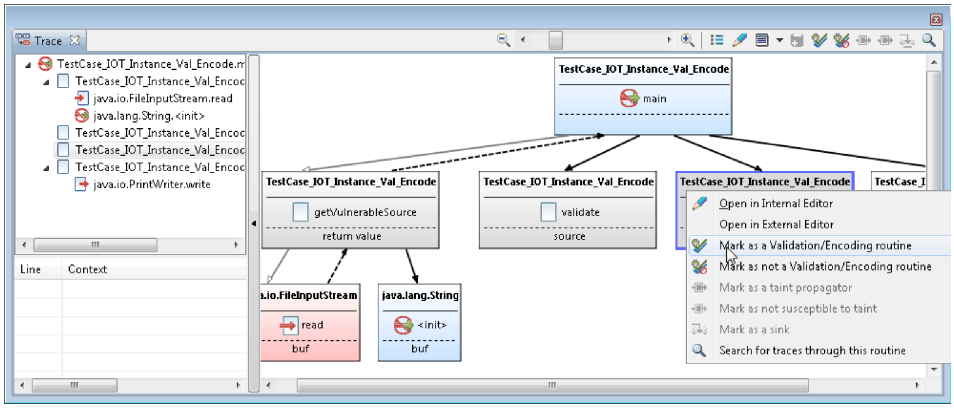
AppScan Source トレース により、汚染されたデータのソースは FileInputStream.read メソッドであることが検出されました。この例では、この検出結果が次回からのスキャンでは検出されないようにするため、検証ルーチンまたはエンコード・ルーチンを作成します。

FileInputStream.read の入力検証ルーチンを作成するには、以下の手順を実行します。

手順

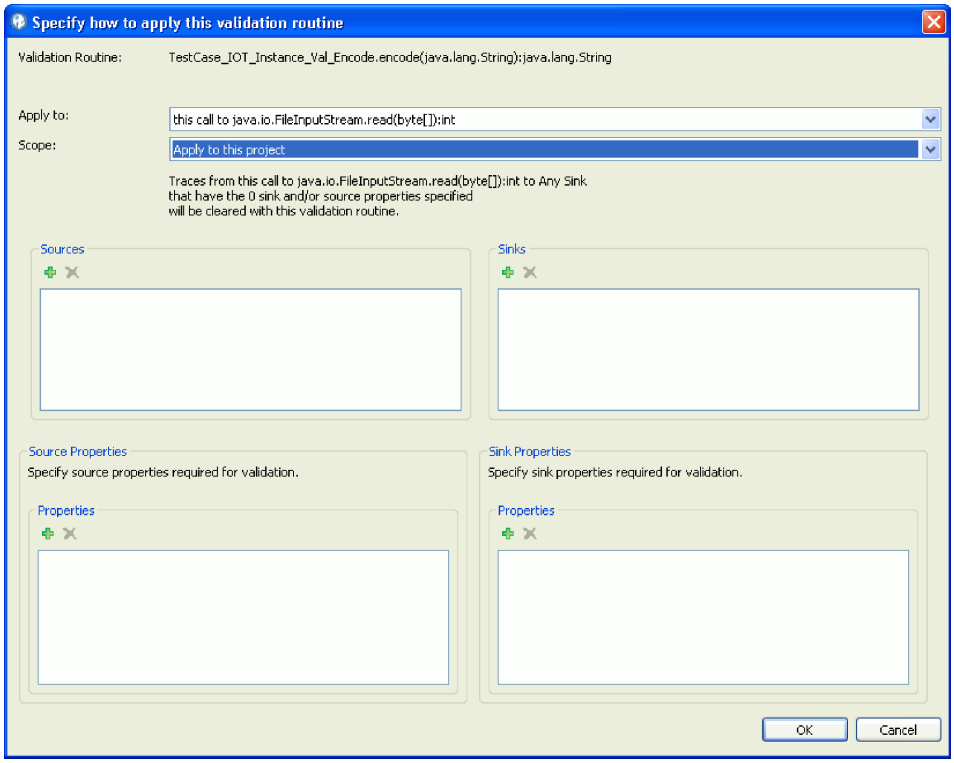
1. 「トレース」ビューの呼び出しグラフで TestCase_IOT_Instance_Val_Encode.encode メソッドを選択して右クリックします。

   ヒント: 作成する検証ルーチンとエンコード・ルーチンがトレース・グラフに表示されない場合は、「トレース」ビューからカスタム・ルール・ウィザードを起動してルーチンを作成できます。 これを実行するのに必要なステップについては、 224 ページの『例 2: カスタム・ルール・ウィザードでの検証ルーチンとエンコード・ルーチンの作成』で説明しています。
2. メニューの「検証/エンコード・ルーチンとしてマークする」を選択します。



3. FileInputStream.read を呼び出すこの特定のインスタンスについてのみ encode ルーチンを適用する場合は、「この検証ルーチンを適用する方法を指定してくだ

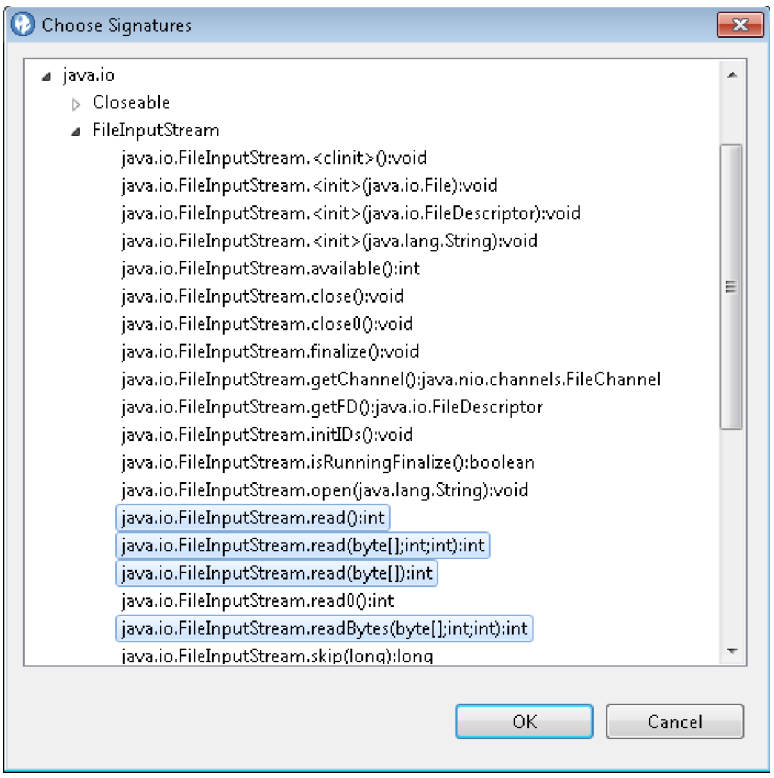
さい」ダイアログ・ボックスで「**java.io.FileInputStream.read** に対するこの呼び出し」を選択します。



`validate` メソッドはクラスに対するプライベート・メソッドで、コードに緊密に関連付けられるため、通常は「**java.io.FileInputStream.read** に対するこの呼び出し」を指定します。

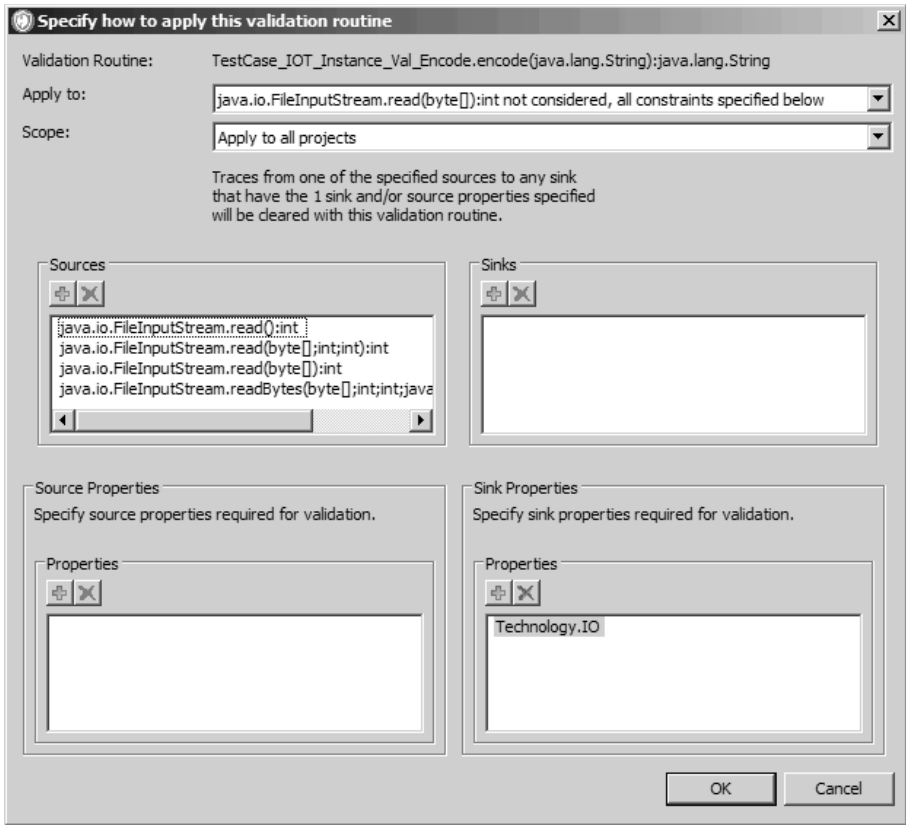
`read` メソッドに対するすべての呼び出しについて検証ルーチンを適用する場合は、「**java.io.FileInputStream.read** に対するすべての呼び出し」を選択します。このオプションを選択する場合、現在のプロジェクトについてのみ有効にするには「このプロジェクトに適用」を選択する必要があります。そうでない場合は、「すべてのプロジェクトに適用」を選択してください。

4. `FileInputStream` クラスのすべての `read` メソッド、および `Technology.IO` プロパティーを持つすべてのシンク (`java.io.PrintWrite.write` メソッドなど) に適用するルーチンをセットアップします。
   a. ソースとしての **read** メソッドの追加: 「**java.io.FileInputStream.read (byte[]):int** に対するすべての呼び出し」を指定して、`java.io.FileInputStream.read(byte[]):int` をソースとして追加することは可能ですが、ここでは代わりに、ソースを個別に追加します。 「この検証ルーチンを適用する方法を指定してください」ダイアログ・ボックスで、「適用対象」メニューの「**java.io.FileInputStream.read(byte[]):int** は考慮されません。すべての制約は下で指定されています」を選択します。 次に、「ソース」セクションの「追加」ボタンをクリックします。「シグニチャーの選択」ダイアログ・ボックスで、`java.io` セクションを展開し、次に `FileInputStream` セクションを展開します。`java.io.FileInputStream.read*` ノードを複数選択して、「**OK**」をクリックします。



b. シンク・プロパティーの追加: 「シンク・プロパティー」セクションの「**VMAT** プロパティーを追加」ボタンをクリックします。「プロパティーの選択」ダイアログ・ボックスで、`Technology.IO` プロパティーを選択して「**OK**」をクリックします

c. すべての設定が完了すると、ダイアログ・ボックスは以下のように表示されているはずです。



5. 「**OK**」をクリックして、検証ルーチンをデータベースに追加します。

例 2: カスタム・ルール・ウィザードでの検証ルーチンとエンコード・ルーチンの作成

作成する検証ルーチンとエンコード・ルーチンがトレース・グラフに表示されない場合は、「トレース」ビューからカスタム・ルール・ウィザードを起動してルーチンを作成できます。

このタスクについて

この例では、 221 ページの『例 2:「トレース」ビューでの検証ルーチンとエンコード・ルーチンの作成』で作成されるものと同じ検証ルーチンを作成します。ただし、この例では、ルーチンを作成するのにカスタム・ルール・ウィザードを使用します。

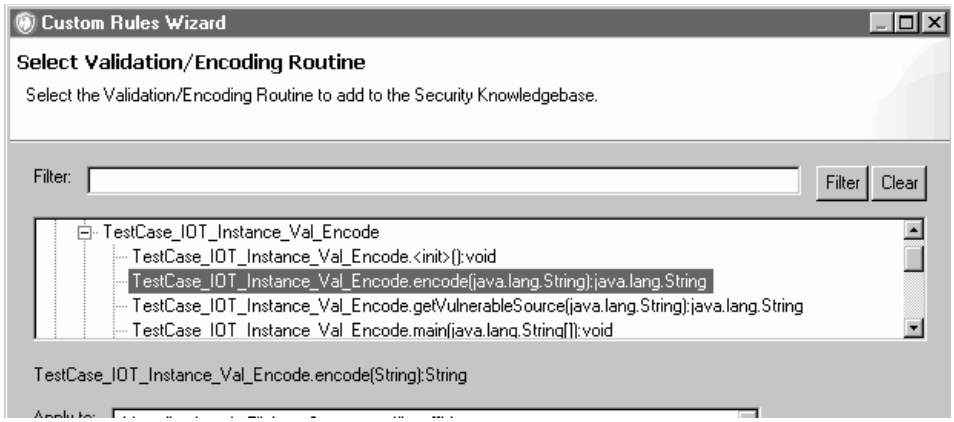
手順

1. 「トレース」ビューで、ツールバーの「カスタム・ルール・ウィザードを起動し、トレース・グラフにない検証ルーチンを追加します」をクリックします。

   注**:** 「カスタム・ルール」ビューからカスタム・ルール・ウィザードを起動した場合、カスタム・ルール・ウィザードから検証ルーチンを作成することはできません。
2. ウィザードの「検証/エンコード・ルーチンの選択」ページで、検証ルーチンの場所を指定します。

この例の場合、次のルーチンを使用します。

```
TestCase_IOT_Instance_Val_Encode.encode(java.lang.String):
java.lang.String
```



3. 221 ページの『例 2:「トレース」ビューでの検証ルーチンとエンコード・ルーチンの作成』の「この検証ルーチンを適用する方法を指定してください」ダイアログ・ボックスで指定したものと同じ設定を使用して、ウィザード・ページの残りのセクションを入力します。
4. 「終了」をクリックして、検証ルーチンをデータベースに追加します。

例 3: ソースとシンクのファイルが異なる場合

以下の例では、シンクとは別のファイル内のソースを示しています。

TestCase_IOT_Xfile_Part1.java:

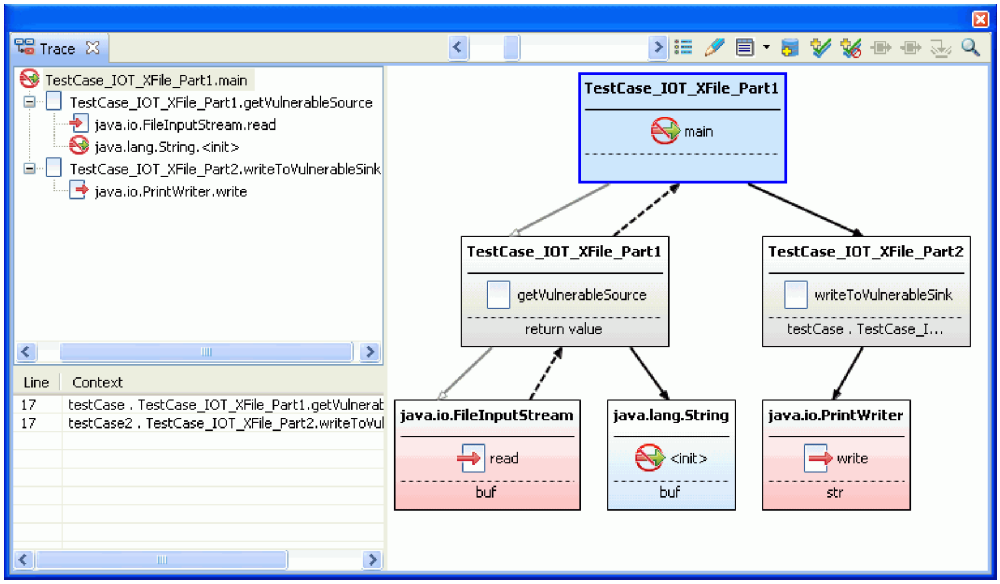
```
public class TestCase_IOT_XFile_Part1 {
 public static void main(String[] args) {
  try {
   TestCase_IOT_XFile_Part1 testCase =
    new TestCase_IOT_XFile_Part1();
   TestCase_IOT_XFile_Part2 testCase2 =
    new TestCase_IOT_XFile_Part2();
   testCase2.writeToVulnerableSink(
    testCase.getVulnerableSource(args[0]));
  } catch (Exception e) {
  }
 }

 public String getVulnerableSource(String file)
  throws IOException, FileNotFoundException {
  FileInputStream fis = new FileInputStream(file);
  byte[] buf = new byte[100];
  fis.read(buf);
  String ret = new String(buf);
  fis.close();
  return ret;
 }
}
```

TestCase_IOT_Xfile_Part2.java:

```
public class TestCase_IOT_XFile_Part2 {
 public void writeToVulnerableSink(String str)
  throws FileNotFoundException {
```

```
  FileOutputStream fos = new FileOutputStream(str);
  PrintWriter writer = new PrintWriter(fos);
  writer.write(str);
 }
}
```

TestCase_IOT_Xfile_Part1.java から TestCase_IOT_Xfile_Part2.java までデータをトレースすることにより、プログラム全体を通じてデータ・フローをトレースすることができます。 スタック・トレースは次のように表示されます。



この例では、TestCase_IOT_XFile_Part1 から TestCase_IOT_XFile_Part2 まで、main メソッド全体のデータ・フローを示しています。

例 4: 詳細な検証

例 4 のコードを調べると、最初のスキャンには、対応するトレース・ルーチンをルートとする 3 つの AppScan Source トレースが記述されていることがわかります。 この例では、FileInputStream.read メソッドを trace1 で選択し、validate ルーチンを追加します。 サンプル・ソース・コードの後のセクションで、検証ルーチンのそれぞれの有効範囲の影響について説明します。

```
public class TestCase_IOT_UserValidation {
    ResultSet resultSet;
    FileInputStream fileInputStream;
    PrintWriter printWriter;
    byte[] buffer;

    public static void main(String[] args) throws Exception {
        TestCase_IOT_UserValidation testCase = new TestCase_IOT_UserValidation();
        testCase.trace1();

        TestCase_IOT_UserValidation testCase2 = new TestCase_IOT_UserValidation();
        testCase2.trace2();

        TestCase_IOT_UserValidation testCase3 = new TestCase_IOT_UserValidation();
        testCase3.trace3();
    }

    private void trace1() throws Exception {
        String source = getVulnerableSource1();
        source = validate(source);
```

```
        writeToVulnerableSink(source);
    }

    private void trace2() throws Exception {
        String source = getVulnerableSource2();
        source = validate(source);
        writeToVulnerableSink(source);
    }

    private void trace3() throws Exception {
        String source = getVulnerableSource3();
        source = validate(source);
        writeToVulnerableSink(source);
    }

    public String getVulnerableSource1() throws Exception {
        fileInputStream.read(buffer);
        return new String(buffer);
    }

    public String getVulnerableSource2() throws Exception {
        fileInputStream.read(buffer);
        return new String(buffer);
    }

    public String getVulnerableSource3() throws Exception {
        return resultSet.getString("x");
    }

    public void writeToVulnerableSink(String str) throws Exception {
        printWriter.write(str);
    }

    private String validate(String source) throws Exception {
        // validate
        return source;
    }
}
```

呼び出しサイト単位の検証ルーチン - FileInputStream.read に対するこの呼び出しの入力

現在の検証ルーチンが非常に限定されたコンテキストしかカバーしていない場合や、入力メソッドが非常に汎用的であるために複数の検証ルーチンが必要になる場合は、呼び出しサイト単位の検証ルーチンを作成します。 trace1 メソッドで「**FileInputStream.read** に対するこの呼び出しに適用」を指定した場合、trace1 の呼び出しスタックには validate メソッドに対する呼び出しが含まれているため、次回のスキャンからは trace1 が検出結果として表示されなくなります。 trace2 も validate を呼び出していますが、検証ルーチンの有効範囲が trace1 呼び出しサイトに関連付けられているため、次回からのスキャンでも引き続き例外として報告されます。 trace3 メソッドも validate を呼び出していますが、ResultSet.getString をソースとして使用しているため、次回からのスキャンでも引き続き例外として報告されます。

API 単位の検証ルーチン - FileInputStream.read に対するあらゆる呼び出しの入力

特定のソースについてのみ検証を適用するには、API 単位の検証ルーチンを作成します。 「**FileInputStream.read** に対するすべての呼び出しに適用」を指定した場合、trace1 メソッドと trace2 メソッドには validate メソッドに対する呼び出し

が記述されているため、次回からのスキャンではいずれのメソッドも検出結果として示されなくなります。 ただし、`trace3` メソッドは `ResultSet.getString` をソースとして使用しているため、同じように `validate` を呼び出しても、引き続き例外として報告されます。

第 7 章 AppScan Source for Analysis および障害追跡

AppScan Source for Analysis は障害追跡システムと統合されているため、確認されたソフトウェアの脆弱性を開発者のデスクトップに直接送信することができます。 障害追跡システムに送信される障害には、バグに関する説明文と、障害と共に送信される検出結果のみが記載されているファイルが含まれています。

AppScan Source for Analysis とさまざまな障害追跡システム (IBM Rational ClearQuest、IBM Rational Team Concert、HP Quality Center、Microsoft Team Foundation Server など) を統合することで、ソフトウェアの脆弱性の障害を追跡できます。

検出結果を障害追跡システムに送信したり、障害を開発者にメールで送信したりするには、障害追跡システムの設定をあらかじめ構成しておくことが必要になる場合があります (110 ページの『設定による障害追跡の有効化』を参照)。

設定による障害追跡の有効化

「障害追跡システム」の設定で、障害追跡システムへの検出結果の送信を有効にし、障害の送信方法を指定することができます。

「障害追跡システム」設定ページの「全般」タブを使用して、AppScan Source における障害追跡システムの統合機能を有効または無効にします。「障害追跡システムの統合を有効にする」チェック・ボックスを選択すると、評価結果に対して「障害の送信」コンテキスト・メニュー・アクションを使用できるようになります。 また、「全般」タブを使用することで、障害の送信時にどの障害追跡システムを利用できるようにするかを個別に制御できます。

サポート対象の障害追跡システムに対して指定可能な設定については、以下のヘルプ・トピックを参照してください。

- 110 ページの『Rational ClearQuest の設定』
- 111 ページの『Quality Center の設定』
- 113 ページの『Rational Team Concert の設定』
- 115 ページの『Team Foundation Server の設定』

Rational ClearQuest の設定

Rational ClearQuest の設定を行うには、必要な Rational ClearQuest の設定が Rational ClearQuest の管理者によって提供されている必要があります。設定は、それぞれの Rational ClearQuest 環境に固有です。

注**:** Rational ClearQuest バージョン 8.0 と統合する場合、Rational ClearQuest スキーマに、**DefectTracking** 事前定義スキーマで使用可能なフィールドが含まれている必要があります。

データベース・セット

1 つ以上の障害データベースの集合。

```
Linux default = Connection Name,
Windows default = Database Set
```

データベース名

障害の送信先データベースの名前。

データベース・ユーザー名

デフォルトの Rational ClearQuest データベース・ユーザー名。

CQPerl 実行可能プログラムの位置

ローカル・コンピューター上の Rational ClearQuest CQPerl 実行可能プログラムの位置。 指定されたデフォルトの位置は、デフォルトの Rational ClearQuest インストール位置にマップされます。

障害レコードのエンティティー

障害オブジェクト用に使用するために Rational ClearQuest インストール済み環境によって構成されたエンティティー (データベース・オブジェクト)。

デフォルトのエンティティーは「**Defect**」です。

レコードの「説明」フィールド

デフォルトの説明は「**Description**」です。

レコードの「ヘッドライン」フィールド

デフォルトのヘッドラインは「**Headline**」です。

検出結果ごとに単一の障害

複数の検出結果を単一の障害として送信するか、複数の障害として送信するかを選択します。 障害を作成するときに送信方式を変更できます。

Quality Center の設定

最初に全般設定で障害追跡システムとして HP Quality Center を有効にしてから、「Quality Center」タブで個別設定を指定する必要があります。

サーバー URL

Quality Center サーバーの URL (`http://<hostname>:<port>/qcbin/` や `https://<hostname>:<port>/qcbin/` など)。

ユーザー名 (オプション)

Quality Center にログインするユーザー名

パスワード (オプション)

ユーザー名を入力した場合は、対応するパスワードを入力してください。

ドメイン

接続先の Quality Center ドメイン。

プロジェクト

接続先の Quality Center プロジェクト

自動ログイン

true の場合、AppScan Source は、検出結果の送信時にログイン情報を要求するプロンプトを表示せず、「設定」で指定されたデフォルトの資格情報を使用してログインします。 false の場合、検出結果を Quality Center に送信するごとにログインする必要があります。

自動送信

true の場合、新規障害を送信するためのダイアログ・ボックスが検出結果の送信時に表示されません。 AppScan Source for Analysis は、「設定」で指定された「デフォルトの障害プロパティー」を使用します。 false の場合、検出結果の送信時に障害情報 (重大度、優先順位、障害タイプ、状態など) の入力を要求するプロンプトが表示されます。

以前に送信した検出結果の再送信

Quality Center に送信された検出結果には、Quality Center 障害情報 (障害 ID、送信ユーザー、および送信日付) のタグが付けられます。 デフォルトの場合、AppScan Source は、同じ検出結果を 2 回以上再送信することはありません。 これにより、複数の検出結果を Quality Center にディスパッチしても、新規の検出結果のみが Quality Center データベース内に入力されます。 選択した場合 (true の場合)、以前に送信した検出結果を Quality Center に再送信できます。

各検出結果を個別のバグとして送信

複数の検出結果を 1 回の操作で送信するときには、すべての検出結果を単一の Quality Center 障害として送信するか、個別の AppScan Source 検出結果ごとに別個の Quality Center 障害として送信するかを選択できます。このチェック・ボックスを選択すると、フラグは true に設定され、個別の検出結果ごとに別個の Quality Center 障害が作成されます。 フラグを false に設定すると、すべての検出結果を一括送信の一部として送信するための単一の Quality Center 障害が作成されます。

バグ概要の自動生成

true の場合、AppScan Source は、Quality Center に送信するための障害の概要を自動的に生成します。 この概要は、障害に含まれる検出結果の数および検出結果のタイプ (`Validation.Required` など) を示します。

false の場合、新規障害の作成時に開くダイアログ・ボックスで障害を送信するときに、「概要」フィールドが表示されて入力できるようになります。

バグ・フィールドの自動ロード

デフォルト設定は true です。 このチェック・ボックスを選択すると、AppScan Source は、Quality Center 内の現在のユーザーおよびグループ設定に基づいて、Quality Center データベースから障害フィールド定義を自動的にロードします。false の場合、 AppScan Source は、新規障害の作成時に開くダイアログ・ボックスに、Quality Center からの障害フィールドを表示しません。

デフォルトの障害プロパティー

さまざまな Quality Center 障害属性のデフォルト値を設定するには、Quality Center の設定タブの「デフォルトの障害プロパティー」をクリックします。デフォルト値は、送信時に「新規障害」ダイアログ・ボックスに事前に取り込まれるか、「自動送信」設定が選択されている場合には Quality Center に自動的に送信されます。

注: 「バグ・フィールドの自動ロード」が選択されている場合、「問題プロパティー」ダイアログ・ボックスが表示されるごとに、障害のプロパティーおよびその使用可能な値が Quality Center から動的に取得されます。したがって、Quality Center データベースに追加された新規フィールドおよび値は、AppScan Source for Analysis 内に自動的に表示されます。「問題プロパティー」ダイアログ・ボックスを開き、Quality Center 情報を取り込むには、サーバー、ログイン、および接続の有効な情報が必要です。

Quality Center の障害フィールドのカスタマイズ

構成ファイルを通じて、「新規障害」ダイアログ・ボックス内のフィールドおよびこれらのフィールドの間の相互作用をカスタマイズできます。カスタマイズのサンプルおよび追加説明が記載されているサンプル構成ファイルは、`<data_dir>¥config¥qc.dts` (`<data_dir>` は、ご使用の AppScan Source プログラム・データの場所です。詳しくは、 360 ページの『インストールとユーザー・データ・ファイルの場所』 を参照してください。) にあります。これらのカスタマイズを使用して、「新規障害」ダイアログ・ボックス内で Quality Center Workflow スクリプト・ロジックを直接モデル化できます。

以下のカスタマイズを選択できます。

- カスタム・フィールドまたは欠落フィールド (あるいはその両方) を表示します
- フィールドを常に強制的に表示します (Quality Center 設定に対し優先します)
- 他のフィールドの選択に基づいて、フィールドに必要な状態を更新します
- 別のフィールド内のリスト・ボックス選択に基づいて、フィールドのリスト・ボックス・オプションを動的に更新します

Rational Team Concert の設定

「Rational Team Concert」設定タブでは、Rational Team Concert サーバーへの接続と、ワークアイテム属性の値を構成できます。

接続情報を入力して正常にログインすると、1 つ以上のプロジェクト・エリアに接続できるようになります。プロジェクト・エリアごとに独自の属性事前設定値を構成できます。

注: Rational Team Concert に接続 (設定を構成するか、または障害を送信することにより接続) すると、SSL 証明書を受け入れるよう指示するプロンプトが出されます。詳しくは、114 ページの『Rational Team Concert の SSL 証明書』を参照してください。

特定のプロジェクト・エリアの属性値を構成するには、そのプロジェクト・エリアを選択して「構成」を選択します。 「構成」ダイアログ・ボックスで、属性値をハードコーディング値に設定したり、選択した検出結果を参照する変数に設定したりすることができます。例えば、属性値で `{Finding.fileName}` を使用すると、障害の送信時に検出結果の実際のソース・コード・ファイル名に置き換えられます。 それらの変数をサポートしている属性値ではコンテンツ・アシスト (`<Ctrl>+<Space>`) が提供されます。 チームではこれらの構成を共有することをお勧めします。それには、「Rational Team Concert」メイン設定ページにある「インポート」ボタンと「エクスポート」ボタンを使用します。

Team Foundation Server の設定

「Team Foundation Server」設定タブでは、Microsoft Team Foundation Server への接続と、ワークアイテム・フィールドの値を構成できます。

接続情報を入力して正常にログインすると、1 つ以上のプロジェクトに接続できるようになります。

注: Team Foundation Server 2010 へのログインを構成するときに、接続先となるチーム・プロジェクトのコレクションを「サーバー URL」に含める必要があります。 例えば、`http://myserver:8080/tfs/`*`DefaultCollection`* のようにします。

プロジェクトごとに独自のフィールド事前設定値を構成できます。

特定のプロジェクトのフィールド値を構成するには、そのプロジェクトを選択して「構成」を選択します。 「構成」ダイアログ・ボックスで、フィールド値をハードコーディング値に設定したり、選択した検出結果を参照する変数に設定したりすることができます。 例えば、フィールド値で `{Finding.fileName}` を使用すると、障害の送信時に検出結果の実際のソース・コード・ファイル名に置き換えられます。 それらの変数をサポートしているフィールドではコンテンツ・アシスト (`<Ctrl>+<Space>`) が提供されます。

チームではこれらの構成を共有することをお勧めします。それには、「Team Foundation Server」メイン設定ページにある「インポート」ボタンと「エクスポート」ボタンを使用します。

HP Quality Center と AppScan Source for Analysis の統合

HP Quality Center を AppScan Source for Analysis と統合するには、Quality Center クライアントをローカル・コンピューターにインストールする必要があります。 Quality Center クライアント・アプリケーションは、ブラウザー・ベースの Quality Center クライアント・インターフェースを使用して Quality Center に初めてログインするときに、ローカル・コンピューターにダウンロードされてインストールされます。

Quality Center 情報の構成

Quality Center は、「Quality Center」タブの「障害追跡システム」設定を使用して構成します。 AppScan Source の検出結果を障害として送信するには、Quality Center を使用可能にして、Quality Center の設定をあらかじめ構成しておく必要があります。 それぞれの設定の説明については、 111 ページの『Quality Center の設定』を参照してください。

注**:** 環境によっては (HP Quality Center バージョン 11 を実行する環境など)、HP Quality Center 統合が機能するように HP ALM Client MSI Generator アドインをインストールする必要があります。

Quality Center への検出結果の送信

検出結果は、AppScan Source for Analysis のいずれかの「検出結果」ビューを使用して Quality Center に送信されます。

手順

1. 表内の検出結果を 1 つ以上選択するか、バンドルを開きます。(バンドルを開く場合は、バンドル内の送信対象の検出結果を選択してください)。
2. 選択項目を右クリックして、メニューから「障害の送信」 > 「**Quality Center** へのディスパッチ」を選択します。
3. Quality Center にログインします。

   現在の設定で自動ログインが指定されている場合、ログイン・ダイアログ・ボックスは表示されません。 この場合、AppScan Source はデフォルトの資格情報を使用してログインを実行します。
4. 検出結果を送信します。

   現在の設定で自動送信が指定されている場合は、AppScan Source の「デフォルトの障害プロパティー」設定を使用して検出結果情報が送信されます。

タスクの結果

検出結果が送信されると、正常に送信された検出結果の数を示す通知メッセージが表示されます。

Quality Center に送信された検出結果の追跡

Quality Center に送信された検出結果には、以下の送信情報タグが付けられます。

- Quality Center 障害 ID

- 送信日
- Quality Center ユーザー名

任意の「検出結果」ビューで、検出結果表の「障害 **ID**」、「障害検出日」、「障害報告ユーザー」の各列に送信情報が表示されます。 ただし、デフォルトの検出結果表には、これらの列は表示されません。 これらの列を表に表示するには、「列の選択と順序付け」ツールバー・ボタンをクリックして、表を構成する必要があります。 「検出結果」ビューに列を追加する方法について詳しくは、 342 ページの『検出結果表のカスタマイズ』を参照してください。

トリアージと修復の実行中に AppScan Source の検出結果の状況を追跡できるようにするため、障害情報は次のスキャンが実行されるまで保存されます。

Quality Center における AppScan Source の検出結果情報

AppScan Source for Analysis によって Quality Center データベース内に障害情報が作成されるときに、検出結果情報が障害の説明として設定されます。 この検出結果情報には、重大度、タイプ、API、分類が格納されます。

また、Quality Center の障害情報データベースには、添付ファイルとして障害情報に追加される AppScan Source バンドル・ファイル (`.ozbdl`) も格納されます。 このバンドル・ファイルには、AppScan Source の検出結果の関連情報 (トレース情報など) がすべて格納されます。 開発者は、バンドルを保存して AppScan Source for Analysis または Developer プラグインで開くことにより、障害のトリアージ処理を行うことができます。

Rational ClearQuest と AppScan Source for Analysis の統合

Rational ClearQuest を AppScan Source for Analysis と統合するには、Rational ClearQuest クライアントをローカル・コンピューターにインストールする必要があります。 このインストールには、`CQPerl` 実行可能ファイルが含まれます。この実行可能ファイルのロケーションは、AppScan Source for Analysis Rational ClearQuest の設定で構成する必要があります。

Rational ClearQuest の統合設定を構成する際に、障害情報データベース・スキーマに関する情報を指定する必要があります。 Rational ClearQuest のエンティティーによって Rational ClearQuest データベース・オブジェクトが参照されるため、障害情報に使用するエンティティーを Rational ClearQuest インストール済み環境で指定する必要があります。

注**:** AppScan Source for Analysis との統合に必要な CQPerl 実行可能ファイルのデフォルトの場所は、Rational ClearQuest のデフォルトのインストール・ディレクトリーです。

注**:** Rational ClearQuest バージョン 8.0 と統合する場合、Rational ClearQuest スキーマに、**DefectTracking** 事前定義スキーマで使用可能なフィールドが含まれている必要があります。

Rational ClearQuest への検出結果の送信

検出結果を自社の障害追跡システムに組み込むことにより、開発者が修復することができます。 ご使用の障害追跡システムに個々の検出結果を送信するか、または 1 つ以上の検出結果をバンドルにして送信することができます。AppScan Source のセッション中に検出結果を AppScan Source から Rational ClearQuest に初めて送信する場合は、ユーザー名とパスワードを入力してログインする必要があります。

バンドルを Rational ClearQuest に送信する場合、バグ番号は、バンドル自体ではなく、バンドル内の個々の検出結果に対して関連付けられます。 これにより、障害の作成時に各検出結果を障害に関連付けたまま、バンドルをさらに操作することができます。

多数の検出結果をバンドルにまとめることができます。すべての検出結果を 1 つの障害にまとめて送信することも、それぞれの検出結果を別々の障害として送信することもできます。 複数の検出結果が存在する場合に「検出結果ごとに単一の障害」設定を選択すると、これらの障害の説明を編集することができます。1 つの障害を送信する場合に編集できるのは説明だけです。

注**:** Rational ClearQuest にログインするには、デフォルトの追跡システム設定をあらかじめ構成しておく必要があります。

注**:** Rational ClearQuest バージョン 8.0 と統合する場合、Rational ClearQuest スキーマに、**DefectTracking** 事前定義スキーマで使用可能なフィールドが含まれている必要があります。

Rational ClearQuest への障害の送信

手順

1. 表内の検出結果を 1 つ以上選択するか、バンドルを開きます。(バンドルを開く場合は、バンドル内の送信対象の検出結果を選択してください)。
2. 選択項目を右クリックして、メニューから「障害の送信」 > 「**ClearQuest** へのディスパッチ」を選択します。
3. Rational ClearQuest にログインして、検出結果を送信します。

タスクの結果

該当のファイルだけが格納された評価ファイルがそれぞれの障害に添付されます。この評価ファイルは、AppScan Source for Analysis または AppScan Source for Development で開くことができます。

注**:** Rational ClearQuest バージョン 8.0 と統合する場合、Rational ClearQuest スキーマに、**DefectTracking** 事前定義スキーマで使用可能なフィールドが含まれている必要があります。

Rational Team Concert と AppScan Source for Analysis の統合

Rational Team Concert と AppScan Source for Analysis を統合する場合、追加の Rational Team Concert クライアントをコンピューターにインストールする必要はありません。

Rational Team Concert への接続を構成するには、「障害追跡システム」設定の「Rational Team Concert」タブに移動します。あるいは、障害を送信すると、その時点でプロンプトが表示されて、ログインして接続を構成するように求められます。

また、障害を送信するときに使用される事前設定フィールド値も、Rational Team Concert の設定で構成できます。 これによって、すべての障害に対して使用する値を設定できます。また、AppScan Source の出荷時に設定されているデフォルト値を変更することもできます。

注**:** Rational Team Concert に接続 (設定を構成するか、または障害を送信することにより接続) すると、SSL 証明書を受け入れるよう指示するプロンプトが出されます。詳しくは、 114 ページの『Rational Team Concert の SSL 証明書』を参照してください。

Rational Team Concert への障害の送信

1 つ以上の検出結果が含まれているバンドルを Rational Team Concert に送信するか、または個々の検出結果を送信することができます。 検出結果を AppScan Source for Analysis から Rational Team Concert に初めて送信する場合には、ユーザー名とパスワードを入力してログインする必要があります。 送信時に使用される事前設定フィールド値は、Rational Team Concert の設定で構成できます。

このタスクについて

バンドルを Rational Team Concert に送信すると、ワークアイテム番号が、バンドル自体ではなく、バンドル内の個々の検出結果に対して関連付けられます。 これにより、各検出結果をワークアイテム番号に関連付けたまま、バンドルをさらに操作することができます。

手順

1. 表内の検出結果を 1 つ以上選択するか、バンドルを開きます (バンドルを開く場合は、バンドル内の送信対象の検出結果を選択してください)。
2. 選択項目を右クリックして、メニューから「障害の送信」 > 「**Rational Team Concert** へのディスパッチ **(Dispatch to Rational Team Concert)**」を選択します。
3. 送信ダイアログ・ボックスに表示される指示に従って、必要に応じてログインし、必須属性の入力などを実行して、プロセスを完了させます。

   注**:** Rational Team Concert に接続 (設定を構成するか、または障害を送信することにより接続) すると、SSL 証明書を受け入れるよう指示するプロンプトが出されます。詳しくは、 114 ページの『Rational Team Concert の SSL 証明書』を参照してください。

タスクの結果

送信されるワークアイテムにバンドルが自動的に追加されます。これは、後で AppScan Source for Analysis または AppScan Source for Development のユーザーが開くことができます。

Rational Team Concert の SSL 証明書

Rational Team Concert サーバーのインストール時に、有効な SSL 証明書を使用するように構成する必要があります。この構成を行わないと、(設定の構成時や障害の送信時に) サーバーにログインするときに、信頼できない接続を通知するメッセージを受け取ります。このトピックでは、Rational Team Concert SSL 証明書の考慮事項について簡単に説明します。

SSL 証明書の保管場所

永続的に受け入れられた証明書は、<user_home>/.jazzcerts (<user_home> は、ご使用のオペレーティング・システムのホーム・ディレクトリーです (例えば Windows では、ディレクトリーは C:¥Documents and Settings¥Administrator¥ などになります)。) に保管されます。 <user_home>/.jazzcerts を削除すると、AppScan Source および Rational Team Concert クライアント用に保管されているすべての証明書が削除されます。

Rational Team Concert クライアントと共有される SSL 証明書

AppScan Source は、その証明書ストアを Rational Team Concert クライアントと共有します。 Rational Team Concert クライアントを使用して証明書を永続的に受け入れると、その証明書は AppScan Source によって再使用されます (AppScan Source で証明書の受け入れを要求するプロンプトが表示されなくなります)。同様に、AppScan Source で証明書を永続的に受け入れると、その証明書が Rational Team Concert クライアントによって再使用されます。

Microsoft Team Foundation Server と AppScan Source for Analysis の統合

Team Foundation Server を AppScan Source for Analysis と統合するには、Microsoft Visual Studio Team Explorer クライアントをローカル・コンピューターにインストールする必要があります。

ご使用の Team Foundation Server への接続を構成するには、「障害追跡システム」設定の「Team Foundation Server」タブに移動します。あるいは、障害を送信すると、その時点でプロンプトが表示されて、ログインして接続を構成するように求められます。

また、障害を送信するときに使用される事前設定フィールド値も、Team Foundation Server の設定で構成できます。これによって、すべての障害に対して使用する値を設定できます。また、AppScan Source の出荷時に設定されているデフォルト値を変更することもできます。

