

MIGRATING INSTALLSHIELD MULTIPLATFORM PROJECTS TO INSTALLANYWHERE

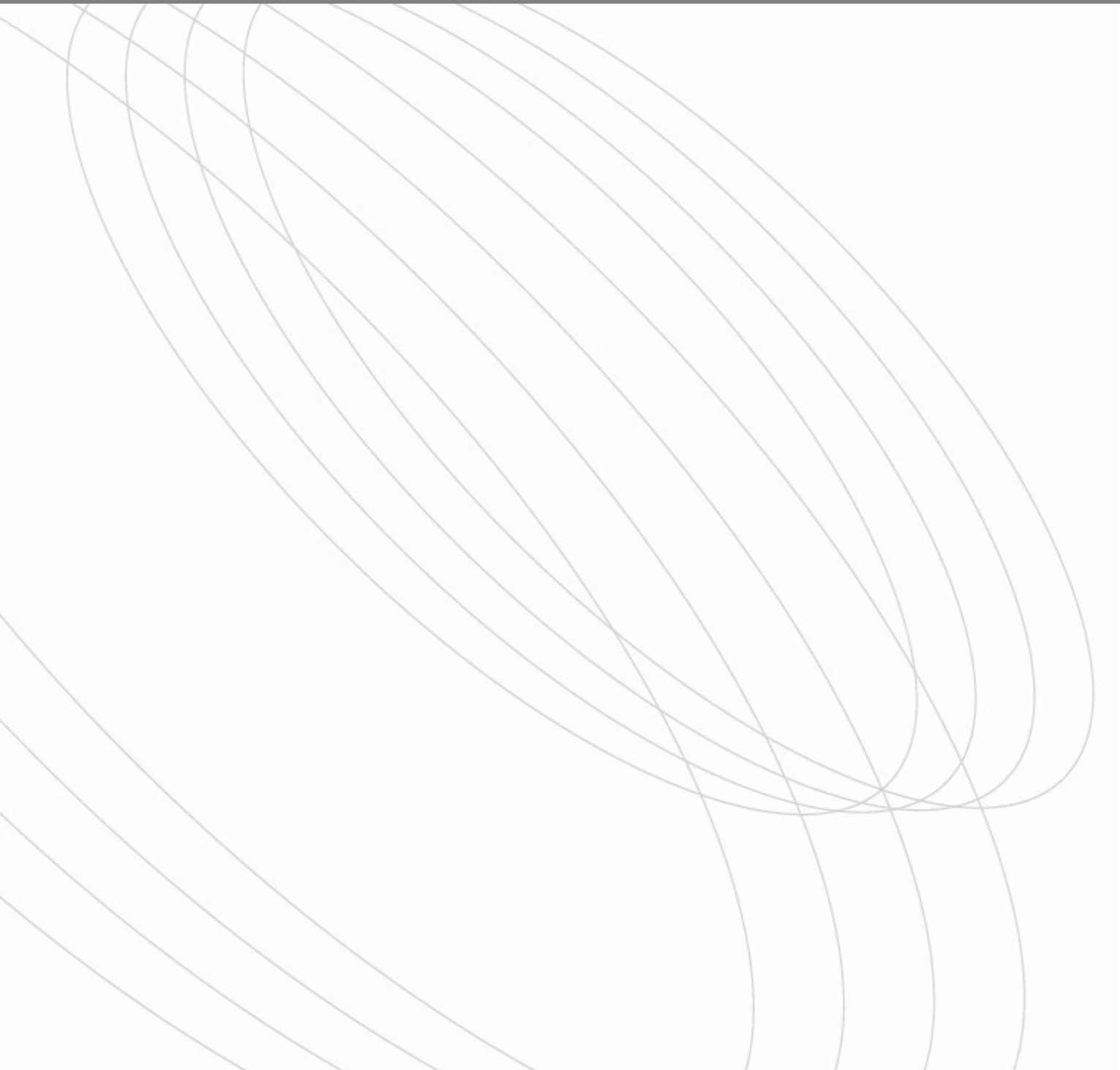


TABLE OF CONTENTS

Introduction.....	3
InstallShield MultiPlatform and InstallAnywhere Differences	3
Migration Best Practices.....	4
The Migration Process	5
Creating a Project Manifest in ISMP	5
Importing a Project Manifest Using InstallAnywhere.....	6
Additional Considerations.....	9
Conditions	9
Custom Code.....	9
Dialogs.....	12
Custom Dialogs	14
Maintaining Existing Installations	14
Summary	16

INTRODUCTION

In order to ease the transition between products for our customers, we've added Project Manifest functionality to the InstallShield MultiPlatform (ISMP) and InstallAnywhere (IA) products. You can download the Project Manifest plug-in from the Macrovision Support Site at <http://www.macrovision.com/support/index.shtml>.

This document outlines a typical use case for migrating ISMP project information and functionality to an InstallAnywhere project. This plug-in is designed to export as much ISMP functionality as possible to an InstallAnywhere project. However, this tool has some limitations. Macrovision has provided support for the ISMP services layer as well as certain utilities with InstallAnywhere to facilitate the migration of projects from InstallShield MultiPlatform to InstallAnywhere. Due to the fact that InstallShield MultiPlatform permitted extensive customization, including the creation of custom Java beans, it is not possible to provide tooling that will guarantee an automated migration of projects from ISMP to IA. This document also highlights the differences between the two products in regards to project settings and explains what functionality is and is not migrated.

