

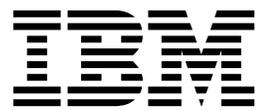
IBM Security AppScan Source
Utilities
Versão 9.0.3.7

Guia do Usuário



IBM Security AppScan Source
Utilities
Versão 9.0.3.7

Guia do Usuário



(C) Copyright IBM Corp. and its licensors 2003, 2017. All Rights Reserved.

IBM, o logotipo IBM, ibm.com, Rational, AppScan, Rational Team Concert, WebSphere e ClearQuest são marcas comerciais ou marcas registradas da International Business Machines Corp., registradas em vários países no mundo todo. Outros nomes de produtos e serviços podem ser marcas comerciais registradas da IBM ou de outras empresas. Uma lista atual de marcas comerciais registradas da IBM está disponível na web em Copyright and trademark information em <http://www.ibm.com/legal/copytrade.shtml>. Linux é uma marca registrada de Linus Torvalds nos Estados Unidos e/ou em outros países. Microsoft, Windows, Windows NT e o logotipo Windows são marcas comerciais registradas da Microsoft Corporation nos Estados Unidos e/ou em outros países. Unix é uma marca registrada da The Open Group nos Estados Unidos e em outros países. Java e todas as marcas comerciais registradas e logotipos baseados em Java são marcas comerciais ou marcas registradas da Oracle e/ou de suas afiliadas.

Esse programa inclui: Jacorb 2.3.0, Copyright 1997-2006 O projeto JacORB; e XOM1.0d22, Copyright 2003 Elliotte Rusty Harold, cada um deles disponível sob a Licença Pública Geral da Biblioteca Gnu (LGPL), uma cópia da qual está disponível no arquivos Avisos que acompanha esse programa.

Índice

Capítulo 1. O Utilitário de Construção

Ounce/Make	1
Requisitos	1
O que o Ounce/Make Suporta	2
Operação	2
Executando Ounce/Make	3
Nomeando Arquivos de Projeto	3
Saída do Ounce/Make	5
Sintaxe de comando Ounce/Make e opções make	5
Arquivo de Propriedades Ounce/Make	8
Elementos do arquivo de propriedades Ouncemake	9
Arquivos de Propriedades de Amostra	14
Exemplos	14
Exemplo 1: Ounce/Make Sem Opções	15
Exemplo 2: Ounce/Make com Opção Recursiva	16
Exemplo 3: Ounce/Make com Projeto Único e Opção Recursiva.	17

Capítulo 2. AppScan Source command line interface (CLI). 19

Objetos e Contexto	19
Permissões do AppScan Source command line interface (CLI)	21
Iniciando o AppScan Source command line interface (CLI)	21
Exibindo a AppScan Source command line interface (CLI) em Idiomas Nacionais.	21
Sintaxe de Comando	22
Comandos do AppScan Source command line interface (CLI)	23
Resumo dos Comandos da AppScan Source command line interface (CLI)	23
about (a)	27
clearcache (cc).	28
delete (del).	31
deleteassess (da)	31
deleteuser (du).	31
delvar (dv)	31
details (det)	32
echo.	33
getaseinfo (gase)	33
help (?)	34
import (im)	34
info (i)	35
list (ls, dir)	35
listassess (la).	36
listgroups (lgrp)	36
listusers (lu)	37
log	37
login (in)	38
login_file	40
login_local (local)	40
logout (out).	40
moduser (mu).	41
newuser (nu)	42

openapplication (oa).	44
openassessmentfile (oaf)	46
password (passwd)	46
printuser (pu)	47
publishassess (pa)	47
publishassessase (pase).	48
quit.	50
record (rc)	51
refresh (rf).	51
register (reg)	51
removeassess (da)	52
report (rpt).	52
scan (sc)	54
script (scr).	55
setaseinfo (sase)	56
setcurrentobject (set, cd)	58
setvar (sv)	58
unregister (unreg)	59
Executando Avaliações Automatizadas	59

Capítulo 3. A Ferramenta de Construção Ounce/Ant. 61

Integração de Ounce/Ant e Apache/Ant	61
Propriedades do Ounce/Ant	62
Configurando Propriedades	62
Criando Projetos.	63
ounceCreateProject	63
ounceSourceRoot.	64
ounceWeb	64
ounceExclude.	64
Nomeando Projetos.	65
Criando e Nomeando Aplicativos	65
Integração de Construção.	66

Capítulo 4. API de Acesso a Dados do AppScan Source 67

Modelo de Objeto de API de Acesso a Dados	68
Usando a API de Acesso a Dados	70
Classes e Métodos de API de Acesso a Dados	70
AssessedFile.	71
Assessment	72
AssessmentDiff	73
AssessmentFilter	74
AssessmentResults	75
Call.	78
ClassificationType	80
DateProximityUnit	80
Factory.	81
Finding.	83
Rastreamento.	89
SeverityType.	90
OunceException	90

Capítulo 5. Plug-in Ounce/Maven 91

Instalando Ounce/Maven	91
----------------------------------	----

Usando Ounce/Maven	92
Cenários de Ounce/Maven	92
Criação Arquivos de Aplicativo e de Projeto	92
Aplicativos de Varredura	93
Relatórios	93
Integrando Relatórios com o Destino do Site	93
Objetivos Ounce/Maven	93
ounce:application	94
ounce:project	94
ounce:project-only	94
ounce:scan	95
ounce:report	95

Capítulo 6. AppScan Source for

Automation 97

Especificando Credenciais de Login do Automation Server a partir da Linha de Comandos	97
O Arquivo de Configuração do Automation Server	98
Criação de Log do Automation Server	99
Usando Ounceauto a partir da Linha de Comandos	99
GenerateReport	100
PublishAssessment	102
PublishAssessmentASE	102
ScanApplication	104
Wait	107

Capítulo 7. APIs de Manipulação de Framework for Frameworks 109

Principais Componentes da API Framework for Frameworks	110
Usando APIs Framework for Frameworks	110
Sobre o Exemplo	111
Importando o Projeto de Exemplo para o Eclipse ou Rational Application Developer for WebSphere Software (RAD).	111

Criando uma Classe que Implementa F4FHandler	114
Criando um Arquivo de Manifesto para seu Manipulador	115
Criando um JAR para seu Manipulador	115
Exportando o JAR para wafgens	118
Ações Comuns Executadas pelo Manipulador Framework for Frameworks API classes and methods	118
F4FActions	119
F4FApp	124
F4FHandler	126
TaintedParam	127
Métodos Sintéticos de Alto Nível	128
Formato VDB	128
Usando Métodos Sintéticos de Alto Nível	129
Exemplo: Criação do Método Sintético	132
Integrando um novo manipulador Framework for Frameworks ao Scanner Customizado da Web Service Description Language Existente.	134
Conectando o Manipulador F4F de Serviço da Web ao Scanner Customizado da WSDL	135
Mapeamento de Assinatura.	136

Capítulo 8. Mensagens de Erro do Componente do Cliente AppScan Source 139

Avisos 153

Índice Remissivo 157

Capítulo 1. O Utilitário de Construção Ounce/Make

Ounce/Make é uma ferramenta que automatiza a importação de informações de configuração para o AppScan Source, a partir de ambientes de construção que usam `makefile`. Ounce/Make elimina a necessidade de importar informações de configuração de `makefiles` manualmente.

O utilitário Ounce/Make fornece uma interface de linha de comandos para gerar arquivos de projeto do AppScan Source (`.ppf`) de `makefiles` (um `makefile` fornece um meio de construir um aplicativo ou uma biblioteca executável a partir de arquivos de origem). Os arquivos gerados `ppf` contêm todas as informações necessárias pelo AppScan Source para acessar o código fonte que o `makefile` correspondente é responsável por compilar. Depois que o Ounce/Make gerar os arquivos `.ppf`, você pode importar ou incluir os arquivos de projeto no AppScan Source.

Para ativar o utilitário:

- Em sistemas Windows, execute `<install_dir>\bin\ouncemake.exe` (em que `<install_dir>` é o local da instalação do AppScan Source), por exemplo no Windows (32 bits):
`C:\Program Files\IBM\AppScan Source\bin\ouncemake.exe`
- Nos sistemas Linux, execute `<install_dir>/bin/ouncemake`, por exemplo:
`/opt/ibm/appscansource/bin/ouncemake`

Make é a ferramenta que automatiza a compilação, o vínculo (e assim por diante), de programas, levando em conta as interdependências de módulos e seus tempos de modificação. Make lê as instruções de `makefile`, que especifica um conjunto de destinos, os arquivos das quais elas dependem e os comandos a serem executados para produzi-las.

Requisitos

Para criar os arquivos de projeto do AppScan Source com sucesso, você deve executar Ounce/Make em um ambiente adequado. A lista a seguir detalha em itens os requisitos para Ounce/Make ser executado com êxito. Se você não atender a todos esses requisitos, o Ounce/Make falha.

- O diretório a partir do qual o Ounce/Make é executado contém um `makefile` válido.
- O ambiente de construção deve poder emitir um comando `make` que será bem-sucedido.
- Execute o comando `make clean` antes de executar Ounce/Make. É possível executar explicitamente `make clean` antes de executar Ounce/Make ou incluí-lo com Ounce/Make especificando a opção `- clean`.
- `Makefiles` que o Ounce/Make encontra não podem conter caminhos absolutos codificados permanentemente sob as seguintes circunstâncias:
 - Para o executável `make` ao chamar um outro `makefile`:
Por exemplo, não faça referência ao caminho `/usr/bin/make -f makefile.mk`. No `makefile`, faça referência a `make` através do executável `make` ou de uma variável. A variável pode ser o *make macro*, `$(MAKE)` ou uma outra variável que você especifica no arquivo de Propriedades.

- Para o executável do compilador ao compilar código fonte:
Por exemplo, `/usr/bin/gcc -I.. -DF00 -o myfile.o myfile.cpp`
- Para o executável vinculador ao vincular arquivos de objeto
Por exemplo, `/usr/bin/ld file1.o file2.o`
- Em instruções `#include`.
Para usar uma instrução `#include`, inclua o seguinte sinalizador no arquivo de projeto como uma opção da configuração:
`--remote_root <dir remoto>`
em que `<dir remoto >` define o ponto de montagem do diretório remoto.

Nota: É possível especificar apenas uma única `remote_root`. Todos os caminhos de códigos permanentes para instruções `#include` devem resolver para um único ponto de montagem.
- Não especifique macros para o make, compilador e executável vinculador na linha de comandos ao chamar make (por exemplo, `make CC=gcc LD=ld`).

O que o Ounce/Make Suporta

O Ounce/Make suporta múltiplas plataformas de computador, compiladores e versões de make.

Plataformas

Você pode executar Ounce/Make em todas as plataformas suportadas pelo AppScan Source.

Versões de Make

O Ounce/Make suporta as seguintes versões de make:

- make: O utilitário make do UNIX é uma ferramenta de engenharia de software para gerenciar e manter programas de computador
- gmake: GNU Make é uma ferramenta que controla a geração de executáveis e outros arquivos não de origem de um programa a partir dos arquivos de origem do programa
- nmake: NMAKE.exe (o Utilitário Microsoft Program Maintenance) é uma ferramenta de 32 bits que constrói projetos baseados em comandos dentro de um arquivo de descrição

Compiladores

O Ounce/Make suporta os seguintes compiladores:

- GCC: A Coleta do Compilador GNU. O AppScan Source suporta os front ends C e C++
- cl: `cl.exe` é o compilador para o Microsoft Visual C++

Operação

Ounce/Make opera com Make para extrair as informações de configuração necessárias dos makefiles e criar arquivos de projeto (ppf) adequados do AppScan Source. Para realizar isso, o Ounce/Make requer que você forneça determinadas informações sobre o ambiente de construção, como a versão de make usada.

Use o arquivo de Propriedades para descrever o ambiente de construção (consulte “Arquivo de Propriedades Ounce/Make” na página 8).

Executando Ounce/Make

É possível executar Ounce/Make a partir de qualquer diretório em que um comando make seja executado com êxito. Na maioria dos casos, o executável `ouncemake` não está no diretório em que você o chama. Portanto, é recomendável incluir o diretório que contém `ouncemake` na variável de ambiente `PATH` para chamar `ouncemake` facilmente.

Por exemplo, se o seu código fonte e/ou `makefile` residir no diretório `\development\srcdir`, para executar `ouncemake` você deve primeiro mudar para esse diretório:

```
%cd \development\srcdir
```

e, em seguida, executado `ouncemake`:

```
%ouncemake <opções>
```

Nota: Para compatibilidade com versões anteriores, o comando `pmake` ainda é aceito. No entanto, `pmake` pode ser removido em um release futuro.

Nomeando Arquivos de Projeto

Ounce/Make usa convenções descritas neste tópico para nomear arquivos de projeto do AppScan Source.

- Ao criar arquivos de projeto do AppScan Source, Ounce/Make usa o caminho de diretório relativo a partir do diretório no qual você chamou `ouncemake` para o diretório onde `ouncemake` cria o arquivo de projeto.
- O diretório no qual você chama `ouncemake` torna-se o primeiro componente no nome do caminho.
- Sublinhados substituem todos os separadores de caminho como barra invertida (`\`) no Windows e barra (`/`) no Linux.
- Quando o comprimento do nome do arquivo excede as limitações do sistema, o Ounce/Make divide componentes do caminho, iniciando à esquerda, até o comprimento do nome do arquivo ser compatível com a convenção de nomenclatura do sistema.
- Ao executar o `ouncemake` na raiz de um sistema de arquivos, como `/` ou `c:\`, o `ouncemake` criará um arquivo de projeto do AppScan Source nomeado `root.ppf`.

O AppScan Source salva o `.ppf` criado no local próximo ao `makefile` que o `ppf` representa. Por exemplo, se você executar Ounce/Make, criando um único arquivo de projeto, o AppScan Source salvará o `ppf` no diretório a partir do qual você chamou Ounce/Make. Consulte “Exemplo 2: Ounce/Make com Opção Recursiva” na página 16 para ver os arquivos `ppf` criados no modo de multiprojetos.

Nota: Se houver mais de um `makefile` em um diretório, o Ounce/Make criará apenas um arquivo `.ppf` no diretório.

Exemplo 1

Esse exemplo ilustra um arquivo `ppf` criado com os separadores de caminho substituídos por sublinhados.

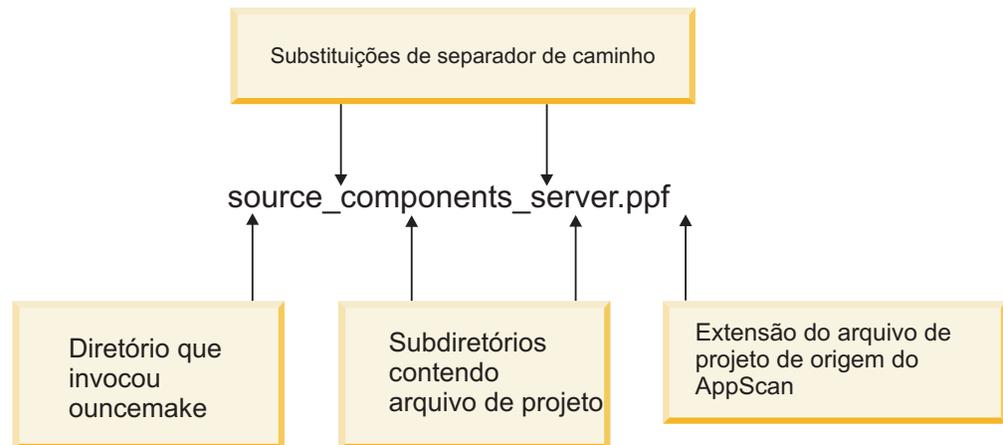
Chame `ouncemake` a partir do seguinte diretório:

C:\development\source

Durante a execução, o Ounce/Make cria um arquivo de projeto do AppScan Source em:

C:\development\source\components\server

O nome do ppf é source_components_server.ppf:



Exemplo 2

Caminho de limite e nomes de arquivo do Microsoft Windows e do Linux. Esses sistemas operacionais limitam o número de caracteres a 255. O Exemplo 2 mostra o caso de um nome do arquivo que excede limitações de comprimento do caminho.

O usuário chama Ounce/Make a partir do seguinte diretório:

C:\path1\path2\path3\path4\path5\development\source

Durante a execução, o Ounce/Make cria um arquivo de projeto do AppScan Source no diretório a seguir:

C:\path1\path2\path3\path4\path5\development\source\components\server

Se o nome do arquivo pode ter um máximo de 25 caracteres, devido a limitações de caminho, o nome do arquivo resultante é:

components_server.ppf

Nomeando Explicitamente um Projeto

É possível usar opcionalmente a variável de ambiente OUNCE_PROJ_NAME para especificar explicitamente um nome para um projeto recém criado. Defina a variável de ambiente OUNCE_PROJ_NAME=value como com qualquer outra variável de ambiente.

Se não configurado, o nome é gerado pela convenção de nomenclatura de projeto existente.

Por exemplo, para especificar o nome do projeto como MyMakeProject a partir da linha de comandos:

```
$(MAKE) $(MAKE_ARGS) all OUNCE_PROJ_NAME=MyMakeProject
```

Saída do Ounce/Make

Enquanto em execução, o Ounce/Make gera saída informando sobre determinados eventos.

Os seguintes eventos acionam mensagens de saída:

- Criação de projeto
- Chamando Make
- Erro

Criação de projeto

Quando o Ounce/Make criar um arquivo de projeto do AppScan Source, ele exibirá uma mensagem que inclui o nome ppf. O seguinte exemplo é uma mensagem de saída da criação de projeto do Ounce/Make:

```
Criado projeto do AppScan Source <nome_do_projeto>.ppf no diretório <diretório>.
```

Chamando Make

Quando o Ounce/Make chama o executável make, o nome do executável e as opções são impressos. A seguinte amostra mostra a saída gerada quando o Ounce/Make chama o make.

```
/usr/bin/gmake -f makefile.mk release
```

Erro

Se ocorrer um erro durante a execução, o Ounce/Make imprime uma mensagem de erro que descreve o erro. Se o erro ocorreu quando o executável make foi executado, a mensagem de erro exibida é o erro do executável make. Se o erro foi específico do Ounce/Make, uma mensagem de erro apropriada descrevendo a condição de erro do Ounce/Make é exibida.

Sintaxe de comando Ounce/Make e opções make

O Ounce/Make suporta múltiplas opções que podem alterar a maneira em que ele se comporta quando é executado.

É possível configurar essas opções no arquivo de Propriedades (consulte “Arquivo de Propriedades Ounce/Make” na página 8 para obter detalhes adicionais) ou inclua-as na linha de comandos. Se você configurar as opções no arquivo de Propriedades, não é necessário especificá-las na linha de comandos sempre que executar Ounce/Make.

Sinopse

O Ounce/Make suporta a seguinte sintaxe:

```
ouncemake [options] [-- make_options]
```

Opções

Um hífen (-) deve preceder todas as opções. É necessário especificar as opções separadamente; você não pode concatená-las após um único hífen. Por exemplo, o comando:

```
ouncemake -sr
```

não é uma sintaxe suportada, mas você pode executar:

```
ouncemake -s -r
```

Nota: Cada opção deve ser separada por um espaço.

Ao executar Ounce/Make, você pode usar a opção abreviada ou a palavra completa.

A seguinte tabela inclui colunas descrevendo cada opção.

- **Opção:** Identifica a opção que Ounce/Make entenderá quando chamado.
- **Padrão:** Se aplicável, explica como Ounce/Make opera por padrão se você não especificar a opção.
- **Descrição:** O comportamento do Ounce/Make ao usar essa opção.

Opção	Padrão quando a opção não estiver especificada	Descrição
-a <nome_do_aplicativo> -application <nome_do_aplicativo>	Desativado	Quando especificado, Ounce/Make cria um arquivo de aplicativo denominado <nome_do_aplicativo>.paf que contém todos os projetos criados por Ounce/Make. O arquivo é criado no diretório no qual ouncemake foi executado.
-b -build	Desativado	Execute o build ao reunir as opções make. Essa opção é incompatível com Cygwin.
-r -recursive	Não-recursivo	O Ounce/Make segue recursivamente todas as chamadas para outros makefiles. Por exemplo, se um makefile existir na raiz de uma árvore de códigos de origem para chamar todos os makefiles nos subdiretórios, chamar ouncemake -r no diretório que contém o makefile raiz faz o Ounce/Make seguir as chamadas para os makefiles de subdiretórios.
-nr -non_recursive	Não-recursivo	O Ounce/Make não segue recursivamente chamadas para outros makefiles.

Opção	Padrão quando a opção não estiver especificada	Descrição
-s -single_project	Modo de múltiplos projetos	Modo de projeto único. Quando no modo de projeto único, o Ounce/Make gera apenas um arquivo de projeto único no diretório a partir do qual ele é chamado. Se não especificado, o Ounce/Make está no modo de múltiplos projetos.
-ns -non_single_project -m -multiple_project	Modo de múltiplos projetos	Modo de múltiplos projetos. Neste modo, o Ounce/Make gera um arquivo de projeto do AppScan Source em cada diretório para cada makefile que encontrar.
-nv -non_verbos -q -quiet	Modo não detalhado	Modo não detalhado. O Ounce/Make só emite suas próprias mensagens de saída. O Ounce/Make suprime a saída a partir do make.
-v -verbos	Modo Detalhado	Modo detalhado. O Ounce/Make emite a saída do make, bem como sua própria, como saída padrão.
-l log_level	1 (desligado)	1 a 10. 10 fornece a criação de log máxima.
-c <comando_clean> -clean <comando_clean>	Desativado	Quando especificado, Ounce/Make interpreta <comando_clean> como um comando e o executa. <comando_clean> deve ser o comando que o usuário executaria normalmente para limpar. Por exemplo, make clean é um comando comum para executar um clean. Observe que você deve colocar o comando entre aspas. Como o Ounce/Make requer um clean antes de executar, se você não especificar essa opção, um prompt aparece perguntando se deseja continuar.
-nc -no_clean	Desativado	Informe o Ounce/Make para não executar clean e não exibir um prompt lembrando de que um clean não irá ser executado.

Opção	Padrão quando a opção não estiver especificada	Descrição
-p <arquivo_de_propriedades>	n/d	Permite aos usuários especificar um arquivo de propriedades para o Ounce/Make usar. <arquivo_de_propriedades> deve ser um caminho absoluto para o arquivo de propriedades que Ounce/Make deve usar.
-? -h -help	n/d	Ajuda para opções do Ounce/Make.

Arquivo de Propriedades Ounce/Make

O arquivo de propriedades Ounce/Make (ouncemake_properties.xml) é um arquivo XML que contém detalhes sobre seu ambiente de construção. Quando o Ounce/Make é executado, ele localiza o arquivo de propriedades com base no caminho de procura ou na opção -p. O arquivo de propriedades deve especificar o compilador usado em seu ambiente de compilação, o vinculador e elementos de construção.

Para saber sobre os elementos requeridos e opcionais especificados no arquivo de propriedades, consulte o “Elementos do arquivo de propriedades Ouncemake” na página 9.

Ao executar o Ounce/Make, se você não especificar a opção -p na linha de comandos, o Ounce/Make examinará seu caminho de procura buscando o arquivo ouncemake_properties.xml (consulte “Sintaxe de comando Ounce/Make e opções make” na página 5 para saber sobre a especificação de opções na linha de comandos). O caminho de procura Ounce/Make depende da plataforma.

O caminho da procura do Microsoft Windows é:

- Diretório de trabalho atual
- <data_dir>\config (em que <data_dir> é o local dos dados do programa do AppScan Source, conforme descrito em Capítulo 9, “Locais de Arquivos de Instalação e de Dados do Usuário”, na página 149)

O caminho da procura do Linux é:

- Diretório de trabalho atual
- <data_dir>/config (em que <data_dir> é o local dos dados do programa do AppScan Source, conforme descrito em Capítulo 9, “Locais de Arquivos de Instalação e de Dados do Usuário”, na página 149)
- Diretório inicial do usuário
- /etc

Nota:

- No Linux, se você não especificar a opção -p e o arquivo de propriedades não existir no caminho de procura, AppScan Source criará automaticamente um no diretório de trabalho atual.

- Para compatibilidade com versões anteriores, o Ounce/Make reconhece `pmake_properties.xml`. Se ambos `pmake_properties.xml` e `ouncemake_properties.xml` existirem no mesmo diretório, `ouncemake_properties.xml` tem precedência.

Elementos do arquivo de propriedades Ouncemake

OuncemakeProperties, o elemento raiz do arquivo de Propriedades, contém os elementos listados neste tópico.

Elementos requeridos:

- Compiler
- Linker
- Make

Elementos opcionais:

- MakeOptions
- Options
- GlobalProjectOptions
- FileOptions
- Executável
- MountRoot

Compiler

O elemento `Compiler` especifica o executável compiler usado em seu ambiente de construção. O valor deste elemento deve ser um caminho absoluto para um executável. Esse elemento consiste em um atributo opcional, `macro`, que especifica o nome da variável que armazena o executável compiler ao construir e um valor específico para o caminho para o compilador. Se você não especificar o atributo `macro`, o Ounce/Make configura o atributo `macro` como `CC` por padrão. O Arquivo de Propriedades pode conter múltiplos elementos `Compiler`, mas pelo menos um é necessário.

O valor para o atributo `macro` deve ser exclusivo em todos os elementos `Compiler`. Por exemplo, não é possível listar `CC` como um atributo `macro` mais de uma vez.

Exemplo

```
<Compiler macro=CXX>/usr/bin/gcc</Compiler>
```

Descrição

`CXX`: Seus makefiles fazem referência a esse compilador com o macro `CXX`.

`/usr/bin/gcc`: Informa Ounce/Make que você usa o compilador `/usr/bin/gcc`.

Linker

O elemento `Linker` especifica um executável vinculador usado no ambiente de construção. O valor deste elemento deve ser um caminho absoluto para um executável. O arquivo de Propriedades pode conter múltiplos elementos `Linker`, mas pelo menos um é necessário.

Esse elemento consiste em um atributo, `macro`, que especifica o nome da variável que armazena o executável linker ao construir. Se você não especificar o atributo `macro`, `LD` é o linker macro padrão.

O valor para o atributo macro deve ser exclusivo em todos os elementos Linker. Por exemplo, não é possível listar LD como um atributo Linker mais de uma vez.

Exemplo

```
<Linker macro=LD>/usr/bin/ld</Linker>
```

Descrição

LD: Informa Ounce/Make que você usa o macro LD

/usr/bin/ld: Informa Ounce/Make que você usa o vinculador /usr/bin/ld

Make

O elemento Make especifica um executável make usado em seu ambiente de construção. O valor deste elemento deve ser um caminho absoluto para um executável. O arquivo de Propriedades pode conter múltiplos elementos Make, mas pelo menos um é necessário.

Esse elemento consiste em um atributo, macro, que especifica o nome da variável que armazena o executável make ao construir. Se você não especificar o atributo macro, Ounce/Make configura o atributo como MAKE por padrão.

O valor para o atributo macro make deve ser exclusivo em todos os elementos Make.

Exemplo

```
<Make macro=MAKE>/usr/bin/make</Make>
```

Descrição

- MAKE: Informa Ounce/Make que você usa o MAKE macro.
- /usr/bin/make: Informa Ounce/Make que você usa o make /usr/bin/make

MakeOptions

O elemento MakeOptions especifica as opções para passar ao executável make. Esse elemento é opcional e não pode aparecer mais de uma vez no Arquivo de Propriedades.

Exemplo

```
<MakeOptions>-f makefile.mk release</MakeOptions>
```

Descrição

As opções:

```
-f makefile.mk release
```

passar para o executável make.

Options

O elemento Options especifica opções to pass para passar para o Ounce/Make na chamada.

Esse elemento fornece uma alternativa para o uso de determinadas opções de linha de comandos do Ounce/Make, conforme descrito em “Sintaxe de comando Ounce/Make e opções make” na página 5.

Atributos usam a sintaxe:

<opção> = <true | false>

O elemento `Options` e seus atributos não são necessários.

O elemento `Options` pode incluir os seguintes atributos:

Atributo	Descrição
<code>recursive</code>	Valor booleano. True ou false. Quando verdade, implica na opção da linha de comandos <code>-r</code> .
<code>single_project</code>	Valor booleano. True ou false. Quando verdade, implica na opção da linha de comandos <code>-s</code> .
<code>verbose</code>	Valor booleano. True ou false. Quando verdade, implica na opção da linha de comandos <code>-v</code> .
<code>clean</code>	Valor de cadeia, colocado entre aspas ("), como <code>"make clean"</code> . Quando verdade, implica na opção da linha de comandos <code>-c</code> . O valor deve ser o comando para executar o clean. Por exemplo, <code>gmake clean</code> .
<code>build</code>	Valor booleano. True ou false. Execute o build ao reunir as opções make. Nota: Incompatível com Cygwin.
<code>application</code>	Valor de cadeia. Quando configurado, implica na opção <code>-a</code> . O valor especificado deve ser o nome do aplicativo que você deseja.
<code>no_clean</code>	Valor booleano. True ou false. Informe o Ounce/Make para não executar clean e não exibir um prompt lembrando de que um clean não irá ser executado.

Nota: O Ounce/Make usa as opções configuradas no arquivo de Propriedades. No entanto, se você executar o Ounce/Make com opções incluídas na linha de comandos, essas opções têm precedência sobre o conjunto de opções no arquivo de Propriedades. Se você executar o Ounce/Make sem opções na linha de comandos, o Ounce/Make aplica as opções do arquivo de Propriedades.

Exemplo

A seguir está uma linha de exemplo a partir do arquivo de Propriedades que faz uso de todos os atributos:

```
<Options recursive="true" single_project="false" verbose="false"
clean="nmake.exe clean" no_clean="false"></Options>
```

Descrição

- `recursive="true"`

- `single_project="false"` direciona o Ounce/Make para operar no modo de múltiplos projetos. Ao operar no modo de múltiplos projetos, o atributo `recursive` também deve ser configurado como `true`.
- `verbose="false"` desliga a verbosidade.
- `clean="nmake.exe clean"` limpa o ambiente de construção.
- `no_clean="false"` informa o Ounce/Make para executar um comando `clean`, suprimindo a mensagem que declara que o `clean` será executado.

GlobalProjectOptions

O elemento `GlobalProjectOptions` especifica as opções de nível do Projeto para todos os arquivos contidos dentro do projeto. O elemento `GlobalProjectOptions` e seus atributos não são necessários.

A seguinte lista descreve os atributos para o elemento `GlobalProjectOptions`:

- `include_paths`: Valor de cadeia. Lista separada por ponto e vírgula de caminhos include a ser aplicados em todos os arquivos no projeto.
- `macros`: Valor de cadeia. Lista separada por ponto e vírgula de macros a ser aplicados em todos os arquivos no projeto.
- `compiler_options`: Valor de cadeia. Lista de opções do compilador, separadas por um espaço, a ser aplicadas em todos os arquivos no projeto. Não especifique caminhos `include` e `macros` aqui.

Exemplo

A seguir está uma linha de exemplo a partir do arquivo de Propriedades que faz uso de todos os atributos:

```
<GlobalProjectOptions include_paths="/usr/include;/usr/local/include"
macros="DEBUG;WIN32" compiler_options="-f non-const-strings"/>
```

Descrição

- `include_paths="/usr/include;/usr/local/include"` inclui o caminho na seção `Includes` da configuração do projeto.
- `macros="DEBUG;WIN32"` inclui macros `DEBUG` e `WIN32` na seção `Macros` da configuração do projeto.
- `compiler_options=` inclui as opções do compilador na seção `Linha de Opções/Comandos` da configuração do projeto.

FileOptions

O elemento `FileOptions` permite incluir caminhos, macros e outras opções do compilador para ser especificados para arquivos com uma determinada extensão. É possível usar `FileOptions` múltiplas vezes para especificar diferentes opções para arquivos com diferentes extensões. Por exemplo, conforme mostrado abaixo, se você tiver um projeto que contenha ambos os arquivos C e C++, crie dois elementos `FileOptions`, um para cada tipo de arquivo.

A seguinte lista descreve os atributos para o elemento `FileOptions`:

- `extensions`: Valor de cadeia. Lista separadas por ponto e vírgula de extensões do arquivo. Cada arquivo com uma extensão que corresponda a uma extensão nessa lista adquire as opções especificadas por essa propriedade. Se uma extensão do arquivo aplica-se a mais de uma ocorrência da propriedade `FileOptions`, a primeira ocorrência no arquivo de propriedades do Ounce Make tem precedência.

- `compiler_options`: Valor de cadeia. Lista de opções do compilador, separadas por um espaço, a ser aplicadas em todos os arquivos com a extensão especificada. Não especifique caminhos `include` e `macros` aqui.
- `include_paths`: Valor de cadeia. Lista separada por ponto e vírgula de caminhos `include` a ser aplicados em todos os arquivos com a extensão especificada.
- `macros`: Valor de cadeia. Lista separada por ponto e vírgula de `macros` a ser aplicados em todos os arquivos com a extensão especificada.

Exemplos

Os seguintes exemplos de `FileOptions` mostram como configurar o arquivo de propriedades do Ounce Make para aplicar as opções corretas em ambos os arquivos C e C++.

O elemento `FileOptions` com `extensions="c"` aplica seus outros valores de atributo apenas em arquivos com uma extensão `c` (`<filename.c>`). O elemento `FileOptions` com `extensions="cpp;cxx"` aplicará seus outros valores de atributo apenas em arquivos com extensões `cpp` (`<filename.cpp>`) ou `cxx` (`<filename.cxx>`).

```
<!-- opções g++ para arquivos C -->
<FileOptions
  extensions="c"
  compiler_options="-gcc_linux_i386"
  include_paths="/usr/local/include;
  /usr/lib/gcc-lib/i386-redhat-linux/3.2.3/include;
  /usr/include"
  macros="" />

<!-- opções g++ para arquivos C++ -->
<FileOptions
  extensions="cpp;cxx"
  compiler_options="-g++_linux_i386"
  include_paths="/usr/include/c++/3.2.3;
  /usr/include/c++/3.2.3/i386-redhat-linux;
  /usr/include/c++/3.2.3/backward;/usr/local/include;
  /usr/lib/gcc-lib/i386-redhat-linux/3.2.3/include;
  /usr/include"
  macros="__GNUG__=3" />
```

Descrição

`extensions="c"` e `extensions="cpp;cxx"`

especifica as extensões do arquivo nas quais essas opções de arquivo se aplicam.

Executável

O elemento `Executável` permite o uso de qualquer executável (diferente de compilador, linker ou make) no ambiente de construção. O valor deste elemento deve ser um caminho absoluto para um executável. O elemento `Executable` contém um atributo, `macro`, que especifica o nome da variável usada para armazenar o nome do executável ao construir. O valor para o atributo `macro` deve ser exclusivo em todos os elementos `Executable`.

O arquivo de Propriedades pode conter qualquer número de elementos `Executable`. O elemento `Executable` é opcional e necessário somente se você usar outros executáveis em seu ambiente de construção que possam fazer o Ounce/Make falhar. Identificando esses executáveis, o Ounce/Make pode impedir a falha.

Exemplo

A seguir está uma linha de exemplo a partir do arquivo de Propriedades:

```
<Executable macro=ARCHIVE>/usr/bin/ar<Executable>
```

Descrição

Esse exemplo identifica o Ounce/Make em que a construção usa o executável /usr/bin/ar e que makefiles fazem referência a esse executável com o macro ARCHIVE.

MountRoot

O elemento MountRoot é necessário se você estiver avaliando o código fonte a partir de um sistema remoto. MountRoot informa Ounce/Make para preceder o ponto de montagem especificado a todos os caminhos include absolutos que ele encontrar.

Especificando um MountRoot, o Ounce/Make implicitamente inclui a opção --remote_root na lista de opções de compiladores no ppf resultante.

MountRoot é válido apenas no Linux.

Exemplo

A seguir está uma linha de exemplo a partir do arquivo de Propriedades:

```
<MountRoot>/mnt/mount_point/usr/include</MountRoot>
```

Descrição

Se um dos caminhos include especificados for /usr/include (como gcc -I/usr/include ...), o Ounce/Make precede /usr/include com o ponto de montagem de forma que o caminho armazenado no arquivo ppf resultante contenha o caminho correto, como /mnt/mount_point/usr/include.

Arquivos de Propriedades de Amostra

A instalação do AppScan Source inclui estes arquivos de propriedades de amostra no diretório <data_dir>\config (em que <data_dir> é o local dos dados do programa do AppScan Source, conforme descrito em Capítulo 9, “Locais de Arquivos de Instalação e de Dados do Usuário”, na página 149):

- SampleOuncemakeProperties-Windows.xml
- SampleOuncemakeProperties-Linux.xml

É possível modificar esses arquivos para criar um arquivo de propriedades customizadas. Se você não estiver familiarizado com XML, use essas amostras como modelos para criar seu arquivo.

Exemplos

Esta seção descreve três maneiras de usar o Ounce/Make.

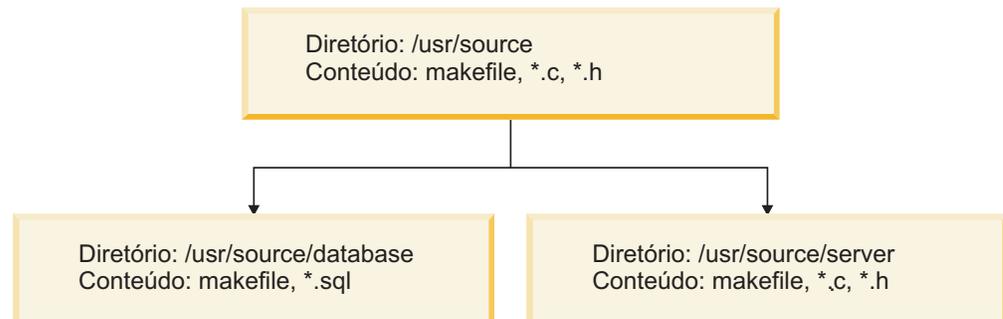
O “Exemplo 1: Ounce/Make Sem Opções” na página 15 ilustra Ounce/Make sem opções, criando um arquivo de projeto do AppScan Source baseando-se apenas no makefile no diretório em que você chama Ounce/Make.

“Exemplo 2: Ounce/Make com Opção Recursiva” na página 16 usa o Ounce/Make com a opção `-r` (recursiva), instruindo o Ounce/Make para operar recursivamente e seguir todas as chamadas para outros makefiles.

No “Exemplo 3: Ounce/Make com Projeto Único e Opção Recursiva” na página 17, Ounce/Make usa as opções `-r` (recursivo) e `-s` (único para criar um único arquivo de projeto do AppScan Source baseando-se no processamento recursivo de todos os makefiles que o Ounce/Make encontra.

Estrutura e Arquivos de Diretório

Todos os três exemplos usam a mesma estrutura e arquivos de diretório:



Esse diagrama mostra um diretório-raiz (`/usr/source`) que contém um makefile e arquivos de origem. O diretório `/usr/source` contém dois subdiretórios: `/usr/source/database` e `/usr/source/server`. O diretório `/usr/source/database` contém um makefile e arquivos SQL. O diretório `/usr/source/server` contém um makefile e arquivos de origem.

Esse exemplo faz as seguintes premissas sobre os três makefiles:

- O makefile em `/usr/source` constrói os arquivos de origem em `/usr/source` e chama os makefiles em `/usr/source/database` e `/usr/source/server`.
- O makefile em `/usr/source/database` importa os arquivos SQL para um banco de dados.
- O makefile em `/usr/source/server` constrói os arquivos de origem em `/usr/source/server`.

Exemplo 1: Ounce/Make Sem Opções

Esse exemplo ilustra o uso de Ounce/Make não recursivamente. No modo não recursivo, o Ounce/Make apenas examina o makefile no diretório a partir do qual ele é chamado. Se o makefile original contiver chamadas para outros makefiles, o Ounce/Make as ignora.

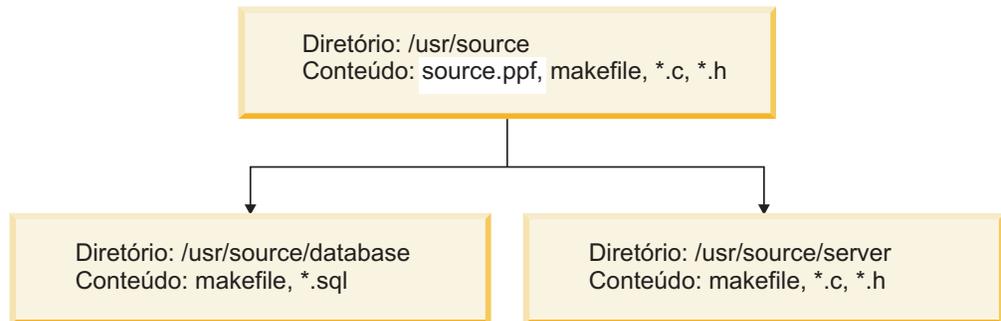
Esse exemplo executa Ounce/Make a partir de `/usr/source`.

