SERENA

SERENA®
RELEASE AUTOMATION

Plug-ins Guide

Serena Proprietary and Confidential Information

Trademarks

U.S. Government Rights

Part number: Serena ALM Product version: 4.5

Publication date: 2013-06-25

Table of Contents

Chapter 1: Serena Release Automation Plug-ins

This guide lists and describes a subset of the many plug-ins included with Serena Release Automation. When you are creating processes, use these plug-ins to configure and execute steps needed for your deployment, such as requests to application servers, build management tools, database and file management systems, and test automation tools.

All of the plug-ins include self-documentation. As you use plug-ins within the processes, click the ? icon beside the properties fields to see a short description of each property.

See the *Serena Release Automation User's Guide* for information on using plug-ins in processes and creating new plug-ins, including examples.

Note: If you require documentation for a plug-in beyond that provided in the Serena Release Automation guides, please check the Serena Support Knowledgebase first and follow your usual Serena Support procedures.

List of Available Plug-ins

The plug-ins provided with Serena Release Automation are listed here, divided into broad usage categories.

Note: New plug-ins are actively in development. If you don't see a plug-in that you need in this list, please search the Serena Support Knowledgebase or follow your usual Serena Support procedures.

Application Lifecycle Management Plug-ins

1. CollabNet TeamForge

Application Server Plug-ins

1. Apache HTTP
2. Apache Tomcat
3. GlassFish
4. IBM WebSphere
   a. WebSphere
   b. WebSphere MQ
   c. WebSphere Message Broker - CMP
5. JBoss
6. Microsoft IIS
   a. IIS AdminScripts

b. IIS AppCmd

c. IIS MS-Deploy

Cloud Support Plug-ins

1. Amazon EC2

Build Tool Plug-ins

1. Apache Ant
2. Apache Maven
3. Microsoft Build Engine (MSBuild)
4. UrbanCode AnthillPro

Change Management Plug-ins

1. HP Quality Center
2. ServiceNow Change Management

Content Management Plug-ins

1. Microsoft SharePoint
2. Serena PVCS
3. Subversion

Database Management Plug-ins

1. DBUpgrader
2. DBDeploy
3. Microsoft SQL Server SQLCMD
4. Oracle SQL*Plus
5. SQL-JDBC

Integration Tool Plug-ins

1. Informatica (Data Integration)
2. Microsoft BizTalk Server (Application Integration)

Issue Tracking Plug-ins

1. Atlassian JIRA
2. Rally Platform

IT/Network Management Plug-ins

1. Citrix NetScaler
2. F5 BIG-IP

3. Nagios XI

Release Automation Plug-ins

1. BMC Application Release Automation (ARA)
2. Serena Release Automation
    a. Application
    b. Component
    c. Configuration Management
    d. Environment
    e. Resource
    f. System
    g. Version
    h. Versioned File Storage (VFS)

Script Automation Plug-ins

1. Groovy

System Tool Plug-ins

1. 7zip
2. Linux System Tools
3. Microsoft
    a. Message Queuing (MSMQ)
    b. Software Installer (MSI)
    c. Windows Service Control Manager
    d. Windows System Information
4. Red Hat Package Manager (RPM)
5. Unix/Linux Autosys
6. Unix/Linux FileUtils
7. Windows/Unix/Linux Shell

Test Automation Plug-ins

1. Deploy Tools
2. QuickTest Pro
3. Selenium

Chapter 2: Amazon EC2 Plug-in

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides environment support in the cloud. This plug-in consists of the following available steps for you to add to your processes in Serena Release Automation:

- Launch Instance
- Create Security Group
- Start Instances
- Stop Instances
- Terminate Instances
- Wait for Instances
- Associate IPs
- Register Instances with LoadBalance
- Deregister Instances with LoadBalance
- Get Public DNS

For information about Amazon EC2, see http://aws.amazon.com.

Important: The properties you need to specify for Serena Release Automation processes are Amazon EC2-specific. You should consult the Amazon EC2 documentation for information on how to find and complete that information if you are not already familiar with Amazon EC2.