This document begins with a table that displays features common to both products as well as caveats. Some best practices are listed that should be considered before exporting your project with the Manifest plug-in. The next section, **The Migration Process** walks through the export process of a sample InstallShield MultiPlatform project as well as the import process in InstallAnywhere. Finally in the **Additional Considerations** section, the process of converting custom code and dialogs is explained.

In general the major steps of converting your project are as follows:

- Install the plug-in for InstallShield MultiPlatform 11.5 SP1.
- Use the Project Manifest tool to export your project.
- Import the project manifest into InstallAnywhere.
- Migrate your custom beans to the InstallAnywhere CustomAction class type.
- Add the required dialogs to your project.

InstallShield MultiPlatform and InstallAnywhere Differences

Since you are porting from one product to another, there are some caveats to consider. Not all of the functionality you have in your ISMP project is migrated to the Project Manifest. None of your custom code is migrated to the Project Manifest. The table below explains which functionality is migrated and which is not.

ISMP	PROJECT MANIFEST	INSTALLANYWHERE
Project Info (name, version)	Project Info	Project Info
Uninstaller Information*	Not Supported	Not Supported
Update Service Information	Update Service Information	Update Service Information
Product Info	Product Info	Product Info

ISMP	PROJECT MANIFEST	INSTALLANYWHERE
Language Support	Language Support	Language Support
Platform Support	Platform Support	Platform Support
Features	Features	Features
Components	Components (Converted to DIMs)	DIM References/Components and File actions
Assemblies	Assemblies (as child manifest)	DIM References added to the Install task
Product Actions* except:	Product Actions (add to DIM files)	Product Actions (located in DIM References)
Ascii File Update	Ascii File Update	Ascii File Update
Environment Variable Update	Environment Variable update	Environment Variable update
Create Folder	Create Folder	Create Folder
Windows Registry Update	Windows Registry Update	Windows Registry Update
Wizard Actions*	Not Supported	Not Supported
Custom Actions*	Not Supported	Not Supported
Dialogs*	Not Supported	Not Supported
Build Variables	Build Variables	Source Paths
Runtime Variables	Runtime Variables	Magic Folder Variables
Installer Launchers*	Not Supported	Not Supported
Product Launcher	Product Launcher	LaunchAnywhere
Short Cuts	ShortCuts	Shortcut
JVM Resolution*	Not Supported	Not Supported
Custom Beans*	Not Supported	Not Supported

* Not exported to Project Manifest

Migration Best Practices

Since there are differences between the two products, consider the following points before attempting to migrate your project.

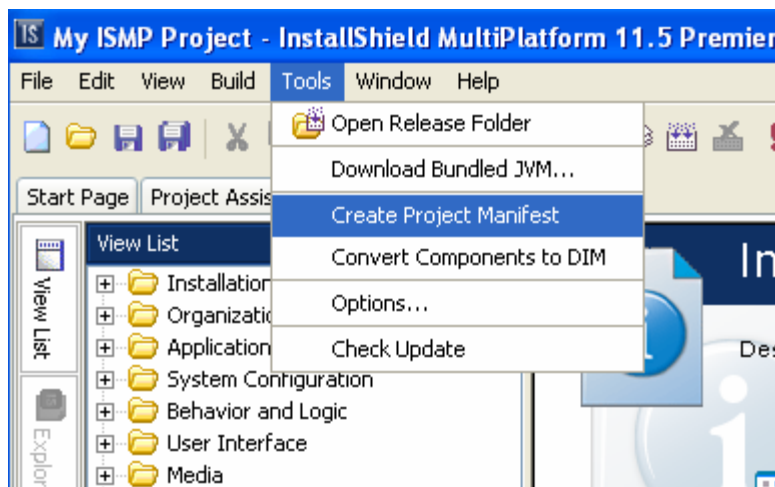
- Always create backup copies of your projects before creating a manifest.
- Only Universal, 5.x, and Condenser project types support the Project Manifest plug-in.
- In order to migrate a Suite, you must create a Project Manifest for each of the child projects.

THE MIGRATION PROCESS

Creating a Project Manifest in ISMP

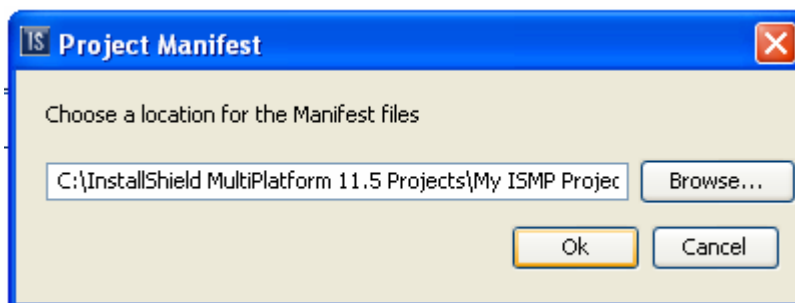
In order to create a Project Manifest from an InstallShield MultiPlatform project, you must download and install the Project Manifest plug-in. If you do not see an entry for “Create Project Manifest” in the Tools menu, you have not downloaded the plug-in. Download it from the Macrovision support site mentioned in the Introduction. Once the plug-in is installed, you can create the manifest by following the steps below.

- 1 Open an existing ISMP project or create a new Universal Installer Project.
- 2 From the **Tools** menu, choose **Create Project Manifest**.



ISMP—Creating a Project Manifest.

- 3 Specify a path for the Project Manifest file when prompted.



ISMP—Specifying a path for the Project Manifest.

- 4 Click **OK**. Progress regarding the creation of the manifest file is shown in the Output window at the bottom of the IDE.

