

openInstaller Developer Tools

Usecases

v0.1

6th August 2007

Vadiraj Deshpande

Table of Contents

Introduction.....	3
Use Cases.....	3
Actors.....	3
UC-01: Download and install the plug-in.....	4
UC-02: Download and install the stand alone app.....	4
UC-03: Create new Installer Project.....	5
UC-03a: Create new Platform independent Installer Project.....	5
UC-03b: Create new Platform dependent Installer Project.....	6
UC-04: Open existing Installer Project.....	6
UC-05: Add New Component.....	7
UC-06: Edit existing Component.....	7
UC-07: Add New Data Model.....	8
UC-08: Edit existing Data Model.....	8
UC-09: Add New Panel View.....	9
UC-10: Edit existing Panel View.....	9
UC-11: Add New Panel Sequence.....	10
UC-12: Edit existing Panel Sequence.....	10
UC-13: Add New Panel Sequence Properties.....	11
UC-14: Edit existing Panel Sequence Properties.....	11
UC-15: Add New Splash image.....	12
UC-16: Add New Layout Preference.....	12
UC-17: Edit existing Layout Preference.....	13
UC-18: Add Installer Launcher.....	13
UC-19: Edit existing installer launcher.....	14
UC-20: Add new Component Configuration Profile.....	14
UC-21: Edit existing Component Configuration Profile.....	15
UC-22: Provide Expert Mode.....	15
UC-23: Perform Integrity check.....	16
UC-24: Build and create Installer distribution.....	16
UC-25: Run and test the installer panel flow.....	17
UC-26: Edit existing Configuration wrapper.....	17
UC-27: Create new configuration wrapper.....	18

Introduction

openInstaller Framework ¹ is a generic framework for developing installers mainly targeted for middle-ware products and product stacks such as Java ES. The openInstaller Developer Tools² is a sub project, which aims at simplifying overall installer development experience and is targeted for product teams who want to build installers on top of openInstaller Framework. The various use cases for the openInstaller Developer tools is the subject matter of this document.

openInstaller Developer Tools (or just 'Developer Tools') addresses all possible requirements³ for any kind of installer development using openInstaller Framework. The following section lists out in detail each of the possible use cases based on these requirements.

Use Cases

Actors

INSTALLER_DEVELOPER	A human who develops Installer
NETBEANS_IDE	The Netbeans IDE
INTEGRATION_ENGINEER	A human who assembles the installer

UC-01: Download and install the plug-in

Summary:	Installer Developer connects to the update center and downloads the plug-in into his/her Netbeans IDE and installs it.
Priority:	Essential
Requirement Id:	TOOL_DISTRIBUTION_AND_DEPLOYMENT
Use Frequency:	Once
Direct Actors:	INSTALLER_DEVELOPER
Pre-condition:	A compatible version of Netbeans IDE installed and running on the compatible version of J2SE.
Notes and Questions	<ol style="list-style-type: none">1. This update center will be hosted on java.net.2. This is the standard Netbeans update center.

UC-02: Download and install the stand alone app.

Summary:	Installer Developer connects to the download center and downloads the tool into his/her system and installs it.
Priority:	Essential
Requirement Id:	TOOL_DISTRIBUTION_AND_DEPLOYMENT
Use Frequency:	Once
Direct Actors:	INSTALLER_DEVELOPER
Pre-condition:	All system requirements are met and there is a compatible J2SE already installed.
Notes and Questions	<ol style="list-style-type: none">1. This download center will be hosted on java.net.2. The distribution can be platform specific or zip based.3. There can be JNLP based distribution apart from normal download bundle.

UC-03: Create new Installer Project

Summary:	INSTALLER_DEVELOPER creates a new Installer project, which acts as a container for all installer related files.
Priority:	Essential
Requirement Id:	CREATE_NEW_PROJECT, STORE_INSTALLER_DATA_ONLY_IN_XMLS, INSTALLER_RUNTIME
Use Frequency:	Often
Direct Actors:	INSTALLER_DEVELOPER
Pre-condition:	<ol style="list-style-type: none">1. A writable place on the disk with sufficient disk space to create project.2. The tool should already be installed.
Notes and Questions	<ol style="list-style-type: none">1. This would be the first step in creating new installer.2. After this, the Installer Developer can create required files that make up the installer.3. The tool creates few supporting files by default in the new project. (like installer launcher etc..)4. This use case has two sub cases : 3a and 3b.

