

Import/Export Plug-in for Domino Designer

Main Details

Summary	Allow XPage developers to import/export resources in/out the Domino Application using Eclipse import/export wizard dialogs.
Technology	Eclipse
Version	2.1.3
Support Domino Designer Version	8.5 or later (tested with 8.5.3 and 8.5.4)

Change History

Version	Change History
1.0.0	Support import/export resources in/out the Domino Application from/to the local file system using Eclipse import/export wizard dialogs.
1.1.0	Support import/export resources via ANT tasks.
1.2.0	Support import reusable control from OpenNTF.
1.2.2	Import the WebContent/WEB-INF/src folder by default. NOTE: Please delete the following xml file after update from previous versions to 1.2.2 or later versions: data\workspace\.metadata\.plugins\com.ibm.designer.extensibility.copyfile\importlist.xml.
1.3.0	Support export reusable control for OpenNTF.
1.3.1	Support bringing up import/export wizard pages from toolbar icons.
2.0.0	1. Better visibility: 1.1. Show just one icon "import from OpenNTF", hide the other options by collapsing them. 1.2. Add OpenNTF menu items to the right click menu. 2. Changes to Import wizard 2.1. Use Rest API to fetch the control download address, and list these controls by SWT instead of showing a web page in an embedded browser. 2.2. Allow users to import multi reused controls from multi catalogs at the same time. 2.3. Change the default importlist.xml to contain almost everything in the NSF. 2.4. Add "Open Temporary Directory" button and menus. 3. Changes to Export wizard 3.1. Use four licenses. 3.2. Change license selection radio to drop down list. 4. Fix some bugs

2.1.0	1. Support import unit test files of controls from OpenNTF 2. Support export control on server to .zip file for OpenNTF
2.1.3	1. Support import files from a local folder to any folder of the Designer Application. 2. Add .classpath to the default importlist.xml and support merge .classpath during import/export. 3. Fix a bug: can not open import file system wizard page from navigate view.

Contributor

Guo Yi, Chen Yang Jun
IBM employee, Regular
Software Engineer
IBM China Software Development LAB
IBM Software Group, IBM Collaboration Solutions
China

Niklas Heidloff
IBM employee, Regular
Lotus Notes/Domino App Dev Architect
IBM Deutschland RD GmbH
IBM Software Group, IBM Collaboration Solutions
Germany

Stephan H Wissel
IBM employee, Regular
Client Technical Specialist
IBM Sales & Distribution, Software Sales
IBM Collaboration & Productivity Advisor
Singapore

Deployment

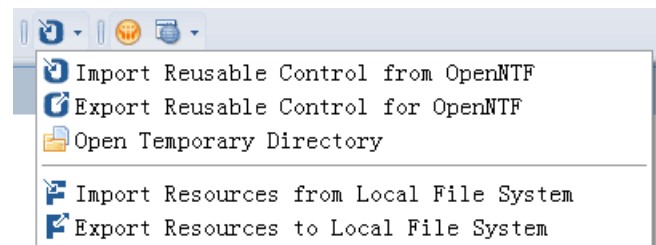
Update site	Local Folder
Feature Name	Import/Export Feature
Feature ID	com.ibm.designer.extensibility.copyfile.feature

Usage

Export/Import from UI

This plug-in extends the Eclipse export/import wizards and it is able to export/import Domino Designer resources out/in the Domino Designer Applications. There are four ways as following to export/import from UI using this plug-in:

1. Bring up import/export wizard pages from toolbar icons.



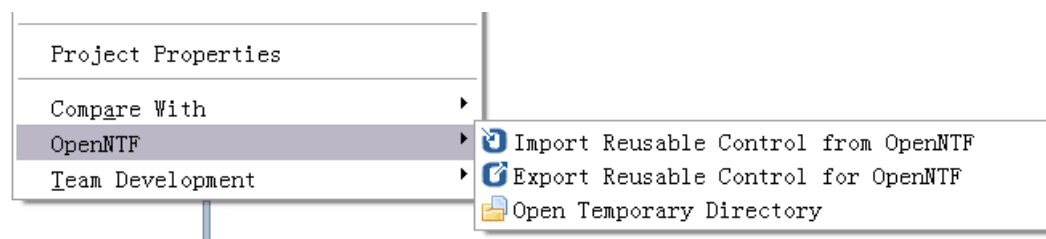
Above five icons represent:

- Import reusable control from OpenNTF
- Export reusable control for OpenNTF use
- Open Temporary Directory to see the Licenses and logs
- Import resources from the local file system
- Export resources to the local file system

from top to bottom respectively.

NOTE that these icons are enabled only when you focus on a Designer application.

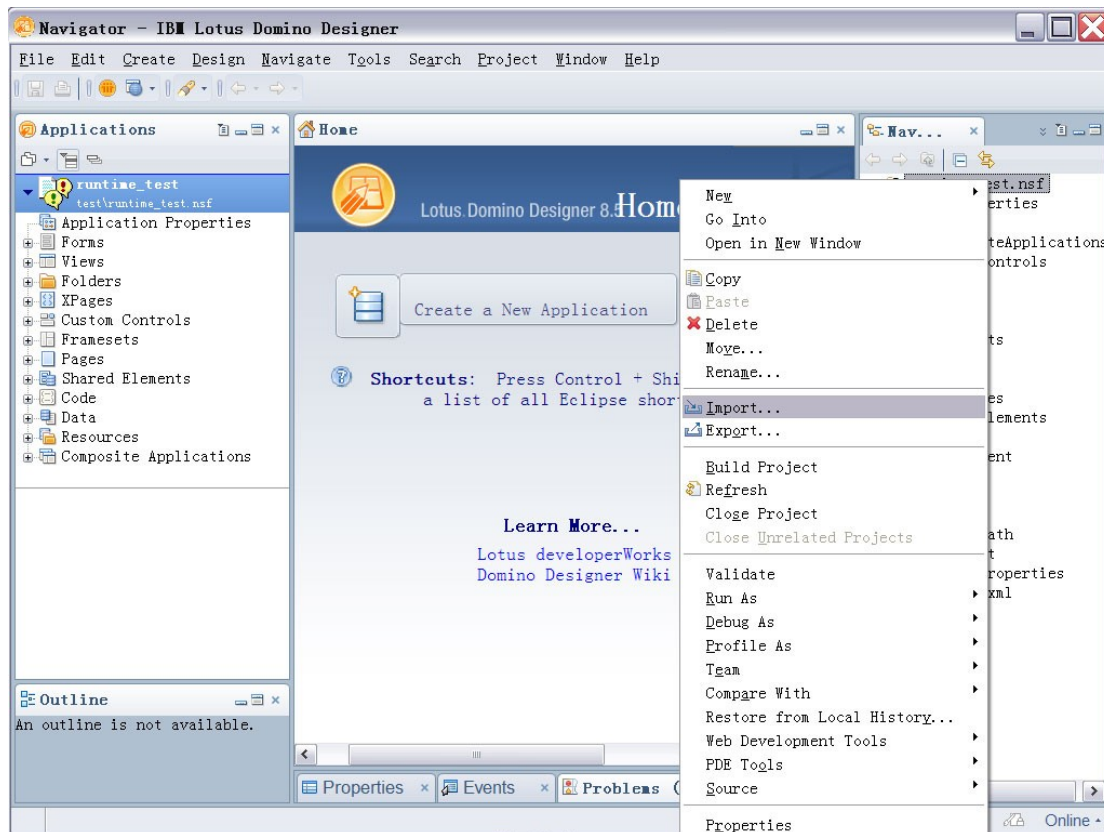
2. Bring up import/export wizard pages from context menu item.



3. Bring up import/export wizard pages from Navigator view.

Open Navigator view: Window → Show Eclipse Views → Other... → General → Navigator

After opening Navigator view in Domino perspective, you are able to import/export resources by right click on the project in the Navigator view and select Import/Export menu.