Microsoft Team Foundation Server への障害の送信

1 つ以上の検出結果が含まれているバンドルを Team Foundation Server に送信するか、または個々の検出結果を送信することができます。 検出結果を AppScan Source for Analysis から Team Foundation Server に初めて送信する場合には、ユーザー名とパスワードを入力してログインする必要があります。送信時に使用される事前設定フィールド値は、Team Foundation Server の設定で構成できます。

このタスクについて

バンドルを Team Foundation Server に送信すると、ワークアイテム番号が、バンドル自体ではなく、バンドル内の個々の検出結果に対して関連付けられます。 これにより、各検出結果をワークアイテム番号に関連付けたまま、バンドルをさらに操作することができます。

注**:** Team Foundation Server 2010 へのログインを構成するときに、接続先となるチーム・プロジェクトのコレクションを「サーバー URL」に含める必要があります。 例えば、`http://myserver:8080/tfs/DefaultCollection` のようにします。

手順

1. 表内の検出結果を 1 つ以上選択するか、バンドルを開きます (バンドルを開く場合は、バンドル内の送信対象の検出結果を選択してください)。
2. 選択項目を右クリックして、メニューから「障害の送信」 > 「**Team Foundation Server** へのディスパッチ」を選択します。
3. 送信ダイアログ・ボックスに表示される指示に従って、必要に応じてログインし、必須フィールドの入力などを実行して、プロセスを完了させます。

タスクの結果

送信されるワークアイテムにバンドルが自動的に追加されます。これは、後で AppScan Source for Analysis または AppScan Source for Development のユーザーが開くことができます。

送信された障害の操作

2 件から 3 件を超える検出結果を個別の障害として送信する場合、その送信プロセスはバックグラウンドで実行されるため、トリアージ・プロセスを引き続き実行することができます。障害を送信した後、障害追跡システムから受信した障害 ID が、関連する検出結果に追加され、その検出結果と共に保持されます。 障害追跡システムに送信された障害を操作するには、このトピックで説明する手順を実行します。

手順

1. ご使用の障害追跡システムを開き、該当する障害を見つけます。
2. 添付ファイルを、AppScan Source のバンドル・ファイル (`.ozbdl`) として保存します。 このファイルは、AppScan Source for Analysis で開くことも、バンドルとして AppScan Source for Development で開くこともできます。

バンドルの障害追跡への送信および E メールによる送信

バンドルでの検出結果は、自社で使用している障害追跡システムに送信できます。あるいは、E メールで送信することもできます。バンドルに検出結果を入れると、開発者による修復のため、これらの検出結果をバグとして送信することができます。

手順

1. バンドルを開きます。

2. 「バンドルを障害追跡に送信」ツールバー・ボタンの下矢印をクリックして、障害追跡システムを選択します。

   注**:** 障害追跡システムによっては、「障害追跡システム」設定を変更してからバンドルを送信することが必要な場合があります。

   あるいは、「バンドル」ツールバーの「バンドルを **E** メールで送信」をクリックして、バンドルを他のユーザーに送信します (E メールの設定は事前に構成しておく必要があります)。
3. 開いている構成ダイアログ・ボックスの設定を完了します。これらのダイアログ・ボックスは、選択した障害追跡システムにより異なります。詳しくは、ヘルプの『*AppScan Source for Analysis* および障害追跡』セクションで説明しています。

E メールによる障害の追跡 (E メールによる検出結果の送信)

このタスクについて

E メール設定を構成済みである場合は、検出結果またはバンドルを E メールで開発者に直接送信して、スキャンで検出された潜在的な障害を通知することができます。この E メールには、検出結果が記録された添付ファイルと、検出結果を説明するテキストが含まれています。

注**:** 一部の Simple Mail Transfer Protocol (SMTP) リレーでは、メールが特定のドメインにのみ送信されます。この場合、`mydomain.com` から E メールを送信すると、`mydomain.com` 内の受信者のみが AppScan Source for Analysis を介してこの E メールを受信することができます。

検出結果表の検出結果を E メールで送信するには、以下の手順を実行します。

手順

1. 表内の検出結果を 1 つ以上選択するか、バンドルを開きます。バンドルを開いた場合は、メールで送信する、バンドルされた検出結果を選択します。
2. 選択項目を右クリックして、メニューから「検出結果の **E** メール送信」を選択します。
3. E メールに、検出結果を含むバンドル添付ファイルが添付されます。「添付ファイル名」ダイアログ・ボックスで、検出結果バンドルの名前を指定します。例えば、「添付ファイル名」フィールドで `my_finding` と指定すると、ファイル名が `my_finding.ozbdl` のバンドルが E メールに添付されます。 「**OK**」をクリックすると、「検出結果の E メール送信」ダイアログ・ボックスが開きます。
4. 「検出結果の E メール送信」ダイアログ・ボックスの「宛先」フィールドには、デフォルトとして、E メール設定に指定されている「宛先アドレス」が指定されていますが、これは、E メールの作成時に容易に変更できます。このダイアログ・ボックスで、E メールの内容を確認してから「**OK**」をクリックして E メールを送信します。

タスクの結果

E メールの内容例:

```
1 findings:
Name: JavaAny.test_DataInput
Type: Vulnerability.Validation.Required
Severity: Low
Classification: Suspect
File Name: C:¥TestApps¥java¥JavaAny¥src¥JavaAny.java
Line / Col: 275 / 0
Context: di . java.io.DataInput.readFully ( ba )
Notes: Check into this vulnerability and report back ASAP.
```

ヒント: 「検出結果の詳細」ビューで、個々の検出結果またはバンドルを E メールで送信することができます。バンドル・ツールバーの「バンドルを **E** メールで送信」をクリックして、バンドルを E メールで送信することもできます。

第 8 章 検出結果レポートと監査レポート

セキュリティー・アナリストおよびリスク・マネージャーは、選択された検出結果のレポート、またはソフトウェア・セキュリティーのベスト・プラクティスおよび法的要件へのコンプライアンスを測定するための一連の監査レポートにアクセスできます。このセクションでは、集約された検出結果データのレポートを作成する方法について説明します。

AppScan Source for Analysis は、2 種類のレポート (検出結果レポートと AppScan Source レポート) を生成します。検出結果レポート は、選択された検出結果のレポートです。 *AppScan Source* レポート は、すべての検出結果のカテゴリー・グループに基づいたレポートであり、特定のセキュリティー・ポリシーに合わせて調整されています。 AppScan Source レポートのリストについては 245 ページの『AppScan Source レポート』を参照してください。

レポートは、特定のスキャン中に収集された検出結果についての詳細を提供します。また、すべての AppScan Source レポートは、検出結果に追加された注記およびトレース・データを含むことがあります。 レポートの長さは、レポートに含まれる検出結果の数によって異なります。レポートは、PDF ファイルまたはハイパーテキスト・マークアップ言語 (HTML) ファイルとして生成できます。 HTML レポートは、Web ページと同様に機能し、ボタンまたはリンクをクリックすると目的のセクションにジャンプできます。 その後、Web ブラウザーのブラウズ機能を使用して、情報をナビゲートできます。

レポートには、検出結果に適用されたスキャン時フィルターもリストされます。スキャン時フィルターについては、 182 ページの『適用済みフィルターの判別』で説明しています。

検出結果レポートの作成

このタスクについて

スキャンの後で、識別された脆弱性についてのレポートを生成できます。 複数の検出結果レポートを生成できます。

- 検出結果
- タイプ別の検出結果
- 分類別の検出結果
- ファイル別の検出結果
- API 別の検出結果
- バンドル別の検出結果
- CWE 別の検出結果 (共通脆弱性タイプ一覧)
- DTS アクティビティー

注**:** 検出結果レポートは、検出結果表内の結果に類似したカテゴリー別の詳細な検出結果を示します。 検出結果レポートを生成するとメモリーを大量に消費する可能性 (https://xmlgraphics.apache.org/fop/1.1/running.html#memory に関連) があるため、最大で 1024 MB の追加のシステム・メモリーが必要になる場合があります。大容量アプリケーションのスキャンに関するレポートを生成するときにメモリーの問題が発生した場合、アプリケーションのパーツごとに個別にスキャンするかスキャン構成を変更してから、レポート (複数の場合あり) の生成を再試行することができます。

検出結果レポート内の CWE ID ハイパーリンクをクリックすると、CWE の Web サイト (http://cwe.mitre.org/) にアクセスすることができます。

検出結果レポートを生成するには、以下のようにします。

手順

1. 検出結果が表示されているビューで、レポートに含める検出結果を選択します。検出結果を選択しない場合は、アクティブなビュー内のすべての検出結果から成るレポートが生成されます。

   「ツール」メニューで、「検出結果レポートの生成」をクリックします。 あるいは、検出結果が表示されているビューで、一連の検出結果を右クリックし、メニューの「検出結果レポートの生成」を選択します。
2. 「検出結果レポートの選択」ダイアログ・ボックスで、レポート・タイプを選択します。

   レポートを生成する場合は「終了」をクリックし、「出力先およびスタイル・シートの指定」ページで以下のオプションの設定を指定する場合は「次へ」をクリックします。

   - レポートの出力先と形式を指定することができます。レポートは、HTML 形式、すべての HTML レポート・コンポーネントを格納する ZIP ファイル形式、または PDF 形式で生成することができます (PDF 形式のレポートを表示するには、Adobe Acrobat Reader が必要です)。レポートの出力先と形式を指定しなかった場合 (または、「検出結果レポートの選択」ページで「終了」をクリックした場合)、HTML 形式がデフォルトで選択され、<data_dir>¥reports (<data_dir> は、ご使用の AppScan Source プログラム・データの場所です。詳しくは、 360 ページの『インストールとユーザー・データ・ファイルの場所』 を参照してください。) にレポートが保存されます。

     注**:** 検出結果レポートではなく、カスタム・レポートを PDF 形式で作成する場合、レポートに記録する詳細レベルを以下の中から指定することができます。

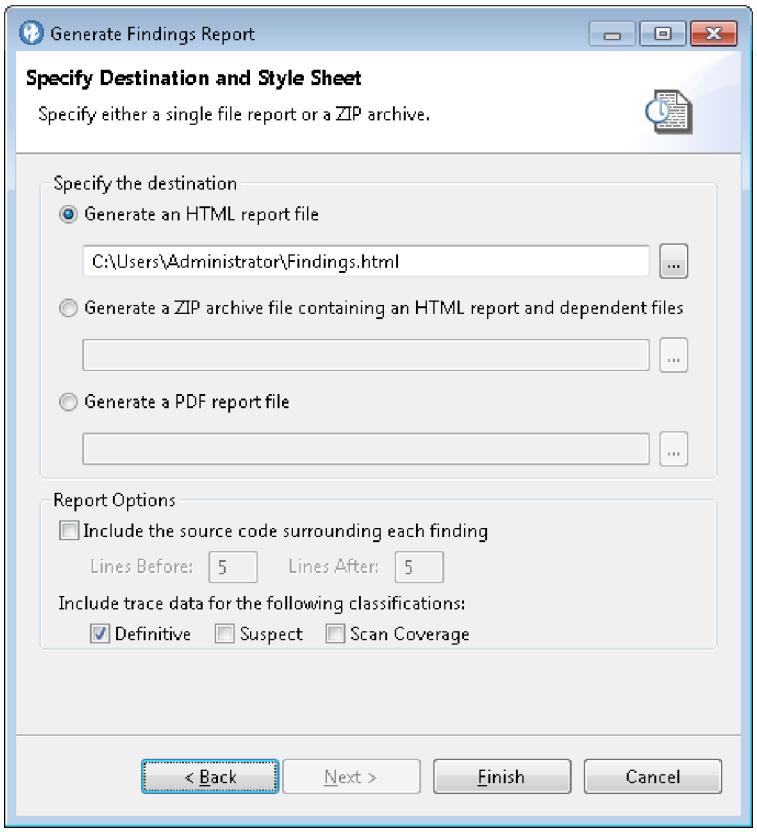
     – 概要: すべてのレポート・グループのカウントが含まれます
     – 詳細: すべての脆弱性プロパティーのすべての API のカウントが含まれます
     – 包括的: すべての API のすべての検出結果から成る表が含まれます
     – 注釈付き: すべての検出結果、および検出結果に含まれるすべての注記、トレース・データ、またはコード・スニペットが含まれます

- コード・スニペットをレポートに含めるには、「各検出結果の周囲のソース・コードを含める」を選択し、レポートに含める脆弱なコード行の前後の行数を指定します。

  ヒント: 「検出結果の詳細」ビューの「レポート作成」セクションで、レポート内の検出結果の前後に組み込むコードの行数を設定することもできます。

  レポートの生成後、注記またはコード・スニペットを含む検出結果を展開すると、ソース・コードが、検出結果の下に青い枠で囲まれて表示されるか、黄色の注記の下に表示されます。脆弱なコード行は、太字の赤いテキストで強調表示されます。
- AppScan Source トレース・データをレポートに含めるには、「以下の分類のトレース・データを含める」の下で、1 つ以上の分類 (「確定」、「要確認」、または「スキャン範囲」) を選択します。

「終了」をクリックして、レポートを生成します。



AppScan Source レポート

AppScan Source レポートは、ソフトウェア・セキュリティー・アナリスト、開発マネージャー、およびリスク管理監査員が、ソフトウェア・セキュリティーのベスト・プラクティスおよび法的要件へのコンプライアンスを測定するために役立ちます。 AppScan Source レポートは、重要なアプリケーションが設定済みのセキュリティー標準を満たしていることを確認するために役立ちます。

AppScan Source では、ソース・コードの脆弱性分析結果を活用して、コンプライアンスについての詳細な見取り図を示す一連のレポートを生成し、セキュリティー、開発、または監査のプロフェッショナルに提供します。

AppScan Source レポートには、以下の機能があります。

- レポート・カード: 各主要カテゴリーのセキュリティー状態の概要を示すレポート・カード
- 詳細監査レビュー: 非適合検出結果の詳細な監査
- ドリルダウン: 非適合コードへの直接アクセス (より詳細な分析および修復と割り当ての優先順位付けのため)

AppScan Source for Analysis は、以下に示すさまざまな AppScan Source レポートを生成します。

- 248 ページの『CWE/SANS Top 25 2011 レポート』
- 248 ページの『DISA Application Security and Development STIG V3R10 レポート』
- 249 ページの『Open Web Application Security Project (OWASP) Mobile Top 10 レポート』
- 249 ページの『Open Web Application Security Project (OWASP) Top 10 2013 レポート』
- 249 ページの『Payment Card Industry Data Security Standard (PCI DSS) バージョン 3.2 レポート』
- 249 ページの『Software Security Profile レポート』: すべての主要な脆弱性カテゴリーにわたって、アプリケーションのセキュリティー状態の概要情報を提供します。

AppScan Source カスタム・レポートの作成

手順

1. 「ツール」メニューで、「レポートの生成」をクリックします。
2. 「レポートの生成」ダイアログ・ボックスで、以下の AppScan Source レポートを選択します。
   - **CWE SANS Top 25 2011**
   - **DISA Application Security and Development STIG V3R10**
   - **OWASP Mobile Top 10**
   - **OWASP Top 10 2013**
   - **PCI Data Security Standard V3.2**
   - **Software Security Profile**

   レポートを生成する場合は「終了」をクリックし、「出力先およびスタイル・シートの指定」ページで以下のオプションの設定を指定する場合は「次へ」をクリックします。
   - レポートの出力先と形式を指定することができます。レポートは、HTML 形式、すべての HTML レポート・コンポーネントを格納する ZIP ファイル形式、または PDF 形式で生成することができます (PDF 形式のレポートを表

示するには、Adobe Acrobat Reader が必要です)。レポートの出力先と形式を指定しなかった場合 (または、「検出結果レポートの選択」ページで「終了」をクリックした場合)、HTML 形式がデフォルトで選択され、<data_dir>¥reports (<data_dir> は、ご使用の AppScan Source プログラム・データの場所です。詳しくは、 360 ページの『インストールとユーザー・データ・ファイルの場所』 を参照してください。) にレポートが保存されます。

注**:** 検出結果レポートではなく、カスタム・レポートを PDF 形式で作成する場合、レポートに記録する詳細レベルを以下の中から指定することができます。

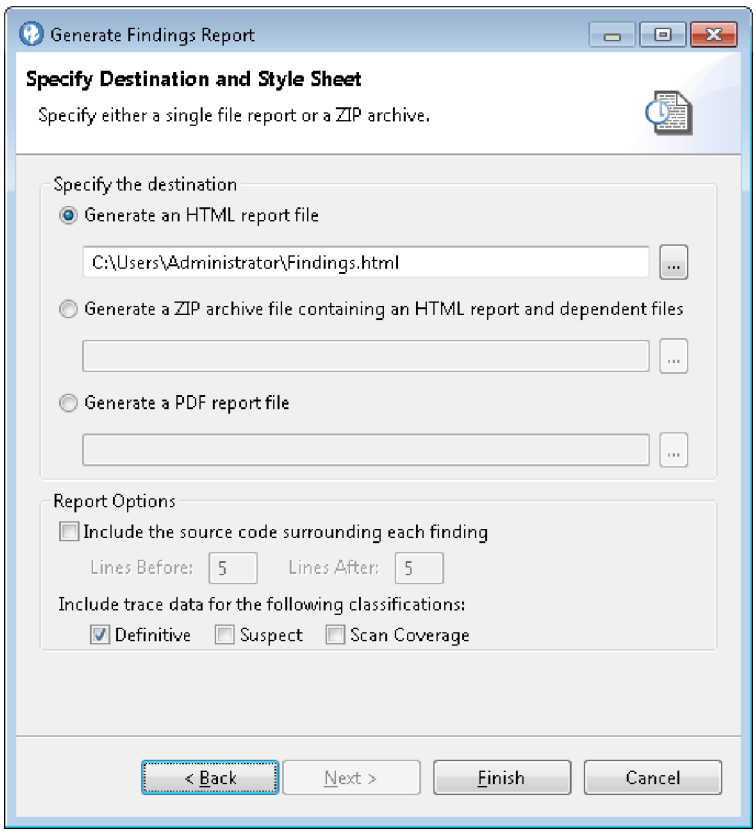
- 概要: すべてのレポート・グループのカウントが含まれます
- 詳細: すべての脆弱性プロパティーのすべての API のカウントが含まれます
- 包括的: すべての API のすべての検出結果から成る表が含まれます
- 注釈付き: すべての検出結果、および検出結果に含まれるすべての注記、トレース・データ、またはコード・スニペットが含まれます

- コード・スニペットをレポートに含めるには、「各検出結果の周囲のソース・コードを含める」を選択し、レポートに含める脆弱なコード行の前後の行数を指定します。

  ヒント**:** 「検出結果の詳細」ビューの「レポート作成」セクションで、レポート内の検出結果の前後に組み込むコードの行数を設定することもできます。

  レポートの生成後、注記またはコード・スニペットを含む検出結果を展開すると、ソース・コードが、検出結果の下に青い枠で囲まれて表示されるか、黄色の注記の下に表示されます。脆弱なコード行は、太字の赤いテキストで強調表示されます。
- AppScan Source トレース・データをレポートに含めるには、「以下の分類のトレース・データを含める」の下で、1 つ以上の分類 (「確定」、「要確認」、または「スキャン範囲」) を選択します。

「終了」をクリックして、レポートを生成します。



CWE/SANS Top 25 2011 レポート

CWE/SANS Top 25 2011 レポートは、「*2011 CWE/SANS* 最も危険なソフトウェア・エラー *TOP 25*」に基づいています。

「*2011 CWE/SANS* 最も危険なソフトウェア・エラー *TOP 25*」について詳しくは、http://cwe.mitre.org/top25/ を参照してください。

AppScan Source にサポートされている共通脆弱性タイプ一覧 (CWE) のすべての脆弱点については、 365 ページの『第 15 章 CWE サポート』を参照してください。

DISA Application Security and Development STIG V3R10 レポート

このトピックでは、「Defense Information Systems Agency (DISA) Application Security and Development Security Technical Implementation Guide (STIG)」Web サイトとガイダンス資料へのリンクを示します。

「DISA Application Security and Development STIG」については、http://iase.disa.mil/を参照してください。

Open Web Application Security Project (OWASP) Top 10 2013 レポート

このトピックでは、「Open Web Application Security Project (OWASP)」Web サイトおよびガイダンス資料へのリンクを示します。

OWASP については、https://www.owasp.org/index.php/Main_Pageを参照してください。さまざまな OWASP 文書およびセキュリティー・リスクへのリンクは、https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project を参照してください。

Open Web Application Security Project (OWASP) Mobile Top 10 レポート

このトピックでは、「Open Web Application Security Project (OWASP)」Web サイトおよびガイダンス資料へのリンクを示します。

OWASP Mobile セキュリティー・プロジェクトについて詳しくは、https://www.owasp.org/index.php/OWASP_Mobile_Security_Projectを参照してください。

Payment Card Industry Data Security Standard (PCI DSS) バージョン 3.2 レポート

このレポートは、クレジット・カード業界データ・セキュリティー基準 (PCI DSS) への準拠を保証するために必要な関連データを提供します。

詳しくは、https://www.pcisecuritystandards.org/security_standards/index.phpを参照してください。

Software Security Profile レポート

Software Security Profile は、アプリケーションのセキュリティーに直接関連する特性について、包括的な分析を提供します。 特定のプロジェクトのソフトウェア内の、重要なセキュリティー機能の詳細な監査を提供します。 このレポートは、デプロイメント用のソフトウェアを認定する前に、暗号化、アクセス制御、ロギング、エラー処理などの要件が実装されていることを確認するために役立ちます。

この複合レポートは、リスクの可能性がある領域を識別し、それらのリスクを最小化するための推奨事項を提案します。 このレポートにより、全体的なアプリケーション・セキュリティーの評価が容易になります。これは、コンプライアンス、ポリシー、アーキテクチャーのレビューに役立ちます。検出結果は、ソース・コードに対する、障害、脆弱性、業界固有の標準、および一般的なベスト・プラクティスのデータベースを使用した、広範囲にわたる静的分析に基づいています。

Software Security Profile には、以下の情報が表示されます。

- レポート・カード: レポート詳細およびセクションの概要を示す重大度インディケーターへのリンクが含まれます。
- 概要: レポートの目的を要約し、アプリケーション構成を記述します。

- メトリック: プロジェクト内のすべてのパッケージのパッケージ、クラス、メソッド、およびコード行の総数を示します。
- カテゴリー別の詳細な検出結果: 検出された各脆弱性カテゴリーの脆弱性カテゴリー名、および脆弱性の重大度レベルを示すアイコンを表示します。

第 9 章 カスタム・レポートの作成

レポート・エディター内で、カスタム・レポートの生成に使用するレポート・テンプレートを作成します。

AppScan Source 検出結果レポートまたは AppScan Source レポートでは、必要とされる的確なデータが提供されないことがあります。その場合には、レポートに含まれる情報を増やしたり減らしたりする必要があります。 AppScan Source for Analysis レポート・エディターを使用すると、カスタム・レポートを作成できます。

通常は、以下の要件がある場合にカスタム・レポートを作成します。

- 固有のセキュリティー・ポリシーにマップされたレポートを、そのポリシーに基づいて生成する場合。 最初にカスタム・レポートを作成し、次にそのレポートを特定の評価に適用します。
- 固有の検出結果および特性を強調するレポートを定義し、生成する場合。
- 既存のレポートを変更する場合、または既存のレポートに要素を追加する場合。

レポート・テンプレートを <data_dir>¥reports (<data_dir> は、ご使用の AppScan Source プログラム・データの場所です。詳しくは、 360 ページの『インストールとユーザー・データ・ファイルの場所』 を参照してください。) に保存すると、すべてのアプリケーションの評価でそのレポートを使用できます。特定のアプリケーションのディレクトリーに保存すると、そのアプリケーションまたはそのアプリケーションのすべてのプロジェクトのスキャンでそのレポートを使用できます。

AppScan Source レポートの作成または編集を開始する前に、レポート・タイプおよび各レポートの構成要素をよく理解してください。 カスタム・レポートを作成するときには、レポート要素を任意の順序でマップできます。 レポート要素には、検出結果情報、コード・スニペット、トレース、修復コンテンツ、およびテキスト要素とグラフィカル要素が含まれます。

レポート・エディター

レポート・エディターを使用して、カスタム・レポートまたはテンプレートを編集したり、新規レポートを作成したりできます。 カスタム・レポートには、検出結果情報、コード・スニペット、AppScan Source トレース、修復コンテンツ、脆弱性マトリックスなど、検出結果レポートで使用可能なすべての項目が含まれます。新規レポートの設計を開始する前に、レポート・エディター内で既存のレポート・テンプレートを変更してみることで、レポート作成プロセスをよく理解することをお勧めします。

レポート・エディターは、「レポート・レイアウト」、「カテゴリー」、および「プレビュー」タブから成り立っています。

- レポート・レイアウト: レポートの外観を設計します。 レイアウトでは、AppScan Source レポート要素を追加、削除、および再配列します。

- カテゴリー: カテゴリーを作成および編集します。カテゴリー は検出結果のグループです。 カテゴリーは、レポートに含める検出結果、それらの検出結果をグループ化する方法、およびグループ化の順序を特定します。
- プレビュー: 編集時に、現在の評価のレポートを表示します。

これら 3 つのタブには、以下の共通フィールドが含まれています。

- ファイル: 保存済みのグループ化ファイル (読み取り専用) のパス。ファイルが保存されるまで、このフィールドには何も表示されません。グループ化ファイルを保存すると、このファイルは、レポートを定義する XML ファイルになります。
- 名前: ユーザー定義のレポート名。

以下のツールバー・ボタンを使用して、カスタム・レポートを保存する、開く、作成する、コピーする、および生成する操作を実行します。

- 新規レポートの作成: 新規カスタム・レポートを作成します。
- 既存からの新規レポート: 既存のレポート・テンプレートから新規カスタム・レポートを作成します。
- 保存済みのレポートを開く: 編集するグループ化ファイルを開きます。
- 保存: 現在のレポートを指定されたファイルに保存します。
- 名前を付けて保存: 現在のレポートを新規ファイルに保存します。
- このレポートのインスタンスの生成: 現在開いている評価のレポートのコピーを作成します。

ヒント: 既存のレポートのサンプルを表示するには、「既存からの新規レポート」をクリックし、AppScan Source のいずれかのレポート・テンプレートを選択します。これらのテンプレートの「レポート・レイアウト」タブと「カテゴリー」タブを使用すると、各レポートの大まかな設計内容を確認することができます。

「レポート・レイアウト」タブ

「レポート・レイアウト」タブは、「パレット」セクションと「レイアウト」セクション、および各ページに表示されるヘッダーまたはフッターを指定できるセクションから構成されています。

ページ・ヘッダーとページ・フッター

「ページ・ヘッダー」フィールドを使用すると、レポートの各ページの上部に表示されるテキストを指定することができ、「ページ・フッター」フィールドを使用すると、レポートの各ページの下部に表示されるテキストを指定することができます。

パレット

「パレット」には、AppScan Source 標準レポートの構成要素のリストが表示されます。 一部の要素については、「カテゴリー」タブで定義されたカテゴリーの情報だけが表示されます (253 ページの表 19 を参照)。

表 *18.* レポート・レイアウト・パレット - カテゴリーに依存しない要素

レポート要素	説明
テキスト・ヘッダー	テキストの太字ブロックをレポート・レイアウトに追加します。
イメージ・ヘッダー	指定のサイズに拡大または縮小されたイメージをピクセル単位で表示します。
AppScan Source ヘッダー	AppScan Source の商標表示を含むレポート・ヘッダー。
タイトルと日付	スキャンされた項目名を含むレポートのタイトルと、スキャン日付およびレポートの生成日付。
テキスト・ブロック	任意のユーザー定義のテキスト。 「ラベル」フィールドで、テキスト・ブロックの見出しを追加することもできます。
脆弱性マトリックス	評価の脆弱性マトリックス (「脆弱性マトリックス」ビューに表示されるものと同じグラフを表示します)。
メトリック	プロジェクト内のすべてのパッケージのパッケージ、クラス、メソッド、およびコード行の総数を示します。
スキャン履歴	現在のスキャンのメトリックと、同じターゲットのスキャンの履歴メトリック。

表 *19.* レポート・レイアウト・パレット - カテゴリーに依存する要素

レポート要素	説明
レポート・カード	「カテゴリー」タブで定義された各カテゴリーの脆弱性レベルの簡単な明細。レポート詳細およびセクションの概要を示す重大度インディケーターへのリンクが含まれます。
脆弱性明細	「カテゴリー」タブで定義されたすべてのカテゴリーにおける脆弱性の数の明細を持つ表 (重大度および分類別)。
部分的レポート・カード	「カテゴリー」タブでの定義による、ユーザー指定カテゴリーの脆弱性レベルの明細。
カテゴリー	「カテゴリー」タブでの定義に従い、カテゴリー化されたすべての検出結果データをリスト表示します。
カテゴリー	「カテゴリー」タブで定義された 1 つ以上のカテゴリーにおけるすべての検出結果をリスト表示します。

レイアウト

パレットから追加した項目は「レイアウト」に表示されます。レイアウト内の項目の削除、変更、移動を行うには、セクション・ツールバーを使用します。

「カテゴリー」タブ

「カテゴリー」タブを使用すると、選択したバンドル、プロパティー、または検出結果に基づいて、検出結果を含むカテゴリーを追加することができます。追加したカテゴリーは、特定の項目を「レイアウト」に追加する際に使用することができます。例えば、「脆弱性明細」を「レイアウト」に追加すると、すべてのカテゴリーにおける脆弱性の数の明細を持つ表 (重大度および分類別) がレイアウトに追加されます。「カテゴリー」タブは、カテゴリーのツリーが表示されるペインと、選択したカテゴリーの属性を編集するためのペインから成り立っています。各カテゴリーには、定義された特定の要件を満たす評価内の検出結果が含まれています。

以下のカテゴリーを使用できます。

- バンドル: バンドル・カテゴリーは、複数のバンドル名のリストから成り立っています。このリストに名前が出現するバンドルに含まれるすべての検出結果が、このカテゴリーに表示されます。 バンドルは現在の評価から選択しますが、バンドルは名前によってマッチングされるため、バンドル・カテゴリーは任意の評価に適用できます。
- 個別の検出結果: カテゴリーに追加する特定の検出結果を選択します。検出結果のスナップショットのみがレポートに追加されます。 レポートに追加された後で検出結果を変更した場合、レポートはその変更を反映しません。
- 脆弱性タイプ、メカニズム、およびテクノロジーのプロパティー: AppScan Source セキュリティー・ナレッジ・データベース 内のプロパティーおよび API からの必須プロパティーのセットを選択します。検出結果に少なくとも 1 つの「プロパティー」およびすべての「必須プロパティー」が含まれる場合、その検出結果はレポートに組み込まれます。

以下の表は、カテゴリー・ペインと、各ペインを構成する項目を示しています。

表 *20.* 「カテゴリー」タブ属性

属性	説明	編集方法
ラベル	カテゴリーの簡単な名前 (Buffer Overflow など)。 このラベルにより、カテゴリーのツリー・リスト内のカテゴリーが識別されます。また、このラベルは、カスタム・レポート内のカテゴリー見出しになります。	1 行のテキスト・フィールドにラベルを入力します。
概要	このカテゴリー内でレポートされる検出結果の数を示す文のテンプレート。 レポート生成中に、実際のカウントが `%FindingCount%` を置き換えます。	カテゴリーの簡略説明を入力し、「カウントの追加」をクリックして、カーソル位置にある語句内に変数 `%FindingCount%` を配置します。
テキスト	カテゴリーの簡単な説明。	カテゴリーを説明するテキストを入力します。

表 20. 「カテゴリー」タブ属性 (続き)

属性	説明	編集方法
プロパティー (プロパティー・カテゴリーのみ)	これらのプロパティーのうち少なくとも 1 つを含む検出結果が、このカテゴリー内でレポートされます。 リストされたすべての必須プロパティーが検出結果に含まれない場合、その検出結果はこのカテゴリーに含まれません。	ツールバーの「追加」をクリックし、「プロパティーの追加」ダイアログ・ボックスからプロパティーを選択します。 「削除」をクリックして、選択した項目をリストから削除します。
必須プロパティー (プロパティー・カテゴリーのみ)	すべての必須プロパティーおよび少なくとも 1 つのプロパティーを含む検出結果が、このカテゴリーのレポート内に表示されます。	ツールバーの「追加」をクリックし、「プロパティーの追加」ダイアログ・ボックスからプロパティーを選択します。 「削除」をクリックして、選択した項目をリストから削除します。
バンドル (バンドル・カテゴリーのみ)	このカテゴリーに含めるバンドルの名前を指定します。	「バンドル」セクションの「バンドルの追加」をクリックし、リストからバンドルを選択します。
検出結果 (検出結果カテゴリーのみ)	このカテゴリーに含める検出結果を指定します。	任意の検出結果表で検出結果を選択し、その表のツールバーで「検出結果の追加」をクリックして、選択した検出結果を追加します。選択した検出結果が複数のビューに存在する場合、どのビューの検出結果を追加するかを選択するためのプロンプトが表示されます。 検出結果を、検出結果表から「レポート・エディター」ビューの表にドラッグすることも、レポート・エディターにドラッグすることも、カテゴリー・ツリー内の既存の検出結果カテゴリーに直接ドラッグすることもできます。

「プレビュー」タブ

テンプレートを編集しながら、AppScan Source for Analysis レポートをプレビューできます。 「プレビュー」ペインで、「プレビュー」をクリックして、開いている評価のレポートを表示します。

カスタム・レポートの生成

このセクションの各トピックでは、既存のカスタム・レポートを使用してレポートの設計と生成を行う手順について説明します。あるいは、新規レポートを作成することもできます。 既存のレポートを編集するには、レポートを開いてから、手順に従って設計、変更、およびプレビューします。

- 『既存のカスタム・レポートからのレポートの設計』
- 『レポートへのカテゴリーの組み込み』
  - 257 ページの『カテゴリーへのバンドルの追加』
  - 257 ページの『カテゴリーへの検出結果の追加』
  - 257 ページの『カテゴリーへのプロパティーの追加』
- 258 ページの『レポートのプレビュー』
- 258 ページの『レポート・テンプレートの保存』

既存のカスタム・レポートからのレポートの設計

手順

1. 「レポート・エディター」ビューで、ツールバーの「既存からの新規レポート」をクリックします。
2. 既存のレポートのリストからレポート・テンプレートを選択します。「レイアウト」ペインで、レポート・テンプレートをプレビューします。
3. レポート名、ヘッダーとフッター、またはテンプレート要素を変更します。
   a. ページ・ヘッダーまたはページ・フッターを追加します。 ページ・ヘッダーおよびページ・フッターは、各ページ上に表示されます。
   b. 追加要素をレポートに追加します。 必要なレポート要素を「パレット」から選択して「挿入」をクリックします (各要素は個別に挿入する必要があります)。
   c. 要素をレポートから削除します。 テンプレートから削除する要素を選択し、ツールバーの「選択したレポート要素の削除」ボタンをクリックします。
4. レポート要素を再配列します。 プレビュー内で要素を選択し、ツールバーの「選択したレポート要素を上へ移動」または「選択したレポート要素を下へ移動」をクリックして、レポート要素を上または下へ移動します。
5. 要素を編集するには、「レイアウト」ペイン内でその要素をダブルクリックするか、その要素を選択してからツールバーの「選択したレポート要素の編集」をクリックします。

   結果のダイアログ・ボックスで、必要な変更を行います。例えば、テキスト・ブロックを編集するには、「テキスト・ブロックの編集」ダイアログ・ボックス内でラベルおよび説明テキストを変更します。

   注: 一部の要素については変更できません。

レポートへのカテゴリーの組み込み

レイアウトを定義したら、次はレポートに組み込むカテゴリーを決定します。

手順

1. 「カテゴリー」ペインで、「新規プロパティー・カテゴリーの作成」、「新規バンドル・カテゴリーの作成」、または「新規検出結果カテゴリーの作成」をクリックします。
2. カテゴリーのラベルを入力して、カテゴリーに名前を付けます。また、カテゴリーの簡単な概要 (カウントを含めることができます) と説明テキストも入力します。

   ツールバーの矢印ボタンを使用して、カテゴリーまたはサブカテゴリーを昇格または降格します。
3. バンドル、検出結果、またはプロパティーをカテゴリーに追加します。

カテゴリーへのバンドルの追加

手順

1. バンドルを含む評価を開きます。 評価に事前にバンドルが含まれていない場合、レポートにバンドルを追加することはできません。
2. 「バンドル」ペインで「バンドルの追加」をクリックし、カテゴリーに組み込むバンドルを 1 つ以上指定します。

カテゴリーへの検出結果の追加

手順

1. 追加する検出結果が含まれている「検出結果」ビューを開きます。 目的の検出結果を選択して、検出結果の表にドラッグするか、レポート・エディターのカテゴリー・ツリーのノードにドラッグします。
2. あるいは、検出結果表の上にあるツールバーの「検出結果の追加」をクリックして、他のビュー内で選択した検出結果を組み込みます。 複数のビュー内で検出結果を選択した場合は、カテゴリーに追加する検出結果を含むビューを選択する必要があります。
3. 検出結果表を含む任意のビューから検出結果を選択します。

カテゴリーへのプロパティーの追加

手順

1. 「プロパティーの追加」をクリックします (プロパティーには、脆弱性、メカニズム、テクノロジーが含まれます)。プロパティーを選択すると、そのプロパティーに関する ナレッジベース・データベース の説明が表示されます (使用可能な場合)。
2. 少なくとも 1 つのプロパティーおよびすべての必須プロパティーを選択します。検出結果をカテゴリーに組み込むには、その検出結果に「必須プロパティー」リスト内のすべてのプロパティーが含まれている必要があります。

   サブカテゴリーを作成するには、カテゴリーを選択し、ツールバーの左矢印または右矢印ボタンをクリックします。

レポートのプレビュー

カスタム・レポートを設計するときには、最終レポートを生成する前に、レポートをプレビューできます。 「プレビュー」ペインで、「プレビュー」をクリックして、現在開いている評価のレポートを表示します。

レポート・テンプレートの保存

「レポート・エディター」ビューのツールバーで、「保存」をクリックして現在のレポート・テンプレートを保持したり、「名前を付けて保存」をクリックして現在のレポート・テンプレートを新規ファイルに保存したりできます。

レポート・テンプレートをアプリケーション・ファイル (.paf または .gaf) と同じディレクトリーに保存した場合、そのテンプレートは、カスタム・レポート・ウィザードおよび「レポート・エディター」ビューのオプションのリストで選択可能になり、そのアプリケーションの後続のスキャンで使用できます。 レポート・テンプレートを <data_dir>¥reports (<data_dir> は、ご使用の AppScan Source プログラム・データの場所です。詳しくは、 360 ページの『インストールとユーザー・データ・ファイルの場所』 を参照してください。) に保存した場合、そのテンプレートは、すべてのアプリケーションのスキャンで使用できます。

第 10 章 脆弱性データベースとパターン・ルールのカスタマイズ

このセクションでは、脆弱性データベースをカスタマイズし、カスタマイズした脆弱性やその他のルーチンをスキャンに組み込む方法について説明します。

スキャン・プロセスには、以下のような複数の段階があります。

- 言語固有のスキャン: 脆弱性データベース (AppScan Source セキュリティー・ナレッジ・データベース) を使用して実行されます。
- トレース: 脆弱性データベースを使用して実行されます。
- パターン・ベースのスキャン: グローバル・パターン・ルール・ライブラリーのパターン・ルールを使用して実行されます。

カスタム・ルールを使用して、AppScan Source セキュリティー・ナレッジ・データベースを自社独自のセキュリティー標準に合わせて調整し、これらの標準を社内全体で一貫して適用することができます。 また、パターン・ルールをカスタマイズすることもできます。

AppScan Source セキュリティー・ナレッジ・データベース の拡張

このセクションでは、脆弱性データベースをカスタマイズし、カスタマイズした脆弱性やその他のルーチンをスキャンに組み込む方法について説明します。カスタム・ルールを使用して、AppScan Source セキュリティー・ナレッジ・データベース (脆弱性データベース) を自社独自のセキュリティー標準に合わせて調整し、これらの標準を社内全体で一貫して適用することができます。

独自の検証ルーチンやエンコード・ルーチンを指定したり、特定のアプリケーション・プログラミング・インターフェース (API) を脆弱性、シンクとソース、汚染伝播元、または情報項目として定義することが重要になる場合がよくあります。これらのルールを作成する場合は、AppScan Source の脆弱性データベースをカスタマイズして拡張します。このデータベースは、AppScan Source セキュリティー・ナレッジ・データベースにおける不可欠な部分です。 カスタム・ルールをデータベースに追加すると、AppScan Source for Analysis によってスキャン時に認識されます。カスタム API の呼び出しは、セキュリティー検出結果またはスキャン範囲検出結果として検出され、これらの検出結果が報告されます。

例えば、`BufferOverflow` 型の `readBuffer()` という名前の API をアナリストが追加するものとします。 この場合、これ以降のスキャンでは、指定に一致する脆弱性が AppScan Source for Analysis によって検出されると、この新しい API が報告されるようになります。 脆弱性タイプについて詳しくは、AppScan Source セキュリティー・ナレッジ・データベースを参照してください (メイン・ワークベンチ・メニューで、「ヘルプ」 > 「セキュリティー・ナレッジ・データベース」と選択します)。

カスタムの検証ルーチンとエンコード・ルーチンを追加すると、これらのルーチンでやり取りされるデータは、AppScan Source for Analysis によって脆弱性として処理されなくなります。カスタム・ルーチンを ナレッジベース・データベース に

追加することにより、検証やエンコードを実行することなく、汚染された入力ソースから出力にデータがフローしているかどうかが AppScan Source for Analysis によって判別されるようになります。

注: AppScan Source セキュリティー・ナレッジ・データベース では、カスタム・レコードに関するオンライン・ヘルプは提供されていませんが、脆弱性タイプに関するヘルプ情報は表示されます。

重要: AppScan Source セキュリティー・ナレッジ・データベース を変更するには、**ナレッジベース・データベース の管理**権限が必要です。

カスタム・ルールの作成

「カスタム・ルール」ビューで、カスタム・ルール・ウィザードを開くことができます。カスタム・ルール・ウィザードは、カスタム・データベース・レコードの作成手順を説明するためのツールです。 作成したカスタム・ルールは、「カスタム・ルール」ビューに表示されます。 このビューの表には、シグニチャー、言語、目的が表示されます。