UC-03a: Create new Platform independent Installer Project

Summary:	This project generates installer for any platform using platform independent packaging format.
Priority:	Essential
Requirement Id:	CREATE_NEW_PROJECT, STORE_INSTALLER_DATA_ONLY_IN_XMLS, INSTALLER_RUNTIME
Use Frequency:	Often
Direct Actors:	INSTALLER_DEVELOPER
Pre-condition:	<ol style="list-style-type: none">1. A writable place on the disk with sufficient disk space to create project.2. The tool should already be installed.
Notes and Questions:	<ol style="list-style-type: none">1. This project enables creation of installers using platform independent packaging format.

UC-03b: Create new Platform dependent Installer Project

Summary:	This project generates installer for supported platform using platform dependent packaging format.
Priority:	Essential
Requirement Id:	CREATE_NEW_PROJECT, STORE_INSTALLER_DATA_ONLY_IN_XMLS, INSTALLER_RUNTIME
Use Frequency:	Often
Direct Actors:	INSTALLER_DEVELOPER
Pre-condition:	<ol style="list-style-type: none">1. A writable place on the disk with sufficient diskspace to create project.2. The tool should already be installed.
Notes and Questions:	<ol style="list-style-type: none">1. This project enables creation of installers using platform dependent packaging format.

UC-04: Open existing Installer Project

Summary:	The tool opens the existing project using the IDE for editing.
Priority:	Essential
Requirement Id:	EDIT_EXISTING_PROJECT, STORE_INSTALLER_DATA_ONLY_IN_XMLS
Use Frequency:	Always
Direct Actors:	INSTALLER_DEVELOPER
Pre-condition:	<ol style="list-style-type: none">1. The tool should already be installed.2. An installer project is already created and is accessible for the tool.
Notes and Questions:	<ol style="list-style-type: none">1. The tool should be capable of detecting the existing projects and open them on request.

UC-05: Add New Component

Summary:	INSTALLER_DEVELOPER creates new component in the installer project.
Priority:	Essential
Requirement Id:	CREATE_NEW_FILES,INCLUDE_INSTALL_UNITS,TYPE_CHECKING_USER_INPUT,INLINE_VALIDATION_OF_USER_INPUT,FILL_DEFAULTS,IMPORT_FILES_BY_REFERENCE,IMPORT_FILES_BY_COPY,IMPORT_FILES_BY_PARTS
Use Frequency:	Always
Direct Actors:	INSTALLER_DEVELOPER
Pre-condition:	<ol style="list-style-type: none">1. The tool should already be installed.2. An installer project is opened by the tool.
Notes and Questions:	<ol style="list-style-type: none">1. The tool should be capable of asking the right questions about the new component that Installer developer wishes to create and validate all inputs before creating the component.2. Files created : Product Descriptor file

UC-06: Edit existing Component

Summary:	INSTALLER_DEVELOPER opens the existing component in IDE for editing.
Priority:	Essential
Requirement Id:	EDIT_REQUIRED_FILES,INCLUDE_INSTALL_UNITS,VIEW_DEPENDENCIES,TYPE_CHECKING_USER_INPUT,INLINE_VALIDATION_OF_USER_INPUT,VIEW_EDIT_GENERATED_XML,ADVANCED_EDITING_XML,IMPORT_FILES_BY_PARTS
Use Frequency:	Always
Direct Actors:	INSTALLER_DEVELOPER
Pre-condition:	<ol style="list-style-type: none">1. The tool should already be installed.2. An installer project is opened by the tool.3. There is a/are component(s) already present in the project.
Notes and Questions:	<ol style="list-style-type: none">1. The tool should be capable of validating all inputs before committing changes to the disk.