4. Bring up import/export wizard pages from Java perspective.

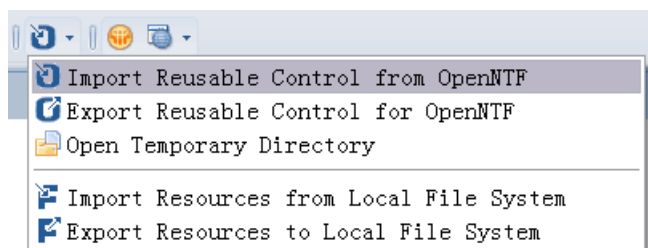
Switch to Java perspective: Window → Open Perspective → Other... → Java

After switching to Java perspective, you are able to import/export resources via Import/Export menu in the File top menu.

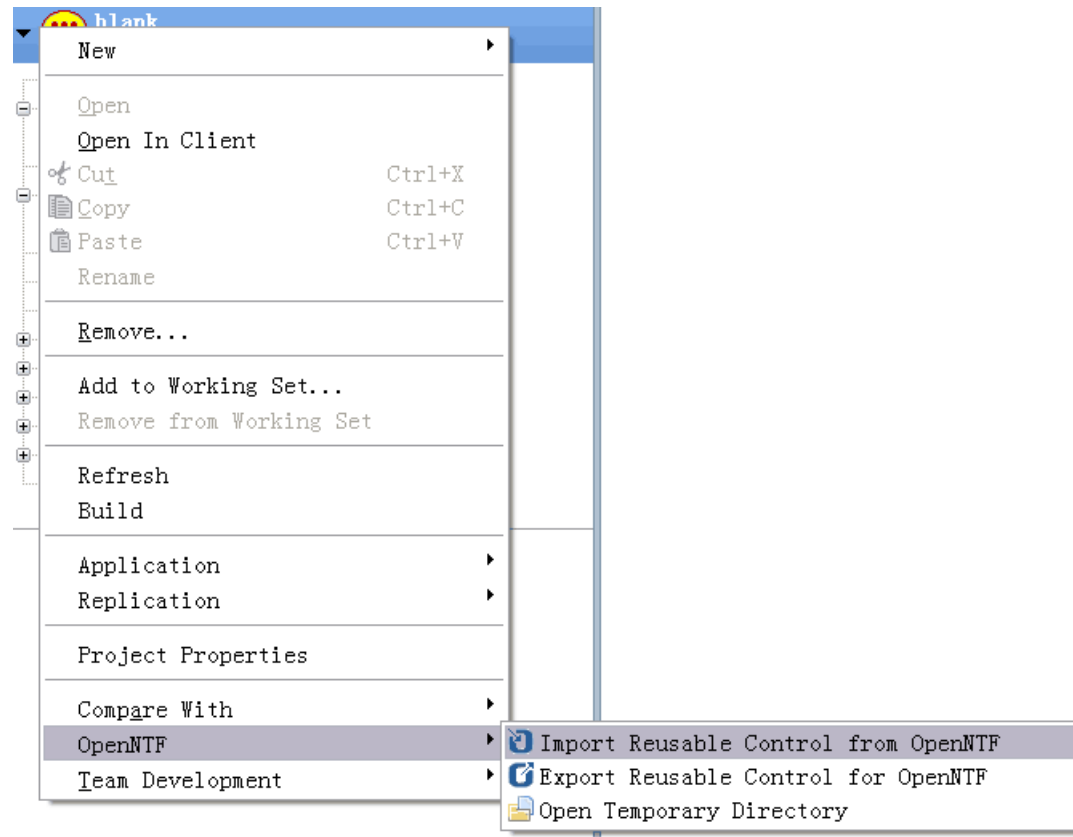
OpenNTF Export/Import

Import from OpenNTF

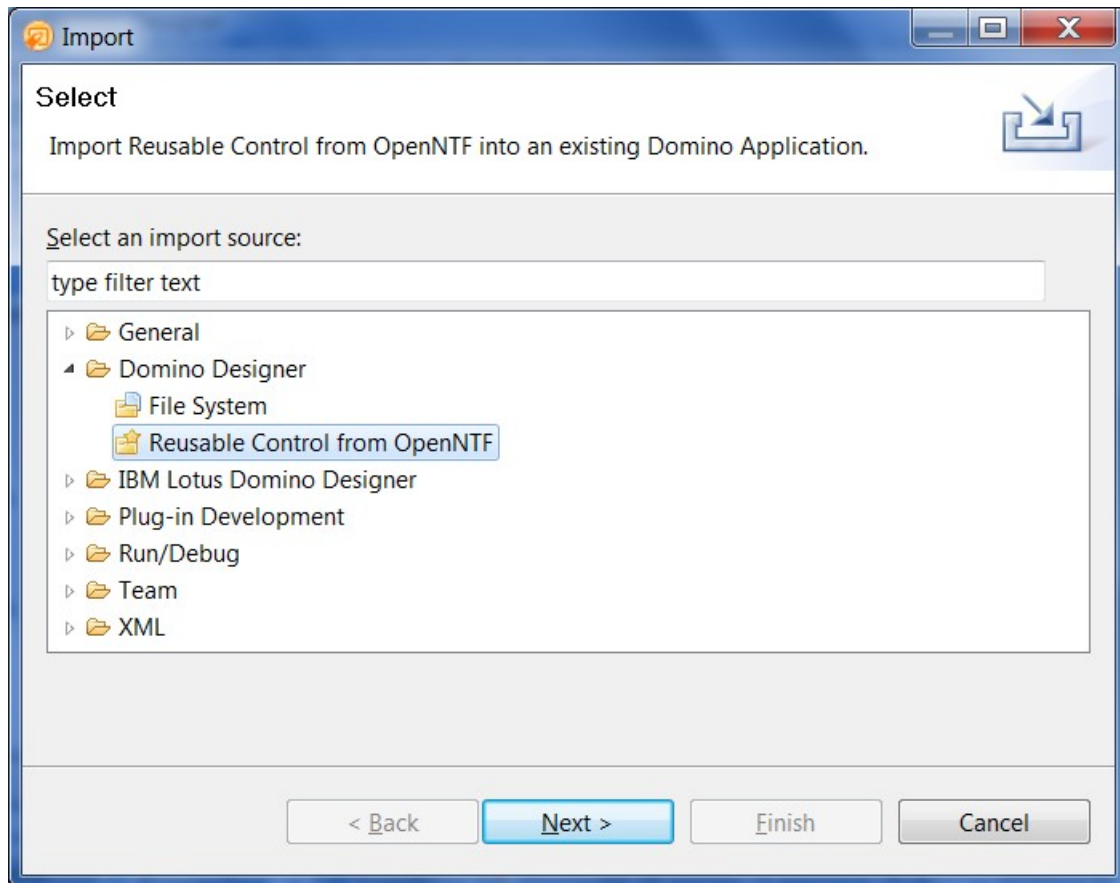
1. Click on the Designer Application, and then click the Import Reusable Control from OpenNTF icon on the toolbar.



Or right click on any Designer element and select the Import Reusable Control from OpenNTF menu item from the context menu.