During the creation process, all components are converted to Developer Installation Manifest (DIM) files. Only product actions that are supported by DIM files are preserved in the Project Manifest file. Project information such as project name and author

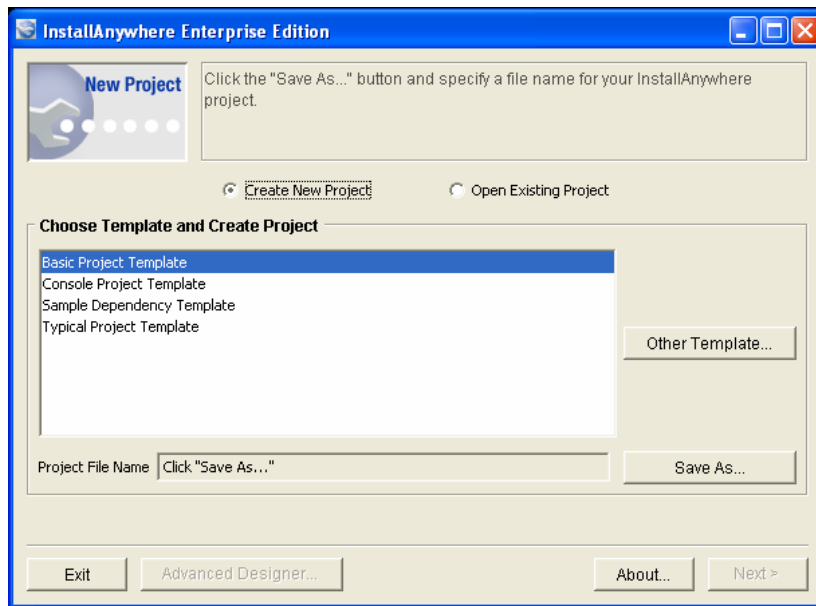
are then stored in the manifest followed by product information like product name and version. Settings for each of the components in the project are stored in the Project Manifest file. Platform and language settings along with conditions that are supported by DIM files are also added to the Project Manifest file.

Once the creation process has finished, a folder structure is created at the location you specified. This folder structure contains the Project Manifest file along with a DIM file for each of the components in the project. If your project contained assembly references, a subfolder for each of them is created that contains a Project Manifest file and DIMs for that assembly's product tree.

Importing a Project Manifest Using InstallAnywhere

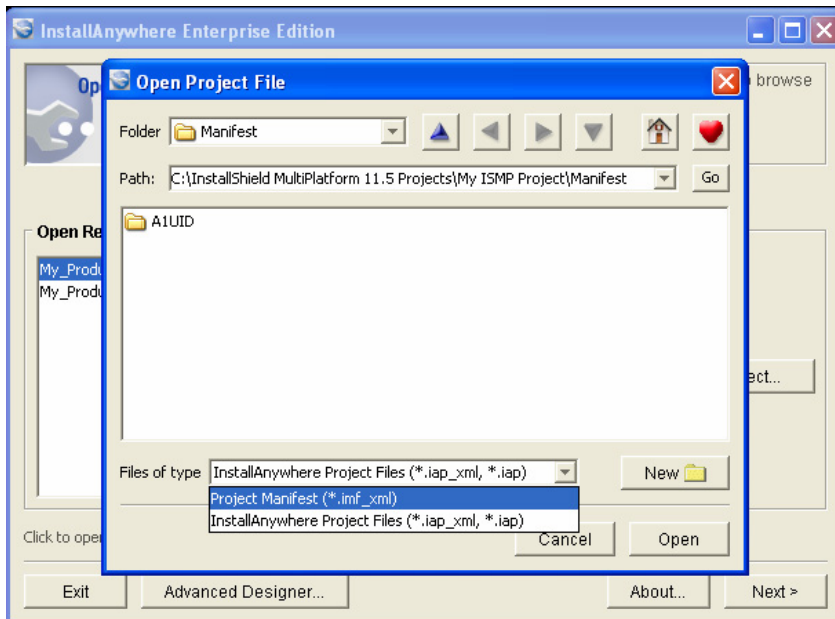
Project Manifests can only be applied to new InstallAnywhere projects. When importing the Project Manifest, InstallAnywhere creates the features, components and install sets with the same settings as the original ISMP project. You can import a Project Manifest using the following steps.

- 1 Open InstallAnywhere and choose the **Open Existing Project** option from the New Project wizard.



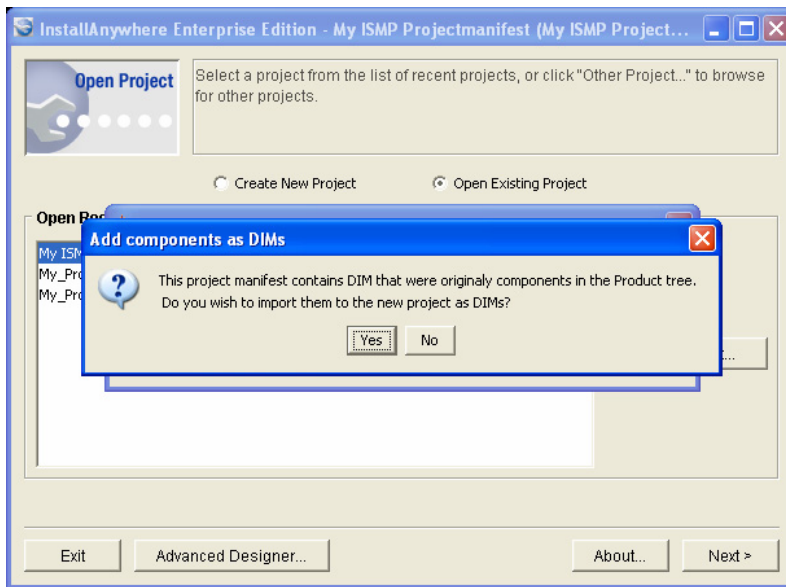
InstallAnywhere—Creating a new project to import the Project Manifest.