プロジェクト固有の検証ルーチンとエンコード・ルーチンが「カスタム・ルール」ビューに表示されるのは、「エクスプローラー」ビューの「すべてのアプリケーション」の下のアプリケーション内に、ルールが適用されるプロジェクトが存在する場合のみです。

- シグニチャー: シグニチャーは、完全修飾関数名です。 例えば、Java シグニチャーには、`com.test.vulnerable.VulnClass.vulnerable(java.lang.string;int):int.` というように、引数と戻りの型が含まれます。
- 言語: C/C++、Java、Visual Basic、Classic ASP、または .NET
- 目的: 特定のメソッドの 1 つ以上のカスタム・レコード・タイプ (`Validation.EncodingRequired` ルーチン、シンク、ソースなど)。

ヒント: ソース・コードを変更することなく、スキャンを繰り返し実行してカスタム・ルールを追加してから再スキャンするという方法でコード・ベースの評価を調整する場合は、脆弱性分析キャッシュを使用するようにプロジェクト・プロパティーを設定することにより、スキャンの時間を大幅に削減することができます。 これを行うには、プロジェクト・プロパティーで、「脆弱性分析キャッシュを有効にする」チェック・ボックスを選択します。 プロジェクト・プロパティーの設定方法については、 281 ページの『選択したプロジェクトの「概要」タブ』の使用方法に関する説明を参照してください。

カスタム・ルール・ウィザードの使用

カスタム・ルール・ウィザードは、メソッドを AppScan Source セキュリティー・ナレッジ・データベース に追加するときに便利です。多くのカスタム・ルールは、グローバルに適用されます (つまり、すべてのプロジェクトに対して適用されます)。カスタム・トレースなしの検出結果、ソース、シンク、汚染伝播元の適用範囲も、常にグローバルになります。 カスタムの検証/エンコード・ルーチンはグローバルではありません。

注: カスタム・ルール・ウィザードでは、選択された項目の妥当性は検証されません。 例えば、あるメソッドを汚染伝播元かつシンクとして識別するカスタム・ルールを定義することもできますが、これは有効なシナリオではありません。

カスタム・ルール・ウィザードの説明に従うことにより、以下の項目を定義して ナレッジベース・データベースに追加することができます。

- シンクとソース
- 汚染伝播元
- 汚染の可能性がないアプリケーション・プログラミング・インターフェース (API)
- 脆弱性
- トレースなしの検出結果を生成する API
- 検証/エンコード・ルーチンではない API
- 汚染されたコールバック
- 情報項目

ソース (汚染)

形式が正しくない可能性のある入力データや、悪質である可能性のある入力データをプログラムに提供するメソッド。

シンク (汚染の可能性あり)

悪質な入力データの影響を受ける可能性のあるファイル、ネットワーク、データベース、他のライブラリー、またはデバイスに対して、データをプログラムから (あるいは、プログラムの可視部分から) 送信する API。

汚染伝播元

汚染伝播元としてマークされたメソッドは、検証されていない入力データ (汚染されたデータ) から API に対する引数を取得して呼び出しを行った場合に、戻り値だけではなく、他の引数によって参照される非定数データも汚染されている可能性があることを示しています。 こうしたデータは、シンクに送信する前に検証またはエンコードを行う必要があります。 このような状況は、通常、汚染された引数からのデータが返された場合や、他の引数にコピーまたは追加された場合に発生します。

汚染の可能性なし

「汚染の可能性なし」(汚染伝播元ではない) としてマークされた API は、検証されていない入力データ (汚染されたデータ) から取得された引数を使用して API を呼び出しても、安全ではない動作や悪質な動作がこの API によって実行されることはないことを示しています。

汚染されたデータが呼び出し処理に到達し、その呼び出し処理が「汚染の可能性なし」としてマークされている場合、トレースに関する限り、AppScan Source はこの呼び出し処理を無視します。 AppScan Source トレース では、逸失トレースは報告されません。また、伝達されたデータも、汚染されたデータとして処理されることはありません。

注: 汚染データがメソッドに到達し、そのメソッドが検証ルーチン、エンコード・ルーチン、シンク、汚染伝播元のいずれでもなく、「汚染の可能性なし」としてもマークされていない場合、このメソッドは逸失トレースとして報告されます。非定数の引数と戻り値は、汚染されている可能性も、汚染されていない可能性もあります。逸失トレースは、AppScan Source トレース の呼び出しグラフに表示されます。

トレースなしの検出結果

常に検出結果として表示されるが、トレースを生成しないメソッドまたは API

検証/エンコード・ルーチンなし

API を「検証/エンコード・ルーチンなし」としてマークすると、この API がデータ検証を実行しないことが示されます。

汚染されたコールバック

コールバックとは、主に他の コードから (例えば、下位のフレームワーク内から) 呼び出されるコード内のルーチンです。 コールバックは、引数として他のコードに渡されるため、汚染されている可能性のある引数を使用して後から呼び出されることがあります。 汚染されたデータがコールバックによって引数に渡された可能性がある場合は、そのコールバックを、汚染されたコールバックとしてマークすることができます。 これにより、ルーチン全体を通じて汚染されたデータのフローが明らかになります。

汚染されたコールバックとしてマークされたルーチンは、すべての入力引数が汚染されているものと認識され、呼び出しグラフのルートに位置するルーチンとして (つまり、不明な外部呼び出し元から呼び出されたルーチンとして) 分析されます。その結果、引数から汚染されたコールバックまでのトレースを使用して、AppScan Source によって検出結果のレポートが作成されます。

アプリケーション・コードによって他のコンテキスト内の同じルーチンが呼び出される場合、そのルーチンは、汚染について特に考慮されることなく処理されます。こうしたコンテキストでは、通常の分析が実行されます。

情報

情報項目として識別されるコード行を脆弱性として指定することはできませんが、セキュリティー監査には含める必要があります。

ルールの追加

この作業のトピックでは、カスタム・ルール・ウィザードを使用してカスタム・ルールを追加する場合の手順について説明します。

このタスクについて

注: セキュリティー検出結果またはスキャン範囲検出結果の追加や削除、重大度の変更を行うと、プロジェクトの V-Density に影響します。

手順

1. 「カスタム・ルール」ビューの「カスタム・ルール・ウィザードの起動」ボタンをクリックして、ウィザードを開きます。
2. 「アプリケーション/プロジェクト/ファイルの選択」ページで、ルールを適用するアプリケーションとプロジェクトを選択します。ナレッジベース・データベースに追加したい項目のソース・コードに、現在のアプリケーションとプロジェクトが関連付けられていることを確認してください。使用可能な場合は、「構成」を選択します。
3. 「有効範囲」セクションで、スキャンの有効範囲を設定します。スキャンする言語によって、以下の有効範囲オプションがあります。

表 *21.* 言語別のプロジェクト・ファイル・オプション

言語	プロジェクト・ファイル・オプション
.NET	• プロジェクト全体を対象にメソッド・シグニチャーのスキャンを実行する • プロジェクト外部のファイルを **1** つ以上選択する .NET プロジェクトには、任意の有効なアセンブリーが格納されます (通常は、`.dll` ファイルまたは `.exe` ファイル)。
Java	• プロジェクト全体を対象にメソッド・シグニチャーのスキャンを実行する • プロジェクト内のファイルを **1** つ以上選択する • プロジェクト外部のファイルを **1** つ以上選択する Java プロジェクトには、`.jar` ファイルまたは `.class` ファイル、あるいはクラス・ファイルのディレクトリー階層が格納されます。
C/C++	• プロジェクト全体を対象にメソッド・シグニチャーのスキャンを実行する • プロジェクト内のファイルを **1** つ以上選択する
Visual Basic	`FRM` (フォーム) ファイル/`CLS` (クラス) ファイル/`BAS` (基本) ファイルをスキャンする
Classic ASP	ASP ファイルのみをスキャンする

- デフォルトのスキャン・モードは「プロジェクト全体を対象にメソッド・シグニチャーのスキャンを実行する」です。 このモードではプロジェクト全体がスキャンされ、使用可能なすべてのシグニチャーが返されます。 このスキャン・モードは、時間がかかる場合があります。
- 「プロジェクト内のファイルを **1** つ以上選択する」オプションは、カスタム・ルールを必要とするメソッドが含まれる特定のプロジェクト・ファイルを選択します。

- 「プロジェクト外部のファイルを **1** つ以上選択する」オプションは、このプロジェクト外部のファイルをスキャン対象として指定します。

4. 「キャッシング」セクションで、変更されたプロジェクトまたは変更されたコードを再読み取りするチェック・ボックスを選択します。この場合、脆弱性分析キャッシュも消去されます (現在のプロジェクトが脆弱性分析をキャッシュするように設定されている場合、次回のスキャン時に脆弱性分析キャッシュが再作成されます)。
5. 文字列解析: 文字列解析は、Java プロジェクトまたは Microsoft .NET プロジェクトでの文字列操作をモニターします。これにより、サニタイズ・プログラムおよび検証プログラムのルーチンを自動検出できます。この検出を使用すると、誤検出や検出漏れを削減できます。文字列解析を有効にするには、「文字列解析で検証プログラムやサニタイズ・プログラムの関数を検索できるようにする」チェック・ボックスを選択します。 「インポートされたルールをグローバル・スコープに適用する」チェック・ボックスは、検出されたサニタイズ・プログラムまたは検証プログラムのルーチンを単一のプロジェクトに適用するか、あるいはグローバル・レベルで (すべてのプロジェクトに) 適用するかを決定します。

   注**:** 文字列解析を適用すると、スキャン速度が低下する場合があります。 したがって、この機能はコード変更後にのみ適用し、その後は、後続のスキャンのために無効にすることをお勧めします。 また、検出されたルーチンは 提案 として表示し、監査員がそれらを確認する必要があります。 これらのルーチンは、「カスタム・ルール」ビューに表示できます。
6. 「次へ」をクリックして、ウィザードの次のページに進みます。
7. 「メソッドの選択」ページで、以下を行います。
   a. ナレッジベース・データベースに追加する 1 つまたは複数のメソッドを選択します。このメソッドは、脆弱 API の名前です。

      メソッドのリストは、以下の 2 とおりの方法でフィルタリングできます。
      - 自動フィルタリング: 「フィルター」フィールドにフィルター・テキストを入力します。入力すると、メソッドのリストにフィルターが自動的に適用されます。これは、デフォルトのフィルター・モードです。
      - 手動フィルタリング: 「フィルター」フィールドにフィルター・テキストを入力して、「フィルター」ボタンをクリックし (または Enter を押し)、リストにフィルターを適用します。手動フィルタリングは、多数のメソッドがあり、自動フィルタリングでは遅くなる場合に使用できます。

      いずれの場合にも、ワイルドカードとしてアスタリスク (*) および疑問符 (?) を使用できます。アスタリスクは、連続した複数の文字 (文字数がゼロの場合も含む) を表し、疑問符は単一の文字を表します。

      フィルター・モードを変更するには、「フィルター」ボタンをトグルとして使用します。これは、「フィルター」ボタンをダブルクリックするか、またはキーボードを使用して「フィルター」ボタンにナビゲートしてからスペース・バーを押すことで行えます。手動フィルターをオンにすると、オンになっていない状態の「フィルター」ボタンと、その吹き出しヘルプ「フィルターを適用します **(**自動フィルタリングにする場合は、ダブルク

リックするかスペース・キーを押します) **(Apply filter (double-click or press space to filter automatically))**」が表示されます。自動フィルタリングをオンにすると、オンになった状態のボタンと、その吹き出しヘルプ「手動フィルタリングにします **(Filter manually)**」が表示されます。

メソッドのリストを見やすくするために、展開および省略のアクションを使用できます。ツリー全体を展開または省略するには、右クリックして「すべて展開」または「すべて省略」を選択します。パッケージまたはクラスとその下の項目をすべてを展開するには、パッケージまたはクラスを右クリックして、「子を展開 **(Expand Children)**」を選択します。

複数のメソッドを選択する場合は、キーボードの Ctrl キーまたは Shift キーを使用します。

「フルシグニチャーを表示する」チェック・ボックスを選択すると、ツリーにメソッドの完全修飾シグニチャーが表示されます。例えば、完全修飾 Java シグニチャーには、パッケージ、クラス、メソッド、引数タイプ、戻りの型が含まれます (例: `com.test.vulnerable.VulnClass.vulnerable (java.lang.string;int):int`)。

b. スキャンを実行したときに、メソッドを以下のいずれかのようにマークするかどうかを指定します。
   - 261 ページの『ソース (汚染)』
   - 261 ページの『シンク (汚染の可能性あり)』
   - 261 ページの『汚染伝播元』
   - 261 ページの『汚染の可能性なし』
   - 262 ページの『トレースなしの検出結果』
   - 262 ページの『検証/エンコード・ルーチンなし』
   - 262 ページの『汚染されたコールバック』
   - 262 ページの『情報』

8. メソッドを 261 ページの『汚染の可能性なし』、 262 ページの『検証/エンコード・ルーチンなし』、 261 ページの『汚染伝播元』、または 262 ページの『汚染されたコールバック』 として追加する場合は、「終了」をクリックして、レコードを AppScan Source セキュリティー・ナレッジ・データベースに追加します。
9. メソッドを 261 ページの『ソース (汚染)』 または 262 ページの『情報』 として追加する場合は、以下の手順を実行します。
   a. 「次へ」をクリックして、「ルール属性の割り当て」ページに進みます。
   b. 追加したメソッドごとに、以下を行います: メソッドに割り当てる 1 つ以上のプロパティーを選択します。メソッドの「タイプ」列は、カスタム・ルールに生成される検出結果の脆弱性タイプを示すように更新されます。

      ヒント**:** 同じプロパティーを複数のメソッドに追加するには、キーボードのCtrl キーまたは Shift キーを使用して、メソッドを複数選択し、メソッドに割り当てるプロパティーを選択します。
   c. 「終了」をクリックして、レコードを AppScan Source セキュリティー・ナレッジ・データベース に追加します。

10. メソッドを 261 ページの『シンク (汚染の可能性あり)』 として追加する場合は、以下の手順を実行します。
    a. 「次へ」をクリックして、「ルール属性の割り当て」ページに進みます。
    b. 追加したメソッドごとに、以下を行います:
        - 脆弱性の影響の「重大度」レベル (「高」、「中」、「低」) を選択します。
        - 「脆弱性タイプ」を選択して、メソッドに適用します。

        ヒント: 同じプロパティーを複数のメソッドに追加するには、キーボードのCtrl キーまたは Shift キーを使用して、メソッドを複数選択し、メソッドに割り当てるプロパティーを選択します。
    c. 「終了」をクリックして、レコードを AppScan Source セキュリティー・ナレッジ・データベース に追加します。
11. メソッドを 262 ページの『トレースなしの検出結果』 として追加する場合は、以下の手順を実行します。
    a. 「次へ」をクリックして、「ルール属性の割り当て」ページに進みます。
    b. 追加したメソッドごとに、以下を行います:
        - 脆弱性の影響の「重大度」レベル (「高」、「中」、「低」) を選択します。
        - メソッドに割り当てる「分類」 (「確定」、「要確認」、または「構成」） を選択します。
        - 「脆弱性タイプ」を選択して、メソッドに適用します。

        ヒント: 同じプロパティーを複数のメソッドに追加するには、キーボードのCtrl キーまたは Shift キーを使用して、メソッドを複数選択し、メソッドに割り当てるプロパティーを選択します。
    c. 「終了」をクリックして、レコードを AppScan Source セキュリティー・ナレッジ・データベース に追加します。

Likelihood ルール属性

`Attribute.Likelihood.High` 属性および `Attribute.Likelihood.Low` 属性は、標準装備のルールの一部で、カスタム・ルールの作成に使用することができます。

AppScan Source では、可能性 は、セキュリティー検出結果が悪用される可能性や機会を表します。 AppScan Source は、https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology#Step_2:_Factors_for_Estimating_Likelihood に示された可能性の定義を取り込み、トレース・プロパティーに基づいて可能性を決定することで、その定義を詳細化します。提供された一連のトレース・プロパティー (例えば、ソース API 名、ソース API タイプ、ソース・テクノロジー、あるいはソース・メカニズムなど) によって、AppScan Source は、将来において特定の脆弱性を使用してトレースが悪用される可能性を判別します。

可能性は、トレースのソース・エレメントと結び付けられます。ソースはプログラムへの入力で、ファイル、サーブレット要求、コンソール入力、ソケットなどがあります。 多くの入力ソースの場合、内容と長さについて制限のないデータが返されます。 チェックされていない入力については、汚染のソースと見なされます。

可能性の例には、以下のものがあります。

- HTTP ソースを持つトレース (Request.getQueryString など) とクロスサイト・スクリプティング・シンク (Response.write など) が提供されると、高可能性が決定されるため、検出結果の信頼性が上昇します。
- システム・プロパティー・ソースを持つトレース (getProperty など) とクロスサイト・スクリプティング・シンク (Response.write など) が提供されると、低可能性が決定されるため、検出結果の信頼性が低下します。

可能性は、即時にアクションを実行するか修正する必要がある、優先度が高い要アクションの検出結果を識別するために使用されます。これは、悪用される可能性が高い汚染のソースと結び付けられ、検出結果を分類するためにより微細化されたアプローチを提供することができます。可能性は、汚染のソースに結び付けられた属性として、AppScan Source 脆弱性データベースに保管されます。この機能は、すぐに使用可能です。

IBM は、ソースの可能性要因を判別するための大規模な研究を実施してきています。カスタム・ルール・ウィザードを使用して、ルール・ベースに追加する新規の汚染ソースに可能性情報を追加することができます。これにより、スキャンによって生成された検出結果の分類が改善され、それによってトリアージ・ワークフロー全体の効率性が向上します。

カスタム・ルール・ウィザードには、「可能性」プロパティーに設定可能な 2 つの値 (「高」および「低」) があります。値「高」は、汚染に対してソースが非常に影響を受けやすいことを意味します。つまり、システムに侵入する汚染に対する障壁が非常に低く、攻撃者が悪意のあるデータを手動あるいは自動のいずれの方法でも容易に送信することが可能になります。値「低」は、このソースを介した悪意のあるデータの侵入に対する障壁が非常に高くなります。これは、攻撃者がソースを汚染させるには、システムの内部知識と、攻撃対象のネットワーク上で操作するための権限が必要になることを意味します。

AppScan Source トレースによる入出力トレースのカスタマイズ

アプリケーション (特に Web アプリケーション) によっては、SQL 注入、コマンド注入、クロスサイト・スクリプティングに関連するセキュリティーの脆弱性を識別するため、入出力トレースが必要になることがあります。 AppScan Source トレースにより、検証ルーチンを指定することができます。この検証ルーチンを使用すると、脆弱性レポートを作成する必要がなくなります。 入力データが検証済みでない場合、その他のすべての出力データは脆弱性としてマークされます。

ユーザー定義の検証ルーチンは、入力データを安全なデータに処理してから出力ルーチンに渡すルーチンです。 検証ルーチンで入力データを処理してから出力ルーチンに渡した場合、入力検証での脆弱性は存在しません。 開発者は、トレース機能と連携する独自の入力検証ルーチンと入力エンコード・ルーチンを指定することができます。

パターン・ベースのルールによるカスタマイズ

AppScan Source のパターン・ベースのスキャンは、カスタマイズされた検索基準に基づいてソース・コードの分析を行う機能です。 パターン・ベースのスキャンは、grep 機能と似ています (grep では、1 つ以上のファイルについて、指定された文字文字列やパターンが検索されます)。 トリアージを行う監査員またはセキュリティー・アナリストは、パターン・ベースのスキャンを使用して、特定のパターンを特定のアプリケーションまたはプロジェクトから検索する場合があります。特定のパターンを特定の脆弱性タイプとして定義しておくことにより、ソース・コードのスキャンを実行したときに、そのパターンを脆弱性として識別することができます。 一致項目が AppScan Source によって検出されると、その項目が検出結果表に表示されます。 すぐに使用できる AppScan Source ルール・ライブラリーに、定義済みのルールとルール・セット (ルールの集合) が格納されています。

パターン・ベースのスキャンでは、正規表現 が検索されます。正規表現 (多くの場合、パターンと呼ばれます) とは、特定の構文規則に従って文字列のセットを表現する (または文字列のセットに一致する) 文字列のことです。 検索内容は、ルールを作成して指定します。ルールは、「カスタム・ルール」ビューを使用して AppScan Source セキュリティー・ナレッジ・データベース に追加するカスタム・ルールに似ています。ルールを作成するときは、重大度、分類、脆弱性タイプなどの基準を定義します。

318 ページの『「パターン・ルール・ライブラリー」ビュー』 を使用すると、新規のパターン・ルールとパターン・ルール・セットを作成し、既存のルールとルール・セットの変更または削除を行うことができます。さらに、選択したアプリケーションの「プロパティー」ビュー、選択したプロジェクトの「プロパティー」ビュー、またはスキャン構成を使用して、パターン・ルールとパターン・ルール・セットを適用します (新規ルールを作成するためのダイアログ・ボックスをこれらのビューから起動することもできます)。ルールとルール・セットの適用について詳しくは、 275 ページの『パターン・ルールおよびパターン・ルール・セットの適用』を参照してください。

例えば、以下のようなパターン・ルールを作成することができます。

- ファイル名パターン・マッチング
- 複数のパターンが定義された単一のルール
- 該当項目のないルール

注: パターン・ルールまたはパターン・ルール・セットの作成やカスタム・ルールとカスタム・ルール・セットの変更と削除を行う場合は、パターンの管理権限が必要です。

パターン・ルール・セット

パターン・ルール・セットとは、パターン・ルールの集合のことです。新規のパターン・ルール・セットを追加したり、既存のパターン・ルール・セットの変更や削除を行うことができます。AppScan Source には言語固有の一連のパターン・ルール・セットが用意されており、プロジェクトまたはアプリケーションにそれらのパターン・ルール・セットを適用するように選択できます (例えば **Java/JSP** プロジェクトに **Java** パターン・ルール・セットを適用することができます)。

318 ページの『「パターン・ルール・ライブラリー」ビュー』 を使用すると、新規のパターン・ルールとパターン・ルール・セットを作成し、既存のルールとルール・セットの変更または削除を行うことができます。さらに、選択したアプリケーションの「プロパティー」ビュー、選択したプロジェクトの「プロパティー」ビュー、またはスキャン構成を使用して、パターン・ルールとパターン・ルール・セットを適用します (新規ルールを作成するためのダイアログ・ボックスをこれらのビューから起動することもできます)。ルールとルール・セットの適用について詳しくは、 275 ページの『パターン・ルールおよびパターン・ルール・セットの適用』を参照してください。

AppScan Source に付属する一部のパターン・ルール・セットには、ルールが設定されていないものがあります。こうしたルール・セットには、組織に適したルールを追加することができます。そのようなルール・セットとして、以下のようなものがあります。

- ColdFusion
- JQuery
- クライアント側 JavaScript
- Visual Basic 6
- MooTools

ヒント: 「パターン・ルール・ライブラリー」ビューで、ルール・セットを右クリックして「プロパティー」を選択すると、ルール・セットに関する情報を表示するダイアログ・ボックスが開きます。この「ルール・セット・プロパティー」ダイアログ・ボックスには、パターン・ルール・セット内で定義されているルールの数や、他のルール・セットとの親子関係などの情報が表示されます。また、「表示名」フィールドと「プロジェクト・タイプ」フィールドでルール・セットの表示名とプロジェクト・タイプを変更することもできます。

注: パターン・ルールまたはパターン・ルール・セットの作成やカスタム・ルールとカスタム・ルール・セットの変更と削除を行う場合は、パターンの管理権限が必要です。

「パターン・ルール・ライブラリー」ビューでのルール・セットの作成

パターン・ルール・セットとは、パターン・ルールの集合のことです。ルール・セットの作成方法を理解するには、このトピックの説明に従ってください。

始める前に

注: パターン・ルールまたはパターン・ルール・セットの作成やカスタム・ルールとカスタム・ルール・セットの変更と削除を行う場合は、パターンの管理権限が必要です。

手順

1. 「パターン・ルール・ライブラリー」ビューで、「新規ルール・セット」をクリックします。
2. 「新規ルール・セット」ダイアログ・ボックスで、「名前」フィールドにルール・セットの名前を入力します

3. ルール・セットの適用対象とするプロジェクトのタイプを 1 つ以上選択します。
4. 「**OK**」をクリックします。
5. 新しいルール・セットがルール・セットのリストに表示されます。以下の 2 つのいずれかの方法で、ルール・セットにルールを追加することができます。
   a. 「パターン・ルール」セクションでルールを 1 つ以上選択して、ルール・セットにドラッグ・アンド・ドロップします。
   b. 「パターン・ルール」セクションでルールを 1 つ以上選択して右クリックし、「ルール・セットへの追加」メニュー項目を選択します。「ルール・セットの選択」ダイアログ・ボックスで、ルールを追加したいルール・セットを選択します。

ルール・セットの変更と削除

製品に付属のパターン・ルール・セット、およびユーザーが作成したルール・セットは、「パターン・ルール・ライブラリー」ビューで変更および削除することができます。

注: パターン・ルールまたはパターン・ルール・セットの作成やカスタム・ルールとカスタム・ルール・セットの変更と削除を行う場合は、パターンの管理権限が必要です。

ルール・セットの変更

ルール・セットは、以下のように変更することができます。

- 269 ページの『「パターン・ルール・ライブラリー」ビューでのルール・セットの作成』の説明に従って新しいルール・セットを設定することにより、既存のルール・セットにルールを追加することができます。
- 1 つ以上のルールをルール・セットから削除するには、削除するルールを選択してから以下のいずれかの操作を実行します。
  - 「セットからのルールの削除」をクリックします。
  - 右クリックして「セットからのルールの削除」を選択します。
- 以下の 2 つのいずれかの方法で、ルール・セットを別のルール・セットに追加することができます。
  - ルール・セットを選択して、別のルール・セットにドラッグ・アンド・ドロップします。
  - ルール・セットを右クリックして「子としてのルール・セットの追加」を選択し、親ルール・セットとして追加するルール・セットを「ルール・セットの選択」ダイアログ・ボックスで選択します。
- ルール・セットの「表示名」および「プロジェクト・タイプ」を変更することができます。ルール・セットを右クリックし、「プロパティー」を選択すると、「ルール・セット・プロパティー」ダイアログ・ボックスが開きます。このダイアログ・ボックスで、「表示名」フィールドを編集することができます。あるいは、「プロジェクト・タイプ」フィールドで「編集」ボタンをクリックし、1 つ以上のプロジェクト・タイプを選択することもできます。

ルール・セットの削除

ルール・セットを削除するには、ルール・セットを選択してから以下のいずれかの操作を実行します。

- 「ルール・セットの削除」をクリックします。
- 右クリックして「削除」を選択します。

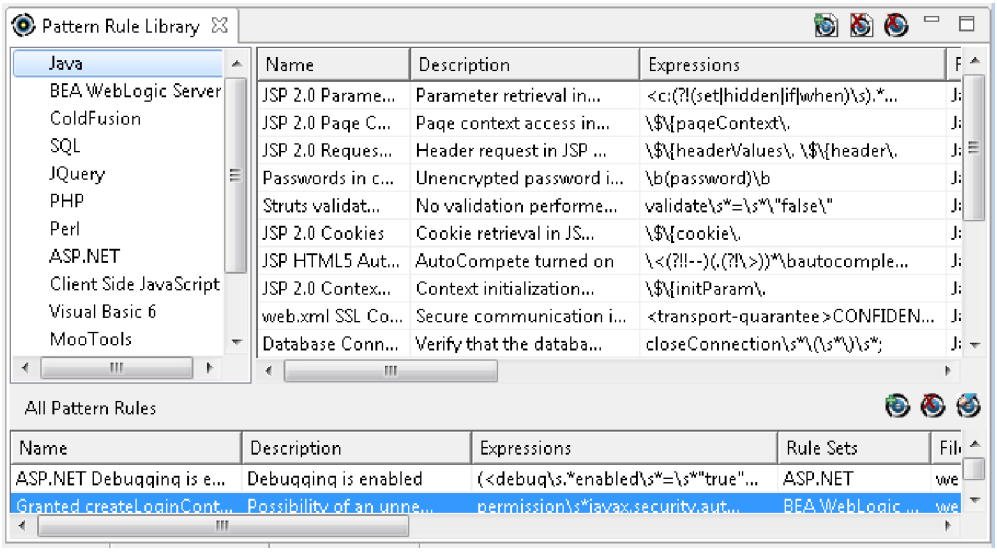
パターン・ルール

AppScan Source のテキスト・ルールは、Extended Global Regular Expressions Print (egrep)、Global Regular Expressions (grep)、Perl のいずれかの正規表現として指定することができます。これらの正規表現 (英数字と特殊文字のみの組み合わせを使用する文字列値を含む表現) により、ルールのマッチングが実行されます。

文字	説明
^	行頭
$	行末
¥n、¥t、または ¥r	リテラル改行、タブ、リターン
[xyz]	列挙されているうちの任意の文字
[^abx]	列挙されている以外の任意の文字
[a-fA-F0-9]	任意の 16 進数文字
.	任意の文字
|	いずれか
\	特殊文字の意味の取り消し ¥$ ¥^ ¥¥ ¥?

パターン・ルールは、(AppScan Source データベース 内の) グローバル・パターン・ルール・ライブラリーに格納され、プロジェクト間とアプリケーション間で共有することができます。また、ルールとルール・セットをすべてのユーザーで共有することもできます。ルールは、参照によって追加されます。そのため、基礎となるルールを削除しなくても、関連するオブジェクト内の参照を削除することにより、ルールを無効にすることができます。

ルールは、「パターン・ルール・ライブラリー」ビュー、「エクスプローラー」ビューの「プロパティー」タブ、またはスキャン構成で作成します。 AppScan Source をインストールすると、AppScan Source に用意されているルールが「パターン・ルール・ライブラリー」ビューに表示されます。 このビューで、ルールの編集、削除、作成を行うことができます。



重要: 検索基準は追加または削除することができますが、それぞれのパターン・ベースのルールには 1 つ以上の検索基準を指定する必要があります。

テキスト・パターンの検索

特定のソース・ファイル内でパターン・ベースのスキャンを実行すると、拡張子別にファイル内のテキスト・パターンが検索されるため、ソース・ファイルや XML 構成ファイルなどのテキスト・ファイル内を検索することができます。

例えば、不適切な E メール・アドレスがアプリケーション内にハードコーディングされないようにするためのパターン検索を作成することができます。以下の例では、企業の E メール・アドレスがアプリケーション内で使用されないようにする場合に、`.*@mycompany.com` などのパターンを検索することができます。

例

このパターンで検出される項目	パターン
E メール・アドレス	`[A-Za-z]¥.[A-Za-z]@[A-Za-z][A-Za-]¥.com`
パターンのすべてのインスタンス (passWord = など)	`[Pp][Aa][Ss][Ss][Ww][Oo][Rr][Dd]¥W*=`
MD5 ハッシュ・アルゴリズムの任意のインスタンス	`getInstance[[:space:]]*¥ ([[:space:]]*"MD5`

パターン・ルールの作成

ルールは、「パターン・ルール・ライブラリー」ビュー、プロジェクトまたはアプリケーションの「プロパティー」ビュー、またはスキャン構成内で作成できます。

始める前に

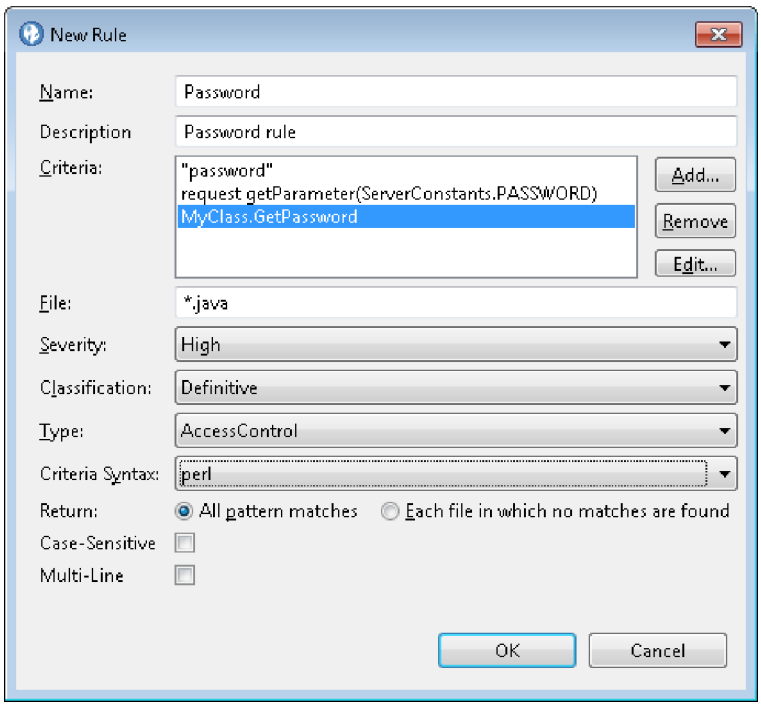
注: パターン・ルールまたはパターン・ルール・セットの作成やカスタム・ルールとカスタム・ルール・セットの変更と削除を行う場合は、パターンの管理権限が必要です。

ルールの作成は「新規ルール」ダイアログ・ボックスで行います。

- 「パターン・ルール・ライブラリー」ビューでこのダイアログ・ボックスを開くには、「新規ルール」をクリックします。
- スキャン構成では、「パターン分析」タブを選択してから、「パターン分析」チェック・ボックスを選択します。そのタブの「パターン・ルール」セクションで、「追加」をクリックして「パターン・ルールの追加」ダイアログ・ボックスを開きます。このダイアログ・ボックスで、「新規ルールの作成」をクリックして「新規ルール」ダイアログ・ボックスを開きます。
- 選択したアプリケーションまたはプロジェクトの「プロパティー」ビューからこのダイアログ・ボックスを開くには、「プロパティー」ビューの「ルールとルール・セット」タブを選択し、「追加」をクリックするか、「ルール」セクション内を右クリックして「追加」を選択します。「ルールの選択」ダイアログ・ボックスで「新規ルール」をクリックします。

手順

1. 「新規ルール」ダイアログ・ボックスの「名前」フィールドにルール名を入力します。
2. オプション: 「説明」フィールドにルールの説明を入力します。
3. 「基準」フィールドを入力します。「追加」をクリックして、各ルールの正規表現を入力します。
4. ファイル・タイプ (`*.java` や `*.xml` など) を指定します。任意のファイル・タイプを入力することができます。その際、ワイルドカード文字を使用することもできます。
5. オプション: 「重大度」フィールドで、以下のいずれかを選択します。
   - 高
   - 中
   - 低
   - 情報
6. オプション: 「分類」フィールドで、以下のいずれかを選択します。
   - 確定
   - 要確認
   - スキャン範囲
7. オプション: スキャンで検索する脆弱性タイプを選択します。 (脆弱性タイプについて詳しくは、AppScan Source セキュリティー・ナレッジ・データベースを参照)



8. オプション: 基準の構文として、以下のいずれかを選択します。
   - **egrep**
   - **grep**
   - **perl**
9. オプション: 返される結果に「パターンに一致する全項目」と「一致項目が見つからなかったファイル」のどちらを含むかを指定します。一致する項目がない場合、そのパターンは該当項目のないルールです。
10. オプション: パターン・マッチングで大/小文字を区別する場合は、「大文字と小文字を区別する」チェック・ボックスを選択します。
11. オプション: 複数行にまたがるパターンにルールが一致する必要がある場合は、「複数行」チェック・ボックスを選択します。
12. 「**OK**」をクリックすると、ルール内の正規表現が正しいかどうかが検証されます。作成されたルールが、パターン・ルール・ライブラリーに追加されます。

パターン・ルールの変更と削除

作成したパターン・ルールは、「パターン・ルール・ライブラリー」ビューで変更および削除することができます。

注: パターン・ルールまたはパターン・ルール・セットの作成やカスタム・ルールとカスタム・ルール・セットの変更と削除を行う場合は、パターンの管理権限が必要です。

ルールの変更

ルールを編集するには、ルールを選択してから以下のいずれかの操作を実行します。

- 「ルールの編集」をクリックします。
- 右クリックして、「編集」を選択します。

「ルールの編集」ダイアログ・ボックスが表示されます。このダイアログ・ボックスで、ルール名以外の設定を変更することができます。

ルールの削除

ルールを 1 つ以上選択してから、以下のいずれかの操作を実行します。

- 「ルールの削除」をクリックします。
- 右クリックして「削除」を選択します。

パターン・ルールおよびパターン・ルール・セットの適用

ルールとルール・セットは、「プロパティー」ビューを使用してアプリケーション・レベルまたはプロジェクト・レベルで適用するか、またはスキャン構成を使用して適用します。適用されたルールを使用してアプリケーションまたはプロジェクトをスキャン (またはルールを含むスキャン構成を使用) すると、ルールによる検索結果が「検出結果」ビューに表示されます。

スキャン構成でのルールとルール・セットの適用

パターン・ベースのスキャンを有効にするには、「パターン分析」チェック・ボックスを選択します。これを選択すると、「パターン・ルール・セット」セクションと「パターン・ルール」セクションが有効になります。

- ルール・セットを追加するには、「パターン・ルール・セット」セクションで「追加」をクリックします。これによって「パターン・ルール・セットの追加」ダイアログ・ボックスが開き、ここで 1 つ以上のルール・セットを選択できます。ルール・セットを選択すると、そこに含まれるルールがダイアログ・ボックスの右側に表示され、そのルール・セットの適用対象のプロジェクト・タイプが「プロジェクト・タイプ」フィールドにリストされます。「**OK**」をクリックして、選択したルール・セットを追加します。
- ルールを追加するには、「パターン・ルール」セクションで「追加」をクリックします。これによって「パターン・ルールの追加」ダイアログ・ボックスが開き、ここで 1 つ以上のルールを選択できます。「新規ルールの作成」をクリックして新規ルールを作成することもできます (272 ページの『パターン・ルールの作成』を参照してください)。新規ルールを作成すると、そのルールはリストに追加され、選択対象になります。ルールを選択または作成したら、「**OK**」をクリックして、スキャン構成に追加します。

  ヒント: 「パターン・ルールの追加」ダイアログ・ボックスでは、ツールチップのヘルプによって各ルールに使用される式が示されます。

「プロパティー」ビューを使用したルールとルール・セットの適用

「エクスプローラー」ビューでプロジェクトまたはアプリケーションを選択し、「プロパティー」ビューの「パターン・ルール/パターン・ルール・セット」タブで、以下に示す変更を行います。アプリケーションまたはプロジェクトに適用するルールとルール・セットを指定したら、アプリケーションまたはプロジェクトのプ

ロパティーを保存します。この状態でアプリケーションまたはプロジェクトのスキャンを実行すると、次回からはここで指定したルールが使用されるようになります。

- ルール・セットを追加するには、「ルール・セット」セクションで「追加」をクリックするか、セクション内で右クリックして「追加」を選択します。これにより、「ルール・セットの選択」ダイアログ・ボックスが開き、追加するルール・セットを選択できます。
- ルール・セットを削除してアプリケーションまたはプロジェクトのスキャンで使用されないようにするには、そのルール・セットを選択して「削除」をクリックするか、ルール・セットを右クリックして「削除」を選択します。
- ルールを追加するには、「ルール」セクションで「追加」をクリックするか、セクション内で右クリックして「追加」を選択します。これにより、「ルールの選択」ダイアログ・ボックスが開き、追加するルールを選択できます。このダイアログ・ボックスでは、「新規ルール」をクリックして、新規ルールを作成することもできます (272 ページの『パターン・ルールの作成』を参照)。新規ルールを作成すると、そのルールはリストに追加され、選択対象になります。ルールを選択または作成した後、「**OK**」をクリックして、そのルールを追加します。
- ルールを削除してアプリケーションまたはプロジェクトのスキャンで使用されないようにするには、そのルールを選択して「削除」をクリックするか、ルールを右クリックして「削除」を選択します。これらのアクションを使用する場合、ルールの複数選択と複数のルールの削除も可能です。

「スキャン構成」ビュー

「スキャン構成」ビューを使用して、スキャンの起動時に使用できる構成を作成することができます。このビューを使用すると、デフォルトのスキャン構成の設定も可能です。スキャン構成では、スキャン時に使用するソース・ルールを指定し、多数のスキャン設定を組み込むことができます。 スキャン構成で設定を行うと、良好なスキャン結果が得られることが多く、また、これらの設定を保存することができるため、スキャンを容易に、しかも短時間で行うことができます。

「スキャン構成」ビューには、以下の主なセクションがあります。

- 130 ページの『スキャン構成の管理』
- 130 ページの『「全般」タブ』
- 131 ページの『「汚染フロー分析」タブ』
- 132 ページの『「パターン分析」タブ』

スキャン構成の管理

このセクションは、スキャン構成を選択、追加、削除、保存、および共有する場合や、スキャン構成をデフォルトとして設定する場合に使用します。

- 新規スキャン構成を作成するには、「新規」をクリックします。スキャン構成の設定が完了したら、「保存」をクリックして変更内容を保存します。 スキャン構成をデフォルトとして設定するには、保存後に「デフォルトとして選択」をクリックします。 デフォルトのスキャン構成がどのように使用されるかについては、 119 ページの『ソース・コードのスキャン』を参照してください。
- 既存のスキャン構成を操作するには、既存のスキャン構成をリストから選択します。

- スキャン構成の設定を変更する場合は、「保存」をクリックして変更内容を保存します (不要な変更内容は、別のスキャン構成に切り替えてから「破棄」をクリックすると、破棄することができます)。
- 選択したスキャン構成を削除するには、「削除」をクリックします。
- スキャン構成を複写するには、「複写」をクリックします。これにより、元のスキャン構成の設定に基づいて新しいスキャン構成が作成されます。
- スキャン構成をデフォルトとして設定するには、「デフォルトとして選択」をクリックします。 デフォルトのスキャン構成がどのように使用されるかについては、 119 ページの『ソース・コードのスキャン』を参照してください。
- スキャン構成を他のユーザーと共有するには、「共有」をクリックします。スキャン構成が AppScan Source データベース に保存されます。

  注**:** スキャン構成を共有する (あるいは共有スキャン構成を変更または削除する) には、「共有構成の管理」権限が必要です。権限の設定について詳しくは、「*IBM Security AppScan Source* インストールと管理のガイド」を参照してください。

