Business Intelligence Solution Providers
Specialized in creating talent resource pool

Getting Started with SalesForce CRM

Working Force.com Migration Utility

Description:

BISP is committed to provide BEST learning material to the beginners and advance learners. In the same series, we have prepared a complete end-to end Hands-on Beginner's Guide for SalesForce. The document focuses on force.com migration utility. Join our professional training program and learn from experts.

History:

Version	Description Change	Author	Publish Date
0.1	Initial Draft	Chandra Prakash Sharma	10th Oct 2013
0.1	Review#1	Amit Sharma	10th Oct 2013

Contents

Force.com Migration Tool Overview

The Force.com Migration Tool is a Java/Ant-based command-line utility for moving metadata between a local directory and a SalesForce organization.

- Development projects where you need to populate a test environment with large amounts of setup changes — Making these changes using a Web interface could take a long time.
- Multistage release processes — A typical development process requires iterative building, testing, and staging before releasing to a production environment. Scripted retrieval and deployment of components can make this process much more efficient.
- Repetitive deployment using the same parameters — You can retrieve all the metadata in your organization, make changes, and deploy a subset of components. If you need to repeat this process, it's as simple as calling the same deployment target again.
- When migrating from stage to production is done by IT — Anyone that prefers deploying in a scripting environment will find the Force.com Migration Tool a familiar process.

Understanding Metadata API :

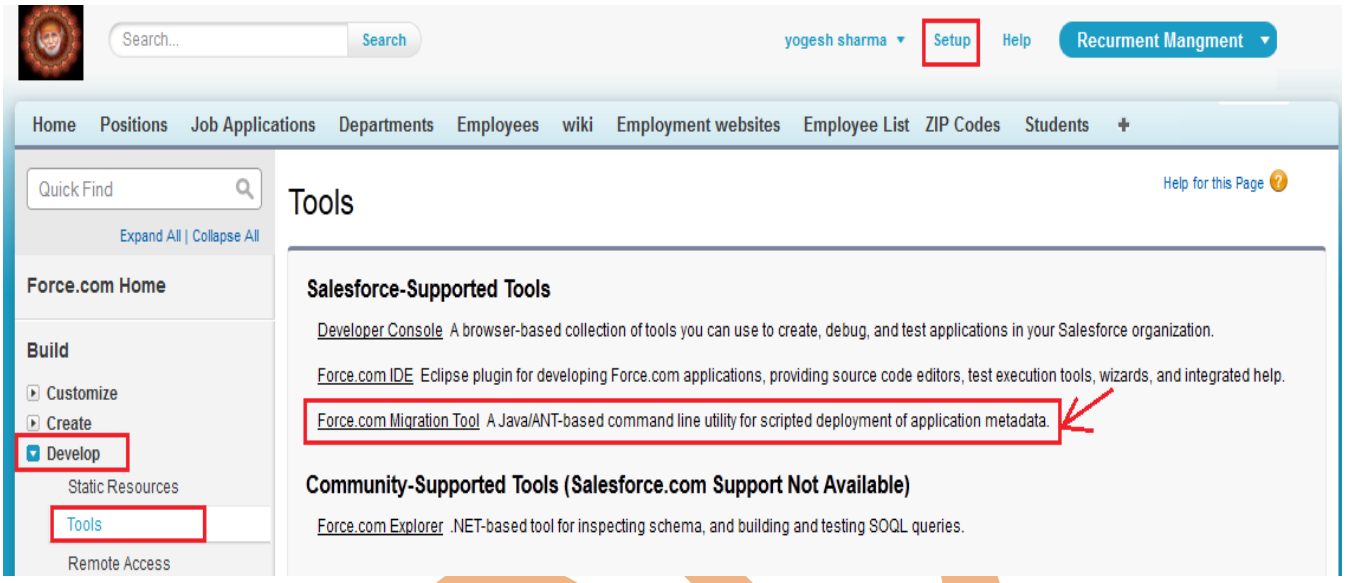
Metadata API contains a set of objects that manage setup and customization information (metadata) for your organizations,
and the SOAP calls that manipulate those objects. With Metadata API you can:

- Work with setup configuration as XML metadata files
- Migrate configuration changes between organizations
- Create your own tools for managing organization and application metadata

Installing the Force.com Migration Tool

Before you install the Force.com Migration Tool you will need Java and Ant installed on your local machine.
you can download on this link : http://ant.apache.org/ and also need JDK 1.6.x or later. .
http://www.oracle.com/technetwork/java/javase/downloads/index.html
1. Install Java and Ant.