- 2 Click **Other Project** and browse to the Project Manifest file.



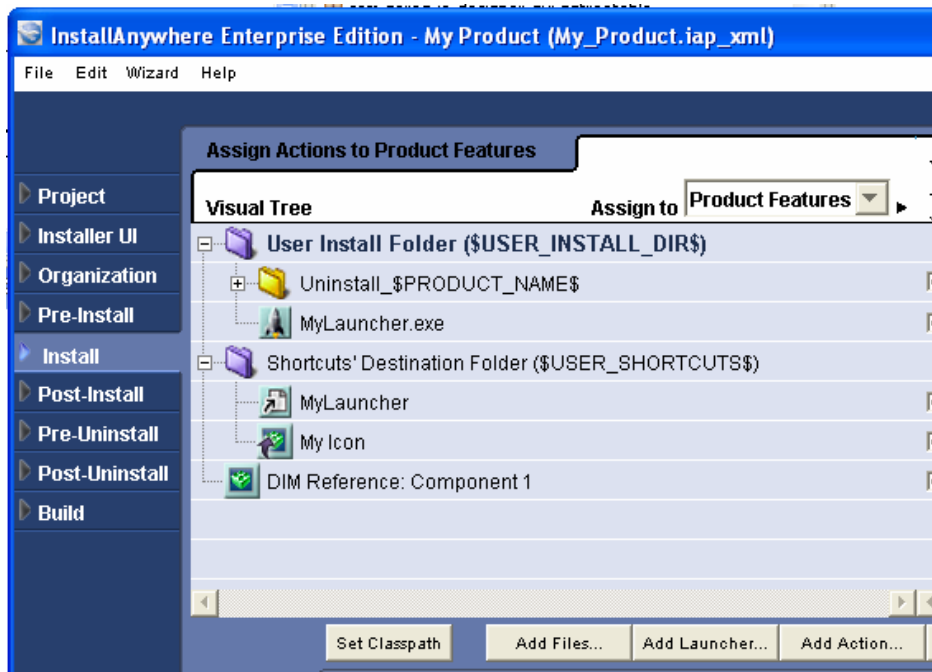
InstallAnywhere—Opening the Project Manifest.

- 3 Click **Open**. If your project contained DIMs along with components, you are prompted to convert components back to components or import everything as DIMs.



InstallAnywhere—Choosing to import the Project Manifest as DIMs.

- 4 Click **Yes** to import everything as DIMs then click **Advanced Designer** to open the InstallAnywhere project with the changes.



InstallAnywhere—DIM references appear in the Advanced Designer's Install task.

Once a new project has been specified, the Project Manifest file that was selected for the existing project is applied to the template that was chosen. Refer the Install task in the InstallAnywhere IDE to see the DIM references, shortcuts and launchers that were migrated from your ISMP project.

ADDITIONAL CONSIDERATIONS

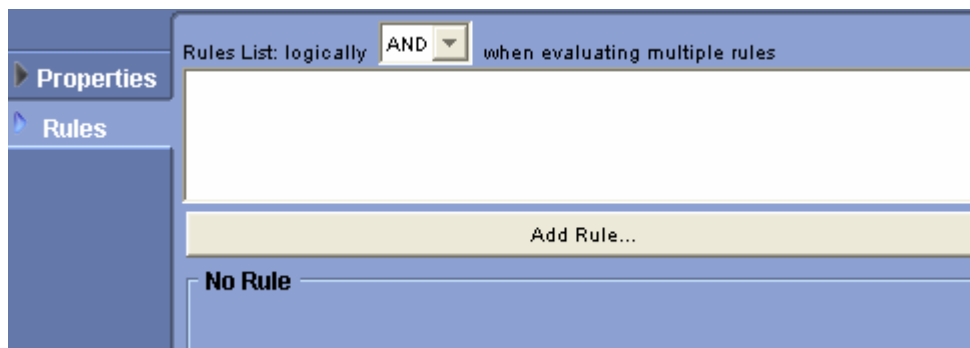
Many of the items not transferred from your original InstallShield MultiPlatform project can be manually created in InstallAnywhere. This section explains how to accomplish this for common components found in a typical MultiPlatform project.

Conditions

In order to maintain the same standard logic used in the ISMP project you need to make use of Rules in your new IA project. Many of the standard conditions found in ISMP have an equivalent Rule in IA such as the Check File\Folder Exists rule and the Check Platform rule. To add a rule to an item in your project, you simply select the item and choose the Rules tab in the bottom half of the IDE, then click the Add Rule button and choose a rule from the list. Rules can be applied to the most of the same items as you would in MultiPlatform.

Properties	
[-] Standard Properties	
Active	True
Display Name	Test Product
Install Location	\$D(install)/\$P(vendor)/\$P(displayName)
[-] Conditions	
Conditions	No Condition

ISMP—Condition property.



InstallAnywhere—Rule properties.

Custom Code

To convert custom logic, you will have to extend the CustomCodeRule class in IA instead of the ProductBeanCondition or WizardBeanCondition in the case of custom conditions. Your custom rule should overwrite the evaluateRule() method. A custom rule can be placed anywhere a standard rule can be added. Just choose the Evaluate Custom Rule from the rules list to add it to an item.