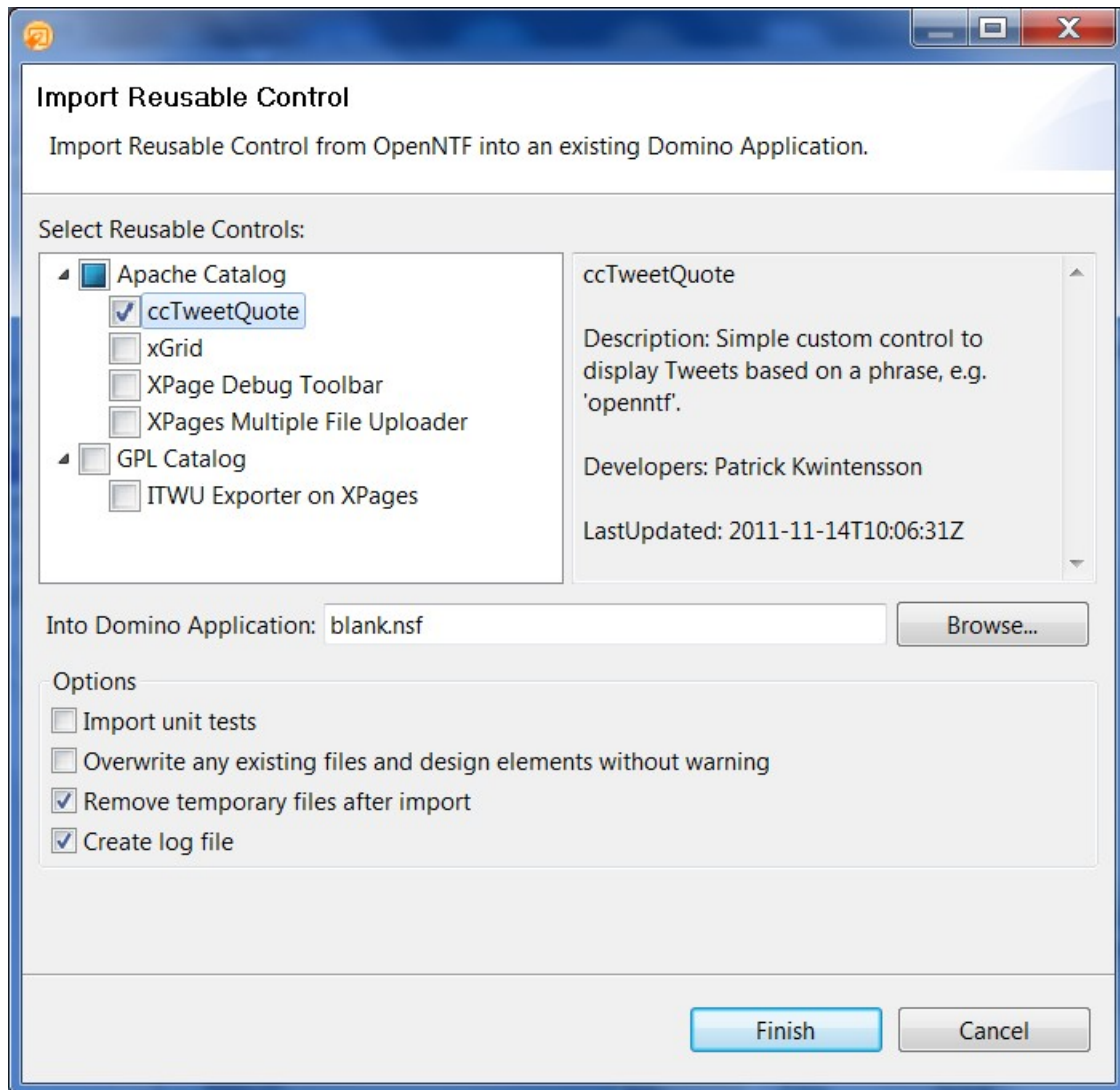


Or open the Eclipse Import wizard page, and select Domino Designer → Reusable Control from OpenNTF.



Click on “Next >”

2. You will get the Reusable Control Import Wizard as below.



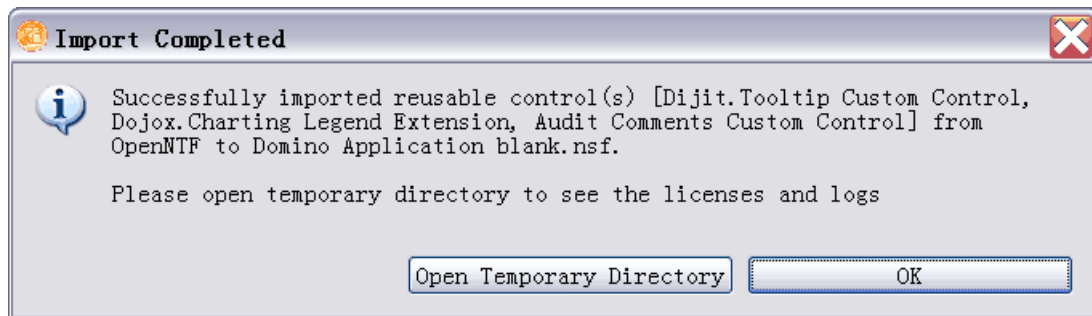
3. Check reusable controls to be imported. Selecting reusable control will display information of this control on the right side.

4. Select the destination Domino Application.

5. Select options:

- Import unit tests: Import demo files which are defined in the importlist.xml file.
- Overwrite any existing files without warning checkbox: import all selected resources.
- Remove temporary files after import: remove the .zip file and the exported files after successfully import the reusable control. **NOTE** that we will not remove the extracted folder because users have to read the LICENSE and NOTICE file in it.
- Create log file checkbox: an import log file will be created in temporary directory if you check on this option. You can click "Open Temporary Directory" menu to open this directory to check the log file.

- Click on "Finish". Importing work will be processed in the backend, and you will be notified when all work is done.



The logic of import is as following:

- Download the .zip file from OpenNTF to the temporary directory <TEMPDIR>.
- Extract the .zip file, assume it is test.zip, to <TEMPDIR>\test folder in the downloaded directory.
- Find the .nsf file in the extracted folder, assume it is control.nsf. And Find the import list file importlist.xml in this directory. If importlist.xml does not exist then generate a default file.

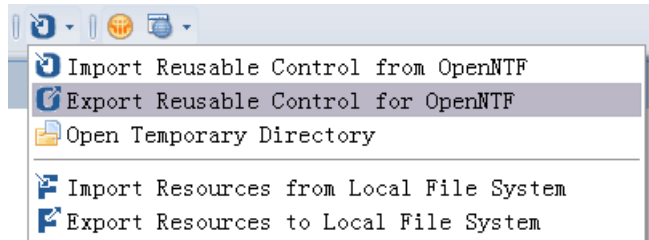
The default importlist.xml is as below:

```
<ImportList>
  <ImportFolders>
    <ImportFolder folder="XPages" />
    <ImportFolder folder="CustomControls" />
    <ImportFolder folder="Forms" />
    <ImportFolder folder="Views" />
    <ImportFolder folder="Folders" />
    <ImportFolder folder="Framesets" />
    <ImportFolder folder="Pages" />
    <ImportFolder folder="SharedElements" />
    <ImportFolder folder="Resources" />
    <ImportFolder folder="Code" />
    <ImportFolder folder="WebContent" />
  </ImportFolders>
  <ImportFiles>
    <ImportFile file=".classpath" />
  </ImportFiles>
</ImportList>
```

