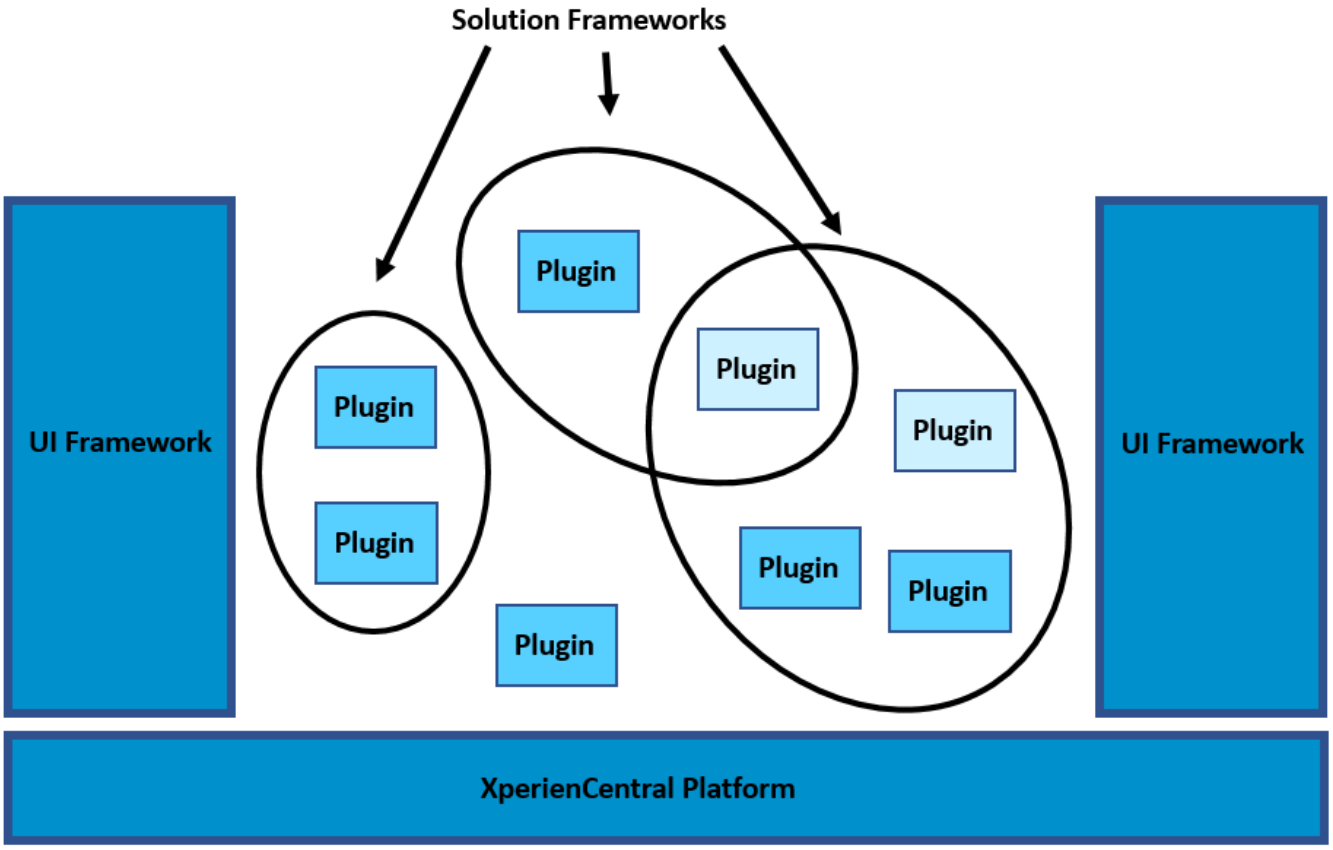
Plugins

In This Section

In XperienCentral, the platform and the components are separated. This allows for a vigorous platform that gives developers the opportunity to easily build solutions or new products on top of the XperienCentral platform. XperienCentral separates the platform and components for the following reasons:

- There is a much clearer distinction between customization and the standard product. By introducing the use of explicit APIs, with which functionalities can be added to the basic platform, the procedure for developers is much clearer. Among other things, this leads to a much more flexible and predictable method of executing updates and migrations than before.
- Partners who want to create their own product with XperienCentral as an OEM product underneath can also do this with the platform strategy.
- With this platform not only partners, but also GX Software itself can easily realize new solutions or products.