2. Login to a SalesForce organization and download the Force.com Migration Tool, as described in Installing the Force.com
Migration Tool.
download the Force.com Migration Tool from a SalesForce organization.
Click on **Setup** > **Develop** > **Tools**, then click Force.com Migration Tool. see below.



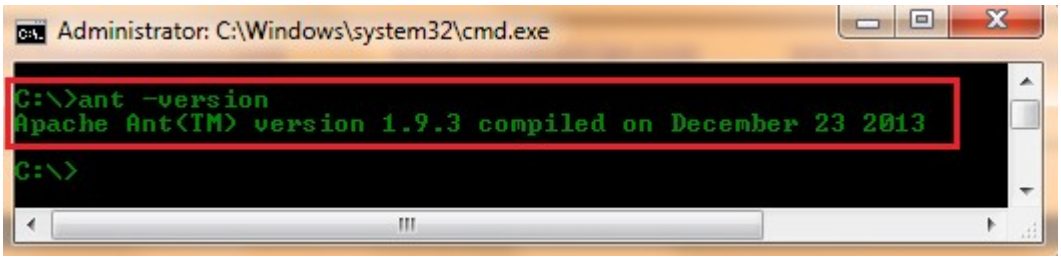
Save the .zip file locally and extract the contents to the directory of your choice.
Copy ant-salesforce.jar and paste into your Ant installation's lib directory. The lib directory is located in the
root folder of your Ant installation.

After install Jdk check java version open command prompt then type java -version, you can see below.

How set ant path :

Syntax	Example
SET ANT_HOME=my_path_to_ant_folder	Ex : SET ANT_HOME= D:\cp sharma\sw\apache-ant-1.9.3-bin\apache-ant-1.9.3
SET JAVA_HOME=my_path_to_jdk_folder	Ex: C:\Program Files\Java\jdk1.7.0_45
SET PATH=%PATH%;%ANT_HOME%/bin; %JAVA_HOME%\bin;	Ex : SET PATH=%PATH%;%ANT_HOME %/bin;%JAVA_HOME%\bin;

After set path you can check ant version by using this syntax.
syntax : ant -version



Using the Force.com Migration Tool

The Force.com Migration Tool is a Java/Ant-based command-line utility for moving metadata between a local directory and a SalesForce organization.
The general procedure you will follow when using the Force.com Migration Tool to copy metadata from one SalesForce organization to another is:

1) Enter credentials and connection information for source SalesForce organization in build.properties
2) Create retrieve targets in build.xml
3) Construct a project manifest in package.xml
4) Run the Force.com Migration Tool to retrieve metadata files from SalesForce
5) Enter credentials and connection information for destination SalesForce organization in build.properties
6) Run the Force.com Migration Tool to deploy metadata files or deletions to SalesForce

SalesForce Connection Information:

1. Go to the location where you extracted the Force.com Migration Tool files and open the sample subdirectory.
2. Open build.properties in a text editor and substitute a valid SalesForce username and password. If you are using a security token, paste the 25-digit token value to the end of your password.

you can see below.

```
1  # build.properties
2  #
3
4  # Specify the login credentials for the desired Salesforce organization
5  sf.username = <Insert your Salesforce username here>
6  sf.password = <Insert your Salesforce password here>
7  #sf.pkgName = <Insert comma separated package names to be retrieved>
8  #sf.zipFile = <Insert path of the zipfile to be retrieved>
9  #sf.metadataType = <Insert metadata type name for which listMetadata or bulkRetrieve operations are to be performed>
10
11 # Use 'https://login.salesforce.com' for production or developer edition (the default if not specified).
12 # Use 'https://test.salesforce.com for sandbox.
13 sf.serverurl = https://login.salesforce.com
14
15 sf.maxPoll = 20
16 # If your network requires an HTTP proxy, see http://ant.apache.org/manual/proxy.html for configuration.
17 #
```