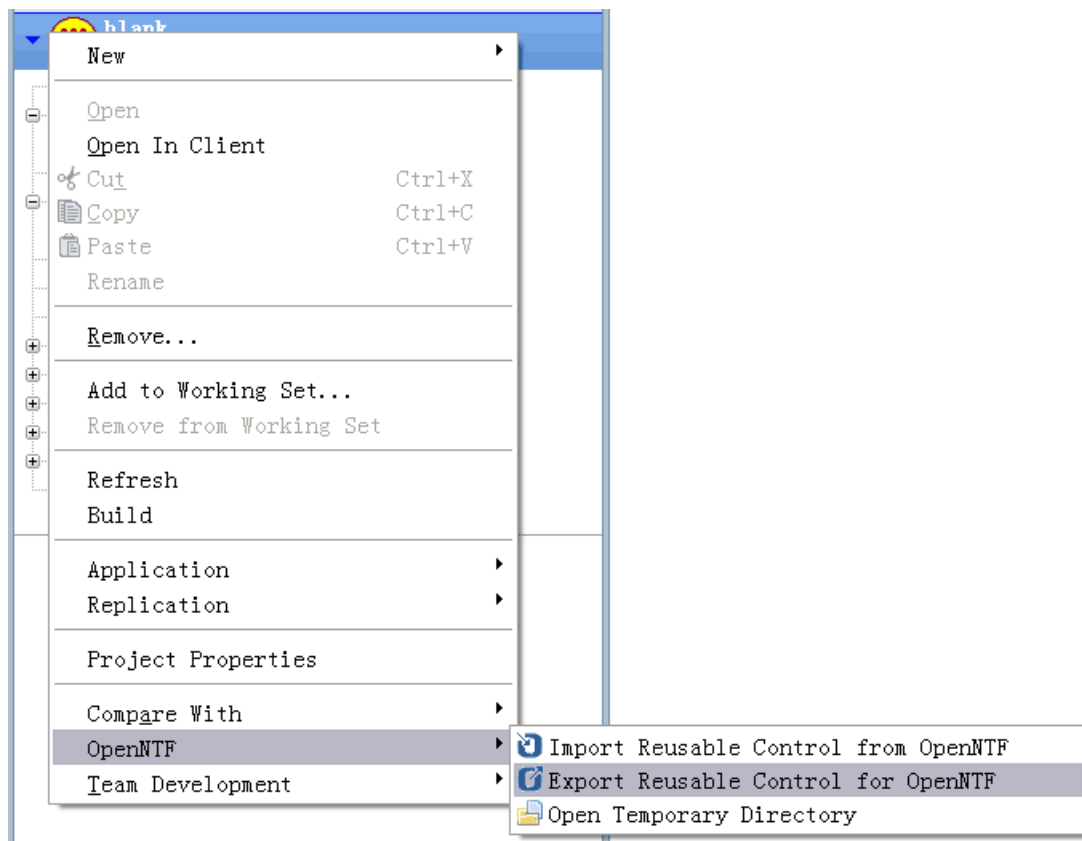
- Open control.nsf in workspace, and export resources specified in importlist.xml from control.nsf application to the directory <TEMPDIR>\control.nsf.
- Import all files from <TEMPDIR>\control.nsf directory to the specified domino application.
- Remove <TEMPDIR>\test.zip and <TEMPDIR>\control.nsf directory. We leave <TEMPDIR>\test directory because users have to read the LICENSE and NOTICE file.

Export for OpenNTF

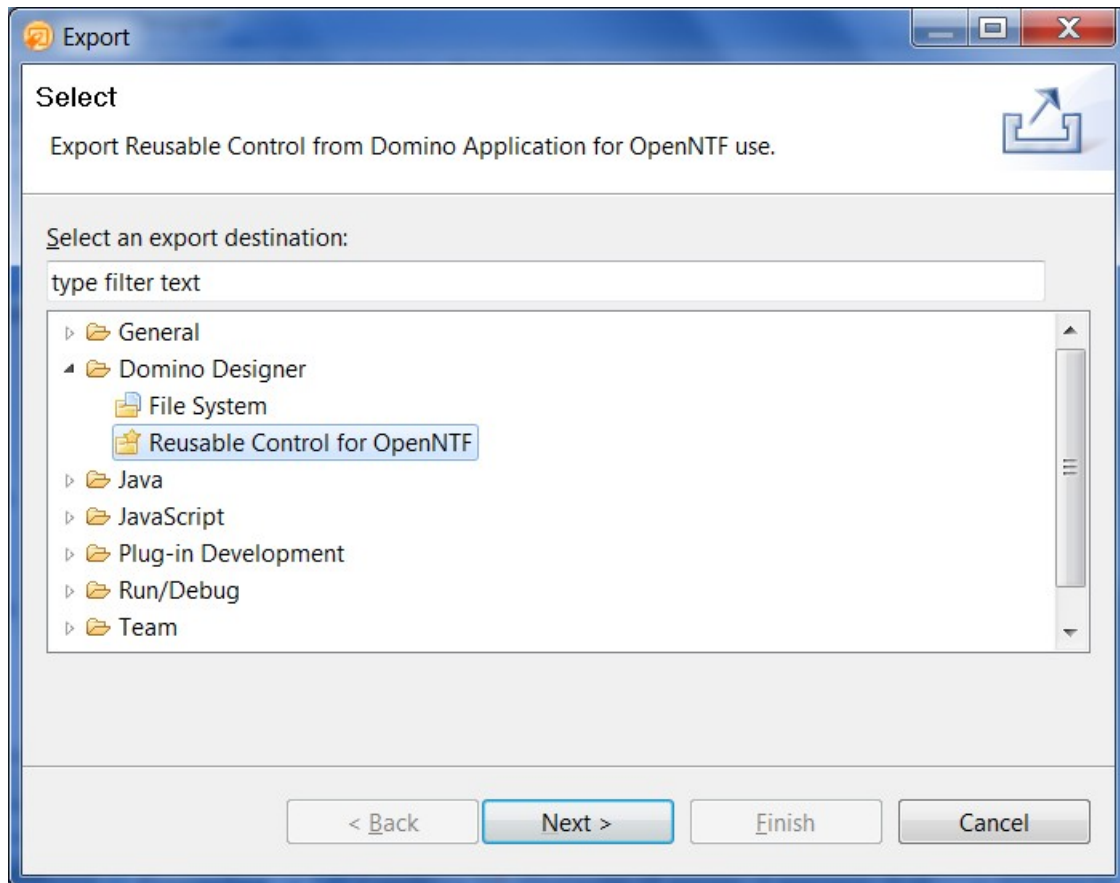
1. Click the Export for OpenNTF icon on the toolbar.



Or right click on a Designer element and select the Export for OpenNTF menu item from the context menu.