注**:** AppScan Source には、標準装備のスキャン構成が用意されています。 これらを変更または削除することはできません。これらのスキャン構成をリストで選択すると、複写したり、その設定を表示したりすることができます。

「全般」タブ

基本情報

このセクションでは、スキャン構成に名前を付けて説明を提供することができます。

フィルター

このセクションでは、スキャン構成を使用すると必ずスキャンに適用されるフィルターを 1 つ以上選択できます。フィルターを選択するときは、AppScan Source 事前定義フィルターまたは 共有フィルター、あるいは自分で作成したフィルターを選択できます。詳しくは、 123 ページの『スキャン構成の管理』を参照してください。

「汚染フロー分析」タブ

汚染フロー分析

汚染フロー分析を有効にし、その有効範囲を設定します。

スキャン・ルール

このセクションは、スキャンで有効になるソース・ルールを判別するために使用します。

ソースはプログラムへの入力で、ファイル、サーブレット要求、コンソール入力、ソケットなどがあります。 一部のソース・ルールを除外することにより、スキャンの速度を上げたり、関係のない入力に起因する脆弱性の検出を避けることができます。

ルールが特定の脆弱性、メカニズム、属性、またはテクノロジーに関連していることを示すには、ルール・プロパティーでルールをタグ付けします。 これらのプロパティーはルール・セットにグループ化され、これらのルール・セットは、関連したルールの共通セットに対応します。 ルール・セットまたは個々のルール・プロパティーのいずれかを指定することにより、スキャンに含めるソース・ルールを制限できます。

- スキャンに組み込む 1 つ以上の脆弱性タイプ (ルール・セット内でタイプ別に編成される) を選択します。
  - すべて: これを選択すると、サポートされるすべての入力のソースに起因する脆弱性が検出されます。
  - ユーザーの入力: これを選択すると、エンド・ユーザーによる入力に起因する脆弱性が検出されます。
  - **Web** アプリケーション: これを選択すると、Web アプリケーションのリスクに起因する脆弱性が検出されます。
  - エラー処理およびロギング: これを選択すると、エラー処理とロギングのメカニズムに起因する脆弱性が検出されます。
  - 環境: これを選択すると、構成ファイル、システム環境ファイル、およびプロパティー・ファイルに起因する脆弱性が検出されます。
  - 外部システム: これを選択すると、外部エンティティーに起因する脆弱性が検出されます。
  - データ・ストア: これを選択すると、データ・ストア (データベースやキャッシュ処理など) に起因する脆弱性が検出されます。
  - 異常な項目: これを選択すると、通常は実動アプリケーションの一部ではないルーチンに起因する脆弱性が検出されます。
  - ファイル・システム: これを選択すると、ファイル・システムに起因する脆弱性が検出されます。
  - 機密データ: これを選択すると、機密データに起因する脆弱性が検出されます。

  吹き出しテキストは、このセクションの各ルール・セットについて記述しています。
- スキャンに組み込む個々のスキャン・ルール・プロパティーを選択します。「選択済みのルール・セットを破棄し、個々のルール・プロパティーを選択」をクリックします。 これにより、「ルール・プロパティーの選択」ダイアログ・ボックスが開き、個々のルール・プロパティーを選択できるようになります。このダイアログ・ボックスでの作業が完了すると、選択されていたルール・セットがすべて破棄されます。 選択されたルール・プロパティーを持つスキャン・ルールがスキャンに使用されます。

詳細設定

このセクションは、上級者向けです。 このセクションには、スキャン結果を向上させるための、さまざまな設定が含まれています。 吹き出しテキストは、このセクションの各設定について記述しています。

「パターン分析」タブ

パターン分析

このセクションを使用して、スキャン構成を使用する場合にパターン・ベースのスキャンを有効にします。パターン・ベースのスキャンは、カスタマイズされた検索基準に基づいてソース・コードの分析を行う機能です。

「パターン・ルール・セット」および「パターン・ルール」

これらのセクションを使用して、パターン分析時に使用するルールとルール・セットを追加します。詳しくは、 268 ページの『パターン・ベースのルールによるカスタマイズ』および 123 ページの『スキャン構成の管理』を参照してください。

「プロパティー」ビュー: 選択したアプリケーション

このビューでは、選択したアプリケーションの属性を構成します。 アプリケーション属性は、前に作成したグローバル属性に依存します。

- 『概要』
- 『除外およびフィルター (Exclusions and Filters)』
- 280 ページの『ルールおよびルール・セット』
- 280 ページの『変更された検出結果』
- 280 ページの『カスタム検出結果』

概要

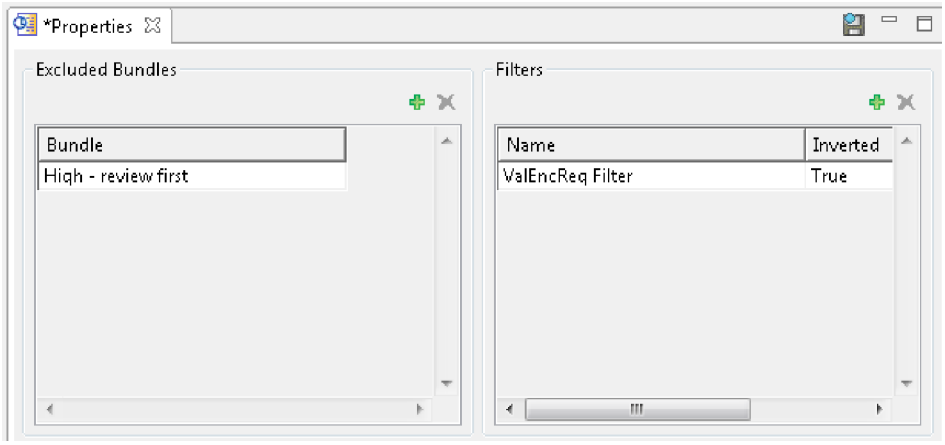
「概要」タブには、以下の内容が表示されます。

- アプリケーション名。アプリケーション名は、フィールドに新しい名前を入力して変更することができます。
- アプリケーション属性

除外およびフィルター (Exclusions and Filters)

このタブでは、選択したアプリケーションに対して既存のフィルターを指定でき、フィルターの適用方法を指定できます (フィルターは直接適用することも、反転して適用することもできます)。このタブでは、スキャンから結果を除外するバンドルを管理することもできます。フィルターについては、 157 ページの『第 5 章 トリアージおよび分析』を参照してください。また、グローバル・フィルターの適用について詳しくは、 182 ページの『グローバル・フィルターの適用』を参照してください。

除外およびフィルタリングされた検出結果はスキャン結果には表示されず、アプリケーションまたはプロジェクトのメトリックで計算に入れられることはありません。



ルールおよびルール・セット

「エクスプローラー」ビューでアプリケーションを選択すると、「プロパティー」ビューの「パターン・ルール/パターン・ルール・セット」タブで、アプリケーションのスキャン時に適用されるパターン・ルールとパターン・ルール・セットを追加することができます。パターン・ベースのスキャンを使用して、検出結果として表示させたいテキスト・パターンを検索します。 個々のルールとルール・セットをアプリケーションとプロジェクトの両方に適用できます。パターン・ベースの分析については 268 ページの『パターン・ベースのルールによるカスタマイズ』を、「プロパティー」ビューでのルールとルール・セットの適用方法については 275 ページの『パターン・ルールおよびパターン・ルール・セットの適用』を参照してください。

変更された検出結果

「変更された検出結果」タブでは、前に変更した検出結果を表示、編集、または削除するか、既存の検出結果を変更します。 変更された検出結果 とは、脆弱性タイプ、重大度、分類、または注が変更された検出結果です。

カスタム検出結果

「カスタム検出結果」タブでは、カスタム検出結果を表示、追加、編集、または削除します。詳しくは、 198 ページの『カスタム検出結果』を参照してください。

「プロパティー」ビュー: 選択したプロジェクト

「プロパティー」ビューのこのモードでは、選択したプロジェクトのパラメーターを構成します。 プロジェクト属性は、前に作成したグローバル属性に依存します。プロパティーは、選択したプロジェクトによって異なります。

- 281 ページの『選択したプロジェクトの「概要」タブ』
- 282 ページの『フィルター』
- 282 ページの『パターン・ルールとパターン・ルール・セット』
- 283 ページの『ファイル拡張子』
- 284 ページの『ソース』
- 284 ページの『JavaServer Page (JSP) プロジェクト依存関係』

- 285 ページの『プロジェクト依存関係』
- 285 ページの『コンパイル』
- 286 ページの『最適化』
- 286 ページの『プリコンパイル・タブ (ASP.NET のみ)』

選択したプロジェクトの「概要」タブ

「概要」タブには、以下の内容が表示されます。

- プロジェクトの名前。 プロジェクト名は、フィールドに新しい名前を入力して変更することができます。
- プロジェクトのファイル名およびパス
- プロジェクト・タイプ
- このセクションには、ターゲットの構成が表示されます。 .NET および C++ のプロジェクトの場合、このセクションには、「プロジェクト依存関係」タブに保存されているターゲットの構成が表示されます。 その他のすべてのプロジェクト・タイプの場合、このセクションには、「デフォルト」が表示されます。
- フィルター・オプション: 「外部ソースに含まれている検出結果をフィルタリング」を選択して、スキャンされたプロジェクトのソース・ファイルではないファイルで検出された検出結果をすべてフィルタリングで除外します。 このオプションにより、検出結果が `ASP.NET` などのコンパイラー生成ファイルまたは一時ファイルで報告されるプロジェクトで、不要な手間が軽減されます。
- 脆弱性分析キャッシュ・オプション: 反復してスキャンを行い、カスタム・ルールを追加してから、ソース・コードを変更せずに再スキャンを行うことによって、コード・ベースの評価を詳細化する場合、脆弱性分析キャッシュを使用するようにプロジェクト・プロパティーを設定すると、スキャン時間を大きく削減することができます。 これを行うには、プロジェクト・プロパティーで、「脆弱性分析キャッシュを有効にする」チェック・ボックスを選択します。 このチェック・ボックスを選択した後に最初にプロジェクトをスキャンすると、脆弱性分析キャッシュが作成されます。 プロジェクトのすべての後続のスキャンで、脆弱性分析キャッシュが使用され、スキャン時間が削減されます。

  脆弱性分析キャッシュおよび Java 増分分析が有効にされた状態で作成されたキャッシュを消去するには、「キャッシュの消去」をクリックします。次にプロジェクトをスキャンすると、完全スキャンが実行され、新規の脆弱性分析キャッシュが作成されます。以下のような場合に、キャッシュの消去が必要となることがあります。

  - 最後のスキャン以降にプロジェクト内のソース・コードが変更された。
  - ソース・ファイルの追加または削除などの、プロジェクト構成の変更を行った。
  - コード構成オプションを変更した。 例えば、Java をスキャンしていて、クラスパスが変更された場合や、C または C++ をスキャンしていて、`include` パスまたはプリプロセッサー定義を変更した場合に、キャッシュを消去したほうがよいことがあります。
  - Java 増分分析を有効にしており、完全スキャンを実行したい、あるいはキャッシュを消去することで修復できる問題が発生している。詳しくは、 132 ページの『Java の増分分析』 を参照してください。

注**:** カスタム・ルール・ウィザードでカスタム・ルールを作成するときに、「キャッシュの消去」チェック・ボックスを選択することによって、脆弱性分析キャッシュを消去することもできます。

- 文字列解析: 文字列解析は、Java プロジェクトまたは Microsoft .NET プロジェクトでの文字列操作をモニターします。これにより、サニタイズ・プログラムおよび検証プログラムのルーチンを自動検出できます。この検出を使用すると、誤検出や検出漏れを削減できます。文字列解析を有効にするには、「文字列解析で検証プログラムやサニタイズ・プログラムの関数を検索できるようにする」チェック・ボックスを選択します。 「インポートされたルールをグローバル・スコープに適用する」チェック・ボックスは、検出されたサニタイズ・プログラムまたは検証プログラムのルーチンを単一のプロジェクトに適用するか、あるいはグローバル・レベルで (すべてのプロジェクトに) 適用するかを決定します。

  注**:** 文字列解析を適用すると、スキャン速度が低下する場合があります。 したがって、この機能はコード変更後にのみ適用し、その後は、後続のスキャンのために無効にすることをお勧めします。 また、検出されたルーチンは 提案 として表示し、監査員がそれらを確認する必要があります。 これらのルーチンは、「カスタム・ルール」ビューに表示できます。
- ファイル・エンコード: プロジェクト内のファイルの文字エンコードは、AppScan Source がファイルを適切に読み取る (そして、例えば、それらをソース・ビューに正しく表示する) ことができるように設定する必要があります。

  注**:** AppScan Source プロジェクトのデフォルトのファイル・エンコードは、**ISO-8859-1** です。 デフォルトのファイル・エンコードは、全般設定ページで変更できます。

フィルター

このタブでは、選択したプロジェクトに対して既存のフィルターを指定でき、フィルターの適用方法を指定できます (フィルターは直接適用することも、反転して適用することもできます)。 フィルターについては、 157 ページの『第 5 章 トリアージおよび分析』を参照してください。また、グローバル・フィルターの適用について詳しくは、 182 ページの『グローバル・フィルターの適用』を参照してください。

パターン・ルールとパターン・ルール・セット

「エクスプローラー」ビューでプロジェクトを選択すると、「プロパティー」ビューの「パターン・ルール/パターン・ルール・セット」タブで、プロジェクトのスキャン時に適用されるパターン・ルールとパターン・ルール・セットを追加することができます。パターン・ベースのスキャンを使用して、検出結果として表示させたいテキスト・パターンを検索します。 個々のルールとルール・セットをアプリケーションとプロジェクトの両方に適用できます。パターン・ベースの分析については 268 ページの『パターン・ベースのルールによるカスタマイズ』を、「プロパティー」ビューでのルールとルール・セットの適用方法については 275 ページの『パターン・ルールおよびパターン・ルール・セットの適用』を参照してください。

ファイル拡張子

プロジェクトの有効なファイル拡張子を構成または追加したり、スキャンからファイルを除外して Web ファイルとして拡張子を指定したりするには、このタブを使用します。

「ファイル拡張子」セクションには、現行プロジェクト・タイプの 117 ページの『プロジェクト・ファイル拡張子』 設定ページでグローバルに設定された拡張子がリストされます (「ファイル拡張子のセット」メニューを使用して、別のプロジェクト・タイプのファイル拡張子を選択できます)。現行プロジェクトのスキャンから拡張子を除外するには、リストでその拡張子を選択し、「拡張子の除外」をクリックします。これにより、その拡張子はタブの「除外する拡張子」セクションにリストされます。

プロジェクトの拡張子を追加するには、「追加の拡張子」セクションで「拡張子の追加」を選択してから、ファイル拡張子を入力し、その拡張子を持つファイルがスキャンされるか、Web ファイルと見なされるか、または除外されるかを示します。

表 22. ファイル拡張子の設定

設定	説明	使用例
「スキャン」または「評価」	指定された拡張子を持つファイルを完全分析に含みます。	• Java プロジェクト用の .xxx 拡張子が作成され、「スキャン」または「評価」のマークが付けられると、その拡張子を持つファイルはコンパイルされ、スキャンされます。 • ファイルのコンパイルとスキャンを行わない場合 (C++ のヘッダー・ファイルなど)、そのファイルはプロジェクトの一部にすることができますが、「スキャン」または「評価」のマークはつきません。これらのファイルは、プロジェクトに含まれ、パターン・ベースの分析時に検索されます。
Web ファイル	JSP コンパイル用に指定の拡張子を持つファイルにマークを付けます。この設定により、AppScan Source は Web ソースを非 Web ソースと分離することができます。	Java プロジェクト用の .yyy 拡張子が作成され、「**Web** ファイル」のマークが付けられると、その拡張子を持つファイルは、プロジェクトで Web ソースとして調整されます。AppScan Source が分析の準備をすると、これらのファイルは分析のためにクラスにプリコンパイルされます。

表 22. ファイル拡張子の設定 (続き)

設定	説明	使用例
除外	指定の拡張子を持つファイル用に、プロジェクトでソース・ファイルを作成しません。この拡張子を持つファイルはスキャンされません。	コンパイルのためにプロジェクトに必要であるものの、分析に組み込む必要がないファイルの `.zzz` 拡張子を作成します。

ソース

スキャンに含めるソースを指定します。

- 作業ディレクトリー: AppScan Source プロジェクト・ファイル (`ppf`) の位置であり、すべての相対パスのベース。
- 「ソース・ルートの追加」および「ソース・ルートの削除」: 「ソース」タブに、プロジェクト構成ウィザードからプロジェクトに対して規定されたプロパティー、またはインポートされた `ppf` で定義されたプロパティーが表示されます。

  「ソース・ルートの削除」は、「ソース・ルート」アイコンが選択されている場合のみ使用可能です。 ソース・コード・ルート・ディレクトリーの削除に使用します。
- ソース・ルートの検出 (Java プロジェクトのみ): AppScan Source for Analysis が自動的にすべての有効なソース・ルートを検索できるようにします。
- プロジェクト・ファイルは、「ソース・ルート」 アイコンの下に表示されます。 スキャンから除外されたファイルには、赤いファイル・アイコンが付いています。 (除外済みファイルを右クリックすると、そのメニューで「除外」は無効に、「含める」は有効になっています)。組み込みファイルを除外するには、ファイルを右クリックして、メニューで「除外」を選択します。 除外済みファイルを組み込むには、ファイルを右クリックして、メニューで「含める」を選択します。

JavaServer Page (JSP) プロジェクト依存関係

「JSP プロジェクト依存関係」タブに、指定された JSP プロジェクト用に規定されたプロパティーが表示されます。

- Web (JSP) コンテンツを含む: プロジェクトが、JavaServer Pages を含む Web アプリケーションであるかどうかを示します。
- Web コンテキスト・ルート: `WEB-INF` ディレクトリーを含む `WAR` ファイルまたはディレクトリー。 Web コンテキスト・ルートは、有効な Web アプリケーションのルートでなければなりません。
- JSP コンパイラー: 製品に付属の Tomcat 7 が、デフォルトの JSP コンパイラー設定です (デフォルト JSP コンパイラーは「Java および JSP」設定ページで変更できます)。 AppScan Source にサポートされるコンパイラーについて詳しくは、http://www.ibm.com/support/docview.wss?uid=swg27027486 を参照してください。

Apache Tomcat バージョン 7 および 8 は、AppScan Source のインストール済み環境に含まれています。「**Tomcat 7**」および「**Tomcat 8**」設定ページが未構成の場合、AppScan Source は、提供されている Tomcat JSP コンパイラー (現在デフォルトとしてマーク) を使用して JSP ファイルをコンパイルします。外部でサポートされている Tomcat コンパイラーを使用したい場合は、Tomcat 設定ページを使用して、ローカルの Tomcat インストール済み環境を示します。

Oracle WebLogic サーバー または WebSphere Application Server を使用する場合は、分析時にアプリケーション・サーバーを JSP コンパイルに使用できるようにするため、適切な設定ページを構成して、アプリケーション・サーバーのローカルのインストール済み環境を示す必要があります。この構成をまだ完了していない場合は、JSP コンパイラーを選択する際に構成を行うようにメッセージによって指示されます。メッセージ内の「はい」をクリックすると、該当する設定ページに進みます。「いいえ」をクリックすると、JSP コンパイラーの選択項目の隣に警告リンクが表示されます (リンクを選択すると、設定ページが開きます)。

プロジェクト依存関係

「プロジェクト依存関係」タブには、プロジェクト・プロパティーが表示されます。 このタブの「構成」の設定は、言語により異なります。以下に例を挙げます。

- 「オプション」を使用すると、追加で必要なコンパイラー・パラメーターを選択できます。
- JDK 設定は Java に固有です。
- プリプロセッサー定義は C/C++ コードに固有です。 プリプロセッサー定義を指定するときは、コンパイラーの `-D` オプションを含めないでください (例えば `-Da=definition1` の代わりに `a=definition1` を使用してください)。複数の定義を指定するときは、セミコロンで区切ったリストを使用します。
- ターゲットの構成は、.NET および C++ のプロジェクトでのみ使用できます。

コンパイル

- オプション: プロジェクト構成に追加で必要なコンパイラー・パラメーター。
- JDK の使用:「設定」で構成した、プロジェクトのコンパイルに使用される JDK を示します。 103 ページの『第 3 章 設定』を参照してください。

  Java プロジェクトは、ローカルの Java Development Kit (JDK) の位置を参照する可能性があります。プロジェクトがサーバーに移動すると、JDK パスは無効になる場合があります。ローカル・プロジェクトをサーバーに転送するには、所定の JDK を指定するプロジェクトごとにデフォルトの JDK パスを指定する必要があります。

  注**:** 製品に付属の JSP プロジェクトのデフォルト・コンパイラーは、Tomcat 7 です。これには、Java バージョン 1.6 以上が必要です。 **Tomcat 7** をデフォルトのまま使用している場合、古い JDK を選択すると、以下のスキャン中のコンパイル・エラーが発生します。
- 検証: 「検証」をクリックして、プロジェクト依存関係が正しく構成されていることを確認します。 Java プロジェクトをチェックして、ソース同士やクラスパス間で構成の競合があるかどうか、およびコンパイル・エラーがあるかどうかを

調べます。 クラスパス内のクラスが、ソース・ルートで重複している場合、競合が存在します。 (競合が存在する場合、クラスパスを変更して、クラスの競合を削除してください。)

競合をチェックした後で、「検証」をクリックして、プロジェクトをコンパイルできるかどうか、およびコンパイル・エラーがレポートされるかどうかを判別します。

最適化

- プリコンパイル済みクラス: スキャン中にコンパイルするのではなく、プリコンパイル済み Java または JSP クラス・ファイルを使用します。 このオプションを選択すると、ソース・ステージ・オプションが無効になります。
- コンパイル・エラーの影響を最小化するためにソース・ファイルをステージする: AppScan Source がソースをステージング・ディレクトリーにコピーするかどうかを制御します。

  「ディレクトリーと一致しないパッケージの修正」では、Java コンパイルが各ソース・ファイルを開く必要があります。

  「スキャンの合間にステージング領域をクリーンアップ」によって、スキャンと次のスキャンの間のパフォーマンスが向上します。

プリコンパイル・タブ (ASP.NET のみ)

プリコンパイルは、Web サイトの特殊なページ (デフォルトでは `precompile.axd`) に対して HTTP 要求を出すことによって実行されます。 このページは、`web.config` で指定された特殊な HTTP ハンドラーによって処理されます。 このハンドラーは、`client.aspx` ファイルを含めたサイト全体をコンパイルして、.NET フレームワーク・ディレクトリー内の ASP.NET 一時ファイル・ディレクトリーに入れます。そこでファイルはすべてスキャンされます。

ASP.NET 1.1 をスキャンするには、Web サイトでデバッグ情報をコンパイルおよびビルドするように、その Web サイトを調整する必要があります。それ以降、Web サイトがデバッグ情報をコンパイルおよびビルドするということ自体が、セキュリティー上の脆弱性となります。 スキャンでこれを必要とするため、この脆弱性を無視しても支障はありません。 ただし、デプロイされたアプリケーションが、`web.config` で `debug=true` と指定してコンパイルされていないことを確認してください。

ASP.NET 1.1 Web サイトをプリコンパイルするには、このエレメントを、ご使用の `web.config` ファイル内の `<system.web>` エレメントに子として追加します。

```
<httpHandlers><add verb="*" path="precompile.axd"
type="System.Web.Handlers.BatchHandler"/></httpHandlers>
```

また、コンパイル・エレメントで `debug=true` と設定する必要があります。例:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
 <system.web>
  <httpHandlers><add verb="*" path="precompile.axd"
  type="System.Web.Handlers.BatchHandler"/>
  </httpHandlers>
   <compilation
```

```
    defaultLanguage="c#"
    debug="true"
   />
...
```

このエレメントは、precompile.axd ページが .Net の特殊な System.Web.Handlers.BatchHandler クラスによって処理されることを、Web サイトに対して指定します。このクラスは、Web サイトのコンテンツをプリコンパイルして、ASP.NET 一時ファイル・ディレクトリーに入れます。

- Web サイト: サイトをプリコンパイルするようにターゲットに要求します。 デフォルトの位置は、precompile.axd です。 precompile.axd は仮想ファイルであり、web.config ファイルで指定されたファイルにマップします。
- 出力ディレクトリー: プリコンパイルのターゲットとなるディレクトリー。 AppScan Source は、このディレクトリーでプリコンパイルの出力を特定します。
- ASP.NET Web サイトのプリコンパイル: AppScan Source は自動的にプリコンパイルを行い、スキャン時には、プリコンパイルされた出力をスキャンします。
- プリコンパイルが失敗した場合はスキャンを停止: 「**ASP.NET Web** サイトのプリコンパイル」および「プリコンパイルが失敗した場合はスキャンを停止」を選択すると、プリコンパイルが失敗した場合にスキャンが停止されます。 そのようにしない場合は、Web サイトのプライマリー出力のみでスキャンが続行されます。
- 直ちにコンパイル: スキャンの前に、現在の設定に基づくプリコンパイルが正常に実行されることを調べるテストを行います。 コンパイルの出力は、「プリコンパイル出力」ペインに表示されます。
- 追加アセンブリー: 任意の .NET プロジェクト・タイプに対して、スキャンする追加アセンブリーを指定します。
- プロジェクト参照: .NET アセンブリー・プロジェクトと既存の .NET プロジェクトで、参照されるアセンブリーを検索するディレクトリーをリストします。

第 11 章 アプリケーション・サーバーのインポート・フレームワークの拡張

AppScan Source では、Apache Tomcat および WebSphere Application Server Liberty プロファイルから Java アプリケーションをインポートできます。このトピックで説明するように、アプリケーション・サーバーのインポート・フレームワークを拡張することにより、他のアプリケーションから Java アプリケーションをインポートできます。

このタスクについて

アプリケーション・サーバーのインポート・フレームワークには、PDF では使用できない付属 API 文書が含まれています。Adobe PDF を使用してこのヘルプ・トピックにアクセスしている場合、AppScan Source for Analysis オンライン・ヘルプを起動して、「プロダクト機能の拡張」 > 「アプリケーション・サーバーのインポート・フレームワークの拡張」 > 「アプリケーション・サーバーのインポート拡張 **API** のクラスおよびメソッド」にナビゲートするか、http://www.ibm.com/support/knowledgecenter/SSS9LM/welcome でヘルプのそのセクションを見つけることでのみ、この API 文書にアクセスできます。

アプリケーション・サーバーのインポート・フレームワークを拡張するには、以下のステップを実行します。これらのステップでは、以下を実行します。

- Eclipse 統合開発環境の構成
- Eclipse での新規プラグインの作成
- 新規作成されたプラグインでの必要な依存関係の設定
- プラグインでのアプリケーション・サーバーの拡張の定義
- プラグインのテスト
- AppScan Source for Analysis に対するプラグインの有効化

手順

1. AppScan Source アプリケーション・サーバーのインポート・フレームワークに必要な依存関係のために Eclipse 統合開発環境を構成します。
   a. Eclipse で、メインメニューから「ウィンドウ」 > 「設定」を選択します。
   b. 「設定」ダイアログ・ボックスで、「プラグイン開発」を展開してから、「ターゲット・プラットフォーム」を選択します。
   c. 「ターゲット・プラットフォーム」設定ページで、「追加」をクリックして、新しいターゲット定義を作成します。
   d. 「ターゲット定義」ウィザード・ページで、「何もしない**:** 空のターゲット定義を使用して開始」を選択してから、「次へ」を押します。
   e. 「ターゲット・コンテンツ」ウィザード・ページで、「名前」フィールドにターゲットの名前を入力して、「追加」をクリックし、AppScan Source インストール・ディレクトリーを追加します (360 ページの『インストールとユーザー・データ・ファイルの場所』を参照)。

f. オプション: 「ロケーション・コンテンツの表示」を選択して、そのプラグインが使用可能であることを確認します。

g. 「完了」をクリックします。

h. 「ターゲット・プラットフォーム」設定ページで、先ほど作成したターゲット・プラットフォームを選択して、「適用」を押します。次に、「**OK**」を押します。

2. Eclipse で新規プラグインを作成します。

a. メインメニューから「ファイル」 > 「新規プロジェクト」を選択して、新規プロジェクト・ウィザードを開きます。

b. 「ウィザードを選択」ページで、「プラグイン・プロジェクト」を選択してから、「次へ」を押します。

c. 「プラグイン・プロジェクト」ページで、「プロジェクト名」フィールドにプラグインの名前 (このヘルプ・トピックでは、`com.example.appserverimporter` を使用します) を入力して、「次へ」を押します。

d. 「コンテンツ」ページで、「アクティベーター **(**プラグインのライフサイクルを制御する **Java** クラス**)** を生成」を選択解除して、「終了」を押します。

3. 先ほど作成したプラグインで、必要な依存関係を設定します。

a. `META-INF¥MANIFEST.MF` を開き、「依存関係」タブを選択します。

b. エディターの「必須プラグイン」セクションで、以下を行います。

- 「追加」をクリックしてから、`com.ouncelabs.core.appserverimporter` および `org.eclipse.core.runtime` を追加します。
- 追加した `com.ouncelabs.core.appserverimporter` プラグインを選択し、「プロパティー」をクリックします。 プラグイン・プロパティーで、「最小バージョン」フィールドと「最大バージョン」フィールドの項目を削除し、「**OK**」をクリックします。
- `org.eclipse.core.runtime` プラグインに対して上記ステップを繰り返します。

c. メインメニューから「ファイル」 > 「保存」を選択して、エディターに対して行われたすべての変更を保存します。

d. 次のステップでは、アプリケーション・サーバーに対する拡張を定義します。このステップのために、`META-INF¥MANIFEST.MF` エディターで作業を続けます。

4. 以下のステップを実行して、アプリケーション・サーバー用にインポーター拡張を定義します。

a. 「拡張」タブを選択して、「追加」をクリックし、`com.ouncelabs.appserver` を追加してから、メインメニューで「ファイル」 > 「保存」を選択します。

b. 「**plugin.xml**」タブを選択します。次のような内容が表示されます。

```
<?xml version="1.0" encoding="UTF-8"?>
<?eclipse version="3.4"?>
<plugin>
```

```
  <extension
    point="com.ouncelabs.appserver">
  </extension>
</plugin>
```

これを編集して、拡張の定義を完了します。例:

```
<?xml version="1.0" encoding="UTF-8"?>
<?eclipse version="3.4"?>
<plugin>
  <extension
    point="com.ouncelabs.appserver">
    <importer
     class="com.example.appserverimporter.MyAppServerImporter"
     id="com.example.appserverimporter.myappserver"
     name="My App Server">
   </importer> </extension>
</plugin>
```

c. メインメニューから「ファイル」 > 「保存」を選択して、**plugin.xml** に対する変更を保存します。

5. インポーター・クラス (この例では、com.example.appserverimporter.MyAppServerImporter) を作成して、新しいアプリケーション・サーバー・インポーターの動作を定義します。このクラスは、BaseAppServerImporter (フレームワークの基本実装) を AppServerImporter インターフェース向けに拡張する必要があります。このクラスで以下を行います。

   a. AppServerImporter.importAppServer(String) を実装します。これは、フレームワークにより、インポートする Java EE プロジェクトとそれらの場所を判別するために使用されます。通常、各プロジェクトには Java EE プロジェクトの名前とパスのみが必要です。 EAR プロジェクトが作成される場合、その中に含まれている Java EE プロジェクトは、AppScan Source ユーザー・インターフェースでプロジェクトを選択するときに非表示になります。この場合は、EAR 全体がインポートされます。そうでない場合、すべてのプロジェクトは、個別に選択できるようにリストされます。

      該当する場合、以下のメソッドの使用が強く推奨されます。

      - BaseAppServerImporter.processDropInsFolder(AppServerProfile, File)
      - BaseAppServerImporter.processEARFile(AppServerProfile, File)

   b. AppServerImporter.isValidLocation(String) を実装します。これは、インストール・ディレクトリーを指定してサーバーのタイプを検出するために使用されます。

   c. オプション: BaseAppServerImporter.getJSPCompilerType() をオーバーライドします。このメソッドは、AppScan Source プロジェクトに使用される JSP コンパイラーを返します。これが実行されない場合、基本実装は NULL を返し、製品のデフォルトの JSP コンパイラーが使用されます。

6. オプション: 拡張オプションとして、プリコンパイル済み JSP コンパイラーを使用するように JSP コンパイルをカスタマイズできます (JSP コンパイルはインポート前またはインポート中に行われます)。

   a. JSPCompilerType.PRECOMPILED を返すように BaseAppServerImporter.getJSPCompilerType() をオーバーライドします。

b. JMX、Java API、外部スクリプトを呼び出して JSP ファイルをコンパイルするか、単にクラス・ファイルを AppScan Source プロジェクトのステージング・ディレクトリーにコピーするように、`BaseAppServerImporter.getJSPCompilerType()` をオーバーライドします。ステージング・ディレクトリーを取得するには、`Application.getStagingDirectory(Project)` を使用します。

c. `JSPCompilerSupport` のカスタム拡張を返すように、`BaseAppServerImporter.createJSPCompilerSupport()` をオーバーライドします。これは、JSP ファイルと生成されたクラス・ファイルとの間のマッピングを保持して、JSP コンパイルの後に検証するために使用されます。

d. `AppServerClasspathProvider` のカスタム実装を返すように、`BaseAppServerImporter.createClasspathProvider()` をオーバーライドします。このクラスは、サーバー・ライブラリーに対する依存関係を持つ Java または JSP ファイルをコンパイルするために必要です。このクラスは、`BaseAppServerClasspathProvider` を拡張する必要があります。`getClasspathEntries()` が呼び出されるときに `BaseAppServerClasspathProvider.installDirectory` が既にアプリケーション・サーバーのインストール・ディレクトリーに設定されていることに注意してください。

7. 以下のステップを実行して、プラグインをテストします。

a. メインメニューから「実行」 > 「構成の実行」を選択します (デバッグ・モードでテストする場合は、「実行」 > 「デバッグ」を選択します)。

b. 新しい「**Eclipse** アプリケーション」構成を作成します。

- 新規構成の「メイン」タブに進みます。「実行するプログラム」セクションで、「プロダクトの実行」を選択して、**com.ouncelabs.osa.rcp.product** を実行するように設定します。
- 「引数」タブに進みます。「作業ディレクトリー」セクションで、「その他」を選択して、フィールドに AppScan Source データ・ディレクトリーを入力します (360 ページの『インストールとユーザー・データ・ファイルの場所』を参照)。
- 「プラグイン」タブで、「起動対象」の選択項目を「以下で選択したプラグインのみ」に設定します。「ワークスペース」を展開して、作成したプラグインが選択されていることを確認してから、「ターゲット・プラットフォーム」でそれらのプラグインを選択解除します。
  - `com.ouncelabs.plugin.base`
  - `com.ouncelabs.plugin.base`
  - `com.ouncelabs.plugin.base.nl`
  - `com.ouncelabs.plugin.base.nl`
  - `com.ouncelabs.plugin.enhanced`
  - `com.ouncelabs.plugin.enhanced`
  - `com.ouncelabs.plugin.enhanced.nl`
  - `com.ouncelabs.plugin.enhanced.nl`

c. 「構成の実行」ダイアログ・ボックスで「実行」をクリックする前に、AppScan Source インストール・ディレクトリーに移動して、bin¥OunceScanner.exe を実行します。

d. 「構成の実行」ダイアログ・ボックスに戻り、「実行」をクリックし、AppScan Source for Analysis を起動して、プラグインをテストします。

8. 以下のステップを実行して、AppScan Source for Analysis に対してプラグインを有効にします。

a. プロジェクトを右クリックして、「エクスポート」を選択します。

b. エクスポート・ウィザードの「選択」ページで、「プラグイン開発」を展開して、「デプロイ可能なプラグインおよびフラグメント」を選択し、「次へ」をクリックします。

c. 「デプロイ可能なプラグインおよびフラグメント」ページで、以下のようにします。

- 「宛先」タブに進み、マシン上の一時ディレクトリーを参照して「ディレクトリー」を設定します。
- 「オプション」タブに進み、「個々の **JAR** アーカイブとしてプラグインをパッケージ」および「限定子の置換」を選択します。
- 「完了」をクリックします。

d. プラグインをエクスポートするための宛先として使用された一時ディレクトリーを見つけて、その中の plugins¥ フォルダーを開きます。このフォルダー内で、作成された .jar ファイルを見つけて <install_dir>¥dropins (<install_dir> は AppScan Source インストール済み環境がある場所です) にコピーします。

注:

- ¥dropins ディレクトリーが存在しない場合は、手動で作成する必要があります。
- AppScan Source インストール・ディレクトリーを変更するには、管理特権が必要になる可能性があります。

e. <install_dir>¥configuration¥org.eclipse.equinox.simpleconfigurator¥bundles.info を見つけます。このファイルのバックアップ・コピーを作成してから、ファイルを編集し、ファイルの終わりに以下を追加します。

