



## Good To Know Part 2

In this part of the Good To Know lesson, we will continue to learn more about the Developer Console, and guide you through the process of working with it.

After starting the Developer Console, we can choose whether we want to create a new dcp project, or open an already existing one.

For new Drivers generated from mySDK website, the dcp project is automatically generated along with the Driver and predefined values.  
For other Plugins, dcp projects still need to be created manually.

To create a new project, you need to select the folder where the dcp file will be placed. If this folder contains an sln file, the dcp file will automatically get the same name as the sln file.

All folders are defined relative to the Project folder.

Plugin Folder defines the path to the compiled set of Plugins that should be processed together in one installation package for the C4 System. The predefined value is @bin.

For the localization of Plugins, there are two options.

The first one is to use the standard Core localization package, which contains localization keys managed by Gamanet. It is available in the Localization section of Portal Services. After downloading the required supported language, you can start using it right away.

If you want to use new, specific localization keys that are not included in the Core localization package, you need to manage these in your own Resx file. For this, you need to enable using of resources and define the path to your localization data.

When creating an installation package, Developer Console adds the localization data to the package, and the C4 System subsequently ensures its correct installation within the System.



More information about the tool for management of localization will be provided later in this video.

## **What is PackageInfo project**

As we already mentioned in the first part of this lesson, one Solution can contain multiple Plugins which are installed in a single installation package. A mandatory part of each such package is a PackageInfo project which creates a PackageInfo DLL. Now, let's learn more about it.

PackageInfo.dll is a file that uniquely identifies the installation package of a set of Plugins. It serves for redistribution of Plugins within the overall topology of the C4 System, and also ensures automatic upgrades of all Plugins within a single installation set.

Also, the version of PluginInfo serves as a key data for upgrading Plugins within the C4 System after the installation of the package. Therefore, the version must be regularly increased with each build of the Solution.

In the AssemblyInfo project, there is a unique identifier - the ID of the package and its name. It's very important to remember that this ID must be unique and remain the same throughout the package's lifetime.

For debugging of Plugin, it's necessary to set the correct variant of Developer Console as the startup external program within the Plugin Project settings.

Then, in the Command line arguments, we enter the full path to the Developer Console project.

## **How to use Developer Console**

The Plugin section of the Developer Console provides separate hosting modules for every category of Plugins. Each of these modules emulates the environment of the specific part of the C4 System in which the Plugin will run.

In our video, we will be working in the module for the Client Panel.

Each hosting module contains a toolbar at the top, with several sections of controls.



On the left side, there is a section for activating and controlling the Plugin. These buttons are specific according to the category of the Plugin we are developing.

In case the Plugin requires contextual data for its operation, this data will be displayed on the left side of the hosting module. Contextual data can be manually refreshed by using the Refresh button.

Then there is a section for selecting the Plugin and registering it to the C4 Server.

If the Plugin supports localization, you will find the localization section on the right side.

Find ID of the C4 Panel is a support tool which helps the developers of the Client Panels to obtain the information they need to correctly define the Panel in the source codes.

The last button on the toolbar is the button for analysis of logs – Trace Analyzer.

The section for management of Plugins contains a dropdown menu with the list of Plugins from the given project.

The Plugins will appear in the list after clicking the Assembly Scanner button.

It is a tool that searches through the directories for information about all the Plugins in the given folder. In this step, the Plugins are just scanned. Loading and activation of the Plugin will happen only after clicking the Activate button.

Searching for Plugins within the individual assemblies is based on the IPlugin attribute or its derived interfaces (IDriver, IISConnector, IPanel, etc).

For each Plugin, the following information is provided: name of the Plugin, category of the Plugin, supported platform and supported runtime.

Plugins that are not compatible with the running variant of the Developer Console are disabled and cannot be used.

In case of any changes in the Plugin interface specification in the project, we recommend to scan the Plugin Folder to correctly configure the parameters of the Developer Console. The list of available Plugins is then stored in the dcp project for quick loading of application when debugging a Plugin.



The last option in this section is registration of Plugins into the C4 Server. This operation consists of two steps.

In the first step, all Plugins, including support assemblies, are packed into an installation package .c4plugin, which is placed in the directory Plugin Folder\@export. In the second step, the installation package is automatically uploaded to the C4 System, where it is unpacked and individual Plugins are distributed to their designated locations of use and then installed.

This process is fully automatic. In case of an error during the package installation, it is necessary in the first step to verify whether it was successfully created – in @export there is a file with suffix .c4plugin. If not, the problem may be in compatibility of the Plugin with the Package Registrator tool running within the Developer Console.

If uploading of package to the C4 System fails, the problem may be in compatibility of the C4 System with the Plugin in the package. For these reasons, Gamanet recommends developing Plugins always on the latest available version of the C4 System, Developer Console, as well as the available nuggets used in the Plugins.

After this registration, it is possible to use the Plugin within the C4 System.

### **Now, let's take a closer look at how to use resources during Plugin development.**

For management of resources in the Developer Console, there is a specialized tool called Resources Tool.

It consists of two separate panels.

The first panel is dedicated to management of the localization keys for various languages.

The second panel is used for browsing the library of C4 icons and images.

As we have mentioned before, in localization, we can work in two ways.

We can either use Core localization keys provided and managed by Gamanet, or use our own localization keys.



The Core keys can be downloaded from the localization web.

After logging in to your account, you get the option to select the languages you want to download.

English Developer/Keys is a special language that serves as a library of all localization keys in English, as they are suggested by developers for translators.

The Core keys are managed by Gamanet, and it also ensures their translation into all supported C4 languages. Developers cannot edit them.

The selected languages are downloaded to the folder Developer Console, `_CoreResources`.

After switching to your own module, you can create your custom localization keys for your Plugin and edit them.

These keys as well as their translations are entirely under your management.

The only mandatory requirement is that they must be available in English language.

ImagesPanel is a library of C4 icons and images provided by Gamanet for developers.

This library of images and their sizes is constantly growing, and its latest version can be downloaded from Gamanet's NuGet server. After downloading it, it is necessary to restart the application for the new images to apply.

If you need to use custom keys within your Panel, you can create your own image resources and merge resources directly to your Client Panel. The C4 Controls is designed to use image keys from all merged resources in the C4 Client application.