UC-07: Add New Data Model

Summary:	INSTALLER_DEVELOPER creates new data model in the installer project.
Priority:	Essential
Requirement Id:	CREATE_NEW_FILES, TYPE_CHECKING_USER_INPUT,INLINE_VALIDATION_OF_USER_INPUT, FILL_DEFAULTS,IMPORT_FILES_BY_REFERENCE, IMPORT_FILES_BY_COPY,IMPORT_FILES_BY_PARTS
Use Frequency:	Always
Direct Actors:	INSTALLER_DEVELOPER
Pre-condition:	<ol style="list-style-type: none">1. The tool should already be installed.2. An installer project is opened by the tool.
Notes and Questions:	<ol style="list-style-type: none">1. The tool should be capable of asking the right questions about the new data model that Installer developer wishes to create and validate all inputs before creating the data model.2. Files created : OOR file

UC-08: Edit existing Data Model

Summary:	INSTALLER_DEVELOPER opens the existing data model in IDE for editing.
Priority:	Essential
Requirement Id:	EDIT_REQUIRED_FILES,TYPE_CHECKING_USER_INPUT,INLINE_VALIDATION_OF_USER_INPUT, VIEW_EDIT_GENERATED_XML,ADVANCED_EDITING_XML, IMPORT_FILES_BY_PARTS
Use Frequency:	Always
Direct Actors:	INSTALLER_DEVELOPER
Pre-condition:	<ol style="list-style-type: none">1. The tool should already be installed.2. An installer project is already created and is accessible for the tool.3. A data model must be created in the current project.
Notes and Questions:	<ol style="list-style-type: none">2. The tool should be capable of validating all inputs before committing changes to the disk.

UC-09: Add New Panel View

Summary:	INSTALLER_DEVELOPER creates new view in the installer project..
Priority:	Essential
Requirement Id:	CREATE_NEW_FILES, TYPE_CHECKING_USER_INPUT,INLINE_VALIDATION_OF_USER_INPUT, FILL_DEFAULTS, IMPORT_FILES_BY_REFERENCE, IMPORT_FILES_BY_COPY, IMPORT_FILES_BY_PARTS
Use Frequency:	Always
Direct Actors:	INSTALLER_DEVELOPER
Pre-condition:	<ol style="list-style-type: none">1. The tool should already be installed.2. An installer project is opened by the tool.
Notes and Questions:	<ol style="list-style-type: none">1. The tool should be capable of asking the right questions about the new panel view that Installer developer wishes to create and validate all inputs before creating the panel view.2. Files created : APOC file

UC-10: Edit existing Panel View

Summary:	INSTALLER_DEVELOPER opens the existing data model in IDE for editing.
Priority:	Essential
Requirement Id:	EDIT_REQUIRED_FILES,TYPE_CHECKING_USER_INPUT,INLINE_VALIDATION_OF_USER_INPUT, VIEW_EDIT_GENERATED_XML, ADVANCED_EDITING_XML, IMPORT_FILES_BY_PARTS
Use Frequency:	Always
Direct Actors:	INSTALLER_DEVELOPER
Pre-condition:	<ol style="list-style-type: none">1. The tool should already be installed.2. An installer project is already created and is accessible for the tool.3. A panel view must be created in the current project.
Notes and Questions:	<ol style="list-style-type: none">1. The tool should be capable of validating all inputs before committing changes to the disk.

UC-11: Add New Panel Sequence

Summary:	INSTALLER_DEVELOPER creates new panel sequence in the installer project..
Priority:	Essential
Requirement Id:	CREATE_NEW_FILES, TYPE_CHECKING_USER_INPUT,INLINE_VALIDATION_OF_USER_INPUT, FILL_DEFAULTS, IMPORT_FILES_BY_REFERENCE, IMPORT_FILES_BY_COPY, IMPORT_FILES_BY_PARTS
Use Frequency:	Always
Direct Actors:	INSTALLER_DEVELOPER
Pre-condition:	<ol style="list-style-type: none">1. The tool should already be installed.2. An installer project is opened by the tool.
Notes and Questions:	<ol style="list-style-type: none">1. The tool should be capable of asking the right questions about the new panel sequence that Installer developer wishes to create and validate all inputs before creating the panel sequence.2. The tool should allow to create a single panel sequence file for a project.3. Files created : Page sequence XML file

UC-12: Edit existing Panel Sequence

Summary:	INSTALLER_DEVELOPER opens the existing panel sequence in IDE for editing.
Priority:	Essential
Requirement Id:	EDIT_REQUIRED_FILES,TYPE_CHECKING_USER_INPUT,INLINE_VALIDATION_OF_USER_INPUT, VIEW_EDIT_GENERATED_XML, ADVANCED_EDITING_XML, IMPORT_FILES_BY_PARTS
Use Frequency:	Always
Direct Actors:	INSTALLER_DEVELOPER
Pre-condition:	<ol style="list-style-type: none">1. The tool should already be installed.2. An installer project is already created and is accessible for the tool.3. A panel sequence must be created in the current project.
Notes and Questions:	<ol style="list-style-type: none">1. The tool should be capable of validating all inputs before committing changes to the disk.