```
<my_plugin>,<my_plugin_version>,
dropins/<my_plugin>_<my_plugin_version>.jar,4,false
```

ここで:

- <my_plugin> は、先ほど作成したプラグインの名前です。
- <my_plugin_version> は、作成したプラグインのバージョン番号です。

注: この項目の先頭で、<my_plugin>、<my_plugin_version>、および dropins/ の場所はコンマ (,) で区切られています。

f. AppScan Source for Analysis を開始します。

g. メインメニューから「ヘルプ」 > 「**AppScan Source for Analysis** について」を選択して、「インストールの詳細」をクリックします。「プラグイン」タブを選択して、プラグインがリストされていることを確認します。

h. 「インストールの詳細」ダイアログ・ボックスを閉じて、アプリケーション・サーバーのインポート・フレームワークの使用を開始します。

第 12 章 AppScan Source for Analysis サンプル

AppScan Source for Analysis には、サンプル・アプリケーションが含まれており、これを使用して製品に慣れることができます。

AppScan Source for Analysis がインストールされた後、サンプル・アプリケーションは <data_dir>¥samples (<data_dir> は、ご使用の AppScan Source プログラム・データの場所です。詳しくは、 360 ページの『インストールとユーザー・データ・ファイルの場所』 を参照してください。) に置かれています。

サンプル Java アプリケーション: simpleIOT

simpleIOT サンプルは、小さな Java アプリケーションで、さまざまなセキュリティー脆弱性が含まれています。手動で AppScan Source for Analysis ワークベンチにインポートすることも、サンプルに含まれるアプリケーション・ファイル (SimpleIOT.paf) またはプロジェクト・ファイル (SimpleIOT.ppf) をインポートすることもできます。アプリケーションおよびプロジェクトを追加する方法については、 35 ページの『第 2 章 アプリケーションおよびプロジェクトの構成』を参照してください。

サンプルを AppScan Source に追加した後、検出結果をスキャンして探索できます。

Framework for Frameworks 処理 API の使用法を確認するためのサンプル・アプリケーション: F4FEjbExample.zip

このプロジェクト・アーカイブ例は、Framework for Frameworks 処理 API の使用例を示すために使用されます。詳しくは、*IBM Security AppScan Source Utilities* ユーザー・ガイド を参照してください。

第 13 章 AppScan Source for Analysis の作業環境

AppScan Source を最大限に活用するには、AppScan Source for Analysis の作業環境の基本概念と、現在のワークフローに最適なオプションの使用方法について理解する必要があります。

AppScan Source for Analysis のワークベンチ

AppScan Source for Analysis のワークフローは、ワークベンチ 内で発生します。ワークベンチは、パースペクティブ、ビュー、およびエディターで構成され、これらの構成要素はコンテキストに応じて表示または非表示になります。

パースペクティブ

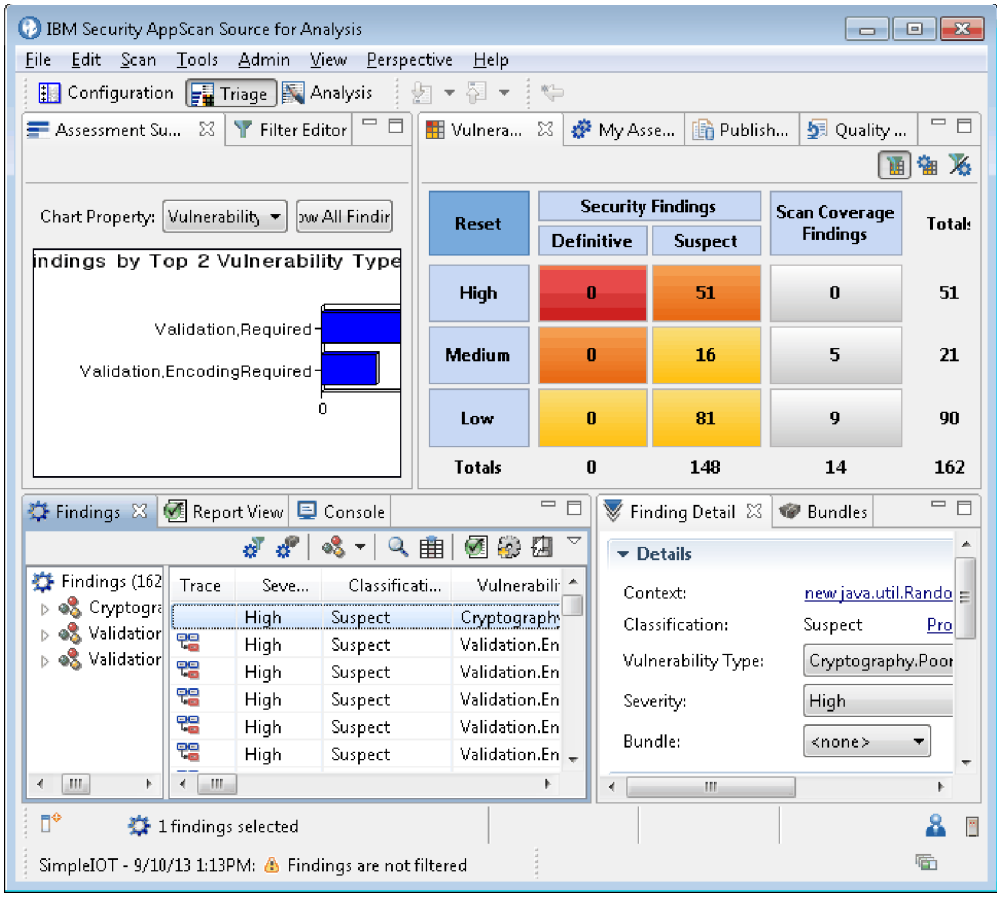
製品の 3 つのパースペクティブ (構成、トリアージ、分析) は、それぞれ複数のビューで構成されています。それぞれのパースペクティブを開くとデフォルトのビューが表示されますが、ビューを再編成して各パースペクティブをカスタマイズすることができます。ビューの詳しい説明は、ヘルプの 311 ページの『第 14 章 ビュー』のセクションを参照してください。

- 「構成」パースペクティブ: アプリケーション、プロジェクト、属性の作成と管理を行います。
- 「トリアージ」パースペクティブ: スキャンの結果を表示して修復ワークフローの優先順位付けを行い、実際の脆弱性と潜在的な脆弱性を区別します。 このパースペクティブを使用して、最初に修正する必要がある問題を特定します。
- 「分析」パースペクティブ: 各検出結果へのドリルダウンや、ソース・コード、修復のアドバイス、および AppScan Source トレース情報のレビューを行います。

ワークベンチ・ウィンドウ

AppScan Source for Analysis ワークベンチ・ウィンドウは、以下の要素で構成されています。

- メインメニュー: AppScan Source for Analysis の機能にアクセスするためのメニューです。
- ツールバー: よく使用する機能のアイコンとボタンが表示されます。
- パースペクティブ: ビューの集合です。
- ビュー: ワークベンチ内の情報をナビゲートするための表示様式です。



ワークベンチの下部にあるツールバーと情報

- 「高速ビュー **(Fast View)**」ツールバー: 高速ビューは、素早く開いたり閉じたりできる非表示のビューです。ワークベンチ・ウィンドウのスペースを占有しないことを除いては、他のビューと同様に機能します。高速ビューは、「高速ビュー (fast view)」バーのツールバー・ボタンによって表されます。このバーは、ワークベンチ・ウィンドウの左下にあるツールバーです。高速ビューのツールバー・ボタンをクリックすると、現行パースペクティブ内でそのビューが一時的に開きます (パースペクティブ上でオーバーレイ表示)。そのビューの外側をクリックするか、ビューのフォーカスが失われると、ビューは再び非表示になります。ビューを高速ビューとして設定するには、「ビューを高速ビューとして表示 **(Show View as a Fast View)**」をクリックして、メニューからビューを選択します。
- 選択された検出結果: 検出結果が選択されると、ワークベンチの下部にある標識に、選択された検出結果の数が表示されます。
- ソース・ファイル情報: ソース・ファイルを開くと、このファイルに関する情報がワークベンチの下部に表示されます。
  - ファイルが書き込み可能か読み取り専用か。読み取り専用ファイルの編集を試みると、AppScan Source for Analysis のプロンプトでファイルを書き込み可能に設定できます。
  - オペレーティング・システムの入力モードが挿入または上書きのどちらか。
  - ファイル内の現行カーソル位置 (行番号と列番号)。

- サーバー接続情報: ユーザー・アイコンの上にカーソルを移動すると、AppScan Enterprise Server に現在ログインしているユーザーが示され、サーバー・アイコンの上にカーソルを移動すると、AppScan Source for Analysis が接続している AppScan Enterprise Server が表示されます。
- 評価が開いている場合は、ワークベンチの下部に次の情報が表示されます。
  - 評価の名前、および評価の作成日時。
  - 評価の検出結果にフィルターがどのように適用されたか素早く確認するために使用できる標識。詳しくは、 182 ページの『適用済みフィルターの判別』を参照してください。
- 進行中のアクションを示す進行標識もワークベンチの下部に表示されます。例えば、この標識はスキャンおよび評価のパブリッシュ時に表示されます。また、このセクションには評価がいつ開かれたかが示されます。

メインメニュー

メインメニュー・バーには、さまざまなアクションを実行できるメニューが含まれています。ユーザー特権に応じて、これらのメニューで使用可能なコマンドが制限されることがあります。

- 『「ファイル」メニュー』
- 304 ページの『「編集」メニュー』
- 306 ページの『「スキャン」メニュー』
- 306 ページの『「ツール」メニュー』
- 307 ページの『「管理」メニュー』
- 307 ページの『「表示」メニュー』
- 308 ページの『「パースペクティブ」メニュー』
- 308 ページの『「ヘルプ」メニュー』

「ファイル」メニュー

「ファイル」メニューには、アプリケーション、プロジェクト、評価に関するオプションが用意されています。また、このメニューで製品を終了することができます。「ファイル」メニュー項目の中には、コンテキストに依存するものがあります。このような項目は、アクティブになっているビュー、およびそのビューで現在選択されている項目によって表示内容が異なります。

表 *23.* 「ファイル」メニュー

メニュー項目	説明	キーボード・ショートカット
「アプリケーションの追加」 > 「新規アプリケーションの作成 **(Create a new application)**」	新規アプリケーションを一連のアプリケーションに追加します。このアクションによって、新規アプリケーション・ウィザードが起動します。	Ctrl+N

表 23. 「ファイル」メニュー (続き)

メニュー項目	説明	キーボード・ショートカット
「アプリケーションの追加」 > 「既存のアプリケーションを開く」	これにより「オープン」ダイアログ・ボックスが起動します。このダイアログ・ボックスで、既存のアプリケーションを参照して、一連のアプリケーションに追加できます。追加できるファイルまたはディレクトリーのタイプとしては、`.paf`、`.sln`、`.dsw`、および `.ewf` があります。	Ctrl+O
「アプリケーションの追加」 > 「既存の **Eclipse** ベースのワークスペースのインポート」	これにより「ワークスペースの追加」ダイアログ・ボックスが起動します。このダイアログ・ボックスで、Java プロジェクトが含まれている既存の Eclipse ワークスペースまたは IBM Rational Application Developer for WebSphere Software (RAD) ワークスペースを追加できます。ワークスペースのインポートが完了したら、そのワークスペースに含まれているすべての Java プロジェクトをスキャンできます。 **注:** ワークスペースをインポートする前に、 52 ページの『Eclipse プロジェクトおよび Rational Application Developer for WebSphere Software (RAD) プロジェクトの開発環境の構成』で説明されているように開発環境がインストールおよび更新されていることを確認してください。	
「アプリケーションの追加」 > 「アプリケーション・サーバーからのインポート」	Apache Tomcat または WebSphere Application Server Liberty アプリケーション・サーバーから既存の Java アプリケーションをインポートします。	

表 23. 「ファイル」メニュー (続き)

メニュー項目	説明	キーボード・ショートカット
「アプリケーションの追加」 > 「複数のアプリケーション」	複数のアプリケーションを一連のアプリケーションに追加します。このアクションは、アプリケーションを検索するディレクトリーを指定するダイアログ・ボックスを起動します。検索した結果、1 つ以上のアプリケーションを選択して追加できます。	
「アプリケーションの追加」 > 「アプリケーションのディスカバー」	これにより、アプリケーション・ディスカバリー・アシスタントが起動します。これを使用すると、Java および Microsoft Visual Studio ソース・コード用のアプリケーションおよびプロジェクトを迅速に作成および構成できます。	
アプリケーションの削除	このアクションは、「エクスプローラー」ビューでアプリケーションを選択している場合に使用可能です。これを選択すると、選択されているアプリケーションが削除されます。	
「プロジェクトの追加」 > 「新規プロジェクト」	このアクションは、「エクスプローラー」ビューでアプリケーションを選択している場合に使用可能です。これを選択すると、アプリケーションに新規プロジェクトを追加できます。このアクションによって、新規プロジェクト・ウィザードが起動します。	
「プロジェクトの追加」 > 「既存のプロジェクト」	このアクションは、「エクスプローラー」ビューでアプリケーションを選択している場合に使用可能です。これを選択すると、アプリケーションに既存のプロジェクトを追加できます。このアクションによりダイアログ・ボックスが起動します。このダイアログ・ボックスで、.ppf、.vcproj、.vcxproj、.csproj、.vbproj、.dsp、または .epf ファイルを参照して開くことができます。	

表 23. 「ファイル」メニュー (続き)

メニュー項目	説明	キーボード・ショートカット
「プロジェクトの追加」 > 「プロジェクトのコピー」	このアクションは、「エクスプローラー」ビューでプロジェクトを選択している場合に使用可能です。これを選択すると、ダイアログ・ボックスが開き、プロジェクトを別のアプリケーションにコピーしたり、現在プロジェクトが含まれているアプリケーションにそのプロジェクトのコピーを作成したりすることができます。	
「プロジェクトの追加」 > 「複数のプロジェクト」	「エクスプローラー」ビューで選択したアプリケーションに複数のプロジェクトを追加します。このアクションは、以下のいずれかのタスクを実行するダイアログ・ボックスを起動します。 • プロジェクトを検索するディレクトリーを指定する。 • プロジェクトを検索するワークスペースを指定する。 • プロジェクトを検索する Microsoft ソリューション・ファイルを指定する。 検索した結果、1 つ以上のプロジェクトを選択して追加できます。	
登録	選択したアプリケーションまたはプロジェクトを AppScan Source に登録します。アプリケーションおよびプロジェクトを AppScan Source データベース に公開するには、事前に登録しておく必要があります。	
登録抹消	選択したアプリケーションまたはプロジェクトの登録を抹消します。	

表 23. 「ファイル」メニュー (続き)

メニュー項目	説明	キーボード・ショートカット
評価を開く	これにより「オープン」ダイアログ・ボックスが起動します。そのダイアログ・ボックスで、AppScan Source 評価ファイルを参照できます。開くことができるファイルのタイプとしては、`.ozasmt` および `.xml` があります。	F7
評価を閉じる	「トリアージ」パースペクティブで現在開かれている評価を閉じます。	
評価の保存	開いている評価をファイルに保存します。	Ctrl+Shift+S
評価に名前を付けて保存	評価に別の名前を付けて保存するか、別のディレクトリーに保存するか、別の名前を付けて別のディレクトリーに保存します。	
AppScan Source に評価を公開	現在の評価を AppScan Source データベース に格納します。公開アクションを完了するには、その前に、スキャンされたアプリケーション (または、そのアプリケーションに含まれるプロジェクトまたはファイル) を登録する必要があります。 アプリケーションを登録していない場合、公開アクションの選択時に、登録するように求めるプロンプトが出されます。	
AppScan Enterprise Consoleに評価を公開	ご使用の AppScan Enterprise Server が Enterprise Console オプションを指定してインストールされている場合は、Enterprise Console に評価を公開することができます。 Enterprise Console に評価を公開するには、AppScan Enterprise Console 設定ページに有効な値を入力しておく必要があります。	

表 23. 「ファイル」メニュー (続き)

メニュー項目	説明	キーボード・ショートカット
保存	このアクションは、以下の場合に使用可能です。 • 「プロパティー」ビューで、アプリケーションのプロパティーが変更された場合。 • 「プロパティー」ビューで、プロジェクトのプロパティーが変更された場合。 • 内部エディターで開かれているファイルが変更された場合。 このアクションを選択して、これらの変更を保存します。	Ctrl+S
終了	AppScan Source for Analysis を終了します。	

注**:** AppScan Source for Analysis、AppScan Source for Automation、および AppScan Source コマンド行インターフェース でサポートされているインポート・ファイルのバージョンを確認するには、http://www.ibm.com/support/docview.wss?uid=swg27027486を参照してください。 このページで、使用している AppScan Source のバージョンのタブを選択してから、使用している AppScan Source コンポーネントを選択します。AppScan Source が他の開発環境からのファイルのオープンおよびスキャンをサポートする場合、そのサポートは、「**Supported Software**」タブの「**Compilers and Languages**」セクションにリストされています。

「編集」メニュー

このメニューで、標準的な変更および検索/置換を制御できます。また、このメニューを使用して、製品設定を起動することもできます。「編集」メニュー項目の中には、コンテキストに依存するものがあります。このような項目は、アクティブになっているビュー、およびそのビューで現在選択されている項目によって表示内容が異なります。

表 24. 「編集」メニュー

メニュー項目	説明	キーボード・ショートカット
切り取り	選択したテキストをコピーして削除します。このアクションは、コンソール、エディター、またはさまざまなテキスト・フィールドで選択したテキストに使用します。	Ctrl+X

表 24. 「編集」メニュー (続き)

メニュー項目	説明	キーボード・ショートカット
コピー	選択したテキストをクリップボードにコピーします。このアクションは、コンソール、エディター、またはさまざまなテキスト・フィールドで選択したテキストに使用します。	Ctrl+C
貼り付け	コピーされたテキストまたは切り取られたテキストを貼り付けます。このアクションは、通常、製品の別の部分に情報を複写および再作成する場合に使用します。	Ctrl+V
名前変更	選択したオブジェクトの名前を変更します。名前を変更できるオブジェクトとしては、アプリケーション、プロジェクト、評価、バンドルがあります。	F2
削除	選択したオブジェクトを除去します。	Delete
すべて選択	テキストの本文全体を選択します。このアクションは、コンソール、エディター、またはさまざまなテキスト・フィールドのテキストに使用します。	Ctrl+A
更新	選択したアプリケーション、プロジェクト、またはビューのコンテンツを更新します。	F5
検索	コンソール、エディター内のテキスト、または検出結果表内の検出結果を検索します。	Ctrl+F
次を検索	コンソールまたはエディターで「検索」アクションを使用してテキストを検索する場合に、このアクションを使用して、テキストの次のインスタンスを検索します。	F3
設定	これを選択すると、「設定」ダイアログ・ボックスが開きます。設定は、AppScan Source for Analysis の外観および操作についての個人の選択項目です。	

「スキャン」メニュー

「スキャン」メニューを使用して、選択したアプリケーション、プロジェクト、またはファイルのスキャンを管理します。

表 25. 「スキャン」メニュー

メニュー項目	説明	キーボード・ショートカット
すべてスキャン	すべてのアプリケーションをスキャンします。デフォルトのスキャン構成を使用してスキャンが実行されます。	
選択項目のスキャン	選択されたアプリケーション、プロジェクト、またはファイルをスキャンします。デフォルトのスキャン構成を使用してスキャンが実行されます。	F4
再スキャン	評価ターゲットを再スキャンします。 項目 (選択した項目) のスキャンに使用された最後のスキャン構成が、再度このスキャンに使用されます。	
スキャンのキャンセル	スキャンを強制終了します。スキャンの結果は生成されません。	
スキャンの停止	スキャンを一時停止します。スキャンの結果は部分的に生成されます。	
構成のビルド	プリプロセッサーの定義やインクルード・パスなどのプロジェクト・ビルド・パラメーターを定義します。 通常は、インポートされたプロジェクトの構成 (**Release** や **Debug** など) が表示されます。 このメニュー項目は、適用されない場合は無効になります。	

「ツール」メニュー

このメニューには、評価を比較したり、レポートを生成したりするためのオプションや、エディターでファイルまたは検出結果をレビューするためのオプションがあります。「ツール」メニュー項目の中には、コンテキストに依存するものがあります。このような項目は、アクティブになっているビュー、およびそのビューで現在選択されている項目によって表示内容が異なります。

表 26. 「ツール」メニュー

メニュー項目	説明
差分評価	このアクションは、比較する 2 つの評価を選択するダイアログ・ボックスを開きます。
検出結果レポートの生成	選択された検出結果またはバンドルの内容に関するレポートを生成します。このアクションを実行する場合は、「検出結果」ビューまたは「バンドル」ビューが選択されている必要があります。ビューで検出結果が選択されていない場合は、ビュー内のすべての検出結果がレポートに含まれます。
レポートの生成	特定の準拠要件またはガイドラインに基づいたすべての検出結果を表示するためのレポートを生成します。
内部エディターで開く	AppScan Source for Analysis の内部エディターでファイルを開きます。検出結果を選択して、このアクションを使用すると、その検出結果に関連したファイルがエディターで開かれます。
外部エディターで開く	外部エディターを使用してファイルを開きます。検出結果を選択して、このアクションを使用すると、その検出結果に関連したファイルがエディターで開かれます。

「管理」メニュー

「管理」メニューのアクションを使用して、ユーザーを管理したり、監査情報の表示画面を起動したりすることができます。

表 27. 「管理」メニュー

メニュー項目	説明
ユーザーの管理	このアクションは、ユーザーおよび許可の作成および編集を行うダイアログ・ボックスを起動します。 ユーザーを管理するには、AppScan Source の管理許可が必要です。
監査	このアクションは、認証イベントなどの監査情報を表示するビューを起動します。

管理作業について詳しくは、「*IBM Security AppScan Source* インストールと管理のガイド」を参照してください。

「表示」メニュー

「表示」メニューを使用して各ビューの表示を制御し、表示するビューを選択します。

AppScan Source for Analysis で使用可能なビューについて詳しくは、AppScan Source for Analysis のビューを参照してください。

「パースペクティブ」メニュー

「パースペクティブ」メニューは、AppScan Source for Analysis のパースペクティブ (ビューおよびオプションの事前構成済みコレクション) の表示を制御します。

表 28. 「パースペクティブ」メニュー

メニュー項目	説明	キーボード・ショートカット
構成	このパースペクティブで、アプリケーション、プロジェクト、および属性の作成と管理を行います。	Alt+1
トリアージ	このパースペクティブで、スキャンの結果を表示して修復ワークフローの優先順位付けを行い、実際の脆弱性と潜在的な脆弱性を区別します。このパースペクティブを使用して、最初に修正する必要がある問題を特定します。	Alt+2
分析	このパースペクティブで、各検出結果へのドリルダウンや、ソース・コード、修復のアドバイス、および AppScan Source トレース情報のレビューを行います。	Alt+3
パースペクティブのリセット	これを選択すると、現在表示されているパースペクティブを、デフォルトの表示およびレイアウトに戻します。	

「ヘルプ」メニュー

「ヘルプ」メニューには、製品の使用に役立つさまざまなツールを開くアクションが用意されています。これらのアクションにより、製品のようこそ画面、オンラインによるユーザー支援、および AppScan Source セキュリティー・ナレッジ・データベースなどが開きます。

表 29. 「ヘルプ」メニュー

メニュー項目	説明
ようこそ	これを選択すると、AppScan Source for Analysis の「ようこそ」ビューが開きます。このビューには、X-Force RSS フィードを含むさまざまなヘルプ・リソースへのクイック・リンクが表示されます。
ヘルプ目次	これを選択すると、AppScan Source for Analysis 製品のユーザー支援が開きます。

表 29. 「ヘルプ」メニュー (続き)

メニュー項目	説明
セキュリティー・ナレッジ・データベース	このアクションにより、AppScan Source セキュリティー・ナレッジ・データベースが開きます。ナレッジベース・データベースは各脆弱性についての情報を提供します。脆弱性の根本原因、リスクの重大度、実施可能な修復アドバイスに関する適格な説明を提供します。
ログ	これを選択すると、「ログ」ビューが開きます。このビューのタブで、表示するログ・ファイルを選択できます。
バージョン情報**IBM Security AppScan Source for Analysis**	これを選択すると、AppScan Source for Analysis に関する製品情報を示すダイアログ・ボックスが開きます。

ツールバー

AppScan Source for Analysis ワークベンチのツールバーには、コマンドへのグラフィック・ショートカットが表示されます。各ツールバー・アイコンの内容を確認するには、マウスをアイコンの上で停止させると、吹き出しヘルプが表示されます。よく使用される操作が、ツールバーのボタンとして用意されています (これらの操作はメインメニューにも表示されます)。ツールバーに表示される操作は、コンテキストに依存します。

メイン・ツールバーには、AppScan Source for Analysis パースペクティブへのクイック・リンクが表示されます。また、ほとんどのビューには、ビューに関連する共通アクションを簡単に起動するためのツールバーがあります。

吹き出しヘルプ

吹き出しヘルプは、マウス・ポインターをインターフェース・エレメントの上に移動すると、小さいポップアップ・ウィンドウに表示される形式のコンテキスト・センシティブ・ヘルプです。インターフェース・エレメントの要旨が、ポップアップ・ウィンドウに表示されます。

AppScan Source for Analysis では、ボタンおよびアイコンの吹き出しヘルプ以外にも、以下のようなさまざまな場所で吹き出しヘルプが用意されています。

- 「エクスプローラー」ビューで、吹き出しヘルプに、アプリケーション、プロジェクト、およびファイルのファイル名とパスが表示されます。吹き出しヘルプには、アプリケーションまたはプロジェクトが登録されているかどうかも示されます。
- 「トレース」ビューで、グラフ内のトレース・ノードの上にマウスを移動すると、そのノードに関する情報が表示されます。
- 「フィルター・エディター」ビューの「トレース」セクションで、トレース項目の上にマウスを移動すると、項目に関する詳細が表示されます。

- 「スキャン構成」ビューの「詳細設定」セクションでは、各設定について吹き出しヘルプが表示されます。
- 「評価の概要」ビューの棒グラフの上にマウスを移動すると、棒グラフで表されている検出結果の正確な数が表示されます。
- ワークベンチのステータス・バー (ワークベンチ下部にあります) で、ユーザー・アイコンの上にマウスを移動すると、ログオン・ユーザーを識別する吹き出しヘルプが起動されます。サーバー・アイコンの上にマウスを移動すると、AppScan Source for Analysis が接続されている Enterprise Server を示す吹き出しヘルプが起動されます。

ステータス・バー

ワークベンチ下部のステータス・バーには、スキャンなどの現在のアクションを示す通知メッセージが表示されます。

例えば、スキャンの実行中、ステータス・バーには、進行標識とともに**「<プロジェクト名> のスキャン中」**と表示されます。また、スキャンの現在のステージも表示されます。例えば、**「脆弱性分析の準備中: 99%」**のように表示されます。スキャンが完了すると、ステータス・バーに経過時間が表示されます。

ステータス・バーには、現在のユーザーおよびサーバー接続に関する情報も示されます。ユーザー・アイコンの上にマウスを移動すると、ログオン・ユーザーを識別する吹き出しヘルプが起動されます。サーバー・アイコンの上にマウスを移動すると、AppScan Source for Analysis が接続されている Enterprise Server を示す吹き出しヘルプが起動されます。

第 14 章 ビュー

AppScan Source for Analysis 作業環境は、異なる評価またはスキャン・データが含まれている複数のパースペクティブおよびビューで構成されています。

AppScan Source for Analysis の各ビューは、検出結果をそれぞれ別の表示形式で提示します (一部のビューでは、コード編集がサポートされています)。これらのビューを使用して、ワークベンチ内で情報をナビゲートすることができます。例えば、「エクスプローラー」ビューには、アプリケーション、プロジェクト、およびその他のリソースが表示されます。ビューは単独で表示される場合も、タブ付きのノートブック形式で他のビューと重なって表示される場合もあります。 ビューを開いたり閉じたり、ワークベンチのウィンドウ内のさまざまな位置で連結したりすることにより、パースペクティブのレイアウトを変更できます。

ビューについては、以下のセクションで詳しく説明しています。

- 『構成ビュー』
- 336 ページの『スキャン出力に役立つビュー』
- 339 ページの『トリアージに役立つビュー』
- 350 ページの『単一の検出結果の調査に使用できるビュー』
- 355 ページの『評価の操作に使用できるビュー』
- 359 ページの『「バンドル」ビュー』

構成ビュー

このセクションのビューは、AppScan Source を構成するために使用されます。

- 『「カスタム・ルール」ビュー』
- 96 ページの『「エクスプローラー」ビュー』
- 318 ページの『「パターン・ルール・ライブラリー」ビュー』
- 319 ページの『「プロパティー」ビュー』
- 129 ページの『「スキャン構成」ビュー』
- 251 ページの『レポート・エディター』

「カスタム・ルール」ビュー

「カスタム・ルール」ビューでは、カスタム・ルール・ウィザードを使用してカスタム・ルールを作成します。 既存のルールを追加、表示、または削除します。

詳しくは、 260 ページの『カスタム・ルールの作成』を参照してください。

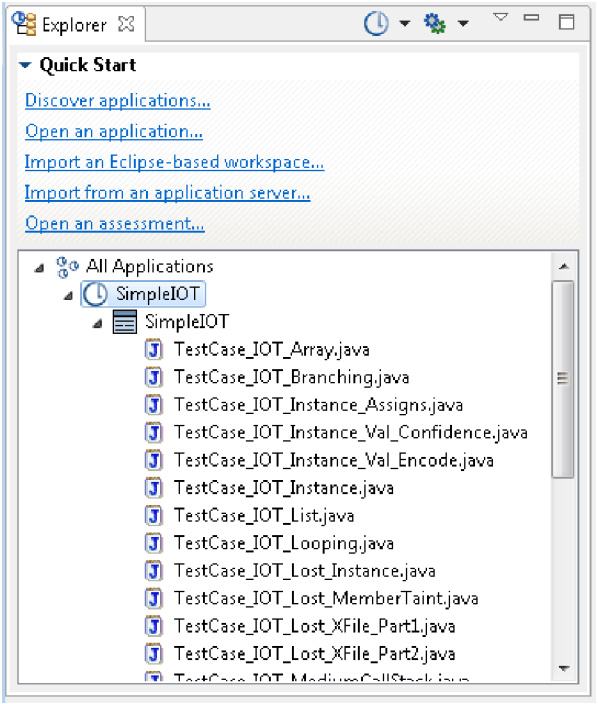
「エクスプローラー」ビュー

「エクスプローラー」ビューには、上部に「クイック・スタート」セクションがあり、下部に「エクスプローラー」セクションがあります。「エクスプローラー」セクションには、「すべてのアプリケーション」という 1 つのノードが含まれています。「クイック・スタート」セクションには、共通のアクションを起動するいくつ

かの便利なリンクが含まれています。 「エクスプローラー」セクションは、「すべてのアプリケーション」をルートとして、ご使用のリソース (アプリケーション、プロジェクト、ディレクトリー、およびプロジェクト・ファイル) を階層的に表示するツリー・ペインで構成されています。 ファイル・ブラウザーとほぼ同じようにして、これらのリソースをナビゲートします。 このビューをナビゲートするとき、このツリーの選択状態によって、「プロパティー」ビューで使用可能なタブが決まります。

- 96 ページの『全般情報』
- 97 ページの『「クイック・スタート」セクション』
- 97 ページの『ツールバー・ボタン』
- 98 ページの『右クリックのメニュー・オプション』
- 101 ページの『アプリケーションおよびプロジェクトのインディケーター』

全般情報



「エクスプローラー」ビューでは、アプリケーションおよびプロジェクトを追加し、ツールバーのボタン、「クイック・スタート」セクションのリンク、および「エクスプローラー」セクションの右クリック・メニュー・コマンドを使用してコードをスキャンします。 アプリケーションを追加したら、「エクスプローラー」セクションに、アプリケーションおよびプロジェクトのビジュアル・インディケーターと、それぞれの状態が表示されます。

ヒント: 「エクスプローラー」ビューで、吹き出しヘルプに、アプリケーション、プロジェクト、およびファイルのファイル名とパスが表示されます。吹き出しヘルプには、アプリケーションまたはプロジェクトが登録されているかどうかも示されます。

「クイック・スタート」セクション

「クイック・スタート」セクションには、共通タスクを起動するための以下のリンクがあります。

- アプリケーションのディスカバー: これにより、アプリケーション・ディスカバリー・アシスタントが起動します。これを使用すると、Java および Microsoft Visual Studio ソース・コード用のアプリケーションおよびプロジェクトを迅速に作成および構成できます。
- アプリケーションを開く: これにより「オープン」ダイアログ・ボックスが起動します。このダイアログ・ボックスで、既存のアプリケーションを参照して、一連のアプリケーションに追加できます。追加できるファイルまたはディレクトリーのタイプとしては、`.paf`、`.sln`、`.dsw`、および `.ewf` があります。
- **Eclipse** ベースのワークスペースのインポート: これにより「ワークスペースの追加」ダイアログ・ボックスが起動します。このダイアログ・ボックスで、Java プロジェクトが含まれている既存の Eclipse ワークスペースまたは IBM Rational Application Developer for WebSphere Software (RAD) ワークスペースを追加できます。ワークスペースのインポートが完了したら、そのワークスペースに含まれているすべての Java プロジェクトをスキャンできます。

  注: ワークスペースをインポートする前に、 52 ページの『Eclipse プロジェクトおよび Rational Application Developer for WebSphere Software (RAD) プロジェクトの開発環境の構成』で説明されているように開発環境がインストールおよび更新されていることを確認してください。
- アプリケーション・サーバーからのインポート: Apache Tomcat または WebSphere Application Server Liberty アプリケーション・サーバーから既存の Java アプリケーションをインポートします。
- 評価を開く: これにより「オープン」ダイアログ・ボックスが起動します。そのダイアログ・ボックスで、AppScan Source 評価ファイルを参照できます。開くことができるファイルのタイプとしては、`.ozasmt` および `.xml` があります。

ツールバー・ボタン

表 *30.* ツールバー・ボタン

アクション	アイコン	説明
アプリケーション・メニューの追加		「アプリケーション・メニューの追加」ボタンの下矢印をクリックすると、新規アプリケーションの作成、既存のアプリケーションのオープン、ワークスペースのインポート、または アプリケーション・ディスカバリー・アシスタントの起動のためのアクションを選択できます。

表 30. ツールバー・ボタン (続き)

アクション	アイコン	説明
選択項目のスキャン		「選択項目のスキャン」ボタンを使用すると、「エクスプローラー」セクションで選択されるオブジェクトをスキャンできます。 スキャンには、デフォルトのスキャン構成が使用されます。別のスキャン構成を選択してスキャンに使用する場合は、「選択項目のスキャン」ボタンの下矢印をクリックします。使用するスキャン構成を選択します。あるいは、「構成の編集」アクションを選択して、別のスキャン構成をデフォルトとして設定します (「スキャン構成」ビューで、デフォルトとして設定する構成を選択し、「デフォルトとして選択」をクリックします)。
「表示」メニュー		「表示メニュー」ボタンは、「エクスプローラー」セクションを更新したり、登録済みの項目を非表示にしたりするためのメニューを開きます。

右クリックのメニュー・オプション

右クリックのメニュー・オプションが使用可能であるかどうかは、「エクスプローラー」セクションで選択されている項目によって決まります。

- 「エクスプローラー」セクションで「すべてのアプリケーション」が選択されている場合は、以下の右クリックのメニュー・オプションが使用可能になります。
  - すべてのアプリケーションのスキャン: すべてのアプリケーションをスキャンします。デフォルトのスキャン構成を使用してスキャンが実行されます。
  - すべてのアプリケーションのスキャン: 使用するスキャン構成を選択します。あるいは、「構成の編集」アクションを選択して、別のスキャン構成をデフォルトとして設定します (「スキャン構成」ビューで、デフォルトとして設定する構成を選択し、「デフォルトとして選択」をクリックします)。
  - アプリケーションの追加
    - 新規レポートの作成: 新規アプリケーションを一連のアプリケーションに追加します。このアクションによって、新規アプリケーション・ウィザードが起動します。
    - 既存のアプリケーションを開く: これにより「オープン」ダイアログ・ボックスが起動します。このダイアログ・ボックスで、既存のアプリケーシ

ョンを参照して、一連のアプリケーションに追加できます。追加できるファイルまたはディレクトリーのタイプとしては、.paf、.sln、.dsw、および .ewf があります。

- 既存の **Eclipse** ベースのワークスペースのインポート: これにより「ワークスペースの追加」ダイアログ・ボックスが起動します。このダイアログ・ボックスで、Java プロジェクトが含まれている既存の Eclipse ワークスペースまたは IBM Rational Application Developer for WebSphere Software (RAD) ワークスペースを追加できます。ワークスペースのインポートが完了したら、そのワークスペースに含まれているすべての Java プロジェクトをスキャンできます。

  注**:** ワークスペースをインポートする前に、 52 ページの『Eclipse プロジェクトおよび Rational Application Developer for WebSphere Software (RAD) プロジェクトの開発環境の構成』で説明されているように開発環境がインストールおよび更新されていることを確認してください。

- アプリケーションのディスカバー: これにより、アプリケーション・ディスカバリー・アシスタントが起動します。これを使用すると、Java および Microsoft Visual Studio ソース・コード用のアプリケーションおよびプロジェクトを迅速に作成および構成できます。

– すべて展開
– すべて縮小
– プロパティー: このオプションを選択すると、選択されている項目の「プロパティー」ビューが開きます。

• 「エクスプローラー」セクションでアプリケーションが選択されている場合は、以下の右クリックのメニュー・オプションが使用可能になります。
  – アプリケーションのスキャン: 選択されたアプリケーション、プロジェクト、またはファイルをスキャンします。デフォルトのスキャン構成を使用してスキャンが実行されます。
  – アプリケーションのスキャン: 使用するスキャン構成を選択します。あるいは、「構成の編集」アクションを選択して、別のスキャン構成をデフォルトとして設定します (「スキャン構成」ビューで、デフォルトとして設定する構成を選択し、「デフォルトとして選択」をクリックします)。
  – プロジェクトの追加
    - 新規プロジェクト: このアクションは、「エクスプローラー」ビューでアプリケーションを選択している場合に使用可能です。これを選択すると、アプリケーションに新規プロジェクトを追加できます。このアクションによって、新規プロジェクト・ウィザードが起動します。
    - 既存のプロジェクト: このアクションは、「エクスプローラー」ビューでアプリケーションを選択している場合に使用可能です。これを選択すると、アプリケーションに既存のプロジェクトを追加できます。このアクションによりダイアログ・ボックスが起動します。このダイアログ・ボックスで、.ppf、.vcproj、.vcxproj、.csproj、.vbproj、.dsp、または .epf ファイルを参照して開くことができます。
    - 複数のプロジェクト: 「エクスプローラー」ビューで選択したアプリケーションに複数のプロジェクトを追加します。このアクションは、以下のいずれかのタスクを実行するダイアログ・ボックスを起動します。

- プロジェクトを検索するディレクトリーを指定する。
- プロジェクトを検索するワークスペースを指定する。
- プロジェクトを検索する Microsoft ソリューション・ファイルを指定する。

検索した結果、1 つ以上のプロジェクトを選択して追加できます。

– アプリケーションの削除: このアクションは、「エクスプローラー」ビューでアプリケーションを選択している場合に使用可能です。これを選択すると、選択されているアプリケーションが削除されます。
– カスタム検出結果の追加: このアクションは、「カスタム検出結果の作成」ダイアログ・ボックスを起動します。このダイアログ・ボックスで、選択したアプリケーションのカスタム検出結果を作成することができます。
– 更新: 選択したアプリケーション、プロジェクト、またはビューのコンテンツを更新します。
– 登録/登録抹消:
  - アプリケーションの登録: 選択したアプリケーションまたはプロジェクトを AppScan Source に登録します。アプリケーションおよびプロジェクトを AppScan Source データベース に公開するには、事前に登録しておく必要があります。
  - アプリケーションに名前を付けて登録...: 新しい名前でアプリケーションを再登録する場合は、このオプションを選択します。
  - アプリケーションの登録抹消: 選択したアプリケーションまたはプロジェクトの登録を抹消します。
  - 位置指定: ローカルのアプリケーション/プロジェクトを、別の AppScan Source ユーザーが登録したアプリケーション/プロジェクトに関連付ける場合は、このオプションを選択します。
– すべて展開
– すべて縮小
– プロパティー: このオプションを選択すると、選択されている項目の「プロパティー」ビューが開きます。

- 「エクスプローラー」セクションでプロジェクトが選択されている場合は、以下の右クリックのメニュー・オプションが使用可能になります。
  – プロジェクトのスキャン: 選択されたアプリケーション、プロジェクト、またはファイルをスキャンします。デフォルトのスキャン構成を使用してスキャンが実行されます。
  – プロジェクトのスキャン: 使用するスキャン構成を選択します。あるいは、「構成の編集」アクションを選択して、別のスキャン構成をデフォルトとして設定します (「スキャン構成」ビューで、デフォルトとして設定する構成を選択し、「デフォルトとして選択」をクリックします)。
  – プロジェクトのコピー: このアクションは、「エクスプローラー」ビューでプロジェクトを選択している場合に使用可能です。これを選択すると、ダイアログ・ボックスが開き、プロジェクトを別のアプリケーションにコピーしたり、現在プロジェクトが含まれているアプリケーションにそのプロジェクトのコピーを作成したりすることができます。

- プロジェクトの削除: 選択したオブジェクトを除去します。
- 登録/登録抹消:
  - プロジェクトの登録: 選択したアプリケーションまたはプロジェクトを AppScan Source に登録します。アプリケーションおよびプロジェクトを AppScan Source データベース に公開するには、事前に登録しておく必要があります。
  - プロジェクトの登録抹消: 選択したアプリケーションまたはプロジェクトの登録を抹消します。
  - 位置指定: ローカルのアプリケーション/プロジェクトを、別の AppScan Source ユーザーが登録したアプリケーション/プロジェクトに関連付ける場合は、このオプションを選択します。
- すべて展開
- すべて縮小
- プロパティー: このオプションを選択すると、選択されている項目の「プロパティー」ビューが開きます。

- 「エクスプローラー」セクションでファイルが選択されている場合は、以下の右クリックのメニュー・オプションが使用可能になります。
  - ファイルのスキャン: 選択されたアプリケーション、プロジェクト、またはファイルをスキャンします。デフォルトのスキャン構成を使用してスキャンが実行されます。
  - ファイルのスキャン: 使用するスキャン構成を選択します。あるいは、「構成の編集」アクションを選択して、別のスキャン構成をデフォルトとして設定します (「スキャン構成」ビューで、デフォルトとして設定する構成を選択し、「デフォルトとして選択」をクリックします)。
  - スキャンから除外: 選択されているファイルをスキャンから除外します。
  - 内部エディターで開く: 選択したファイルを AppScan Source エディター (「分析」パースペクティブ内) で開きます。
  - 「外部エディターで開く」:選択したファイルを開く外部エディターを選択します。
  - プロパティー: このオプションを選択すると、選択されている項目の「プロパティー」ビューが開きます。

アプリケーションおよびプロジェクトのインディケーター

次の表に、「エクスプローラー」ビューでのアプリケーションおよびプロジェクトのアイコンを示します。

表 *31.* アプリケーションおよびプロジェクトのアイコン

アプリケーションまたはプロジェクトのタイプ	未登録	登録済み	存在しない/未検出
インポートされたアプリケーション			

表 31. アプリケーションおよびプロジェクトのアイコン (続き)

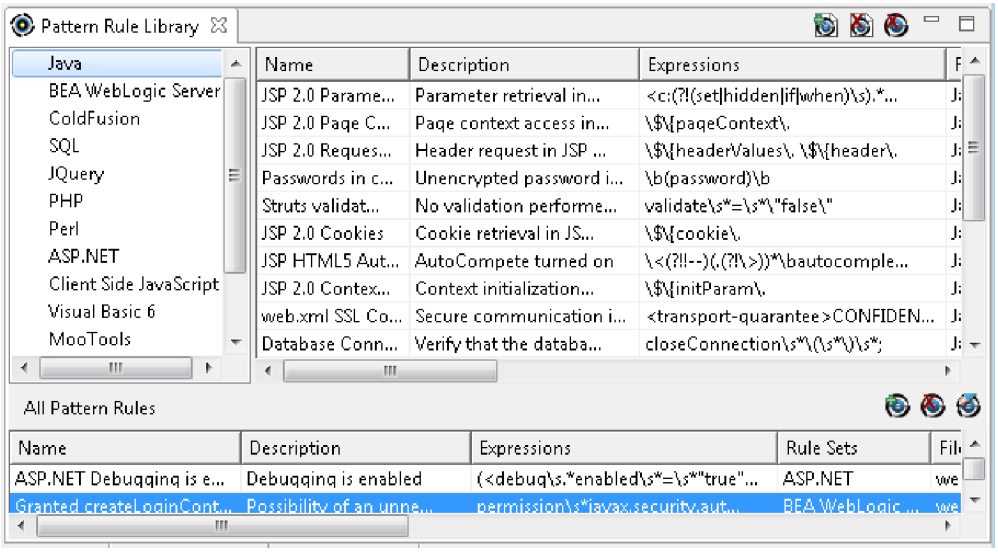
アプリケーションまたはプロジェクトのタイプ	未登録	登録済み	存在しない/未検出
手動で作成されたか、または アプリケーション・ディスカバリー・アシスタントを使用して作成されたアプリケーション			
インポートされたプロジェクト			
手動で作成されたか、または アプリケーション・ディスカバリー・アシスタントを使用して作成されたプロジェクト			

「エクスプローラー」ビューには、ローカル・アプリケーションおよびローカル・プロジェクトだけでなく、サーバーに登録されているアプリケーションおよびプロジェクトも表示されます (例えば、他のユーザーが登録したアプリケーションおよびプロジェクトなど、サーバーに登録されていてもローカルに保存されていないアプリケーションおよびプロジェクトはグレー表示されています)。 ツールバーの「表示メニュー」ボタンをクリックし、「サーバーに登録されている項目の非表示」メニュー項目を切り替えて選択解除すると、既存のサーバー・アプリケーションおよびプロジェクトを表示できます。プロジェクトがグレー表示されている場合は、右クリックして、メニューの「位置指定」を選択できます。

「パターン・ルール・ライブラリー」ビュー

パターン・ベースのスキャンは、カスタマイズされた検索基準に基づいてソース・コードの分析を行う機能です。「パターン・ルール・ライブラリー」ビューを使用すると、既存のパターン・ベースのルールを言語別に表示できます (すぐに使用できる AppScan Source パターン・ルール・ライブラリーも含まれています)。また、このビューで、パターン・ベースのスキャンに使用するルールやパターンを追加できます。

ルール・ライブラリーをビルドすると、特定のアプリケーションまたはプロジェクトにパターン分析を適用できます。パターン検索について詳しくは、 268 ページの『パターン・ベースのルールによるカスタマイズ』を参照してください。



「プロパティー」ビュー

「プロパティー」ビューのコンテンツは、「エクスプローラー」ビューで選択されている項目によって決まります。プロパティーは、すべてのアプリケーション、個別のアプリケーション、プロジェクト、またはファイルに適用されます。 表示されるプロパティーは、言語または選択されたプロジェクト・タイプによって異なります。

- 『「プロパティー」ビュー: すべてのアプリケーション』
- 279 ページの『「プロパティー」ビュー: 選択したアプリケーション』
- 280 ページの『「プロパティー」ビュー: 選択したプロジェクト』
- 329 ページの『ファイル・プロパティー』

「プロパティー」ビュー: すべてのアプリケーション

「エクスプローラー」ビューで「すべてのアプリケーション」を選択した場合、「プロパティー」ビューには「概要」タブおよび「フィルター」タブが表示されます。

概要

「概要」タブには、グローバル属性が表示されます。 属性 は、類似した特性を持つユーザー定義項目の名前付きグループです。 属性とその値は、自分で追加または削除します。

フィルター

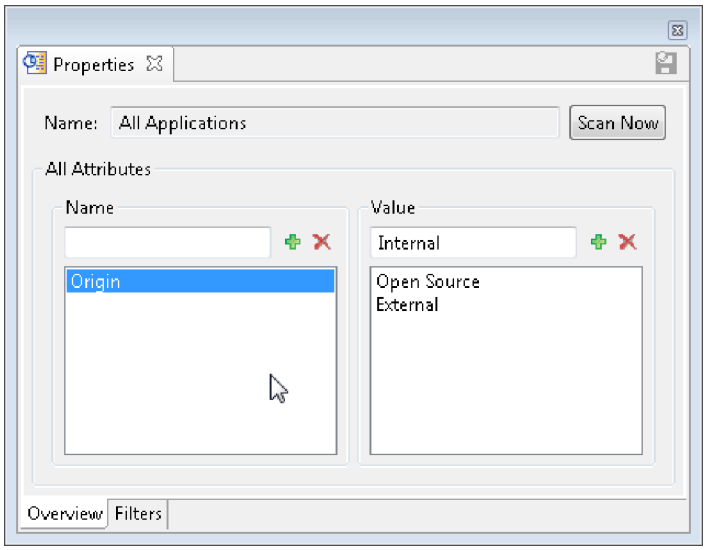
このタブでは、すべてのアプリケーションに対して既存のフィルターを指定でき、フィルターの適用方法を指定できます (フィルターは直接適用することも、反転して適用することもできます)。フィルターについては、 157 ページの『第 5 章 トリアージおよび分析』を参照してください。また、グローバル・フィルターの適用について詳しくは、 182 ページの『グローバル・フィルターの適用』を参照してください。

フィルタリングされた検出結果はスキャン結果には表示されず、アプリケーションまたはプロジェクトのメトリックで計算に入れられることはありません。

グローバル属性の追加および削除:

アプリケーションの属性をグループ化する前に、「すべてのアプリケーション」の属性を定義する必要があります。

このタスクについて



グローバル属性またはその値を削除するには、属性名または属性値を選択し、「属性の削除」をクリックします。その名前または値は、リストに表示されなくなります。

注: 属性を削除しても、履歴結果には影響しません。

グローバル属性およびその値を追加するには、次の手順で行います。

手順

1. 「すべてのアプリケーション」を選択します。
2. 「プロパティー」ビューの「概要」タブで、属性の名前を入力します。
3. 「属性の追加」をクリックします。 属性名が「名前」リストに表示されます。
4. 名前を付けた属性を選択します。
5. 属性の「値」を入力します。
6. 「値の追加」をクリックします。 属性値が値リストに表示されます。

「プロパティー」ビュー: 選択したアプリケーション

このビューでは、選択したアプリケーションの属性を構成します。 アプリケーション属性は、前に作成したグローバル属性に依存します。

- 279 ページの『概要』
- 279 ページの『除外およびフィルター (Exclusions and Filters)』
- 280 ページの『ルールおよびルール・セット』

- 280 ページの『変更された検出結果』
- 280 ページの『カスタム検出結果』

概要

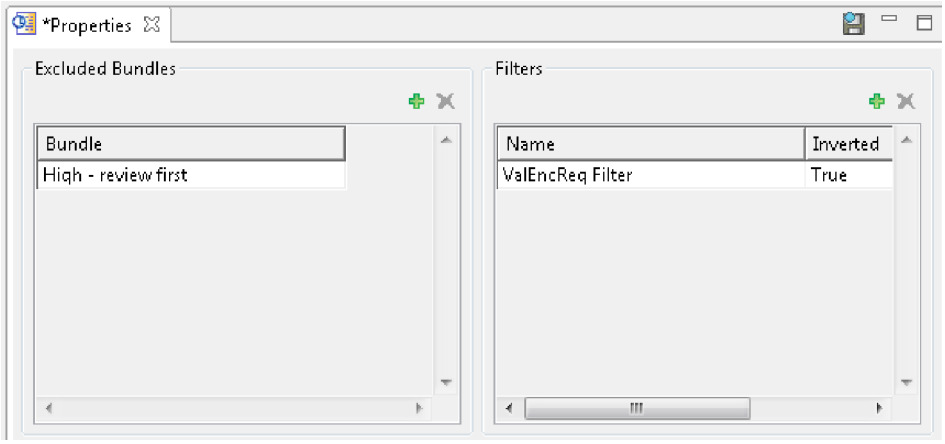
「概要」タブには、以下の内容が表示されます。

- アプリケーション名。アプリケーション名は、フィールドに新しい名前を入力して変更することができます。
- アプリケーション属性

除外およびフィルター (Exclusions and Filters)

このタブでは、選択したアプリケーションに対して既存のフィルターを指定でき、フィルターの適用方法を指定できます (フィルターは直接適用することも、反転して適用することもできます)。このタブでは、スキャンから結果を除外するバンドルを管理することもできます。フィルターについては、 157 ページの『第 5 章 トリアージおよび分析』を参照してください。また、グローバル・フィルターの適用について詳しくは、 182 ページの『グローバル・フィルターの適用』を参照してください。

除外およびフィルタリングされた検出結果はスキャン結果には表示されず、アプリケーションまたはプロジェクトのメトリックで計算に入れられることはありません。



ルールおよびルール・セット

「エクスプローラー」ビューでアプリケーションを選択すると、「プロパティー」ビューの「パターン・ルール/パターン・ルール・セット」タブで、アプリケーションのスキャン時に適用されるパターン・ルールとパターン・ルール・セットを追加することができます。パターン・ベースのスキャンを使用して、検出結果として表示させたいテキスト・パターンを検索します。 個々のルールとルール・セットをアプリケーションとプロジェクトの両方に適用できます。パターン・ベースの分析については 268 ページの『パターン・ベースのルールによるカスタマイズ』を、「プロパティー」ビューでのルールとルール・セットの適用方法については 275 ページの『パターン・ルールおよびパターン・ルール・セットの適用』を参照してください。

変更された検出結果

「変更された検出結果」タブでは、前に変更した検出結果を表示、編集、または削除するか、既存の検出結果を変更します。 変更された検出結果 とは、脆弱性タイプ、重大度、分類、または注が変更された検出結果です。

カスタム検出結果

「カスタム検出結果」タブでは、カスタム検出結果を表示、追加、編集、または削除します。詳しくは、 198 ページの『カスタム検出結果』を参照してください。

アプリケーション属性の作成**:**

手順

1. 「概要」タブで、「属性の追加」をクリックします。
2. 「グローバル属性」ダイアログ・ボックスで、アプリケーションに適用する属性の名前を選択します。
3. 「値」列をクリックし、リストから属性値を選択します。

「プロパティー」ビュー**:** 選択したプロジェクト

「プロパティー」ビューのこのモードでは、選択したプロジェクトのパラメーターを構成します。 プロジェクト属性は、前に作成したグローバル属性に依存します。プロパティーは、選択したプロジェクトによって異なります。

- 281 ページの『選択したプロジェクトの「概要」タブ』
- 282 ページの『フィルター』
- 282 ページの『パターン・ルールとパターン・ルール・セット』
- 283 ページの『ファイル拡張子』
- 284 ページの『ソース』
- 284 ページの『JavaServer Page (JSP) プロジェクト依存関係』
- 285 ページの『プロジェクト依存関係』
- 285 ページの『コンパイル』
- 286 ページの『最適化』
- 286 ページの『プリコンパイル・タブ (ASP.NET のみ)』

選択したプロジェクトの「概要」タブ

「概要」タブには、以下の内容が表示されます。

- プロジェクトの名前。 プロジェクト名は、フィールドに新しい名前を入力して変更することができます。
- プロジェクトのファイル名およびパス
- プロジェクト・タイプ
- このセクションには、ターゲットの構成が表示されます。 .NET および C++ のプロジェクトの場合、このセクションには、「プロジェクト依存関係」タブに保存されているターゲットの構成が表示されます。 その他のすべてのプロジェクト・タイプの場合、このセクションには、「デフォルト」が表示されます。

- フィルター・オプション: 「外部ソースに含まれている検出結果をフィルタリング」を選択して、スキャンされたプロジェクトのソース・ファイルではないファイルで検出された検出結果をすべてフィルタリングで除外します。 このオプションにより、検出結果が ASP.NET などのコンパイラー生成ファイルまたは一時ファイルで報告されるプロジェクトで、不要な手間が軽減されます。
- 脆弱性分析キャッシュ・オプション: 反復してスキャンを行い、カスタム・ルールを追加してから、ソース・コードを変更せずに再スキャンを行うことによって、コード・ベースの評価を詳細化する場合、脆弱性分析キャッシュを使用するようにプロジェクト・プロパティーを設定すると、スキャン時間を大きく削減することができます。 これを行うには、プロジェクト・プロパティーで、「脆弱性分析キャッシュを有効にする」チェック・ボックスを選択します。 このチェック・ボックスを選択した後に最初にプロジェクトをスキャンすると、脆弱性分析キャッシュが作成されます。 プロジェクトのすべての後続のスキャンで、脆弱性分析キャッシュが使用され、スキャン時間が削減されます。

  脆弱性分析キャッシュおよび Java 増分分析が有効にされた状態で作成されたキャッシュを消去するには、「キャッシュの消去」をクリックします。次にプロジェクトをスキャンすると、完全スキャンが実行され、新規の脆弱性分析キャッシュが作成されます。以下のような場合に、キャッシュの消去が必要となることがあります。

  – 最後のスキャン以降にプロジェクト内のソース・コードが変更された。
  – ソース・ファイルの追加または削除などの、プロジェクト構成の変更を行った。
  – コード構成オプションを変更した。 例えば、Java をスキャンしていて、クラスパスが変更された場合や、C または C++ をスキャンしていて、include パスまたはプリプロセッサー定義を変更した場合に、キャッシュを消去したほうがよいことがあります。
  – Java 増分分析を有効にしており、完全スキャンを実行したい、あるいはキャッシュを消去することで修復できる問題が発生している。詳しくは、 132 ページの『Java の増分分析』 を参照してください。

  注: カスタム・ルール・ウィザードでカスタム・ルールを作成するときに、「キャッシュの消去」チェック・ボックスを選択することによって、脆弱性分析キャッシュを消去することもできます。
- 文字列解析: 文字列解析は、Java プロジェクトまたは Microsoft .NET プロジェクトでの文字列操作をモニターします。これにより、サニタイズ・プログラムおよび検証プログラムのルーチンを自動検出できます。この検出を使用すると、誤検出や検出漏れを削減できます。文字列解析を有効にするには、「文字列解析で検証プログラムやサニタイズ・プログラムの関数を検索できるようにする」チェック・ボックスを選択します。 「インポートされたルールをグローバル・スコープに適用する」チェック・ボックスは、検出されたサニタイズ・プログラムまたは検証プログラムのルーチンを単一のプロジェクトに適用するか、あるいはグローバル・レベルで (すべてのプロジェクトに) 適用するかを決定します。

  注: 文字列解析を適用すると、スキャン速度が低下する場合があります。 したがって、この機能はコード変更後にのみ適用し、その後は、後続のスキャンのために無効にすることをお勧めします。 また、検出されたルーチンは 提案 として

表示し、監査員がそれらを確認する必要があります。 これらのルーチンは、「カスタム・ルール」ビューに表示できます。

- ファイル・エンコード: プロジェクト内のファイルの文字エンコードは、AppScan Source がファイルを適切に読み取る (そして、例えば、それらをソース・ビューに正しく表示する) ことができるように設定する必要があります。

  注: AppScan Source プロジェクトのデフォルトのファイル・エンコードは、**ISO-8859-1** です。 デフォルトのファイル・エンコードは、全般設定ページで変更できます。

フィルター

このタブでは、選択したプロジェクトに対して既存のフィルターを指定でき、フィルターの適用方法を指定できます (フィルターは直接適用することも、反転して適用することもできます)。 フィルターについては、 157 ページの『第 5 章 トリアージおよび分析』を参照してください。また、グローバル・フィルターの適用について詳しくは、 182 ページの『グローバル・フィルターの適用』を参照してください。

パターン・ルールとパターン・ルール・セット

「エクスプローラー」ビューでプロジェクトを選択すると、「プロパティー」ビューの「パターン・ルール/パターン・ルール・セット」タブで、プロジェクトのスキャン時に適用されるパターン・ルールとパターン・ルール・セットを追加することができます。パターン・ベースのスキャンを使用して、検出結果として表示させたいテキスト・パターンを検索します。 個々のルールとルール・セットをアプリケーションとプロジェクトの両方に適用できます。パターン・ベースの分析については 268 ページの『パターン・ベースのルールによるカスタマイズ』を、「プロパティー」ビューでのルールとルール・セットの適用方法については 275 ページの『パターン・ルールおよびパターン・ルール・セットの適用』を参照してください。

ファイル拡張子

プロジェクトの有効なファイル拡張子を構成または追加したり、スキャンからファイルを除外して Web ファイルとして拡張子を指定したりするには、このタブを使用します。

「ファイル拡張子」セクションには、現行プロジェクト・タイプの 117 ページの『プロジェクト・ファイル拡張子』 設定ページでグローバルに設定された拡張子がリストされます (「ファイル拡張子のセット」メニューを使用して、別のプロジェクト・タイプのファイル拡張子を選択できます)。現行プロジェクトのスキャンから拡張子を除外するには、リストでその拡張子を選択し、「拡張子の除外」をクリックします。これにより、その拡張子はタブの「除外する拡張子」セクションにリストされます。

プロジェクトの拡張子を追加するには、「追加の拡張子」セクションで「拡張子の追加」を選択してから、ファイル拡張子を入力し、その拡張子を持つファイルがスキャンされるか、Web ファイルと見なされるか、または除外されるかを示します。

表 32. ファイル拡張子の設定

設定	説明	使用例
「スキャン」または「評価」	指定された拡張子を持つファイルを完全分析に含みます。	• Java プロジェクト用の .xxx 拡張子が作成され、「スキャン」または「評価」のマークが付けられると、その拡張子を持つファイルはコンパイルされ、スキャンされます。 • ファイルのコンパイルとスキャンを行わない場合 (C++ のヘッダー・ファイルなど)、そのファイルはプロジェクトの一部にすることができますが、「スキャン」または「評価」のマークはつきません。これらのファイルは、プロジェクトに含まれ、パターン・ベースの分析時に検索されます。
Web ファイル	JSP コンパイル用に指定の拡張子を持つファイルにマークを付けます。この設定により、AppScan Source は Web ソースを非 Web ソースと分離することができます。	Java プロジェクト用の .yyy 拡張子が作成され、「**Web** ファイル」のマークが付けられると、その拡張子を持つファイルは、プロジェクトで Web ソースとして調整されます。AppScan Source が分析の準備をすると、これらのファイルは分析のためにクラスにプリコンパイルされます。
除外	指定の拡張子を持つファイル用に、プロジェクトでソース・ファイルを作成しません。この拡張子を持つファイルはスキャンされません。	コンパイルのためにプロジェクトに必要であるものの、分析に組み込む必要がないファイルの .zzz 拡張子を作成します。

ソース

スキャンに含めるソースを指定します。

- 作業ディレクトリー: AppScan Source プロジェクト・ファイル (ppf) の位置であり、すべての相対パスのベース。
- 「ソース・ルートの追加」および「ソース・ルートの削除」: 「ソース」タブに、プロジェクト構成ウィザードからプロジェクトに対して規定されたプロパティー、またはインポートされた ppf で定義されたプロパティーが表示されます。

「ソース・ルートの削除」は、「ソース・ルート」アイコンが選択されている場合のみ使用可能です。 ソース・コード・ルート・ディレクトリーの削除に使用します。

- ソース・ルートの検出 (Java プロジェクトのみ): AppScan Source for Analysis が自動的にすべての有効なソース・ルートを検索できるようにします。
- プロジェクト・ファイルは、「ソース・ルート」 アイコンの下に表示されます。 スキャンから除外されたファイルには、赤いファイル・アイコンが付いています。 (除外済みファイルを右クリックすると、そのメニューで「除外」は無効に、「含める」は有効になっています)。組み込みファイルを除外するには、ファイルを右クリックして、メニューで「除外」を選択します。 除外済みファイルを組み込むには、ファイルを右クリックして、メニューで「含める」を選択します。

JavaServer Page (JSP) プロジェクト依存関係

「JSP プロジェクト依存関係」タブに、指定された JSP プロジェクト用に規定されたプロパティーが表示されます。

- Web (JSP) コンテンツを含む: プロジェクトが、JavaServer Pages を含む Web アプリケーションであるかどうかを示します。
- Web コンテキスト・ルート: `WEB-INF` ディレクトリーを含む `WAR` ファイルまたはディレクトリー。 Web コンテキスト・ルートは、有効な Web アプリケーションのルートでなければなりません。
- JSP コンパイラー: 製品に付属の Tomcat 7 が、デフォルトの JSP コンパイラー設定です (デフォルト JSP コンパイラーは「Java および JSP」設定ページで変更できます)。 AppScan Source にサポートされるコンパイラーについて詳しくは、http://www.ibm.com/support/docview.wss?uid=swg27027486 を参照してください。

  Apache Tomcat バージョン 7 および 8 は、AppScan Source のインストール済み環境に含まれています。「**Tomcat 7**」および「**Tomcat 8**」設定ページが未構成の場合、AppScan Source は、提供されている Tomcat JSP コンパイラー (現在デフォルトとしてマーク) を使用して JSP ファイルをコンパイルします。外部でサポートされている Tomcat コンパイラーを使用したい場合は、Tomcat 設定ページを使用して、ローカルの Tomcat インストール済み環境を示します。

  Oracle WebLogic サーバー または WebSphere Application Server を使用する場合は、分析時にアプリケーション・サーバーを JSP コンパイルに使用できるようにするため、適切な設定ページを構成して、アプリケーション・サーバーのローカルのインストール済み環境を示す必要があります。この構成をまだ完了していない場合は、JSP コンパイラーを選択する際に構成を行うようにメッセージによって指示されます。メッセージ内の「はい」をクリックすると、該当する設定ページに進みます。「いいえ」をクリックすると、JSP コンパイラーの選択項目の隣に警告リンクが表示されます (リンクを選択すると、設定ページが開きます)。

プロジェクト依存関係

「プロジェクト依存関係」タブには、プロジェクト・プロパティーが表示されます。 このタブの「構成」の設定は、言語により異なります。以下に例を挙げます。

- 「オプション」を使用すると、追加で必要なコンパイラー・パラメーターを選択できます。
- JDK 設定は Java に固有です。
- プリプロセッサー定義は C/C++ コードに固有です。 プリプロセッサー定義を指定するときは、コンパイラーの `-D` オプションを含めないでください (例えば `-Da=definition1` の代わりに `a=definition1` を使用してください)。複数の定義を指定するときは、セミコロンで区切ったリストを使用します。
- ターゲットの構成は、.NET および C++ のプロジェクトでのみ使用できます。

コンパイル

- オプション: プロジェクト構成に追加で必要なコンパイラー・パラメーター。
- JDK の使用:「設定」で構成した、プロジェクトのコンパイルに使用される JDK を示します。 103 ページの『第 3 章 設定』を参照してください。

  Java プロジェクトは、ローカルの Java Development Kit (JDK) の位置を参照する可能性があります。プロジェクトがサーバーに移動すると、JDK パスは無効になる場合があります。ローカル・プロジェクトをサーバーに転送するには、所定の JDK を指定するプロジェクトごとにデフォルトの JDK パスを指定する必要があります。

  注**:** 製品に付属の JSP プロジェクトのデフォルト・コンパイラーは、Tomcat 7 です。これには、Java バージョン 1.6 以上が必要です。 **Tomcat 7** をデフォルトのまま使用している場合、古い JDK を選択すると、以下のスキャン中のコンパイル・エラーが発生します。
- 検証: 「検証」をクリックして、プロジェクト依存関係が正しく構成されていることを確認します。 Java プロジェクトをチェックして、ソース同士やクラスパス間で構成の競合があるかどうか、およびコンパイル・エラーがあるかどうかを調べます。 クラスパス内のクラスが、ソース・ルートで重複している場合、競合が存在します。 (競合が存在する場合、クラスパスを変更して、クラスの競合を削除してください。)

  競合をチェックした後で、「検証」をクリックして、プロジェクトをコンパイルできるかどうか、およびコンパイル・エラーがレポートされるかどうかを判別します。

最適化

- プリコンパイル済みクラス: スキャン中にコンパイルするのではなく、プリコンパイル済み Java または JSP クラス・ファイルを使用します。 このオプションを選択すると、ソース・ステージ・オプションが無効になります。
- コンパイル・エラーの影響を最小化するためにソース・ファイルをステージする: AppScan Source がソースをステージング・ディレクトリーにコピーするかどうかを制御します。

  「ディレクトリーと一致しないパッケージの修正」では、Java コンパイルが各ソース・ファイルを開く必要があります。

  「スキャンの合間にステージング領域をクリーンアップ」によって、スキャンと次のスキャンの間のパフォーマンスが向上します。

プリコンパイル・タブ (ASP.NET のみ)

プリコンパイルは、Web サイトの特殊なページ (デフォルトでは precompile.axd) に対して HTTP 要求を出すことによって実行されます。 このページは、web.config で指定された特殊な HTTP ハンドラーによって処理されます。 このハンドラーは、client.aspx ファイルを含めたサイト全体をコンパイルして、.NET フレームワーク・ディレクトリー内の ASP.NET 一時ファイル・ディレクトリーに入れます。そこでファイルはすべてスキャンされます。

ASP.NET 1.1 をスキャンするには、Web サイトでデバッグ情報をコンパイルおよびビルドするように、その Web サイトを調整する必要があります。それ以降、Web サイトがデバッグ情報をコンパイルおよびビルドするということ自体が、セキュリティー上の脆弱性となります。 スキャンでこれを必要とするため、この脆弱性を無視しても支障はありません。 ただし、デプロイされたアプリケーションが、web.config で debug=true と指定してコンパイルされていないことを確認してください。

ASP.NET 1.1 Web サイトをプリコンパイルするには、このエレメントを、ご使用の web.config ファイル内の <system.web> エレメントに子として追加します。