Enter User name and Password with token

Parameter	Value
sf.username	The salesforce.com username for login. The username associated with this connection must have the "Modify All Data" permission. Typically, this is only enabled for System Administrator users. When connecting to a sandbox instance your sandbox name is appended to your username. For example, if your production username is foo@salesforce.com, and one of your sandboxes is called bar, then your sandbox username is foo@salesforce.com.bar.
sf.password	The password you use to log into the organization associated with this project. If you are using a security token, paste the 25-digit token value to the end of your password.
sf.serverurl	The salesforce server URL. Use https://login.salesforce.com to connect to a production or Developer Edition organization. To connect to a sandbox instance, change this to https://test.salesforce.com.

Constructing a Project Manifest :

The package.xml file is a project manifest that lists all the components you want to retrieve or deploy in a single request. You can retrieve or deploy only a single package at a time.
The following elements may be defined in package.xml :

Name	Description
<fullName>	The name of the server-side package to deploy into. If the <fullName> field is omitted, components will not be assigned to a package when deployed, and will be in the unpackaged package.
<types>	This element contains one or more <members> tags and one <name> tag, and is used to list the metadata components of a certain type to retrieve or deploy.
<members>	The full name of a component. There is one <members> element defined for each component in the directory. You can replace the value in this member with the wildcard character * (asterisk) instead of listing each member separately.
<name>	There is one name defined for each component type in the directory. This is a child element of <types>.
<version>	The Metadata API version number of the files being retrieved or deployed. When deploying, all the files must conform to the same version of the Metadata API.

Specifying Standard Objects :
To retrieve standard objects and/or custom fields on standard objects, you must name the component in package.xml.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
<types>
<members>Case.Employee__c</members>
<name>CustomField</name>
</types>
<types>
<members>Account</members>
<name>CustomObject</name>
</types>
<version>29.0</version>
</Package
```

Specifying Named Components :

To retrieve a component, specify the type of component in the <name> element and declare each component to be retrieved or deployed in the <members> element. The following is a sample package.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
<types>
<members>MyCustomObject__c</members>
<members>MyHelloWorldObject__c</members>
<name>CustomObject</name>
</types>
<version>29.0</version>
</Package>
```

Specifying all Components of a Type :

To retrieve all components of a particular type, use the wildcard symbol (*). For example, to retrieve all custom objects:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
<types>
<members>*</members>
<name>CustomObject</name>
</types>
<version>29.0</version>
</Package>
```

Specifying Standard Objects :

To retrieve standard objects and/or custom fields on standard objects, you must name the component in package.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
<types>
<members>Case.EngineeringReqNumber__c</members>
<name>CustomField</name>
</types>
<types>
<members>Account</members>
<name>CustomObject</name>
</types>
<version>29.0</version>
</Package>
```

Describing Metadata Types :

Field	Description
username	Required. The SalesForce username for login. The username associated with this connection must have the "Modify All Data" permission.
password	Required. The password you use to log into the organization associated with this project. If you are using a security token, paste the 25-digit token value to the end of your password.
serverurl	Optional. The SalesForce server URL (if blank, defaults to login.salesforce.com). To connect to a sandbox instance, change this to test.salesforce.com.
apiVersion	Optional. The API version to use for the metadata. The default is 29.0.
resultFilePath	Optional. The path of the output file where results are stored. The default output is the console. Directing the output to a file makes it easier to extract the relevant information for your package.xml manifest file.
trace	Optional. Defaults to false. Prints the SOAP requests and responses to the console. Note that this will show the user's password in plain text during login.