Consulte “Estrutura e Arquivos de Diretório” para obter uma explicação gráfica da estrutura de diretório e arquivos que são assumidos por este exemplo.

Comando

`ouncemake`

O seguinte diagrama mostra o conteúdo dos diretórios após o Ounce/Make ser executado:



Após a execução do Ounce/Make, agora o /usr/source contém um arquivo de projeto do AppScan Source nomeado source.ppf. Esse arquivo de projeto contém todas as informações necessárias para avaliar arquivos de origem em /usr/source. Ounce/Make no modo não recursivo ignora as chamadas no makefile em /usr/source chamando os makefiles em /usr/source/database e /usr/source/server.

Exemplo 2: Ounce/Make com Opção Recursiva

Com a opção `-r`, o Ounce/Make opera recursivamente (segue chamadas para outros makefiles contidos dentro de um makefile). Desde que o modo de arquivo de diversos projetos seja o modo padrão, quando usado com a opção `-r`, o Ounce/Make cria um arquivo de projeto do AppScan Source para cada makefile encontrado que compila o código fonte.

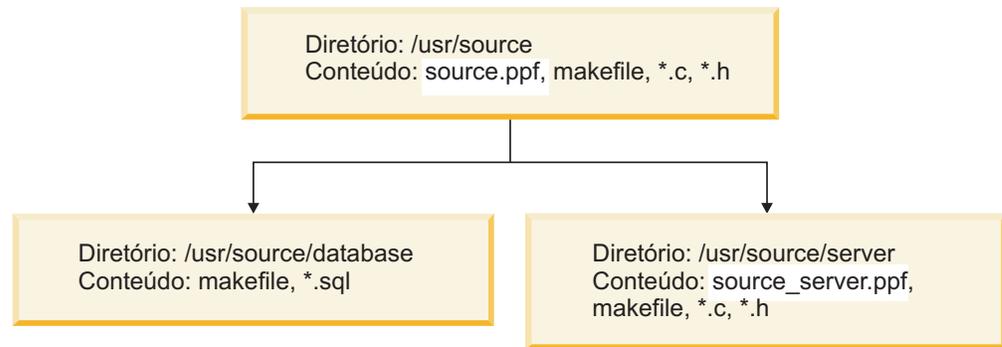
Consulte “Estrutura e Arquivos de Diretório” na página 15 para obter uma explicação gráfica da estrutura de diretório e arquivos que são assumidos por este exemplo.

Comando

```
ouncemake -r
```

A opção `-r` (recursiva) instrui o Ounce/Make a seguir chamadas do makefile para outros makefiles. Para obter uma descrição mais detalhada da opção recursiva, consulte a tabela em “Sintaxe de comando Ounce/Make e opções make” na página 5.

O seguinte diagrama mostra o conteúdo dos diretórios após o Ounce/Make ser executado:



Neste exemplo, o Ounce/Make cria um arquivo de projeto do AppScan Source no `/usr/source` e `/usr/source/server`. Como o makefile em `/usr/source` chamou os makefiles em `/usr/source/database` e `/usr/source/server`, o Ounce/Make verificou se esses makefiles compilaram código fonte.

No caso do makefile no `/usr/source/database`, o Ounce/Make determinou que esse makefile não compila o código fonte; portanto, ele não criou um arquivo de projeto do AppScan Source. No entanto, o Ounce/Make determinou que o makefile no `/usr/source/server` compilou os arquivos de origem naquele diretório e, desse modo, gerou um arquivo de projeto do AppScan Source para os arquivos de origem no `/usr/source/server`.

Exemplo 3: Ounce/Make com Projeto Único e Opção Recursiva

O Exemplo 3 ilustra o uso do Ounce/Make recursivamente no modo de projeto único. Ounce/Make gera um único arquivo de projeto do AppScan Source para uma combinação do código fonte compilado por todos os makefiles encontrados.

Consulte “Estrutura e Arquivos de Diretório” na página 15 para obter uma explicação gráfica da estrutura de diretório e arquivos que são assumidos por este exemplo.

Execute o seguinte comando a partir do diretório `/usr/source`:

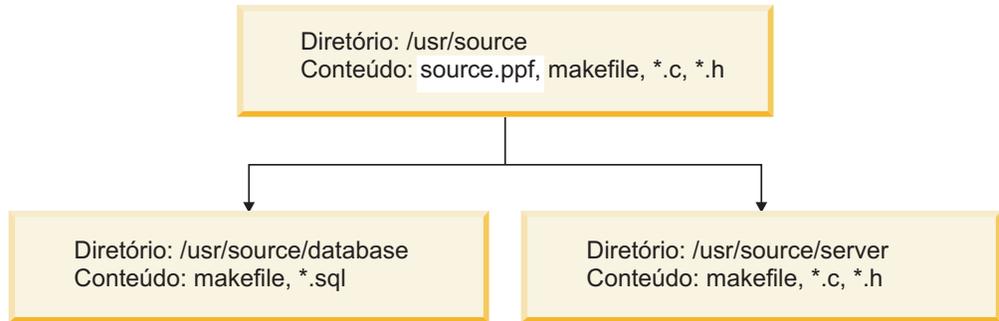
Comando

```
ouncemake -r -s
```

A opção `-r` (recursiva) instrui o Ounce/Make a seguir chamadas do makefile para outros makefiles. Para obter uma descrição mais detalhada da opção recursiva, consulte a tabela em “Sintaxe de comando Ounce/Make e opções make” na página 5.

A opção `-s` instrui Ounce/Make para gerar apenas um arquivo de projeto do AppScan Source no diretório a partir do qual ele é chamado, o que é oposto a criar um novo projeto para cada makefile encontrado.

O seguinte diagrama mostra o conteúdo dos diretórios após o Ounce/Make ser executado.



Um único arquivo de projeto do AppScan Source existe em `/usr/source`. Esse arquivo de projeto do AppScan Source contém as informações de configuração para o código fonte inteiro em `/usr/source` e em `/usr/source/server`.

Capítulo 2. AppScan Source command line interface (CLI)

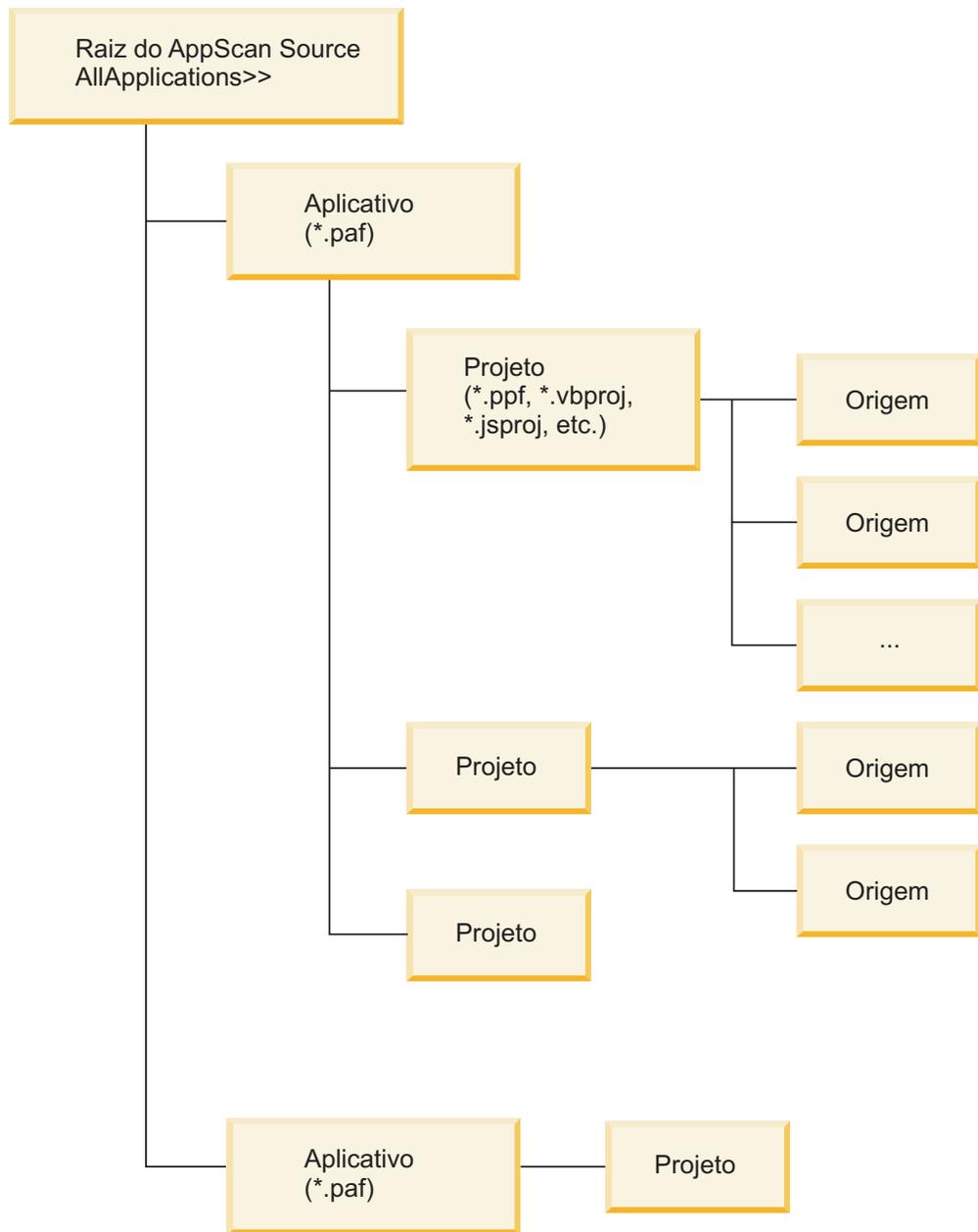
A CLI é uma interface com a funcionalidade principal do AppScan Source.

O CLI suporta a execução das funções do AppScan Source a partir da linha de comandos ou a execução de arquivos de script para registrar comandos enviados durante uma sessão interativa. Em execução, a CLI fornece o feedback do usuário ao console ou, como opção, a um arquivo de log.

Objetos e Contexto

A Árvore de Objetos do AppScan Source

A árvore de objetos mostra a estrutura básica dos objetos do AppScan Source. É possível varrer qualquer objeto (aplicativos, projetos ou arquivos de origem) dentro da estrutura em árvore.



Objetos Ativos e Contexto Atual

Muitos comandos do AppScan Source command line interface (CLI) agem apenas dentro do contexto atual; ou seja, dentro do diretório atual ou em um arquivo especificado. Use o comando `SetCurrentObject (SET, CD)` para mover-se para o objeto antes de chamar qualquer comando que requer um contexto atual.

Adicionalmente, alguns comandos do CLI agem apenas em um objeto ativo, que pode ser um arquivo de origem, projeto ou aplicativo com uma avaliação atualmente selecionado.

Permissões do AppScan Source command line interface (CLI)

Muitos comandos da CLI requerem que tenha as permissões apropriadas do AppScan Source.

Se você não tiver permissão para uma ação, uma mensagem de erro aparecerá. Observe que os comandos Scan (SC), Publish e Report (RPT) da CLI requerem uma licença do AppScan Source for Automation.

Para obter informações adicionais, consulte, consulte seu administrador.

Iniciando o AppScan Source command line interface (CLI)

- Nos sistemas Windows, execute `<install_dir>\bin\AppScanSrcCli.exe` (em que `<install_dir>` é o local da instalação do AppScan Source). Por exemplo, no Windows (32 bits):
`C:\Program Files\IBM\AppScanSource\bin\AppScanSrcCli.exe`
- Em sistemas Linux, execute `<install_dir>/bin/appscansrccli`, por exemplo:
`/opt/ibm/appscansource/bin/appscansrccli`

O CLI permite que um comando válido seja passado na linha de comandos na inicialização do aplicativo.

Exemplo:

O CLI passa o comando `script` para executar uma sessão inteira do CLI automaticamente.

```
<dir_instal>\bin\AppScanSrcCli script myscript.txt
```

Exibindo a AppScan Source command line interface (CLI) em Idiomas Nacionais

Este tópico descreve as etapas necessárias para exibir a CLI em idiomas nacionais.

Procedimento

- Abra `<data_dir>\config\cli.cp` (em que `<data_dir>` é o local dos dados do programa do AppScan Source, conforme descrito em Capítulo 9, “Locais de Arquivos de Instalação e de Dados do Usuário”, na página 149) em um editor de texto. Inclua esta linha no arquivo:

```
-Duser.language=<language>
```

Em que `<language>` é o idioma que você deseja exibir. Estes valores são suportados:

- de para alemão
- es para espanhol
- fr para francês
- it para italiano
- ja para japonês
- ko para coreano
- ru para russo
- pt_BR ou pt-BR para português do Brasil
- zh_CN ou zh-CN para chinês simplificado
- zh_TW ou zh-TW para chinês tradicional

Salve o arquivo após incluir essa linha.

- Abra `<data_dir>\config\ounce.ozsettings` em um editor de texto. Localize a configuração local no arquivo. Esta configuração será semelhante a:

```
<Setting
  name="locale"
  value=""
  default_value=""
  description="The language (and optionally the Country/Region)
    in which UI and messages should be displayed. E.g. fr for
    French, or pt-BR for Portuguese (Brazil)"
  display_name="Locale"
  type="text"
  available_values=""
  read_only="false"
  hidden="false"
  force_upgrade="false"
/>
```

Nessa configuração, modifique o atributo `value` para que ele seja configurado com a mesma configuração de idioma especificada na etapa acima. Por exemplo, se na etapa acima você incluiu `-Duser.language=fr`, altere o atributo `value` para `value="fr"`.

Salve o arquivo após modificar essa configuração.

- Inicie a CLI (se essas mudanças foram feitas enquanto a CLI já estava em execução, será necessário reiniciar a CLI).

Sintaxe de Comando

Os comandos da AppScan Source command line interface (CLI) estão de acordo com o modelo de uso com os argumentos necessários e opcionais, semelhante a um shell de comando. Os comandos da CLI não fazem distinção entre maiúsculas e minúsculas e não requerem comutadores para argumentos diferentes.

A sintaxe para todos os comandos da CLI é:

```
command required_args [optional_args]
```

As descrições de comando e o uso respeitam as seguintes convenções:

- **command**: Comando real, identificado em uma fonte **bold Courier**.
- Argumentos necessários: Identificados na fonte *courier*.
- Literais: Identificados por *italics*.
- Argumentos opcionais: Colocados entre [].
- Argumentos necessários que possuem opções: Colocados entre chaves {}.
- Alguns argumentos requerem valores associados (identificados por sinais de maior e menor < >). Se esses argumentos forem usados sem incluir os valores, você receberá uma mensagem de erro.
- Cada comando possui um nome longo e uma abreviação. Localize as abreviações entre parênteses no título do comando.
- Todos os nomes de comando são uma palavra sem espaços ou guias.
- Argumentos não requerem comutadores de comando, mas são **dependentes de ordem**.

Importante: É importante lembrar que argumentos são dependentes de ordem.

Argumentos podem conter espaços e guias. Coloque argumentos que contêm espaços ou guias entre aspas duplas (" "); caso contrário, eles são analisados como múltiplos argumentos.

Comandos do AppScan Source command line interface (CLI)

Esta seção descreve todos os comandos disponíveis da CLI, argumentos necessários e opcionais e exemplos. Cada comando corresponde a uma categoria específica conforme observado em "Resumo dos Comandos da AppScan Source command line interface (CLI)".

Cada comando consiste em um nome longo e uma abreviação. Por exemplo, o comando para varrer um aplicativo, projeto ou arquivo e criar uma avaliação é scan com sc como a abreviação.

Resumo dos Comandos da AppScan Source command line interface (CLI)

A seguinte tabela lista os comandos da CLI, uma descrição de cada comando e a necessidade ou não de um login.

Tabela 1. Comandos do CLI

Comando e abreviação	Descrição	Login necessário
about (a)	Exibir as informações sobre versão e copyright da Interface da Linha de Comandos do AppScan Source.	
clearcache (cc)	Remova o cache de análise de vulnerabilidade e os dados de assinatura das regras customizadas. Depois de abrir um aplicativo usando o comando openapplication (oa), é possível limpar o cache emitindo clearcache.	
delete (del)	Exclui um objeto-filho do objeto atual.	Sim
deleteassess (da)	Este comando foi renomeado. Consulte o removeassess (da).	
deleteuser (du)	Excluir um usuário a partir do banco de dados do AppScan Source.	Sim
delvar (dv)	Excluir uma única variável.	Sim
details (det)	Listar os detalhes de avaliação para o objeto atual.	Sim
echo	Ecoa todas as entradas e saídas para a tela.	
getaseinfo (gase)	Imprimir configurações do AppScan Enterprise Server.	Sim

Tabela 1. Comandos do CLI (continuação)

Comando e abreviação	Descrição	Login necessário
help (?)	Lista a ajuda para todos os comandos ou um único comando.	
import (im)	Usar o comando importar para incluir projetos (tais como .ppf) para um aplicativo existente.	Sim
info (i)	Lista informações sobre as propriedades e os valores do objeto atual.	Sim
list (ls, dir)	Listar todos os objetos sob o objeto atual na árvore de objetos. A árvore é exibida como uma representação gráfica do ID do objeto, nome e tipo.	Sim
listassess (la)	Listar o ID do objeto e a data/horário de avaliação para o objeto atual na árvore de objetos. Use listassess para obter um ID para uso com o comandodetails.	Sim
listgroups (lgrp)	Lista todos os grupos, suas permissões e uma descrição de cada um.	Sim
listusers (lu)	Lista todos os usuários do AppScan Source.	Sim
log	Ligar ou Desligar criação de log de mensagens.	
login (in)	Efetuar login no AppScan Enterprise Server (substitui login_local (local)).	
login_file	Efetue login no AppScan Enterprise Server usando um arquivo de token (arquivos de token são criados usando a opção -persist com o comando "login (in)" na página 38 do CLI ou ao criar o usuário do AppScan Source for Automation).	
logout (out)	Efetue logout do AppScan Source e finalize a sessão da interface da linha de comandos da AppScan Source command line interface (CLI).	Sim
moduser (mu)	Modifique as informações sobre o usuário, tais como permissões, ID do usuário e nome para um usuário do AppScan Source.	Sim

Tabela 1. Comandos do CLI (continuação)

Comando e abreviação	Descrição	Login necessário
newuser (nu)	Crie um novo usuário do AppScan Source (um nome de usuário válido, senha e nome completo são requeridos). Os usuários do AppScan Source podem existir no repositório do usuário AppScan Enterprise Server e no banco de dados AppScan Source - ou, se você tiver motivos para ter usuários que não possam acessar o servidor, eles podem ser criados localmente como usuários AppScan Source. Você também pode criar um novo usuário do AppScan Source que já existe no AppScan Enterprise Server.	Sim
openapplication (oa)	Esse comando pode ser usado para abrir um aplicativo existente - ou para criar uma novo arquivo de aplicativo AppScan Source.	Sim
openassessmentfile (oaf)	Abre um arquivo de avaliação do AppScan Source (file_name.ozasmt).	Sim
password (passwd)	O comando password permite que você altere sua senha ou, se você tiver permissões de administrador, altere outra senha do usuário.	Sim
printuser (pu)	O comando printuser (pu) exibe informações sobre um único usuário na tela.	Sim
publishassess (pa)	Publique a avaliação atual ou uma avaliação selecionada para o AppScan Source Database. Quando este comando for usado, a avaliação será disponibilizada para um cliente AppScan Source como AppScan Source for Analysis - mas não será disponibilizada para AppScan Enterprise Console (use o comando "publishassessase (pase)" na página 48 para publicar no AppScan Enterprise Console).	Sim

Tabela 1. Comandos do CLI (continuação)

Comando e abreviação	Descrição	Login necessário
publishassessase (pase)	Publique a avaliação atual ou uma avaliação selecionada para o AppScan Enterprise Console. Quando este comando for usado, a avaliação não ficará disponível para clientes AppScan Source como AppScan Source for Analysis (use o comando "publishassess (pa)" na página 47 para publicar nos clientes AppScan Source).	Sim
quit	Termina e fecha a sessão da interface da linha de comandos do AppScan Source. Emite um logout se você estiver conectado.	
record (rc)	Liga ou desliga a gravação.	
refresh (rf)	O Refresh restaura o projeto ou aplicativo atual do disco.	Sim
register (reg)	Projetos e aplicativos de registro com o banco de dados do AppScan Source.	Sim
removeassess (da)	Remova a avaliação selecionada ou atual da memória.	Sim
report (rpt)	Report gera um relatório do AppScan Source do tipo especificado, incluindo relatórios de descobertas e relatórios do AppScan Source. Uma licença válida do AppScan Source for Automation é requerida para uso deste comando.	Sim
scan (sc)	Varre um aplicativo (ou todos os aplicativos), projeto ou arquivo. Uma licença válida do AppScan Source for Automation é requerida para uso deste comando.	Sim
script (scr)	Executar um script de comandos.	
setaseinfo (sase)	Especifique configurações do Enterprise Console.	Sim
setcurrentobject (set, cd)	Use setcurrentobject para navegar na árvore de objetos.	Sim
setvar (sv)	Cria uma nova variável ou modifica uma variável existente.	Sim

Tabela 1. Comandos do CLI (continuação)

Comando e abreviação	Descrição	Login necessário
unregister (unreg)	Este comando é usado para cancelar o registro de um aplicativo ou projeto do nó atual registrado anteriormente.	Sim

Os seguintes comandos foram removidos da CLI ou foram descontinuados:

- add: Descontinuado. Não usar.
- listproducts (lprod): Descontinuado. Não usar.
- liststopobject (lstop): Descontinuado. Não usar.
- login_admin: Removido. Não usar.
- login_local (local): Descontinuado. Use login (in).
- new: Descontinuado. Use openapplication (oa).
- reset (r): Removido. Não usar.
- runassess (ra): Descontinuado. Use scan (sc).

about (a)

Descrição

Exibir as informações sobre versão e copyright da Interface da Linha de Comandos do AppScan Source.

Sintaxe

sobre

Exemplo

```
AllApplications>> About
```

```
-----
Security AppScan Source 8.5.0.0 Construção 175.

Licensed Materials - Property of IBM Corp. (C) Copyright IBM"
Corp. e seus licenciadores 2003, 2011. Todos os Direitos Reservados. IBM,
o logotipo IBM, ibm.com, Rational, AppScan e ClearQuest são
marcas comerciais ou marcas registradas da International Business
Machines Corp., registradas em vários países no mundo todo.
Outros nomes de produtos e serviços podem ser marcas comerciais
registradas da IBM ou de outras empresas. Uma lista atual de marcas comerciais
registradas da IBM está disponível na Web em "Copyright and trademark information"
em www.ibm.com/legal/copytrade.shtml. Linux é uma marca
registrada de Linus Torvalds nos Estados Unidos e/ou em outros
países. Microsoft, Windows, Windows NT e o logotipo
Windows são marcas comerciais registradas da Microsoft Corporation nos
Estados Unidos e/ou em outros países. UNIX é uma marca
registrada do The Open Group nos Estados Unidos e em outros países. Java e todas
as marcas comerciais registradas e logotipos baseados em Java são
marcas comerciais registradas da Sun Microsystems, Inc. nos Estados Unidos
e/ou em outros países.

Esse programa inclui: Jacobr 2.3.0, Copyright 1997-2006 O
projeto JacORB; Jericho HTML Parser 2.1, Copyright 2005 Martin
Jericho; e XOM 1.0d22, Copyright 2003 Elliotte Rusty Harold,
estando cada um disponível sob a Gnu Library General Public
License (LGPL), cuja cópia está disponível no arquivo Avisos
que acompanhou esse programa.
-----
```

clearcache (cc)

Descrição

Remova o cache de análise de vulnerabilidade e os dados de assinatura das regras customizadas. Depois de abrir um aplicativo usando o comando `openapplication (oa)`, é possível limpar o cache emitindo `clearcache`.

Depois de emitir esse comando, se tiver a análise incremental Java ativada, a próxima varredura iniciada para o aplicativo ou projeto será uma varredura integral.

Sintaxe

```
clearcache
```

Exemplo

```
AllApplications>> openapplication SimpleIOT
AllApplications\SimpleIOT>> clearcache
AllApplications\SimpleIOT>>
```

Análise incremental para Java

Quando a análise incremental está ativada, os dados de análise são armazenados em cache pelo AppScan Source. Ao varrer novamente o projeto ou o aplicativo, o AppScan Source usa esses dados para determinar as mudanças no código e somente as partes do código impactadas pelas mudanças são analisadas novamente. O resultado final é uma análise integral de seu código, mas em uma fração do tempo.

Sobre Esta Tarefa

A análise incremental é suportada no Windows e no Linux. Quando ativada, a análise incremental é executada em projetos ou aplicativos AppScan Source ou em projetos ou áreas de trabalho do Eclipse. Depois de ativar a análise incremental, a primeira varredura a ser executada no projeto, aplicativo ou área de trabalho é sempre uma varredura integral (o cache de análise de vulnerabilidade é somente atualizado durante uma varredura integral). Isso permite que o AppScan Source armazene dados em cache para varreduras subsequentes. As varreduras de seu projeto, aplicativo ou área de trabalho a partir daí serão varreduras incrementais, contanto que o cache de análise de vulnerabilidade não tenha sido limpo e contanto que o número de arquivos alterados não exceda uma configuração de limite que possa ser determinada.

Para ativar e usar a análise incremental, siga estas etapas:

Procedimento

1. Abra `<data_dir>\config\scan.ozsettings` em um editor de texto (em que `<data_dir>` é o local dos dados do programa do AppScan Source, conforme descrito em Capítulo 9, “Locais de Arquivos de Instalação e de Dados do Usuário”, na página 149). Localize a configuração `incremental_analysis` no arquivo. Esta configuração será semelhante a:

```
<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"
  />

```

Nesta configuração, modifique o atributo `value`. Se o atributo for configurado como `true`, esta configuração será `on`. Se estiver configurado como `false`, o AppScan Source não executará a análise incremental durante a varredura.

- Em `<data_dir>\config\scan.ozsettings`, localize a configuração `percentage_of_files_changed`:

```

<Setting
  name="percentage_of_files_changed"
  read_only="false"
  default_value="50"
  description="Na varredura incremental, se a porcentagem de arquivos
  sendo mudados desde a última varredura exceder o limite, a varredura
  integral será iniciada. A porcentagem varia entre 0 e 100.
  O limite padrão é 50, o que representa 50%."
  type="int"
  value="50"
  display_name="Porcentagem de arquivos sendo mudados"
  hidden="true"
/>

```

Essa configuração permite especificar a porcentagem de arquivos que precisam ser mudados antes do início de uma varredura integral. Por padrão, essa porcentagem de limite é de 50%, o que significa que, se varrer novamente depois de 50% ou mais dos arquivos do projeto, aplicativo ou área de trabalho terem mudado, uma varredura integral será iniciada em vez de uma varredura de análise incremental. Nessa configuração, mude o atributo `value`, conforme desejado, em sua porcentagem de limite preferencial.

- Salve `<data_dir>\config\scan.ozsettings` depois de ter modificado todas as configurações relevantes e, então, inicie ou reinicie o produto do AppScan Source que suporta a análise incremental. Por exemplo, reinicie o AppScan Source for Analysis, o Plug-in do AppScan Source for Development Eclipse ou o AppScan Source command line interface (CLI) ou reinicie o serviço do AppScan Source for Automation.
- Agora, ao varrer novamente aplicativos ou projetos Java com a mesma configuração de varredura, a análise incremental será executada se o número de arquivos mudados não exceder o limite e se o cache de análise de vulnerabilidade não tiver sido limpo.
- Limpeza do cache de análise de vulnerabilidade:** se houver problemas com a varredura incremental ou se desejar executar uma varredura de análise integral quando a análise incremental estiver ativada, limpe o cache de vulnerabilidade antes de varrer novamente:
 - AppScan Source for Analysis:
 - Abra a visualização Propriedades para o projeto do AppScan Source. Se estiver varrendo um aplicativo, abra a visualização de propriedades de qualquer projeto-filho (excluir o cache de um projeto também excluirá o cache de seu aplicativo).
 - Na guia Visão Geral, clique em **Limpar cache**.
 - Plug-in do AppScan Source for Development Eclipse: Exclua `<data_dir>\temp\workspace\<project>`, em que:
 - `<data_dir>` é o local dos dados do programa do AppScan Source, conforme descrito em Capítulo 9, “Locais de Arquivos de Instalação e de Dados do Usuário”, na página 149.

- <workspace> é o nome da área de trabalho do Eclipse que está sendo varrida. Para excluir o cache da área de trabalho inteira, exclua todo o diretório <data_dir>\temp\<workspace>.
- <project> é o nome do projeto do Eclipse que está sendo varrido. Para excluir o cache do projeto, exclua o diretório <data_dir>\temp\<workspace>\<project>.
- O AppScan Source command line interface (CLI): use o comando `clearcache`, conforme descrito no *IBM Security AppScan Source Utilities User Guide*.
- AppScan Source for Automation: use o argumento `-clearcache` do comando `ScanApplication`, conforme descrito no *IBM Security AppScan Source Utilities User Guide*.

Resultados

Depois de varrer no AppScan Source for Analysis, é possível usar o recurso **Diff de Avaliação** para comparar as avaliações antes e depois das mudanças no código.

Dica:

- Para forçar uma varredura de análise integral, desative a análise incremental ou limpe o cache de análise de vulnerabilidade.
- Ao executar a análise incremental, é necessário executar uma varredura de análise integral após fazer qualquer uma dessas modificações:
 - Mudanças na regra de segurança ou mudanças nas regras customizadas aplicáveis ao projeto ou ao aplicativo.
 - Mudanças na configuração de varredura.
 - Mudanças nos arquivos `.ozsettings` que afetam as varreduras.
 - Mudanças em propriedades do aplicativo ou do projeto. Por exemplo, qualquer mudança feita na visualização Propriedades do AppScan Source for Analysis para Todos os Aplicativos ou em um aplicativo ou projeto selecionado.
 - Incluir um novo projeto em um aplicativo ou excluir um projeto existente.
 - Excluir arquivos de varreduras. Por exemplo, no AppScan Source for Analysis, é possível optar por excluir um arquivo da varredura, clicando nele como o botão direito na visualização Explorer e escolhendo **Excluir das varreduras**.
- Informações atuais sobre a análise incremental podem ser localizadas em <http://www.ibm.com/support/docview.wss?uid=swg21994390>.

Nota:

- Após uma varredura incremental, marcadores de provas no editor podem não estar mais na localização correta.
- Provas corrigidas que não possuem um rastreo poderão aparecer em resultados de varredura incremental.
- Não é possível ter mais de um produto ou componente AppScan Source aberto ao mesmo tempo durante varreduras incrementais. Além disso, outro usuário não pode varrer o mesmo aplicativo ou projeto conforme você estiver varrendo, ao mesmo tempo e na mesma máquina.

delete (del)

Descrição

Exclui um objeto-filho do objeto atual.

Use o comando `list (ls, dir)` para visualizar os filhos do objeto atual.

Identifique o objeto-filho para excluir por nome ou ID.

Sintaxe

```
del {name|id object_id}
```

- `name`: Nome do objeto-filho. Para visualizar os filhos do objeto atual, emita um comando `list`.
- `id`: Literal. Indica a exclusão por ID do objeto.
- `object_id`: Argumento necessário ao excluir por ID. O `object_id` é o ID numérico do filho para excluir.

Exemplos

- Para excluir um aplicativo denominado `App1`:

```
AllApplications>> Delete App1
```
- Excluir um projeto com ID 40:

```
AllApplications\App1>> Delete id 40
```

deleteassess (da)

Este comando foi renomeado. Consulte o `removeassess (da)`.

deleteuser (du)

Descrição

Excluir um usuário a partir do banco de dados do AppScan Source.

Sintaxe

```
du username
```

`username`: O usuário do AppScan Source a ser excluído.

Exemplo

Para excluir o usuário, `agraham`:

```
AllApplications>> deleteuser agraaham  
AllApplications>> Excluiu o usuário 'agraham'.
```

delvar (dv)

Descrição

Excluir uma única variável.

Sintaxe

```
delvar <nome da variável>
```

variable name: O nome da variável para excluir. Se uma variável desse nome existir, ela será excluída imediatamente, seguida por uma mensagem de êxito.

Exemplo

Para excluir a variável myvar:
AllApplications>> delvar myvar
Excluída a variável '%myvar%'.

details (det)

Descrição

Listar os detalhes de avaliação para o objeto atual.

Recuperar detalhes sobre uma avaliação atual ou uma avaliação selecionada por ID de Avaliação. Use o listassess (1a) para obter uma lista de avaliações e de seus IDs.

Dica: Use o comando log para ativar a criação de log antes de usar o comando details. É possível gerar um arquivo de log em formato de texto delimitado por vírgulas (.csv) que o Microsoft Excel ou um outro programa que aceita dados CSV pode abrir.

Sintaxe

details [id]

id: ID do objeto para uma avaliação da sessão atual. Se um ID não for especificada, são exibidos detalhes (ou enviados para um arquivo de log) para a avaliação mais recente.

Exemplo

Para exibir os detalhes da avaliação mais recente:

```
AllApplications\Myapps>> details
Arquivo, linha, col, nome, tipo, gravidade, confiabilidade
C:\MyApps\WebGoat\webgoat\src\lessons\CommandInjection.java
, 119
, 0
, java.lang.Throwable.printStackTrace
, Vulnerability.Info
, 3-SeverityType_info
, 3-ConfidenceType_low
.
.
C:\MyApps\WebGoat\webgoat\src\session\ECSFactory.java
, 625
, 0
, java.lang.String.equalsIgnoreCase
, Vulnerability.Info
, 3-SeverityType_info
, 3-ConfidenceType_low

-----
Total de Sites de Chamada: 1283
Total Definitive Security Findings with High Severity: 10
Total Definitive Security Findings with Medium Severity: 15
Total Definitive Security Findings with Low Severity: 53
```

```
Total Suspect Security Findings with High Severity: 24
Total Suspect Security Findings with Medium Severity: 25
Total Suspect Security Findings with Low Severity: 6
Total Scan Coverage Findings with High Severity: 123
Total Scan Coverage Findings with Medium Severity: 69
Total Scan Coverage Findings with Low Severity: 56
Total de Linhas: 17197
Max V-Density: 929.8482293423273
Max V/kloc: 22.155027039599933
V-Density: 929.8482293423273
V/kloc: 22.155027039599933
```

echo

Descrição

Ecoa todas as entradas e saídas para a tela.

Este comando geralmente é usado em um script.

Sintaxe

`echo`

Exemplo

Para enviar comentários:

```
3/11/08 2:15 PM AllApplications>>
3/11/08 2:16 PM echo "esse é um teste"
3/11/08 2:16 PM esse é um teste
```

getaseinfo (gase)

Descrição

Imprimir configurações do AppScan Enterprise Server.

Esse comando exibe a configuração do AppScan Enterprise Server que foi configurada pelo comando `setaseinfo (sase)`.

Sintaxe

`getaseinfo`

Nota:

- Você deve ter assinado o AppScan Source command line interface (CLI) com a permissões Gerenciar Configurações do AppScan Enterprise para poder configurar a url. Para obter informações sobre contas de usuários e permissões, consulte a seção *Administrando AppScan Source* do *IBM Security AppScan Source Installation and Administration Guide*.
- O ID do usuário e senha são armazenados na máquina que está executando o cliente AppScan Source (por exemplo, AppScan Source for Analysis) - enquanto a url é armazenada no Enterprise Server (que pode estar localizado em uma máquina remota). Não é possível acessar as informações `userid` e `password` da máquina remota (por exemplo, emitindo o comando `getaseinfo` a partir dela).

Exemplo

```
AllApplications>> getaseinfo
```

```
Username:      my_domain\my_username  
URL:          http://my_aseserver/ase
```

help (?)

Descrição

Lista a ajuda para todos os comandos ou um único comando.

A ajuda da AppScan Source command line interface (CLI) exibe a sintaxe de comando, os aliases e o uso da amostra de cada comando. A ajuda está disponível para comandos individuais ou você pode solicitar uma lista de todos os comandos com uma visão geral de síntese de cada comando.

Sintaxe

```
help [command]
```

command: o comando da CLI para o qual exibir a ajuda online. Você deve estar em uma posição na árvore em que a ajuda esteja disponível para o objeto específico.

Exemplos

- Listar a ajuda para todos os comandos válidos:

```
AllApplications>> ?
```
- Listar a Ajuda para o comando register (reg):

```
AllApplications>> Help register
```

import (im)

Descrição

Usar o comando importar para incluir projetos (tais como .ppf) para um aplicativo existente.

Ao usar este comando para incluir um projeto, esses arquivos de projeto são suportados:

- AppScan Source arquivos de projeto (.ppf).
- Microsoft Visual Studio C/C++ (.vcproj)
- Microsoft Visual Studio C# (.csproj)
- Microsoft Visual Studio Visual Basic (.vbproj)

Ao importar projetos, caracteres curinga podem ser usados em lugar de nomes de arquivo e/ou extensões de arquivo e você pode especificar um argumento adicional para indicar se todos os projetos sob o diretório especificado devem ser importados (importação recursiva).

Sintaxe

```
import path [true|false]
```

- O número de argumentos varia dependendo do tipo do objeto importado.
- path: Caminho completo do objeto importado.

- true ou false: Opcional. Inclua ou omita subdiretórios. O padrão é false (omita subdiretórios).

Exemplo

```
AllApplications\testit>> import c:\testapps\java\webgoat\*.ppf
AllApplications\testit>> ls
23942: webgoat (Project [local])
```

info (i)

Descrição

Lista informações sobre as propriedades e os valores do objeto atual.

Sintaxe

Info

Exemplo

Para listar o ID e o nome do aplicativo atual:

```
AllApplications\Myapp>> Info
```

A saída é semelhante a:

```
object: Myapp(Application)
id: 1
```

...

list (ls, dir)

Descrição

Listar todos os objetos sob o objeto atual na árvore de objetos. A árvore é exibida como uma representação gráfica do ID do objeto, nome e tipo.

Sintaxe

list [*all*]

- all: Opcional. Exibe todos os objetos na árvore de objetos.
- Transmitir nenhum argumento lista somente os filhos imediatos do objeto atual.

Exemplos

- Listar os objetos na árvore de objetos atual:

```
AllApplications>> List
1: WebGoat_4_0 (Application [local] [registered])
2: test_application (Application [local] [registered])
```

- To list all objects in the Object Tree:

```
AllApplications>> LS all
```

```
1: WebGoat_4_0 (Application [local] [registered])
|----198: compile (Project [local] [registered])
|----|----470: WebContent\lp\JavaScriptValidation.html (Source)
|----|----475: WebContent\lp\RemoteAdminFlaw.html (Source)
|----|----483: WebContent\lp>WelcomeScreen.html (Source)
|----|----474: WebContent\lp\ReflectedXSS.html (Source)
|----|----477: WebContent\lp\SqlInjection.html (Source)
. . .
```

listassess (1a)

Descrição

Listar o ID do objeto e a data/horário de avaliação para o objeto atual na árvore de objetos. Use listassess para obter um ID para uso com o comandodetails.

Sintaxe

```
listassess [all]
```

all: Opcional. Lista todas as avaliações na memória.

Exemplos

- Para listar as informações do objeto de avaliação atual:

```
AllApplications\WebGoat_4_0>> listassess
```

```
AllApplications\WebGoat_4_0>> 4762: WebGoat_4_0 (Application, Tue Mar 11 14:20:23 EDT 2008)
```

```
AllApplications\WebGoat_4_0>> 9033: WebGoat_4_0 (Application, Tue Mar 11 14:43:31 EDT 2008)
```

- Para listar todas as informações do objeto de avaliação:

```
AllApplications\Applications>> ListAssess all
```

```
542: putty (Application, Fri Aug 04 08:38:18 EDT 2008)
```

```
1804: JavaAny (Application, Tue Apr 01 13:17:33 EDT 2008)
```

```
2971: snare (Application, Tue Apr 01 08:42:53 EDT 2008)
```

```
4773: SimpleIoT (Project, Tue Apr 01 07:31:11 EDT 2008)
```

```
4874: JSPWiki (Application, Tue Apr 01 13:22:08 EDT 2008)
```

listgroups (1grp)

Descrição

Lista todos os grupos, suas permissões e uma descrição de cada um.