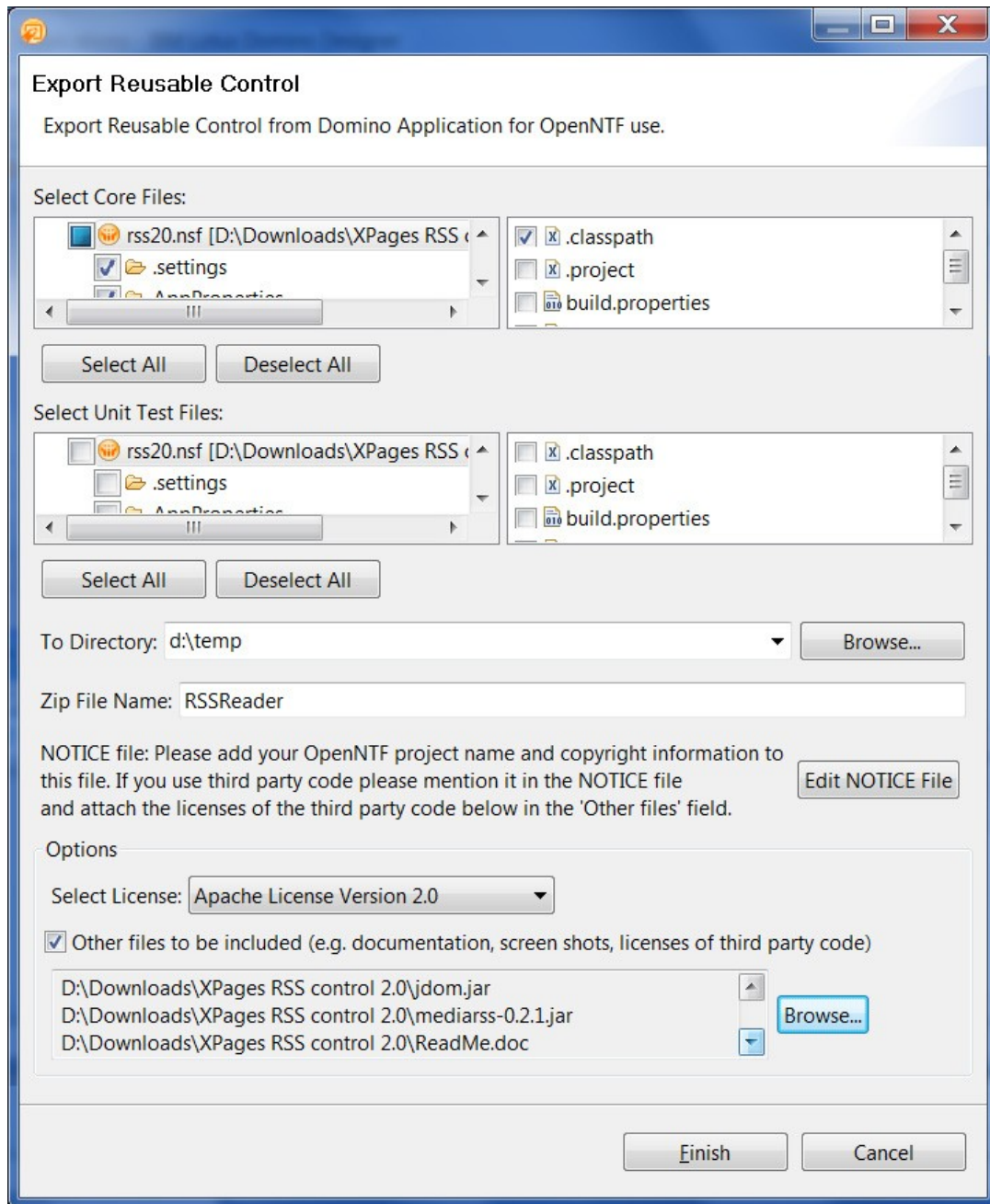


Or open the Eclipse Export wizard page, and select Domino Designer → Reusable Control for OpenNTF.



Click on “Next >”

2. You will get the Reusable Control Export Wizard as below.



3. Select core resources and unit test resources that need to be export from a Domino Designer Application. Select destination directory and type the target .zip file name.
4. Click Edit NOTICE File button, input your project name and copyright into the prompted NOTICE file and save it.
5. Select license in the options group. **NOTE** that If you use third party code please click “Edit NOTICE file” to mention this in the NOTICE file and attach the licenses of the third party code in the Other Files field.
6. Attach other files such as documentation or screenshots.

7. Click on “Finish”.

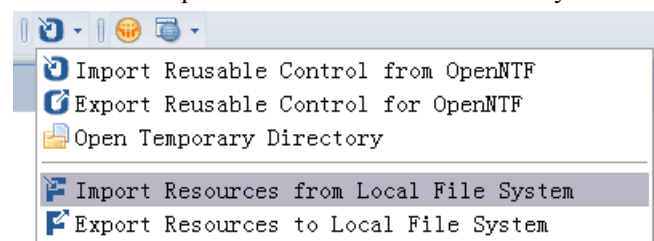
The logic of export is as following:

1. Create a directory d:\temp\RSSReader. Assume the destination directory is d:\temp and the .zip file name is RSSReader.zip.
2. Copy the Designer Application rss20.nsf to d:\temp\RSSReader directory.
3. Copy the specified LICENSE file to d:\temp\RSSReader.
4. Clicking “Edit NOTICE file” will create a temporary NOTICE file in directory data\workspace\metadata\plugins\com.ibm.designer.extensibility.copyfile\temp and edit this file, and clicking finish will copy this temporary NOTICE file to d:\temp\RSSReader.
5. Create an xml file importlist.xml in d:\temp\RSSReader.
6. Copy other files listed in the other files field to d:\temp\RSSReader.
7. Compress d:\temp\ RSSReader to RSSReader.zip.
8. Delete d:\temp\ RSSReader directory.

