



Good To Know Part 1

Hi Everybody, Welcome to the human soul of your security engine!

This lesson will provide you with information that is good to know when developing Plugins for the C4 System.

In the first part of this lesson, let us make an introduction to a tool provided by Gamanet for developers of C4 Plugins, called Developer Console.

The C4 System enables running of Plugins of different categories.

Each of them can be developed in different platform, runtime and .NET version.

This choice depends on location of the Plugin within the C4 System, as well as whether the Plugin operates within the C4 System environment as its extension, or connects an external system.

Plugins designed for extension of the C4 Client, such as Client Panels, must be developed in the same platform as the C4 Client itself, which is .NET Framework 64-bit, version 4.8.

Plugins of type Report, which are running in the C4 Server core module, operate in .NET Core 64-bit, version 6.0.

Smart Routines running in the C4 Server environment, in the module for management of incidents and workflow, are also developed in .NET Core 64-bit, version 6.0.

In general, if possible, we recommend developing all Plugins in the runtime Any.

In case of Drivers and IS Connectors, which are designed for communication with external devices, the choice of the platform and runtime depends on whether the manufacturer provides an SDK.

If there is an SDK available, then it is necessary to follow the platform and runtime in which the SDK is programmed.

If there is no SDK provided, Gamanet recommends to develop in platform .NET Standard and runtime Any.



As we can see, there are four different Plugin Runners available, to be able to run Drivers under their specific platforms and runtimes.

For debugging of Plugins during their development, Gamanet provides a unified environment – an application called Developer Console. It is released in four variants covering all four possible combinations of platforms and runtimes.

Developer Console is the only place where it is possible to run and test the Plugins which are not yet licensed.

Developer Console is distributed independently of the C4 System, we can download it from the Portal Services, section Downloads.

By default, Developer Console is installed to the folder *Program Files\Gamanet\Developer Console*.

As we have already mentioned, it is installed in four variants.

The Core directory contains DeveloperConsoleCore_x64.exe and DeveloperConsoleCore_x86.exe.

The Framework directory contains DeveloperConsoleFramework_x64.exe and DeveloperConsoleFramework_x86.exe.

All variants of Developer Console use the same gamanet.config file. It enables unified login to a single C4 Server for all variants.

How to activate Developer Console

When launching Developer Console for the first time, we need to set its connection to the C4 System.

In the Application Settings window, enter the credentials which you use to log in to the C4 System.

As we mentioned in previous lessons, the logged-in user must have permissions to install Drivers into the C4 System.

Click the Check button to verify if the entered data is correct.



When the Developer Console is launched and a newer version is available, it automatically upgrades to the latest official release.

You can enable downloading of Beta versions as well, but we recommend to use only the officially released versions.

The Log button enables quick access to a special directory in the user's Temp folder, which stores logs generated by the Developer Console itself, as well as by the developed Plugins.

The entered login details will be implemented into Developer Console by clicking the Apply button.

After establishing connection with the C4 System, you can start working with Developer Console.

A Visual Studio Solution can contain one or more Plugin projects in different platforms and runtimes.

When creating a new project, the developer selects the platform and runtime they will work in. It's important to note that for compatibility with the C4 System, Plugins need to be developed in .NET versions older than or equal to those supported by the C4 System. Currently, the highest supported versions are Core 6.0 and Framework 4.8.

One project can contain multiple Plugins, but we recommend that they are all of the same category. It is important for the distribution of Plugins into different parts of the C4 System as well as for their upgrades.

How to prepare Plugin Solution

Let's now have a closer look at the recommended structure of the Solution folder.

In Gamanet convention for Plugin development, source codes and binary codes are kept in separate subdirectories.

The src subdirectory contains source codes for individual projects.

As we already mentioned, the Solution folder can contain one or more projects of Plugins.



The @bin subdirectory contains compiled DLL files.

These are further divided into subdirectories according to the specific projects.

It is important to build all Plugins into the common root @bin directory, to be able to create a single installation package from the entire Solution.

A mandatory component in development of each installation package is PackageInfo DLL, which must be placed directly in the root @bin directory. It is a specific project whose purpose will be explained in detail later in this video.

For each Visual Studio Solution, it is possible to set up the Developer Console.

All its configurations are stored in a dcp file.

We recommend to place this file to the root directory of the entire Solution, with the same name as the name of the Solution file.

If we stick to this described structure of subdirectories, as well as recommended location of the dcp file, Developer Console will automatically prepare the settings of default subdirectories with the final DLLs of the individual Plugins.

In the case of managing localizations within the Plugin development, it will also set up a directory for it.

The next part of the lesson will guide you through the particular steps of working with the Developer Console.