Sintaxe

```
listgroups
```

Exemplo

```
AllApplications>> ListGroups
```

```
-----  
Grupo: APPS  
Descrição: Gerenciamento do Aplicativo e do Projeto  
Permissões:  
REGISTER (Registrar)  
ATTRAPPLY (Aplicar Atributos)  
ATTRMODIFY (Gerenciar Atributos)  
SCAN (Scan)  
VIEWREGISTER (Visualizar Registrado)  
Grupo: ASSESSMENTS  
Descrição: Gerenciamento de Avaliação  
Permissões:  
ASMNTDELETE (Excluir Avaliações Publicadas)  
ASMNTPUBLISH (Publicar Avaliações)  
ASMNTSAVE (Salvar Avaliações)  
ASMNTVIEWPUBLISH (Visualizar Avaliações Publicadas)  
Grupo: KB  
Descrição: Gerenciamento da Base de Conhecimento  
Permissões:
```

```
CUSTOM (Gerenciar Regras Customizadas)
PATTERN (Gerenciar Padrões)
Grupo: ADMIN
Descrição: Administração
Permissões:
ASE (Gerenciar Configurações Corporativas do AppScan)
LDAP (Gerenciar Configurações do LDAP)
USER (Gerenciar Usuários)
```

listusers (lu)

Descrição

Lista todos os usuários do AppScan Source.

Se disponível, a saída incluirá o ID do usuário, o nome e o repositório do usuário.

Sintaxe

```
listusers
```

Exemplo

Para listar todos os usuários do AppScan Source:

```
AllApplications>> Listusers
All Users
  admin      (myASEServer\admin)
  jsmith     (John Smith) local
  joandarcy  (Joan Darcy) local
AllApplications>>
```

log

Descrição

Ligar ou Desligar criação de log de mensagens.

Registre uma sessão em um arquivo especificado ou visualize o status da criação de log atual. A sessão inteira, incluindo a saída, é registrada no arquivo especificado.

Dica: Para registrar somente a entrada do comando, use o comandorecord (rc).

Sintaxe

```
log {on file|off}
```

- O argumento on ou off é necessário para ligar ou desligar a criação de log. Log sem argumentos identifica o estado atual.
- on: Necessário para ligar a criação de log. On requer um nome de file.
- file: Nome do arquivo de log. É possível especificar o tipo de arquivo a gerar, como .txt ou .csv.
- off: Desliga a sessão atual de criação de log.

Exemplos

- Para registrar um arquivo denominado MyLogFile.txt:
AllApplications>> Log on c:\MyLogFile.txt

- Para desligar a criação de log:
AllApplications>> Log off

login (in)

Descrição

Efetuar login no AppScan Enterprise Server (substitui login_local (local)).

Se o login for bem-sucedido, o prompt muda para AllApplications>> para indicar que você está com login efetuado.

Sintaxe

login server user_id password [-persist] [-acceptssl]

- **server**: Obrigatório. Especifique a URL para a instância do AppScan Enterprise Server. O formato dessa URL é `http(s)://<hostname>:<port>/ase`, em que `<hostname>` é o nome da máquina na qual o AppScan Enterprise Server foi instalado e `<port>` é a porta na qual o servidor está em execução. Um exemplo dessa URL é `https://myhost.mydomain.ibm.com:9443/ase`.
- **user_id**: requerido.
 - Se o seu método de autenticação do AppScan Enterprise Server for ID do usuário e senha, insira seu ID do usuário.
 - Se o seu AppScan Enterprise Server estiver configurado para usar autenticação do Windows, insira o domínio e o nome do usuário usado para conectar ao Enterprise Console (separe o domínio e o nome do usuário com \ - por exemplo, `my_domain\my_username`).
 - Se seu AppScan Enterprise Server estiver configurado com LDAP, insira o nome de usuário que você usa para se conectar ao Enterprise Console.
 - No Windows, se o AppScan Enterprise Server estiver ativado para autenticação Common Access Card (CAC), insira `common_name<cn_certificate>`, em que `common_name` é o seu Nome Comum (CN) do CAC e `cn_certificate` é o CN do emissor do certificado. Se o seu `common_name` ou `cn_certificate` contiver espaços, coloque o valor entre aspas ("`common_name<cn_certificate>`").
- **password**: Obrigatório se o método de autenticação do AppScan Enterprise Server for ID do usuário e senha. Especifique a senha para seu ID de usuário do Enterprise Server.

Nota: Essa opção não será usada quando o AppScan Enterprise Server estiver ativado para autenticação de CAC

- **-persist**: Opcional. Persista credenciais de login através de uma chave criptografada. Se o argumento `persist` for usado, o `login` salva o nome de usuário e a senha em um arquivo criptografado chamado `cli.token`. O `cli.token` é salvo na pasta `.ounce` do diretório `home` dos usuários.
- **-acceptssl**: Opcional. Aceitar certificados SSL automaticamente. Para obter mais informações, consulte o “Certificados SSL do AppScan Enterprise Server” na página 39

Importante: Se este argumento não estiver incluído, efetuar login ou publicar no AppScan Enterprise Console irá falhar com erro se um certificado inválido for encontrado (se você ainda não aceitou permanentemente o certificado enquanto efetua o login por outro produto do cliente AppScan Source).

Exemplos

- O usuário John Smith efetua login no AppScan Enterprise Server que foi configurado por seu administrador:

```
>> login https://myserver:9443/ase/ johnsmith mypassword -acceptssl
Login successful.
AllApplications>>
```
- No Windows, o usuário Jane Smith efetua login em um AppScan Enterprise Server que está ativado para autenticação CAC. Seu CN é janesmith e o CN do certificado da empresa é mydomain-co-ABC123-C0:

```
>> login https://myserver:9443/ase/ janesmith<mydomain-co-ABC123-C0>
```

Ela é então solicitada por uma caixa de diálogo da Segurança do Windows que insira o PIN do cartão CAC. Quando concluir o PIN na caixa de diálogo, a linha de comandos exibirá isso:

```
Login successful.
AllApplications>>
```

Certificados SSL do AppScan Enterprise Server

Quando o AppScan Enterprise Server estiver instalado, ele deverá ser configurado para usar um certificado SSL válido. Se isto não estiver feito, você receberá uma mensagem de conexão não confiável quando efetuar o login no servidor a partir do AppScan Source for Analysis ou do AppScan Source command line interface (CLI) - ou AppScan Source for Development on Windows e Linux.

Local de Armazenamento do Certificado SSL

Os certificados que foram permanentemente aceitos são armazenados em <data_dir>\config\cacertspersonal e em <data_dir>\config\cacertspersonal.pem (em que <data_dir> é o local dos dados do programa do AppScan Source, conforme descrito em Capítulo 9, “Locais de Arquivos de Instalação e de Dados do Usuário”, na página 149). Remova estes dois arquivos se não desejar mais que os certificados sejam armazenados permanentemente.

AppScan Source for Automation e a Validação de Certificados SSL

Por padrão, os certificados são automaticamente aceitos ao usar o AppScan Source for Automation. Esse comportamento é determinado pela configuração ounceautod_accept_ssl no arquivo de configuração do Automation Server (<data_dir>\config\ounceautod.ozsettings (em que <data_dir> é o local dos dados do programa do AppScan Source, conforme descrito em Capítulo 9, “Locais de Arquivos de Instalação e de Dados do Usuário”, na página 149)). Se esta configuração for editada para que value="true" seja configurado como value="false", a validação de SSL será tentada e a criação de log ou publicação no AppScan Enterprise Console falhará com um erro, se for encontrado um certificado inválido.

AppScan Source command line interface (CLI) e a Validação de Certificados SSL

Por padrão, ao usar o comando CLI login, a validação SSL será tentada e efetuar login ou publicar no AppScan Enterprise Console irá falhar com erro se um certificado inválido for encontrado (se você ainda não tiver aceitado permanentemente o certificado enquanto efetua o login por outro produto do cliente AppScan Source). Esse comportamento pode ser modificado usando o parâmetro -acceptssl da opção ao emitir o comando login. Quando esse parâmetro for usado, os certificados SSL serão aceitos automaticamente.

login_file

Descrição

Efetue login no AppScan Enterprise Server usando um arquivo de token (arquivos de token são criados usando a opção `-persist` com o comando `login (in)` na página 38 do CLI ou ao criar o usuário do AppScan Source for Automation).

Se o login for bem-sucedido, o prompt muda para `AllApplications>>` para indicar que você está com login efetuado.

Sintaxe

`login_file server token_file [-acceptssl]`

- `server`: Obrigatório. Especifique a URL para a instância do Enterprise Server.
- `token_file`: Obrigatório. Indica o caminho e o nome do arquivo de token. Se o arquivo de token foi criado com o comando `login (in)` na página 38 do CLI, o arquivo será `ouncecli.token` e residirá na pasta `.ounce` do diretório home dos usuários. Se o arquivo de token foi criado para AppScan Source for Automation, o arquivo será `<data_dir>/config/ounceautod.token` (em que `<data_dir>` é o local dos dados do programa do AppScan Source, conforme descrito em Capítulo 9, “Locais de Arquivos de Instalação e de Dados do Usuário”, na página 149).
- `-acceptssl`: Opcional. Aceitar certificados SSL automaticamente. Para obter mais informações, consulte o “Certificados SSL do AppScan Enterprise Server” na página 39

Importante: Se este argumento não estiver incluído, efetuar login ou publicar no AppScan Enterprise Console irá falhar com erro se um certificado inválido for encontrado (se você ainda não aceitou permanentemente o certificado enquanto efetua o login por outro produto do cliente AppScan Source).

Exemplo

O usuário John Smith efetua login no AppScan Enterprise Server, usando um arquivo de token que foi criado no Windows a partir do CLI:

```
>> login_file https://myserver:9443/ase/  
C:\Users\Administrator\.ounce\ouncecli.token  
Login successful.  
AllApplications>>
```

login_local (local)

Descrição

Descontinuado no AppScan Source Versão 6. Use `login (in)`.

logout (out)

Descrição

Efetue logout do AppScan Source e finalize a sessão da interface da linha de comandos da AppScan Source command line interface (CLI).

Se você efetuou login, o `logout` é um comando inválido.

Sintaxe

Logout

Exemplo

Para efetuar logout:

```
AllApplications>> Logout
>> Logout Efetuado.
>> Saindo a Interface da Linha de Comandos do Security AppScan Source...
```

moduser (mu)

Descrição

Modifique as informações sobre o usuário, tais como permissões, ID do usuário e nome para um usuário do AppScan Source.

Sintaxe

```
moduser --userid|-u <user id>
[--fullname|-f <user first and last name>]
[--group [group[:permission[:permission...]]
        [--group...]]
[--removegroup [group[:permission
                [:permission...]] [--removegroup...]]]
```

Nome do usuário e nome

- --username|-u: Necessário. Um nome do usuário válido do AppScan Source.
- --fullname|-f: Opcional. Nome completo do usuário. Se a entrada incluir espaços, coloque-a entre os símbolos " (por exemplo, -f "Joe Smith").

Grupos e Permissões

Opcional.

Grupos e permissões identificam as tarefas permitidas do AppScan Source para aquele usuário. Tarefas não identificadas especificamente como parte de uma permissão estão disponíveis a todos os usuários:

--group: Os grupos e permissões de grupo a ser incluídos para esse usuário. Especificar um grupo sem nenhuma permissão concede ao usuário todas as permissões nesse grupo.

ou

--removegroup: Os grupos e permissões de grupo a ser removidos desse usuário. Especificar um grupo sem nenhuma permissão remove todas as permissões nesse grupo.

Os grupos e permissões são:

- ASSESSMENTS: Permissões de nível de avaliação.
 - ASMNTDELETE: Excluir avaliações publicadas.
 - ASMNTPUBLISH: Publicar avaliações.
 - ASMNTSAVE: Salvar avaliações.
 - ASMNTVIEWPUBLISH: Visualizar avaliações publicadas.

- ADMIN: Permissões administrativas.
 - ASE: Gerenciar configurações do AppScan Enterprise
 - USER: Gerenciar configurações do usuário incluindo a inclusão e exclusão de usuários e alteração de permissões de usuário.
- APPS: Permissões de nível de Aplicativo e de Projeto
 - ATTRAPPLY: Aplicar atributos a aplicativos.
 - ATTRMODIFY: Criar, excluir e modificar atributos.
 - VIEWREGISTER: Visualizar aplicativos e projetos registrados.
 - REGISTER: Registrar/cancelar registro de aplicativos e projetos. Implica permissão VIEWREGISTER.
 - SCAN: Varre aplicativos e projetos.
- KB: Permissões de gerenciamento do Base de conhecimento.
 - CUSTOM: Gerenciar regras customizadas.
 - PATTERN: Criar, editar ou excluir padrões.
- FILTER: Gerenciamento de filtro.
 - SHAREDFILTERS: Gerenciar filtros compartilhados.
- SCANCONFIG: Varrer gerenciamento de configuração
 - SHAREDCONFIGS: Gerenciar configurações de varredura compartilhadas.

Exemplo

Após as credenciais de usuário de Joan Darcy serem criadas (com o comando `newuser (nu)`), o Administrador do Sistema determinou que ela precisa somente salvar, publicar e visualizar as permissões; mas não excluir a permissão. Adicionalmente, Joan precisa das permissões Padrões do Base de conhecimento:

```
moduser --userid joandarcy --removegroup
ASSESSMENTS:ASMNTDELETE --group KB:PATTERN
```

newuser (nu)

Descrição

Crie um novo usuário do AppScan Source (um nome de usuário válido, senha e nome completo são requeridos). Os usuários do AppScan Source podem existir no repositório do usuário AppScan Enterprise Server e no banco de dados AppScan Source - ou, se você tiver motivos para ter usuários que não possam acessar o servidor, eles podem ser criados localmente como usuários AppScan Source. Você também pode criar um novo usuário do AppScan Source que já existe no AppScan Enterprise Server.

Nota: O comando `newuser (nu)` não será aplicado se o AppScan Enterprise Server estiver ativado para autenticação Common Access Card (CAC).

Sintaxe

```
newuser --userid|-u <user id>
--password|-p <senha>
--fullname|-f <user first and last name>
[--group [group[:permission[;permission...]]
[--group...]]
```

Informações de Identificação

- `--userid|-u`: Obrigatório. ID do usuário. Nenhum espaço é permitido.
- `--password|-p`: Senha de usuário.
- `--fullname|-f`: Nome completo do usuário. Se a entrada incluir espaços, coloque-a entre os símbolos " (por exemplo, `-f "Joe Smith"`).

Grupos e Permissões

Optional. Grupos e permissões identificam as tarefas permitidas do AppScan Source para aquele usuário. Tarefas não identificadas especificamente como parte de uma permissão estão disponíveis a todos os usuários:

`--group`: Os grupos e permissões de grupo para esse usuário. Especificar um grupo sem nenhuma permissão concede ao usuário todas as permissões nesse grupo. Os grupos e suas permissões são:

- **ASSESSMENTS**: Permissões de nível de avaliação.
 - **ASMNTDELETE**: Excluir avaliações publicadas.
 - **ASMNTPUBLISH**: Publicar avaliações.
 - **ASMNTSAVE**: Salvar avaliações.
 - **ASMNTVIEWPUBLISH**: Visualizar avaliações publicadas.
- **ADMIN**: Permissões administrativas.
 - **ASE**: Gerenciar configurações do AppScan Enterprise
 - **USER**: Gerenciar configurações do usuário incluindo a inclusão e exclusão de usuários e alteração de permissões de usuário.
- **APPS**: Permissões de nível de Aplicativo e de Projeto
 - **ATTRAPPLY**: Aplicar atributos a aplicativos.
 - **ATTRMODIFY**: Criar, excluir e modificar atributos.
 - **VIEWREGISTER**: Visualizar aplicativos e projetos registrados.
 - **REGISTER**: Registrar/cancelar registro de aplicativos e projetos. Implica permissão **VIEWREGISTER**.
 - **SCAN**: Varre aplicativos e projetos.
- **KB**: Permissões de gerenciamento do Base de conhecimento.
 - **CUSTOM**: Gerenciar regras customizadas.
 - **PATTERN**: Criar, editar ou excluir padrões.
- **FILTER**: Gerenciamento de filtro.
 - **SHAREDFILTERS**: Gerenciar filtros compartilhados.
- **SCANCONFIG**: Varrer gerenciamento de configuração
 - **SHAREDCONFIGS**: Gerenciar configurações de varredura compartilhadas.

Autenticação LDAP

Você não pode incluir usuários LDAP no repositório do usuário do AppScan Source se eles ainda não estiverem no repositório do usuário do AppScan Enterprise Server. Para incluir um usuário do AppScan Source que será autenticado por meio do LDAP, você deverá ter configurado o repositório do usuário do AppScan Enterprise Server para usar um repositório LDAP. Para obter informações sobre isso, consulte o AppScan Enterprise Server *Planning & Installation Guide*.

Se você estiver usando a autenticação LDAP e desejar incluir um usuário do AppScan Source que não é parte de um grupo de usuários LDAP, emita o comando `newuser`.

Exemplo

Crie um usuário chamado Joan Darcy no AppScan Enterprise Server. Seu nome de usuário é `joandarcy` e sua senha é `123456`. Joan pode usar o AppScan Source com todas as permissões nos grupos `APPS` e `ASSESSMENTS`, bem como permissões de regras customizadas dentro do grupo `KB`:

```
AllApplications>> newuser --userid joandarcy --password 123456
--fullname "Joan Darcy" --group APPS --group ASSESSMENTS --group KB:CUSTOM
AllApplications>> Created user 'joandarcy'. ID do Usuário: 888
```

openapplication (oa)

Descrição

Esse comando pode ser usado para abrir um aplicativo existente - ou para criar uma novo arquivo de aplicativo AppScan Source.

Ao usar este comando para abrir um aplicativo, estes arquivos de aplicativo são suportados:

- Arquivos de Aplicativo do AppScan Source (`.paf`).
- Eclipse ou áreas de trabalho Rational Application Developer for WebSphere Software (RAD) (`.ewf`)

Nota: Os arquivos `.ewf` são gerados quando você usa `openapplication` para abrir um diretório da área de trabalho (especificando seu caminho).

- Arquivos WAR (`.war`)
- Arquivos EAR (`.ear`)
- Somente Windows: Microsoft arquivos Visual C++ Workspace (`.dsw`)
- Somente Windows: Microsoft arquivo da solução Visual Studio.NET (`.sln`)

Nota: Para saber quais versões de arquivos importados são suportados pelo AppScan Source for Analysis, pelo AppScan Source for Automation e pelo Interface da linha de comandos do AppScan Source, consulte <http://www.ibm.com/support/docview.wss?uid=swg27027486>. Nesta página, selecione a guia para a versão do AppScan Source que está usando e, em seguida, selecione o componente do AppScan Source que está usando. Se o AppScan Source suportar a abertura e a varredura de arquivos de outros ambientes de desenvolvimento, esse suporte será listado na seção **Compiladores e idiomas** da guia **Software suportado**.

Sintaxe

```
openapplication file [-appserver_type]
[-include_all_lib_jars] [-include_lib_jars] [-no_ear_project]
```

- `file`: Obrigatório.
 - Se estiver usando o comando para abrir um aplicativo, especifique o caminho e o nome do arquivo do aplicativo existente.
 - Se você estiver usando o comando para criar um aplicativo, AppScan Source especifique um caminho válido e um novo nome de arquivo (assegure-se que o novo nome de arquivo ainda não existi).

- Se você estiver usando o comando para abrir uma área de trabalho do Eclipse ou Rational Application Developer for WebSphere Software (RAD), é possível especificar apenas o caminho da área de trabalho.
- -appserver_type: Opcional. Se o aplicativo que você está abrindo inclui páginas de JavaServer (por exemplo, um arquivo WAR ou EAR), use essa configuração para especificar o servidor de aplicativos para usar a compilação de JSP. Especifique um desses, entre aspas duplas:
 - Tomcat 7
 - Tomcat 8
 - WebSphere 7.0
 - WebSphere 8.0
 - WebSphere 8.5
 - WebLogic 11g
 - WebLogic 12c

Nota:

- Antes de especificar um servidor de aplicação, certifique-se de que tem sido configurado corretamente na preferência AppScan Source for Analysis.
- Se -appserver_type não for usado, o compilador JSP padrão que está definido em AppScan Source for Analysis será usada para compilação de JSP. Fora-da-caixa, Tomcat 7 é o compilador JSP padrão.
- Para arquivos WAR:
 - -include_all_lib_jars: use essa configuração para incluir todas as bibliotecas no arquivo WAR durante a varredura.
 - -include_lib_jars: use essa configuração para especificar as bibliotecas no arquivo WAR que deseja incluir durante a varredura. Ao usar a configuração, não inclua informações do caminho da biblioteca e separe várias bibliotecas com vírgulas.
- -no_ear_project: ao importar um arquivo EAR, um projeto será criado automaticamente para armazenar bibliotecas compartilhadas. Se não houver bibliotecas compartilhadas, o projeto será criado, mas estará vazio. Quando essa configuração for usada, nenhum projeto será criado para o arquivo EAR.

Exemplos

- Para abrir um arquivo no Windows Microsoft Visual Studio .NET solution:
AllApplications>> openapplication c:\mysln.sln
- Para abrir uma área de trabalho Eclipse:
AllApplications>> oa "C:\Users\myname\My Documents\myworkspace"
ou
AllApplications>> oa "C:\Users\myname\My Documents\myworkspace\myworkspace.ewf"
- Para abrir um arquivo WAR e incluir somente algumas de suas bibliotecas ao varrer:
AllApplications>> oa c:\mywar.war -include_lib_jars lib1.jar,lib2.jar

openassessmentfile (oaf)

Descrição

Abre um arquivo de avaliação do AppScan Source (file_name.ozasmt).

Uma avaliação aberta torna-se a avaliação atual.

Sintaxe

```
openassessmentfile path_file
```

path_file: Caminho e nome do arquivo do arquivo de avaliação existente.

Exemplo

Para abrir uma avaliação salva anteriormente:

```
AllApplications>> OpenAssessmentFile c:\priority.ozasmt
```

password (passwd)

Descrição

O comando password permite que você altere sua senha ou, se você tiver permissões de administrador, altere outra senha do usuário.

Para alterar uma senha, digite a nova senha duas vezes. Ambas as entradas da nova senha devem corresponder para que a mudança tenha efeito. Ao alterar sua própria senha, você deve digitar sua senha atual primeiro.

Sintaxe

```
password [user name]
```

username: Nome do usuário cuja senha está sendo alterada. Se você não especificar um nome de usuário, está alterando sua própria senha.

Exemplos

- Para alterar sua própria senha:

```
AllApplications>> Password
Digite a senha atual: *****
Digite a nova senha: *****
Confirme a nova senha: *****
Senha alterada.
AllApplications>>
```

- Para alterar a senha de outro usuário (se você tiver permissões de administrador):

```
AllApplications>> passwd johnsmith
Alterando senha para o usuário johnsmith
Digite a nova senha: *****
Confirme a nova senha: *****
Senha alterada.
AllApplications>>
```

printuser (pu)

Descrição

O comando `printuser (pu)` exibe informações sobre um único usuário na tela.

Sintaxe

```
printuser username
```

username: Nome do usuário do AppScan Source.

Exemplo

Exibir os dados do usuário originais de Joan Darcy:

```
printuser joandarcy
Usuário: joandarcy
Primeiro Nome: Joan
Sobrenome: Darcy
Endereço de e-mail: jdarcy@example.com
Grupos:
- APPS      (Gerenciamento de Aplicativo e de Projeto)
  REGISTER  (Registrar)
  ATTRAPPLY (Aplicar Atributos)
  ATTRMODIFY (Gerenciar Atributos)
  SCAN      (Scan)
  VIEWREGISTER (Visualizar Registrado)
- (ASSESSMENTS) (Gerenciamento de Avaliação)
  ASMTDELETE (Excluir Avaliações Publicadas)
  ASMNTPUBLISH (Publicar Avaliações)
  ASMNTSAVE  (Salvar Avaliações)
  ASMNTVIEWPUBLISH (Visualizar Avaliações Publicadas)
- [KB]      (Gerenciamento de Base de Conhecimento)
  CUSTOM    (Gerenciar Regras Customizadas)
- [ADMIN]   (Administração)
```

Nota: Colchetes quadrados [] indicam que esse usuário não tem todas as permissões nesse grupo.

publishassess (pa)

Descrição

Publique a avaliação atual ou uma avaliação selecionada para o AppScan Source Database. Quando este comando for usado, a avaliação será disponibilizada para um cliente AppScan Source como AppScan Source for Analysis - mas não será disponibilizada para AppScan Enterprise Console (use o comando “`publishassess (pase)`” na página 48 para publicar no AppScan Enterprise Console).

Uma licença válida do AppScan Source for Automation é requerida para uso deste comando.

Quando uma avaliação for publicada com este comando, a avaliação será automaticamente registrada com AppScan Enterprise Server. Não é necessário registrar manualmente a avaliação ao usar esse comando.

Sintaxe

```
pa [id]
```

id: Literal opcional. Identifica o ID de avaliação. Se nenhum ID for incluído, o comando aplica-se à varredura mais recente. É possível usar o comando `listassess (1a)` para localizar o ID de avaliação.

Exemplo

No exemplo a seguir, nós configuramos o objeto atual para o pebble do aplicativo, varremos projeto de pebble e, em seguida, publicamos a avaliação (que é registrada automaticamente ao mesmo tempo).

```
AllApplications>> cd pebble
AllApplications\pebble>> scan pebble
New Scan started

Scanning Project pebble (1 of 1)
Preparing project for scan...
.
.
Preparing for Vulnerability Analysis...
Performing Vulnerability Analysis...
Generating Findings...
Preparing project for scan...
.
.
Scanned Project pebble:
  Total files: 15
  Total findings: 167
  Total lines: 385
vkloc: 0.44448395412925595
v-Density: 22.446439683527426
Scanned Application pebble:
  Total files: 15
  Total findings: 167
  Total lines: 385
vkloc: 0.44448395412925595
v-Density: 22.446439683527426
Scan completed:
  Total files: 15
  Total findings: 167
  Total lines: 385
vkloc: 0.44448395412925595
v-Density: 22.446439683527426
Elapsed Time - 18 Seconds

AllApplications\pebble>> publishassess

Assessment Successfully Published.
```

publishassessase (pase)

Descrição

Publique a avaliação atual ou uma avaliação selecionada para o AppScan Enterprise Console. Quando este comando for usado, a avaliação não ficará disponível para clientes AppScan Source como AppScan Source for Analysis (use o comando “publishassess (pa)” na página 47 para publicar nos clientes AppScan Source).

Sintaxe

```
publishassessase
-aseapplication <ase_application> [id] [path]
[-folder <location>] [-name <published_assessment_name>] [-preventOverwrite]
```

- `-aseapplication <ase_application>`: Esta opção é necessária quando se está conectado ao AppScan Enterprise Server Versão 9.0.3 e superior (a menos que você desative o requisito, conforme descrito aqui). A associação de um aplicativo é opcional quando se está conectado a versões anteriores do AppScan Enterprise Server. Use essa opção para especificar o aplicativo Enterprise Console ao qual associar a avaliação.
- `id`: Literal opcional. Identifica o ID de avaliação. É possível usar o comando `listassess (1a)` para localizar o ID de avaliação.
- `path`: Literal opcional. Caminho e nome do arquivo do arquivo de avaliação.
- `-folder <location>`: Opcional. Essa opção se aplica somente quando estiver conectada a versões do AppScan Enterprise Server anteriores a Versão 9.0.3. Especifique a pasta do Enterprise Console para a qual publicar. Se este argumento não for usado, a avaliação será publicada em sua pasta Enterprise Console padrão.
- `-name <published_assessment_name>`: Opcional. Nome que a avaliação será salva como no Enterprise Console. Se este argumento não for usado, um nome será gerado com base no aplicativo AppScan Source digitalizado para produzir a avaliação (este nome será pré-anexado com AppScan Source:).
- `-preventOverwrite`: opcional. Inclua esse argumento para evitar a publicação se uma avaliação do mesmo nome já existir no servidor.

Quando o argumento opcional for um número inteiro, o comando assume que ele é o ID de avaliação. Quando ele não for um número inteiro, o comando assume que ele é um caminho para um arquivo de avaliação salvo.

Se uma avaliação não for especificada usando o `id` ou o comando `path`, a avaliação gerada pela varredura mais recente será assumida.

Importante:

Ao fazer upgrade para o AppScan Source Versão 9.0.3.4, essas mudanças serão notadas:

- Ao publicar uma avaliação no AppScan Enterprise Console, agora deve-se associar a avaliação com um aplicativo em AppScan Enterprise (se você estiver executando o AppScan Enterprise Server Versão 9.0.3 e superior). Como resultado, scripts de automação podem falhar se não incluírem a associação do aplicativo. No AppScan Enterprise Server, a associação do aplicativo é necessária se desejar aproveitar os recursos de gerenciamento de risco de segurança do aplicativo AppScan Enterprise Server. Consulte http://www.ibm.com/support/knowledgecenter/SSW2NF_9.0.3/com.ibm.ase.help.doc/topics/c_overview.html.
- Além disso, remova a porta da URL do AppScan Enterprise.
 1. No AppScan Source for Analysis, clique em **Editar > Preferências**.
 2. Nas configurações do Console AppScan Enterprise, remova a porta do campo **URL do Console Enterprise**.
- Após publicar sua avaliação, ela estará disponível somente na visualização do monitor do AppScan Enterprise (em liberações anteriores, a avaliação estava disponível na visualização de varredura do AppScan Enterprise). A migração para essa visualização está descrita em http://www.ibm.com/support/knowledgecenter/SSW2NF_9.0.3/com.ibm.ase.help.doc/topics/t_workflow_for_applications.html.

Esse é o resultado de um protocolo de comunicação alterado entre o AppScan Source e o AppScan Enterprise Server que é necessário para publicar para o AppScan Enterprise Server ao usar a autenticação Common Access Card (CAC).

Se não desejar publicar avaliações no AppScan Enterprise Server quando a autenticação CAC estiver habilitada ou se não desejar aproveitar os recursos de gerenciamento de risco de segurança do aplicativo Enterprise Server, é possível reverter para o protocolo de comunicação anterior da seguinte maneira:

1. Abra <data_dir>\config\ounce.ozsettings (em que <data_dir> é o local dos dados do programa do AppScan Source, conforme descrito em Capítulo 9, “Locais de Arquivos de Instalação e de Dados do Usuário”, na página 149)).

2. Neste arquivo, localize esta configuração:

```
<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. Na configuração, mude value="false" para value="true" e, em seguida, salve o arquivo.

4. Reinicie o produto AppScan Source do qual você publicará avaliações.

Quando essa configuração estiver definida como value="true":

- Se você associar uma avaliação a um aplicativo no AppScan Enterprise ao publicar, a avaliação ficará disponível nas visualizações de Monitor e de Varredura.
- Se você não associar uma avaliação a um aplicativo ao publicar, a avaliação ficará disponível na visualização de Varredura.
- Não será possível publicar avaliações no AppScan Enterprise Server quando a autenticação CAC estiver habilitada.

Para obter informações adicionais, consulte <http://www.ibm.com/support/docview.wss?uid=swg21993010>.

quit

Descrição

Termina e fecha a sessão da interface da linha de comandos do AppScan Source. Emite um logout se você estiver conectado.

Sintaxe

quit

Exemplo

```
AllApplications>> Quit
```

```
>> Logout Efetuado.
```

```
>> Saíndo a Interface da Linha de Comandos do Security AppScan Source...
```

record (rc)

Descrição

Liga ou desliga a gravação.

Record captura somente comandos e argumentos emitidos, mas não captura saída.

Dica: Para capturar a saída, use o comando `log`.

Sintaxe

```
record {on file|off}
```

- `on` ou `off`: Necessário para ligar ou desligar a gravação.
- `file`: Nome do arquivo registrado. Record `On` requer um arquivo `file` que inclua o caminho de arquivo válido. Se o arquivo ainda não existir, a AppScan Source command line interface (CLI) o cria. Se o arquivo já existir, os dados do log serão anexados ao final do arquivo.
- Record sem argumentos identifica se a gravação está ligada ou desligada.

Exemplo

- Para registrar um arquivo denominado `MyCommands.txt`:

```
AllApplications>> Record on C:\MyCommands.txt
```
- Para desligar a gravação:

```
AllApplications>> RC off
```

refresh (rf)

Descrição

O Refresh restaura o projeto ou aplicativo atual do disco.

Use o refresh quando desejar recarregar um projeto ou aplicativo que possa ter sido alterado.

Sintaxe

```
refresh
```

Exemplo

Para atualizar `MyProject`:

```
AllApplications\MyApplication\MyProject>> Refresh
```

register (reg)

Descrição

Projetos e aplicativos de registro com o banco de dados do AppScan Source.

O registro permite registrar o aplicativo ou o projeto no nó atual com o AppScan Source. O registro torna o AppScan Source ciente dos aplicativos e projetos. Aplicativos e projetos registrados podem ser publicados (usando `publishsess (pa)`).

Após registrados, somente é possível cancelar o registro de aplicativos ou projetos usando `unregister` (`unreg`).

Sintaxe

`register`

Exemplo

```
AllApplications>> cd pebble
AllApplications\pebble>> register
AllApplications\pebble>> 'pebble' registrado com êxito.
```

`removeassess` (da)

Descrição

Remova a avaliação selecionada ou atual da memória.

Sintaxe

`removeassess` [*id*]

id: Literal opcional. Identifica o ID de avaliação. Se nenhum ID for incluído, o comando aplica-se à avaliação mais recente. Use `listassess` (`la`) para localizar um ID de avaliação.

Exemplo

Para remover uma avaliação, primeiro obtenha o ID de avaliação:

```
AllApplications\Applications>> ListAssess
542: putty (Application, Tue Apr 01 08:38:18 EDT 2008)
180: JavaAny (Application, Tue Apr 01 13:17:33 EDT 2008)
```

Em seguida, remova a avaliação com o ID 180:

```
AllApplications\Applications>> RemoveAssess 180
```

`report` (rpt)

Descrição

Report gera um relatório do AppScan Source do tipo especificado, incluindo relatórios de descobertas e relatórios do AppScan Source. Uma licença válida do AppScan Source for Automation é requerida para uso deste comando.

Os formatos disponíveis de saída de relatórios são HTML, PDF e zip.

Sintaxe

```
report "<report type>" <output format> <output location>
[<ID de avaliação>] [-includeSrcBefore:<n>] [-includeSrcAfter:<n>]
[-includeTrace:<definitive|suspect|coverage>]
```

- `report type`: O nome do relatório, dentro de aspas duplas, a ser gerado. Especifique um dos seguintes elementos:
 - Um relatório de Descobertas:
 - Descobertas por Pacote Configurável
 - Descobertas por API
 - Descobertas por Classificação

- Descobertas
- Atividade do DTS
- Descobertas por Tipo
- Descobertas por CWE
- Descobertas por Arquivo
- Um relatório do AppScan Source:
 - 25 principais do CWE SANS de 2011
 - DISA Application Security and Development STIG V3R10
 - 10 principais do dispositivo móvel OWASP
 - 10 principais do OWASP de 2013
 - PCI Data Security Standard V3.2
 - Perfil de Segurança de Software
- Um relatório customizado, se disponível.

Ao inserir o tipo de relatório, dentro de aspas duplas, insira exatamente conforme especificado na lista acima - por exemplo Provas por Classificação ou Perfil de Segurança do Software.

- output format : Especifique um dos seguintes formatos para esse relatório:
 - html: Gera o relatório como HTML e o exibe on-line.
 - zip: Cria um arquivo ZIP que contém todos os componentes do relatório de HTML
 - Para relatórios em formato PDF, é possível especificar o nível de detalhe:
 - pdf-summary: Contém contagens para cada grupo de relatórios customizados
 - pdf-detailed: Contém contagens para cada API para cada propriedade de vulnerabilidade
 - pdf-comprehensive: Contém tabelas compostas por todas as descobertas para cada API
 - pdf-annotated: Contém todas as descobertas, qualquer nota incluída com as descobertas e fragmentos de código designados
 - output location: O caminho de arquivo para gravar o relatório.
- output location: Especifique o caminho e o nome do arquivo absoluto no qual deseja salvar o relatório.
- assessment id: Opcional. O ID de avaliação, o qual é obtido no comando listassess (1a). Se você omitir o ID de avaliação, o relatório é gerado a partir da varredura mais recente.
- -includeSrcBefore:<n>: Opcional. O número de linhas de código fonte a ser incluídas em um relatório antes de cada descoberta.
- -includeSrcAfter:<n>: Opcional. O número de linhas de código fonte a ser incluídas em um relatório após cada descoberta.
- -includeTrace:<definitive|suspect|coverage>: Opcional. Inclui informações de rastreamento no relatório para descobertas de cobertura definitivas, suspeitas ou de varredura (consulte "Classificações" na página 151 para aprender mais sobre classificações de descobertas). Para incluir informações de rastreamento para mais de uma classificação de descoberta, especifique esta opção diversas vezes. Por exemplo, para incluir informações de rastreamento para descobertas definitivas e suspeitas, especifique -includeTrace:definitive -includeTrace:suspect.

Exemplos

- Solicite um relatório de *Descobertas por API* por escrito para HTML. No relatório, inclua informações de rastreo para descobertas definitivas:

```
AllApplications>> report "Findings by API" html
C:\reports\findings.html -includeTrace:definitive
```

- Solicite um relatório *10 Principais do OWASP* do AppScan Source gravado em PDF com detalhes abrangentes usando a avaliação existente 542:

```
AllApplications>> report "OWASP Top 10 2013" pdf-comprehensive
/reports/webgoat_OWASP_13_comp.pdf 542
```

scan (sc)

Descrição

Varre um aplicativo (ou todos os aplicativos), projeto ou arquivo. Uma licença válida do AppScan Source for Automation é requerida para uso deste comando.

Importante: Se você estiver trabalhando com um projeto AppScan Source que tenha dependências em um ambiente de desenvolvimento (por exemplo, um projeto IBM® MobileFirst Platform), assegure-se de construir o projeto no ambiente de desenvolvimento antes de importá-lo. Após importá-lo, se modificar arquivos nele, assegure-se de recriá-lo no ambiente de desenvolvimento antes de varrer no AppScan Source (se você não fizer isso, as modificações feitas nos arquivos serão ignoradas pelo AppScan Source).

Sintaxe

```
scan [path][config <proj_config>][-name <assessment_name>]
[-scanconfig <scan_configuration_name>]
```

- path: Opcional. O caminho completo e o nome do arquivo (.ozasmt) que a avaliação exportada será salva como.

Nota:

- Se você especificar um diretório válido sem um nome de arquivo, um arquivo de avaliação(.ozasmt) será criado para você basear no nome do aplicativo, nome do projeto e na configuração de varredura usados ao criar a avaliação.
- Se você especificar um diretório válido com um nome de arquivo que não existe, um arquivo de avaliação será criado naquele local usando o nome do arquivo especificado.
- Se você especificar um arquivo que já existe, o arquivo existente será sobrescrito.
- Se você especificar um nome do arquivo(.ozasmt) em um diretório não existente, nenhuma avaliação será salva.
- config <proj_config>: Opcional. Esse argumento é válido apenas para avaliações no nível do projeto. Se seu projeto tiver um arquivo de configuração, especifique-o usando esse argumento.
- -name <assessment_name>: Opcional. Forneça um nome para a avaliação. Esse nome é usado nos produtos do cliente AppScan Source para distinguir as avaliações umas das outras (por exemplo, em AppScan Source for Analysis, o nome apareceria na coluna **Nome** da visualização Minhas Avaliações).
- -scanconfig <scan_configuration_name>: Opcional. Especifique o nome de uma configuração de varredura para usar para a varredura. Se uma configuração de varredura não for especificada, a configuração de varredura padrão será usada para a varredura.

Exemplos

- Para varrer as configurações padrão de projetos em todos os aplicativos:

```
AllApplications>> Scan
```

Os resultados aparecem como:

```
New Scan started
.
.
Preparing for Vulnerability Analysis...
Performing Vulnerability Analysis...
Generating Findings...
Preparing project for scan...
.
.
Scanned Project:
  Total files: 15
  Total findings: 167
  Total lines: 385
  vkloc: 0.44448395412925595
  v-Density: 22.446439683527426
Scanned Application:
  Total files: 15
  Total findings: 167
  Total lines: 385
  vkloc: 0.44448395412925595
  v-Density: 22.446439683527426
Scan completed:
  Total files: 15
  Total findings: 167
  Total lines: 385
  vkloc: 0.44448395412925595
  v-Density: 22.446439683527426
Elapsed Time - 18 Seconds
New Scan started. Please wait...
Assessment complete
-----
Total Call Sites: 75
Total Definitive Security Findings with High Severity: 25
Total Definitive Security Findings with Medium Severity: 37
Total Definitive Security Findings with Low Severity: 9
Total Suspect Security Findings with High Severity: 20
Total Suspect Security Findings with Medium Severity: 80
Total Suspect Security Findings with Low Severity: 60
Total Scan Coverage Findings with High Severity: 50
Total Scan Coverage Findings with Medium Severity: 33
Total Scan Coverage Findings with Low Severity: 17
Total Lines: 3000
...
```

- Para varrer a configuração de depuração de Prj1:

```
AllApplications\Prj1>> SC config debug
```

script (scr)

Descrição

Executar um script de comandos.

O script pode conter qualquer comando válido da AppScan Source command line interface (CLI) com argumentos. O comando record (rc) pode gerar o arquivo de script.

Scripts podem referenciar variáveis ambientais, usando a notação `${VAR}` (em que VAR é o nome de uma variável ambiental).

Quando o comando `script` é passado para `AppScanSrcCli` a partir da linha de comandos, ele sai automaticamente do `AppScanSrcCli` no final do script.

Sintaxe

```
script script_file
```

`script_file`: Caminho completo e nome do arquivo do arquivo de script que deseja executar.

Comentários em Scripts

Uma linha precedida por um sinal de número (#) é um comentário. A CLI ignora os comentários quando executado interativamente ou a partir de um arquivo de script.

O seguinte script de exemplo contém comentários:

```
login localhost User 123456
#navegar para minha área de trabalho
cd MyApplication
#varrer meu aplicativo
varredura
logout
```

Exemplos

Um usuário que normalmente efetua login e usa o CLI interativamente pode desejar registrar um script para ser executado repetidamente enquanto é executado a partir do console. É possível gravar um script de uma sessão inteira usando o comando `record (rc)` e executá-lo repetidamente ou em intervalos programados. Use um script para automatizar uma sessão inteira ao permitir que um único arquivo em lote ou script para efetue login, crie projetos, faça varredura, copie arquivos e assim por diante.

- Para executar um script, denominado `myscript.txt`:
AllApplications>> Script c:\myscript.txt
- O seguinte script de exemplo efetua login, ativa a criação de log, cria um aplicativo, importa todos os projetos ppf sob um determinado diretório, executa uma avaliação e sai:

```
log on C:\mylogfile.log
new MyApplication C:\myAppFolder
cd MyApplication
im c:\Projects\*.ppf
varredura
logout
```

Observe que `logout` não é necessário se o script for sempre executado a partir da linha de comandos `AppScanSrcCli`.

setaseinfo (sase)

Se o AppScan Enterprise Server tiver sido instalado com a opção AppScan Enterprise Console, será possível publicar avaliações nele. O Enterprise Console oferece uma variedade de ferramentas para trabalhar com suas avaliações, tais como recursos de relatórios, gerenciamento de problema, análise de tendência e painéis.

Descrição

Especifique configurações do Enterprise Console.

Se você pretende publicar avaliações do AppScan Source para o Enterprise Console, use o comando `setaseinfo` para especificar configurações de conexão do Enterprise Console.

Sintaxe

```
setaseinfo --userid|-u <nome de usuário>  
--password|-p <senha>  
--url|-l <url>
```

- `userid`: requerido.
 - Se o seu método de autenticação do AppScan Enterprise Server for ID do usuário e senha, insira seu ID do usuário.
 - Se o seu AppScan Enterprise Server estiver configurado para usar autenticação do Windows, insira o domínio e o nome do usuário usado para conectar ao Enterprise Console (separe o domínio e o nome do usuário com \ - por exemplo, `my_domain\my_username`).
 - Se seu AppScan Enterprise Server estiver configurado com LDAP, insira o nome de usuário que você usa para se conectar ao Enterprise Console.
 - No Windows, se o AppScan Enterprise Server estiver ativado para autenticação Common Access Card (CAC), insira `common_name<cn_certificate>`, em que `common_name` é o seu Nome Comum (CN) do CAC e `cn_certificate` é o CN do emissor do certificado. Se o seu `common_name` ou `cn_certificate` contiver espaços, coloque o valor entre aspas ("`common_name<cn_certificate>`").

No mínimo, você deve ser um usuário do QuickScan. Se estiver conectado a um AppScan Enterprise Server anterior à Versão 9.0.3, você deverá ter sua própria pasta de usuário no Enterprise Server.

- `password`: Obrigatório se o método de autenticação do AppScan Enterprise Server for ID do usuário e senha. Especifique a senha para seu ID de usuário do Enterprise Server.

Nota: Essa opção não será usada quando o AppScan Enterprise Server estiver ativado para autenticação de CAC

- `url`: Especifique a URL para a instância do AppScan Enterprise Server. O formato dessa URL é `http(s)://<hostname>:<port>/ase`, em que `<hostname>` é o nome da máquina na qual o AppScan Enterprise Server foi instalado e `<port>` é a porta na qual o servidor está em execução. Um exemplo dessa URL é `https://myhost.mydomain.ibm.com:9443/ase`.

Nota: Esse argumento será opcional se a URL do Enterprise Console já tiver sido configurada.

Nota:

- Você deve ter assinado o AppScan Source command line interface (CLI) com a permissões Gerenciar Configurações do AppScan Enterprise para poder configurar a `url`. Para obter informações sobre contas de usuários e permissões, consulte a seção *Administrando AppScan Source* do *IBM Security AppScan Source Installation and Administration Guide*.
- O ID do usuário e senha são armazenados na máquina que está executando o cliente AppScan Source (por exemplo, AppScan Source for Analysis) - enquanto a `url` é armazenada no Enterprise Server (que pode estar localizado em uma

máquina remota). Não é possível acessar as informações userid e password da máquina remota (por exemplo, emitindo o comando getaseinfo a partir dela).

Exemplo

```
AllApplications>> setaseinfo --userid my_username --password my_password
--url https://my_aseserver/ase
AppScan Enterprise Integration Information Configured.
AllApplications>>
```

setcurrentobject (set, cd)

Descrição

Use setcurrentobject para navegar na árvore de objetos.

Este comando configura o objeto selecionado atualmente na árvore de objetos. O objeto deve ser um filho válido do objeto atual. Para visualizar os filhos do objeto atual, use o comando list (ls, dir). Selecione o objeto por nome ou ID.

Sintaxe

setcurrentobject {name|id object id}

- *name*: Nome do filho a ser configurado. Necessário ao navegar na árvore por nome do objeto.
- *id*: Literal. Selecionar por ID do objeto.
- *object id*: ID Numérico do filho a ser configurado. Necessário ao selecionar por ID.

Exemplos

- Para navegar a partir da raiz até o Aplicativo cujo ID é igual a 1:

```
AllApplications>> SetCurrentObject applications
AllApplications>> CD id 1
AllApplications\Applications>>
```

- Para recuar um nível:

```
AllApplications\Applications>> CD ..
```

- Para retornar para a parte superior (raiz) da Árvore de Objetos:

```
AllApplications\Applications>> Set /
ou
AllApplications\Applications>> Set \
```

setvar (sv)

Descrição

Cria uma nova variável ou modifica uma variável existente.

Sintaxe

setvar <nome> <caminho>

- *nome*: O nome da variável. Se a variável já existir, então a AppScan Source command line interface (CLI) atualiza o caminho. Se a variável não existir, a CLI cria a variável com o valor do caminho nomeado.
- *path*: Um caminho existente para a variável. Se o caminho não existir, a variável não será criada.

Exemplos

- Para criar a variável AppScan Source Myvar com o caminho C:\Myapps:

```
AllApplications>> SetVar Myvar C:\Myapp
AllApplications>> Criada nova variável '%Myvar%'
```
- Para criar uma variável AppScan Source nomeada SRC_ROOT com base no valor da variável de ambiente do sistema SRC_ROOT:

```
# login
login localhost admin
# criar a variável SRC_ROOT com o valor
# da variável de ambiente SRC_ROOT
setvar SRC_ROOT ${SRC_ROOT}
```

unregister (unreg)

Descrição

Este comando é usado para cancelar o registro de um aplicativo ou projeto do nó atual registrado anteriormente.

Logo que o registro é cancelado, o aplicativo ou projeto não pode ser publicado. Para obter mais informações, consulte o comandoregister (reg).

Sintaxe

unregister

Exemplo

```
AllApplications\Myapp>> Unregister
AllApplications\Myapp>> 'Myapp' teve o registro cancelado com êxito.
```

Executando Avaliações Automatizadas

O AppScan Source command line interface (CLI) permite importar automaticamente um arquivo de projeto do AppScan Source (.ppf) e varrer o código fonte. A partir da linha de comandos, você pode executar um script, como a seguinte amostra, Run_Assessments.txt.

```
AppScanSrcCli scr c:\<dir_instal>\bin\Run_Assessments.txt
```

Run_Assessments.txt de Amostra

```
# Login.
Login <nome_do_host> <nome_de_usuario> <senha>
# Ativar a criação de log.
log on c:\myLogFile.log
# Criar um novo aplicativo denominado "testit."
new testit c:\AppTest
# Navegar para o aplicativo recém criado.
cd testit
# Importar o Arquivos de projeto (.ppfs) sob c:\projects\joans.
im c:\projects\joans\*.ppf
# Atualizar o Projeto.
refresh

# Executar uma avaliação.
varredura
# Registrar a avaliação
register
```

```
# Publicar a avaliação
publishassess
# Efetuar logout e terminar a sessão de CLI.
quit
```

Buffer

Efetuando Login em 'c:\mylogfile.log'...

```
AllApplications>> new testit c:\AppTest
AllApplications>> cd testit
AllApplications\testit>> import c:\TestApps\testproj\*.ppf
AllApplications\testit>> refresh
AllApplications\testit>> ls
```

214: testproj (Project [local])

```
AllApplications\testit>> la
```

testit não possui nenhuma avaliação atual.

```
scan
Nova Varredura iniciada às 15:41:55
Varrendo Projeto testproj (1 de 1)
Preparing project for scan...
```

```
.
.
.
```

```
Procurando Arquivo C:\TestApps\testproj\src\se\bluefish\blueblog\metarepository\Meta
Category.java (21 de 33)
Procurando Arquivo C:\TestApps\testproj\src\se\bluefish\blueblog\metarepository\Meta
Repository.java (22 de 33)
```

```
-----
Total de Sites de Chamada: 348
Total Definitive Security Findings with High Severity: 5
Total Definitive Security Findings with Medium Severity: 1
Total Definitive Security Findings with Low Severity: 4
Total Suspect Security Findings with High Severity: 0
Total Suspect Security Findings with Medium Severity: 8
Total Suspect Security Findings with Low Severity: 0
Total Scan Coverage Findings with High Severity: 16
Total Scan Coverage Findings with Medium Severity: 27
Total Scan Coverage Findings with Low Severity: 16
Total de Linhas: 7386
Max V-Density: 732.2772813430815
Max V/kloc: 10.42512862171676
V-Density: 732.2772813430815
V/kloc: 10.42512862171676
```

```
AllApplications\testit>> register
AllApplications\testit>> 'testit' registrado com êxito.
AllApplications\testit>> pa
```

Assessment Successfully Published.

```
AllApplications\testit>> la
```

```
AllApplications\testit>> 27001: testit (Application, Fri Mar 14 15:41:55 EDT 2008)
```

```
AllApplications\testit>>
```

Capítulo 3. A Ferramenta de Construção Ounce/Ant

Esta seção descreve como usar o Ounce/Ant, um utilitário de construção do AppScan Source que integra o AppScan Source e o Apache Ant. Integrar Ounce/Ant com seu ambiente Ant o ajuda a automatizar construções e avaliações de código.

O Ounce/Ant é uma ferramenta de construção do AppScan Source que se integra em seu ambiente de construção do Apache Ant. Em vez de criar e configurar os arquivos de projeto e aplicativo do AppScan Source manualmente no AppScan Source for Analysis, o Ounce/Ant gera automaticamente projetos e aplicativos do AppScan Source que contêm todas as informações necessárias, como arquivos de origem e caminho de classe. O AppScan Source pode então varrer os arquivos de projeto e aplicativo.

Apache Ant Versão 1.7 ou posterior é necessário para uso com Ounce/Ant.

Integração de Ounce/Ant e Apache/Ant

As etapas descritas nesse tópico de tarefa são necessárias para integrar Ounce/Ant no ambiente de construção do Apache/Ant.

Procedimento

1. Copie o arquivo `ounceant.jar` e, opcionalmente, `ant-contrib-1.0b3.jar` para o diretório `ant lib`.

A instalação do AppScan Source coloca `ounceant.jar` e `ant-contrib-1.0b3.jar` no `<install_dir>\lib` (em que `<install_dir>` é o local da instalação do AppScan Source).

2. Opcional: Substitua a propriedade `build.compiler`.

Consulte “Criando Projetos” na página 63 para obter informações adicionais sobre como substituir `build.compiler`.)

Substitua essa propriedade usando um dos seguintes métodos:

- Use uma tag de propriedade no arquivo `build.xml`.

```
<property name="build.compiler"
value="com.ouncelabs.ounceant.OunceCompilerAdapter"/>
```
- Especifique o valor para `build.compiler` na linha de comandos ao chamar Ant usando a opção `-D`.

```
ant -Dbuild.compiler=
com.ouncelabs.ounceant.OunceCompilerAdapter
```
- Inclua o valor para `build.compiler` em um arquivo de texto e instrua Ant para carregar as propriedades nesse arquivo usando a opção `properties` conforme descrito na documentação do Ant.

3. Crie `taskdefs`.

Para usar tarefas do Ounce/Ant, você deve fazer referência a `ounceant.jar` em uma tarefa `taskdef`. Por exemplo,

```
<taskdef resource="com/ouncelabs/ounceant/task/ounce.xml"
classpath="ounceant.jar"/>
```

Para usar a tarefa `var`, `ant-contrib-1.0b3.jar` deve fazer referência à tarefa `taskdef` como segue:

```
<taskdef resource="net/sf/antcontrib/antlib.xml"/>
```

Propriedades do Ounce/Ant

A tabela nesse tópico descreve propriedades do Ounce/Ant opcionais.

Propriedade	Valores válidos / Valores padrão	Descrição
ounce.default.configuration_name	<nome_de_config>	Nome de configuração criado no projeto AppScan Source. Se não configurado, usa a Configuração 1.
ounce.build.compiler	nome do compilador	O nome do executável compilador a ser usado quando a propriedade build.compiler do Ant estiver configurada para usar OunceCompilerAdapter. Se não configurado, usa javac.
ounce.project_name	Nome	Nome do projeto. Se não configurado, o Ounce/Ant gera um nome. Consulte "Nomeando Projetos" na página 65 para obter informações adicionais.
ounce.project_dir (Necessário para configurar ounce.project_name)	<dir_projeto>	Diretório ativo do projeto. Se não configurado, usa o diretório de arquivo de construção atual.
ounce.application_name	<nome_do_aplicativo>	Nome do aplicativo. Consulte "Criando e Nomeando Aplicativos" na página 65.
ounce.application_dir (Necessário para configurar ounce.application_name)	<dir_aplicat>	Diretório ativo do aplicativo. Você deve configurar ounce.application_name e ounce.application_dir para incluir um projeto em um aplicativo.
ounce.projects.store_full_paths (opcional)	true ou false	Caminhos absolutos de armazenamento criados em aplicativos e projetos. Se não configurado, usa caminhos relativos.

Configurando Propriedades

Você pode usar a tarefa var para configurar propriedades quantas vezes quiser. Quando chamado, explicitamente ou implicitamente, ounceCreateProject usa os valores de propriedade atuais. É possível configurar propriedades através de um dos métodos descritos nesse tópico de tarefa.

Procedimento

- Use um atributo para uma tarefa de Ounce/Ant se o atributo existir.
- Use uma tarefa var antes da tarefa adequada do AppScan Source. Propriedades configuradas com var permanecem no escopo apenas para o arquivo no qual elas são configuradas. Para usar a tarefa var, a biblioteca de contribuição do Ant deve estar no diretório lib do Ant e a tarefa taskdef deve referenciá-lo.

```
<taskdef resource="net/sf/antcontrib/antlib.xml"/>
```
- Use -D na linha de comandos do Ant.
- Coloque a propriedade no arquivo de propriedades de construção (normalmente chamado de build.properties).

Criando Projetos

A tarefa ounceCreateProject gera projetos do AppScan Source.

Use ounceCreateProject explicitamente ou implicitamente:

- Explicitamente chame ounceCreateProject em um arquivo de construção de Ant.
- Implicitamente chame ounceCreateProject sempre que a tarefa javac for chamada através de substituição da propriedade build.compiler.

Projetos podem ser agrupados em aplicativos. É possível usar o atributo de aplicativo opcional ou as propriedades de Ounce/Ant apropriadas para colocar o projeto em um aplicativo.

Projetos também podem ser ampliados usando Ounce/Ant. Atributos de projeto de diversos projetos podem ser mesclados em um projeto do AppScan Source. Se mais de um javac ou ounceCreateProject for ampliar o mesmo projeto, mantenha o mesmo nome e diretório.

Nota: Se você substituir build.compiler para criar projetos, execute ant clean primeiro.

Consulte “Exemplo da Tarefa ounceCreateProject” na página 64 para obter o uso de propriedades e seus atributos aninhados.

ounceCreateProject

ounceCreateProject aceita os seguintes parâmetros e elementos aninhados:

Parâmetros especificados como atributos	Descrição
Nome	Nome para o projeto.
workingDir	Diretório ativo para o projeto.
classpath	Caminho da classe para o projeto.
sourcepath	Caminho de origem para as dependências de origem do projeto. Esses arquivos não são varridos para vulnerabilidades.
jdkName	Nome JDK do AppScan Source a ser usado ao varrer o projeto. (Deve ser criado no AppScan Source for Analysis)
appName	Aplicativo que contém esse projeto (opcional).
appDir	Diretório de aplicativo (opcional)

Parâmetros especificados como elementos aninhados	Descrição
ounceSourceRoot	Especifique as raízes de origem do projeto
ounceWeb	Especifique o conteúdo da Web do projeto
ounceExclude	Permite a exclusão de determinados arquivos a partir do conjunto de arquivos especificado no elemento-pai.

Exemplo da Tarefa ounceCreateProject

```
<ounceCreateProject
  name="myProjectName"
  workingDir="{sandbox}/myProject"
  classpath="{my.class.path}"
  sourcepath="{my.source.path}"
  jdkName="jdk15"
  appName="myApp"
  appDir="{sandbox}>
<ounceSourceRoot dir="src"/>
<ounceExclude dir="src/test"/>
<ounceExclude file="src/mydir/Bad.java"/>
<ounceSourceRoot dir="src2"/>
<ounceSourceRoot file="src3/mydir.java"/>
<ounceWeb webContextRoot="web/myProject.war"/>
<ounceExclude dir="web/test"/>
<ounceExclude file="web/partial.jsp"/>
</ounceCreateProject>
```

ounceSourceRoot

ounceSourceRoot aceita os seguintes parâmetros especificados como atributos:

Parâmetros Especificados como Atributos	Descrição
dir	Caminho para uma raiz de origem válida de Java™
file	Caminho para um arquivo individual

ounceWeb

ounceWeb aceita os seguintes parâmetros especificados como atributos.

Parâmetros Especificados como Atributos	Descrição
webContextRoot	Caminho para uma raiz de contexto da Web ou um arquivo war.

ounceExclude

ounceExclude aceita os seguintes parâmetros especificados como atributos:

Parâmetros Especificados como Atributos	Descrição
dir	Caminho para um diretório a ser excluído

Parâmetros Especificados como Atributos	Descrição
file	Caminho para um arquivo individual a ser excluído

Nomeando Projetos

É possível nomear um projeto usando a convenção de nomenclatura padrão ou criando um nome de projeto manualmente.

Nomenclatura de Projeto Padrão

O esquema de nomenclatura padrão do Ounce/Ant permite a geração de nomes de projeto exclusivos.

<diretório>_<nome_de_destino>[_<número>]

- <diretório>: Nome do diretório de projeto (não inclui caminho completo)
- <nome_de_destino>: Nome de destino atual
- <número>: Um número inteiro que começa em 1 e é incrementado conforme necessário para garantir que seja exclusivo. Observe que <número> aparece apenas estiver houver um conflito.

Exemplo

Com os seguintes parâmetros, o nome do projeto torna-se test_compile. Se dois projetos forem criados no destino de compilação, o segundo nome do projeto torna-se test_compile_1.

```
<diretório_de_trabalho> = C:\mydir\test
<diretório> = test
<nome_de_destino> = compile
```

Nomeando Projetos Manualmente

Se você criar um projeto com ounceCreateProject, pode usar o atributo name para especificar um nome do projeto. Se o atributo name não for especificado ou você criar projetos usando javac, a tarefa do AppScan Source verá a propriedade ounce.project_name e, se ela estiver configurada, usará o valor como o nome do projeto. Portanto, a propriedade ounce.project_name pode ser configurada usando var antes de cada chamada para chamar a tarefa javac, fornecendo um nome exclusivo para cada projeto criado.

Criando e Nomeando Aplicativos

Ounce/Ant pode criar aplicativos do AppScan Source durante a operação e agrupar os projetos criados do AppScan Source nos aplicativos.

Você cria esses aplicativos com duas propriedades Ant:

- ounce.application_name
- ounce.application_dir

Importante: Para criar o aplicativo, você deve configurar ambas as propriedades ou configurar os atributos appName e AppDir da tarefa ounceCreateProject.

Se essas propriedades forem configuradas durante a criação do projeto, o projeto é incluído em um aplicativo com o nome especificado por ounce.application_name e um diretório ativo especificado por ounce.application_dir.

Você pode configurar essas propriedades diversas vezes usando a tarefa `var` para agrupar projetos nos aplicativos que deseja. Um nome do aplicativo apropriado indica a combinação de projetos em um único aplicativo.

Dica: Se você usar `ounceCreateProject`, também pode usar os atributos `appName` e `appDir` para especificar o aplicativo no qual será incluído o projeto.

Integração de Construção

A tarefa `ounceCli` permite acesso à AppScan Source command line interface (CLI) para varredura durante uma construção. A tarefa `ounceCli` pode chamar a `cli` do Ant.

`ounceCli` requer os seguintes parâmetros especificados como atributos:

- `dir`: Local do diretório de instalação do AppScan Source
- `script`: Local do arquivo de script da `cli` para execução
- `output`: Local do arquivo de saída da `cli`

Exemplo

```
<ounceCli
dir="${installDir}"
script="${scripts}/cli_script.txt"
output="log.txt"/>
```

Capítulo 4. API de Acesso a Dados do AppScan Source

A API de Acesso a Dados fornece acesso aos resultados de avaliação gerados pelo AppScan Source, incluindo descobertas e detalhes de descoberta. Ela também fornece acesso a métricas de avaliação, como data e hora de análise, linhas de código, densidade V e número de descobertas.

A API de Acesso a Dados está incluída com a instalação destes componentes do AppScan Source:

- AppScan Source for Analysis
- AppScan Source command line interface (CLI)

A API Data Access API é instalada como <install_dir>\sdk\ouncesdk.jar no Windows e <install_dir>/sdk/ouncesdk.jar no Linux (em que <install_dir> é o local da instalação do AppScan Source).

A API de Acesso a Dados requer JDK 1.5 ou posterior.

No Windows: ao executar um programa que usa o SDK, especifique esses argumentos da Java Virtual Machine (JVM) na linha de comandos:

```
java -classpath
<install_dir>\lib\avalon-framework-4.1.5.jar;
<install_dir>\lib\commons-lang3-3.3.2.jar;
<install_dir>\lib\icu4j-52_1.jar;
<install_dir>\lib\jacob.jar;
<install_dir>\lib\log4j-1.2.8.jar;
<install_dir>\lib\logkit-1.2.jar;
<install_dir>\sdk\ouncesdk.jar;
<install_dir>\lib\xml-apis.jar;
<install_dir>\lib\saxon9.jar
... com.company.product.ClassName
```

No Linux: ao executar um programa que usa o SDK, especifique esses argumentos da Java Virtual Machine (JVM) na linha de comandos:

```
java -classpath
<install_dir>/lib/avalon-framework-4.1.5.jar:
<install_dir>/lib/commons-lang3-3.3.2.jar:
<install_dir>/lib/icu4j-52_1.jar:
<install_dir>/lib/jacob.jar:
<install_dir>/lib/log4j-1.2.8.jar:
<install_dir>/lib/logkit-1.2.jar:
<install_dir>/sdk/ouncesdk.jar:
<install_dir>/lib/xml-apis.jar:
<install_dir>/lib/saxon9.jar
... com.company.product.ClassName
```

Modelo de Objeto de API de Acesso a Dados

Diagrama 1 - Diagrama da UML (Linguagem de Modelagem Unificada) detalhando o modelo do objeto de avaliações

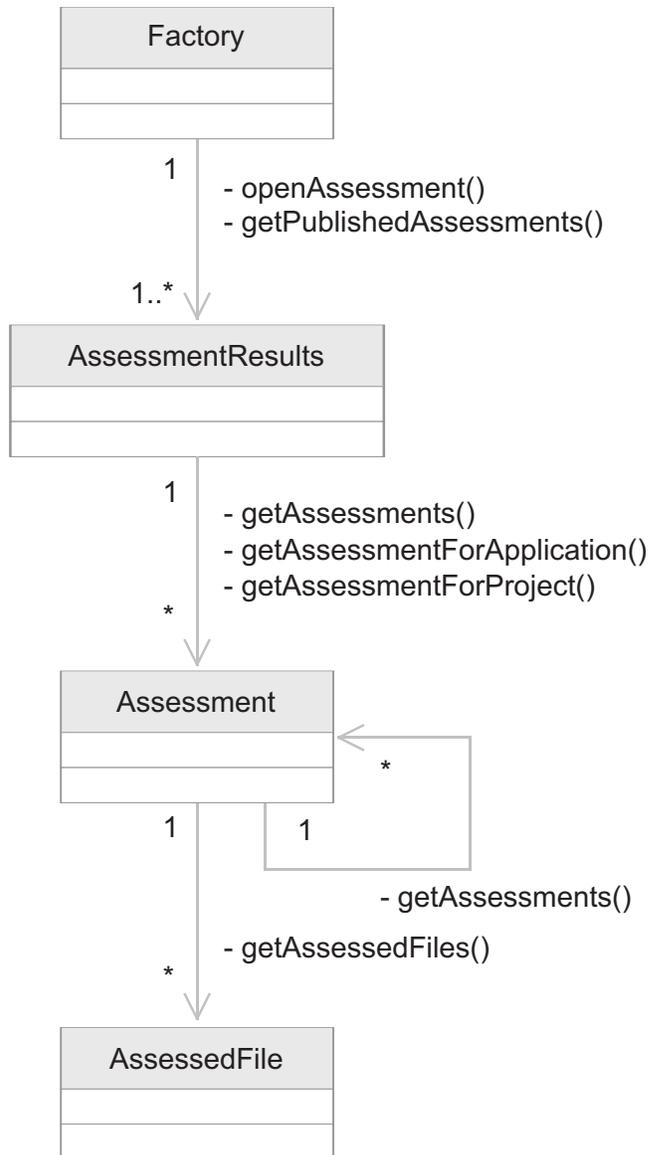


Diagrama 2 - Diagrama da UML (Linguagem de Modelagem Unificada) Detalhando o Modelo do Objeto de Descobertas

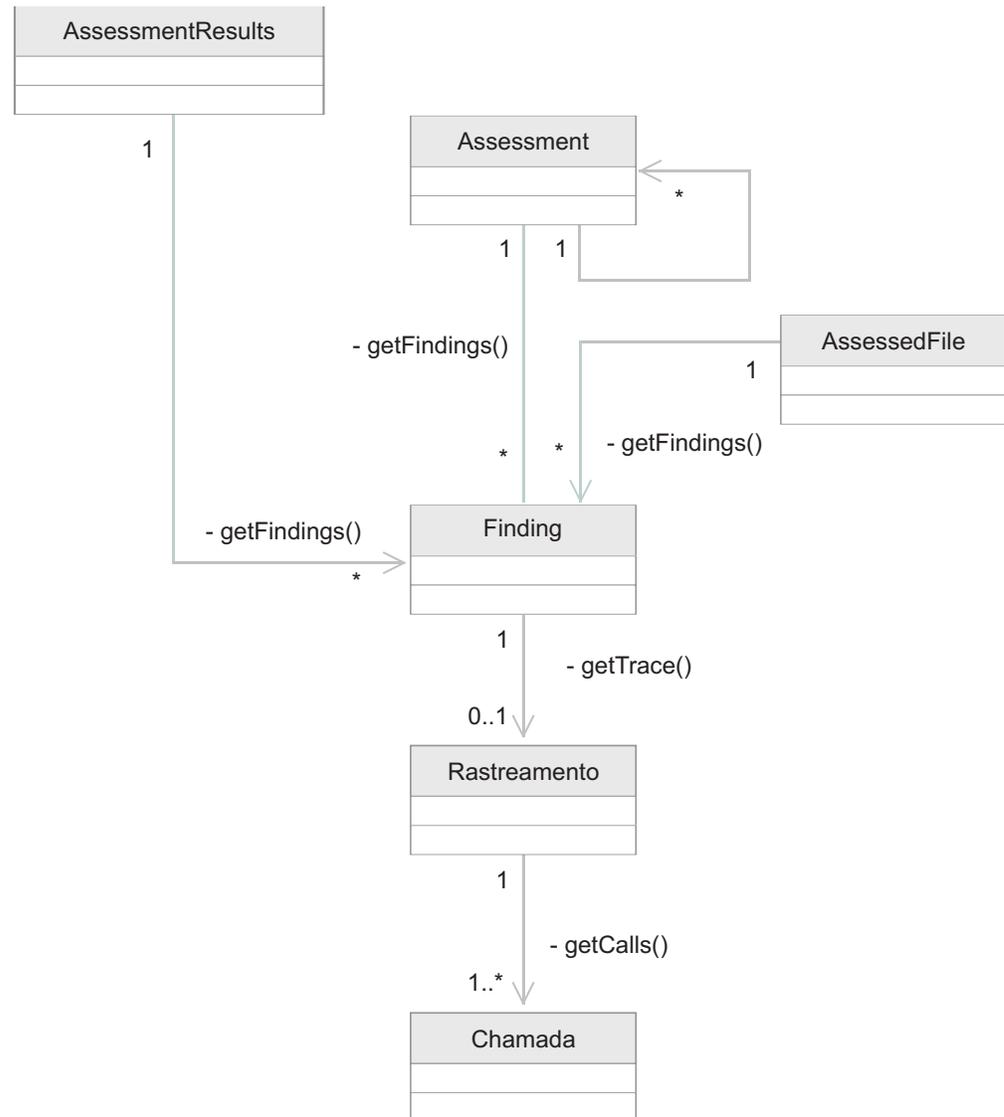


Diagrama 1 - Diagrama da UML (Linguagem de Modelagem Unificada) detalhando o modelo do objeto de avaliações

Na parte superior do diagrama encontra-se a classe Factory. A classe Factory é mostrada como estando relacionada à classe AssessmentResults por meio de dois métodos na classe Factory: `openAssessment()` e `getPublishedAssessments()`. O relacionamento entre essas duas classes é uma classe Factory para um ou mais classes AssessmentResults.

Abaixo da classe Factory, o diagrama representa a classe AssessmentResults. A classe AssessmentResults é mostrada como estando relacionada à classe Assessment por meio de três métodos: `getAssessments()`, `getAssessmentForApplication()` e `getAssessmentForProject()`. O relacionamento entre essas duas classes é uma classe AssessmentResults para zero ou mais classes Assessment.

Abaixo da classe `AssessmentResults`, o diagrama representa a classe `Assessment`. Uma classe `Assessment` é mostrada como estando relacionada a zero ou mais classes `AssessedFile`, por meio do método `getAssessedFiles()`. A classe `Assessment` também representa um relacionamento com ela própria por meio do método `getAssessments()`. Esse relacionamento é mostrado como uma classe `Assessment` para zero ou mais classes filhas `Assessment`.

Abaixo da classe `Assessment`, o diagrama representa a classe `AssessedFile`. Uma classe `Assessment` está relacionada a zero ou mais classes `AssessedFile` por meio do método `getAssessedFiles()`.

Diagrama 2 - Diagrama da UML (Linguagem de Modelagem Unificada) Detalhando o Modelo do Objeto de Descobertas

Na parte superior do diagrama encontram-se três classes: `AssessmentResults`, `Assessment` e `AssessFile`. Cada uma destas três classes está representada como estando relacionada à classe `Finding` por meio do método `getFindings()`. O relacionamento é mostrado como um para zero ou mais classes `Finding`. A classe `Assessment` também representa um relacionamento com ela própria de uma classe `Assessment` para zero ou mais classes filhas `Assessment`.

Abaixo da classe `Finding`, o diagrama representa a classe `Trace`. Uma classe `Finding` está relacionada a zero ou mais classes `Trace` por meio do método `getTrace()`.

Abaixo da classe `Trace`, o diagrama representa a classe `Call`. Uma classe `Trace` está relacionada a uma ou mais classes `Call` por meio do método `getCalls()`.

Usando a API de Acesso a Dados

É possível localizar exemplos de vários cenários da API de acesso a dados nos arquivos `SamplePublished.java` e `SampleSdk.java` incluídos em `<install_dir>\sdk\sample\com\ouncelabs\sdk\sample` (em que `<install_dir>` é o local da instalação do AppScan Source).

Classes e Métodos de API de Acesso a Dados

Esta seção descreve em detalhes as classes e métodos de Acesso a Dados do AppScan Source. As classes são:

- `com.ouncelabs.sdk.Factory`
- `com.ouncelabs.sdk.assessment.AssessedFile`
- `com.ouncelabs.sdk.assessment.Assessment`
- `com.ouncelabs.sdk.assessment.AssessmentDiff`
- `com.ouncelabs.sdk.assessment.AssessmentResults`
- `com.ouncelabs.sdk.assessment.Finding`
- `com.ouncelabs.sdk.assessment.Trace`
- `com.ouncelabs.sdk.assessment.Call`
- `com.ouncelabs.sdk.assessment.SeverityType`
- `com.ouncelabs.sdk.assessment.ClassificationType`
- `com.ouncelabs.sdk.assessment.AssessmentFilter`
- `com.ouncelabs.sdk.assessment.OnceException`

AssessedFile

A classe com.uncelabs.sdk.assessment.AssessedFile representa uma avaliação de um arquivo individual. Ela fornece acesso a dados de avaliação para um arquivo, como suas descobertas e estatísticas.

AssessedFile.getFindings

Assinatura

```
public Finding[] getFindings()
```

Descrição

Fornece acesso a todas as descobertas para esse AssessedFile.

Retornos

Uma matriz de objetos de descoberta para esse arquivo avaliado.

Emite

OnceException, se a API de Acesso a Dados não puder recuperar as descobertas.

AssessedFile.getStats

Assinatura

```
public AssessmentStats getStats()
```

Descrição

Fornece acesso a estatísticas com rolagem para esse arquivo, como o número total de descobertas, o total de linhas, e assim por diante.

Retornos

Objeto AssessmentStats contendo as estatísticas para essa avaliação.

Emite

OnceException, se a API de Acesso a Dados não puder recuperar as estatísticas.

AssessedFile.getFilename

Assinatura

```
public String getFilename()
```

Descrição

Fornece acesso ao caminho absoluto para o arquivo representado por esse AssessedFile.

Retornos

Cadeia contendo o caminho absoluto para o arquivo representado por esse AssessedFile.

Assessment

A classe com `ouncelabs.sdk.assessment.Assessment` representa uma avaliação de um aplicativo ou um projeto.

Assessment.getFindings

Assinatura

```
public Finding[] getFindings()
```

Descrição

Fornece acesso a todas as descobertas para essa Avaliação.

Retornos

Uma matriz de objetos de descoberta para essa avaliação.

Assessment.getStats

Assinatura

```
public AssessmentStats getStats()
```

Descrição

Uma lista completa de estatísticas incluindo: o número total de arquivos varridos, o número total de vulnerabilidades encontradas, tempo de varredura e densidade de vulnerabilidade.

Retornos

Objeto `AssessmentStats` contendo as estatísticas para essa avaliação

Assessment.getAssessments

Assinatura

```
public Assessment[] getAssessments()
```

Descrição

Retorna os objetos-filhos da `Assessment` para essa avaliação. Como avaliações de projeto não possuem filhos de avaliação, objetos-filhos de avaliação são retornados apenas para objetos de avaliação de aplicativo.

Retornos

- Para avaliações de aplicativo, retorna uma matriz de objetos `Assessment`, uma para cada projeto varrido no aplicativo.
- Para avaliações de projeto, retorna uma matriz vazia.

Emite

`OnceException`, se a API de Acesso a Dados não puder recuperar as avaliações.

Assessment.GetFiles

Assinatura

```
public AssessedFile[] getFiles()
```

Descrição

Fornece acesso a cada arquivo que foi parte dessa avaliação.

Retornos

Uma matriz de objetos `AssessedFile`. Cada objeto representa um arquivo que foi parte dessa avaliação.

Emit

`OnceException`, se a API de Acesso a Dados não puder recuperar os arquivos avaliados.

Assessment.getFileByPath

Assinatura

```
public AssessedFile getFileByPath(filePath)
```

Descrição

Fornece acesso a um arquivo avaliado por caminho.

Retornos

Objeto `AssessmentFile` correspondente ao caminho de arquivo especificado ou nulo se o caminho de arquivo especificado não corresponder a um arquivo avaliado.

Emit

`OnceException`, se a API de Acesso a Dados não puder recuperar os arquivos avaliados.

Assessment.getAssesseeName

Assinatura

```
public String getAssesseeName()
```

Descrição

Fornece o nome do aplicativo ou projeto a partir do qual essa avaliação foi gerada.

Retornos

Nome de um aplicativo ou um projeto.

AssessmentDiff

A classe `com.uncelabs.sdk.assessment.AssessmentDiff` mantém a diferença entre as duas avaliações, fornecendo o delta entre as duas avaliações.

AssessmentDiff.getCommonFindings

Assinatura

```
public Finding[] getCommonFindings()
```

Descrição

Obtenha as descobertas que foram comuns para as duas avaliações.

AssessmentDiff.getLostFindings

Assinatura

```
public Finding[] getLostFindings()
```

Descrição

Obtenha as descobertas que estavam na avaliação mais antiga, mas não na mais recente.

AssessmentDiff.getNewFindings

Assinatura

```
public Finding[] getNewFindings()
```

Descrição

Obtenha as descobertas que estavam na avaliação mais recente, mas não na mais antiga.

AssessmentFilter

Use a classe `com.ouncelabs.sdk.assessment.AssessmentFilter` para especificar critérios de filtragem ao recuperar avaliações publicadas.

AssessmentFilter.AssessmentFilter

Assinatura

```
public AssessmentFilter(int maxResults)
```

Descrição

Construtor que somente especifica o número máximo de avaliações publicadas a recuperar. É possível especificar opções de filtragem adicionais chamando os métodos de configuração na classe (por exemplo, `setUserName()` para filtrar por usuário publicado).

Parâmetros

- `maxResults`: O número máximo de resultados a retornar.

AssessmentFilter.AssessmentFilter

Assinatura

```
public AssessmentFilter(String appName,  
String userName,  
Double dateProximityDuration,  
int DateProximityUnit,  
Long dateRangeStart, Long dateRangeEnd,  
int maxResults)
```

Descrição

Construtor utilizando todas as opções de critérios como argumentos.

Parâmetros

- `appName`: O nome do aplicativo. (Nulo se não filtrado por `appName`.)
- `userName`: O usuário que publicou o aplicativo. (Nulo se não filtrado por nome de usuário.)
- `dateProximityDuration`: Quando correlacionado com `dateProximityUnit`, o número de unidades (dias, semanas, e assim por diante) para filtrar iniciando com a data atual. (Nulo se não filtrado por proximidade de data.)
- `dateProximityUnit`: Quando correlacionada com `dateProximityDuration`, a unidade, como dias, semanas, e assim por diante, pela qual contar. Necessário se `dateProximityDuration` for especificada. Consulte “`DateProximityUnit.value`” na página 81 para obter unidades válidas.
- `dateRangeStart`: O início de um intervalo de data. (Nulo se não filtrado por intervalo de data.)
- `dateRangeEnd`: O final do intervalo de data (nulo se não filtrado por intervalo de data).
- `maxResults`: O número máximo de resultados a retornar.

AssessmentResults

A classe `com.ouncelabs.sdk.assessment.AssessmentResults` representa a avaliação inteira.

AssessmentResults.getFindings

Assinatura

```
public Finding[] getFindings()
```

Descrição

Fornecer acesso a todas as descobertas para esses `AssessmentResults`.

Retornos

Uma matriz de objetos de descoberta para essa avaliação.

Emite

`OnceException`, se a API de Acesso a Dados não puder recuperar as descobertas.

AssessmentResults.getAssessments

Assinatura

```
public Assessment[] getAssessments()
```

Descrição

Fornece acesso aos dados de avaliação para cada aplicativo que foi avaliado como parte desses `AssessmentResults`.

Retornos

Uma matriz de objetos de Avaliação, uma para cada aplicativo que era parte dos `AssessmentResults`.

Emite

`IOException`, se a API de Acesso a Dados não puder recuperar as avaliações.

AssessmentResults.getStats

Assinatura

```
public AssessmentStats getStats()
```

Descrição

Fornece acesso a estatísticas com rolagem para essa avaliação, como o número total de arquivos avaliados, o total de linhas, e assim por diante.

Retornos

Objeto `AssessmentStats` contendo as estatísticas para essa avaliação.

Emite

`IOException`, se a API de Acesso a Dados não puder recuperar as estatísticas de avaliação.

AssessmentResults.getAssessmentForApplication

Assinatura