Creating an Amazon EC2 Launch Instance Step

The basic building blocks of Amazon EC2 are the *Amazon Machine Images* (AMI). An *AMI* is a template that contains a software configuration, such as an operating system, application server, or applications that you can run in an Amazon environment. Amazon has a variety of AMIs available and you can also create you own.

To launch an instance from AMI, you need to create an Amazon EC2 `Launch Instance` step.

To create an Amazon EC2 launch instance step:

1. Enter a `Name` for your Launch Instance step.
2. Specify values for the following required properties:

Property	Description
Access Key Id	The EC2 access key ID to use to log in.
Secret Key	The EC2 secret key.
# of instances	The number of EC2 instances to startup.
Instance Type	The type of instances to run. Allowed values are m1.small, m1.large, m1.xlarge, m2.xlarge, m2.4xlarge, c1.medium, c1.xlarge.
AMI ID	The AMI ID of the instances to be started.
AWS Jar	The full path to the AWS SDK .jar file.

3. You may choose to specify values for the following optional properties:

Property	Description
Security Group	A comma-separated list of security group names to use.
Availability Zone	The zone to start these instances in.
Keypair	The keypair to start these instances with.
User data	The user data to be passed to the instance.

Creating an Amazon EC2 Security Group Step

The Amazon EC2 security group acts as a firewall that controls the traffic allowed into a group of instances.

To create a security group step:

1. Enter a Name for your Security Group step.
2. Specify values for the following required properties:

Property	Description
Access Key Id	The EC2 access key ID to use to log in.
Secret Key	The EC2 secret key.
AWS Jar	The full path to the AWS SDK .jar file.

3. You may choose to specify values for the following optional properties:

Property	Description
Name	The name of the security group to create, if not using a file to add permissions.
Description	The description of the security group to create, if not using a file to add permissions.
VPC ID	The ID of the Virtual Private Cloud (VPC) to use, if not using a file to add permissions.
Definition File	The file describing the security group allowed ips/ports. If left blank, an empty security group will be created. Otherwise, the file has the following format: <SecurityGroup name="name" description="description" vpcId="id_optional"> <ipPermission protocol="tcp" fromPort="nn" toPort="nn"> <ipRange value="0.0.0.0/0"/> </ipPermission> </SecurityGroup>

Amazon EC2 Start, Stop, and Terminate Instance Steps

To start, stop, or terminate an Amazon EC2 instance, you pass the same set of parameters. These give the information needed to identify the instance that you want to start, stop, or terminate. The only difference is start and stop steps accept only one Instance ID whereas the terminate step accepts more than one instance ID.

To create a start, stop, or terminate instance step:

1. Enter a `Name` for your start, stop, or terminate instance step.
2. Specify values for the following required properties:

Property	Description
Access Key Id	The EC2 access key ID to use to log in.
Secret Key	The EC2 secret key.
instance ID	The Instance ID to start or stop, or for the terminate step, a list of instance IDs separated by commas.

Property	Description
AWS Jar	The full path to the AWS SDK .jar file.

Creating an Amazon EC2 Wait for Instance Step

Use to create a step that waits until the instance(s) specified in the `Instance IDs` field switch to the state specified in the `State` field.

To create a Wait for Instance step:

1. Enter a `Name` for your Wait for Instance step.
2. Specify values for the following required properties:

Property	Description
Access Key Id	The EC2 access key ID to use to log in.
Secret Key	The EC2 secret key.
Instance IDS	A comma-separated list of instance IDs that correspond to the IPs to associate.
State	The state for instances to be in.
Timeout	The timeout for this step, in milliseconds.
AWS Jar	The full path to the AWS SDK .jar file.

Creating an Amazon EC2 Associate IPs Step

In an AWS, an Elastic IP Address (EIP) enables you to reserve an IP address that you can then assign to any AMI instance you have running. If needed, at any time you can also change the assignment to a different instance.

According to Amazon, this feature is designed for "dynamic cloud computing". Once an EIP has been associated with an instance, it remains associated with that instance until you release it.