Get the list of metadata types enabled for your organization, specify a target in the build.xml file using

```
<sf:describeMetadata>.
<target name="describeMetadata">
<sf:describeRetrieve
username="${sf.username}"
password="${sf.password}"
serverurl="${sf.serverurl}"
resultFilePath="describeMetadata/describe.log"/>
</target>
```

Listing Components for a Metadata Type :

The listMetadata target retrieves property information about metadata components in your organization. This target is useful when you want to identify individual components in package.

Field	Description
username	Required. The SalesForce username for login. The username associated with this connection must have the “Modify All Data” permission.
password	Required. The password you use to log into the organization associated with this project. If you are using a security token, paste the 25-digit token value to the end of your password.
serverurl	Optional. The SalesForce server URL (if blank, defaults to login.salesforce.com). To connect to a sandbox instance, change this to test.salesforce.com.
metadataType	Required. The name of the metadata type for which you are retrieving property information; for example, CustomObject for custom objects, or Report for custom reports.
folder	The folder associated with the component. This field is required for components that use folders, such as Dashboard, Document, Email Template,

	or Report.
apiVersion	Optional. The API version to use for the metadata. The default is 29.0.
resultFilePath	Optional. The path of the output file where results are stored. The default output is the console. Directing the output to a file makes it easier to extract the relevant information for your package.xml manifest file.
trace	Optional. Defaults to false. Prints the SOAP requests and responses to the console. Note that this will show the user's password in plain text during login.

Get property information for components of one metadata type, such as CustomObject, specify a target in the build.xml file using

```
<target name="listMetadata">
<sf:listMetadata
username="${sf.username}"
password="${sf.password}"
serverurl="${sf.serverurl}"
metadataType="CustomObject"
resultFilePath="listMetadata/list.log"/>
</target>
```

Creating Retrieve Targets :

The build.xml file specifies a series of commands to be executed by Ant. Within the build.xml file are named targets that process a series of commands when you run Ant with a target name. The following parameters may be set for each <sf:retrieve> target :

Field	Description
username	Required. The SalesForce username for login. The username associated with this connection must have the “Modify All Data” permission.
password	Required. The password you use to log into the organization associated with this project. If you are using a security token, paste the 25-digit token value to the end of your password.
serverurl	Optional. The SalesForce server URL (if blank, defaults to login.salesforce.com). To connect to a sandbox instance, change this to test.salesforce.com.
retrieveTarget	Required. The root of the directory structure into which the metadata files are retrieved.
packageNames	Required if unpackaged is not specified. A comma-separated list of the names of the packages to retrieve.
apiVersion	Optional. The API version to use for the metadata. The default is 29.0.
pollWaitMillis	Optional. Defaults to 10000. The number of milliseconds to wait between attempts when polling for results of the retrieve request.
maxPoll	Optional. Defaults to 20. The number of times to poll the server for the results of the retrieve request.
singlePackage	Optional. Defaults to true. This must be set to false if you are retrieving multiple

	packages. If set to false, the retrieved zip file includes an extra top-level directory containing a subdirectory for each package.
trace	Optional. Defaults to false. Prints the SOAP requests and responses to the console. Note that this will show the user's password in plain text during login.
unpackaged	Required if packageNames is not specified. The path and name of a file manifest that specifies the components to retrieve.
unzip	Optional. Defaults to true. If set to true, the retrieved components are unzipped.

Retrieving Metadata from a SalesForce Organization :

To retrieve Force.com components :

1. Open a command prompt.
2. Run Ant by specifying a target name in build.xml. If this is the first time you are running Ant, use ant
retrieveUnpackaged to retrieve unpackaged components specified in package.xml.

Running Tests in a Deployment :

For deployment to a production organization, all the tests in your organization, except for those that originate from installed
If the deployment includes components for any of the following metadata types, all the tests are automatically run.

- ApexClass
- ApexComponent
- ApexPage
- ApexTrigger
- ArticleType
- CriteriaBasedSharingRule
- CustomDataType
- CustomField
- CustomObject
- DataCategoryGroup
- Flow
- InstalledPackage
- NamedFilter
- OwnerSharingRule
- PermissionSet
- Profile
- Queue
- RemoteSiteSettingRecordType
- Role
- SharingReason

- Territory
- Validation Rules
- Workflow