```
public Assessment[] getAssessmentForApplication(String applicationName)
```

Descrição

Fornece acesso aos dados de avaliação para o aplicativo especificado.

Parâmetros

`applicationName`: O nome do aplicativo.

Retornos

Uma matriz de objetos de Avaliação, uma para cada aplicativo que corresponde ao nome especificado.

Emit

OnceException, se a API de Acesso a Dados não puder recuperar as avaliações.

AssessmentResults.getAssessmentForProject

Assinatura

```
public Assessment[]  
getAssessmentForProject(String projectName, String applicationName)
```

Descrição

Fornecer acesso aos dados de avaliação para o projeto especificado no aplicativo especificado.

Parâmetros

projectName: O nome do projeto.

applicationName: O nome do aplicativo no qual o projeto reside.

Retornos

Uma matriz de objetos de Avaliação, uma para cada projeto que corresponde ao nome especificado.

Emit

OnceException, se a API de Acesso a Dados não puder recuperar as avaliações.

AssessmentResults.getName

Assinatura

```
public String getName()
```

Descrição

Retorna o nome dos AssessmentResults.

Retornos

Cadeia contendo o nome dos AssessmentResults.

AssessmentResults.getMessages

Assinatura

```
public Message[] getMessages()
```

Descrição

Fornecer acesso a todas as mensagens de status que apareceram no console ao executar a avaliação no AppScan Source for Analysis.

Retornos

Uma matriz de objetos de Mensagem.

Emit

OnceException, se a API de Acesso a Dados não puder recuperar as mensagens.

Call

A classe com.uncelabs.sdk.assessment.Call representa um nó em um AppScan Source trace.

Call.getCalls

Assinatura

```
public Call[] getCalls()
```

Descrição

Fornecer acesso às chamadas que foram feitas no contexto dessa chamada.

Retornos

Uma matriz de objetos de Chamada, na ordem em que eles foram chamados

Emit

OnceException, se a API de Acesso a Dados não puder recuperar as chamadas.

Call.getFilename

Assinatura

```
public String getFilename()
```

Descrição

Fornecer acesso ao nome do arquivo no qual essa chamada ocorreu.

Retornos

O nome do arquivo para essa chamada.

Call.getLineNumber

Assinatura

```
public int getLineNumber()
```

Descrição

Fornecer acesso ao número da linha no qual a chamada ocorreu.

Retornos

O número da linha dessa chamada.

Call.getColumnNumber

Assinatura

```
public int getColumnNumber()
```

Descrição

Essa chamada fornece acesso ao número da coluna na qual a chamada ocorreu.

Retornos

O número da coluna na qual a chamada foi originada.

Call.getSignature

Assinatura

```
public String getSignature()
```

Descrição

Fornecer acesso à assinatura da API que foi chamada.

Retornos

A assinatura dessa chamada.

Call.getSrcContext

Assinatura

```
public String getSrcContext()
```

Descrição

Fornecer acesso ao contexto de origem da chamada.

Retornos

O contexto de origem dessa chamada.

Call.getMethodName

Assinatura

```
public String getMethodName()
```

Descrição

Fornecer acesso ao nome do método que foi chamado.

Retornos

O nome do método para essa chamada.

Call.getClassName

Assinatura

```
public String getClassName()
```

Descrição

Fornecer acesso ao nome de classe ao qual o método chamado pertence.

Retornos

O nome de classe para esse método de chamada.

Call.getTraceType

Assinatura

```
public TraceType getTraceType()
```

Descrição

Fornece acesso ao tipo de rastreo para essa chamada. O tipo de rastreo, por exemplo, pode especificar que uma chamada é uma origem ou um depósito.

Retornos

Objeto TraceType que representa o tipo de rastreo para a chamada.

ClassificationType

A classe com.uncelabs.sdk.assessment.ClassificationType representa a classificação de uma descoberta.

Membros Estáticos

- ClassificationType.Vulnerability
- ClassificationType.Type1
- ClassificationType.Type2
- ClassificationType.Unknown

ClassificationType.value

Assinatura

```
public int value()
```

Descrição

Fornece acesso ao valor desse ClassificationType. O valor retornado corresponde a um dos valores dos membros estáticos.

Retornos

O valor de classificação para esse objeto ClassificationType.

DateProximityUnit

Quando com.uncelabs.sdk.assessment.DateProximityUnit é correlacionada com dateProximityDuration, a unidade, como dias, semanas, e assim por diante, pela qual contar. Necessário quando dateProximityDuration for especificada. Unidades válidas são descritas nessa seção.

Membros Estáticos

- DateProximityUnit.Hour
- DateProximityUnit.Day
- DateProximityUnit.Week
- DateProximityUnit.Month

- `DateProximityUnit.Year`

DateProximityUnit.value

Assinatura

```
public EnumType value()
```

Descrição

Fornecer acesso ao valor dessa `DateProximityUnit`. O valor retornado corresponde a um dos valores dos membros estáticos.

Retornos

O valor de data para esse objeto `DateProximityUnit`.

Factory

`com.ouncelabs.sdk.Factory` fornece métodos para inicialização, criação de login e avaliações de abertura. Essa classe é o ponto de entrada na API de Acesso a Dados.

Factory.init

Assinatura

```
public void init(String installDir, boolean acceptInvalidSSLCerts)
```

Descrição

Inicializa a API de Acesso a Dados. Você deve chamar `Factory.init` antes de chamar qualquer outro método.

Parâmetros

- `installDir`: Caminho para o diretório de instalação do AppScan Source (consulte “Local de instalação padrão” na página 149).
- `acceptInvalidSSLCerts`: Configure o booleano como `true` se deseja aceitar certificados SSL inválidos (certificados SSL não exigirão validação).

Emit

`OnceException`, se a inicialização falhar. Geralmente, isso ocorre apenas se você especificar o diretório errado.

Factory.shutdown

Assinatura

```
public void shutdown()
```

Descrição

Sai da API de Acesso a Dados. Se `Factory.init` for chamado, você deve chamar `Factory.shutdown` para a API de Acesso a Dados ser encerrada corretamente. Se não for chamado, `OnceScanner.exe` deve continuar a ser executado como um processo não autorizado.

Factory.clearCache

Assinatura

```
public void clearCache()
```

Descrição

Essa chamada limpa o cache para a API de Acesso a Dados. Ela deve ser usada periodicamente quando processar muitos resultados de descobertas no uso de memória excessivo.

Factory.login

Assinatura

```
public void login(String user name, String password, String host name[int :port])
```

Descrição

Efetua login no AppScan Enterprise Server executando como a URL especificada ou no nome do host especificado. Você deve efetuar login antes de poder obter avaliações publicadas no AppScan Source Database.

Parâmetros

- username: O nome de usuário do usuário que está efetuando login.
- password: A senha do usuário que está efetuando login.
- hostname: A URL ou nome do host do AppScan Enterprise Server no qual deseja efetuar login. hostname pode ser um endereço IP ou um nome de domínio.

Se necessário, você pode especificar o número da porta do AppScan Enterprise Server anexando `<port>`. Por exemplo, `public void login("hwall", "shhh2008", "MyHost:2880")`.

Emit

0unceException, se o login falhar.

Factory.openAssessment

Assinatura

```
public AssessmentResults openAssessment(String pathname)
```

Descrição

Abre a avaliação especificada por nome do caminho.

Parâmetros

pathname: O caminho de arquivo para um arquivo de avaliação.

Retornos

O objeto AssessmentResults que representa a avaliação especificada por pathname.

Emit

- `FileNotFoundException`, se o nome do arquivo especificado não existir ou não puder ser aberto.
- `OnceException`, se a API de Acesso a Dados não puder carregar o arquivo de avaliação.

Factory.getPublishedAssessments

Assinatura

```
public AssessmentResults[] getPublishedAssessments(AssessmentFilter filter)
```

Descrição

Fornecer acesso a avaliações publicadas. Utiliza um objeto `AssessmentFilter` que você pode utilizar para especificar o conjunto de avaliações publicadas para recuperar.

Parâmetros

`filter`: Um objeto `AssessmentFilter` que especifica o conjunto de avaliações publicadas para recuperar.

Retornos

Uma matriz de objetos `AssessmentResults` correspondentes ao filtro especificado.

Emit

`OnceException`, se a API de Acesso a Dados não puder recuperar os resultados de avaliação.

Factory.diffAssessments

Assinatura

```
public AssessmentDiff diffAssessments  
(AssessmentResults result1, AssessmentResults result2)
```

Descrição

Localiza as diferenças entre os dois resultados de avaliação.

Emit

`OnceException`.

Finding

A classe `com.uncelabs.sdk.assessment.Finding` representa uma descoberta individual em uma avaliação. Ela fornece acesso a todos os dados associados com uma descoberta, como classificação e gravidade.

Finding.getFilename

Assinatura

```
public String getFilename()
```

Descrição

Fornece acesso ao nome do arquivo no qual essa descoberta foi localizada.

Retornos

Uma cadeia contendo o caminho absoluto para o arquivo no qual essa descoberta foi localizada.

Finding.getLineNumber

Assinatura

```
public int getLineNumber()
```

Descrição

Fornece acesso ao número da linha no qual a descoberta foi localizada.

Retornos

O número da linha para essa descoberta.

Finding.getApiName

Assinatura

```
public String getApiName()
```

Descrição

Fornece acesso ao nome da API para essa descoberta.

Retornos

O nome da API para essa descoberta.

Finding.getCallerName

Assinatura

```
public String getCallerName()
```

Descrição

Fornece acesso à assinatura do método que continha a chamada para a API que essa descoberta representa.

Retornos

A assinatura do responsável pela chamada.

Finding.getClassification

Assinatura

```
public ClassificationType getClassification()
```

Descrição

Fornece acesso à classificação da descoberta.

Retornos

Objeto `ClassificationType` que representa a classificação da descoberta

Finding.getSeverity

Assinatura

```
public SeverityType getSeverity()
```

Descrição

Obtenha o valor atual da gravidade na descoberta.

Retornos

Objeto `SeverityType` que representa a gravidade da descoberta. `SeverityType` está descrito em “`SeverityType.value`” na página 90.

Finding.getVulnerabilityType

Assinatura

```
public String getVulnerabilityType()
```

Descrição

Fornece acesso ao tipo de vulnerabilidade atual da descoberta.

Retornos

O tipo de vulnerabilidade para essa descoberta. Tipos de vulnerabilidade, como Estouro de Buffer ou Injeção de SQL, estão descritos no Base de Conhecimento de Segurança do AppScan Source.

Finding.getOriginalClassification

Assinatura

```
public ClassificationType getOriginalClassification()
```

Descrição

Fornece acesso à classificação original da descoberta. Se a descoberta não foi modificada, `getOriginal` é o mesmo que `getClassification`.

Retornos

Objeto `ClassificationType` que representa a classificação original de uma descoberta em que a classificação foi modificada.

Finding.getOriginalSeverity

Assinatura

```
public SeverityType getOriginalSeverity()
```

Descrição

Obtém o valor original da gravidade da descoberta. Se a descoberta não foi modificada, `getOriginalSeverity` é o mesmo que `getSeverity`.

Retornos

Objeto `SeverityType` que representa a gravidade original para uma descoberta em que a gravidade foi modificada.

Finding.getOriginalVulnerabilityType

Assinatura

```
public String getOriginalVulnerabilityType()
```

Descrição

Fornecer acesso ao tipo de vulnerabilidade original da descoberta. Se a descoberta não foi modificada, `getOriginalVulnerabilityType` é o mesmo que `getVulnerability`.

Retornos

Um objeto `VulnerabilityType` que representa o tipo de vulnerabilidade original para uma descoberta em que a vulnerabilidade foi modificada.

Finding.getModifiedClassification

Assinatura

```
public ClassificationType getModifiedClassification()
```

Descrição

Fornecer acesso à classificação modificada da descoberta.

Retornos

Objeto `ClassificationType` que representa a classificação de uma descoberta em que a classificação foi modificada. (Nulo se não corresponder.)

Finding.getModifiedSeverity

Assinatura

```
public SeverityType getModifiedSeverity()
```

Descrição

Obtém o valor modificado da gravidade. O mesmo que `getSeverity()` se modificado.

Retornos

Objeto `SeverityType` que representa a gravidade para uma descoberta em que a gravidade foi modificada. (Nulo se não corresponder.)

Finding.getModifiedVulnerabilityType

Assinatura

```
public String getModifiedVulnerabilityType()
```

Descrição

Fornece acesso ao tipo de vulnerabilidade modificado da descoberta.

Retornos

Um objeto `VulnerabilityType` que representa o tipo de vulnerabilidade para uma descoberta em que a vulnerabilidade foi modificada. (Nulo se não corresponder.)

Finding.getSrcContext

Assinatura

```
public String getSrcContext()
```

Descrição

Fornece acesso ao contexto de origem para essa descoberta.

Retornos

O contexto de origem para essa descoberta.

Finding.getDefectSubmissionUser

Assinatura

```
public String getDefectSubmissionUser()
```

Descrição

Fornece acesso ao nome do usuário que submeteu essa descoberta por último a um sistema de rastreamento de defeito externo. Observe que é o nome de usuário do sistema de rastreamento de defeitos, que pode ser diferente do nome de usuário do AppScan Source.

Retornos

O usuário submetente a partir da última submissão do sistema de rastreamento de defeito dessa descoberta.

Finding.getDefectSubmissionDate

Assinatura

```
public String getDefectSubmissionDate()
```

Descrição

Fornece acesso à cadeia de datas a partir do momento em que essa descoberta foi submetida pela última vez a um sistema de rastreamento de defeito externo.

Retornos

A cadeia de datas de submissão a partir da última submissão do sistema de rastreamento de defeito dessa descoberta.

Finding.getDefectInfo

Assinatura

```
public String getDefectInfo()
```

Descrição

Fornece acesso ao ID do sistema de rastreamento de defeito externo a partir do momento em que essa descoberta foi submetida pela última vez a um sistema de rastreamento de defeito externo.

Retornos

Uma cadeia com IDs de defeito associados com essa descoberta.

Finding.getProperties

Assinatura

```
public String[] getProperties()
```

Descrição

Essa chamada fornece acesso às propriedades associadas com a descoberta. Essas propriedades fornecem informações sobre esse tipo de vulnerabilidade e sobre essa API da descoberta.

Retornos

Matriz de propriedades:

- `Vulnerability.value`: Tipo de vulnerabilidade relatado nas visualizações de Descobertas, como:
`Vulnerability.BufferOverflow`
`Vulnerability.Injection.SQL`
- `Mechanism.value`: Mecanismo de segurança de API; usado para definir categorias de relatório customizado, como:
`Mechanism.AccessControl`
`Mechanism.Cryptography`
- `Technology.value`: Tecnologia de API; usada para definir categorias de relatório customizado, como:
`Technology.Communications.HTTP`
`Technology.Database`
- `Attribute.value`: Tag de método de API, como:
`Attribute.Deprecated`
`Attribute.Modifier.Protected`

Consulte o Base de Conhecimento de Segurança do AppScan Source para obter informações detalhadas sobre valores de propriedade.

Emit

OnceException, se a API de Acesso a Dados não puder recuperar as propriedades.

Finding.isExcluded

Assinatura

```
public boolean isExcluded()
```

Descrição

Indica se essa descoberta é excluída da avaliação. Descobertas excluídas não contribuem com nenhuma estatística para a avaliação.

Retornos

Se excluídas, true, se não excluídas, false.

Finding.getTrace

Assinatura

```
public Trace getTrace()
```

Descrição

Fornecer acesso ao AppScan Source trace associado a essa descoberta, se houver alguma.

Retornos

Objeto de rastreamento para a descoberta se houver uma, nulo se não houver.

Emit

OnceException, se a API de Acesso a Dados não puder recuperar o rastreamento.

Finding.getNotes

Assinatura

```
public String getNotes()
```

Descrição

Fornecer acesso às notas em uma descoberta.

Retornos

Qualquer nota para a descoberta, nula se não houver nenhuma nota.

Emit

OnceException, se a API de Acesso a Dados não puder recuperar as notas.

Rastreamento

A classe com.uncelabs.sdk.assessment.Trace representa o AppScan Source trace para uma descoberta.

Trace.getCalls

Assinatura

```
public Call[] getCalls()
```

Descrição

Fornece acesso à chamada raiz do AppScan Source trace.

Retornos

Uma matriz de objetos Call. Atualmente sempre contém um único objeto.

Emit

OnceException, se a API de Acesso a Dados não puder recuperar as chamadas.

SeverityType

A classe com.ouncelabs.sdk.assessment.SeverityType representa a gravidade de uma descoberta.

Membros Estáticos

Os tipos de gravidade são:

- SeverityType.High
- SeverityType.Medium
- SeverityType.Low
- SeverityType.Info
- SeverityType.Unknown

SeverityType.value

Assinatura

```
public int value()
```

Descrição

Fornece acesso ao valor desse SeverityType. O valor retornado corresponde a um dos valores dos membros estáticos.

Retornos

O valor de gravidade para esse objeto SeverityType.

OnceException

com.ouncelabs.sdk.assessment.OnceException é a classe de exceção usada pela API de Acesso a Dados. Para encontrar informações adicionais sobre uma exceção, chame getMessage() na exceção.

Capítulo 5. Plug-in Ounce/Maven

Esta seção descreve o plug-in Ounce/Maven, que usa Maven, uma ferramenta de construção do Apache, para integrar o AppScan Source no fluxo de trabalho do Maven.

O plug-in Ounce/Maven permite criar os arquivos de aplicativo e projeto do AppScan Source. Se o AppScan Source for Automation também estiver instalado, use o Ounce/Maven para executar as varreduras de segurança do código fonte e gerar relatórios de segurança abrangentes.

Para obter informações sobre o AppScan Source for Automation, consulte Capítulo 6, “AppScan Source for Automation”, na página 97.

Para obter informações adicionais sobre a família de produtos do AppScan Source, consulte <http://www.ibm.com/software/rational/products/appscan/source/>.

Instalando Ounce/Maven

Antes de Iniciar

Os seguintes são pré-requisitos para instalar e executar Ounce/Maven:

- Apache Maven Versão 2.x ou posterior: Para obter informações sobre implantação e uso do Maven plug-ins, ver o website Apache Maven Project em <http://maven.apache.org/>.
- Para varredura usando o Maven plug-in, você necessita AppScan Source for Automation: Para obter mais informações, consulte Capítulo 6, “AppScan Source for Automation”, na página 97.

Sobre Esta Tarefa

Assim que instalar e configurar Maven, Ounce/Maven é transferido por download na primeira vez em que você referenciá-lo.

A documentação do site de Ounce/Maven inclui descrições dos objetivos Ounce/Maven, seus parâmetros, exemplos, notas de uso e exemplos detalhados.

É possível localizar a documentação do site em <http://mojo.codehaus.org/ounce-maven-plugin>.

Procedimento

1. Se Maven ainda não estiver instalado, instale Maven a partir da <http://maven.apache.org/>. Siga as orientações no Web site de Maven.
2. Execute um dos seguintes:
 - Edite um arquivo pom de Maven para incluir um ou mais objetivos Ounce/Maven, conforme descrito na documentação do site de Ounce/Maven
 - Chame um ou mais objetivos Ounce/Maven a partir da linha de comandos, conforme descrito na documentação do site de Ounce/Maven

Usando Ounce/Maven

O plug-in Ounce/Maven permite usar o Ounce/Maven para criar projetos e aplicativos do AppScan Source, varrer os aplicativos, publicar as avaliações resultantes e gerar relatórios do AppScan Source. Especifique os objetivos e parâmetros Ounce/Maven conforme faria para qualquer outro plug-in Maven.

É possível chamar comandos Ounce/Maven de duas maneiras:

- Usando um arquivo `pom` (construção) do Maven: O arquivo `pom` permite criar arquivos de aplicativo e projeto do AppScan Source como parte de sua construção. Depois desinstalar o Ounce/Maven, você pode modificar um arquivo `pom` do Maven para especificar os objetivos `ounce:application` e `ounce:project-only` conforme necessário para suas tarefas do AppScan Source.
- A partir da linha de comandos: Chame os objetivos `ounce:project`, `ounce:scan` e `ounce:report` a partir da linha de comandos para criar arquivos de projeto do AppScan Source (ou substituir parâmetros do arquivo de projeto a partir do arquivo `pom`), inicie as varreduras do AppScan Source, publique avaliações e gere relatórios do AppScan Source.

Cada um dos objetivos Ounce/Maven inclui uma série de parâmetros.

- Para obter informações sobre objetivos Ounce/Maven, consulte “Objetivos Ounce/Maven” na página 93.
- Para obter informações sobre os parâmetros para cada objetivo, consulte a documentação do site de Ounce/Maven em <http://mojo.codehaus.org/ounce-maven-plugin>.

Cenários de Ounce/Maven

Ounce/Maven permite:

- Criar arquivos de aplicativo e projeto do AppScan Source
- Varrer aplicativos e publicar os resultados (avaliações) no banco de dados do AppScan Source
- Gerar relatórios a partir de resultados de avaliação
- Integrar relatórios do AppScan Source no destino site

Esta seção descreve essas tarefas em resumo. Para obter informações detalhadas sobre conceitos e linguagem específicos do AppScan Source, consulte o *AppScan Source for Analysis User Guide*.

Criação Arquivos de Aplicativo e de Projeto

O AppScan Source usa um modelo de aplicativo e projeto para gerenciar dados, como segue:

- **Aplicativo:** Um aplicativo contém dados de configuração e informações customizáveis, junto com uma lista de projetos nesse aplicativo.
- **Projeto:** Um projeto consiste em um conjunto de arquivos (incluindo código fonte) e suas informações relacionadas (como dados de configuração). Para varrer um projeto, ele deve ser parte de um aplicativo.

O objetivo `ounce:application` cria arquivos (`.paf`) do aplicativo AppScan Source - enquanto os objetivos `ounce:project` e `ounce:project-only` criam arquivos (`.ppf`) de projeto do AppScan Source.

Consulte o *AppScan Source for Analysis User Guide* para obter informações sobre outros formatos de arquivo de aplicativo e projeto.

Aplicativos de Varredura

Uma varredura do AppScan Source analisa o código fonte para vulnerabilidades de segurança. O resultado de uma varredura é uma avaliação, que é um arquivo XML.

Nota: Para obter informações detalhadas sobre as capacidades do AppScan Source, consulte o *IBM Security AppScan Source para Analysis Guia do Usuário*.

Use o objetivo `ounce:scan` da linha de comandos para varrer o aplicativo e seus projetos e, opcionalmente, gerar um relatório a partir da avaliação.

Após uma varredura ser concluída, Ounce/Maven permite:

- Publicar a avaliação para o AppScan Source Database. Isso disponibiliza os resultados da avaliação para outros usuários com acesso ao banco de dados e com privilégios necessários.
- Gerar um relatório.

Relatórios

Para muitos, a saída preferida para o AppScan Source é um relatório detalhado que fornece dados sobre as avaliações. O objetivo `ounce:report` permite gerar esses relatórios a partir de avaliações novas ou anteriores. O objetivo `ounce:report` pode, se necessário, varrer um aplicativo e, em seguida, publicar a avaliação ou gerar relatórios sobre a avaliação. Ao contrário do `ounce:scan`, no entanto, você pode usar `ounce:report` para gerar relatórios para avaliações existentes.

Integrando Relatórios com o Destino do Site

Para integrar `ounce:report` com o destino do site, inclua o plug-in Ounce/Maven na seção de relatório do pom. O objetivo do site executa `ounce:report` e a saída de relatório torna-se parte da documentação do site para o seu aplicativo.

Ao integrar com o destino do site, não é necessário especificar qualquer parâmetro adicional. É possível especificar o tipo de relatório como um dos valores descritos em "Valores de `reportType`" na página 95. Se você não especificar um tipo de relatório, o relatório padrão é Descobertas.

Objetivos Ounce/Maven

O Ounce/Maven fornece os seguintes objetivos para executar as funções do AppScan Source:

- `ounce:application`: gera um arquivo de aplicativo do AppScan Source que contém referências para todos os projetos filhos, conforme definido pelo arquivo pom. Um aplicativo é necessário para varrer (e assim relatar).
- `ounce:project`: Cria um ou mais arquivos de projeto do AppScan Source dependendo do número de projetos-filho do Maven. O objetivo `ounce:project` destina-se a executar a partir da linha de comandos e incorporar a construção de Maven no objetivo.
- `ounce:project-only`: Cria um ou mais arquivos de projeto do AppScan Source dependendo do número de projetos-filho do Maven. O objetivo `ounce:project-only` é fornecido para integrar a criação dos arquivos de projeto do AppScan Source no ciclo de vida de construção do Maven.

- `ounce:scan`: Varre um aplicativo. É possível opcionalmente publicar a avaliação ou gerar um relatório. Execute o objetivo `ounce:scan` a partir da linha de comandos.
- `ounce:report`: gera um relatório do AppScan Source. Se necessário atualizar resultados, execute uma varredura antes de gerar o relatório. Execute o objetivo `ounce:report` a partir da linha de comandos.

Nota:

- Para tornar seus arquivos de aplicativo e projeto móveis, crie variáveis de caminho para mapear caminhos de arquivo para seus locais.
- Para obter exemplos de como usar objetivos Ounce/Maven, consulte a documentação do site de Ounce/Maven em <http://mojo.codehaus.org/ounce-maven-plugin/>.

ounce:application

Descrição

Gera um arquivo de aplicativo do AppScan Source que contém referências para todos os projetos filhos, conforme definido pelo arquivo pom. Um aplicativo é necessário para varrer e relatar. Enquanto `ounce:application` não cria arquivos de projeto do AppScan Source, você pode executar `ounce:project-only` para criar o projeto dentro do mesmo arquivo de construção.

ounce:project

Descrição

Use `ounce:project` a partir da linha de comandos. O objetivo `ounce:project` cria um ou mais arquivos de projeto do AppScan Source dependendo do número de projetos-filho do Maven.

Usando ounce:project

Quando você invocar o objetivo `ounce:project` a partir da linha de comandos, ele constrói automaticamente qualquer dependência necessária. Se nenhum arquivo de aplicativo existir, `ounce:project` o constrói.

Todos os parâmetros para `ounce:project` são opcionais. O nome do projeto é determinado pelo `artifactId` de Maven e você não pode substituí-lo.

Execute Maven com um objetivo `ounce:project` a partir do diretório que contém o arquivo `pom.xml` de nível superior (ou, se apenas um subconjunto da construção for necessário, a partir de um subdiretório que contém um `pom.xml` filho).

ounce:project-only

Descrição

O objetivo `ounce:project-only` é usado a partir de um arquivo pom. O objetivo `ounce:project-only` cria um ou mais arquivos de projeto do AppScan Source dependendo do número de projetos-filho do Maven.

Usando ounce:project-only

Todos os parâmetros para ounce:project-only são opcionais. O nome do projeto é determinado pelo artifactId de Maven e não pode ser substituído.

ounce:scan

Descrição

O objetivo ounce:scan gera uma varredura de um determinado aplicativo, seguido opcionalmente por um relatório na avaliação resultante. Execute ounce:scan a partir da linha de comandos.

Nota: O objetivo ounce:scan não se destina a ser parte do ciclo de vida de construção de Maven.

Usando ounce:scan

Quando você especifica ounce:scan, pode solicitar que Ounce/Maven execute um relatório a partir da avaliação seguindo imediatamente a varredura. Nesse caso, especifique os parâmetros de relatório descritos em “Valores de reportType” e “Valores de reportOutputType” na página 96. Se você especificar o reportType, também deve especificar reportOutputType e reportOutputPath.

Nota: The O objetivo ounce:scan cria ou atualiza arquivos de aplicativo e projeto, se necessário.

ounce:report

Descrição

O objetivo ounce:report gera um relatório a partir de uma avaliação. Se você não especificar uma avaliação existente, ounce:report executa ounce:scan antes de gerar o relatório. Execute ounce:report a partir da linha de comandos.

Especifique os parâmetros de relatório descritos em “Valores de reportType” e “Valores de reportOutputType” na página 96. Se você especificar o reportType, também deve especificar reportOutputType e reportOutputPath.

Valores de reportType

- Um relatório de Descobertas:
 - Descobertas por Pacote Configurável
 - Descobertas por API
 - Descobertas por Classificação
 - Descobertas
 - Atividade do DTS
 - Descobertas por Tipo
 - Descobertas por CWE
 - Descobertas por Arquivo
- Um relatório do AppScan Source:
 - 25 principais do CWE SANS de 2011
 - DISA Application Security and Development STIG V3R10

- 10 principais do dispositivo móvel OWASP
- 10 principais do OWASP de 2013
- PCI Data Security Standard V3.2
- Perfil de Segurança de Software
- Um relatório customizado, se disponível.

Valores de reportOutputType

- Especifique um dos seguintes formatos para esse relatório:
 - html: Gera o relatório como HTML e o exibe on-line.
 - zip: Cria um arquivo ZIP que contém todos os componentes do relatório de HTML.
- Para relatórios em formato PDF, é possível especificar o nível de detalhe:
 - pdf-summary: Contém contagens para cada grupo de relatórios customizados
 - pdf-detailed: Contém contagens para cada API para cada propriedade de vulnerabilidade
 - pdf-comprehensive: Contém tabelas compostas por todas as descobertas para cada API
 - pdf-annotated: Contém todas as descobertas, qualquer nota incluída com as descobertas e fragmentos de código designados
 - pdf-annotated: Gera um relatório anotado como um arquivo PDF.

Capítulo 6. AppScan Source for Automation

O Automation Server (ou `onceautod`) permite automatizar aspectos-chave do fluxo de trabalho do AppScan Source e integrar segurança com ambientes de construção durante o ciclo de vida do desenvolvimento de software (SDLC). O Automation Server permite enfileirar solicitações para varrer e publicar avaliações e gerar relatórios sobre a segurança do código do aplicativo.

Através do executável da linha de comandos do cliente do AppScan Source for Automation (ou `onceauto`), você submete solicitações ao servidor. Ao processar solicitações, o Automation Server é executado como um cliente do AppScan Enterprise Server associado e pode ser conectado apenas a um único AppScan Enterprise Server. Ele atende em uma porta TCP (padrão 13205) para conexões a partir do host local apenas.

- Nos sistemas Windows, o Automation Server é executado como o serviço **IBM Security AppScan Source Automation**.
- Em sistemas Linux, ele é executado como um daemon:
 - Para parar o daemon, emita este comando: `/etc/init.d/onceautod stop`
 - Para iniciar o daemon, emita este comando: `/etc/init.d/onceautod start`

O Automation Server processa solicitações como um usuário do AppScan Source especificado e, portanto, herda as permissões desse usuário. Este ID do usuário deve ter todas as permissões necessárias, dependendo dos comandos que precisar executar. Por exemplo, se o ID do usuário precisar executar o comando `PublishAssessment`, o ID do usuário poderá receber permissões de publicação e registro e não exigirá permissão para varredura (consulte a seção *Administering AppScan Source* do *AppScan Source Installation and Administration Guide* para obter detalhes adicionais). Submeter uma solicitação para o Automation Server não requer credenciais do usuário.

Especificando Credenciais de Login do Automation Server a partir da Linha de Comandos

Se você não tiver especificado as credenciais de login do Automation Server durante o processo de instalação, conforme descrito no *IBM Security AppScan Source Installation and Administration Guide*, deverá configurar o Automation Server após a instalação para ser executado como um usuário do AppScan Source.

Sobre Esta Tarefa

Neste tópico da tarefa,

- `<install_dir>` é o local da instalação do AppScan Source.
- `user_id` é o usuário com o qual o Automation Server é autenticado ao processar uma solicitação. O usuário deve ser definido no AppScan Source com as permissões requeridas.

Nota: Se o usuário não existir ainda, você precisará especificá-lo no painel de instalação ou via linha de comandos - e depois criá-lo manualmente (após a instalação) com o AppScan Source for Analysis ou o AppScan Source command line interface (CLI). Para criar o novo usuário pela linha de comandos, use o comando `"newuser (nu)"` na página 42. Ao criar o novo usuário, assegure-se de

especificar o mesmo nome de usuário e senha que foram especificados para o login do Automation Server. Outras configurações, como permissão, podem ser definidas de acordo com suas necessidades.

- password é a senha do usuário.
- A opção --persist preserva as credenciais de login no disco e cria um arquivo-chave criptografado com o nome do usuário e a senha especificados.
- Somente Windows: common_name é seu Nome Comum (CN) do CAC.
- Somente Windows: cn_certificate é o CN do emissor do certificado.

Procedimento

- **Nos sistemas Windows:**
 - Se o método de autenticação do AppScan Enterprise Server for ID do usuário e senha, emita este comando:

```
<install_dir>\bin\ounceautod.exe -u <user name> -p <password> --persist
```
 - Se o AppScan Enterprise Server estiver ativado para autenticação Common Access Card (CAC), insira este comando:

```
<install_dir>\bin\ounceautod.exe -u <common_name<cn_certificate>> --persist
```

Nota: Se o seu common_name ou cn_certificate contiver espaços, coloque o valor entre aspas ("common_name<cn_certificate>").

- **Nos sistemas Linux:**
 1. Mude seu LD_LIBRARY_PATH para incluir <install_dir>/bin (em que <install_dir> é o local da instalação do AppScan Source), por exemplo:

```
export LD_LIBRARY_PATH=/opt/ibm/appscansource/  
bin:$LD_LIBRARY_PATH
```
 2. Emita este comando:

```
<install_dir>/ibm/appscansource/bin/ounceautod  
-u <user_id> -p <password> --persist
```

Resultados

Depois de especificar as credenciais de login, elas são salvas em disco.

O Arquivo de Configuração do Automation Server

O arquivo de configuração ounceautod, ounceautod.ozsettings, especifica as propriedades para o daemon Automation Server. O arquivo ounceautod.ozsettings reside em <data_dir>\config (em que <data_dir> é o local dos dados do programa do AppScan Source, conforme descrito em Capítulo 9, "Locais de Arquivos de Instalação e de Dados do Usuário", na página 149).

O arquivo ounceautod.ozsettings contém estas propriedades:

Propriedade	Valor	Descrição
ounceautod_max_concurrent_requests	1 (Padrão)	Número de solicitações simultâneas permitidas.
ounceautod_accept_ssl	true (Padrão)	Aceitar certificados SSL automaticamente. Para obter informações adicionais, consulte "Certificados SSL do AppScan Enterprise Server" na página 39.

Propriedade	Valor	Descrição
ounceautod_port	13205 (padrão)	Número da porta na qual executar ounceautod. Nota: O valor de porta nesta configuração deve corresponder à configuração de porta no arquivo ounceauto.ozsettings em <data_dir>\config.
ounceautod_server_hostname	<nome_do_host>:porta	Nome do host do computador no qual o AppScan Enterprise Server foi instalado e a porta na qual ele executa nesse computador.

Se este arquivo for modificado, você deverá reiniciar o serviço Automation Server (descrito em Capítulo 6, “AppScan Source for Automation”, na página 97).

Criação de Log do Automation Server

O Automation Server gera um arquivo de log, ounceautod.log, que registra qualquer solicitação feita para o Automation Server bem como a resposta. As entradas de log residem em <data_dir>\logs\ounceautod.log (em que <data_dir> é o local dos dados do programa do AppScan Source, conforme descrito em Capítulo 9, “Locais de Arquivos de Instalação e de Dados do Usuário”, na página 149).

O Automation Server registra os eventos a seguir para todas as solicitações:

- Solicitação recebida
- Solicitação iniciada
- Solicitação concluída
- Falha na solicitação

Cada entrada de log contém:

- Registro de data e hora de entrada de log
- Tipo de evento
- Tipo de solicitação
- ID do Pedido
- Responsável pela chamada, se especificado
- Informações específicas da solicitação

Usando Ounceauto a partir da Linha de Comandos

A interface para AppScan Source for Automation, ounceauto, emite comandos para o servidor especificado pelos argumentos da linha de comandos, imprime a saída, se houver alguma, e retorna. Cada solicitação recebe um ID do Pedido e todas as entradas de log relativas à solicitação contêm o ID do Pedido para propósitos de identificação.

Execute ounceauto a partir da linha de comandos como segue:

```
ounceauto <nome_do_comando> <argumentos_de_comando>
```

Comandos `Onceauto` incluem:

- `GenerateReport`
- `PublishAssessment`
- `PublishAssessmentASE`
- `ScanApplication`
- `Wait`

GenerateReport

Descrição

Crie um relatório a partir de uma avaliação.

Sintaxe

`onceauto GenerateReport`

```
-assessment <caminho_de_avaliação>  
-type <tipo_de_relatório>  
-output <formato de saída>  
-file <local de saída>  
[-caller <responsável pela chamada>]  
[-includeSrcBefore <n>]  
[-includeSrcAfter <n>]  
[-includeTraceDefinitive]  
[-includeTraceSuspect]  
[-includeTraceCoverage]
```

- `-assessment <assessment path>`: Caminho para o arquivo de avaliação para o qual você deseja gerar o relatório.
- `-type "<report type>"`: Nome do tipo de relatório dentro de aspas duplas. Os tipos de relatório consistem em relatórios de Descobertas, relatórios do AppScan Source e relatórios customizados.

Os tipos de relatórios do AppScan Source incluem:

- Um relatório de Descobertas:
 - Descobertas por Pacote Configurável
 - Descobertas por API
 - Descobertas por Classificação
 - Descobertas
 - Atividade do DTS
 - Descobertas por Tipo
 - Descobertas por CWE
 - Descobertas por Arquivo
- Um relatório do AppScan Source:
 - 25 principais do CWE SANS de 2011
 - DISA Application Security and Development STIG V3R10
 - 10 principais do dispositivo móvel OWASP
 - 10 principais do OWASP de 2013
 - PCI Data Security Standard V3.2
 - Perfil de Segurança de Software
- Um relatório customizado, se disponível.

Ao inserir o tipo de relatório, dentro de aspas duplas, insira exatamente conforme especificado na lista acima - por exemplo Provas por Classificação ou Perfil de Segurança do Software.

- -output <output format> : Especifique um desses formatos para este relatório,
 - html: Gera o relatório como HTML e o exibe on-line.
 - zip: Cria um arquivo ZIP que contém todos os componentes do relatório de HTML
 - Para relatórios em formato PDF, é possível especificar o nível de detalhe:
 - pdf-summary: Contém contagens para cada grupo de relatórios customizados
 - pdf-detailed: Contém contagens para cada API para cada propriedade de vulnerabilidade
 - pdf-comprehensive: Contém tabelas compostas por todas as descobertas para cada API
 - pdf-annotated: Contém todas as descobertas, qualquer nota incluída com as descobertas e fragmentos de código designados
 - output location: O caminho de arquivo para gravar o relatório.
- -file <output location>: Especifique o caminho e o nome do arquivo no qual deseja salvar o relatório.
- -caller <responsável pela chamada>: Opcional. Designe um responsável pela chamada à operação de geração de relatórios. O responsável pela chamada pode ser o nome do usuário real, mas isto não é necessário. O nome do responsável pela chamada é gravado no arquivo de log ounceauto.
- -includeSrcBefore <n>: Opcional. O número de linhas de código fonte a ser incluídas antes de cada descoberta.
- -includeSrcAfter <n>: Opcional. O número de linhas de código fonte a ser incluídas após cada descoberta.
- -includeTraceDefinitive: Opcional. Inclua as informações de rastreamento no relatório para provas definitivas (consulte "Classificações" na página 151 para saber mais sobre classificações de provas).
- -includeTraceSuspect: Opcional. Inclua informações de rastreamento no relatório para provas suspeitas.
- -includeTraceCoverage: Opcional. Inclua informações de rastreamento no relatório para verificar provas suspeitas.

Retornar Valor

O ID de Pedido se for bem-sucedido ou -1 se o envio de pedido for malsucedido.

Exemplos

- Gere um relatório Provas por API como um arquivo HTML. No relatório, inclua as informações de rastreamento para provas definitivas:


```
ounceauto generatereport -assessment C:\Ounce\Data\Webgoat.ozasmt
-type "Descobertas por API" -output html
-file C:\reports\Webgoat_Findings.html
-includeTraceDefinitive
```
- Para gerar 10 principais relatórios de um OWASP de 2013AppScan Source como um PDF:


```
ounceauto generatereport -assessment C:\Ounce\Data\Webgoat.ozasmt
-type "OWASP Top 10 2013" -output pdf-annotated
-file C:\Reports\Webgoat_OWASP.pdf
```

PublishAssessment

Descrição

Publique a avaliação selecionada para o banco de dados do AppScan Source.

Sintaxe

```
ounceauto PublishAssessment -file <assessment.ozasmt>  
[-caller <responsável pela chamada>]
```

- -file <arquivo de avaliação>: Caminho completo para um arquivo de avaliação.
- -caller <responsável pela chamada>: Opcional. Designe um responsável pela chamada à operação de geração de relatórios. O responsável pela chamada pode ser o nome do usuário real, mas isto não é necessário. O nome do responsável pela chamada é gravado no arquivo de log ounceauto.

Retornar Valor

O ID de Pedido se for bem-sucedido ou -1 se o envio de pedido for malsucedido.

Exemplo

Para publicar a avaliação WebGoat_Internal:

```
ounceauto publishassessment -file C:\Ounce\Data\WebGoat_Internal.ozasmt
```

PublishAssessmentASE

Descrição

Publique a avaliação selecionada para o AppScan Enterprise Console.

Sintaxe

```
ounceauto PublishAssessmentASE -file <assessment_file>  
[-aseapplication <ase_application>] [-caller <caller>]  
[-folder <location>] [-name <published_assessment_name>]  
[-preventOverwrite]
```

- -file <assessment_file>: Necessário. Caminho e nome do arquivo do arquivo de avaliação.
- -aseapplication <ase_application>: Esta opção é necessária quando se está conectado ao AppScan Enterprise Server Versão 9.0.3 e superior (a menos que você desative o requisito, conforme descrito aqui). A associação de um aplicativo é opcional quando se está conectado a versões anteriores do AppScan Enterprise Server. Use essa opção para especificar o aplicativo Enterprise Console ao qual associar a avaliação.
- -caller <responsável pela chamada>: Opcional. Designe um responsável pela chamada à operação de geração de relatórios. O responsável pela chamada pode ser o nome do usuário real, mas isto não é necessário. O nome do responsável pela chamada é gravado no arquivo de log ounceauto.
- -folder <location>: Opcional. Essa opção se aplica somente quando estiver conectada a versões do AppScan Enterprise Server anteriores a Versão 9.0.3. Especifique a pasta do Enterprise Console para a qual publicar. Se este argumento não for usado, a avaliação será publicada em sua pasta Enterprise Console padrão.

- `-name <published_assessment_name>`: Opcional. Nome que a avaliação será salva como no Enterprise Console. Se este argumento não for usado, um nome será gerado com base no aplicativo AppScan Source digitalizado para produzir a avaliação (este nome será pré-anexado com AppScan Source:).
- `-preventOverwrite`: opcional. Inclua esse argumento para evitar a publicação se uma avaliação do mesmo nome já existir no servidor.

Retornar Valor

O ID de Pedido se for bem-sucedido ou -1 se o envio de pedido for malsucedido.

Exemplo

Para publicar a avaliação WebGoat_Internal para o AppScan Enterprise Server Versão 9.0.3 ou superior:

```
ounceauto publishassessmentase -file C:\Ounce\Data\WebGoat_Internal.ozasmt
-aseapplication myapplication
```

Importante:

Ao fazer upgrade para o AppScan Source Versão 9.0.3.4, essas mudanças serão notadas:

- Ao publicar uma avaliação no AppScan Enterprise Console, agora deve-se associar a avaliação com um aplicativo em AppScan Enterprise (se você estiver executando o AppScan Enterprise Server Versão 9.0.3 e superior). Como resultado, scripts de automação podem falhar se não incluírem a associação do aplicativo. No AppScan Enterprise Server, a associação do aplicativo é necessária se desejar aproveitar os recursos de gerenciamento de risco de segurança do aplicativo AppScan Enterprise Server. Consulte http://www.ibm.com/support/knowledgecenter/SSW2NF_9.0.3/com.ibm.ase.help.doc/topics/c_overview.html.
- Além disso, remova a porta da URL do AppScan Enterprise.
 1. No AppScan Source for Analysis, clique em **Editar** > **Preferências**.
 2. Nas configurações do Console AppScan Enterprise, remova a porta do campo **URL do Console Enterprise**.
- Após publicar sua avaliação, ela estará disponível somente na visualização do monitor do AppScan Enterprise (em liberações anteriores, a avaliação estava disponível na visualização de varredura do AppScan Enterprise). A migração para essa visualização está descrita em http://www.ibm.com/support/knowledgecenter/SSW2NF_9.0.3/com.ibm.ase.help.doc/topics/t_workflow_for_applications.html.

Esse é o resultado de um protocolo de comunicação alterado entre o AppScan Source e o AppScan Enterprise Server que é necessário para publicar para o AppScan Enterprise Server ao usar a autenticação Common Access Card (CAC).

Se não desejar publicar avaliações no AppScan Enterprise Server quando a autenticação CAC estiver habilitada ou se não desejar aproveitar os recursos de gerenciamento de risco de segurança do aplicativo Enterprise Server, é possível reverter para o protocolo de comunicação anterior da seguinte maneira:

1. Abra `<data_dir>\config\ounce.ozsettings` (em que `<data_dir>` é o local dos dados do programa do AppScan Source, conforme descrito em Capítulo 9, “Locais de Arquivos de Instalação e de Dados do Usuário”, na página 149)).
2. Neste arquivo, localize esta configuração:

```

<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. Na configuração, mude `value="false"` para `value="true"` e, em seguida, salve o arquivo.
4. Reinicie o produto AppScan Source do qual você publicará avaliações.