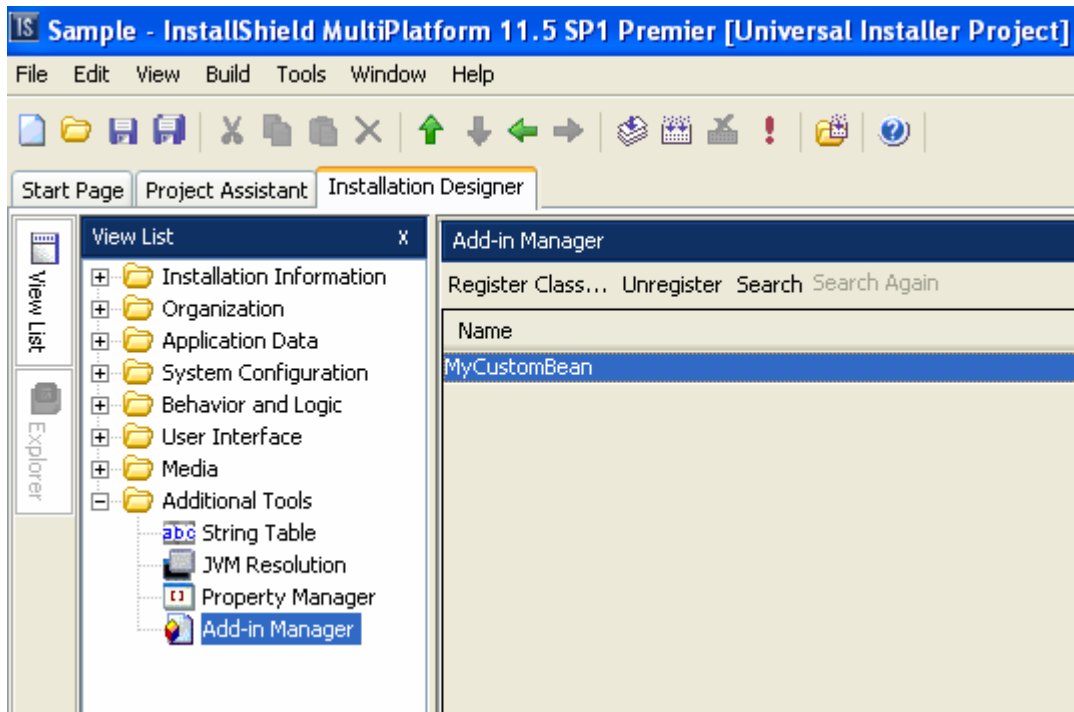
Additional Considerations

If your ISMP project contains custom beans that use the ISMP Services, these may convert over as well. The File service, Security service, SystemUtil service, Win32 service, Win32 Registry service and the Windows Account Privileges service from ISMP are available in InstallAnywhere. In order to make use of your existing code you will have to create a new class that extends the CustomCodeAction class and overwrite the install() and/or uninstall() methods. You need to use the InstallerProxy reference to access the services. To illustrate the differences, see below for examples of an ISMP custom wizard bean and an InstallAnywhere custom action:

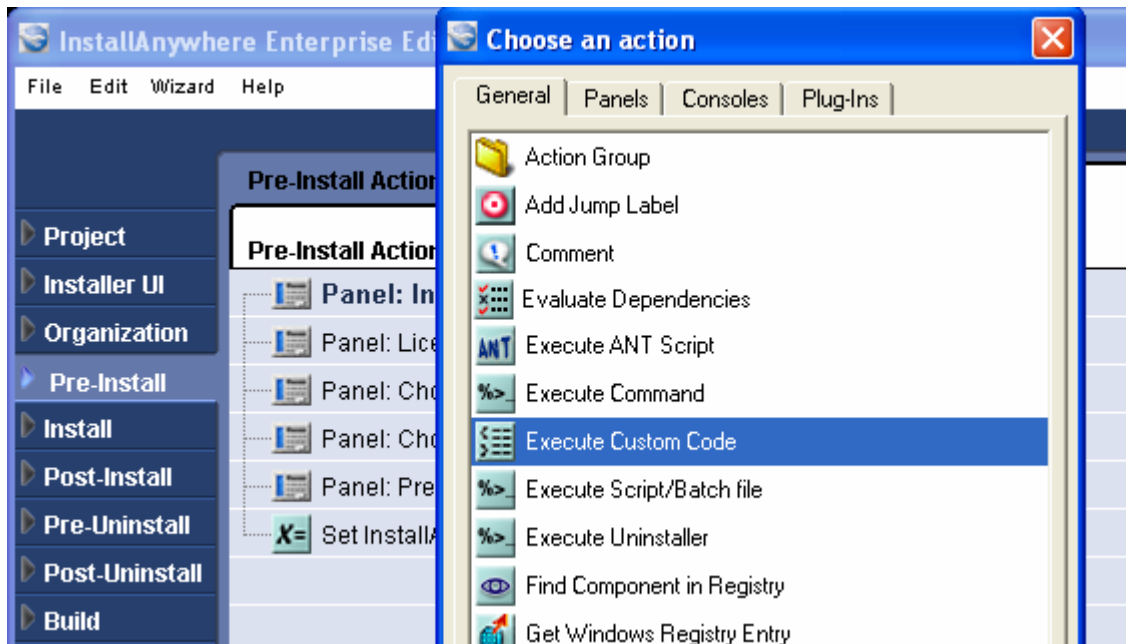
```
//ISMP Custom action
public class MyWizardAction extends WizardAction
{
    public void execute(WizardBeanEvent event){
    {
        FileService fileSvc = (FileService) svcAccess
            .getService(FileService.class);
        if (fileSvc == null) {
            printMessage("FileService not available!!!");
        } else {
            fileSvc.createDirectory("//myfolder");
        }
    }
}

//InstallAnywhere Custom Action
public class MyCustomCodeAction extends CustomCodeAction {
    public void install(InstallerProxy proxy) throws InstallException {
        FileService fileSvc = (FileService) proxy
            .getService(FileService.class);
        if (fileSvc == null) {
            printMessage("FileService not available!!!");
        } else {
            fileSvc.createDirectory("//myFolder");
        }
    }
}
```