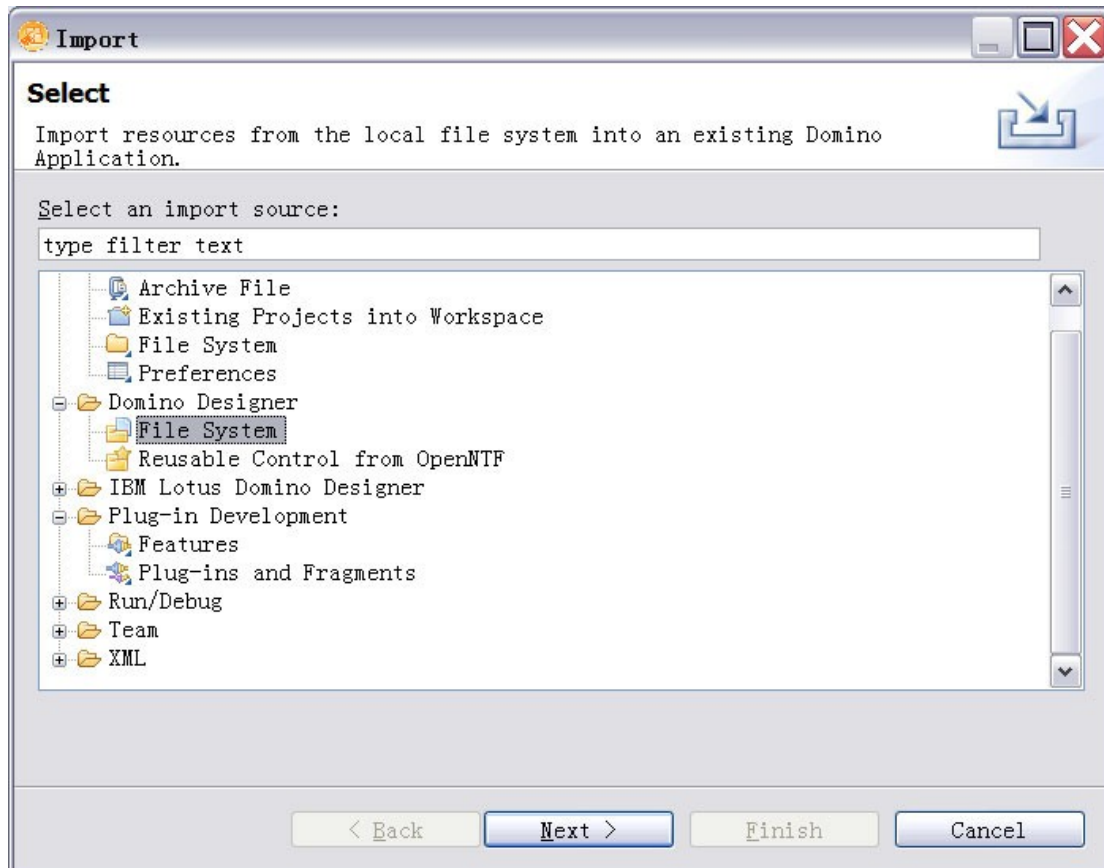
Local File System Export/Import

Import from Local File System

1. Click the Import Resources from Local File System icon on the toolbar.

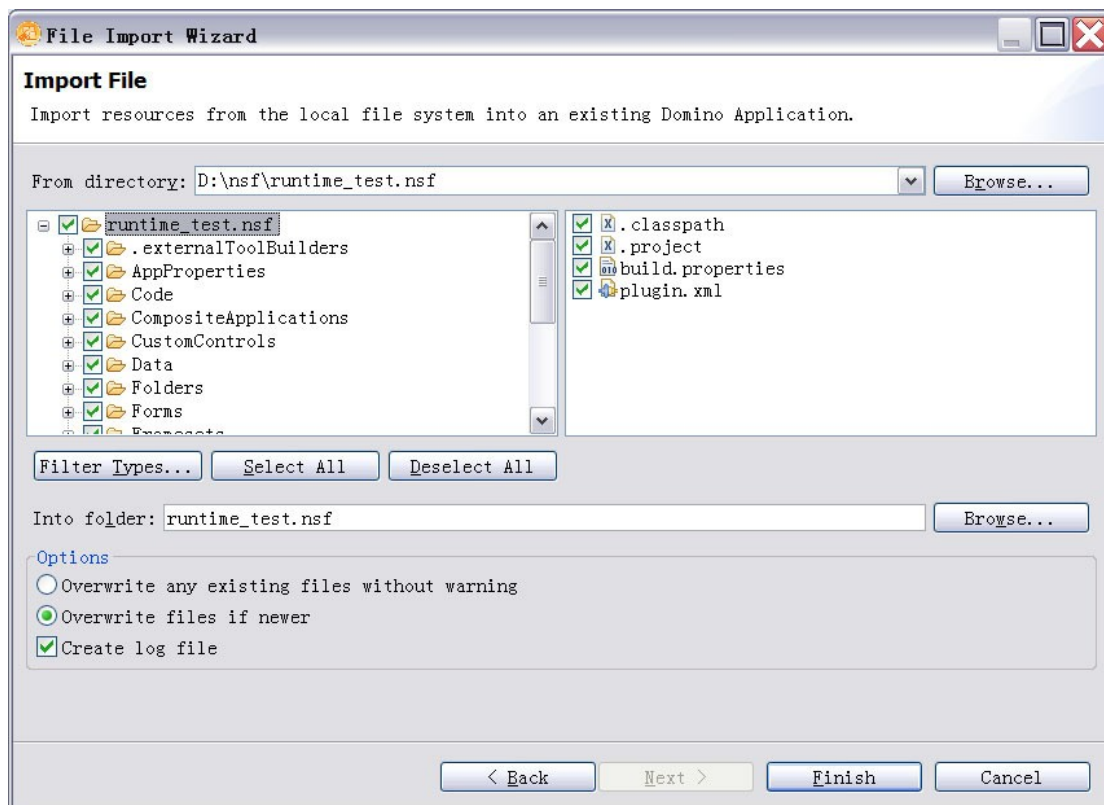


Or open the Eclipse Import wizard page, and select Domino Designer → File System.



Click on “Next >”

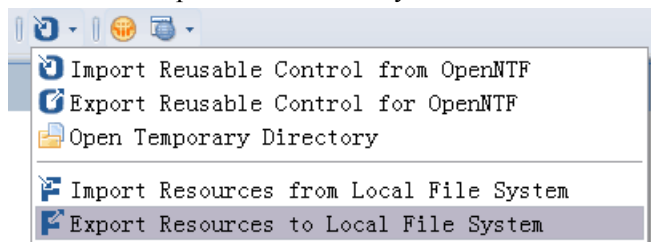
2. You will get the File Import Wizard as below.



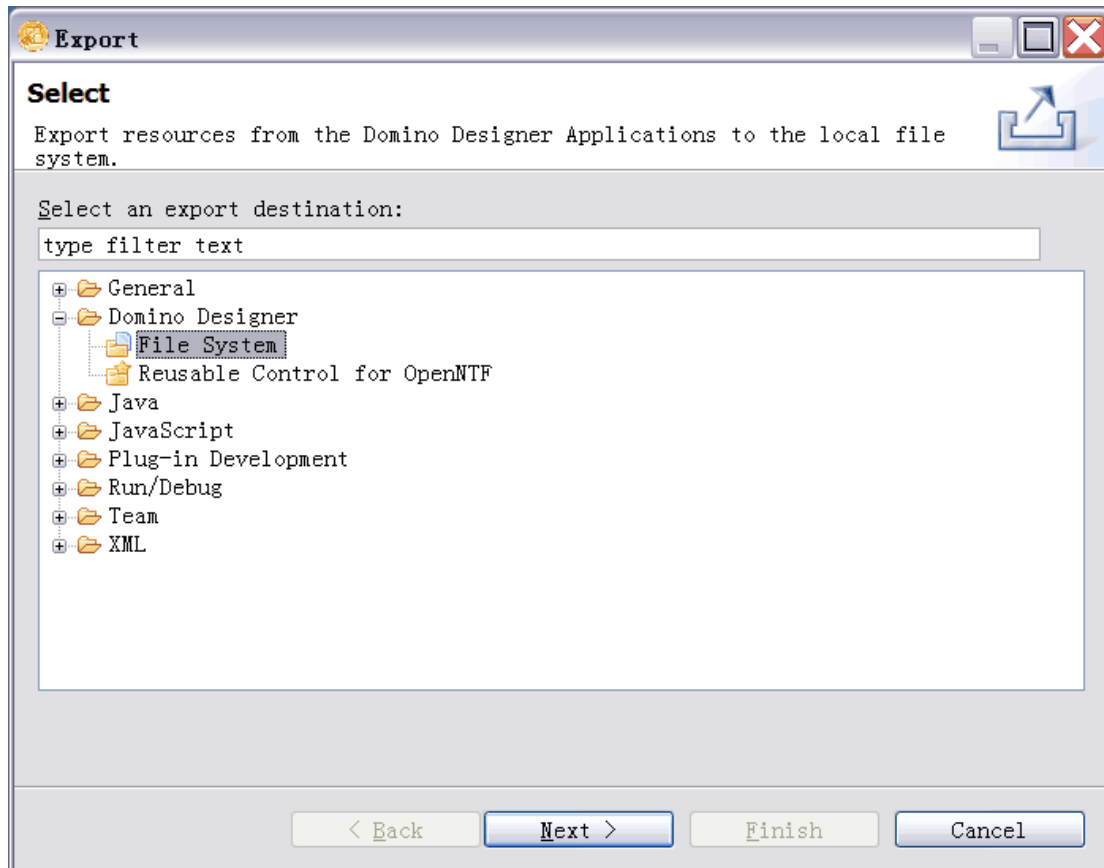
3. Select source directory.
4. Select source files to be imported. All types of designer elements files are able to be imported by this plug-in.
5. Select the destination directory.
6. Select options:
 - Overwrite any existing files without warning radio: import all selected resources.
 - Overwrite files if newer radio: only import resources when they are newer than the ones in the Domino Designer project or the ones do not exist in the project.
 - Create log file checkbox: an import log file will be created in the parent folder of the source directory if you check on this option.
7. Click on “Finish”.

Export to Local File System

1. Click the Export to Local File System icon on the toolbar.

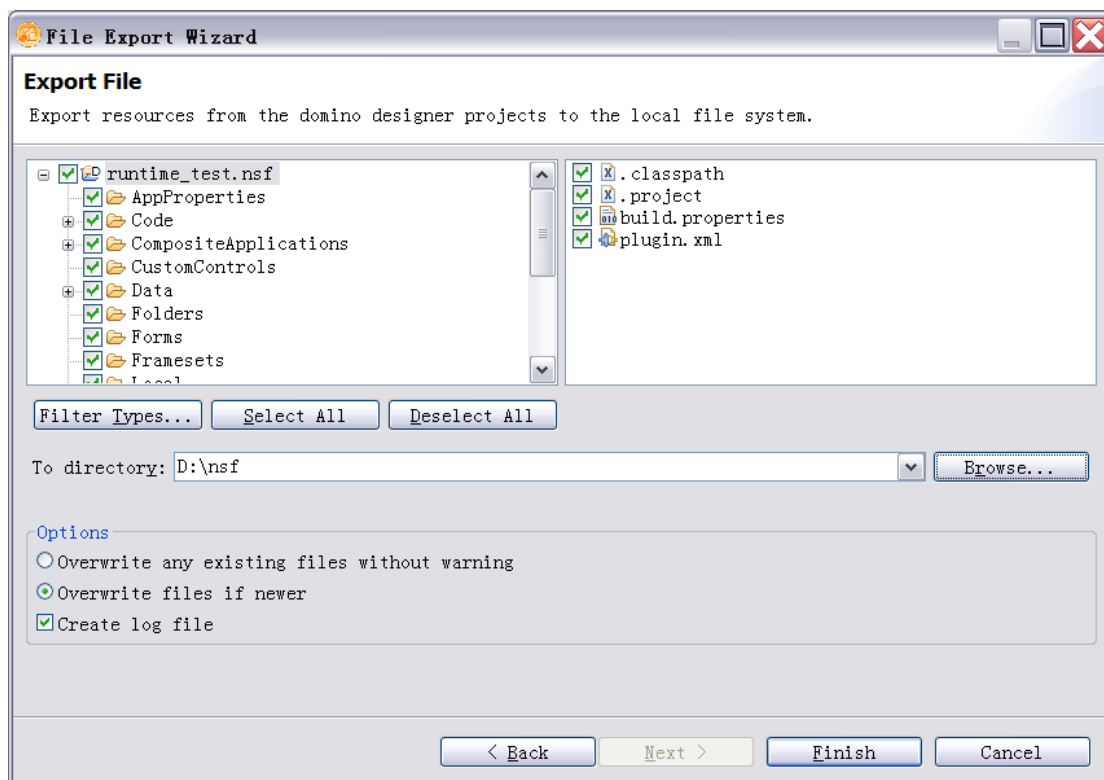


Or open the Eclipse Export wizard page, and select Domino Designer → File System.