Quando essa configuração estiver definida como `value="true"`:

- Se você associar uma avaliação a um aplicativo no AppScan Enterprise ao publicar, a avaliação ficará disponível nas visualizações de Monitor e de Varredura.
- Se você não associar uma avaliação a um aplicativo ao publicar, a avaliação ficará disponível na visualização de Varredura.
- Não será possível publicar avaliações no AppScan Enterprise Server quando a autenticação CAC estiver habilitada.

Para obter informações adicionais, consulte <http://www.ibm.com/support/docview.wss?uid=swg21993010>.

ScanApplication

Descrição

Varra o aplicativo especificado e execute outras ações relacionadas à varredura.

Sintaxe

ounceauto ScanApplication

```

-application <nome do aplicativo>|
  -application_file <caminho do arquivo de aplicativo>
[-name <nome da avaliação>]
[-scanconfig <scan_configuration_name>]
[-save <nome do arquivo>]
[-caller <responsável pela chamada>]
[-publish]
[-clearcache]
[-report <tipo de relatório> <formato de saída>
<local de saída>]
[-includeSrcBefore <n>]
[-includeSrcAfter <n>]
[-includeTraceDefinitive]
  on[-includeTraceSuspect]
[-includeTraceCoverage]
[-appserver_type]
[-include_all_lib_jars]
[-include_lib_jars]
[-no_ear_project]

```

- `-application <nome do aplicativo>` **ou** `-application_file <caminho do arquivo de aplicativo>`: Um desses são necessários.
 - Se você especificar `-application <nome do aplicativo>`, indique o nome do aplicativo a ser verificado.

- Se você especificar `-application_file <path to application file>`, indica o caminho completo e o nome do arquivo para um destes tipos de arquivos:
 - Arquivos de Aplicativo do AppScan Source (.paf).
 - Eclipse ou áreas de trabalho Rational Application Developer for WebSphere Software (RAD) (.ewf)

Nota: Os arquivos .ewf são gerados quando você usa `openapplication` para abrir um diretório da área de trabalho (especificando seu caminho).

- Arquivos WAR (.war)
- Arquivos EAR (.ear)
- Somente Windows: Microsoft arquivos Visual C++ Workspace (.dsw)
- Somente Windows: Microsoft arquivo da solução Visual Studio.NET (.sln)

Nota: Para saber quais versões de arquivos importados são suportados pelo AppScan Source for Analysis, pelo AppScan Source for Automation e pelo Interface da linha de comandos do AppScan Source, consulte <http://www.ibm.com/support/docview.wss?uid=swg27027486>. Nesta página, selecione a guia para a versão do AppScan Source que está usando e, em seguida, selecione o componente do AppScan Source que está usando. Se o AppScan Source suportar a abertura e a varredura de arquivos de outros ambientes de desenvolvimento, esse suporte será listado na seção **Compiladores e idiomas** da guia **Software suportado**.

- `-name <nome da avaliação>`: Opcional. Um nome para a avaliação.
- `-scanconfig <scan_configuration_name>`: Opcional. Especifique o nome de uma configuração de varredura para usar para a varredura. Se uma configuração de varredura não for especificada, a configuração de varredura padrão será usada para a varredura.
- `-save <nome do arquivo>`: Opcional. Salve os resultados da avaliação nesse arquivo.
- `-caller <responsável pela chamada>`: Opcional. Atribua um responsável pela chamada para a operação. O responsável pela chamada pode ser o nome do usuário real, mas isto não é necessário. O nome do responsável pela chamada é gravado no arquivo de log `ounceauto`.
- `-publish`: Opcional. Publicar a avaliação após a varredura.
- `-clearcache`: Opcional. Remova o cache de análise de vulnerabilidade e os dados de assinatura das regras customizadas antes da varredura. Se tiver a análise incremental Java ativada, a varredura será integral.
- `-report`: Opcional. Gerar um relatório após a varrer.
 - Opções do comando `-report` necessárias:
 - `<report type>`: Tipo de relatório. Os tipos de relatório consistem em relatórios de Descobertas, relatórios do AppScan Source e relatórios customizados. Consulte as opções em “GenerateReport” na página 100.
 - `<formato de saída>`: Especifique o formato de saída. Consulte as opções em “GenerateReport” na página 100.
 - `<output location>`: O local para salvar o relatório.
 - Opções do comando `-report` opcionais:
 - `-includeSrcBefore <n>`: O número de linhas do código-fonte a serem incluídas antes de cada descoberta.
 - `-includeSrcAfter <n>`: O número de linhas do código-fonte a serem incluídas após cada descoberta.

- `-includeTraceDefinitive`: Inclua as informações de rastreamento no relatório para provas definitivas (consulte “Classificações” na página 151 para saber mais sobre classificações das provas).
- `-includeTraceSuspect`: Inclua as informações de rastreamento no relatório para provas suspeitas.
- `-includeTraceCoverage`: Inclua informações de rastreamento no relatório para descobertas de cobertura de varredura.
- `-appserver_type`: Opcional. Se o aplicativo que você está abrindo inclui páginas de JavaServer (por exemplo, um arquivo WAR ou EAR), use essa configuração para especificar o servidor de aplicativos para usar a compilação de JSP. Especifique um desses, entre aspas duplas:
 - Tomcat 7
 - Tomcat 8
 - WebSphere 7.0
 - WebSphere 8.0
 - WebSphere 8.5
 - WebLogic 11g
 - WebLogic 12c

Nota:

- Antes de especificar um servidor de aplicação, certifique-se de que tem sido configurado corretamente na preferências AppScan Source for Analysis.
- Se `-appserver_type` não for usado, o compilador JSP padrão que está definido em AppScan Source for Analysis será usada para compilação de JSP. Fora-da-caixa, Tomcat 7 é o compilador JSP padrão.
- Para arquivos WAR:
 - `-include_all_lib_jars`: use essa configuração para incluir todas as bibliotecas no arquivo WAR durante a varredura.
 - `-include_lib_jars`: use essa configuração para especificar as bibliotecas no arquivo WAR que deseja incluir durante a varredura. Ao usar a configuração, não inclua informações do caminho da biblioteca e separe várias bibliotecas com vírgulas.
- `-no_ear_project`: ao importar um arquivo EAR, um projeto será criado automaticamente para armazenar bibliotecas compartilhadas. Se não houver bibliotecas compartilhadas, o projeto será criado, mas estará vazio. Quando essa configuração for usada, nenhum projeto será criado para o arquivo EAR.

Retornar Valor

O ID de Pedido se for bem-sucedido ou -1 se o envio de pedido for malsucedido.

Exemplos

- Verifique o aplicativo WebGoat, publique-o e anote o log com John Smith como o responsável pela chamada:


```
ounceauto scanapplication -application_file C:\WebGoat\WebGoat.paf
-publish -caller JohnSmith
```
- Verifique o aplicativo WebGoat e crie um relatório de Provas no diretório C:\WebGoat. No relatório, inclua as informações de rastreamento para provas definitivas:


```
ounceauto scanapplication -application WebGoat
-report Findings html C:\WebGoat\MyReport.html
-includeTraceDefinitive
```

- Varra um arquivo WAR e inclua somente algumas de suas bibliotecas:
ounceauto scanapplication -application_file c:\mywar.war
-include_lib_jars lib1.jar,lib2.jar

Wait

Descrição

Bloquear até que a solicitação especificada seja concluída.

Sintaxe

```
ounceauto wait -requestid <id_do_pedido>
```

-requestid <id_do_pedido>: O ID do pedido para aguardar.

Retornar Valor

Se a solicitação correspondente ao <id_do_pedido> foi concluída com êxito, o valor de retorno será 0, caso contrário -1.

Exemplo

O seguinte exemplo do Windows ilustra uma varredura de um arquivo de aplicativo e, em seguida, aguarda a varredura ser concluída.

```
ounceauto scanapplication -application_file WG0.paf  
ounceauto wait -requestid %errorlevel%
```

No Linux, o equivalente deste exemplo é:

```
ounceauto scanapplication -application_file WG0.paf  
ounceauto wait -requestid $?
```

Capítulo 7. APIs de Manipulação de Framework for Frameworks

O AppScan Source fornece um conjunto de APIs Java que permitem incluir suporte para estruturas que são usadas em seus aplicativos. As classes e os métodos oferecidos nessas APIs permitem que você se responsabilize por estruturas para as quais o suporte integrado não é fornecido.

Nota: AppScan Source inclui suporte integrado para estas estruturas:

- Apache Struts 1 e 2
- Spring MVC 2.5 e 3
- ASP .NET MVC (Windows apenas)
- Enterprise JavaBeans (EJB) 2
- ASP .NET (Windows apenas)
- J2EE
- JavaServer Faces (JSF) 2
- .NET 4.5 (Windows apenas)
- Jax - RS (V1.0 e V1.1)
- Jax - WS (V2.2)

Estruturas modernas movimentam uma grande quantidade de informações que afeta o comportamento de tempo de execução dos aplicativos fora do código fonte normal e nos arquivos de configuração e anotações. Antes isso resultava em pontos cegos durante análises estáticas. Equipes de produtos poderiam criar regras customizadas para aplicativos individuais, porém, não existia nenhuma estrutura que pudesse descrever de forma flexível as atividades dessas estruturas de maneira automatizada.

Ao usar as APIs do Framework for Frameworks, você pode rápida e facilmente incluir suporte para novas estruturas diretamente no AppScan Source. Isso é realizado pelo processamento das informações de configuração associadas à estrutura e fornecimento desses dados de volta para o AppScan Source através das APIs associadas.

As APIs Framework for Frameworks estão incluídas na instalação destes produtos:

- AppScan Source for Automation
- AppScan Source for Analysis
- AppScan Source for Development

As APIs são instaladas em <install_dir>\walalib (em que <install_dir> é o local da instalação do AppScan Source).

Um archive de projeto de exemplo é instalado em <data_dir>\samples\F4FEjbExample.zip (em que <data_dir> é o local dos dados do programa do AppScan Source, conforme descrito em Capítulo 9, “Locais de Arquivos de Instalação e de Dados do Usuário”, na página 149).

Nota: Os nós Rastreo com nomes de classe que iniciam com `Appscan.Synthetic`, `Appscan.Synthetic.Validator` e `AppScan.Synthetic.Replacement` correspondem aos métodos que são sintetizados por AppScan Source.

- Os métodos `AppScan.Synthetic` são usados para atar os rastreios no código do aplicativo que usa as estruturas.
- Um método `AppScan.Synthetic.Validator` modela a validação subjacente executada pelo tempo de execução da estrutura. É possível selecionar um método validador e marcá-lo como **Validator** se necessário.
- Um método `AppScan.Synthetic.Replacement` indica que um método no código do aplicativo foi substituído por AppScan Source para capturar o fluxo de dados entre os componentes de logout (como controladores e visualizações) da estrutura.

Principais Componentes da API Framework for Frameworks

F4Handler

F4Handler é a classe abstrata que todos os novos manipuladores de estrutura devem estender. Ela força o fluxo de trabalho de um manipulador de estrutura por meio dos métodos abstratos que devem ser substituídos. Ela também fornece acesso aos outros dois componentes de API, F4App e F4Action.

F4App

Fornecer informações sobre o aplicativo sendo varrido, incluindo informações de sua configuração (como raízes de origem e raiz da Web). Esse componente também fornece acesso a cada classe no aplicativo com o objetivo de determinar informações necessárias, como anotações e superclasses.

F4Action

Fornecer informações para o AppScan Source sobre o aplicativo que está sendo varrido. Pontos de entrada, conexões entre métodos abstratos e suas implementações e muito mais podem ser descritos para o mecanismo de varredura por meio da chamada dos métodos fornecidos nessa API.

Usando APIs Framework for Frameworks

Esta seção destaca as etapas necessárias durante o uso de APIs Framework for Frameworks. Quando apropriado, essas etapas são relacionadas com o exemplo fornecido, permitindo que você o acompanhe.

Neste exemplo, vamos:

- Criar uma classe que estenda `F4Handler`
- Implementar dois métodos abstratos do `F4Handler`
- Criar um Java archive (.jar) com um arquivo de manifesto que identifique sua classe manipuladora
- Exportar o .jar para o diretório AppScan Source `wafgens`
- “Sobre o Exemplo” na página 111
- “Importando o Projeto de Exemplo para o Eclipse ou Rational Application Developer for WebSphere Software (RAD)” na página 111
- “Criando uma Classe que Implementa `F4Handler`” na página 114

- “Criando um Arquivo de Manifesto para seu Manipulador” na página 115
- “Criando um JAR para seu Manipulador” na página 115
- “Exportando o JAR para waf1gens” na página 118

Sobre o Exemplo

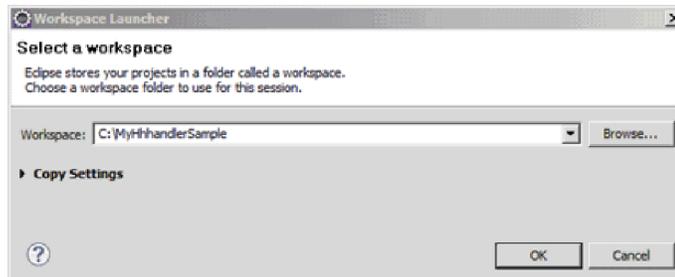
Incluído com a instalação do AppScan Source há um manipulador de amostra que lê e fornece informações do Framework for Frameworks sobre os aplicativos Enterprise Java Bean (EJB) 2 (também referido como Enterprise Java Beans ou apenas Beans).

EJB é uma estrutura que tenta facilitar o trabalho e a reutilização de componentes de lógica de negócios. Cada bean representa um componente de negócios e deve ter classes de interface associadas que outros Beans devem usar para interagir com ele. Essas classes de interface são associadas ao Bean por meio do arquivo de configuração EJB do aplicativo (EJB-jar.xml). Isso cria um ponto cego, já que o mecanismo de análise estática não selecionará essas associações na origem, o que torna a estrutura EJB particularmente adequada para manipulação por meio de Framework for Frameworks.

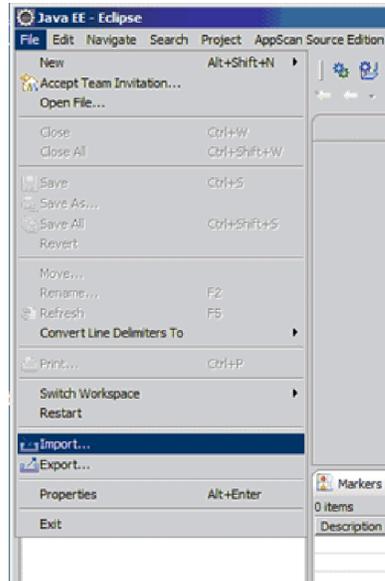
Importando o Projeto de Exemplo para o Eclipse ou Rational Application Developer for WebSphere Software (RAD)

Procedimento

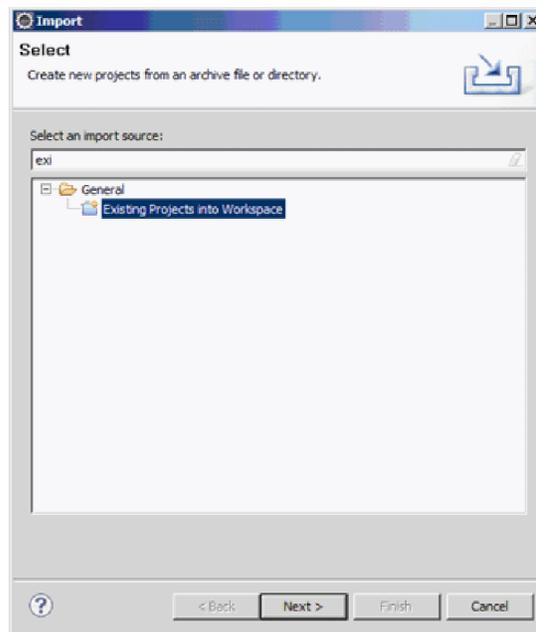
1. Inicie o Eclipse.
2. Crie uma nova área de trabalho para seu projeto manipulador.



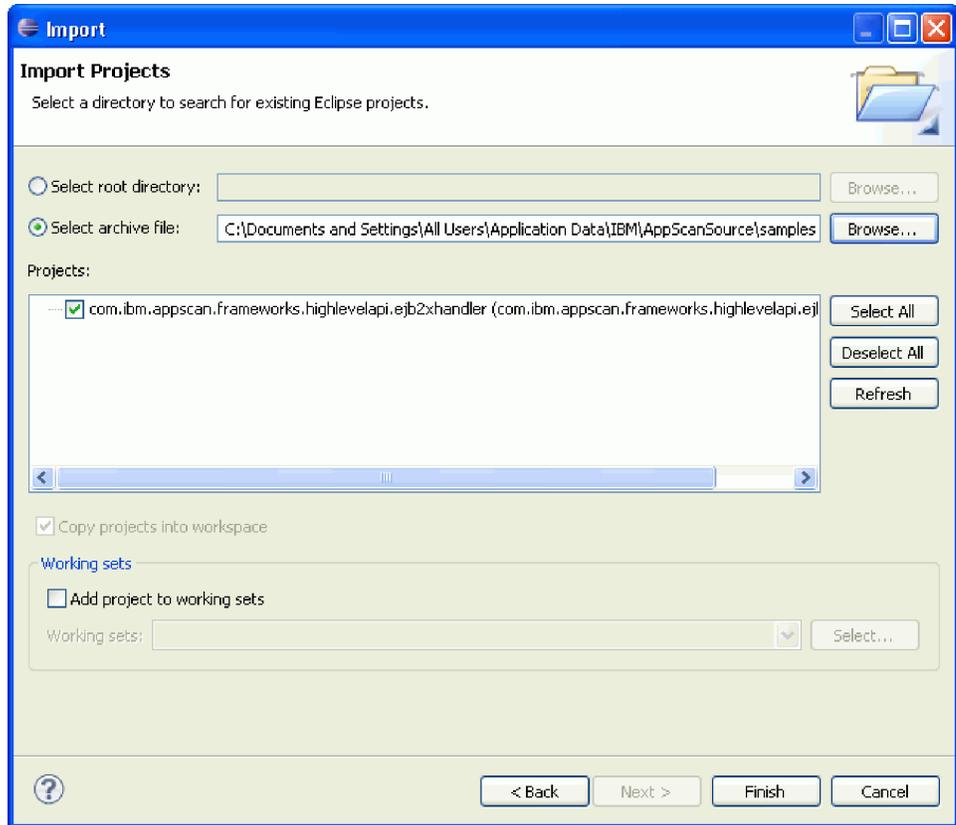
3. Selecione **Arquivo > Importar** no menu do ambiente de trabalho principal.



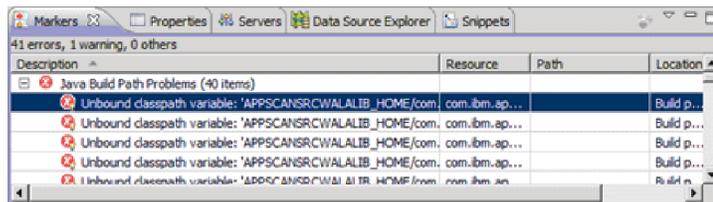
4. Na página **Selecionar** do Assistente de importação, selecione **Projetos Existentes para a Área de Trabalho** e, então, clique em **Avançar**.



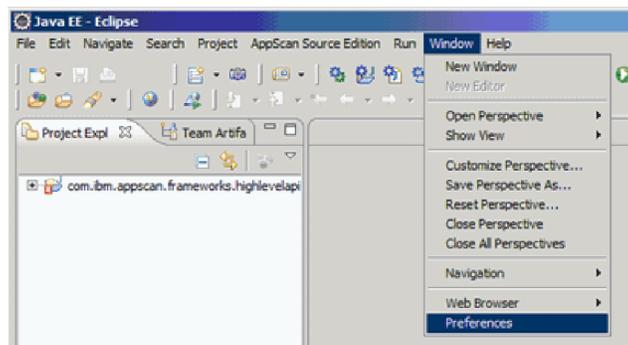
5. Selecione o botão de opções **Selecionar Archive** e em seguida clique em **Navegar**. Na caixa de diálogo **Selecionar archive contendo os projetos a serem importados**, localize `<data_dir>\samples\F4FEjbExample.zip` (em que `<data_dir>` é o local dos dados do programa do AppScan Source, conforme descrito em Capítulo 9, “Locais de Arquivos de Instalação e de Dados do Usuário”, na página 149) e, então, clique em **Abrir**.
Esse arquivo zip é um archive Eclipse que contém o código fonte e configuração para obter o exemplo de manipulador de estrutura Ejb2xHandler. Clique em **Concluir** para importar o archive.



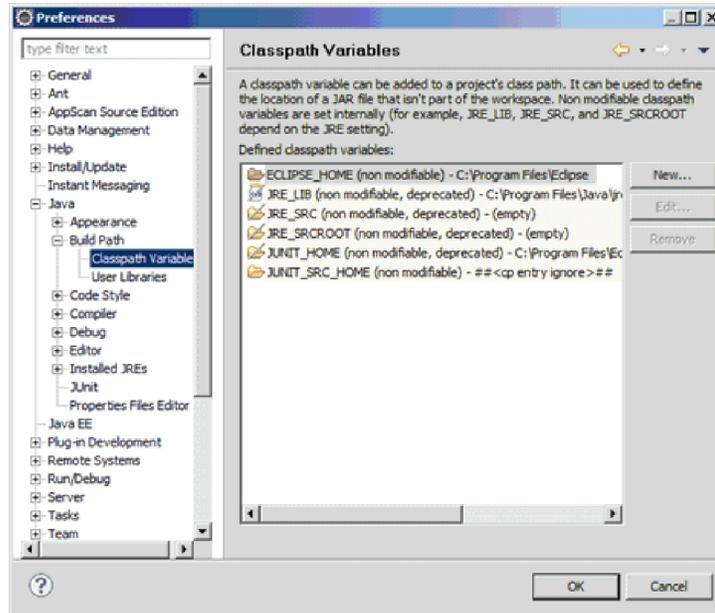
6. Após o archive ter sido importado, você vai observar inúmeros erros de construção. O exemplo tem uma variável de caminho de classe que aponta para o diretório que contém suas bibliotecas.



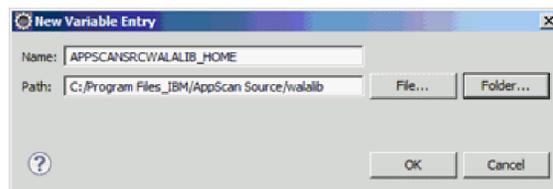
7. Em seguida, você vai definir uma variável. Selecione **Janela > Preferências** do menu do ambiente de trabalho principal.



8. Na caixa de diálogo Preferências, selecione **Java > Caminho de Construção > Variáveis de Caminho de Classe** para abrir a página de preferências Variáveis de Caminho de Classe. Clique em **Novo**.



9. Crie uma nova variável denominada APPSCANSRCWALALIB_HOME e configure o **Caminho** como <install_dir>\walalib (em que <install_dir> é o local da instalação do AppScan Source).



Após você clicar em **OK** em todas as caixas de diálogo, a amostra deverá ser completamente construída sem erros.

Criando uma Classe que Implementa F4Handler

Para criar um novo manipulador de estrutura, primeiro você deve criar uma classe que implemente F4Handler. Dois métodos devem ser implementados para o suporte da funcionalidade Framework for Frameworks.

Substituir isApplicable

Objetivo: O AppScan Source chama isApplicable para determinar se ele deve executar o manipulador. Se você retornar True, ele executará seu manipulador chamando handleApp. Se você retornar False, nada será chamado.

Nota: isApplicable inclui uma verificação no início do método para assegurar que o aplicativo seja um aplicativo Java antes de continuar.

Observe no exemplo: No Ejb2xHandler, isApplicable primeiramente verifica se o idioma é apropriado (já que EJB está apenas em aplicativos Java). Se o aplicativo é baseado em Java, isApplicable procura por quaisquer instâncias do ejb-jar.xml, que é o arquivo de configuração necessário para um aplicativo EJB 2. Se os arquivos de configuração forem localizados, eles serão lidos no manipulador e True será retornado para permitir que o AppScan Source saiba que ele deve chamar handleApp para lidar com as informações contidas nos arquivos de configuração.

Substituir handleApp

Propósito: O AppScan Source chama handleApp para permitir determinar e configurar informações sobre a estrutura ou estruturas em uso pelo aplicativo atual sendo varrido. Os parâmetros F4FApp e F4FAction são usados para obter informações sobre o aplicativo e configurar características específicas sobre como manipular as estruturas que estão presentes e que são manipuladas pelo seu manipulador.

Criando um Arquivo de Manifesto para seu Manipulador

Crie um arquivo denominado Manifest.txt para conter as informações necessárias para seu manifesto do JAR quando ele for exportado. Especificamente, o AppScan Source procurará a entrada Framework-Handler ao carregar o manipulador JAR para determinar qual classe contém a funcionalidade F4FHandler.

Coloque essas duas linhas no arquivo Manifest.txt:

```
Manifest-Version: 1.0  
Framework-Handler: package.HandlerClassName
```

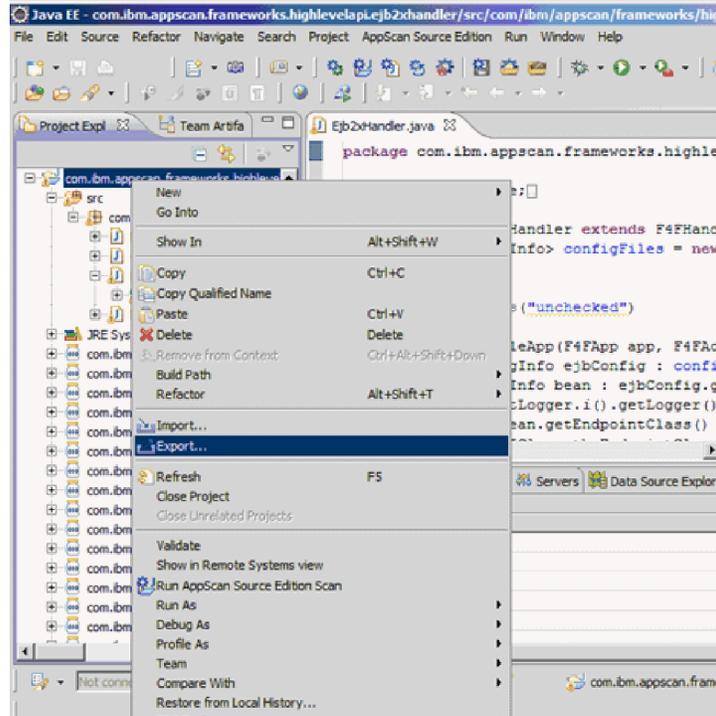
Em que package.HandlerClassName é o nome de classe e o pacote completo para sua classe que implementa a interface F4FHandler.

Observe no exemplo: Manifest.txt contém as informações que serão colocadas no manifesto do arquivo JAR do Ejb2xHandler quando ele for criado. Há apenas duas linhas, uma listando a versão e a outra especificando a classe de manipulação de estrutura, Framework-Handler:
com.ibm.appscan.frameworks.highlevelapi.ejb2xhandler.Ejb2xHandler.

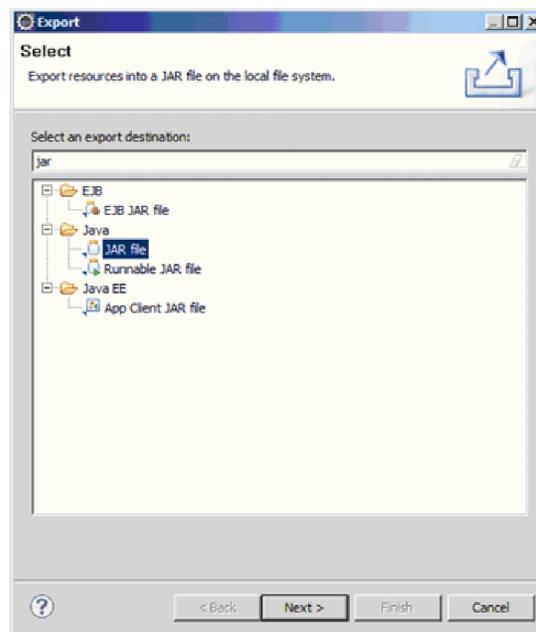
Criando um JAR para seu Manipulador

Procedimento

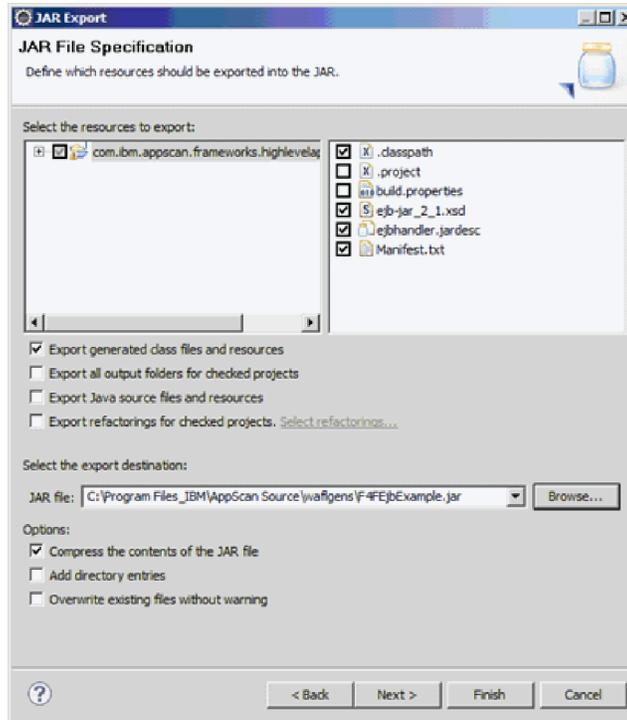
1. Clique com o botão direito do mouse em seu projeto na visualização Eclipse Project Explorer e selecione **Exportar**.



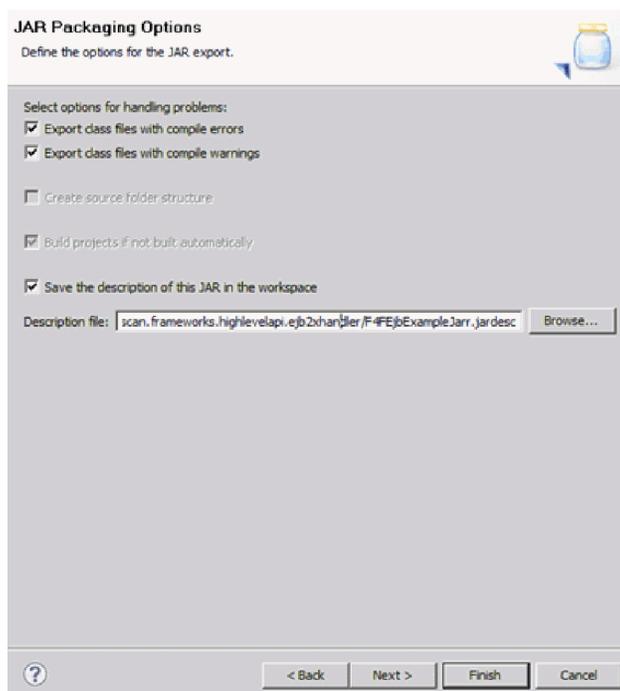
2. No Assistente de exportação, selecione **Arquivo JAR** e, então, clique em **Avançar**.



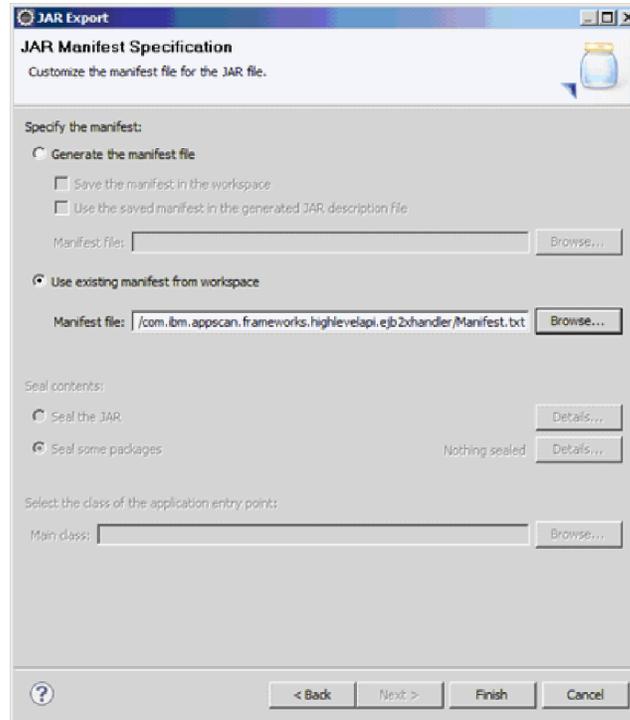
3. Na seção **Selecionar o destino de exportação** de Exportação Java, selecione um local adequado para seu arquivo JAR e forneça um nome. Clique em **Avançar**.



4. Clique em **Avançar** novamente. Opcionalmente, também é possível configurar um local e um nome para um arquivo de descrição. Isso salvará um arquivo `.jardesc` em seu projeto que conterá todas as configurações de exportação de JAR dessa exportação. Com isso, é possível repetir a exportação sem especificar as configurações novamente.



5. Selecione o botão de opções **Usar manifesto existente da área de trabalho** e clique em **Navegar**. Selecione o arquivo `Manifest.txt` criado anteriormente.



6. Clique em **Concluir**

Exportando o JAR para wafgens

Se você não exportou o JAR do manipulador diretamente para o diretório wafgens durante sua criação, copie-o agora. O caminho completo é <install_dir>\wafgens (em que <install_dir> é o local da instalação do AppScan Source).

Ações Comuns Executadas pelo Manipulador

Criar um Ponto de Entrada de Serviço da Web

Muitas estruturas fornecem seus próprios pontos de entrada em um aplicativo. Um exemplo comum é expor serviços da Web que são identificados em um arquivo de configuração ou em anotações no código. Após a procura nos arquivos de configuração do aplicativo ou diretamente no bytecode para pontos de entrada designados, o método `F4FAction.addTaintedCallback` pode ser usado para criar um ponto de entrada de dados corrompido no método apropriado.

Observe no exemplo: No EJB 2, pontos de entrada de serviço da Web são declarados definindo *terminais* no arquivo de configuração do aplicativo (`ejb-jar.xml`). Em seguida, `handleApp` faz um loop nos beans declarados no `ejb-jar.xml` e, sempre que uma classe de terminal é definida, obtém a lista de nomes de métodos. Então ele declara suas implementações como um ponto de entrada de serviço da Web usando o método `addTaintedCallback`.

Substituir um Método

Estruturas modernas fazem uso de abstração e funções virtuais com frequência para acoplar fracamente componentes de negócios. Embora possa ser uma melhoria no processo de desenvolvimento, isso cria dificuldades para a análise

estática quando a conexão entre função virtual e sua implementação é manipulada em um arquivo de configuração ou via anotações no código. `F4FAction.replaceCalls` permite que um manipulador designe essas conexões.

Observe no exemplo: No EJB 2, cada bean tem um conjunto de interfaces (locais e remotas) que declara como outros beans podem interagir com ele. Isso significa que, sempre que a interface de um bean `class.method` é chamada, a estrutura a substitui pelo verdadeiro `ImplementationClass.method`.

A partir da linha 62, nosso manipulador de exemplo efetua loop por meio de cada bean e toma suas interfaces locais e as substitui por suas implementações reais.