ISMP and InstallAnywhere—A comparison between an ISMP custom wizard bean and an InstallAnywhere custom action.

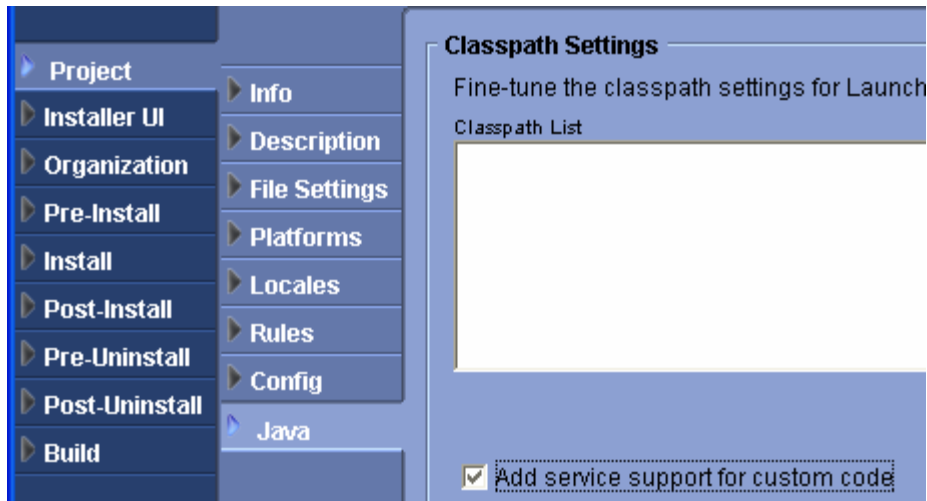


ISMP ---- Custom action is added to the project by registering the custom class in the Add-ins section.



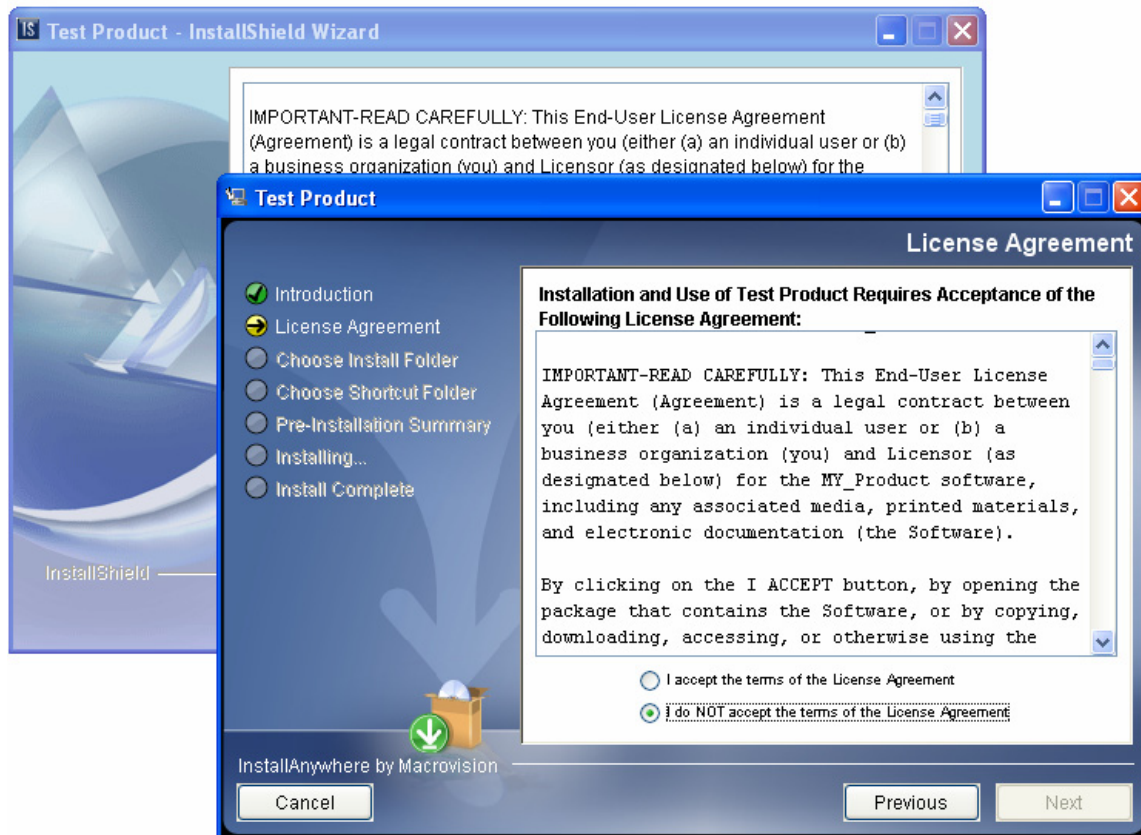
InstallAnywhere—You can add a custom action by choosing the Action from the Actions list and set the path to your class.