To create an associate IPs step:

1. Enter a `Name` for your Associate IPs step.
2. Specify values for the following required properties:

Property	Description
Access Key Id	The EC2 access key ID to use to log in.
Secret Key	The EC2 secret key.
IPS	A new-line separated list of elastic ips to associate with instances.
Instance IDS	A comma-separated list of instance IDs that correspond to the IPs to associate.
AWS Jar	The full path to the AWS SDK .jar file.

Creating an Amazon EC2 Register Instances with LoadBalance Step

Elastic Load Balancing automatically distributes incoming application traffic across multiple Amazon EC2 instances. It enables you to achieve even greater fault tolerance in your applications, seamlessly providing the amount of load balancing capacity needed in response to incoming application traffic. Elastic Load Balancing detects unhealthy instances within a pool and automatically reroutes traffic to healthy instances until the unhealthy instances have been restored.

You can enable Elastic Load Balancing within a single *Availability Zone* or across multiple zones for even more consistent application performance. Elastic Load Balancing can also be used in an Amazon Virtual Private Cloud (VPC) to distribute traffic between application tiers. For more details, see the *Amazon Elastic Compute Cloud* documentation.

To create an Amazon EC2 Register Instance with LoadBalance Step:

1. Enter a `Name` for your Register Instance with LoadBalance step.
2. Specify values for the following required properties:

Property	Description
Access Key Id	The EC2 access key ID to use to log in.
Secret Key	The EC2 secret key.
Load Balancer Name	The name of the load balancer in this EC2 account to register the instances from.
Instance Ids	Instance ID to start.
AWS Jar	The full path to the AWS SDK .jar file.

Creating an Amazon EC2 Deregister Instances with LoadBalance Step

The Deregister Instances With LoadBalancer works the same as the re-registration except that you have to use another step for it.

To create an Amazon EC2 Deregister Instance with LoadBalance Step:

1. Enter a Name for your Register Instance with LoadBalance step.
2. Specify values for the following required properties:

Property	Description
Access Key Id	The EC2 access key ID to use to log in.
Secret Key	The EC2 secret key.
Load Balancer Name	The name of the load balancer in this EC2 account to de-register the instances from.
Instance Ids	Instance ID to start.
AWS Jar	The full path to the AWS SDK .jar file.

Creating an Amazon EC2 Get Public DNS Step

You may get the list of public DNSs that a list of Amazon EC2 instances are running on.

To create an Amazon EC2 Get Public DNS Step:

1. Enter a Name for your Get Public DNS step.
2. Specify values for the following required properties:

Property	Description
Access Key Id	The EC2 access key ID to use to log in.
Secret Key	The EC2 secret key.
Instance Ids	A comma-separated list of instance IDs to be retrieved. The DNS list returned will be in the same order as the IDs given.
AWS Jar	The full path to the AWS SDK .jar file.

Chapter 3: DBDeploy Plug-in

The DBDeploy plug-in enables Serena Release Automation to upgrade a database. This plug-in consists of the following step for you to add to your processes in Serena Release Automation:

- Run DbDeploy

For information about DBDeploy, see http://dbdeploy.com/.

Creating a Run DbDeploy Step

You may use this step to upgrade your database.

To create a Run DbDeploy step:

1. Enter a `Name` for your Run DbDeploy step.
2. Specify values for the following required properties:

Property	Description
Driver Classname	Class name of the driver for the database.
DB Driver Jar Base Path	Location of the DB driver jar files.
DB Driver Jars	Name of the database driver .jar file. Supports the * wildcard.
URL	Database URL. For example, `jdbc:hsqldb:hsql://localhost/xdb`
User	Username for the database login.
Password	Password for the username for the database login.
SQL File path	The path to the directory within which the SQL files reside.
Changelog table name	Name of the change log table to use. This is useful if the DDL and DML need to be separate when deploying to replicated environments.
SQL statement delimiter	The delimiter to use to separate scripts into statements. Default is a semicolon ";"