Criação de Log

Um manipulador pode usar a classe `com.ibm.wala.andromeda.util.logging.TaintLogger` para registrar mensagens informativas durante a execução - e fazer com que mensagens de erro apareçam na interface com o usuário do AppScan Source. A classe `TaintLogger` emprega a biblioteca `log4j`. Para registrar uma mensagem, primeiro obtenha um objeto `Logger` chamando `TaintLogger.i().getLogger()`. Em seguida, chame os métodos de criação de log no `Logger` (por exemplo, `Logger.warn`) para registrar mensagens desejadas. As mensagens de log aparecerão em `<data_dir>\logs\scanner_exceptions.log` (em que `<data_dir>` é o local dos dados do programa do AppScan Source, conforme descrito em Capítulo 9, “Locais de Arquivos de Instalação e de Dados do Usuário”, na página 149). Se `Logger.error` ou `Logger.fatal` for usado para registrar uma mensagem, a mensagem de erro também aparecerá na visualização Console na interface com o usuário AppScan Source.

Framework for Frameworks API classes and methods

This section contains the Framework for Frameworks API classes and methods, abbreviated for Adobe PDF. To access the complete set of API documentation, launch the AppScan Source for Analysis online help and navigate to Reference > Framework for Frameworks handling APIs > Framework for Frameworks API classes and methods.

The complete set of API documentation can also be found at <http://www.ibm.com/support/knowledgecenter/SSS9LM/welcome>.

- “F4FActions”
- “F4FApp” na página 124
- “F4FHandler” na página 126
- “TaintedParam” na página 127

F4FActions

```
java.lang.Object
  extended by com.ibm.appscan.frameworks.highlevelapi.F4FActions

public class F4FActions
  extends java.lang.Object
```

Class for specifying how the application's framework constructs should be modeled. An `F4FHandler` mutates the `F4FAction` object passed to `F4FHandler.handleApp(F4FApp, F4FActions)` as it analyzes the application.

Constructor Detail

F4FActions

```
public F4FActions()
```

Create an empty F4FActions object. Should not be needed for implementing a new framework handler, as the relevant F4FActions object will be passed to F4FHandler.handleApp(F4FApp, F4FActions).

addTaintedCallback

```
public void addTaintedCallback(IMethod method,  
                               int numParams)
```

Same as “addTaintedCallback” (String, int), but takes an IMethod directly rather than a VDB signature

addTaintedCallback

```
public void addTaintedCallback(java.lang.String vdbMethodSig,  
                               int numParams)
```

Make a method a tainted callback, with all parameters tainted.

Nota: For .NET apps, we need fully-qualified VDB signatures. So, instead of int as a parameter type, we need System.Int32, etc. To see the full mapping from fully-qualified names to the names usually used in VDB, see DotNetVDBUtil.systemName2VDBShortName.

Parameters:

- vdbMethodSig - the signature of the callback method
- numParams - the number of parameters for the callback method, including the this parameter

replaceCalls

```
public void replaceCalls(java.lang.String oldVDBSig,  
                        java.lang.String newVDBSig)
```

Replace all calls to one method with calls to another method. We require that the descriptors for the old and new method (i.e., the number of arguments, argument type, and return type) are identical.

Nota: replacement will only occur when oldVDBSig is the `_declared_` target at a call site. So, if oldVDBSig is `Integer.toString()`, and we see a call to `Object.toString()`, we will `_not_` perform a replacement at that call site, even though it might invoke `Integer.toString()`.

Nota: for .NET apps, we need fully-qualified VDB signatures. So, instead of int as a parameter type, we need System.Int32, etc. To see the full mapping from fully-qualified names to the names usually used in VDB, see DotNetVDBUtil.systemName2VDBShortName

Parameters:

- oldVDBSig - signature of method whose calls should be replaced
- newVDBSig - signature of method to replace calls with

replaceCallsWithSyntheticExpr

```
public void replaceCallsWithSyntheticExpr(java.lang.String vdbSig,  
    com.ibm.appscan.frameworks.specinfo.SyntheticExpr expr)
```

Replace all calls to a method with an arbitrary WAFL SyntheticExpr. For example, one could replace calls with an assignment via an AssignmentExpr.

Nota: replacement will only occur when oldVDBSig is the `_declared_` target at a call site. So, if oldVDBSig is `Integer.toString()`, and we see a call to `Object.toString()`, we will `_not_` perform a replacement at that call site, even though it might invoke `Integer.toString()`.

Nota: for .NET apps, we need fully-qualified VDB signatures. So, instead of `int` as a parameter type, we need `System.Int32`, etc. To see the full mapping from fully-qualified names to the names usually used in VDB, see `DotNetVDBUtil.systemName2VDBShortName`

Parameters:

- `vdbSig` - signature of method whose calls should be replaced
- `expr` - synthetic expression to replace calls with

replaceCallsWithParamPattern

```
public void replaceCallsWithParamPattern(java.lang.String oldVDBSig,  
    java.util.Map<java.lang.String,  
    java.util.Map<java.lang.Integer,  
    java.util.regex.Pattern>>  
    newSig2Pattern)
```

Replace calls to one method with calls to another method only if the parameters of String type are constants meeting specified patterns. We require that the descriptors for the old and new method (i.e., the number of arguments, argument type, and return type) are identical.

Nota: replacement will only occur when oldVDBSig is the `_declared_` target at a call site. So, if oldVDBSig is `Integer.toString()`, and we see a call to `Object.toString()`, we will `_not_` perform a replacement at that call site, even though it might invoke `Integer.toString()`.

Nota: for .NET apps, we need fully-qualified VDB signatures. So, instead of `int` as a parameter type, we need `System.Int32`, etc. To see the full mapping from fully-qualified names to the names usually used in VDB, see `DotNetVDBUtil.systemName2VDBShortName`

Parameters:

- `oldVDBSig` - signature of method whose calls should be replaced
- `newSig2Pattern` - maps VDB signature of each possible replacement method `m` to a map `M` from integer parameter positions to `Patterns`. If the string constant parameters in the appropriate positions match the patterns in `M` at some call site, a replacement to `m` will be performed.

addFrameworkInfo

```
public void addFrameworkInfo  
(com.ibm.appscan.frameworks.specinfo.IFrameworkInfo info)
```

Add arbitrary additional framework info. This method should only be needed for rare cases where the other APIs provided are insufficient.

addTaintedCallback

```
public void addTaintedCallback(java.lang.String vdbMethodSig,  
                               java.util.Collection<TaintedParam>  
                               taintedParams)
```

Make some method a tainted callback, with only certain parameter access paths being treated as tainted.

Nota: for .NET apps, we need fully-qualified VDB signatures. So, instead of `int` as a parameter type, we need `System.Int32`, etc. To see the full mapping from fully-qualified names to the names usually used in VDB, see `DotNetVDBUtil.systemName2VDBShortName`

Parameters:

- `vdbMethodSig` - the signature of the callback method, in VDB format
- `taintedParams` - information on which parameter access paths should be tainted

addHighLevelSyntheticMethod

```
public void addHighLevelSyntheticMethod(HighLevelSyntheticMethod m)
```

equivalent to `addHighLevelSyntheticMethod(m, true)`

addHighLevelSyntheticMethod

```
public void addHighLevelSyntheticMethod(HighLevelSyntheticMethod m,  
                                         boolean isEntrypoint)
```

Add a high-level synthetic method. A corresponding WAFL synthetic method (possibly an entrypoint) will be generated.

Parameters:

- `m` - the method
- `isEntrypoint` - should the method be marked as an entrypoint in WAFL?

createGlobal

```
public Global createGlobal(java.lang.String name,  
                           java.lang.String declaredVDBType,  
                           boolean isEntrypointScoped)
```

Create a new global that can be accessed from `HighLevelSyntheticMethods`.

Parameters:

- `name` - name for the global
- `declaredVDBType` - the declared type of the global (e.g., `java.lang.String`).

Nota: for .NET apps, we need a fully-qualified VDB type. So, instead of `int` as a parameter type, we need `System.Int32`, etc. To see the full mapping from fully-qualified names to the names usually used in VDB, see `DotNetVDBUtil.systemName2VDBShortName`

- `isEntrypointScoped` - if true, the global is scoped to a single entrypoint (i.e., it is request-scoped). Otherwise, the global is scoped across entrypoints (i.e., it is "session" or "application" scoped)

Returns:

- a Global object, which can be read/written inside a HighLevelSyntheticMethod

createGlobal

```
public Global createGlobal(java.lang.String name,  
                           IClass declaredClass,  
                           boolean isEntrypointScoped)
```

Just like “createGlobal” na página 122(String, String, boolean), but takes an IClass for the declared type instead of a type name

getGlobals

```
public java.util.Collection<Global> getGlobals()
```

For internal usage.

getAdditionalFrameworkInfo

```
public java.util.Collection  
<com.ibm.appscan.frameworks.specinfo.IFrameworkInfo>  
getAdditionalFrameworkInfo()
```

For internal usage.

getCallReplacement2SigsInfo

```
public java.util.Map  
<java.lang.String,java.util.Map  
<java.lang.String,java.util.Map  
<java.lang.Integer,java.util.regex.Pattern>>>  
getCallReplacement2SigsInfo()
```

For internal usage.

getCallReplacement2ExprInfo

```
public java.util.Map  
<java.lang.String,com.ibm.appscan.frameworks.specinfo.SyntheticExpr>  
getCallReplacement2ExprInfo()
```

For internal usage.

getCallback2TaintedParams

```
public java.util.Map  
<java.lang.String,java.util.Collection<TaintedParam>>  
getCallback2TaintedParams()
```

For internal usage.

getHighLevelSyntheticMethods

```
public java.util.List  
<com.ibm.wala.util.collections.Pair  
<HighLevelSyntheticMethod,java.lang.Boolean>>  
getHighLevelSyntheticMethods()
```

For internal usage.

toString

```
public java.lang.String toString()
```

Overrides:

- toString in class java.lang.Object

F4FApp

```
java.lang.Object  
  extended by com.ibm.appscan.frameworks.highlevelapi.F4FApp
```

```
public class F4FApp  
  extends java.lang.Object
```

Representation of an application, with methods to query various properties of classes, methods, etc. Implemented mostly by delegating to methods from the T.J. Watson Libraries for Analysis (WALA); the goal is to consolidate the most useful WALA methods in a single type. See the WALA home page (<http://wala.sourceforge.net>) for full details on WALA APIs.

Constructor Detail

```
public F4FApp(IClassHierarchy cha)
```

Should not be needed to implement a new handler. The relevant F4FApp object will be passed as a parameter to F4FHandler.handleApp(F4FApp, F4FActions)

getAppClass

```
@Deprecated  
public IClass getAppClass(java.lang.String vdbClassName)
```

Deprecated. Use getClass(String) instead; this method simply delegates to that one.

getClass

```
public IClass getClass(java.lang.String vdbClassName)
```

get the IClass for some class in the application, including library jars/DLLs. If no class with the provided name is found, return null

Parameters:

- vdbClassName - class name in VDB format, e.g., java.lang.String

getClassAnnotations

```
public java.util.Collection<Annotation>  
  getClassAnnotations(IClass klass)
```

Get the annotations/attributes for a class. For .NET, the result will include inherited attributes.

Parameters:

- klass - the class whose annotations are desired

getMethodAnnotations

```
public java.util.Collection<Annotation>  
  getMethodAnnotations(IMethod method)
```

Get the annotations / attributes for a method. For .NET, these will include inherited attributes.

Parameters:

- method - the method whose annotations are desired

getFieldAnnotations

```
public java.util.Collection<Annotation>  
getFieldAnnotations(IField field)
```

Get the annotations / attributes for a field.

Parameters:

- field - the field whose annotations are desired

getMethodParametersAnnotations

```
public java.util.Collection<Annotation>[]  
getMethodParametersAnnotations(IMethod method)
```

Get annotations on parameters as an array of Collections, where each array element gives the annotations on the corresponding parameter. Note that the this parameter for an instance method cannot have annotations.

Parameters:

- method - the method whose parameter annotations are desired

getAllApplicationClasses

```
public java.util.Collection<IClass>  
getAllApplicationClasses()
```

Get all the classes in the application (i.e., excluding those in library jars).

getClassHierarchy

```
public IClassHierarchy getClassHierarchy()
```

Get the WALA class hierarchy for the application. Most handlers should be able to work via the other methods in this class and shouldn't need to operate directly on the class hierarchy. But, access is provided for advanced use.

getMethodsDeclaredInClass

```
public java.util.Collection<IMethod>  
getMethodsDeclaredInClass(IClass klass)
```

Get all the static and instance methods declared in klass

getClassMethods

```
public java.util.Collection<IMethod>  
getClassMethods(java.lang.String className,  
                java.lang.String methodName)
```

Get all the methods in an class with a particular name. If the class cannot be found, returns an empty Collection.

Parameters:

- `className` - the class name in VDB (i.e., source-level) format, e.g., `java.lang.String`
- `methodName` -

getClassMethods

```
public java.util.Collection<IMethod>
getClassMethods(IClass appClass,
               java.lang.String methodName)
```

Get all the methods in an class with a particular name.

Parameters:

- `appClass` - the class
- `methodName` -

getStringConstantsReturnedByMethod

```
public java.util.Collection<java.lang.String>
getStringConstantsReturnedByMethod(IMethod method)
```

Get the possible String constants returned by the method. E.g., if the method has a statement `return "result";`, then "result" will be in the returned Collection. Throws an `IllegalArgumentException` if the return type of method is not String.

F4FHandler

```
java.lang.Object
com.ibm.appscan.frameworks.highlevelapi.F4FHandler
```

```
public abstract class F4FHandler
extends java.lang.Object
```

The abstract class that any new framework handler should extend.

Nota: Any sub-class of `F4FHandler` must have a no-argument constructor, for instantiation using reflection

Constructor Detail

```
F4FHandler
```

```
public F4FHandler()
```

handleApp

```
public abstract void handleApp(F4FApp app,
                              F4FActions actions)
```

Define what actions should be represented in the generated WAFL specification to handle the given application.

Parameters:

- `app` - the application to be analyzed
- `actions` - the actions to be taken; implementations of `handleApp(F4FApp, F4FActions)` should mutate this parameter and store the desired actions

isApplicable

```
public abstract boolean isApplicable()
```

Is the framework handler applicable for the target application? Implementations should check the language of the app, whether relevant configuration files are present, etc.

setFrameworksInput

```
public void setFrameworksInput(FrameworksInput input)
```

Should not be invoked by a framework handler.

getFrameworksInput

```
protected FrameworksInput getFrameworksInput()
```

Get the parsed representation of the input parameters to the frameworks code.

TaintedParam

```
java.lang.Object  
    extended by com.ibm.appscan.frameworks.highlevelapi.TaintedParam
```

```
public class TaintedParam  
    extends java.lang.Object
```

A type used to represent how some method parameter may reference tainted data.

Constructor Detail

```
public TaintedParam(int paramPos,  
                    java.lang.String accessPath)
```

Creates a new TaintedParam object for a particular parameter position and access path.

Parameters:

- paramPos - the position of the tainted parameter, numbered starting from 0 (where the this parameter of an instance method is parameter 0)
- accessPath - access path of fields from the parameter that should be tainted, e.g., "f.g". Use "" to indicate the parameter itself should be tainted.

getParamPos

```
public int getParamPos()
```

getAccessPath

```
public java.lang.String getAccessPath()
```

hashCode

```
public int hashCode()
```

Overrides:

- hashCode in class java.lang.Object

equals

```
public boolean equals(java.lang.Object obj)
```

Overrides:

- equals in class java.lang.Object

toString

```
public java.lang.String toString()
```

Overrides:

- toString in class java.lang.Object

Métodos Sintéticos de Alto Nível

Os métodos sintéticos são uma construção útil para modelar fluxo de dados avançado em estruturas. Por exemplo, muitas estruturas padrão (como Struts e Spring) encorajam uma arquitetura *model-view-controller* (MVC) para o aplicativo. Com uma estrutura MVC, o envio de formulário do cliente é frequentemente manipulado desta maneira:

1. Com base na URL, determine a classe M do *modelo* para manter dados do formulário enviado e a classe C do *controlador* contendo a lógica de negócios.
2. Crie um objeto de modelo M e configure suas propriedades com base nos dados do formulário (não confiável) na solicitação de HTTP. As propriedades são normalmente configuradas através de *setter* JavaBeans (por exemplo, setName() ou setAddress()).
3. Execute alguma validação nos dados no objeto M.
4. Crie um objeto do controlador C e passe o objeto M para um método C.execute() que executa a lógica de negócios. Normalmente, execute() retornará o nome de uma visualização para renderizar o resultado.
5. Com base no nome da visualização, determine o arquivo de visualização adequado (por exemplo, uma página JavaServer) a ser mostrado. Frequentemente, os dados no objeto M são passados para a visualização através de atributos da solicitação ou do objeto de sessão.

Todos os recursos acima podem ser modelados com os métodos sintéticos do Framework for Frameworks, expondo dessa maneira os comportamentos para análise pelo AppScan Source. A API do Framework for Frameworks fornece métodos sintéticos de alto nível para facilitar a geração de métodos sintéticos.

Nota: Os nós Rastreo com nomes de classe que iniciam com Appscan.Synthetic, Appscan.Synthetic.Validator e Appscan.Synthetic.Replacement correspondem aos métodos que são sintetizados por AppScan Source.

- Os métodos AppScan.Synthetic são usados para atar os rastreios no código do aplicativo que usa as estruturas.
- Um método AppScan.Synthetic.Validator modela a validação subjacente executada pelo tempo de execução da estrutura. É possível selecionar um método validador e marcá-lo como **Validator** se necessário.
- Um método AppScan.Synthetic.Replacement indica que um método no código do aplicativo foi substituído por AppScan Source para capturar o fluxo de dados entre os componentes de logout (como controladores e visualizações) da estrutura.

Formato VDB

Os métodos na API Framework for Frameworks requerem que Strings que representam nomes do tipo ou assinaturas de método estejam no formato VDB. Os nomes de tipo VDB são simplesmente nomes de nível de origem completos (por exemplo, java.lang.String) - ou, no caso de classes internas, representado com um símbolo de dólar (\$) (por exemplo,

`javax.swing.text.DefaultEditorKit$DefaultKeyTypedAction`). Para .NET, nomes abreviados como `int` e `string` devem ser evitados ao usar a API Framework for Frameworks. Em vez disso, use os nomes completos como `System.Int32` e `System.String`.

Uma assinatura de método VDB consiste nestas duas partes:

1. O nome do método, incluindo o nome completo da classe de fechamento (por exemplo, `java.lang.String.substring`).
2. Um descritor para o método, fornecendo os tipos de parâmetro e tipo de retorno. Os tipos de parâmetros ficam entre parênteses e separados por ponto-e-vírgula (;). O parêntese de fechamento é seguido por dois-pontos (:), e, em seguida, o tipo de retorno.

Uma assinatura de método VDB de exemplo é:

```
java.lang.String.substring(int;int):java.lang.String
```

Uma assinatura de método VDB de exemplo de uma classe interna é:

```
javax.swing.text.DefaultEditorKit$DefaultKeyTypedAction.  
actionPerformed(ActionEvent):void
```

Para a maioria dos métodos de API que requerem `String` no formato VDB, frequentemente há um método equivalente que usa um objeto `IClass` ou `IMethod`. Por exemplo, o método `F4FActions.addTaintedCallback(String, int)` requer que o primeiro parâmetro seja uma assinatura de método formatada para VDB - enquanto `F4FActions.addTaintedCallback(IMethod, int)` identifica o método com um objeto `IMethod`. Usar os métodos da API que utilizam os objetos `IClass` e `IMethod` pode ser mais conveniente por estas razões:

1. O usuário não precisa se preocupar sobre a formatação VDB para esses métodos porque isso é manipulado internamente.
2. Ao obter o objeto `IClass` ou `IMethod` (por exemplo, através de `F4FApp.getClassMethods()`), assegura-se que a classe ou o método correspondente exista realmente no código do aplicativo.

Usando Métodos Sintéticos de Alto Nível

Este tópico ilustra como usar alguns dos recursos-chave de métodos sintéticos de alto nível. Para obter detalhes completos, consulte a documentação Javadoc das classes e métodos Framework for Frameworks API.

- “Criando um Método Sintético de Alto Nível”
- “Criando Locais” na página 130
- “Incluindo Chamadas” na página 130
- “Incluindo um Valor de Retorno” na página 130
- “Globals” na página 131

Criando um Método Sintético de Alto Nível

Para criar um método sintético de alto nível e incluí-lo no objeto `actions` de `F4FActions`, use o código como:

```
HighLevelSyntheticMethod m = HighLevelSyntheticMethod.make();  
actions.addHighLevelSyntheticMethod(m);
```

Você pode especificar o nome, os tipos de parâmetros e o tipo de retorno do método sintético passando uma assinatura de método VDB para `HighLevelSyntheticMethod.make()`.

Criando Locais

Uma nova variável local para um método sintético de alto nível `m` pode ser criada desta maneira:

```
Local l = m.newLocal("java.lang.String");
```

Construtores não precisam ser chamados em métodos sintéticos. Fornecido o código acima, pode-se assumir que `l` refere-se a um objeto Sequência não-nulo ao incluir instruções adicionais no método sintético.

Incluindo Chamadas

A maioria das instruções nos métodos sintéticos de alto nível serão chamadas de métodos, incluídas através do método `addCall()`. Os parâmetros para `addCall()` representam o método a ser chamado, as informações de posição do arquivo para a chamada e os parâmetros a serem passados na chamada. Isso é um exemplo para incluir uma chamada em um método setter `sample.BeanType.setName()`, assumindo um objeto `app` de `F4FApp`:

```
Collection<IMethod> setterMethods =  
    app.getClassMethods("sample.BeanType", "setName");  
// assumir que há exatamente um método "setName" em BeanType  
IMethod setter = setterMethods.iterator().next();  
HighLevelSyntheticMethod m = HighLevelSyntheticMethod.make();  
Local l = m.newLocal("sample.BeanType");  
m.addCall(setter, null, l, Taint.taint());
```

As etapas 2–5 de manipulação de envio de formulário do cliente em uma arquitetura MVC (conforme descrito em “Métodos Sintéticos de Alto Nível” na página 128) podem ser modeladas para algum outro grau incluindo chamadas adequadas para um método sintético da maneira mostrada a seguir.

Os parâmetros para uma chamada são representados por objetos do tipo `Param`, que podem ser:

- Um objeto `Local`, representando um local
- Um objeto `Taint`, representando dados não confiáveis ou *contaminados*.
- Um objeto `EnclosingFormal`, representando um parâmetro formal do método sintético de alto nível. Por exemplo, se você tiver um método sintético com a assinatura `synthMethod(int):void`, `EnclosingFormal.FIRST` fará referência ao parâmetro `int` do método.
- Um objeto `Global` representando um global (discutido em “Globals” na página 131).

Nota: Para métodos de instância não estáticos, o valor a ser passado como `this` deve ser fornecido para `addCall()`. No exemplo acima, o valor em `l` será passado como `this` para `setName()`.

Incluindo um Valor de Retorno

Um método sintético pode retornar um valor usando o método `setReturnedValue()`. Valores de retorno podem ser úteis para gerar métodos de marcador para modelar validação de estrutura complexa. Por exemplo, se você estiver usando uma estrutura que executa validação complexa de um valor de solicitação de HTTP contaminado antes de passar para o método setter de um objeto de modelo, é possível expor a validação em rastreios descobertos pelo `AppScan Source` usando código como este:

```
String validatorSig =
    "Synth.validatorModel(java.lang.String):
    java.lang.String";
HighLevelSyntheticMethod validator =
    HighLevelSyntheticMethod.make(validatorSig);
// just return the parameter passed in
validator.setReturnedValue(EnclosingFormal.FIRST);
HighLevelSyntheticMethod m = ...;
Local validated = m.addCall(validatorSig, null, Taint.taint());
// now, validated can be passed to a setter
```

O método sintético `Synth.validatorModel()` simplesmente retorna a `String` que é passada a ele como um parâmetro. Em seguida, uma chamada para `Synth.validatorModel()` é incluída em outro método sintético, passando o valor contaminado como um argumento. O resultado dessa chamada é armazenado em `validated`, que pode ser passado em uma chamada subsequente para um método setter (conforme ilustrado pelo exemplo em “Incluindo Chamadas” na página 130). Rastreios envolvendo esses dados contaminados incluirão a chamada em `Synth.validatorModel()`, e um usuário do AppScan Source pode escolher filtrar os rastreios se a validação estiver resolvida o suficiente.

Globals

Globals são um recurso avançado que pode ser usado para expor fluxo entre partes desiguais de um aplicativo. Por exemplo, *globals* podem ser usados para modelar o fluxo de dados de um controlador para uma visualização através de atributos de uma solicitação ou sessão. As operações-chave para criar e acessar *globals* são:

- Para criar um objeto `Global` representando um *global*, use `F4FActions.createGlobal()`.
- Para gravar em um *global*, use `HighLevelSyntheticMethod.addGlobalWrite()`.
- Para ler de um *global*, passe o objeto `Global` como um parâmetro `Param` em uma chamada para `HighLevelSyntheticMethod.addCall()` - ou faça com que seja retornado de um método sintético através de `HighLevelSyntheticMethod.setReturnedValue()`.

Exemplo: Uma classe do controlador grava o primeiro nome de um usuário no atributo de solicitação `firstName` e, em seguida, a visualização lê esse atributo de solicitação e renderiza o valor para a resposta. Em um alto nível, esse fluxo poderia ser modelado como:

```
Global firstNameGlobal = actions.createGlobal
    ("firstName", "java.lang.String", true);
HighLevelSyntheticMethod controller = ...;
Local firstNameLocal = controller.newLocal("java.lang.String");
controller.addGlobalWrite(firstNameGlobal, firstNameLocal, null);
HighLevelSyntheticMethod view = ...;
view.addCall("Response.write(java.lang.String):void",
    null, firstNameGlobal);
```

Ao incluir uma gravação em `firstNameGlobal` no método sintético do controlador e, em seguida, passar `firstNameGlobal` para `Response.write()` no método sintético `view`, o fluxo de dados do controlador para a visualização é exposto.

Nota: Este exemplo contém muitas simplificações. Entre outras coisas, uma versão completa precisaria expor o fluxo de dados adequado para `firstNameLocal`.

Exemplo: Criação do Método Sintético

Este exemplo demonstra a criação de um método sintético. Neste método sintético, primeiro contaminamos um campo de classe e, em seguida, incluímos uma chamada a um método no código do usuário e finalmente passamos o resultado desta chamada de método para um sorvedouro.

- “Cenário de Aplicativo”
- “O Que é Modelado no Método Sintético”
- “Etapa 1: Criar um Método Sintético Vazio”
- “Etapa 2: Modelar uma Contaminação de um Campo de Classe” na página 133
- “Etapa 3: Modelar a Chamada do Método createUser()” na página 133
- “Etapa 4: Modelar Retorno de Dados para o Cliente” na página 134

Cenário de Aplicativo

Neste exemplo, o createUser é uma REST API implementada na estrutura JAX-RS. Um usuário pode chamar esta API usando uma URL como http://host:port/users/createUser. O tempo de execução da estrutura delega esta chamada para UserRestService.createUser() porque o caminho corresponde. Além disso, o tempo de execução da estrutura inicializa a variável de classe urlInfo a partir dos dados do cliente antes de chamar createUser(). Finalmente, os dados retornados por createUser() são enviados de volta para o cliente pelo tempo de execução.

O Que é Modelado no Método Sintético

Para capturar o “Cenário de Aplicativo” usando um método sintético, nós temos estes elementos:

- A contaminação da variável de classe urlInfo de UserRestService.
- A chamada para UserRestService.createUser().
- O retorno de dados para o cliente.

Esse é o código para o método createUser():

```
import java.io.BufferedWriter;

@Path("users")
public class UserRestService {
    @Context
    UriInfo urlInfo;

    @Path("/createUser")
    public User createUser(){
        MultivaluedMap<String, String> queryParams =
            urlInfo.getQueryParameters();
        String name = queryParams.getFirst("name");
        User user = new User(name);
        return user;
    }
}
```

Etapa 1: Criar um Método Sintético Vazio

```
22 public class JAXRSHandler extends F4FHandler{
23     @Override
        public void handleApp(F4FApp app, F4FActions actions) {
24         HighLevelSyntheticMethod synthMethod = HighLevelSyntheticMethod.make();
```

- O objeto do método sintético é inicializado na linha 24.

Etapa 2: Modelar uma Contaminação de um Campo de Classe

O fragmento de código abaixo demonstra como os campos de classe específica podem ser contaminados. Neste exemplo, desejamos contaminar os campos de classe que estão indicados com a anotação @Context.

```
27 // create a local variable of the appropriate type
28 Local userRestServiceClazzInstance =
    synthMethod.newLocal("com.ibm.appscan.UserRestService");
29
30 // get all the class fields
31 for(IField field: app.getIClass
    ("com.ibm.appscan.UserRestService").getDeclaredInstanceFields()){
32
33     //get all the annotations associated with the field
34     Collection<Annotation> fieldAnnotations = app.getFieldAnnotations(field);
35
36     //for each annotation of the field check if it is an @Context annotation
37     for (Annotation annotation : fieldAnnotations) {
38
39         if (annotation.getType().getName().toString().equals
            ("Ljavax/ws/rs/core/Context")) {
40
41             // call the F4F API to assign taint to the field
42             synthMethod.addInstanceVariableWrite(
43                 userRestServiceClazzInstance /*Variable representing
44                 class instance*/,
45                 field /*field to taint */,
46                 Taint.taint() /* taint */,
47                 null);
48         }
49     }
50 }
```

A Framework for Frameworks API `addInstanceVariableWrite` executa quatro argumentos:

1. O primeiro argumento é a referência de classe cujo campo desejamos contaminar. No exemplo, a variável local `userRestServiceClazzInstance` se refere a este argumento.
2. O segundo argumento é o campo de classe que desejamos contaminar.
3. O terceiro argumento é o novo valor que será designado ao campo da variável de classe. No exemplo, desejamos contaminar esta variável para que um `Taint.taint()` seja passado.
4. O último argumento é `FilePositionInfo`, que é `null` neste exemplo.

Etapa 3: Modelar a Chamada do Método `createUser()`

Nesta etapa, simulamos a chamada em nosso método sintético.

```
50 // call createUser() and store the returned value to a local variable
51 Local returnObj = synthMethod.addCall(
52     "com.ibm.appscan.UserRestService.createUser():com.ibm.appscan.User"
53     /* signature of the method to be called */,
54     null, /* FilePositionInfo */
55     userRestServiceClazzInstance /* Object on which the method
56     will be called */);
```

A API de alto nível `addCall` executa três argumentos:

- O primeiro argumento é a assinatura do método que desejamos chamar. Nesse caso, desejamos chamar `User.createUser()`, portanto, sua assinatura é usada.

- O segundo argumento é `FilePositionInfo`. Ele é passado como `null` porque não é necessário para este exemplo.
- O último argumento representa a lista de parâmetros que são necessários para chamar o método `createUser()`.

Uma vez que `createUser()` não executa argumento, o único argumento passado para essa chamada é o objeto `this`, uma variável `Local` (`userRestServiceClazzInstance`) do tipo `User`. O método `createUser()` retorna um `User` recém-criado que é armazenado em uma nova variável `Local` denominada `returnedObj` (conforme mostrado na linha 51).

Etapa 4: Modelar Retorno de Dados para o Cliente

Nesta etapa final, criamos um sorvedouro para modelar o retorno de dados de volta para o cliente.

```

58 // create a PrintWriter object
59 Local printWriterObj = synthMethod.newLocal("java.io.PrintWriter");
60
61 // we want to call the print API on the PrintWriter object.
62 // The print API takes a single argument of Object type returns void
63
64 // we create a param list of size 2. The first item in this list is always the
65 // 'this' object and rest are actual method arguments.
66 Param[] printWriterObjParam = new Param[2];
67 printWriterObjParam[0] = printWriterObj; // The "this" object
68 // the value returned by the call to the createUser method
69 printWriterObjParam[1] = returnObj;
70
71 // Now add a call to the print method
72 synthMethod.addCall("java.io.PrintWriter.print(java.lang.Object):void",
73 null, printWriterObjParam);
74 }

```

- Na linha 59, nós criamos um objeto `Local` para a classe `PrintWriter` no método sintético.
- Nas linhas 64 a 66, preparamos a lista de parâmetros que é necessária para chamar o método `print` na instância da classe `PrintWriter` (`printWriterObj`):
 - O primeiro parâmetro é a referência do objeto (`printWriterObj`), sozinha.
 - O segundo parâmetro é o argumento para o método `print`. Desejamos passar o valor de retorno que é armazenado na variável `Local` `returnObj`.
- Na linha 69, incluímos uma chamada para o método `PrintWriter.print` para modelar o retorno dos dados de volta para o cliente.

Integrando um novo manipulador Framework for Frameworks ao Scanner Customizado da Web Service Description Language Existente

Sobre Esta Tarefa

Ao varrer um aplicativo JAX-RS ou JAX-WS com o AppScan Source, deverá assegurar-se de poder capturar todos os pontos de entrada de serviço da web, para que você não perca vulnerabilidade no aplicativo. Este documento descreve uma técnica para fazer isso. Ele descreve as etapas que deverão ser realizadas para integrar um novo manipulador Framework for Frameworks (F4F) usado para analisar estruturas de serviço da web não suportadas com o scanner customizado do Web Service Description Language (WSDL).

Este trabalho é necessário, para que o scanner customizado da WSDL seja capaz de diferenciar entre aqueles pontos de entrada de serviço da web já descobertos por esse novo manipulador F4F e aqueles pontos de entrada de serviço da web que existem no arquivo WSDL. Com estas informações, o scanner customizado da WSDL eliminará as provas de configuração que foram analisadas por este novo manipulador F4F.

- “Conectando o Manipulador F4F de Serviço da Web ao Scanner Customizado da WSDL”
- “Mapeamento de Assinatura” na página 136

Conectando o Manipulador F4F de Serviço da Web ao Scanner Customizado da WSDL

Antes de Iniciar

Esta seção supõe que o manipulador F4F principal e a geração de método sintético já foram desenvolvidos e testados.

Procedimento

1. Inclua uma dependência do plug-in em `com.ibm.appscan.frameworks.wsdl.util`, que inclui as APIs Auxiliares para relatar os serviços localizados. Ela também existe dentro da pasta `walalib`.
2. No manipulador principal (que estende a classe `F4FHandler`), no início do método `handleApp()`, inclua estas linhas:

```
String filePath = WSDLReportingUtil.getServicesFilePath
    (getFrameworksInput().getFileLocs());
if (filePath!=null) {
WSDLReportingUtil.initXmlDocument(filePath);
}
```

Essas linhas criam um arquivo temporário que será usado pelo Scanner Customizado da WSDL.
3. No término do método `handleApp()`, inclua esta linha para assegurar que todos os serviços relatados sejam salvos antes que o Manipulador F4F saia:

```
WSDLReportingUtil.saveXmlDocument();
```
4. Para relatar um serviço, inclua uma chamada em uma dessas APIs a partir da API `WSDLReportingUtil`:
 - `reportService(String targetNameSpace, String serviceName, String methodSignature)`
 - `reportService(String targetNamespace, String serviceName, String operationName, ArrayList<String> wsdlParams)`em que:
 - `targetNameSpace` é o namespace de destino do serviço. Em JAX-WS, por exemplo, está localizado nas anotações ou corresponde ao nome do pacote.
 - `serviceName` é o nome do serviço. Este pode ser localizado nas anotações ou pode corresponder ao nome da classe de serviço.
 - `methodSignature` é a assinatura de método do arquivo WSDL com alguma simplificação.
 - `operationName` é o nome do método, que é vinculado a um nome de operação em um Tipo de Porta no arquivo WSDL.
 - `wsdlParams` é a lista de tipos de parâmetros em um formato simples da WSDL.

5. Atualmente o Scanner Customizado da WSDL funciona como um scanner independente ou como parte de uma varredura do Java/JSP. Se o novo Manipulador F4F de Serviço da Web for projetado para trabalhar com uma linguagem diferente da do Java (por exemplo, .NET), certifique-se de que o Scanner Customizado da WSDL seja chamado com o tipo de varredura adequada seguindo estas etapas:
 - a. Acesse o diretório <data_dir>/ltd (em que <data_dir> é o local dos dados do programa do AppScan Source, conforme descrito em Capítulo 9, “Locais de Arquivos de Instalação e de Dados do Usuário”, na página 149).
 - b. Faça uma cópia de backup de todos os arquivos, para evitar quaisquer problemas inesperados.
 - c. Abra o arquivo .ltd que corresponde ao tipo de varredura adequada. Por exemplo, para C++, abra o arquivo cpp.ltd.
 - d. Inclua esta linha antes da tag de fechamento <LanguageTypeDefinition>:
`<Scanner name="wsdl"/>`
 - e. Copie o valor file_extention_set_name, que será necessário para a próxima etapa.
 - f. Abra o file_extensions.xml e localize o FileExtensionSet que possui o nome que foi copiado na etapa anterior.
 - g. Inclua esta linha neste ponto no arquivo:
`<FileExtension extension="wsdl" assess="true"/>`
 - h. Salve todos os arquivos e reinicie o AppScan Source. O Scanner Customizado da WSDL será executado agora como parte da varredura integral do aplicativo.

Resultados

Neste ponto, o scanner customizado da WSDL apenas mostrará as provas de configuração (na visualização Provas) dos arquivos WSDL que não são implementadas pela estrutura. Os serviços dos arquivos WSDL que já estão implementados no aplicativo não aparecerão como uma prova de configuração. Estes serão analisados pelo novo manipulador F4F de serviços da web.

Mapeamento de Assinatura

Sobre Esta Tarefa

O scanner customizado da WSDL tentará corresponder aos métodos relatados aos Serviços da WSDL correspondente, Tipos de Portas WSDL e Operações da WSDL. Este é o motivo pelo qual os serviços da web são relatados no formato WSDL. Algumas simplificações foram incluídas para facilitar o entendimento das coisas:

- Para os tipos primitivos de XSD, não há necessidade de incluir o namespace de destino. Em vez disso, apenas use o nome (por exemplo, string, int, float, double).
- Para coleções e matrizes, [] é incluído após o tipo (por exemplo, string[], int[]).
- Para os tipos de classe que serão mapeados para os tipos complexos de XSD (usando JAX-B ou quaisquer outras tecnologias), o tipo aparecerá como isto:
`http://my-types/myxsdnamespace/:Customer`

O namespace deverá estar disponível como parte da anotação Java. Se ele não fizer parte da anotação Java, deverá corresponder ao namespace de serviço do responsável pela chamada ou o nome do pacote, dependendo da implementação da estrutura.

Nota: Os erros de mapeamento provavelmente impedirão o scanner customizado da WSDL de corresponder a API de serviço da web relatada à operação associada da WSDL. Se isso ocorrer, uma prova de configuração extra aparecerá. Essa prova extra deverá ser considerada um positivo falso e indicativo de um problema de mapeamento.

Se você não tiver certeza de como a assinatura esperada deve se parecer, execute uma varredura WSDL do AppScan Source nos próprios arquivos WSDL. Isso gerará provas para todas as operações nos serviços WSDL. Clicar em cada prova exibirá a assinatura de método na visualização Detalhe das Provas.

Capítulo 8. Mensagens de Erro do Componente do Cliente AppScan Source

CRWSA0900E O Pacote Configurável Excluído Não Pode Ser Renomeado

O Pacote Configurável Excluído é um pacote configurável integrado que contém todas as provas excluídas. Ele não pode ser renomeado.

CRWSA0907E Falha ao incluir <> porque ele já existe no banco de dados e não pode ser sobrescrito

A regra customizada já existe no banco de dados AppScan Source e não pode ser recriada ou sobrescrita. Isso ocorre em uma destas situações:

- A regra que foi enviada no banco de dados está marcada como um sorvedouro - e você está tentando criar uma regra customizada que marca o sorvedouro como propagação de contaminação.
- A regra que foi enviada no banco de dados está marcada como um propagador de contaminação - e você está tentando criar uma regra customizada que altera essa regra, incluindo a propagação de contaminação.

CRWSA0920E Houve Erros ao Importar a Assinatura no Projeto

Verifique se o projeto está configurado adequadamente antes de importar assinaturas.

CRWSA0925E O Valor Inserido para "{0}" Não É Válido.

O valor inserido não está em um formato válido. Insira um formato válido.

CRWSA0930E Formato de Data Inválido Inserido para "{0}".

CRWSA0935E Formato de Número Inválido Inserido para "{0}".

CRWSA0940E Usuário Desconhecido

Certifique-se de que o usuário exista no Enterprise Server e de que as informações tenham sido inseridas corretamente.

CRWSA0945E O ID do Usuário Especificado É Muito Curto. Os IDs do Usuário Devem Ter pelo menos um Caractere de Comprimento.

CRWSA0950E A Senha Especificada É Muito Curta. As Senhas Devem Ter pelo menos 6 Caracteres de Comprimento.

CRWSA0955E O ID do Usuário Especificado É Muito Longo. Os IDs do Usuário Não Podem Exceder 255 Caracteres de Comprimento.

CRWSA0960E O ID do Usuário Especificado Contem Caracteres Inválidos.

CRWSA0965E A Senha Especificada É Muito Longa. As Senhas Não Podem Exceder 16 Caracteres de Comprimento.