In order to compile these classes, you must have the IAClasses.zip file as well as the services.jar class in your classpath. You will also have to check the Add services support for custom code check box locate in the Project/Java tab in the IDE as shown in the figure below. Refer to the InstallAnywhere help file as well as the JavaDocs included with IA.



InstallAnywhere—The Add service support for custom code check box is essential.

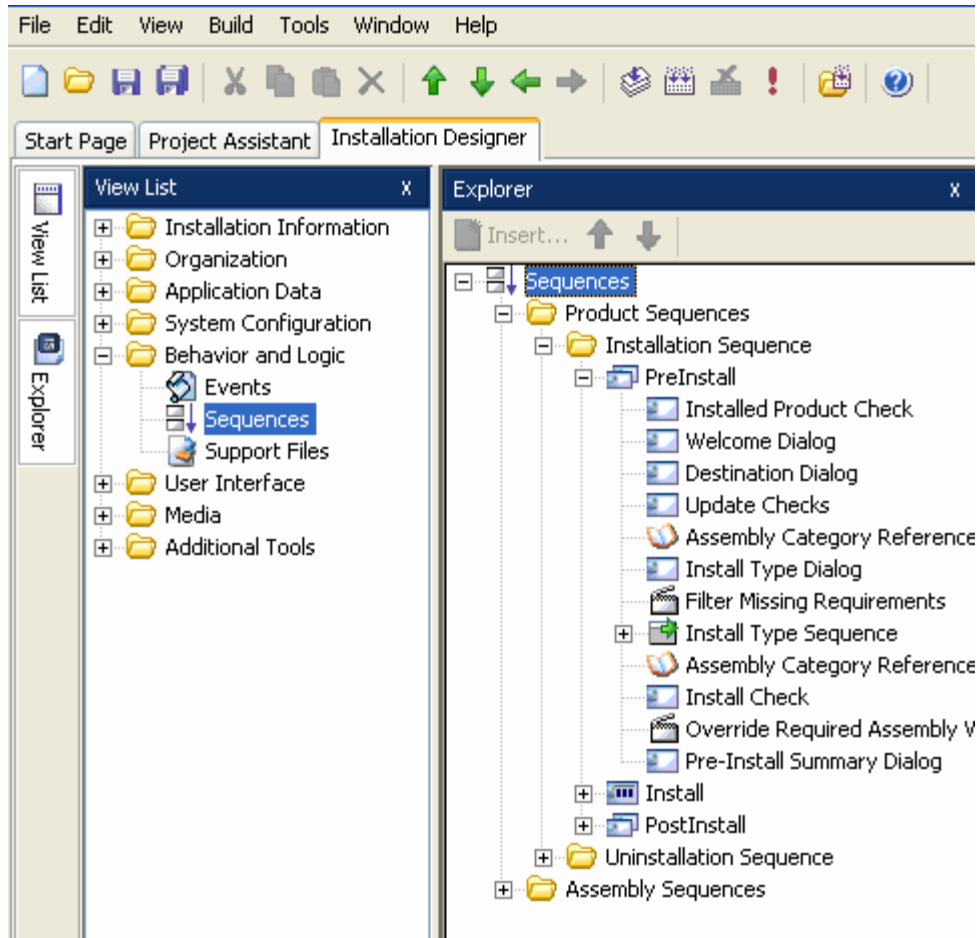
Dialogs



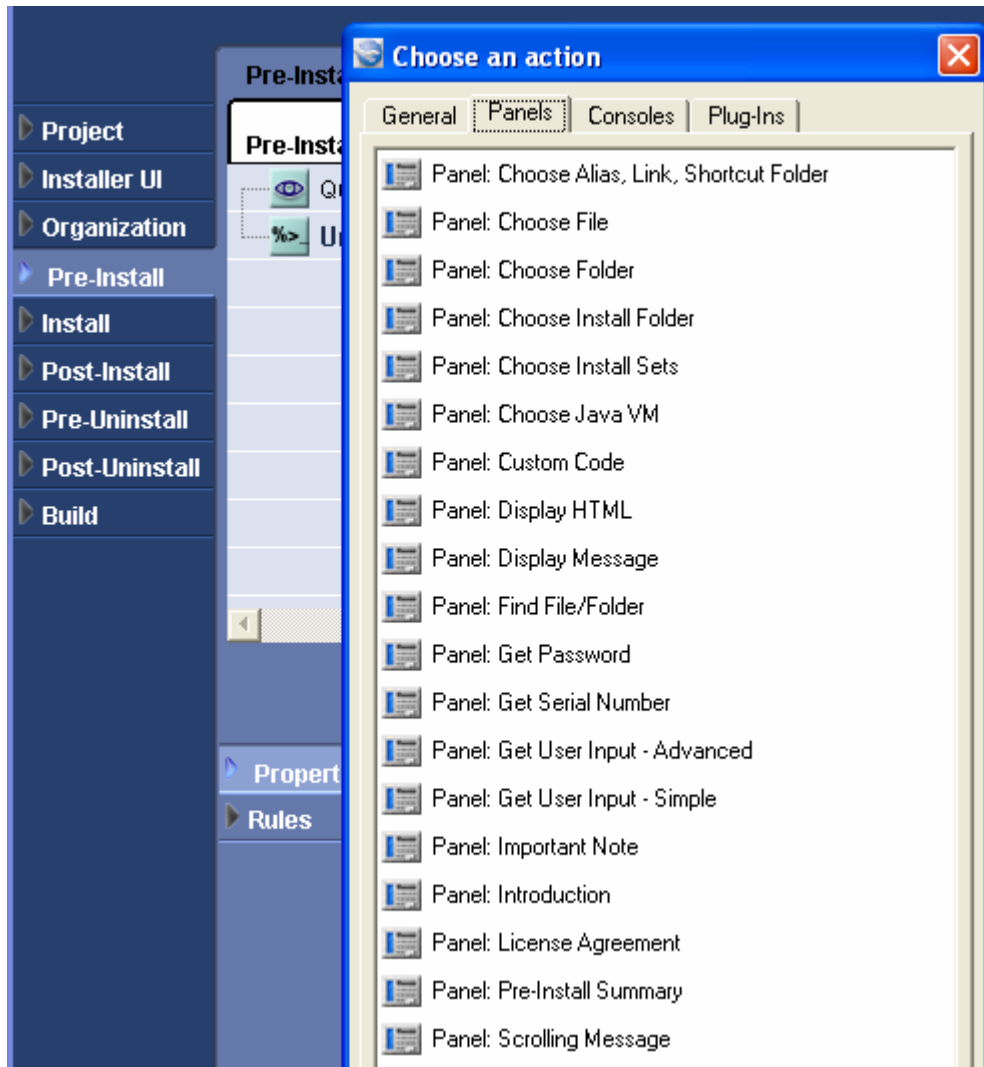
ISMP and InstallAnywhere—A comparison between ISMP and InstallAnywhere dialogs.