Nomenclature

The origin of plugins in XperienCentral lies in the identical functionality available in GX WebManager 9 which was known as a WebManager Component Bundle, referred to using the abbreviation "WCB". Because of this legacy, you will come across the string "wcb" in functions, methods, properties and parameters in the source code and Java classes exposed by XperienCentral for developing plugins. The phased-out term "WCB" is identical to the term "plugin" used in all XperienCentral documentation.

XperienCentral Components

In XperienCentral, a component is a small piece of software that provides a particular service. This may be a GUI component, but can also be a headless service. Each component is of a particular component type which identifies the properties and logic the component provides. In XperienCentral the possible component types are:

Component Type	Description
Element	Represents a content element in XperienCentral. An element can be inserted in a page, page section, and a media item.
Panel	Represents a panel popup in XperienCentral. A panel is accessible from the Configuration menu.
MediaItem	Represents a particular media type in the Content Repository.
Page metadata	This component allows the plugin programmer to add additional metadata fields to each page in XperienCentral.
Form	With this component type, new Handlers, Validators and Routers can be added to forms.
Presentation	Contains a collection of design entities (JSPs, images, stylesheets) defining the appearance of the website environment.
Service	This is a so-called headless service; it has no user interface. Examples of services are mailing services, newsfeed imports, storage services and scheduling services.
Servlet	A servlet is an object that receives a request and generates a response based on that request.
Widget	A widget is a plugin that implements the Dojo Toolkit framework. See Widgets for a comprehensive explanation of how to develop plugins containing widgets for XperienCentral.

Multiple components can be combined into one plugin. A plugin is a set of components that logically belong to the same software component. For example, an authorization plugin may contain an authorization service as well as a panel in which users and authorization can be maintained.

Multiple plugins can be combined into a WCA, a WebManager Component Archive. This is particularly useful for system administrators because it is a set of plugins that can be uploaded into an XperienCentral installation in a single update.

Plugin Certification

Plugins can be certified when they conform to the plugin guidelines. Conforming to these guidelines will improve the overall quality of the plugin in all its aspects - well-designed, well-documented, consistent, compatible, Internationalization ready and migration ready. For that reason it is important to know and understand these guidelines before you start developing them. Follow this link to see the Plugin Development Guidelines.

Getting Started

The best way to start developing a plugin is by using an archetype. The example command below creates an element archetype:

```
mvn archetype:generate -DinteractiveMode=false -DarchetypeGroupId=nl.gx.webmanager.archetypes -
DarchetypeArtifactId=XperienCentral-element-archetype
-DarchetypeVersion=<XperienCentral Version> -DgroupId=com.gxXperienCentral.helloworld -
DartifactId=helloworldelement
-Dclassprefix=HelloWorld -s ..\XperienCentral\settings.xml
```

The following is a sample `activator.java` file with key:

```
...
public class Activator extends ComponentBundleActivatorBase {
  /**
   * Creates and returns the bundle definition of the plugin.
   * @return the bundle definition of the plugin.
   */
  @Override
  protected
ComponentBundleDefinition getBundleDefinition() {
    ComponentBundleDefinitionImpl componentBundleDefinition =
      new ComponentBundleDefinitionImpl();
componentBundleDefinition.setId(WCBConstants.BUNDLE_ID);
componentBundleDefinition.setName(WCBConstants.BUNDLE_NAME);
componentBundleDefinition.setNameSpace(WCBConstants.NAMESPACE_URI);
componentBundleDefinition.setDescription(WCBConstants.BUNDLE_DESCRIPTION);
componentBundleDefinition.setComponentDefinitions(getComponentDefinitions());
    return componentBundleDefinition;
  }
...
```