UC-13: Add New Panel Sequence Properties

Summary:	INSTALLER_DEVELOPER creates new panel sequence properties in the installer project..
Priority:	Essential
Requirement Id:	CREATE_NEW_FILES, TYPE_CHECKING_USER_INPUT,INLINE_VALIDATION_OF_USER_INPUT, FILL_DEFAULTS, IMPORT_FILES_BY_REFERENCE, IMPORT_FILES_BY_COPY,IMPORT_FILES_BY_PARTS
Use Frequency:	Always
Direct Actors:	INSTALLER_DEVELOPER
Pre-condition:	<ol style="list-style-type: none">1. The tool should already be installed.2. An installer project is opened by the tool.
Notes and Questions:	<ol style="list-style-type: none">1. The tool should be capable of asking the right questions about the new panel sequence properties that Installer developer wishes to create and validate all inputs before creating the panel sequence properties.2. The tool should allow to create single panel sequence properties.3. Files created : Page sequence properties file.

UC-14: Edit existing Panel Sequence Properties

Summary:	INSTALLER_DEVELOPER opens the existing panel sequence properties in IDE for editing.
Priority:	Essential
Requirement Id:	EDIT_REQUIRED_FILES,TYPE_CHECKING_USER_INPUT,INLINE_VALIDATION_OF_USER_INPUT, IMPORT_FILES_BY_PARTS
Use Frequency:	Always
Direct Actors:	INSTALLER_DEVELOPER
Pre-condition:	<ol style="list-style-type: none">1. The tool should already be installed.2. An installer project is already created and is accessible for the tool.3. A panel sequence properties must be created in the current project.
Notes and Questions:	<ol style="list-style-type: none">1. The tool should be capable of validating all inputs before committing changes to the disk.

UC-15: Add New Splash image

Summary:	INSTALLER_DEVELOPER creates new splash image in the installer project..
Priority:	Essential
Requirement Id:	CREATE_NEW_FILES, TYPE_CHECKING_USER_INPUT,INLINE_VALIDATION_OF_USER_INPUT
Use Frequency:	Always
Direct Actors:	INSTALLER_DEVELOPER
Pre-condition:	<ol style="list-style-type: none">1. The tool should already be installed.2. An installer project is opened by the tool.
Notes and Questions:	<ol style="list-style-type: none">1. The tool should be capable of creating a new image.

UC-16: Add New Layout Preference

Summary:	INSTALLER_DEVELOPER creates new layout preference in the installer project..
Priority:	Desired
Requirement Id:	CREATE_NEW_FILES, TYPE_CHECKING_USER_INPUT,INLINE_VALIDATION_OF_USER_INPUT, FILL_DEFAULTS, IMPORT_FILES_BY_REFERENCE, IMPORT_FILES_BY_COPY, IMPORT_FILES_BY_PARTS
Use Frequency:	Rarely
Direct Actors:	INSTALLER_DEVELOPER
Pre-condition:	<ol style="list-style-type: none">1. The tool should already be installed.2. An installer project is opened by the tool.
Notes and Questions:	<ol style="list-style-type: none">1. The tool should be capable of asking the right questions about the new layout preference that Installer developer wishes to create and validate all inputs before creating the layout preference.2. Files created : ui.pref, gui.pref, cui.pref

UC-17: Edit existing Layout Preference

Summary:	INSTALLER_DEVELOPER opens the existing layout preference in IDE for editing.
Priority:	Essential
Requirement Id:	EDIT_REQUIRED_FILES,TYPE_CHECKING_USER_INPUT,INLINE_VALIDATION_OF_USER_INPUT, IMPORT_FILES_BY_PARTS
Use Frequency:	Always
Direct Actors:	INSTALLER_DEVELOPER
Pre-condition:	<ol style="list-style-type: none">1. The tool should already be installed.2. An installer project is already created and is accessible for the tool.3. A layout preference must be created in the current project.
Notes and Questions:	<ol style="list-style-type: none">1. The tool should be capable of validating all inputs before committing changes to the disk.