```
<httpHandlers><add verb="*" path="precompile.axd"
type="System.Web.Handlers.BatchHandler"/></httpHandlers>
```

また、コンパイル・エレメントで debug=true と設定する必要があります。例:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
 <system.web>
  <httpHandlers><add verb="*" path="precompile.axd"
  type="System.Web.Handlers.BatchHandler"/>
  </httpHandlers>
   <compilation
    defaultLanguage="c#"
    debug="true"
   />
...
```

このエレメントは、precompile.axd ページが .Net の特殊な System.Web.Handlers.BatchHandler クラスによって処理されることを、Web サイトに対して指定します。このクラスは、Web サイトのコンテンツをプリコンパイルして、ASP.NET 一時ファイル・ディレクトリーに入れます。

- Web サイト: サイトをプリコンパイルするようにターゲットに要求します。 デフォルトの位置は、precompile.axd です。 precompile.axd は仮想ファイルであり、web.config ファイルで指定されたファイルにマップします。
- 出力ディレクトリー: プリコンパイルのターゲットとなるディレクトリー。 AppScan Source は、このディレクトリーでプリコンパイルの出力を特定します。
- ASP.NET Web サイトのプリコンパイル: AppScan Source は自動的にプリコンパイルを行い、スキャン時には、プリコンパイルされた出力をスキャンします。
- プリコンパイルが失敗した場合はスキャンを停止: 「**ASP.NET Web** サイトのプリコンパイル」および「プリコンパイルが失敗した場合はスキャンを停止」を選

択すると、プリコンパイルが失敗した場合にスキャンが停止されます。 そのようにしない場合は、Web サイトのプライマリー出力のみでスキャンが続行されます。

- 直ちにコンパイル: スキャンの前に、現在の設定に基づくプリコンパイルが正常に実行されることを調べるテストを行います。 コンパイルの出力は、「プリコンパイル出力」ペインに表示されます。
- 追加アセンブリー: 任意の .NET プロジェクト・タイプに対して、スキャンする追加アセンブリーを指定します。
- プロジェクト参照: .NET アセンブリー・プロジェクトと既存の .NET プロジェクトで、参照されるアセンブリーを検索するディレクトリーをリストします。

ファイル・プロパティー

ファイル・プロパティーは、一般に C/C++ アプリケーション用に構成されるプロジェクト依存関係に類似しています。

プロジェクトから構成データを組み込みます。このときファイル構成にプロジェクト構成データを組み込みます。 これにより、ファイル構成は、累積のプロジェクト構成およびファイル構成から成ります。 ファイル構成は、プロジェクト構成よりも優先します。

「スキャン構成」ビュー

「スキャン構成」ビューを使用して、スキャンの起動時に使用できる構成を作成することができます。このビューを使用すると、デフォルトのスキャン構成の設定も可能です。スキャン構成では、スキャン時に使用するソース・ルールを指定し、多数のスキャン設定を組み込むことができます。 スキャン構成で設定を行うと、良好なスキャン結果が得られることが多く、また、これらの設定を保存することができるため、スキャンを容易に、しかも短時間で行うことができます。

「スキャン構成」ビューには、以下の主なセクションがあります。

- 130 ページの『スキャン構成の管理』
- 130 ページの『「全般」タブ』
- 131 ページの『「汚染フロー分析」タブ』
- 132 ページの『「パターン分析」タブ』

スキャン構成の管理

このセクションは、スキャン構成を選択、追加、削除、保存、および共有する場合や、スキャン構成をデフォルトとして設定する場合に使用します。

- 新規スキャン構成を作成するには、「新規」をクリックします。スキャン構成の設定が完了したら、「保存」をクリックして変更内容を保存します。 スキャン構成をデフォルトとして設定するには、保存後に「デフォルトとして選択」をクリックします。 デフォルトのスキャン構成がどのように使用されるかについては、 119 ページの『ソース・コードのスキャン』を参照してください。
- 既存のスキャン構成を操作するには、既存のスキャン構成をリストから選択します。

- スキャン構成の設定を変更する場合は、「保存」をクリックして変更内容を保存します (不要な変更内容は、別のスキャン構成に切り替えてから「破棄」をクリックすると、破棄することができます)。
- 選択したスキャン構成を削除するには、「削除」をクリックします。
- スキャン構成を複写するには、「複写」をクリックします。これにより、元のスキャン構成の設定に基づいて新しいスキャン構成が作成されます。
- スキャン構成をデフォルトとして設定するには、「デフォルトとして選択」をクリックします。 デフォルトのスキャン構成がどのように使用されるかについては、 119 ページの『ソース・コードのスキャン』を参照してください。
- スキャン構成を他のユーザーと共有するには、「共有」をクリックします。スキャン構成が AppScan Source データベース に保存されます。

  注**:** スキャン構成を共有する (あるいは共有スキャン構成を変更または削除する) には、「共有構成の管理」権限が必要です。権限の設定について詳しくは、「*IBM Security AppScan Source* インストールと管理のガイド」を参照してください。

注**:** AppScan Source には、標準装備のスキャン構成が用意されています。 これらを変更または削除することはできません。これらのスキャン構成をリストで選択すると、複写したり、その設定を表示したりすることができます。

「全般」タブ

基本情報

このセクションでは、スキャン構成に名前を付けて説明を提供することができます。

フィルター

このセクションでは、スキャン構成を使用すると必ずスキャンに適用されるフィルターを 1 つ以上選択できます。フィルターを選択するときは、AppScan Source 事前定義フィルターまたは 共有フィルター、あるいは自分で作成したフィルターを選択できます。詳しくは、 123 ページの『スキャン構成の管理』を参照してください。

「汚染フロー分析」タブ

汚染フロー分析

汚染フロー分析を有効にし、その有効範囲を設定します。

スキャン・ルール

このセクションは、スキャンで有効になるソース・ルールを判別するために使用します。

ソースはプログラムへの入力で、ファイル、サーブレット要求、コンソール入力、ソケットなどがあります。 一部のソース・ルールを除外することにより、スキャンの速度を上げたり、関係のない入力に起因する脆弱性の検出を避けることができます。

ルールが特定の脆弱性、メカニズム、属性、またはテクノロジーに関連していることを示すには、ルール・プロパティーでルールをタグ付けします。 これらのプロパティーはルール・セットにグループ化され、これらのルール・セットは、関連したルールの共通セットに対応します。 ルール・セットまたは個々のルール・プロパティーのいずれかを指定することにより、スキャンに含めるソース・ルールを制限できます。

- スキャンに組み込む 1 つ以上の脆弱性タイプ (ルール・セット内でタイプ別に編成される) を選択します。
  - すべて: これを選択すると、サポートされるすべての入力のソースに起因する脆弱性が検出されます。
  - ユーザーの入力: これを選択すると、エンド・ユーザーによる入力に起因する脆弱性が検出されます。
  - **Web** アプリケーション: これを選択すると、Web アプリケーションのリスクに起因する脆弱性が検出されます。
  - エラー処理およびロギング: これを選択すると、エラー処理とロギングのメカニズムに起因する脆弱性が検出されます。
  - 環境: これを選択すると、構成ファイル、システム環境ファイル、およびプロパティー・ファイルに起因する脆弱性が検出されます。
  - 外部システム: これを選択すると、外部エンティティーに起因する脆弱性が検出されます。
  - データ・ストア: これを選択すると、データ・ストア (データベースやキャッシュ処理など) に起因する脆弱性が検出されます。
  - 異常な項目: これを選択すると、通常は実動アプリケーションの一部ではないルーチンに起因する脆弱性が検出されます。
  - ファイル・システム: これを選択すると、ファイル・システムに起因する脆弱性が検出されます。
  - 機密データ: これを選択すると、機密データに起因する脆弱性が検出されます。

  吹き出しテキストは、このセクションの各ルール・セットについて記述しています。
- スキャンに組み込む個々のスキャン・ルール・プロパティーを選択します。「選択済みのルール・セットを破棄し、個々のルール・プロパティーを選択」をクリックします。 これにより、「ルール・プロパティーの選択」ダイアログ・ボックスが開き、個々のルール・プロパティーを選択できるようになります。このダイアログ・ボックスでの作業が完了すると、選択されていたルール・セットがすべて破棄されます。 選択されたルール・プロパティーを持つスキャン・ルールがスキャンに使用されます。

詳細設定

このセクションは、上級者向けです。 このセクションには、スキャン結果を向上させるための、さまざまな設定が含まれています。 吹き出しテキストは、このセクションの各設定について記述しています。

「パターン分析」タブ

パターン分析

このセクションを使用して、スキャン構成を使用する場合にパターン・ベースのスキャンを有効にします。パターン・ベースのスキャンは、カスタマイズされた検索基準に基づいてソース・コードの分析を行う機能です。

「パターン・ルール・セット」および「パターン・ルール」

これらのセクションを使用して、パターン分析時に使用するルールとルール・セットを追加します。詳しくは、 268 ページの『パターン・ベースのルールによるカスタマイズ』および 123 ページの『スキャン構成の管理』を参照してください。

レポート・エディター

レポート・エディターを使用して、カスタム・レポートまたはテンプレートを編集したり、新規レポートを作成したりできます。 カスタム・レポートには、検出結果情報、コード・スニペット、AppScan Source トレース、修復コンテンツ、脆弱性マトリックスなど、検出結果レポートで使用可能なすべての項目が含まれます。新規レポートの設計を開始する前に、レポート・エディター内で既存のレポート・テンプレートを変更してみることで、レポート作成プロセスをよく理解することをお勧めします。

レポート・エディターは、「レポート・レイアウト」、「カテゴリー」、および「プレビュー」タブから成り立っています。

- レポート・レイアウト: レポートの外観を設計します。 レイアウトでは、AppScan Source レポート要素を追加、削除、および再配列します。
- カテゴリー: カテゴリーを作成および編集します。カテゴリー は検出結果のグループです。 カテゴリーは、レポートに含める検出結果、それらの検出結果をグループ化する方法、およびグループ化の順序を特定します。
- プレビュー: 編集時に、現在の評価のレポートを表示します。

これら 3 つのタブには、以下の共通フィールドが含まれています。

- ファイル: 保存済みのグループ化ファイル (読み取り専用) のパス。ファイルが保存されるまで、このフィールドには何も表示されません。グループ化ファイルを保存すると、このファイルは、レポートを定義する XML ファイルになります。
- 名前: ユーザー定義のレポート名。

以下のツールバー・ボタンを使用して、カスタム・レポートを保存する、開く、作成する、コピーする、および生成する操作を実行します。

- 新規レポートの作成: 新規カスタム・レポートを作成します。
- 既存からの新規レポート: 既存のレポート・テンプレートから新規カスタム・レポートを作成します。
- 保存済みのレポートを開く: 編集するグループ化ファイルを開きます。

- 保存: 現在のレポートを指定されたファイルに保存します。
- 名前を付けて保存: 現在のレポートを新規ファイルに保存します。
- このレポートのインスタンスの生成: 現在開いている評価のレポートのコピーを作成します。

ヒント**:** 既存のレポートのサンプルを表示するには、「既存からの新規レポート」をクリックし、AppScan Source のいずれかのレポート・テンプレートを選択します。これらのテンプレートの「レポート・レイアウト」タブと「カテゴリー」タブを使用すると、各レポートの大まかな設計内容を確認することができます。

「レポート・レイアウト」タブ

「レポート・レイアウト」タブは、「パレット」セクションと「レイアウト」セクション、および各ページに表示されるヘッダーまたはフッターを指定できるセクションから構成されています。

ページ・ヘッダーとページ・フッター

「ページ・ヘッダー」フィールドを使用すると、レポートの各ページの上部に表示されるテキストを指定することができ、「ページ・フッター」フィールドを使用すると、レポートの各ページの下部に表示されるテキストを指定することができます。

パレット

「パレット」には、AppScan Source 標準レポートの構成要素のリストが表示されます。 一部の要素については、「カテゴリー」タブで定義されたカテゴリーの情報だけが表示されます (253 ページの表 19 を参照)。

表 *33.* レポート・レイアウト・パレット - カテゴリーに依存しない要素

レポート要素	説明
テキスト・ヘッダー	テキストの太字ブロックをレポート・レイアウトに追加します。
イメージ・ヘッダー	指定のサイズに拡大または縮小されたイメージをピクセル単位で表示します。
AppScan Source ヘッダー	AppScan Source の商標表示を含むレポート・ヘッダー。
タイトルと日付	スキャンされた項目名を含むレポートのタイトルと、スキャン日付およびレポートの生成日付。
テキスト・ブロック	任意のユーザー定義のテキスト。 「ラベル」フィールドで、テキスト・ブロックの見出しを追加することもできます。
脆弱性マトリックス	評価の脆弱性マトリックス (「脆弱性マトリックス」ビューに表示されるものと同じグラフを表示します)。
メトリック	プロジェクト内のすべてのパッケージのパッケージ、クラス、メソッド、およびコード行の総数を示します。

表 33. レポート・レイアウト・パレット - カテゴリーに依存しない要素 (続き)

レポート要素	説明
スキャン履歴	現在のスキャンのメトリックと、同じターゲットのスキャンの履歴メトリック。

表 34. レポート・レイアウト・パレット - カテゴリーに依存する要素

レポート要素	説明
レポート・カード	「カテゴリー」タブで定義された各カテゴリーの脆弱性レベルの簡単な明細。レポート詳細およびセクションの概要を示す重大度インディケーターへのリンクが含まれます。
脆弱性明細	「カテゴリー」タブで定義されたすべてのカテゴリーにおける脆弱性の数の明細を持つ表 (重大度および分類別)。
部分的レポート・カード	「カテゴリー」タブでの定義による、ユーザー指定カテゴリーの脆弱性レベルの明細。
カテゴリー	「カテゴリー」タブでの定義に従い、カテゴリー化されたすべての検出結果データをリスト表示します。
カテゴリー	「カテゴリー」タブで定義された 1 つ以上のカテゴリーにおけるすべての検出結果をリスト表示します。

レイアウト

パレットから追加した項目は「レイアウト」に表示されます。レイアウト内の項目の削除、変更、移動を行うには、セクション・ツールバーを使用します。

「カテゴリー」タブ

「カテゴリー」タブを使用すると、選択したバンドル、プロパティー、または検出結果に基づいて、検出結果を含むカテゴリーを追加することができます。追加したカテゴリーは、特定の項目を「レイアウト」に追加する際に使用することができます。例えば、「脆弱性明細」を「レイアウト」に追加すると、すべてのカテゴリーにおける脆弱性の数の明細を持つ表 (重大度および分類別) がレイアウトに追加されます。「カテゴリー」タブは、カテゴリーのツリーが表示されるペインと、選択したカテゴリーの属性を編集するためのペインから成り立っています。各カテゴリーには、定義された特定の要件を満たす評価内の検出結果が含まれています。

以下のカテゴリーを使用できます。

- バンドル: バンドル・カテゴリーは、複数のバンドル名のリストから成り立っています。このリストに名前が出現するバンドルに含まれるすべての検出結果が、このカテゴリーに表示されます。 バンドルは現在の評価から選択しますが、バンドルは名前によってマッチングされるため、バンドル・カテゴリーは任意の評価に適用できます。
- 個別の検出結果: カテゴリーに追加する特定の検出結果を選択します。検出結果のスナップショットのみがレポートに追加されます。 レポートに追加された後で検出結果を変更した場合、レポートはその変更を反映しません。

- 脆弱性タイプ、メカニズム、およびテクノロジーのプロパティー: AppScan Source セキュリティー・ナレッジ・データベース 内のプロパティーおよび API からの必須プロパティーのセットを選択します。検出結果に少なくとも 1 つの「プロパティー」およびすべての「必須プロパティー」が含まれる場合、その検出結果はレポートに組み込まれます。

以下の表は、カテゴリー・ペインと、各ペインを構成する項目を示しています。

表 *35.* 「カテゴリー」タブ属性

属性	説明	編集方法
ラベル	カテゴリーの簡単な名前 (Buffer Overflow など)。 このラベルにより、カテゴリーのツリー・リスト内のカテゴリーが識別されます。また、このラベルは、カスタム・レポート内のカテゴリー見出しになります。	1 行のテキスト・フィールドにラベルを入力します。
概要	このカテゴリー内でレポートされる検出結果の数を示す文のテンプレート。 レポート生成中に、実際のカウントが `%FindingCount%` を置き換えます。	カテゴリーの簡略説明を入力し、「カウントの追加」をクリックして、カーソル位置にある語句内に変数 `%FindingCount%` を配置します。
テキスト	カテゴリーの簡単な説明。	カテゴリーを説明するテキストを入力します。
プロパティー (プロパティー・カテゴリーのみ)	これらのプロパティーのうち少なくとも 1 つを含む検出結果が、このカテゴリー内でレポートされます。 リストされたすべての必須プロパティーが検出結果に含まれない場合、その検出結果はこのカテゴリーに含まれません。	ツールバーの「追加」をクリックし、「プロパティーの追加」ダイアログ・ボックスからプロパティーを選択します。 「削除」をクリックして、選択した項目をリストから削除します。
必須プロパティー (プロパティー・カテゴリーのみ)	すべての必須プロパティーおよび少なくとも 1 つのプロパティーを含む検出結果が、このカテゴリーのレポート内に表示されます。	ツールバーの「追加」をクリックし、「プロパティーの追加」ダイアログ・ボックスからプロパティーを選択します。 「削除」をクリックして、選択した項目をリストから削除します。
バンドル (バンドル・カテゴリーのみ)	このカテゴリーに含めるバンドルの名前を指定します。	「バンドル」セクションの「バンドルの追加」をクリックし、リストからバンドルを選択します。

表 35. 「カテゴリー」タブ属性 (続き)

属性	説明	編集方法
検出結果 (検出結果カテゴリーのみ)	このカテゴリーに含める検出結果を指定します。	任意の検出結果表で検出結果を選択し、その表のツールバーで「検出結果の追加」をクリックして、選択した検出結果を追加します。選択した検出結果が複数のビューに存在する場合、どのビューの検出結果を追加するかを選択するためのプロンプトが表示されます。 検出結果を、検出結果表から「レポート・エディター」ビューの表にドラッグすることも、レポート・エディターにドラッグすることも、カテゴリー・ツリー内の既存の検出結果カテゴリーに直接ドラッグすることもできます。

「プレビュー」タブ

テンプレートを編集しながら、AppScan Source for Analysis レポートをプレビューできます。 「プレビュー」ペインで、「プレビュー」をクリックして、開いている評価のレポートを表示します。

スキャン出力に役立つビュー

このセクションのビューは、スキャン出力を表示および管理するために使用されます。

- 『「コンソール」ビュー』
- 337 ページの『「メトリック」ビュー』
- 337 ページの『「自分の評価」ビュー』
- 338 ページの『「公開された評価」ビュー』

「コンソール」ビュー

「コンソール」ビューには、現在のスキャンの出力が表示されます。出力には、状況情報、出力テキスト、およびエラー・メッセージが含まれます。 このビューには、2 つのコンソールが表示される場合があり、その場合 1 つは現在実行中のスキャン、もう 1 つは完了したスキャンのコンソールです。

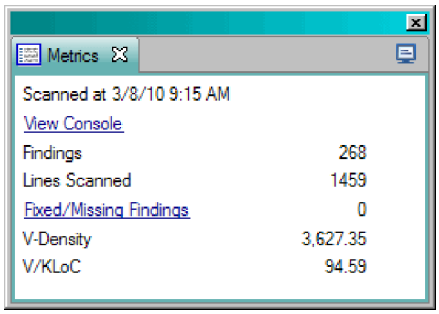
出力コンソールには、スキャンされたファイル、スキャンされたファイルの総数、検出された脆弱性の総数、スキャンの時間、脆弱性の密度を含む、完全なスキャン出力が表示されます。

ツールバーのボタンで、コンソール出力を操作します。

エラー・コンソールには、出力エラー・メッセージと、スキャンでのエラーの数が表示されます。 エラーの値は、スキャン中に更新されます。

「メトリック」ビュー

「メトリック」ビューは、評価ごとの統計を表示します。スキャンされたコード行、検出結果の総数、 V-Density、および V/KLoC が含まれます。



コンソールの表示

現在のスキャンの出力を表示するための「コンソール」ビューを開くハイパーリンク。

検出結果

スキャンで特定された検出結果の数。

スキャンされた行

スキャンされたコードの行数。

修正された/存在しない検出結果

アプリケーション・バンドル内に含まれているが、このスキャンでは検出されなかった項目の数。

V-Density

アプリケーションの脆弱性を評価するための一貫性のある方法を可能にする数式。V-Density は、検出結果の数と重要性を、分析されているアプリケーションまたはプロジェクトのサイズに関係付けることによって計算されます。