Additional Considerations

Standard dialogs will also have to be added to your InstallAnywhere project manually. Most of the standard dialogs found in ISMP are also available in InstallAnywhere such as the License dialog shown above. In addition, IA has a Welcome panel, Password panel, Message panel, InstallSet panel, Get User Input (easy and advanced), among others.



ISMP—Many of the dialogs found in ISMP have an equivalent dialog in IA.



InstallAnywhere—Add dialogs from the list.

Custom Dialogs

Custom dialogs cannot be exported from ISMP and imported to IA. Instead you can leverage the InstallAnywhere UserInputPanel advanced version. This allows you to add various controls to a panel and customize them. You can also utilize the new HTML panel that will allow you to display a stylesheet, javascript, html page or website within the wizard panel. A custom panel can also be created by writing a custom action that extends the CustomCodePanel class. This can be added to the Pre- or Post-Install sequence by clicking the Add Action button and choosing the Panel tab. Choose Panel: Custom Code and set the path property to the location of your custom class.

Maintaining Existing Installations

If you are creating upgrades or patches for existing installations that were originally installed using ISMP, the recommending course is to use the Uninstall InstallShield Universal Software action in IA. To use this action correctly first use the Query InstallShield Software Information to retrieve the location of the previously installed application. You can use the variable

Additional Considerations

\$ISMP_COMPONENT_LOCATIONS\$ in the Uninstall InstallShield Software action to silently uninstall the old application before installing the current version.

The screenshot shows the 'Get InstallShield Universal software information' configuration window. On the left, there is a sidebar with 'Properties' and 'Rules' tabs. The main area contains the following fields and settings:

- InstallShield Universal software:**
 - UUID: v9fsg0grgs0sn86stk30j67m30ls854l25673884yeg
 - ☐ Compare version: Equal to (dropdown) <Please enter InstallShield Universal software UUID>
 - ☒ Highest version only.
 - ☒ Install location: \$USER_INSTALL_DIR\$
 - VPD location: (empty field)
- Set variables:**
 - Components found count: \$ISMP_COMPONENT_COUNT\$
 - Components found versions: \$ISMP_COMPONENT_VERSIONS\$
 - Components found locations: \$ISMP_COMPONENT_LOCATIONS\$

InstallAnywhere—Query ISMP VPD for install location of previous version

The screenshot shows the 'Uninstall InstallShield Universal Software' configuration window. On the left, there is a sidebar with 'Properties' and 'Rules' tabs. The main area contains the following fields and settings:

- InstallShield Universal software:**
 - UUID: <Please enter InstallShield Universal software UUID>
 - ☒ Install location: \$USER_INSTALL_DIR\$
 - ☐ Compare version: Equal to (dropdown) <Please enter InstallShield Universal software UUID>
 - VPD location: (empty field)
- Uninstall options:**
 - ☐ Show indeterminate dialog: Executing...
 - Uninstaller UI mode: ☐ same as parent ☒ always show GUI
- Uninstaller arguments:** -silent
- Store process's stdout in:** \$EXECUTE_STDOUT\$
- Store process's stderr in:** \$EXECUTE_STDERR\$
- Store process's exit code in:** \$EXECUTE_EXITCODE\$

InstallAnywhere—Uninstall InstallShield bases installs.

SUMMARY

With the features of InstallShield MultiPlatform 11.5 SP 1 and InstallAnywhere it is possible to migrate your ISMP projects to InstallAnywhere and replicate much of the custom ISMP functionality without too much re-tooling.



Macrovision Corporation
2830 De La Cruz Boulevard
Santa Clara, CA 95050
USA

Santa Clara (Global Headquarters):
+1 888-755-0861
Chicago: +1 800-809-5659
New York: +1 800-804-0103

United Kingdom (Europe,
Middle East Headquarters):
+44 870-871-1111
+44 870-873-6300

Japan (Asia, Pacific Headquarters):
+81 3-5774-6253

www.macrovision.com