UC-18: Add Installer Launcher

Summary:	The tool adds a new installer launcher script to the current project.
Priority:	Desired
Requirement Id:	CREATE_NEW_FILES,TYPE_CHECKING_USER_INPUT,INLINE_VALIDATION_OF_USER_INPUT, FILL_DEFAULTS
Use Frequency:	Sometimes
Direct Actors:	INSTALLER_DEVELOPER
Pre-condition:	<ol style="list-style-type: none">1. The tool should already be installed.2. An installer project is already created and is accessible for the tool.
Notes and Questions:	<ol style="list-style-type: none">1. The tool should be capable of validating all inputs before committing changes to the disk.2. The launcher script is created when a new project is created. This option enables the installer developer to re-create it if its removed from the project.3. Files created : installer.sh, installer.vbs

UC-19: Edit existing installer launcher

Summary:	INSTALLER_DEVELOPER opens the existing installer script in IDE for editing.
Priority:	Desired
Requirement Id:	EDIT_REQUIRED_FILES,TYPE_CHECKING_USER_INPUT,INLINE_VALIDATION_OF_USER_INPUT
Use Frequency:	Sometimes
Direct Actors:	INSTALLER_DEVELOPER
Pre-condition:	<ol style="list-style-type: none">1. The tool should already be installed.2. An installer project is already created and is accessible for the tool.3. An installer launcher script is already present in the current project.
Notes and Questions:	<ol style="list-style-type: none">1. The tool should be capable of validating all inputs before committing changes to the disk.

UC-20: Add new Component Configuration Profile

Summary:	INSTALLER_DEVELOPER creates a new configuration profile.
Priority:	Desired
Requirement Id:	CREATE_NEW_FILES, TYPE_CHECKING_USER_INPUT, INLINE_VALIDATION_OF_USER_INPUT, FILL_DEFAULTS
Use Frequency:	Often
Direct Actors:	INSTALLER_DEVELOPER
Pre-condition:	<ol style="list-style-type: none">1. The tool should already be installed.2. An installer project is already created and is accessible for the tool.
Notes and Questions:	<ol style="list-style-type: none">1. The tool should allow creating this feature as this is required in most of the installer cases like Application Server. ()2. We observed that there is a need to have more than one configuration set for a product. The best example is Application Server, it has Personal Edition and Enterprise Edition. These two differ in the way the product is configured, but use the same install units. In the current use case, this new config. profile will group related data models and panel views, thus creating different configuration sets for a single product. This feature will be a add on to the underlying installer framework.

UC-21: Edit existing Component Configuration Profile

Summary:	INSTALLER_DEVELOPER edits an existing configuration profile.
Priority:	Desired
Requirement Id:	CREATE_NEW_FILES, TYPE_CHECKING_USER_INPUT, INLINE_VALIDATION_OF_USER_INPUT
Use Frequency:	Often
Direct Actors:	INSTALLER_DEVELOPER
Pre-condition:	<ol style="list-style-type: none">1. The tool should already be installed.2. An installer project is already created and is accessible for the tool.
Notes and Questions:	<ol style="list-style-type: none">1. The tool should allow editing as this is required in most of the installer cases like Application Server. (Personal Edition and Enterprise Edition).2. Only renaming/deleting of the existing config. Profile is allowed.

UC-22: Provide Expert Mode

Summary:	The tool operates in expert mode.
Priority:	Desired
Requirement Id:	CUSTOMIZE_INSTALLER_PROJECT, SCM_ACCESS, EDIT_PARTIAL_XML
Use Frequency:	Often
Direct Actors:	INSTALLER_DEVELOPER
Pre-condition:	<ol style="list-style-type: none">1. The tool should already be installed.2. An installer project is already created and is accessible for the tool.
Notes and Questions:	<ol style="list-style-type: none">1. The tool should allow operating in two modes, one being 'Expert' mode, which turns on advanced features such as customizing installer layout and source control access.2. The tool should allow switching between the modes.(Expert and normal)