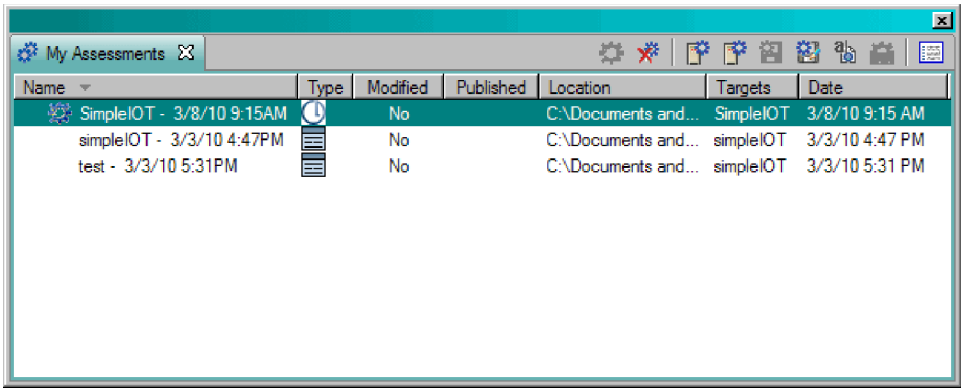
V/KLoC

コード 1000 行当たりの検出された脆弱性の数。

「自分の評価」ビュー

「自分の評価」ビューには、評価のリストが表示されます。リストには、現在開かれている評価と、自分が保存した評価が含まれています。評価の現行作業セットが変更されると (例えば、新しい評価を追加したり、評価を変更したりすると)、作業セット内に保存されていない変更があることを示すアスタリスクが、ビューのタイトルの横に表示されます。

- 名前: 評価名。
- タイプ: スキャンの対象がアプリケーション ( )、プロジェクト ( )、またはファイル ( ) のいずれであるかを示すアイコン。評価名の横の星印は、評価が現在開かれていることを示します。
- スキャン構成: スキャンに使用したスキャン構成。
- 変更済み: 「はい」または「いいえ」で評価の変更状態を示します。
- 公開済み: 評価が AppScan Source データベース に公開されているというインディケーター。
- 位置: 評価ファイルへのパス (`<file_name>.ozasmt`)。
- ターゲット: スキャンされたアプリケーション、プロジェクト、またはファイル。
- 日付: スキャン完了日。



スキャンが完了すると、スキャンが自動的に「マイ評価」ビューに表示されます。このビューに表示される評価には、このコンピューターからのスキャン、または自分が追加したコンピューターのスキャンが含まれます。

このビューでは、評価のオープン、追加、削除、公開、保存、名前変更、または比較を実行できます。 保存も公開もせずにこのビューから評価を削除した場合、その評価は永久に削除されます。 すべての保存済みの評価に、すべての結果、出力、およびエラー・ログが含まれることに注意してください。 評価の保存および公開について詳しくは、 152 ページの『評価の保存』を参照してください。

評価の比較について詳しくは、 197 ページの『「差分評価」ビューでの 2 つの評価の比較』を参照してください。

ヒント: 「全般」設定で、「公開された評価」ビューおよび「自分の評価」ビュー内に表示する評価の最大数を設定できます。

「公開された評価」ビュー

「公開された評価」ビューには、AppScan Source データベースに公開されている評価がリストされています。

- 名前: 評価名。
- タイプ: スキャンの対象がアプリケーション ( )、プロジェクト ( )、またはファイル ( ) のいずれであるかを示すアイコン。評価名の横の星印は、評価が現在開かれていることを示します。

- スキャン構成: スキャンに使用したスキャン構成。
- 公開者: 評価を公開した人のユーザー名
- ターゲット: スキャンされたアプリケーション、プロジェクト、またはファイル。
- 日付: スキャン完了日。

「公開された評価」ビューでは、以下の操作を実行できます。

- 「マイ評価」ビューに評価を追加する
- 評価をフィルタリングする
- 評価を開いたり削除したりする
- 評価を閉じる
- 評価を比較する
- 評価を保存する
- 評価を名前変更する
- メトリックを表示する

ヒント**:** 「全般」設定で、「公開された評価」ビューおよび「自分の評価」ビュー内に表示する評価の最大数を設定できます。

トリアージに役立つビュー

このセクションのビューは、細分化されたスキャン出力を表示および管理するために使用されます。

- 『「差分評価」ビュー』
- 340 ページの『「カスタム検出結果」ビュー』
- 345 ページの『「除外された検出結果」ビュー』
- 343 ページの『「検出結果」ビュー』
- 346 ページの『「修正された/存在しない検出結果」ビュー』
- 345 ページの『「変更された検出結果」ビュー』
- 346 ページの『「検索結果」ビュー』
- 347 ページの『「レポート」ビュー』
- 349 ページの『「ソースとシンク」ビュー』

「差分評価」ビュー

差分評価は、「マイ評価」ビューと「検出結果」ビューの組み合わせを表します。比較する 2 つの評価を選択すると、その 2 つの評価の間の差異が表示されます。

このビューでは、新規の検出結果、修正された/存在しない検出結果、および共通の検出結果の総数が表示されます。

- 共通の検出結果は、両方の評価の表示に出現します。
- 新規の検出結果は、2 つの評価のうちの新しい方にのみ出現する検出結果です (青で強調表示)。
- 修正された/存在しない検出結果は、先に行われた評価のみに出現する検出結果です (緑のイタリック体で強調表示)。

右側のペインに検出結果が表示されます。 以下の操作を行うには、表で検出結果を右クリックします。

- 検出結果レポートの生成
- 検出結果を障害として送信
- 外部エディターで開く
- 内部エディターで開く

左側のペインに、比較される評価がリストされます。

注: 「差分評価」ビューではフィルターを無視します。

「カスタム検出結果」ビュー

「カスタム検出結果」ビューには、現在開いている評価に存在するユーザー定義の検出結果 (カスタム検出結果) が表示されます。 このビューでは、現在の評価のカスタム検出結果の作成、削除、および変更を実行できます。 「カスタム検出結果」ビューでカスタム検出結果が作成されると、新規の検出結果が現在の評価に追加され、評価メトリックが更新されます。

フィルターおよびバンドルは、「カスタム検出結果」ビュー内の検出結果に影響しません。 このビューでは、カスタム・レポート結果を表示したり、選択した検出結果を保存したりすることはできません。

検出結果を含むビュー

多くの AppScan Source for Analysis ビューに、以下の検出結果が含まれています。

- 「検出結果」ビュー
- 「変更された検出結果」ビュー
- 「カスタム検出結果」ビュー
- 「除外された検出結果」ビュー
- 「バンドル」ビュー
- 「修正された/存在しない検出結果」ビュー
- 「レポート」ビュー
- 「検索結果」ビュー
- 「差分評価」ビュー

検出結果表

以下の表に、検出結果表で使用可能な列を示します。 使用できない列は、表に表示されないことがあります。表示する列を選択するには (または表でその他のカスタマイズ作業を行うには)、 342 ページの『検出結果表のカスタマイズ』の指示に従ってください。

表 *36.* 検出結果表

列見出し	説明
トレース	この列のアイコンは、逸失シンクまたは既知のシンクのトレースが存在することを示します。
重大度	• 高: データの機密性や保全性、可用性、および/または処理リソースの保全性や可用性にリスクをもたらします。 重大度の高い状態は、即時に修復されるように優先順位付けする必要があります。 • 中: データ・セキュリティーおよびリソース保全性にリスクをもたらしますが、攻撃の影響は比較的受けにくい状態です。 重大度が中の状態は、可能であれば検討し修復します。 • 低: データ・セキュリティーおよびリソース保全性にもたらすリスクは最小です。 • 情報: 検出結果自体は、セキュリティーを侵害するわけではありません。これは、コードで使用されているテクノロジー、アーキテクチャー特性、またはセキュリティー・メカニズムについて説明するものです。
分類	検出結果のタイプ: 「確定」または「要確認」セキュリティー検出結果 - または「スキャン範囲」検出結果。 **注:** 場合によっては、「なし」の分類を使用して、セキュリティー検出結果でもスキャン範囲検出結果でもない分類が示されることがあります。
脆弱性タイプ	脆弱性カテゴリー (`Validation.Required` または `Injection.SQL` など)。
API	脆弱な呼び出し。API と、それに渡される引数の両方を表示します。
ソース	ソースはプログラムへの入力で、ファイル、サーブレット要求、コンソール入力、ソケットなどがあります。 多くの入力ソースの場合、コンテキストと長さについてはバインドされていないデータが返されます。 チェックされていない入力については、汚染されているものと見なされます。
シンク	シンクは、データの書き込み先となる任意の外部フォーマットです。 シンクの例としては、データベース、ファイル、コンソール出力、ソケットなどがあります。データをチェックせずにシンクに書き込むと、重大なセキュリティー脆弱性となる可能性があります。
ディレクトリー	スキャンされたファイルの絶対パス。

表 *36.* 検出結果表 *(*続き*)*

列見出し	説明
ファイル	セキュリティー検出結果またはスキャン範囲検出結果が発生するコード・ファイルの名前。検出結果内のファイル・パスは、スキャンされたプロジェクト作業ディレクトリーからの相対パスです。
呼び出し側メソッド	脆弱な呼び出しが行われている関数 (またはメソッド)。
行	脆弱な API を含むコード・ファイル内の行番号。
バンドル	この検出結果を含むバンドル。
CWE	コミュニティーが作成した、共通のソフトウェア脆弱性の辞書の ID およびトピック (共通脆弱性タイプ一覧 (CWE) のトピック)。

検出結果表のカスタマイズ**:**

検出結果を含むすべてのビュー (AppScan Source for Analysis の「差分評価」ビューを除く) で、表示したい列と列順序のみを指定することで、検出結果表をカスタマイズできます。 各ビューで異なる設定を使用できますが、オプションをすべてのビューに適用することもできます。 列順序をカスタマイズするには、このタスク・トピックの手順に従ってください。

このタスクについて

検出結果表内の列について詳しくは、 340 ページの『検出結果表』を参照してください。

手順

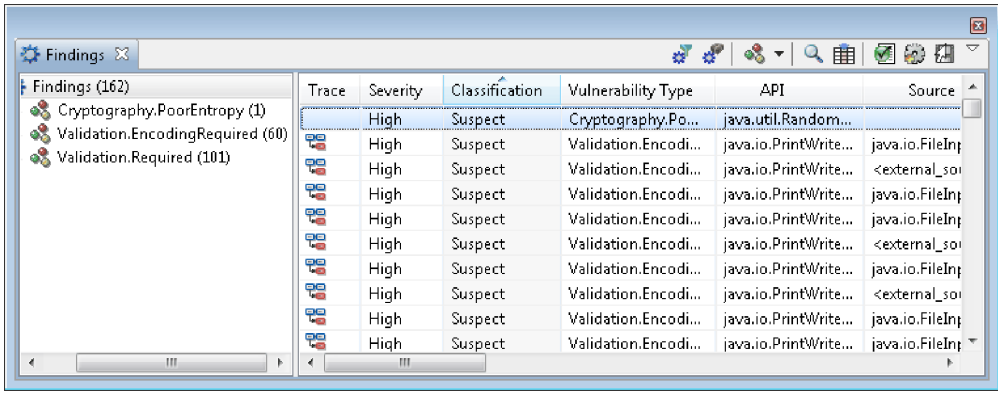
1. 「列の選択と順序付け」ツールバー・ボタンをクリックします。

   注**:** AppScan Source for Development (Visual Studio プラグイン) では、「表の列の選択と順序付け」ツールバー・ボタンをクリックします。
2. 「列の選択と順序付け」ダイアログ・ボックスで、列名を選択し、次に上矢印または下矢印をクリックして、列の位置を移動します。
3. 「列の追加」ボタンをクリックして、列をビューに追加します。 あるいは、「列の削除」ボタンをクリックして、ビューから列を削除します。

   注**:** AppScan Source for Development (Visual Studio プラグイン) では、これらのボタンにはそれぞれ「挿入」および「削除」というラベルが付けられています。
4. 「デフォルトの復元」をクリックして、デフォルトの列と列順序をリセットします。
5. 「**OK**」をクリックして、設定を保存します。

「検出結果」ビュー

「検出結果」ビューには、評価における検出結果のデータが表示されます。このトピックにリストされているパラメーターによって検出結果をグループ化することができます。



要確認**:** AppScan Source for Development (Eclipse プラグイン) および AppScan Source for Analysis では、これらはユーザー・インターフェースで「ビュー」と呼ばれます。 AppScan Source for Development (Visual Studio プラグイン) ではこれらはユーザー・インターフェースで「ウィンドウ」と呼ばれています。 この資料では、ビュー とウィンドウ のどちらに対しても主にビュー という用語を使います。

検出結果表のパラメーターのグループ分け

「検出結果」ビューで、「ツリー階層の選択」ツールバー・ボタンの下矢印を選択して、検出結果をグループ化するためのパラメーターを選択します。

表 *37.* 検出結果表のパラメーターのグループ分け

モード	グループ化
脆弱性タイプ	タイプ、重大度、分類
分類	分類、重大度、タイプ
ファイル	プロジェクト、ディレクトリー、ファイル、方法
API	API、タイプ
バンドル	バンドル、タイプ、API
CWE	CWE
表	グループ化なし

ツールバー・ボタン

表 *38.* ツールバー・ボタン

アクション	アイコン	説明
フィルターに一致しない検出結果の表示		このボタンで、「検出結果」ビューのフィルター済み検出結果の表示を切り替えます。

表 38. ツールバー・ボタン (続き)

アクション	アイコン	説明
バンドルされている検出結果の表示		このボタンで、「検出結果」ビューのバンドルされている検出結果の表示を切り替えます。このアクションは、自分で作成した、含まれているすべてのバンドル内の検出結果を非表示にします。この設定は、除外されたバンドルの検出結果の表示には影響しません。これらの検出結果は「検出結果」ビューに表示されることはありません。
ツリー階層の選択	選択されるグループによって異なります。	343 ページの『検出結果表のパラメーターのグループ分け』を参照してください。
検索		このボタンは、検出結果を検索できるダイアログ・ボックスを開きます。このダイアログ・ボックスでは、さまざまな検索オプションを使用できます。検索を行うと、結果が「検索結果」ビューに表示されます。
列の選択と順序付け		このボタンは、「列の選択と順序付け」ダイアログ・ボックスを開きます。このダイアログ・ボックスで、列を追加/削除したり、既存の列を変更したりすることができます。
「レポート」ビュー		このボタンは、「レポート」ビューを開きます。「レポート」ビューには、ソフトウェア・セキュリティーのベスト・プラクティスと規制要件への準拠性を測定する包括的な監査レポートに基づく検出結果が表示されます。
カスタム検出結果の作成		このボタンは、AppScan Source for Analysis でのみ使用できます。このボタンを選択すると「カスタム検出結果の作成」ダイアログ・ボックスが開きます。このダイアログ・ボックスで、カスタム検出結果を現在の評価に追加することができます。

表 38. ツールバー・ボタン (続き)

アクション	アイコン	説明
選択した検出結果の保存		1 つ以上の検出結果を選択している場合、このボタンは「選択した検出結果の保存」ダイアログ・ボックスを開きます。このダイアログ・ボックスで、選択した検出結果を新しい評価ファイルに保存することができます。
「表示」メニュー		このメニューを使用すると、すべてのツールバー・ボタンのアクションに迅速にアクセスできます。

「検出結果」ビューでは、以下の操作を実行できます。

- コード・エディターで検出結果を開く
- 除外を作成する
- 検出結果を変更する
- 異なるグループ化によって検出結果を表示する
- 検出結果で特定の項目を検索する

AppScan Source for Analysis では、このビューの使用中に以下の操作も行うことができます。

- 検出結果をバンドルに移動する
- 障害追跡システムに障害を送信する
- カスタム検出結果を作成する
- 検出結果レポートの生成
- 検出結果またはバンドルを E メールで送信する

「除外された検出結果」ビュー

「除外された検出結果」ビューには、除外された検出結果のみが表示されます。 除外された検出結果とは、スキャンから除外した検出結果です。このビューでは、特定の検出結果を検索できます。 このビュー内の列は、「検出結果」ビュー内の列と同一です。

除外された検出結果を再度組み込む場合は、 185 ページの『除外としてマークされた検出結果の再組み込み』の説明に従ってください。

「変更された検出結果」ビュー

「変更された検出結果」ビューには、現行アプリケーションの、変更されたすべての検出結果が含まれています。 変更された検出結果 とは、脆弱性タイプ、重大度、分類、または注が変更された検出結果です。 失われた検出結果 (現在開いている評価には存在しない検出結果) は、緑のイタリック体で表示され、変更できません。

このビューでは、以下の操作を実行できます。

- 特定の検出結果を検索する
- 追加の変更を加える

AppScan Source for Analysis では、このビューで以下のアクションを行うこともできます。

- 検出結果をバンドルに追加する
- 障害追跡システムに障害を送信する
- 検出結果 (障害) を E メールで送信する
- 検出結果レポートの生成

「修正された/存在しない検出結果」ビュー

「修正された/存在しない検出結果」ビューでは、バンドル内にはあっても、現在の評価には含まれない検出結果を示します。 検出結果が修正された/存在しない検出結果として識別されるのは、それが解決されたか、削除されたか、またはソース・ファイルがスキャンされなかったためです。

「検索結果」ビュー

検出結果を検索すると、結果が「検索結果」ビューに表示されます。

このビューでは、以下の操作を実行できます。

- 検出結果をソートする
- 内部エディターまたは外部エディターでコードを編集する
- 脆弱性タイプを設定する
- 「要確認」およびスキャン範囲の検出結果を「確定」に昇格する
- 重大度レベルを設定する
- 検出結果に注釈を付ける
- 特定の検出結果を除外する
- 後続の検索を実行する

AppScan Source for Analysis では、このビューの使用中に以下の操作も行うことができます。

- 検出結果をバンドルに追加する
- 障害追跡システムに障害を送信するか、検出結果を E メールで送信する
- 検出結果レポートの生成

「検索結果」ビューには、検索基準に一致する項目のみが含まれ、最大 5 回までの検索の結果を保持します。 例えば、「検索結果」ビューでバッファー・オーバーフローの脆弱性タイプを検索し、次に「確定」の分類を検索すると、検索結果は両方の検索の重なる部分になります。

検索基準は、「検索」フィールドに `"<keyword>" in <originating_view>: <fields searched>` のような検索の表現として表示されます。例えば、`"shutdown" in Findings [Context, API, Method]` となります。 現在の評価を閉じた場合、すべての検索結果は破棄され、「検索」フィールドには、**「現行の検索はありません」**というテキストが表示されます。

「レポート」ビュー

「レポート」ビューで、ソフトウェア・セキュリティーのベスト・プラクティスと規制要件への準拠性を測定するさまざまな監査レポートに基づいてスキャン結果を編成することができます。

このビューには、以下のレポートに基づいた検出結果が表示されます。

- 248 ページの『CWE/SANS Top 25 2011 レポート』
- 248 ページの『DISA Application Security and Development STIG V3R10 レポート』
- 249 ページの『Open Web Application Security Project (OWASP) Mobile Top 10 レポート』
- 249 ページの『Open Web Application Security Project (OWASP) Top 10 2013 レポート』
- 249 ページの『Payment Card Industry Data Security Standard (PCI DSS) バージョン 3.2 レポート』
- 249 ページの『Software Security Profile レポート』

AppScan Source for Analysis を使用してカスタム・レポートを作成し、`<data_dir>¥reports¥profile` (`<data_dir>` は、ご使用の AppScan Source プログラム・データの場所です。詳しくは、 360 ページの『インストールとユーザー・データ・ファイルの場所』 を参照してください。) に保存すると、「レポート」ビューでカスタム・レポートによる検出結果を表示することもできます。

「レポート」ビュー内の列は、 343 ページの『「検出結果」ビュー』内の列と同一です。

検出結果の検索

検出結果を含む複数のビューで、特定の検出結果を検索できます。 検索基準には、バンドル、コード、ファイル、プロジェクト、または脆弱性タイプが含まれます。検索結果が「検索結果」ビューに表示されます。

コードの検索時に、以下のような複数の項目またはすべての項目を検索の対象とすることができます。

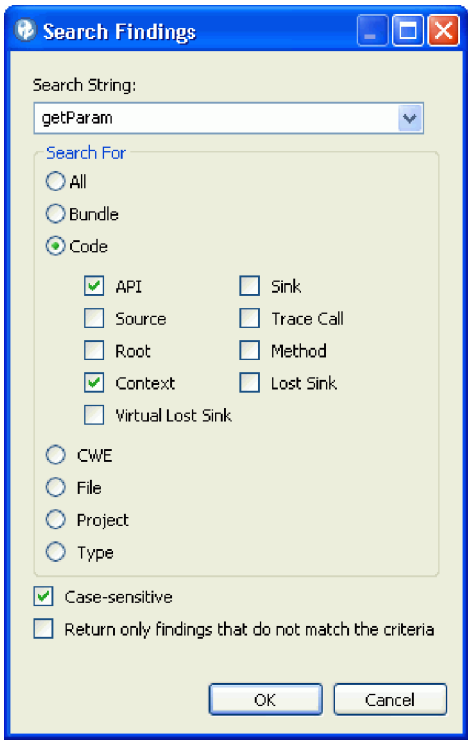
- API
- コンテキスト
- 方法
- ソース
- シンク
- 逸失シンク
- ルート
- トレース呼び出し

すべての検出結果での、ある項目のすべての出現箇所の検索**:**

手順

1. 検索を行うビューを選択します。

2. メインメニューから「編集」 > 「検索」を選択します (AppScan Source for Development (Eclipse プラグイン) では、「編集」 > 「検索/置換」を選択し、AppScan Source for Development (Visual Studio プラグイン) では検出結果が表示されているビューで「検索」ボタンをクリックします)。



3. 「検出結果の検索」ダイアログ・ボックスで検索文字列を入力します。
4. その文字列を、「バンドル」、「コード」、「**CWE**」、「ファイル」、「プロジェクト」、「タイプ」、または「すべて」で検索します。 一致する検出結果が「検索結果」ビューに表示されます。

   大文字と小文字を区別してテキストを検索するには、「大文字と小文字を区別」を選択します。

   AppScan Source for Analysis または AppScan Source for Development (Eclipse プラグイン) を使用する場合、検索基準に対応しない検出結果を返すには、「基準に一致しない検出結果のみを返す」を選択します。

検出結果表での検出結果の検索**:**

手順

1. ツールバーで「検索」をクリックします。
2. 検索の特性を指定し、「**OK**」をクリックします。

検出結果ツリーでの検索**:**

手順

1. ツールバーで「検索」をクリックします。
2. 検索の特性を指定し、「**OK**」をクリックします。

タスクの結果

「検出結果」ビューでは、表示される検出結果のサブセットの中で検索を行うこともできます。 例えば、「脆弱性タイプ」などの特定のサブセットで検出結果を検索したい場合があります。

「ソースとシンク」ビュー

「ソースとシンク」ビューでは、入力および出力のトレースに基づいて検出結果を表示することができます。

「ソースとシンク」ビューは、以下の 3 つのセクションに分かれています。

- ソースとシンク: 左側のパネルに、以下の 3 つの最上位ノードがあります。
  - ソース: ソースは、プログラムに対する入力情報です。ソースには、ファイル、サーブレット要求、コンソール入力、ソケットなどがあります。 多くの入力ソースの場合、内容と長さについて制限のないデータが返されます。 チェックされていない入力については、汚染されているものと見なされます。ソースは、すべての検出結果表の「ソース」列に表示されます。
  - シンク: シンクは、データの書き込み先となる任意の外部フォーマットです。シンクの例としては、データベース、ファイル、コンソール出力、ソケットなどがあります。 データをチェックせずにシンクに書き込むと、重大なセキュリティー脆弱性となる可能性があります。
  - 逸失シンク: 逸失シンクとは、トレースできなくなった API メソッドのことです。

  各ノードは、影響を受けるパッケージを表示するように展開することができます。 次に、今度はパッケージを展開して、影響を受けるクラス、さらにメソッドも表示することができます。これらのメソッドを展開すると、トレースの反対側にあるパッケージ、クラス、およびメソッドを表示できます。 例えば、特定のシンクに関心がある場合、**Sinks** ルートの下のメソッドにドリルダウンすることができます。 そこに達すると、そのメソッドの下のツリーには、そのシンクに通じたすべてのソースに戻るパスが示されます。

```
- Sources
 - packageA
  - classA
   - methodA
    - packageB
     - classB
      - methodB (at opposite end of trace)
- Sinks
 - packageB
  - classB
   - methodB
    - packageA
     - classA
      - methodA
- Lost Sinks
```

  このツリー・ビューでの選択内容によって、ビューの他の 2 つのセクションに表示される内容が決まります。

- 中間ノード: ビューのこのセクションには、「ソースとシンク」セクションでの選択内容に該当するトレースのすべての中間ノードを結合したものが表示されます。 ここで、検出結果表に表示される内容を詳細化することができます。

このセクションは、デフォルトでは表示されません。 「中間呼び出し表の表示/非表示」をクリックすると、表示したり、再度非表示にしたりすることができます。

1 つのパッケージ、クラス、またはメソッドの検出結果のみを表示するには、その「必要」列のチェック・ボックスを選択します。 1 つのパッケージ、クラス、またはメソッドの検出結果をフィルタリングで除外するには、その「削除」列のチェック・ボックスを選択します。このセクションで行ったフィルター設定を使用して、新規フィルターを作成できます。

使用例:「ソースとシンク」セクションで以下のツリー・ノードがあるものとします。

```
- Sources
 - java.util
  - Properties
   - getProperty
```

`getProperty` が選択された場合、検出結果表には、`getProperty` をソースとするトレースを含む検出結果のみが表示されます。 この時点で、中間ノード・セクションには、ソースが `getProperty` であるすべてのトレースのすべての中間ノード (トレース内の、ソースおよびシンク以外のノード) が表示されます。ただし、トレースが特定の API を通過するかどうかには関心がないこともあります。 例えば、`getProperty` からのデータが有効であることを保証する検証ルーチンがあるので、この検証ルーチンを通過するトレースは表示させたくない場合があります。中間ノード・セクションには、この検証ルーチンが含まれます。これがトレースの中間ノードであるためです。 中間ノード・セクションで検証ルーチンを参照し、その「削除」チェック・ボックスをクリックします。 これにより、検出結果表から、この中間ノードを通過するトレースを含むすべての検出結果が削除されます。

- 検出結果: このセクションには、 343 ページの『「検出結果」ビュー』および検出結果を含む他のビューにある物と同じ 340 ページの『検出結果表』 (および関連アクション) が含まれています。 ビューの他の 2 つのセクションで表示対象として選択したソース、シンク、および中間ノードの検出結果が表示されます。

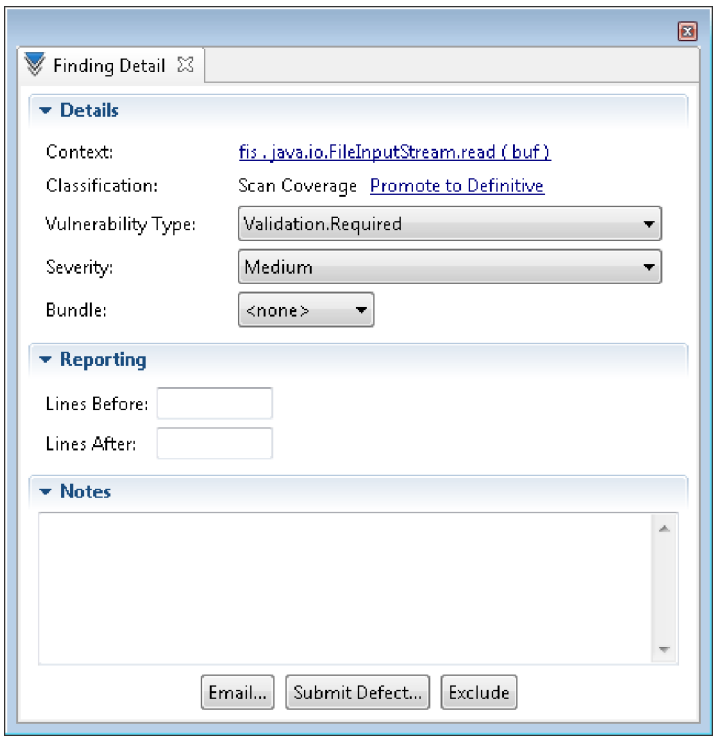
単一の検出結果の調査に使用できるビュー

このセクションのビューは、単一の検出結果を調査するために使用されます。

- 193 ページの『「検出結果の詳細」ビュー』
- 353 ページの『「修復支援」ビュー』
- 353 ページの『「トレース」ビュー』

「検出結果の詳細」ビュー

検出結果を選択すると、「検出結果の詳細」ビューが表示され、そのプロパティーを変更できます。このビューでは、個々の検出結果を変更できます。



- 193 ページの『詳細セクション』
- 194 ページの『レポート作成セクション (AppScan Source for Analysis および AppScan Source for Development (Eclipse プラグイン) でのみ使用可能)』
- 194 ページの『注釈セクション』
- 194 ページの『「検出結果の詳細」ビューのアクション』
- 195 ページの『カスタム検出結果の「検出結果の詳細」ビュー (AppScan Source for Analysis でのみ使用可能)』

詳細セクション

- コンテキスト: 脆弱性のある部分を囲むコードのスニペット
- 分類: 「確定」または「要確認」のセキュリティー検出結果あるいは「スキャン範囲」検出結果 (分類が変更された場合に検出結果を 「確定」に昇格するか、元の値に戻すためのリンク付き)
- 脆弱性タイプ
- 重大度: 高、中、低、または情報
- バンドル: 検出結果を含むバンドルの名前 (AppScan Source for Development (Visual Studio プラグイン) では使用不可)

レポート作成セクション (AppScan Source for Analysis および AppScan Source for Development (Eclipse プラグイン) でのみ使用可能)

レポート内の検出結果の前または後 (またはその両方) に組み込むコードの行数を指定します。

注釈セクション

検出結果に注釈を付けます。

「検出結果の詳細」ビューのアクション

- 除外: 検出結果を検出結果表から除外 (削除) するには、「除外」をクリックします。 除外された検出結果を表示するには、「除外された検出結果」ビューを開きます。
- AppScan Source for Analysis でのみ使用可能:
  - **E** メール: E メール設定を構成した場合、検出結果バンドルを直接開発者に E メールで送信し、スキャン後に検出された潜在的な障害について通知することができます。 この E メールには、検出結果を含むバンドル添付ファイルが含まれ、E メール・テキストでは検出結果を説明しています。
    1. 「検出結果の詳細」ビューで現在の検出結果を E メールで送るには、「**E** メール」をクリックします。
    2. 「添付ファイル名」ダイアログ・ボックスで、E メールに添付される検出結果バンドルの名前を指定します。 例えば、「添付ファイル名」フィールドで `my_finding` と指定すると、ファイル名が `my_finding.ozbdl` のバンドルが E メールに添付されます。
    3. 「**OK**」をクリックすると、「検出結果の E メール送信」ダイアログ・ボックスが開きます。 デフォルトで、「検出結果の E メール送信」ダイアログ・ボックスの「宛先」フィールドには、E メール設定で指定されている「宛先アドレス」が入力されていますが、これは E メールの作成時に容易に変更できます。このダイアログ・ボックスで、E メールの内容を確認してから「**OK**」をクリックして E メールを送信します。
  - 障害の送信: 検出結果を障害として送信するには、「障害の送信」をクリックします。 これにより、「障害追跡システムの選択」ダイアログ・ボックスが開きます。
    - 「**ClearQuest**」を選択し、「**OK**」をクリックすると、「添付ファイル名」ダイアログ・ボックスが開きます。そこで、障害に添付される検出結果バンドルの名前を指定してから、「**OK**」をクリックします。 Rational ClearQuest にログインして、検出結果を送信します。
    - **Quality Center** を選択して「**OK**」をクリックすると、ログイン・ダイアログ・ボックスが開き、Quality Center にログインして検出結果を送信できます。
    - いずれかの「**Team Foundation Server**」オプションを選択すると、ダイアログ・ボックスが開き、障害追跡システムにログインしてその他の構成詳細を入力するように求めるプロンプトが出されます。

    注**:** Rational Team Concert は、macOS でサポートされている唯一の障害追跡システムです。

カスタム検出結果の「検出結果の詳細」ビュー **(AppScan Source for Analysis** でのみ使用可能**)**

カスタム検出結果の「検出結果の詳細」ビューには、以下の編集可能な追加情報が表示されます。

- ファイル
- 行
- 列
- API

また、 193 ページの『詳細セクション』を編集する方法は、一部のフィールドについては標準的な検出結果と異なります (例えば、カスタム検出結果の分類はリスト形式で表示されます)。

「修復支援」ビュー

AppScan Source セキュリティー・ナレッジ・データベースでは、それぞれの脆弱性について、コンテキスト固有の情報が提供されます。 ナレッジベース・データベース によって、脆弱性の内容、安全ではない理由、解決方法、および今後回避する方法を知ることができます。 ソース・コードをスキャンすると、ナレッジベース・データベース はミッション・クリティカルなアプリケーションからリスクを排除するために必要な固有情報を提供します。 ナレッジベース・データベース の修復のアドバイスが、「修復支援」ビューに表示されます。 スキャンすると、ナレッジベース・データベース はミッション・クリティカルなアプリケーションからリスクを排除するために必要な固有情報を提供します。

ナレッジベース・データベース を表示して修復のアドバイスを得るには

- 検出結果表で検出結果を 1 つ選択してから、ナレッジベース・データベース ヘルプまたは「修復支援」ビューを開きます。
- AppScan Source for Analysis では、メニューから「ヘルプ」 > 「セキュリティー・ナレッジベース・データベース」を選択して、ナレッジベース・データベース 全体を表示することもできます。

データベース内の特定の API は、重大度レベルおよび重大度タイプをリストします。 例えば、`strcpy()` という API (バッファー・オーバーフロー・タイプ) は、重大度レベルが「高」です。 `strcpy()` では、出力先バッファーの長さを認識しないため、出力先バッファーが上書きされないようにするためのチェックを実行できないことから、出力先バッファーでのオーバーフローが起こりやすくなると説明されています。 長さのパラメーターをとる `strncpy()` を使用して、この問題を解決してください。

検出結果に、関連する共通脆弱性タイプ一覧 (CWE) ID がある場合、「修復支援」ビューから、`http://cwe.mitre.org/data/definitions/<CWE_ID>.html` にある CWE のトピック (`CWE: <id>`) へのハイパーリンクが表示されます。

「トレース」ビュー

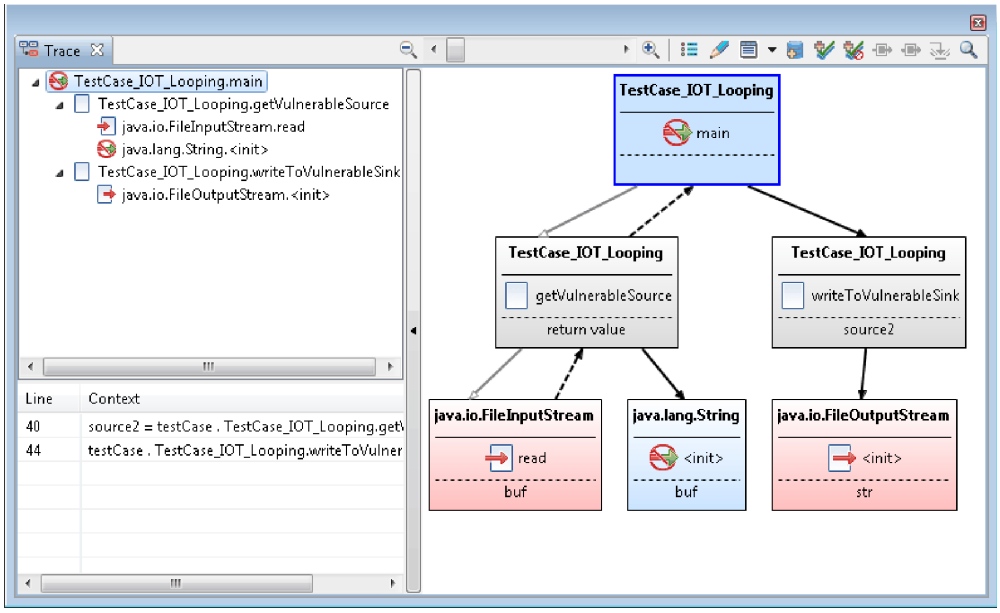
AppScan Source は、入出力の分析を実行し、それらの脆弱性を特定して表示します。 AppScan Source トレース・グラフを含む行を示すアイコンが、検出結果リストに表示されます。

「トレース」ビューでは、ルート・ノードが表示されます。ここで入力と出力のスタックが一緒になっています。 入力スタックは、汚染されたデータを提供することが分かっているソース につながる一連の呼び出しです。 出力スタックは、シンク

につながる一連の呼び出しです。 AppScan Source トレースは、分析されたコードが、保護されていないシンクへの保護されていないソースの使用を追跡できる場合に生成されます。

- ソース: ソースは、プログラムに対する入力情報です。ソースには、ファイル、サーブレット要求、コンソール入力、ソケットなどがあります。 多くの入力ソースの場合、内容と長さについて制限のないデータが返されます。 チェックされていない入力については、汚染されているものと見なされます。ソースは、すべての検出結果表の「ソース」列に表示されます。
- シンク: シンクは、データの書き込み先となる任意の外部フォーマットです。 シンクの例としては、データベース、ファイル、コンソール出力、ソケットなどがあります。 データをチェックせずにシンクに書き込むと、重大なセキュリティー脆弱性となる可能性があります。
- 逸失シンク: 逸失シンクとは、トレースできなくなった API メソッドのことです。

次の図に、入力スタックおよび出力スタックを通じたルートからの呼び出しシーケンスを示します。



図では、以下のようになっています。

- 中空の矢印は、既知の汚染されたデータ・フローを含まない呼び出しを示します。
- 中実の矢印は、汚染されている可能性のあるデータを含みます。 破線は戻りパスを示します。
- 実線が、メソッド呼び出しを表します。

ヒント:

- 「トレース」ビューで、グラフのトレース・ノードの上にマウスを移動すると、ノードに関する情報が表示されます。
- ビューの左側にある 2 つパネル (「入出力スタック」パネルと「データ・フロー」パネル) は、呼び出し図グラフを見やすくするために省略表示することがで

きます。これらのパネルを省略表示するには、「ツリー・ビューの非表示」矢印ボタンを選択してください。非表示になったこれらのパネルを表示するには、「ツリー・ビューの表示」矢印ボタンを選択します。

- スクロール・バーを移動してズームインまたはズームアウトすることにより、詳細情報を拡大表示したり、表示範囲を広げたりすることができます。ポインターをズーム・スクロール・バー上に移動すると、現在のズーム・レベルが示されます。最大レベルまでズームインするには、「ズーム率 **200%**」をクリックします。可能な限りズームアウトするには、「適合ズーム」をクリックします。

評価の操作に使用できるビュー

このセクションのビューは、評価のハイレベルな操作に使用されます。

- 『「評価の概要」ビュー』
- 356 ページの『「フィルター・エディター」ビュー』
- 357 ページの『「脆弱性マトリックス」ビュー』

「評価の概要」ビュー

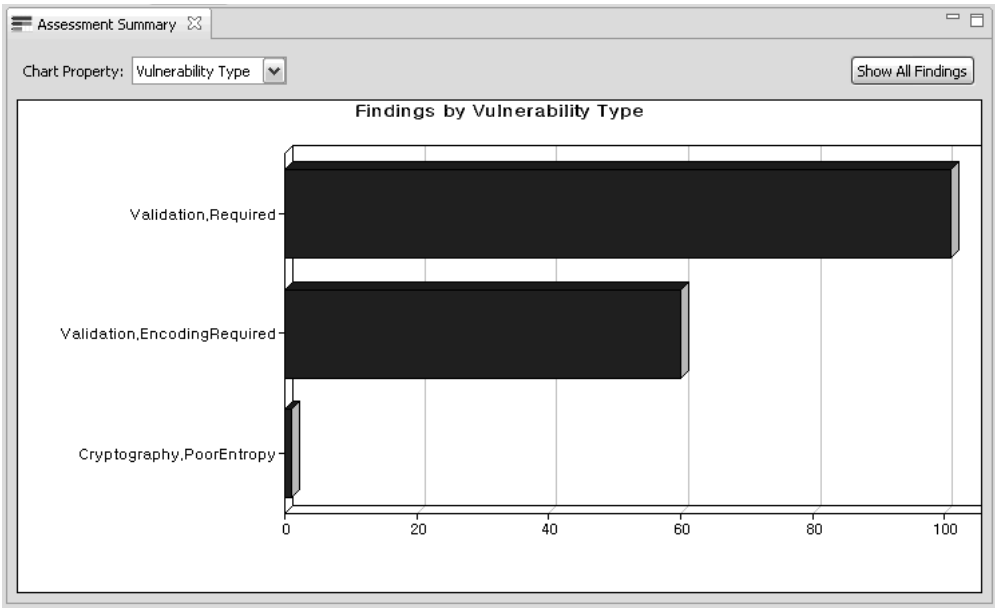
開いている評価を棒グラフでグラフィカルに示す「評価の概要」ビューには、選択された検出結果の情報が表示されます。

注**:**

- 「評価の概要」ビューは、macOS では使用できません。
- AppScan Source for Development (Visual Studio プラグイン) では、このビューは「フィルターの編集」ウィンドウの一部です。

グラフ・プロパティーによって、以下の内容を表示できます。

- 脆弱性タイプ: `Validation.Encoding` または `Injection.SQL` などの脆弱性タイプ
- **API**: 脆弱性が発生する API の名前
- プロジェクト: 複数のプロジェクトが存在する場合、プロジェクトごとの検出結果
- ファイル: 脆弱性が発生する個別のファイル