Click on “Next >”

2. You will get the File Export Wizard as below.



3. Select source files to be exported. All types of designer elements are able to be exported by the original Eclipse export wizard and this plug-in just changes the timestamp of the target file to the local timestamp of the designer element.

4. Select the destination directory. A folder with the same name of the designer project will be created in this destination directory. And sub-folders will be created to follow the hierarchies of each exported resource's complete path.

5. Select options:

- Overwrite any existing files without warning radio: export all selected resources.
- Overwrite files if newer radio: only export resources when they are newer than the ones in file system or the ones do not exist in file system.
- Create log file checkbox: an export log file will be created in the destination directory if you check on this option.

6. Click on “Finish”.

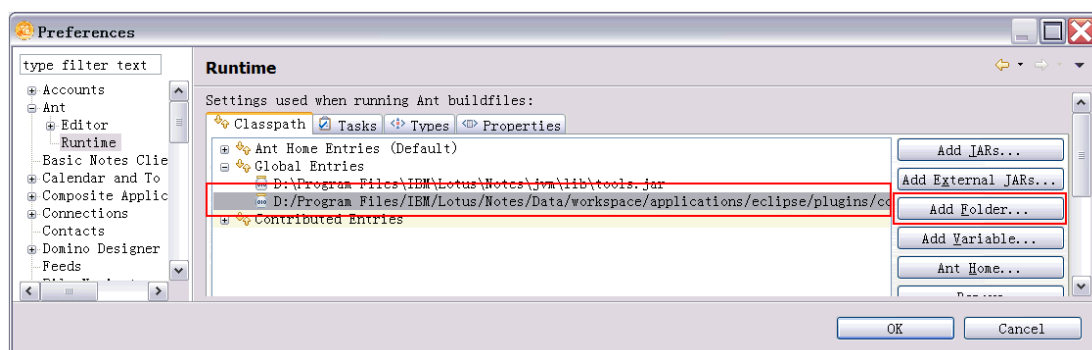
Export/Import using ANT tasks

This feature allows users export/import resources using ANT tasks without UI. **NOTE** that this only works for ANT builder, NOT works when manually executing an ANT script.

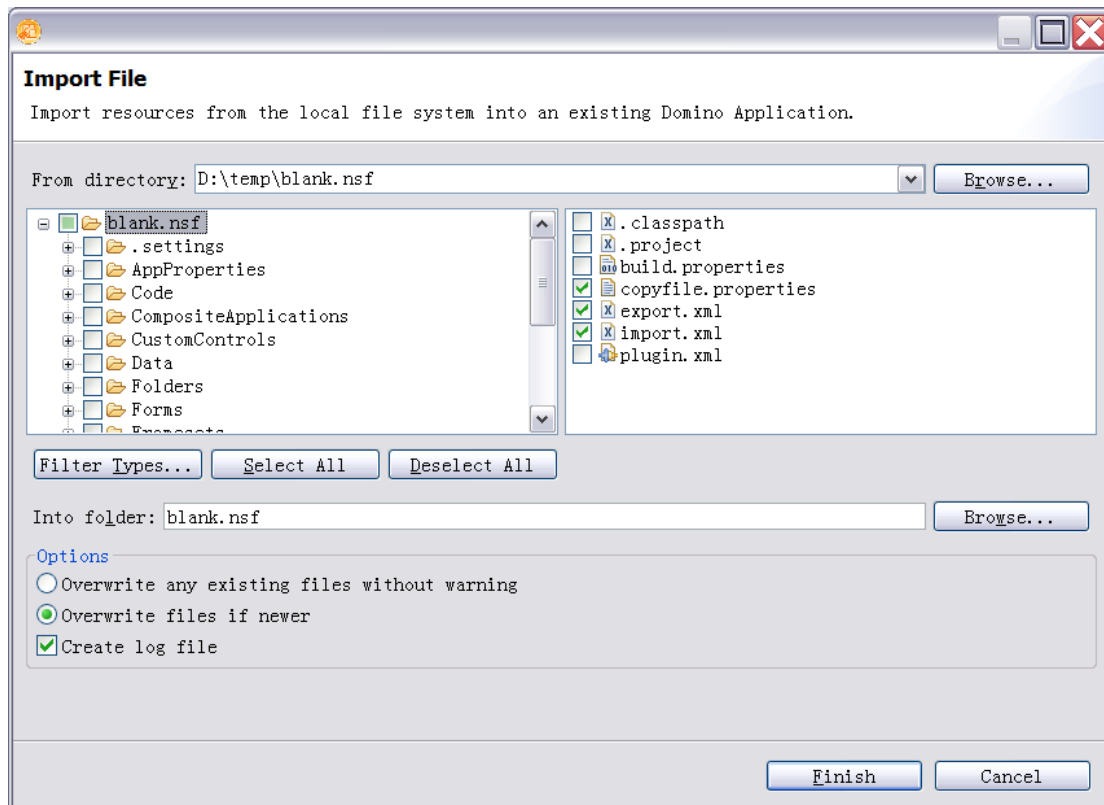
Configuration

1. Add ImportTask and ExportTask definition into the ANT configuration.

Click File -> Preferences menu, and navigate to Ant->Runtime preference page, add <NotesData> workspace\applications\eclipse\plugins\com.ibm.designer.extensibility.copyfile_2.0.0 folder to Global entries section.