UC-23: Perform Integrity check

Summary:	The tool validates all aspects of installer as a whole (and its files) to be consistent.
Priority:	Essential
Requirement Id:	HIDE_COMPLEXITY_OF_INSTALLER_FRAMEWORK, BEST_PRACTICE_CHECK, BRANDING_INSTALLER, BRANDING_INSTALLER_PLUGIN, INSTALLER_PROJECT_INTEGRITY_CHECK
Use Frequency:	Often
Direct Actors:	INSTALLER_DEVELOPER
Pre-condition:	<ol style="list-style-type: none">1. The tool should already be installed.2. An installer project is already created and is accessible for the tool.
Notes and Questions:	<ol style="list-style-type: none">1. The tool should have logic in built to validate and verify all installer related aspects correctly and should help installer developers to resolve any errors quickly in a user friendly manner.2. Often its required that error hints are provided as possible.

UC-24: Build and create Installer distribution

Summary:	The tool should be able to build (assemble) the installer and create distributions.
Priority:	Essential
Requirement Id:	BUILD_INSTALLER_AND_TEST, INSTALLER_RUNTIME
Use Frequency:	Always
Direct Actors:	INSTALLER_DEVELOPER, INTEGRATION_ENGINEER
Pre-condition:	<ol style="list-style-type: none">1. The tool should already be installed.2. An installer project is already created and is accessible for the tool.3. The Integrity check should report no errors.
Notes and Questions:	

UC-25: Run and test the installer panel flow

Summary:	The tool should allow running and testing the installer from within itself.
Priority:	Expected
Requirement Id:	BUILD_INSTALLER_AND_TEST, VIEW_PANEL MOCKUP
Use Frequency:	Often
Direct Actors:	INSTALLER_DEVELOPER
Pre-condition:	<ol style="list-style-type: none">1. The tool should already be installed.2. An installer project is already created and is accessible for the tool.3. The integrity check should report no errors.
Notes and Questions:	<ol style="list-style-type: none">1. The tool should allow running the installer and test panel sequence.2. Only panel sequence and some configuration can be tested.

UC-26: Edit existing Configuration wrapper

Summary:	The tool should allow editing of existing configuration wrappers.
Priority:	Desired
Requirement Id:	EDIT_EXISTING_FILES, TYPE_CHECKING_USER_INPUT, INLINE_VALIDATION_OF_USER_INPUT
Use Frequency:	Often
Direct Actors:	INSTALLER_DEVELOPER
Pre-condition:	<ol style="list-style-type: none">3. The tool should already be installed.4. An installer project is already created and is accessible for the tool.
Notes and Questions:	<ol style="list-style-type: none">1. The tool should allow editing of configuration wrappers and associated files.2. There can be a feature which allows dry-testing of the same.

UC-27: Create new configuration wrapper

Summary:	The tool should help creating/including configuration wrappers for component configuration process.
Priority:	Desired
Requirement Id:	CREATE_NEW_FILES, TYPE_CHECKING_USER_INPUT, INLINE_VALIDATION_OF_USER_INPUT, FILL_DEFAULTS
Use Frequency:	Often
Direct Actors:	INSTALLER_DEVELOPER
Pre-condition:	<ol style="list-style-type: none">5. The tool should already be installed.6. An installer project is already created and is accessible for the tool.
Notes and Questions:	<ol style="list-style-type: none">1. The tool should allow creation of configuration wrapper and actual configuration files for components.2. There can be a feature which allows dry-testing of the same.3. Files created : init-config.sh

UC-27: Create easy installer for Netbeans module suite projects

Summary:	The tool should help generating installers for Netbeans module suite projects.
Priority:	Desired
Requirement Id:	EASY_INSTALLER_GENERATOR_FOR_NETBEANS
Use Frequency:	Often
Direct Actors:	INSTALLER_DEVELOPER/NETBEANS_DEVELOPER
Pre-condition:	<ol style="list-style-type: none">7. The tool should already be installed.8. Netbeans module suite is already created and is branded.
Notes and Questions:	<ol style="list-style-type: none">1. The tool should ask minimal questions about the project (like name, version and license) and generate the installer.2. This feature will be a highlight of the openInstaler developer tools.

- 1 www.OpenInstaller.org
- 2 (todo-external link to dev. Tools here)
- 3 (todo-external link to dev. Tools requirements page)