CRWSA0970E As Senhas Especificadas Não Correspondem. Insira as Senhas Novamente.

CRWSA0975E O Usuário Admin Não Pode Ser Modificado.

CRWSA0980E O Endereço de Email Especificado É Inválido. - O Formato Necessário É <user>@<domain>, em que <domain> Não Pode Exceder 255 Caracteres.

CRWSA0985E O 'Email' É Necessário.

CRWSA0990E O 'Nome' É Necessário.

CRWSA0995E O ID do Usuário Especificado Existe no Repositório, os Campos 'Nome' e 'Email' Devem Ser Atualizados.

CRWSA1010W O Usuário {0} Não Pôde Ser Incluído.

Consulte a ajuda do AppScan Enterprise Server para obter informações.

CRWSA1050E Erro Inesperado ao Carregar o Editor Externo.

Certifique-se de que o editor externo esteja disponível e funcionando corretamente.

CRWSA1055E Erro ao Carregar o Editor Externo. O Arquivo de Propriedades Está Corrompido.

Certifique-se de que o editor externo esteja disponível e funcionando corretamente.

CRWSA1060E Erro ao Carregar o Editor Externo. O Arquivo de Propriedades Está Ausente.

Certifique-se de que o editor externo esteja disponível e funcionando corretamente.

CRWSA1065E Erro ao Carregar o Editor Externo. Não É Possível Carregar o Editor Preferencial

Certifique-se de que o editor externo esteja disponível e funcionando corretamente.

CRWSA1070E Erro ao Carregar o Editor Externo. O Plug-in do Editor Retornou uma Falha.

Certifique-se de que o editor externo esteja disponível e funcionando corretamente.

CRWSA1075E Erro ao Carregar o Editor Externo. Não É Possível Criar a Instância Executável.

Certifique-se de que o editor externo esteja disponível e funcionando corretamente.

CRWSA1080E Erro ao Carregar o Eclipse. A Conexão Falhou.

Certifique-se de que o Eclipse tenha sido configurado adequadamente para ser usado como editor externo (use um Eclipse que contenha

com.ouncelabs.core.filelauncher.jar em seu diretório plugins\).

CRWSA1085E Erro ao Carregar o Eclipse. Sem Resposta do Servidor.

Certifique-se de que o Eclipse tenha sido configurado adequadamente para ser usado como editor externo (use um Eclipse que contenha com.ouncelabs.core.filelauncher.jar em seu diretório plugins\).

CRWSA1090E Erro ao Carregar o Eclipse. Falha do Protocolo.

CRWSA1095E Erro ao Carregar o Eclipse. O Caminho para o Eclipse É Inválido.

Certifique-se de que o Eclipse tenha sido configurado adequadamente para ser usado como editor externo (use um Eclipse que contenha com.ouncelabs.core.filelauncher.jar em seu diretório plugins\).

CRWSA1100E Erro ao Carregar o Editor Customizado. Não É Possível Executar {0}

CRWSA1120E Varredura com Falha

CRWSA1125E Ocorreu um Erro ao Recuperar os Resultados da Varredura

CRWSA1150W Aviso: Marcar "{0}" como Rotina de Validação Terá Efeito na Descoberta Atual ou no Gráfico de Rastreamento. Tem Certeza de que Deseja Incluir o Nó Raiz como Rotina de Validação?

CRWSA1155E Pelo menos uma Origem, Dissipador, Propriedade de Origem ou Propriedade de Dissipador É Necessária.

CRWSA1160E Não É Possível Ativar o Assistente de Regras Customizadas enquanto uma Avaliação Está em Execução.

CRWSA1165E Erro ao Incluir Regras Customizadas

CRWSA1170E Falha ao Incluir {0}.

CRWSA1175E Falha ao Incluir {0} porque Ele Já Existe no Banco de Dados e Não Pode Ser Sobrescrito.

CRWSA1185E Validação com Falha devido a um Erro na Compilação

CRWSA1190E Compilação com Falha

CRWSA1195E Ocorreu um Erro ao Tentar Abrir um Editor.

Quando o editor falha ao abrir, tente reiniciar o AppScan Source.

CRWSA1200E O Arquivo Selecionado Não Foi Localizado neste Sistema.

CRWSA1205E Ocorreu um Erro ao Criar Marcadores de Código Fonte para a Avaliação.

CRWSA1210E Um ou mais Arquivos de Origem Não Puderam ser Localizados para o Recurso de Fragmento de Código Fonte. Deseja que Seja Solicitado o Local dos Arquivos Ausentes?

CRWSA1215E Falha ao Tentar se Conectar a AppScan Enterprise Console. - Verifique se o Servidor Está em Execução no Local Especificado.

CRWSA1220E O Login Atinge o Tempo Limite após {0,número} Segundos - É Recomendado Reiniciar os Serviços do AppScan Enterprise Console.

CRWSA1225E Erro ao Armazenar Filtros Não Salvos. Mudanças nos Filtros Não Serão Salvas.

CRWSA1230E Não É Possível Salvar a Avaliação durante a Varredura.

CRWSA1240E Aplicativo Não Localizado

CRWSA1245E Não Foi Possível Localizar {0}

Verifique o local do JRE especificado.

CRWSA1250E Não Foi Possível Localizar o Plug-in startup.jar ou org.eclipse.equinox.launcher

Verifique a instalação baseada em Eclipse especificada.

CRWSA1290W Um JDK Estava Ausente em um ou mais Projetos. O JDK Padrão Foi Configurado para Esses Projetos.

O JDK especificado de um projeto importado não foi reconhecido pelo AppScan Source. Para corrigir isso, inclua um JDK reconhecido usando a página de preferências do Java e JSP.

CRWSA1295W Um JDK Estava Ausente no Projeto. O JDK Padrão Foi Configurado para Esse Projeto.

O JDK especificado de um projeto importado não foi reconhecido pelo AppScan Source. Para corrigir isso, inclua um JDK reconhecido usando a página de preferências do Java e JSP.

CRWSA1300E Um JDK Está Ausente em um ou mais Projetos e Não Há Padrão Especificado. Será Solicitada a Escolha do JDK Padrão após Clicar no Botão OK. No Futuro, Esse JDK Será Usado para Todos os Projetos que Não Configuraram um JDK Explicitamente.

CRWSA1305E O Aplicativo É Somente Leitura. Mudanças Feitas na Avaliação não Persistirão em Varreduras Futuras.

É possível que o aplicativo esteja remoto ou que ele precise ser salvo. Em qualquer um dos casos, as mudanças feitas na avaliação não estarão disponíveis em varreduras futuras.

CRWSA1340E "{0}" Falhou ao se Registrar

CRWSA1345E O Arquivo do Aplicativo Não É Gravável.

CRWSA1350E Erro ao Excluir Entradas

CRWSA1355E Não É Possível Excluir um Aplicativo enquanto uma Avaliação Está Sendo Executada.

CRWSA1360E Não É Possível Excluir um Projeto enquanto uma Avaliação Está Sendo Executada.

CRWSA1365W Quaisquer Projetos nos quais Você Deseja que as Mudanças Sejam Aplicadas Devem Ser Atualizados.

CRWSA1370W Algumas das Provas Seleccionadas Possuem Notas Existentes. Tem Certeza de que Deseja Sobrescrevê-las?

CRWSA1375W Falha ao Exportar para o Arquivo

CRWSA1380W Sem Provas para Exportar. Todas as Provas Seleccionadas Estão Ausentes nos Resultados da Avaliação.

CRWSA1383E Sem Provas para Submeter. Todas as Provas Seleccionadas Foram Submetidas e a Preferência "Submeter Defeitos apenas uma Vez" Está Configurada.

CRWSA1385W Estes Campos Devem Ser Compilados para Submeter um Defeito:

CRWSA1395W Pacote Configurável Salvo com Sucesso em {0}.

CRWSA1400E O Arquivo de Projeto "{0}" Não Pôde Ser Localizado.

Verifique se o arquivo de projeto do AppScan Source existe.

CRWSA1405E O Diretório do ASP.NET "{0}" Não Pôde Ser Localizado

Verifique se o diretório do ASP.NET existe.

CRWSA1410E Erro ao Importar o Arquivo de Projeto "{0}"

CRWSA1415E Erro ao Importar o Diretório do ASP.NET "{0}"

CRWSA1420E Erro ao Criar o Projeto

CRWSA1425E O Diretório Ativo "{0}" Não Existe. Selecione um Diretório Válido.

CRWSA1430E A Raiz de Contexto da Web "{0}" Não Existe. Selecione um Diretório ou Arquivo WAR Válido.

CRWSA1435E A Raiz de Contexto da Web "{0}" Não Pode Ser um Diretório-pai no mesmo Diretório que o Diretório de Projeto "{1}".

CRWSA1440E A Raiz de Conteúdo "{0}" Não Existe. Selecione um Diretório Válido.

CRWSA1445E O Arquivo ou Diretório "{0}" Não Existe. Selecione um Arquivo ou Diretório Válido.

CRWSA1450E A Raiz de Conteúdo "{0}" Não Pode Ser um Diretório-pai ou o mesmo Diretório do Diretório de Projeto "{1}".

CRWSA1455E O Método {0} já Possui um ActionObject para o Tipo {1}.

CRWSA1460E "{0}" Não É um Diretório Válido

CRWSA1465E JDK Ausente para o Projeto

CRWSA1470E Este Projeto Java/JSP Faz Referência a um JDK Inválido ({0}). Um JDK Padrão Será Fornecido.

CRWSA1475E O Nome de JDK já Existe

CRWSA1480E Um JDK Denominado "{0}" já Existe.

Especifique um nome de JDK exclusivo.

CRWSA1485E O Nome de JDK Deve Ter pelo menos um Caractere.

CRWSA1490E Um JDK Deve Ser Selecionado para a Operação

Um JDK Existente Deve Ser Selecionado.

CRWSA1495E Não É Possível Salvar

CRWSA1500E Não É Possível Salvar enquanto uma Avaliação Está em Execução

CRWSA1505E A Raiz de Origem "{0}" já Existe como uma Raiz.

CRWSA1510E O Diretório "{0}" Não Existe. Selecione um Diretório Válido.

CRWSA1515E O Diretório Pré-compilado "{0}" Não Existe. Selecione um Diretório Válido.

CRWSA1520E A Classe {0} Existe na Dependência {1} e no Caminho de Origem.

Verifique as dependências especificadas.

CRWSA1525E Tipo de Arquivo Inesperado no Caminho da Classe ({0}).

Certifique-se de que seu projeto Java contenha apenas arquivos .jar, .class ou .java.

CRWSA1530E Os Projetos Não Podem Ser Validados enquanto uma Varredura Está em Execução.

CRWSA1535E

CRWSA1540E O Nome de Configuração Está Vazio

CRWSA1545E Raiz de Origem Inválida

CRWSA1550E A Raiz de Origem "{0}" É Inválida. - Raízes de Origem que São Arquivos Devem Estar sob o Diretório Ativo.

CRWSA1555E Uma Raiz de Contexto da Web Válida Não Pôde Ser Localizada.

CRWSA1565E Rotina de Codificação Criada com Sucesso para {0}

CRWSA1568E Não Foi Possível Resolver as Variáveis no Caminho "{0}"

Inclua a variável do caminho usando a página de preferências Alterar Variáveis.

CRWSA1570E Nome do Atributo Inválido {0}

CRWSA1575E Valor do Atributo Inválido {0}

CRWSA1580E Você Deve Criar um Atributo antes de Incluir Valores.

CRWSA1585E O Aplicativo Deve Ter um Nome Não Vazio.

CRWSA1590E Entrada Duplicada "{0}". Especifique um Valor Exclusivo.

CRWSA1595E O Nome Não Pode Estar Vazio.

CRWSA1600E Uma Visualização com Provas Deve Ser Seleccionada para Ser Gerado um Relatório.

CRWSA1625E Acesso Negado. A Conta Atual Não Tem Permissão para {0}.

CRWSA1630E Ocorreu um Erro ao Tentar Importar a Área de Trabalho. Certifique-se de que o Importador Esteja Configurado Adequadamente em Preferências > Importadores de Área de Trabalho do Eclipse .

CRWSA1640E Não É Possível Incluir o Projeto. {0} É Referência a um Aplicativo Somente Leitura.

O aplicativo é somente para leitura - ou é uma referência para uma solução da área de trabalho do Eclipse ou do Microsoft Visual Studio. Projetos não podem ser incluídos.

CRWSA1650E Não É Possível Localizar a Folha de Estilo {0}.

Erro CRWSA1653E

Esse erro é seguido por dois pontos (:) e uma mensagem que descreve o erro. Estes documentos fornecem uma possível resolução para algumas destas mensagens:

- Tenta gerar um relatório usando o resultado da integração do AppScan Source Maven no erro "Sintaxe incorreta para opção de relatório".
- Executar `ounceauto scanapplication` resulta em "CRWSA1653E: deve-se efetuar login primeiro"
- Resultados da varredura em "inicializando applicationmanager do gerenciador" no AppScan Source for Security
- Publicar a avaliação no AppScan Enterprise usando o AppScan Source CLI falha com o código de erro CRWSA1653E
- Publicar avaliação a partir da CLI resulta no erro CRWSA1653E

CRWSA1655E Um Nome é Necessário para o Padrão

CRWSA1660E Pelo menos um Critério Deve Ser Especificado para o Padrão

CRWSA1665E Os Critérios Não Podem Estar Vazios.

CRWSA1670E O Arquivo É Necessário para o Padrão.

CRWSA1675E Uma Regra de Varredura do Nome "{0}" já Existe. Escolha um Nome Exclusivo.

CRWSA1700E O Layout do Relatório Atualmente Está Vazio. Inclua Elementos do Relatório na Guia Layout do Relatório.

CRWSA1705E Não Há Pacotes Configuráveis para Exportar

CRWSA1710E Não Há Avaliação Aberta para Exportar

CRWSA1715E A Senha Inserida nos Campos Nova Senha e Confirmar Nova Senha Não Correspondem

CRWSA1725E A Senha É Muito Curta. Insira uma Senha com 6 a 16 Caracteres.

CRWSA1730E A Senha É Muito Longa. Insira uma Senha com 6 a 16 Caracteres.

CRWSA1735E O Arquivo de Configuração "{0}" Não Possui Nome de Exibição e Será Ignorado.

CRWSA1736E Não foi possível abrir o arquivo <filename>. Certifique-se de que o usuário com o qual o servidor de automação está sendo executado tenha permissões ao arquivo

CRWSA1737E Não foi possível abrir o arquivo <filename>. Certifique-se de que o usuário com o qual o servidor de automação está sendo executado tenha permissões ao arquivo e que seja um aplicativo local

CRWSA1740E O Arquivo de Configuração Denominado "{0}" já Está Aberto.

CRWSA1745E O Arquivo de Configuração "{0}" Não Está Visível ao Usuário.

CRWSA1750I Esta Guia Contém Algumas Configurações que Requerem a Reinicialização dos Serviços do AppScan Source para que as Mudanças Entrem em Vigor. Antes de Reiniciar os Serviços do AppScan Source, Escolha Todos os Aplicativos Clientes do AppScan Source em Execução ou Conectados a Este Computador. Deseja Reiniciar os Serviços Agora?

CRWSA1755E "{0}" Possui uma Configuração em que Estão Ausentes um ou mais Atributos Necessários (Nome, Nome_de_Exibição ou Tipo). Este Arquivo Não Será Carregado."

CRWSA1760E Ocorreu um Erro de Licenciamento:

CRWSA1775E A Conexão do AppScan Enterprise Console Falhou ao Inicializar. Verifique a Disponibilidade do AppScan Enterprise Console com Essas Configurações em um Navegador Externo.

CRWSA1780E A Conexão do AppScan Enterprise Console Falhou ao Inicializar. Verifique os Logs e/ou a Disponibilidade do AppScan Enterprise Console com Essas Configurações em um Navegador Externo. - Detalhes do Erro: - {0}

CRWSA1790E Falha ao Publicar a Avaliação no AppScan Enterprise Console. Verifique os Logs e/ou Entre em Contato com o Administrador do Sistema.

CRWSA1795E Falha ao Inicializar o Serviço de Publicação. Verifique as Configurações de Conexão do AppScan Enterprise Console.

CRWSA1800E Falha ao Inicializar o Serviço de Publicação. Verifique a Disponibilidade do AppScan Enterprise Console e sua Capacidade de Efetuar Logon.

CRWSA1805E Não É Possível Efetuar o Download do Arquivo. - Ocorreu um Erro ao Tentar se Conectar ao Site.

Certifique-se de que haja uma conexão com a Internet ativa e de que o site esteja disponível.

CRWSA1810E O Login é Necessário antes de o Pacote Configurável Criado no Produto da versão 7.0.0 Possa Ser Aberto

CRWSA9999E O AppScan Source for Analysis Encontrou um Erro Crítico e Deve Ser Fechado.

Capítulo 9. Locais de Arquivos de Instalação e de Dados do Usuário

Ao instalar o AppScan Source, os arquivos de dados do usuário e de configuração são armazenados fora do diretório de instalação.

- “Local de instalação padrão”
- “Diretório de Dados Padrão do AppScan Source”
- “Local do Arquivo Temporário do AppScan Source” na página 150

Local de instalação padrão

Quando o AppScan Source é instalado, o software é colocado em um destes locais padrão:

- Versões de 32 bits do Microsoft Windows:
<SYSTEMDRIVE>:\Program Files\IBM\AppScanSource
- Versões de 64 bits do Microsoft Windows:
<SYSTEMDRIVE>:\Program Files (x86)\IBM\AppScanSource
- Linux: Se você for o usuário padrão, o Assistente de Instalação instala o software em /opt/ibm/appscansource. Se você não for o usuário raiz, será possível instalar o plug-in do EclipseAppScan Source for Development - que, por padrão, é instalado no <home_directory>/AppScan_Source.
- macOS: /Applications/AppScanSource.app

Importante:

- O nome do diretório de instalação pode conter somente caracteres em inglês. Não são permitidas pastas que contêm caracteres que não estejam em inglês.
- Se estiver instalando no Windows, você deverá ter privilégios de administrador para instalar componentes do AppScan Source.
- Se estiver instalando no Linux, você deverá ter privilégios de administrador para instalar os componentes do servidor do AppScan Source.

Diretório de Dados Padrão do AppScan Source

Os dados do AppScan Source são compostos de itens como arquivos de configuração, amostra e log. Quando o AppScan Source é instalado, os arquivos de dados são colocados, por padrão, nestes locais:

- Microsoft Windows: <SYSTEMDRIVE>:\ProgramData\IBM\AppScanSource

Nota: ProgramData\ é uma pasta oculta e, para vê-la, você deve modificar as preferências de visualização no **Explorer** para mostrar arquivos e pastas ocultos.

- Linux: /var/opt/ibm/appscansource
- macOS: /Users/Shared/AppScanSource

Para saber como alterar o local do diretório de dados do AppScan Source, consulte “Alterando o Diretório de Dados do AppScan Source” na página 150.

Local do Arquivo Temporário do AppScan Source

Algumas operações do AppScan Source resultam na criação de arquivos temporários, que são armazenados, por padrão, nestes locais:

- Microsoft Windows: <SYSTEMDRIVE>:\ProgramData\IBM\AppScanSource\temp

Nota: ProgramData\ é uma pasta oculta e, para vê-la, você deve modificar as preferências de visualização no **Explorer** para mostrar arquivos e pastas ocultos.

- Linux: /var/opt/ibm/appscansource/temp
- macOS: /Users/Shared/AppScanSource/temp

O local do arquivo temporário está localizado em um diretório temp no diretório de dados do AppScan Source. É possível alterar o local do arquivo temporário alterando o diretório de dados, conforme descrito em “Alterando o Diretório de Dados do AppScan Source”. Isso fará com que o temp seja localizado no diretório de dados escolhido.

Alterando o Diretório de Dados do AppScan Source

É possível que queira alterar o local do diretório de dados AppScan Source com o propósito de gerenciar o espaço em disco rígido. É possível alterar o local após a instalação do AppScan Source, ao seguir as etapas neste tópico.

Antes de Iniciar

Antes de concluir esta tarefa, certifique-se de que tenha saído ou encerrado todos os aplicativos cliente do AppScan Source. Os aplicativos cliente do AppScan Source incluem:

- AppScan Source for Analysis
- AppScan Source for Development (plug-in do Eclipse ou Visual Studio)(suportado somente no Windows e Linux)
- AppScan Source command line interface (CLI)
- AppScan Source for Automation

Além disso, se tiver instalado o AppScan Source for Automation, certifique-se da saída ou do encerramento do Automation Server:

- No Windows, pare o serviço **IBM Security AppScan Source Automation**.
- No Linux, emita este comando: /etc/init.d/ounceautod stop
- No macOS, emita este comando: launchctl stop com.ibm.appscan.autod

Procedimento

1. Defina uma variável de ambiente APPSCAN_SOURCE_SHARED_DATA=<data_dir>, em que <data_dir> é o local no qual deseja que os dados do AppScan Source sejam armazenados.

Nota:

- O local <data_dir> deve ser um caminho completo e absoluto que já exista na mesma máquina que a instalação do AppScan Source.
- O nome do diretório <data_dir> pode conter apenas caracteres em inglês. Não são permitidas pastas que contêm caracteres que não estejam em inglês.

2. Localize o diretório de dados padrão que foi criado quando o AppScan Source foi instalado (consulte o “Diretório de Dados Padrão do AppScan Source” na página 149 para saber mais sobre os locais do diretório de dados padrão).
3. Copie ou mova o conteúdo do diretório de dados padrão para o local <data_dir> especificado na variável de ambiente.
4. **Aplica-se somente ao AppScan Source for Automation instalado no Linux:**
 - a. Edite o arquivo /etc/init.d/ounceautod.
 - b. Localize esta linha,

```
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 &
```

e substitua por esta:

```
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 &
```
 - c. Salve o arquivo /etc/init.d/ounceautod.

Nota: O comando acima é uma linha.

O que Fazer Depois

Se tiver instalado do AppScan Source for Automation, inicie o Automation Server:

- No Windows, inicie o serviço **IBM Security AppScan Source Automation**.
- No Linux, emita este comando: /etc/init.d/ounceautod start
- No macOS, emita este comando: launchctl start com.ibm.appscan.autod

Classificações

As descobertas são classificadas pelo AppScan Source para indicar se elas são descobertas de segurança ou de cobertura de varredura. As descobertas de segurança representam vulnerabilidades de segurança reais ou prováveis - enquanto que descobertas de cobertura de varredura representam áreas nas quais a configuração poderia ser melhorada para fornecer melhor cobertura de varredura.

Cada descoberta pertence a uma destas *classificações*:

- Descoberta de segurança **Definitiva**: Uma descoberta que contém um design, uma implementação ou uma violação de política definitiva que apresenta uma oportunidade para que um invasor faça com que o aplicativo opere de uma maneira indesejada.

Esse ataque poderia resultar no acesso não autorizado, furto ou distorção de dados, sistemas ou recursos. Cada descoberta de segurança definitiva é completamente articulada e o padrão subjacente específico da condição vulnerável é conhecido e descrito.

- Descoberta de segurança **Suspeita**: Uma descoberta que indica uma condição suspeita e potencialmente vulnerável que requer informações ou requerer adicional. Um elemento de código ou estrutura que pode criar uma vulnerabilidade quando usada incorretamente.

Uma descoberta suspeita difere de uma descoberta definitiva porque há alguma condição desconhecida que impede uma determinação conclusiva da

vulnerabilidade. Exemplos dessa incerteza podem ser o uso de elementos dinâmicos ou de funções de biblioteca para os quais o código fonte não está disponível. Como resultado, há um nível adicional de pesquisa que é necessário para confirmar ou rejeitar uma descoberta suspeita como definitiva.

- Descoberta de **Cobertura de Varredura**: Descobertas que representam áreas nas quais a configuração poderia ser melhorada para fornecer melhor cobertura de varredura (por exemplo, descobertas de dissipador perdido).

Nota: Em alguns casos, uma classificação igual a **Nenhum** pode ser usada para denotar uma classificação que não é uma descoberta de segurança nem uma descoberta de cobertura de varredura.

Avisos

Essas informações foram desenvolvidas para produtos e serviços oferecidos nos Estados Unidos. É possível que a IBM não ofereça em outros países os produtos, serviços ou recursos discutidos neste documento. Consulte um representante IBM local para obter informações sobre produtos e serviços disponíveis atualmente em sua área. Qualquer referência a um produto, programa ou serviço IBM não deve expor ou implicar que apenas esse produto, programa ou serviço IBM pode ser usado. Qualquer produto, programa ou serviço funcionalmente equivalente, que não infrinja nenhum direito de propriedade intelectual da IBM poderá ser usado. No entanto, é responsabilidade do usuário avaliar e verificar a operação de qualquer produto, programa ou serviço não IBM.

A IBM pode ter patentes ou solicitações de patentes pendentes relativas a assuntos tratados nesta publicação. O fornecimento desta publicação não garante ao Cliente nenhum direito sobre tais patentes. É possível enviar suas dúvidas sobre licenças, por escrito, para:

Gerência de Relações Comerciais e Industriais da IBM Brasil
Av. Pasteur, 138-146
Botafogo
Rio de Janeiro, RJ
CEP 22290-240

Para pedidos de licença relacionados a informações de DBCS (Conjunto de Caracteres de Byte Duplo), entre em contato com o Departamento de Propriedade Intelectual da IBM em seu país ou envie pedidos de licença, por escrito, para:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

O parágrafo a seguir não se aplica a nenhum país em que tais disposições não estejam de acordo com a lei local:

A INTERNATIONAL BUSINESS MACHINES CORPORATION FORNECE ESTA PUBLICAÇÃO "NO ESTADO EM QUE SE ENCONTRA" SEM GARANTIA DE NENHUM TIPO, SEJA EXPRESSA OU IMPLÍCITA, INCLUINDO, MAS NÃO LIMITADO ÀS GARANTIAS IMPLÍCITAS DE NÃO INFRAÇÃO, COMERCIALIZAÇÃO OU ADEQUAÇÃO A UM DETERMINADO PROPÓSITO.

Alguns estados não permitem a exclusão de garantias expressas ou implícitas em certas transações; portanto, esta disposição pode não se aplicar ao Cliente.

Estas informações podem conter imprecisões técnicas ou erros tipográficos. Periodicamente, são feitas alterações nas informações aqui contidas; tais alterações serão integradas em futuras edições desta publicação. A IBM pode, a qualquer momento, aperfeiçoar e/ou alterar os produtos e/ou programas descritos nesta publicação, sem aviso prévio.

Referências nestas informações a websites não IBM são fornecidas apenas por conveniência e não representam de forma alguma um endosso a esses websites. Os materiais contidos nesses Websites não fazem parte dos materiais deste produto IBM e a utilização desses Websites é de inteira responsabilidade do Cliente.

A IBM pode utilizar ou distribuir as informações fornecidas da forma que julgar apropriada sem incorrer em qualquer obrigação para com o Cliente.

Licenciados deste programa que desejam obter informações sobre este assunto com objetivo de permitir: (i) a troca de informações entre programas criados independentemente e outros programas (incluindo este) e (ii) a utilização mútua das informações trocadas, devem entrar em contato com:

Gerência de Relações Comerciais e Industriais da IBM Brasil
Av. Pasteur, 138-146
Botafogo
Rio de Janeiro, RJ
CEP 22290-240

Tais informações podem estar disponíveis, sujeitas a termos e condições apropriadas, incluindo em alguns casos, o pagamento de uma taxa.

O programa licenciado descrito nesta publicação e todo o material licenciado disponível são fornecidos pela IBM sob os termos do Contrato com o Cliente IBM, do Contrato Internacional de Licença do Programa IBM ou de qualquer outro contrato equivalente.

Todos os dados de desempenho aqui contidos foram determinados em um ambiente controlado. Sendo assim, os resultados obtidos em outros ambientes operacionais podem variar significativamente. Algumas medidas podem ter sido tomadas em sistemas em nível de desenvolvimento e não há garantia de que estas medidas serão iguais em sistemas geralmente disponíveis. Além disso, algumas medidas podem ter sido estimadas por extrapolação. Resultados reais podem variar. Os usuários desta publicação devem verificar os dados aplicáveis a seu ambiente específico.

As informações sobre produtos não IBM foram obtidas junto aos fornecedores dos respectivos produtos, de seus anúncios publicados ou de outras fontes disponíveis publicamente. A IBM não testou estes produtos e não pode confirmar a precisão de seu desempenho, compatibilidade nem qualquer outra reivindicação relacionada a produtos não IBM. Dúvidas sobre os recursos de produtos não IBM devem ser encaminhadas diretamente a seus fornecedores.

Todas as declarações relacionadas aos objetivos e intenções futuras da IBM estão sujeitas a alterações ou cancelamento sem aviso prévio e representam apenas metas e objetivos.

Todos os preços IBM mostrados são preços de varejo sugeridos pela IBM, são atuais e estão sujeitos a alteração sem aviso prévio. Os preços do revendedor podem variar.

Estas informações foram projetadas apenas com o propósito de planejamento. As informações aqui contidas estão sujeitas a alterações antes que os produtos descritos estejam disponíveis.

Esta publicação contém exemplos de dados e relatórios utilizados em operações diárias de negócios. Para ilustrá-los da forma mais completa possível, os exemplos podem incluir nomes de indivíduos, empresas, marcas e produtos. Todos estes nomes são fictícios e qualquer semelhança com nomes e endereços utilizados por uma empresa real é mera coincidência.

LICENÇA DE COPYRIGHT:

Estas informações contêm programas de aplicativos de exemplo no idioma de origem, ilustrando as técnicas de programação em várias plataformas operacionais. É possível copiar, modificar e distribuir esses programas de exemplo sem a necessidade de pagar à IBM, com objetivos de desenvolvimento, uso, marketing ou distribuição de programas aplicativos em conformidade com a interface de programação do aplicativo para a plataforma operacional para a qual os programas de exemplo são criados. Esses exemplos não foram completamente testados sob todas as condições. Portanto, a IBM não pode garantir ou implicar confiabilidade, manutenção ou função destes programas. O Cliente pode copiar, modificar e distribuir estes programas de amostra de qualquer maneira sem pagamento à IBM, com objetivos de desenvolvimento, utilização, marketing ou distribuição de programas aplicativos em conformidade com interfaces de programação de aplicativos da IBM.

Cada cópia ou parte destes programas de amostra ou qualquer trabalho derivado deve incluir um aviso de copyright com os dizeres:

© (nome da sua empresa) (ano). Partes deste código são derivadas dos Programas de Amostra da IBM Corp. © Copyright IBM Corp. _digite o ano ou anos_. Todos os direitos reservados.

Se essas informações estiverem sendo exibidas em cópia eletrônica, as fotografias e ilustrações coloridas podem não aparecer.

Marcas registradas

IBM, o logotipo IBM e ibm.com são marcas ou marcas registradas da International Business Machines Corp. nos Estados Unidos e/ou em outros países. Outros nomes de produtos e serviços podem ser marcas comerciais registradas da IBM ou de outras empresas. Uma lista atual de marcas registradas da IBM está disponível na web em Informações de copyright e marca registrada em www.ibm.com/legal/copytrade.shtml.

Adobe, Acrobat, PostScript e todas as marcas registradas baseadas em Adobe são marcas ou marcas registradas da Adobe Systems Incorporated nos Estados Unidos e/ou em outros países.

IT Infrastructure Library é uma marca registrada da Agência Central de Computação e Telecomunicações, que agora faz parte do Departamento de Comércio do Governo.

Intel, o logotipo Intel, Intel Inside, o logotipo Intel Inside, Intel Centrino, o logotipo Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium e Pentium são marcas ou marcas registradas da Intel Corporation ou de suas subsidiárias nos Estados Unidos e em outros países.

Linux é uma marca registrada da Linus Torvalds nos Estados Unidos e/ou em outros países.

Microsoft, Windows, Windows NT e o logotipo Windows são marcas comerciais da Microsoft Corporation nos Estados Unidos e/ou em outros países.

ITIL é uma marca registrada, e uma marca registrada da comunidade do Departamento de Comércio do Governo, e está registrada no U.S. Patent and Trademark Office.

UNIX é uma marca registrada da The Open Group nos Estados Unidos e em outros países.

Java e todas as marcas comerciais registradas e logotipos baseados em Java são marcas comerciais ou marcas registradas da Oracle e/ou de suas afiliadas.

Cell Broadband Engine é uma marca comercial da Sony Computer Entertainment, Inc. nos Estados Unidos e/ou em outros países e, sendo assim, é usada sob licença.

Linear Tape-Open, LTO, o logotipo LTO, Ultrium e o logotipo Ultrium são marcas comerciais da HP, IBM Corp. e Quantum nos EUA e em outros países.

Índice Remissivo

A

- API de Acesso a Dados 67
 - Argumentos da JVM 67
 - modelo de objeto 68
 - usando 70
- aplicativos
 - criando com Ounce/Ant 65
- AppScan Enterprise Server
 - certificado SSL 39
- AppScan Source
 - Login no AppScan Enterprise Server
 - certificado SSL 39
- AppScan Source for Automation 91, 97
 - arquivo de configurações 98
 - max_concurrent_requests 98
 - port 98
 - server_hostname 98
 - efetuando login 97
 - linha de comandos 99
 - logs 99
- Automation Server
 - arquivo de configurações 98
 - max_concurrent_requests 98
 - port 98
 - server_hostname 98
- ID do Pedido 100
- linha de comandos 99
- logs 99

C

- Classes e Métodos de API de Acesso a Dados 70
 - AssessedFile 71
 - getFilename 71
 - getFindings 71
 - getStats 71
 - Assessment 72
 - getAssesseeName 73
 - getAssessments 72
 - getFileByPath 73
 - getFiles 73
 - getFindings 72
 - getStats 72
 - AssessmentDiff 74
 - AssessmentFilter 74
 - AssessmentFilter 74, 75
 - AssessmentResults 75
 - getAssessmentForApplication 76
 - getAssessmentForProject 77
 - getAssessments 76
 - getFindings 75
 - getMessages 77
 - getName 77
 - getStats 76
 - Chamada 78
 - getCalls 78
 - getClassName 79
 - getColumnNumber 78
 - getFilename 78

- Classes e Métodos de API de Acesso a Dados (*continuação*)
 - Chamada (*continuação*)
 - getLineNumber 78
 - getMethodName 79
 - getSignature 79
 - getSrcContext 79
 - getTraceType 80
 - ClassificationType 80
 - membros estáticos 80
 - valor 80
 - com.ouncelabs.sdk.AssessmentDiff
 - getCommonFindings 74
 - getLostFindings 74
 - getNewFindings 74
 - com.ouncelabs.sdk.Factory 81
 - clearCache 82
 - diffAssessments 83
 - encerramento 81
 - getPublishedAssessments 83
 - init 81
 - login 82
 - openAssessment 82
 - DateProximityUnit 80
 - membros estáticos 80
 - valor 81
 - Finding 83
 - getApiName 84
 - getCallerName 84
 - getClassification 84
 - getDefectInfo 88
 - getDefectSubmissionDate 87
 - getDefectSubmissionUser 87
 - getFilename 83
 - getLineNumber 84
 - getModifiedClassification 86
 - getModifiedSeverity 86
 - getModifiedVulnerabilityType 87
 - getNotes 89
 - getOriginalClassification 85
 - getOriginalSeverity 85
 - getOriginalVulnerabilityType 86
 - getProperties 88
 - getSeverity 85
 - getSrcContext 87
 - getTrace 89
 - getVulnerabilityType 85
 - isExcluded 89
 - OunceException 91
 - Rastreamento 90
 - getCalls 90
 - SeverityType 90
 - membros estáticos 90
 - valor 90
 - classificação
 - cobertura de varredura 151
 - definitivo 151
 - suspeito 151
 - CLI (consulte a interface da linha de comandos) 19

D

- descobertas
 - classificação 151
- diretório de instalação padrão 149

F

- formatos de saída da interface com a linha de comandos 52
- formatos de saída de relatório 100
 - html 100
 - pdf-abrangente 100
 - pdf-anotado 100
 - pdf-detalhado 100
 - pdf-resumo 100
 - zip 100
- Framework for Frameworks 109
 - ações comuns executadas 118
 - exemplo 110
 - criar jar 115
 - criar manifesto 115
 - importar 111
 - JAR de exportação 118
 - sobre 111
 - métodos sintéticos de alto nível 128
 - exemplo 132
 - formato VDB 129
 - usando 129
 - principais componentes 110
 - scanner customizado da WSDL 134
 - usando
 - criar classe F4FHandler 114

I

- ID do Pedido 99
- instalação
 - local do arquivo 149
 - local dos dados 149
 - alterando 150
- interface de linha de comandos 19
 - acessando a partir do utilitário de construção Ounce/Ant 66
 - árvore de objetos 19
- comandos 23
 - clearcache 28
 - delete 31
 - deleteassess 31
 - deleteuser 31
 - delvar 31
 - details 32
 - echo 33
 - getaseinfo 33
 - help 34
 - importar 34
 - info 35
 - list 35
 - listassess 36
 - listgroups 36
 - listusers 37

interface de linha de comandos
(*continuação*)

comandos (*continuação*)

- log 37
- login 38
- login_file 40
- login_local 40
- logout 40
- moduser 41
- newuser 42
- openapplication 44
- openassessmentfile 46
- printuser 47
- publishassess 47
- publishassessase 48
- quit 50
- record 51
- refresh 51
- register 51
- relatório 52
- removeassess 52
- resumo 23
- script 55
- senha 46
- setaseinfo 57
- setcurrentobject 58
- setvar 58
- sobre 27
- unregister 59
- varredura 54

contexto 19

executando avaliações

- automatizadas 59

exibição em idioma nacional 21

iniciando 21

permissões 21

sintaxe 22

L

Linux

- convenções do utilitário de construção Ounce/Make 3
- iniciando a interface com a linha de comandos 21
- Utilitário de construção Ounce/Make

 - caminho da procura 8

M

mensagens de erro

- componentes do cliente 139

O

Ounce/Maven 91

- cenários 92
- instalando 91
- objetivos 93
- ounce:application 94
- ounce:project 94
- ounce:project-only 94
- ounce:report 95
- ounce:scan 95
- usando 92

ounceautod 97

ounceautod (*continuação*)

- arquivo de configurações 98
- comandos 99

 - GenerateReport 100
 - PublishAssessment 102
 - PublishAssessmentASE 102
 - ScanApplication 104
 - Wait 107

- linha de comandos 99
- login de linha de comandos 97
- logs 99

P

permissões

- comando moduser 41
- comando newuser 42

Plug-in Ounce/Maven

- cenários

 - aplicativos de varredura 93
 - criando arquivos de aplicativo e de projeto 92
 - integrando relatórios com o destino do site 93
 - relatórios 93

projetos

- arquivos de nomenclatura com o utilitário de construção Ounce/Make 3
- nomenclatura explícita 4
- criando com Ounce/Ant 63
- ounceCreateProject 63
- ounceExclude 64
- ounceSourceRoot 64
- ounceWeb 64
- nomenclatura com Ounce/Ant 65

R

relatórios 52

T

tipos de relatório de descobertas 100

U

Utilitário de construção Ounce/Ant 61

- arquivo jar ant-contrib 61
- criando aplicativos 65
- criando projetos 63

 - ounceCreateProject 63
 - ounceExclude 64
 - ounceSourceRoot 64
 - ounceWeb 64

- Integração Apache/Ant 61
- integração de construção 66
- nomeando projetos 65
- ounceant.jar 61
- Propriedades 62

 - configurando 63

- tarefa ounceCli 66
- Versão de Java JDK 61

Utilitário de construção Ounce/Make 1

- arquivo de propriedades 8

Utilitário de construção Ounce/Make
(*continuação*)

- amostras 14
- localizando 8
- ativando 1
- caminho da procura 8
- comando 1
- comandos

 - make 1
 - make clean 1

- compiladores suportados 2
- elementos de arquivo de propriedades 9

 - Compiler 9
 - Executable 13
 - FileOptions 12
 - GlobalProjectOptions 12
 - Make 10
 - MakeOptions 10
 - MountRoot 14
 - Options 10
 - Vinculador 9

- executando 3

 - opções 10

- exemplos 14

 - Ounce/Make com opção recursiva 16
 - Ounce/Make com projeto único e opção recursiva 17
 - Ounce/Make sem opções 15

- incluindo na variável de ambiente PATH 3
- mensagens de saída 5
- nomeando arquivos de projeto 3

 - nomenclatura explícita 4

- opções 5

 - opções make 5

- operação 3
- ouncemake_properties.xml (consulte também o arquivo de propriedades) 8
- plataformas suportadas 2
- requisitos 1
- sintaxe de comando 5
- uso 3
- versões make suportadas 2

V

varredura

- incremental 28

W

Windows

- convenções do utilitário de construção Ounce/Make 3
- iniciando a interface com a linha de comandos 21
- Utilitário de construção Ounce/Make

 - caminho da procura 8



Impresso no Brasil