Property	Description
SQL statement delimeter type	To split on the delimiter whenever it occurs, specify `normal`. To split on the delimiter only if it is featured on a line by itself, specify `row`. Default is `normal`.
Line Ending	How lines should be separated in SQL statements that are issued by way of JDBC. By default, the step uses the appropriate line ending for the platform it is running on and this is normally satisfactory. However, due to a bug in some Oracle drivers, the Windows default of CRLF may not always work. This property takes the values: • `platform` • `cr` • `lf` • `crlf`

3. (Optional) You may choose to specify the following:
   a. The `Output File` property can be used to switch to output script mode. The name of the script that DBDeploy will output. This should include a full or relative path.
   b. The `DB Type` is required if script mode is used. This refers to the target DBMS.
   c. The `Template Directory` is the directory from which to read customized template scripts. This is only relevant in script mode.

Chapter 4: Serena Release Automation System Plug-in

The Serena Release Automation System plug-in enables you to enter system, or global, properties to be used by a process. This plug-in consists of the following step for you to add to your processes in Serena Release Automation:

- Create System Property

For information about Serena Release Automation System Properties, see the *Serena Release Automation User's Guide*.

Creating a Serena Release Automation System Property Step

You may use this step to enter system, or global, properties to be used by a process..

To create a Serena Release Automation System Property step:

1. Enter a `Name` for your Serena Release Automation System Property step.
2. In the `Property Name` field, enter the name of the Serena Release Automation system property to be set by this step.
3. (Optional) You may choose to specify the following:
   a. The `Property Value` to which you want the Serena Release Automation system property set by this step.
   b. `Secure` the Serena Release Automation system property so that it is stored encrypted and its display is obscured in the Serena Release Automation user interface.

Chapter 5: 7zip Plug-in

7zip is an open source file archiver that combines a number of files together into a single file for easier transportation and storage. The 7zip plug-in provides the following step for you to add to your processes in Serena Release Automation:

- Extract Archive

For information about 7zip, see http://www.7-zip.org/.

Creating a 7zip Extract Archive Step

You may use this step to extract files from a 7zip archive.

To create a 7zip extract archive step:

1. Enter the `Name` for your extract archive step.
2. Use the `Include Files` field to list the file filters you want this step to use to select the files to include.

   

   Important: Files must each be listed on a new line separated from the next.
3. Specify from where the files should be extracted in the `Extract Directory` field.
4. (Optional) You may choose to specify the following:
   a. The `Directory Offset` relative to the current working directory where the step should run.
   b. Use the `Exclude Files` field to list file filters you want this step to exclude from using.

      

      Important: Files must each be listed on a new line separated from the next.

Chapter 6: Red Hat Package Manager (RPM) Plug-in

The Red Hat Package Manager (RPM) is an open packaging system that runs on Red Hat Enterprise Linux and other Linux and UNIX systems. This plug-in consists of the following available steps for you to add to your processes in Serena Release Automation:

- Install RPM
- Uninstall RPM
- Update RPM

For information about Red Hat Package Manager (RPM), see http://www.rpm.org/.

Creating an Install RPM Step

You may use this step to install RPM packages.

To create an install RPM step:

1. Enter the `Name` for your install RPM step.
2. Use the `RPM Packages` field to list the RPM packages you want this step to install.

   

   Important: Packages must each be listed on a new line separated from the next.

3. (Optional) You may choose to specify the following: `Install Options` (also in a new-line separated list) to be used during the install.

Creating an Uninstall RPM Step

You may use this step to uninstall RPM packages.

To create an uninstall RPM step:

1. Enter the `Name` for your uninstall RPM step.
2. Use the `RPM Packages` field to list the RPM packages you want this step to uninstall.

   

   Important: Packages must each be listed on a new line separated from the next.

3. (Optional) You may choose to specify the following: `Erase Options` (also as a new-line separated list) to be used during the uninstall.

Creating an Update RPM Step

You may use this step to update RPM packages.

To create an update RPM step:

1. Enter the `Name` for your update RPM step.
2. Use the `RPM Packages` field to list the RPM packages you want this step to install.

   

   Important: Packages must each be listed on a new line separated from the next.
3. (Optional) You may choose to specify the following: `Update Options` (also in a new-line separated list) to be used during the install.