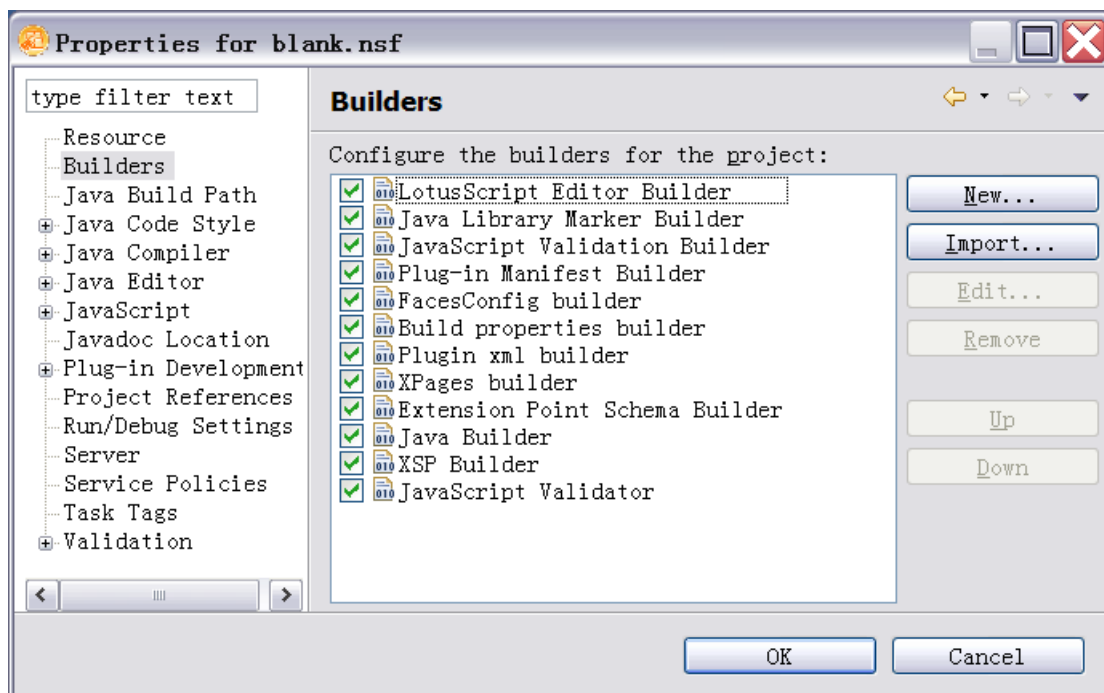


2. Import your ANT scripts and properties to your Designer Application via this import plug-in.

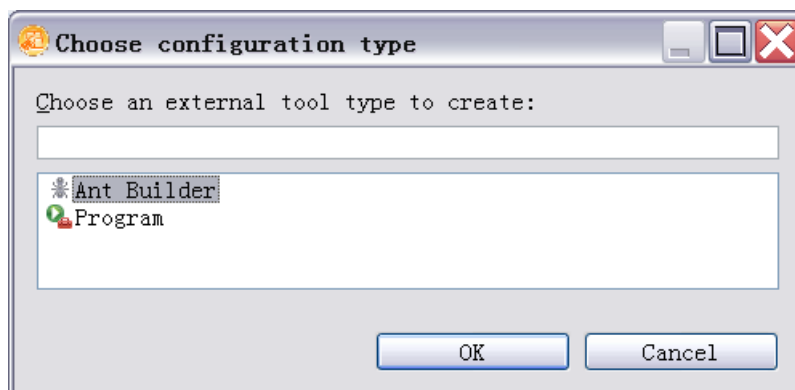


3. Create an ANT builder.

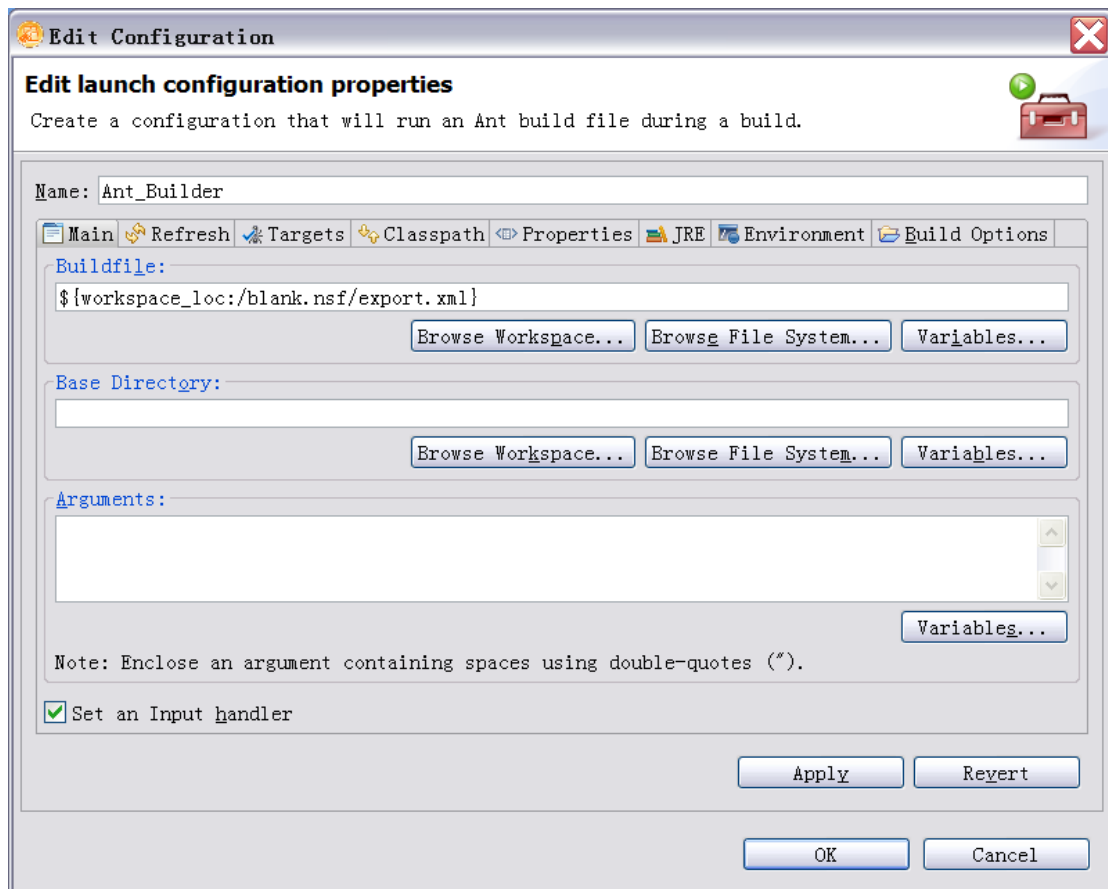
Click Project->Properties menu



Click New... button and Select Ant Builder.



Type in the builder name and chose the buildfile.



Edit Configuration

Edit launch configuration properties

Create a configuration that will run an Ant build file during a build.

Name:

Main Refresh Targets Classpath Properties JRE Environment Build Options

Buildfile:

Browse Workspace... Browse File System... Variables...

Base Directory:

Browse Workspace... Browse File System... Variables...

Arguments:

Variables...

Note: Enclose an argument containing spaces using double-quotes (").

☒ Set an Input handler

Apply Revert

OK Cancel

Click OK. Now your ANT script will be executed when you build this project.

Sample Export ANT script

Below is a sample export ANT script:

```
<project name="ExportTask" default="main" basedir=". ">
<taskdef name="eclipseexport"
classname="com.ibm.designer.extensibility.copyfile.anttasks.ExportTask"
/>
<property file="copyfile.properties" />
<target name="main">
    <eclipseexport todir="${export.todir}" project="${project}"
    overwriteolder="yes" createlog="yes" debug="yes">
        <fileset dir="${project}/Views" />
        <filelist dir="${project}/XPages" files="test.xsp, test1.xsp" />
    </eclipseexport>
</target>
</project>
```

where,

- todir: The destination directory.
- project: The project to be exported.
- overwriteolder: Whether to only overwrite the older files.
 - yes means to only overwrite the older files;
 - no means to overwrite all files in file system.
 - The default value is yes.
- createlog: Whether to create log file. The default value is yes.
- debug: Whether to print out debug information. The default value is false.

You can define these variables in a properties file.

NOTE that Export Task supports fileset and filelist, but does not support include/exclude patterns in fileset.

Sample Import ANT script

Below is a sample import ANT script:

```
<project name="ImportTask" default="main" basedir=". ">
<property file="copyfile.properties" />
<taskdef name="eclipseimport"
classname="com.ibm.designer.extensibility.copyfile.anttasks.ImportTask"
/>
<target name="main">
  <eclipseimport fromdir="${import.fromdir}" project="${project}"
  overwriteolder="yes" createlog="yes" debug="yes">
    <fileset dir="${import.fromdir}/${project}/XPages"
    includes="**/*.xsp" />
    <fileset dir="${import.fromdir}/${project}/Views" />
    <filelist dir="${import.fromdir}/${project}/XPages"
    files="test.xsp,test1.xsp" />
  </eclipseimport>
</target>
</project>
```

where,

- fromdir: The source directory.
- project: The project to be exported.
- overwriteolder: Whether to only overwrite older files.
 - yes means only overwrite older files;
 - no means overwrite all files in the project.
 - The default value is yes.
- createlog: Whether to create log file. The default value is yes.
- debug: Whether to print out debug information. The default value is false.

You can define these variables in a properties file.

NOTE that Import Task supports fileset and filelist, and supports include/exclude patterns in fileset.