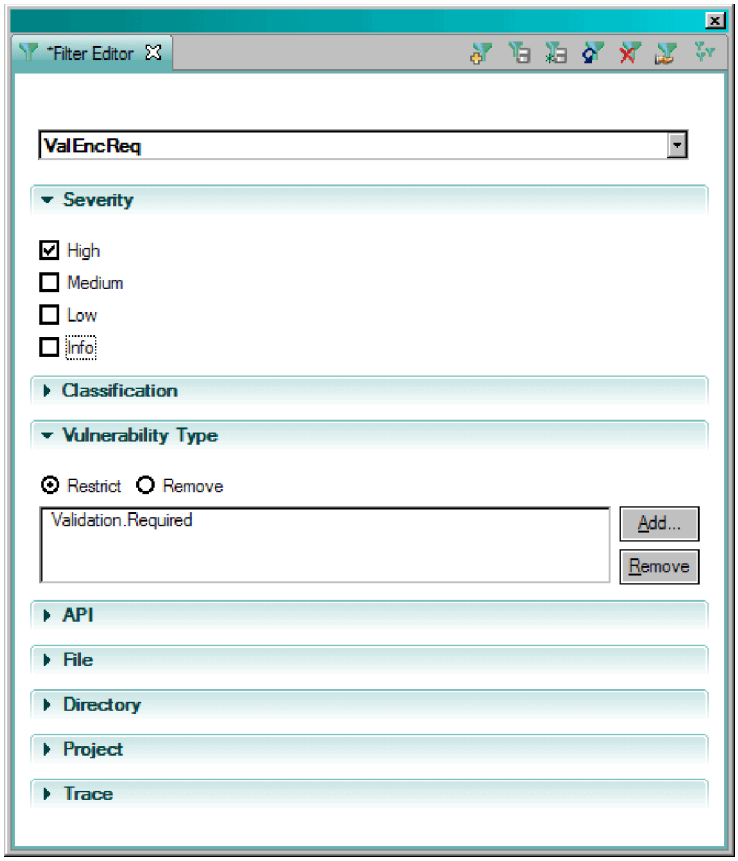
グラフをクリックすると、検出結果の詳細をドリルダウンして、トリアージを開始できます。

ヒント: 「評価の概要」ビューの棒グラフの上にマウスを移動すると、その棒グラフで表されている検出結果の正確な数が表示されます。

「フィルター・エディター」ビュー

「フィルター・エディター」ビューでは、現在選択されているフィルターを、AppScan Source ビューよりもきめ細かく操作することができます。このビューは、フィルタリングが可能なすべての基準で構成されています。

注: AppScan Source for Development (Visual Studio プラグイン) では、このビューは「フィルターの編集」ウィンドウの一部です。

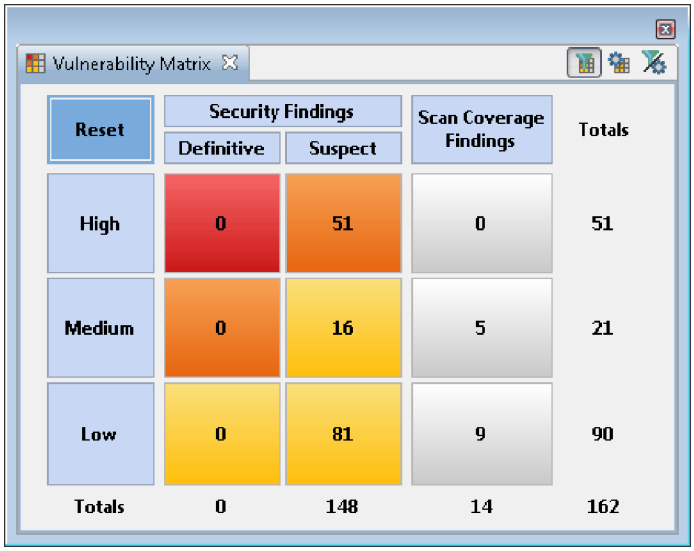


ヒント: 「フィルター・エディター」ビューの「トレース」セクションで、トレース項目の上にマウスを移動すると、項目に関する詳細が表示されます。

「脆弱性マトリックス」ビュー

「脆弱性マトリックス」ビューには、スキャンに含まれるすべてのアプリケーションの検出結果の総数が表示されます。検出結果を変更すると、マトリックスが更新されます。

注: AppScan Source for Development (Visual Studio プラグイン) では、このビューは「フィルターの編集」ウィンドウの一部です。



セキュリティー検出結果とスキャン範囲検出結果は、その検出結果を調査または対処する優先順位を示す色付きの四角形内に表示されます。

1. 「高」重大度の「確定」セキュリティー検出結果は、最も優先順位が高いことを指す赤い色で示されます。
2. 「中」重大度の「要確認」セキュリティー検出結果はオレンジ色で、次に処理する必要があります。
3. 以下のマトリックスのエントリーは黄色で示され、その次に検討する必要があります。
   - 「低」重大度の「確定」セキュリティー検出結果
   - 「中」および「低」重大度の「要確認」セキュリティー検出結果
4. スキャン範囲検出結果は、グレーの四角形内に表示され、最も低い優先順位が付与されます。

「脆弱性マトリックス」内のセル、行見出し、または列見出しをクリックすると、そのセル、行、または列内の結果のみを含むように現在のフィルターが更新されます。 すべての検出結果のビューに戻るには、「リセット」をクリックします。

「脆弱性マトリックス」ビューでは、ツールバー・ボタンによって、色付きの正方形に入っている数値が操作できます。 以下の項目を表示できます。

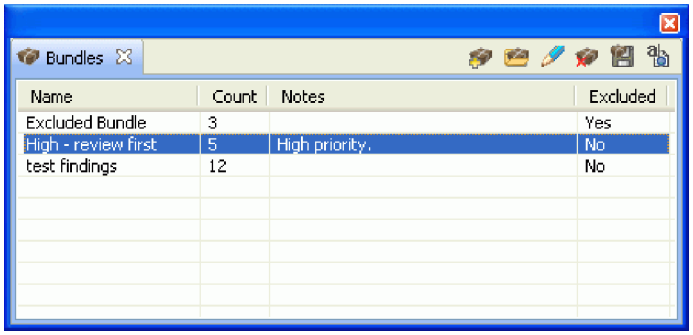
- フィルタリングされた検出結果のみのカウントおよび合計
- 検出結果のカウントおよび総数

  注: 「品質」検出結果、および「情報」重大度レベルに分類される検出結果は、「脆弱性マトリックス」ビューに含まれません。
- フィルタリングされた検出結果とすべての検出結果のカウントおよび総数

注: 「脆弱性マトリックス」ビューの外部で適用されるフィルターは、「脆弱性マトリックス」ビューに作用しない可能性があります。フィルターが「脆弱性マトリックス」ビューに反映されるようにするには、「脆弱性マトリックス」ビューの「フィルターに掛けた検出結果の数を表示」ツールバー・ボタンを選択する必要があります。

「バンドル」ビュー

「バンドル」ビューでは、新規バンドルの作成、バンドルへの検出結果の追加、バンドルおよび注の表示、バンドルの名前の変更、またはバンドルの削除を実行できます。 このビューには、バンドル名、バンドルに付加されたすべての注、バンドル内の検出結果の数、およびバンドルが除外されているかどうかが表示されます。 一度バンドルを開いて内容を表示させると、検出結果を他のバンドルに移動したり、検出結果を変更したり、コードを編集したり、バンドルを障害追跡システムに送信したりすることができます。



詳しくは、 187 ページの『バンドルを使用したトリアージ』を参照してください。

「バンドル」ビュー

「バンドル」ビューには、1 つのバンドル内のすべての検出結果が表示されます。バンドルは、AppScan Source for Analysis で作成される検出結果のセットです。

あるバンドル内のすべての検出結果を表示するには、「バンドル」ビューでバンドル名をダブルクリックします。 バンドル名は、「バンドル」ビュー内でタイトルとして表示されます。 バンドルをインポートし、その内容を「バンドル」ビューで表示することもできます。 バンドル内の検出結果を変更または削除することはできません。

「バンドル」ビューには、検出結果表と同様に、次の詳細情報が含まれています。

表 39. 「バンドル」ビューの列

列	説明
トレース	この列のアイコンは、逸失シンクまたは既知のシンクのトレースが存在することを示します。
ファイル	セキュリティー検出結果またはスキャン範囲検出結果が発生するコード・ファイルの名前。検出結果内のファイル・パスは、スキャンされたプロジェクト作業ディレクトリーからの相対パスです。

表 39. 「バンドル」ビューの列 (続き)

列	説明
分類	検出結果のタイプ: 「確定」または「要確認」セキュリティー検出結果 - または「スキャン範囲」検出結果。 **注:** 場合によっては、「なし」の分類を使用して、セキュリティー検出結果でもスキャン範囲検出結果でもない分類が示されることがあります。
重大度	• 高: データの機密性や保全性、可用性、および/または処理リソースの保全性や可用性にリスクをもたらします。 重大度の高い状態は、即時に修復されるように優先順位付けする必要があります。 • 中: データ・セキュリティーおよびリソース保全性にリスクをもたらしますが、攻撃の影響は比較的受けにくい状態です。重大度が中の状態は、可能であれば検討し修復します。 • 低: データ・セキュリティーおよびリソース保全性にもたらすリスクは最小です。 • 情報: 検出結果自体は、セキュリティーを侵害するわけではありません。これは、コードで使用されているテクノロジー、アーキテクチャー特性、またはセキュリティー・メカニズムについて説明するものです。
脆弱性タイプ	脆弱性カテゴリー (`Validation.Required` または `Injection.SQL` など)。
コンテキスト	脆弱性のある部分を囲むコードのスニペット。
呼び出し側メソッド	脆弱な呼び出しが行われている関数 (またはメソッド)。
CWE	コミュニティーが作成した、共通のソフトウェア脆弱性の辞書の ID およびトピック (共通脆弱性タイプ一覧 (CWE) のトピック)。
行	脆弱な API を含むコード・ファイル内の行番号。
注	この検出結果に追加された注。
障害 **ID**	障害追跡システムから受信した障害 ID。

インストールとユーザー・データ・ファイルの場所

AppScan Source をインストールすると、ユーザー・データ・ファイルおよび構成ファイルは、インストール・ディレクトリー以外の場所に保管されます。

- 361 ページの『デフォルトのインストール場所』
- 361 ページの『デフォルトの AppScan Source データ・ディレクトリー』

- 『AppScan Source 一時ファイルの場所』

デフォルトのインストール場所

AppScan Source をインストールすると、ソフトウェアは以下のいずれかのデフォルトの場所に配置されます。

- 32 ビット・バージョンの Microsoft Windows:
  `<SYSTEMDRIVE>:¥Program Files¥IBM¥AppScanSource`
- 64 ビット・バージョンの Microsoft Windows:
  `<SYSTEMDRIVE>:¥Program Files (x86)¥IBM¥AppScanSource`
- Linux: root ユーザーの場合は、インストール・ウィザードによって `/opt/ibm/appscansource` にソフトウェアがインストールされます。 root ユーザーではない場合は、デフォルトでは `<home_directory>/AppScan_Source` にインストールされる AppScan Source for Development Eclipse プラグインをインストールできます。
- macOS: `/Applications/AppScanSource.app`

重要:

- インストール・ディレクトリー名には、英文字のみを含めることができます。非英文字を含む名前があるフォルダーは許可されません。
- Windows にインストールする場合、AppScan Source コンポーネントをインストールするには管理者特権が付与されている必要があります。
- Linux にインストールする場合、AppScan Source サーバー・コンポーネントをインストールするには、root 特権が必要です。

デフォルトの AppScan Source データ・ディレクトリー

AppScan Source データは、構成ファイル、サンプル・ファイル、およびログ・ファイルなどの項目からなります。AppScan Source をインストールすると、データ・ファイルはデフォルトで以下の場所に配置されます。

- Microsoft Windows: `<SYSTEMDRIVE>:¥ProgramData¥IBM¥AppScanSource`

  注: `ProgramData¥` は隠しフォルダーです。このフォルダーを表示するには、エクスプローラーの表示設定を変更して、隠しファイルと隠しフォルダーが表示されるようにする必要があります。
- Linux: `/var/opt/ibm/appscansource`
- macOS: `/Users/Shared/AppScanSource`

AppScan Source データ・ディレクトリーの場所を変更する方法については、 362 ページの『AppScan Source データ・ディレクトリーの変更』を参照してください。

AppScan Source 一時ファイルの場所

AppScan Source 操作のなかには、一時ファイルが作成されるものがあります。これらの一時ファイルは、デフォルトで以下の場所に保管されます。

- Microsoft Windows: `<SYSTEMDRIVE>:¥ProgramData¥IBM¥AppScanSource¥temp`

注: ProgramData¥ は隠しフォルダーです。このフォルダーを表示するには、エクスプローラーの表示設定を変更して、隠しファイルと隠しフォルダーが表示されるようにする必要があります。

- Linux: /var/opt/ibm/appscansource/temp
- macOS: /Users/Shared/AppScanSource/temp

一時ファイルの場所は、常に、AppScan Source データ・ディレクトリー内の temp ディレクトリーになります。データ・ディレクトリーを変更することにより、一時ファイルの場所を変更できます。変更方法については、『AppScan Source データ・ディレクトリーの変更』を参照してください。これによって、ユーザーが選択したデータ・ディレクトリーに temp が配置されます。

AppScan Source データ・ディレクトリーの変更

ハード・ディスク・スペースを管理するために、AppScan Source データ・ディレクトリーの場所を変更することができます。AppScan Source のインストール後に、このトピックに記載されている手順に従って、データ・ディレクトリーの場所を変更できます。

始める前に

このタスクを実行する前に、すべての AppScan Source クライアント・アプリケーションを終了またはシャットダウンしていることを確認してください。AppScan Source クライアント・アプリケーションには、以下のものがあります。

- AppScan Source for Analysis
- AppScan Source for Development (Eclipse または Visual Studio プラグイン)(Windows および Linux でのみサポートされます)
- AppScan Source コマンド行インターフェース (CLI)
- AppScan Source for Automation

また、AppScan Source for Automation をインストールしている場合は、Automation Server が以下のようにしてシャットダウンされていることを確認してください。

- Windows の場合は、**IBM Security AppScan Source Automation** サービスを停止します。
- Linux の場合は、次のコマンドを実行します: /etc/init.d/ounceautod stop
- macOS の場合は、コマンド launchctl stop com.ibm.appscan.autod を発行します。

手順

1. APPSCAN_SOURCE_SHARED_DATA=<data_dir> 環境変数を定義します。ここで、<data_dir> は AppScan Source データを保管する場所です。

   注:

   - <data_dir> の場所は、AppScan Source がインストールされているマシンと同じマシン上の既存の完全な絶対パスでなければなりません。
   - <data_dir> ディレクトリー名には、英文字のみを含めることができます。非英文字を含む名前があるフォルダーは許可されません。

2. AppScan Source のインストール時に作成されたデフォルトのデータ・ディレクトリーを見つけます (データ・ディレクトリーのデフォルトの場所については、361 ページの『デフォルトの AppScan Source データ・ディレクトリー』を参照してください)。
3. デフォルト・データ・ディレクトリーのコンテンツを、環境変数で指定した <data_dir> の場所にコピーまたは移動します。
4. **AppScan Source for Automation** を **Linux** 上にインストールしている場合のみ、以下を行います。
   a. /etc/init.d/ounceautod ファイルを編集します。
   b. 以下の行を見つけます。

```
su - ounce -c
'export LD_LIBRARY_PATH="/opt/IBM/AppScan_Source/bin":$LD_LIBRARY_PATH &&
cd "/opt/IBM/AppScan_Source/bin" &&
"/opt/IBM/AppScan_Source/bin/ounceautod" -s' >>
"/var/opt/ibm/appscansource/logs/ounceautod_output.log" 2>&1 &
```

   この行を以下の行に置き換えます。

```
su - ounce -c
'export APPSCAN_SOURCE_SHARED_DATA=<new data directory path here> &&
export LD_LIBRARY_PATH="/opt/IBM/AppScan_Source/bin":$LD_LIBRARY_PATH &&
cd "/opt/IBM/AppScan_Source/bin" &&
"/opt/IBM/AppScan_Source/bin/ounceautod" -s' >>
"<new data directory path here>/logs/ounceautod_output.log" 2>&1 &
```

   注: 上記のコマンドは、1 行で入力してください。
   c. /etc/init.d/ounceautod ファイルを保存します。

次のタスク

AppScan Source for Automation をインストールしている場合は、以下のようにして Automation Server を開始します。

- Windows の場合は、**IBM Security AppScan Source Automation** サービスを開始します。
- Linux の場合は、次のコマンドを実行します: /etc/init.d/ounceautod start
- macOS の場合は、コマンド launchctl start com.ibm.appscan.autod を発行します。

第 15 章 CWE サポート

共通脆弱性タイプ一覧 (CWE) は、公開されているソフトウェア脆弱性に共通名を提供する業界標準リストです。 このトピックでは、AppScan Source の現行バージョンでサポート対象の CWE ID を示します。

スキャン時に、AppScan Source は以下の CWE 一覧の ID とその親 ID と子 ID を検索します。

表 40. CWE サポート

15、16、20、73、74、77、79、88、89、90、91、95、98
105、109、112、113、116、117、120、129、130、131、134、185、190
201、209、242、250、257、264、266、267、285、287、288、295
310、311、312、319、327、331、335、345、352、359、367、382、388、390、398
400、404、407、425、434、447、470、472、477、489、497
506、507、511、517、520、521、522、523、524、525、532、538、543、544、546、547、565、569、586
601、613、615、624、643、645

用語集

この用語集には、AppScan Source の用語と定義が記載されています。

この用語集では、以下の相互参照が使用されています。

- 「を参照」は、当該用語から、優先的に使用される同義語を参照します。または、頭字語や省略語から、定義が示されている完全な形式の用語を参照します。
- 「も参照」は、関連用語や対義語を参照します。

他の IBM 製品の用語集を参照するには、www.ibm.com/software/globalization/terminology にアクセスしてください。

A

アプリケーション **(application)**
特定の 1 つ以上のビジネス・プロセスを直接サポートする機能を提供する、1 つ以上のコンピューター・プログラムまたはソフトウェア・コンポーネント。

アセンブリー **(assembly)**
.NET Framework アプリケーションにおいてデプロイメント、バージョン管理、再利用、アクティブ化のスコープ設定、およびセキュリティー権限の設定を行うための単位となる、タイプおよびリソースのコレクション。

評価 **(assessment)**
コードをスキャンした結果として得られる検出結果のコレクション。ユーザーは、これを操作したり保存したりすることができます。また、他のユーザーと共有することもできます。

攻撃 **(attack)**
ソフトウェア・プログラムまたはネットワーク・システムの動作を妨げることを目的として無許可の個人によって試行される任意の行為。

属性 **(attribute)**
スキャン結果を意味のあるグループ (所属別、プロジェクト・リーダー別など) に編成するのに役立つ、アプリケーションの特性。

B

バンドル **(bundle)**
ユーザーが作成する検出結果のセット。 バンドルは、エクスポートして、複数のユーザーやアプリケーションの間で共有できます。

C

コールバック **(callback)**
特定のイベントが発生したことを、あるスレッドが別のアプリケーション・スレッドに通知するための方法。

呼び出しグラフ **(call graph)**
プログラム内のサブルーチン間でのデータの流れを線で表したグラフ。

クロスサイト・スクリプティング **(cross-site scripting)**
クライアントから提供されたデータを Web サイトでエコー出力させて、それをユーザーの Web ブラウザーで実行させる攻撃手法。

D

障害 **(defect)**
変更要求の一種で、作業成果物に含まれる異常または不備を識別します。

E

エンコード **(encode)**
コンピューター・セキュリティーの文脈においては、コード化のための特定の体系を使用して、理解不能な形式にプレーン・テキストを変換すること。

例外 **(exception)**
脆弱性が疑われる状況や潜在的な脆弱性が存在することを示します。この場合は、さらなる情報や調査が必要になります。

除外 **(exclusion)**
ユーザーがマークして無視できる検出結果。

F

フィルター **(filter)**
特定の特徴を持つ検出結果を定義するルールのセット。

検出結果 **(finding)**
セキュリティー侵害につながる不具合がコード内で検出されたことを示します。 AppScan は、検出結果を脆弱性と例外の 2 つのカテゴリーに分けます。

L

逸失シンク **(lost sink)**
トレースできなくなった API メソッド。

P

パースペクティブ **(perspective)**
ワークベンチにおいてリソースのさまざまな側面を表示するビューのグループ。

R

修復 **(remediation)**
問題の修正方法の提案。

S

スキャン **(scan)**
AppScan がアプリケーションを探索およびテストしてその結果を提供するプロセス。

パターン・ルール **(pattern rule)**
スキャン中に検索されるパターンまたは正規表現。

シンク **(sink)**
データの書き込み先となる任意の外部フォーマット。 シンクの例としては、データベース、ファイル、コンソール出力、ソケットなどがあります。

ソケット **(socket)**
TCP/IP が使用する通信ハンドル。

スタック **(stack)**
後入れ先出し (LIFO) の原則に基づいて管理されるメモリー内の領域。一般に、一時的な登録情報、パラメーターの値、サブルーチンの戻りアドレスなどの情報が格納されます。

T

汚染 **(taint)**
コードで処理することを許可されている、セキュアでないデータ。

トリアージ **(triage)**
検出結果を評価し、それらの解決方法を決定するプロセス。

V

V-Density
アプリケーションの脆弱性を評価するための一貫性のある方法を可能にする数式。 V-Density は、脆弱性と例外の数と重要性を、分析されているアプリケーションまたはプロジェクトのサイズに関係付けることによって計算されます。

脆弱性分析キャッシュ **(vulnerability analysis cache)**
ソース・コードのスキャン中に検出された脆弱性を格納するためのキャッシュ。以降のスキャンでは、スキャン時間を短縮するために、このキャッシュを使用することができます。

W

ワークベンチ **(workbench)**
Eclipse や、Eclipse ベースのツール (IBM Rational Application Developer など) で提供される、ユーザー・インターフェースおよび統合開発環境 (IDE)。

X

XSS 「クロスサイト・スクリプティング (cross-site scripting)」を参照。

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒103-8510
東京都中央区日本橋箱崎町19番21号
日本アイ・ビー・エム株式会社
法務・知的財産
知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。

IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態で提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。

国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。 本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。 IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。 そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性がありますが、その測定値が、一般に利用可能なシステムのものと同じである保証はありません。 さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。 お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。 IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確証できません。 IBM 以外の製品の性能に関する質問は、それらの製品の供給者にお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があり、単に目標を示しているものです。

表示されている IBM の価格は IBM が小売り価格として提示しているもので、現行価格であり、通知なしに変更されるものです。卸価格は、異なる場合があります。

本書はプランニング目的としてのみ記述されています。記述内容は製品が使用可能になる前に変更になる場合があります。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。 より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラット

フォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。お客様は、IBM のアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。

© (お客様の会社名) (西暦年).このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。 © Copyright IBM Corp. _年を入れる_. All rights reserved.

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

商標

IBM、IBM ロゴおよび ibm.com は、世界の多くの国で登録された International Business Machines Corporation の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。 現時点での IBM の商標リストについては、http://www.ibm.com/legal/copytrade.shtml をご覧ください。

Adobe、PostScript は、Adobe Systems Incorporated の米国およびその他の国における登録商標または商標です。

IT Infrastructure Library は英国 Office of Government Commerce の一部である the Central Computer and Telecommunications Agency の登録商標です。

インテル、Intel、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Intel Centrino、Intel Centrino ロゴ、Celeron、Intel Xeon、Intel SpeedStep、Itanium、および Pentium は、Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

ITIL は英国 Office of Government Commerce の登録商標および共同体登録商標であって、米国特許商標庁にて登録されています。

UNIX は The Open Group の米国およびその他の国における登録商標です。

Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。

Cell Broadband Engine は、Sony Computer Entertainment, Inc.の米国およびその他の国における商標であり、同社の許諾を受けて使用しています。

Linear Tape-Open、LTO、LTO ロゴ、Ultrium、および Ultrium ロゴは、HP、IBM Corp. および Quantum の米国およびその他の国における商標です。

索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

[ア行]

アプリケーション (application)
  アプリケーション・サーバーからのインポート 48
    アプリケーション・サーバーのインポート・フレームワークの拡張 289
    Liberty プロファイル用のプリコンパイル済み JSP の生成 50
  既存の追加 45
    ドラッグ・アンド・ドロップ 46
    ユーザー・インターフェース・アクション 45
  削除 95
  新規作成 40
  定義 21
  開く 40
  複数の追加 47
    ドラッグ・アンド・ドロップ 48
    ユーザー・インターフェース・アクション 47
アプリケーション除外 183
アプリケーション属性 95
アプリケーション・ディスカバリー・アシスタント 41
  デフォルトの除外ルール 45
逸失シンク (lost sink) 209, 211
インストール
  データの場所 136, 360
    変更 362
  ファイルの場所 136, 360
  Application Developer インポーター 52
  Eclipse インポーター 52
インターネット・プロトコル・バージョン 6 3
失われた検出結果 346
「エクスプローラー」ビュー 94, 96, 280, 312, 322
エディター
  外部 307
  内部 307
エラー・コンソール 336
エラー・ログ 308
エンコード 208
エンコード・ルーチン 215
汚染されたコールバック 261
汚染伝播元 261
オンライン・ヘルプ 308

[カ行]

該当項目のないルール 272
外部エディター 202, 214, 307
  ノートパッド 202, 214
  Eclipse 202, 214
  vi 202, 214
  Visual Studio.NET 202, 214
確定 262
カスタム検出結果 198, 279, 320
  「検出結果」ビューでの作成 200
  ソース・コード・エディターでの作成 201
  「プロパティー」ページでの削除 200
  「プロパティー」ページでの作成 199
  「プロパティー」ページでの変更 200
「カスタム検出結果」ビュー 340
カスタム・ルール
  汚染されたコールバック 261
  汚染伝播元 261
  汚染の可能性なし 261
  検証/エンコード・ルーチン 260, 261
  情報 261
  シンク (sink) 261
  ソース 261
  トレースなしの検出結果 261
  likelihood 属性 266
「カスタム・ルール」ビュー 260, 268, 311
カスタム・ルール・ウィザード 260
カスタム・レポート 251
  カテゴリーの組み込み 257
    検出結果の追加 257
    バンドルの追加 257
    プロパティーの追加 257
  生成 256
  設計 256
  テンプレートの保存 258
  プレビュー 258
カスタム・レポートの作成 246
カテゴリー 251, 332
「管理」メニュー 307
共通脆弱性タイプ一覧 158, 243
共通脆弱性タイプ一覧のサポート 365
共有フィルター 160, 176
グローバル除外 183
グローバル属性 94, 95, 320
グローバル・フィルターの適用 182
検索、検出結果の 209, 348
「検索結果」ビュー 346
検出結果
  解決済み/存在しない 197
    カスタム 198
  カスタム 198, 279, 320
    「検出結果」ビューでの作成 200
    ソース・コード・エディターでの作成 201
    「プロパティー」ページでの削除 200
    「プロパティー」ページでの作成 199
    「プロパティー」ページでの変更 200
  共通の 197
  検索 347
    ある項目のすべての出現箇所 347
    検出結果表での 348
  「検出結果の詳細」ビューでの変更 193
  検出結果表からの変更 191
    重大度の変更 192
    脆弱性の変更 192
    注釈付け 192
    分類の昇格 192
  「差分評価」ビューでの比較 197
  重大度の変更 192
  除外された 279, 308, 320
  存在しない 308, 337
  「バンドル」ビューでの注釈付け 191
  比較 197
  表 202, 214
  表示 158
  分類 22
  分類の昇格 192
  変更 191
  変更された 160, 279, 308, 320
  変更の取り消し 195
  メインメニューからの比較 197
  new 197
「検出結果」ビュー 343
「検出結果の詳細」ビュー 193, 351
検出結果レポート 243
検証 208
  呼び出しサイト単位 216
  API 単位 216
検証ルーチン 215

検証ルーチン (続き)
追加 259
呼び出しサイト単位 226
API 単位 226
検証/エンコード・ルーチン 261
コード例 218
例 1: ソースからシンク 218
例 2: 検証ルーチンとエンコード・ルーチンの作成
カスタム・ルール・ウィザード 224
「トレース」ビュー 221
例 2: ソースからシンクへの変更 220
例 3: ソースとシンクのファイルが異なる場合 225
例 4: 詳細な検証 226
コード・スニペット 243
「公開された評価」ビュー 145, 147, 338
削除 146
構成 35, 297
アプリケーション 39
プロジェクト 52
「構成」パースペクティブ 297
コンソール
エラー 336
出力 336
「コンソール」ビュー 336
コンパイラー
JSP 117
Tomcat 117
WebLogic 117
WebSphere Application Server 117

[サ行]

作業環境 297
削除ルール 163
差分評価 160, 197
「差分評価」ビュー 160, 339
サンプル 295
自動登録 144
修正された/存在しない検出結果 189
カスタム 198
「修正された/存在しない検出結果」ビュー 346
重大度 158, 262
「修復支援」ビュー 353
出力コンソール 336
障害追跡 229
E メール 240
HP Quality Center 234
検出結果情報 235
送信、検出結果の 234
追跡、検出結果の 234
IBM Rational ClearQuest 235
障害の送信 236
障害追跡 (続き)
Rational ClearQuest
障害の保存 239
送信、検出結果の 236
Rational Team Concert 237
障害の送信 237
SSL 証明書 114, 238
Team Foundation Server 238
障害の送信 239
情報 261
除外 160, 183, 279, 280, 319, 320, 322, 345
アプリケーション (application) 183
グローバル 183
検出結果表で検出結果にマーク付け 184, 185
指定 184
バンドル (bundle) 183, 187
フィルター 183
フィルターの指定 185
例 1 185
例 2 186
プロジェクト 183
除外された検出結果 279, 308, 320
「除外された検出結果」ビュー 345
新規アプリケーション構成ウィザード 39
新機能 4
新規プロジェクトの追加 56, 57, 58
シンク (sink) 209, 353
スキャン 119
スキャン構成 124
スキャン (scan)
増分 132
「スキャン」メニュー 306
スキャンからのファイルの除外 135
「スキャン構成」ビュー 129, 276, 329
スキャンのキャンセル 136
ステータス・バー 310
正規表現 268, 269, 271, 272
egrep 272
grep 271
Perl 271
制限ルール 163
脆弱性 (vulnerability)
定義 21
脆弱性タイプ 163
脆弱性マトリックス 180, 357
「脆弱性マトリックス」ビュー 357
製品 1
製品の概要 20
絶対パス 152
設定 103, 304
アプリケーション・サーバー 107
障害追跡システム 110, 229
サーバー名の変更 115
HP Quality Center 111, 230
設定 (続き)
障害追跡システム (続き)
Rational ClearQuest 110, 229
Rational Team Concert 114, 115, 233
Team Foundation Server 115, 233
全般 103
ナレッジベース・データベースの記事 117
プロジェクト・ファイル拡張子 117
AppScan Enterprise Console 106, 150
E メール 116
Eclipse インポーター 53, 115
HP Quality Center 111, 230
フィールドのカスタマイズ 113, 232
Java 117
JavaServer Page 117
JSP 117
Rational ClearQuest 110, 229
Rational Team Concert 114, 233
サーバー名の変更 115
Team Foundation Server 115, 233
Tomcat 7 108
WebLogic 11 108
WebLogic 12 108
WebSphere 109
ソース 209, 353
ソース・ルート 65, 77
ソート順 158
重大度 158
分類 158
属性 279, 280, 319, 320, 322
アプリケーション (application) 95, 279, 280, 320, 322
グローバル 95, 279, 280, 320, 322
作成 322
定義 21
属性のサポート 203
存在しない検出結果 308

[タ行]

注釈のサポート 203
「ツール」メニュー 307
ツールバー 309
データ・フロー 211
テキスト・パターン
定義 272
テキスト・パターン脆弱性 268, 269
デバッグ情報 65
デフォルト JDK 117
デフォルトのインストール・ディレクトリー 136, 360
トリアージ 157
除外を使用した 183

トリアージ (続き)
バンドルを使用した 187
プロセス 160
例 161
「トリアージ」パースペクティブ 297
トレース 207, 267
検索 209
スキャン結果 208
「トレース」ビュー 210, 353

[ナ行]

内部エディター 202, 214, 307
ナレッジベース・データベース 1, 308, 353
ナレッジベース・データベース 管理 259
権限 259
入出力トレース 209
入出力分析 353
ノートパッド 202, 214

[ハ行]

パースペクティブ 297
トリアージ 297
分析 297
「パースペクティブ」メニュー 308
バグ・フィールドの自動ロード 111, 230
パターン 268, 269, 279, 280, 320, 322
検索、テキスト 272
パターン・ベースのスキャン 279, 280, 320, 322
パターン・ベース分析 75
パターン・ルール 259, 268, 269
削除 274
定義 271
適用 275
「パターン・ルール・ライブラリー」ビューでの作成 272
変更 274
パターン・ルール・セット
削除 270
作成、「スキャン・ルール・ライブラリー」ビューでの 269
適用 275
変更 270
「パターン・ルール・ライブラリー」ビュー 271, 318
バンドル 21, 160, 187, 279, 320
検出結果 189
検出結果の移動 189
検出結果の追加 188
検出結果の表示 189
作成 187
「検出結果」ビューでの 188
バンドル (続き)
作成 (続き)
「バンドル」ビューでの 188
修正された/存在しない検出結果 189
除外された 279, 320
除外済み 183, 187
ディスパッチング 191, 239
保存 190
「バンドル」ビュー 359
バンドルの作成 187
「検出結果」ビューでの 188
「バンドル」ビューでの 188
ビュー 311
エクスプローラー 94, 96, 280, 312, 322
カスタマイズ 342
検出結果表 340
カスタム検出結果 340
カスタム・ルール 260, 268, 311
検索結果 346
検出結果 343
検出結果の詳細 193, 351
検出結果を含む 340
公開された評価 338
構成 311
コンソール 336
差分評価 160, 339
修正された/存在しない検出結果 346
修復支援 353
除外された検出結果 345
スキャン構成 129, 276, 329
スキャン出力 336
脆弱性マトリックス 357
ソースとシンク 349
単一の検出結果の調査 350
トリアージ 339
トレース 210, 353
パターン・ルール・ライブラリー 271, 318
バンドル 359
評価 (assessment) 355
「評価の概要」 355
フィルター・エディター 356
プロパティー 94, 319
変更された検出結果 346
マイ評価 338
管理 137
メトリック 337
レポート 347
評価
公開 144
削除 153
「差分評価」ビューでの比較 197
自動保存 153
保存 152
評価 (続き)
「マイ評価」ビューまたは「公開された評価」ビューからの比較 197
評価 (assessment) 21, 119
クラウド分析 138
公開済み 119, 144
比較 160, 339
保存済み 119
「評価の概要」ビュー 355
評価の公開 119, 144, 299
AppScan Enterprise Console 147
AppScan Source 145
削除 146
評価の削除 153
評価の比較 160, 339
評価の保存 119, 152
自動的に 153
「表示」メニュー 308
「ファイル」メニュー 299
ファイル・プロパティー 329
フィルター 160
共有 160, 176
公開 146
作成 175
「ソースとシンク」ビューでの 181
「評価の概要」ビューからの 178
「フィルター・エディター」ビューでの 176
事前定義 169
事前定義アーカイブ 173
アクセス権限 174
脆弱性タイプ 163
「脆弱性マトリックス」ビューからの 180
定義 163
適用 182
グローバル 182
判別 183
ローカル 160
フィルター除外 183
フィルターの作成 175
「ソースとシンク」ビュー 181
「評価の概要」ビューからの 178
フィルター・エディター 176
フィルターの適用 182
「フィルター・エディター」ビュー 180, 356
吹き出しヘルプ 309
プリコンパイル済み Java クラス・ファイル 65
プレビュー 251, 332
プロジェクト
アプリケーションへの追加 54
既存の追加 55
ドラッグ・アンド・ドロップ 56

プロジェクト (続き)
既存の追加 (続き)
ユーザー・インターフェース・アクション 55
コピー 93
削除 95
定義 21
パターン分析
追加 75
複数の追加 56
ドラッグ・アンド・ドロップ 58
ユーザー・インターフェース・アクション 57
変更 93
Arxan
追加 58
ASP
追加 59
Classic ASP 54
COBOL
追加 63
ColdFusion
追加 64
C/C++ 54
追加 61
Java 54
追加 65
JavaScript
追加 73
JSP 54
コンテンツの追加 71
Perl
追加 76
PHP
追加 77
PL/SQL
追加 90
T-SQL
追加 91
Visual Basic 54
追加 92
.NET アセンブリー
追加 74
プロジェクト依存関係 65, 77, 280, 322
プロジェクト除外 183
プロジェクトのコピー 299
プロジェクト・ファイル拡張子 117
プロパティー
ファイル (file) 329
「プロパティー」ビュー 94, 319
「分析」パースペクティブ 297
分類 158, 163, 262
確定 22
スキャン範囲 22
要確認 22
米国連邦情報・技術局 3
変更された検出結果 160, 279, 308, 320
「変更された検出結果」ビュー 346
「編集」メニュー 304
変数
定義 109, 154
公開時および保存時の 154
例 155
変数の定義
公開時および保存時の 154
例 155

[マ行]

マイグレーション 16
「マイ評価」ビュー 338
管理 137
未解決の PHP インクルード式 80, 85
メインメニュー 299
「メトリック」ビュー 337
メニュー
管理者 307
スキャン 306
ツール 307
パースペクティブ 308
表示 308
ファイル 299
編集 304
メイン 299
問題
解決 202

[ヤ行]

ユーザー管理 307
有効範囲 215
呼び出しサイト単位 215
API 単位 215
要確認 262
用語集 367
呼び出しグラフ (call graph) 211
呼び出しサイト単位 (有効範囲) 215
呼び出しサイト単位のルーチン 215

[ラ行]

ルール
削除 163
制限 163
ルール条件
重大度 163
信頼性 163
タイプ 163
列の選択と順序付け 234
レポート
検出結果 243
レポート (続き)
プロファイル 243
AppScan Source レポート 243
「レポート」ビュー 347
レポート・エディター 251, 332
カテゴリー 251, 332
「カテゴリー」タブ 254, 334
プレビュー 251, 332
「プレビュー」タブ 256, 336
レイアウト 252, 333
レポート・レイアウト 251, 332
レポート・エディター 251, 252, 332, 333
連邦情報処理標準 3
ローカル・フィルター 160

[ワ行]

ワークスペース
追加 51
ワークフロー 20
ワークベンチ (workbench) 297

A

API 単位 (有効範囲) 215
API 単位のルーチン 215
AppScan Enterprise Console の統合 147
AppScan Enterprise Server
パスワード変更 28
SSL 証明書 28
AppScan Source
アクセシビリティーの問題 29
製品ファミリー 1
AppScan Enterprise Server へのログイン 23
パスワード変更 28
CAC 26
SSL 証明書 28
for Analysis 1, 20, 122
概念 21
for Automation 1
for Development 1
AppScan Source 製品 1
AppScan Source セキュリティー・ナレッジ・データベース 1, 259, 353
AppScan Source トレース 207, 347
AppScan Source ファイル
epf 35
ewf 35
gaf 35
gpf 35
paf 35
ppf 35
AppScan Source レポート 245

Arxan プロジェクト 58
ASP コンテンツ・ルート 59
ASP プロジェクト 59

C

COBOL プロジェクト 63
ColdFusion プロジェクト 64
CQPerl 実行可能ファイル 235
CWE 158, 243, 342, 347, 353
CWE ID ハイパーリンク 243
CWE サポート 365
CWE/SANS Top 25 2011 レポート 248

D

DISA Application Security and Development 245, 248

E

E メール、検出結果 240
Eclipse 202, 214
egrep 272

F

FIPS 3

G

grep 268, 269, 271

H

HP Quality Center 110, 229, 234
検出結果情報 235
送信、検出結果の 234
追跡、検出結果の 234

I

IBM Rational ClearQuest 235
障害の送信 236
IPv6 3

J

JAR ファイル 71
Java API
構文要件 267
Java Development Kit 117, 304
Java クラス・ファイル 65
Java クラス・ファイル (続き)
プリコンパイル済み 65
Java プロジェクト依存関係 65
JavaScript ステートメント・グラフ 213
JavaScript プロジェクト 73
JavaServer Pages 304
JDK 65, 117, 280, 304, 322
デフォルト 65, 117
JSP 304
コンパイラー 117
JSP コンパイル 107
JSP ファイル構造 71
JSP プロジェクト 71
JSP プロジェクト依存関係 65

M

Microsoft Visual Studio 39

N

NIST 3

O

Open Web Application Security Project 245
Ounce/Ant 35, 54, 55
Ounce/Make 35, 54, 55
Ounce/Maven Plug-in 35
OWASP 245
OWASP Mobile Top 10 249
OWASP Top 10 2013 レポート 249

P

Payment Card Industry Security Standard レポート 243, 245
PBSA 75
PCI Data Security Standard レポート バージョン 3.0 249
PCI レポート 243
Perl 271, 272
Perl プロジェクト 76
PHP 文書ルート 77
PL/SQL プロジェクト 90

Q

Quality Center 234
検出結果情報 235
送信、検出結果の 234
追跡、検出結果の 234

R

RAD 52
Rational Application Developer for WebSphere Software (RAD) 52
Rational ClearQuest 110, 229
障害の保存 239
送信、検出結果の 236
Rational Team Concert 110, 229, 237
障害の送信 237
SSL 証明書 114, 238

S

SMTP メール・サーバー構成 116
Software Security Profile 245, 249
strncpy() 202, 353

T

Team Foundation Server 238
障害の送信 239
Tomcat 108
コンパイラー 117
T-SQL プロジェクト 91

V

vi 202, 214
Visual Studio.NET 202, 214
V-Density 262, 268, 337
V/KLoC 337

W

WAR ファイル 71, 202, 214
Web コンテキスト・ルート 65, 71, 280, 322
WebLogic 65, 108, 117, 280, 322
コンパイラー 117
WebSphere 109
WebSphere Application Server 117
コンパイラー 117
WEB-INF ディレクトリー 65, 71, 280, 322

[特殊文字]

.dsp 55
.ewf 39
.jsp 71
.jspx 71
.NET アセンブリー・プロジェクト 74
.ozasmt 152
.ozbdl 190, 239

.paf 40
.sln 39
.vcproj 55
.war 65, 280, 322

Printed in Japan

日本アイ・ビー・エム株式会社
〒103-8510 東京都中央区日本橋箱崎町19-21