



Pocket Investor

**Plugin Development
Reference**

Pocket Investor is an innovative investment manager designed for Windows Mobile Classic and Professional (touch-screen enabled) devices which combines many features in a compact and powerful solution especially designed for small and medium investors in a mobile scenario.

Pocket Investor allows 3rd party developers to extend the program functionality using a simple plugin architecture. It is possible to create custom data viewers, special notification mechanisms for prices and alerts, export views in various formats or integrate PI with other applications and device interfaces, for example.

See examples of plugins at www.acquasys.com/investor/plugins.htm

Creating a Plugin

- 1) Using Microsoft Visual Studio .NET, create a Smart Device project of type **Class Library**. Plugins can be written in any .NET language (VB.NET, C#);
- 2) In project properties, enter an assembly name ending in "**plugin.dll**" (ex. HelloWorld.plugin.dll or HelloWorldPlugin.dll). This is necessary for the plugin to be recognized by Pocket Investor.
- 3) Copy assembly **PIPlugInt.dll** from Pocket Investor installation folder to your project folder and add a reference to it in your project.
- 4) Create a public class that implements one or more of the interfaces described below.

That's it, happy coding!

Plugin Interfaces

Plugins can be written in any .NET language (VB.NET, C#) and are deployed as a class library (dll).

A plugin class must implement one or more of the following interfaces, that are declared in assembly PIPlugInt.dll:

Acquasys.PocketInvestor.Plugin.IPlugin

This is the base interface that all plugin interfaces inherit from. Normally you will not implement this interface directly, unless you are writing a plugin that will be executed only when PI is started or finished.

The HTC Wheel plugin is an example of a plugin that implements this interface.

Method to implement:

public void Dispose()

The Dispose() method gives the plugin an opportunity to deallocate resources or perform any cleanup code at program finalization. If this is not needed by your plugin, you must implement an empty method.

Acquasys.PocketInvestor.Plugin.IPriceUpdatePlugin

Implement this interface if you want to notify your plugin class each time a price update is performed in Pocket Investor.

The Watch plugin is an example that implements this interface.

Methods to implement:

public void Execute(PriceInfo[] prices)

The Execute() method receives an array of PriceInfo objects as parameter. Each PriceInfo instance contains the symbol, date, price and price change for each investment updated.

public void Dispose()

See IPlugin interface.

Acquasys.PocketInvestor.Plugin.IAlertPlugin

Implement this interface if you want to notify your plugin class each time alerts are fired in Pocket Investor.

The Watch plugin is an example that implements this interface.

Methods to implement:

public void Execute(string[] alerts)

The Execute() method receives an array of strings containing the alert messages.

public void Dispose()

See IPlugin interface.

Acquasys.PocketInvestor.Plugin.IViewUpdatePlugin

This interface must be implemented for plugins that want to receive notifications of any changes in the current view displayed in Pocket Investor.

The PC Viewer plugin is an example that implements this interface.

Methods to implement:

public void Execute(string viewName, string[] columns, string[,] data)

At each view update, the Execute method is called with the following arguments:

viewName: String containing the name of the current view

columns: Array of strings containing all column titles

data: Multi-dimensional array of strings that represents the data displayed in the view as grid cells. Each element in the array correspond to a cell in the view. For example, data[0,1] contains the second column in the first line of the view.

public void Dispose()

See IPlugin interface.

Acquasys.PocketInvestor.Plugin.IViewExportPlugin

This interface is identical to IViewUpdatePlugin interface, but plugins implementing it are notified only when the user explicitly request a view export using Menu/Tools/Export View from within Pocket Investor.

The CSV Export plugin is an example that implements this interface.

Methods to implement:

public void Execute(string viewName, string[] columns, string[,] data)

See IViewUpdatePlugin interface.

public void Dispose()

See IPlugin interface.

Acquasys.PocketInvestor.Plugin.ICustomUpdatePlugin

Implement this interface to provide price updates for investments from a custom source (ex. a different Quote provider, downloaded text files etc).

Methods to implement:

public PriceInfo[] Execute(string[] symbols)

The Execute() method receives a string array with all the investment symbols to get current prices for, and must return a PriceInfo array with quote dates and prices. This method will be called on each update event with all investments that have Update Mode = Custom (Plugin).

public void Dispose()

See IPlugin interface.

Considerations

- As an explicit initialization method is not provided, you must call any startup code needed directly in the class constructor.
- Depending on the interface implemented, the Execute method may be called again when still executing. Prepare it for this possibility.
- Design your plugin to be as small as possible and avoid allocating large resources. Plugins are instantiated at program startup and will live in memory until Pocket Investor is terminated.
- Try to keep a single plugin per assembly. This allows users to choose what functionalities to install on their devices without unnecessary overhead.
- Although a plugin assembly may contain any number of helper classes, declare only the plugin class itself as public and not abstract. This helps Pocket Investor to recognize plugin classes faster.
- You may choose to use the Registry to provide customizable settings for end users. In this case, create a key for your plugin under Pocket Investor settings, as in this example:

HKEY_CURRENT_USER\Software\Acquasys\Pocket Investor\Plugins\MyPlugin

- Let us know about your creations. This way we can indicate them for PI users, and better analyze the impact of future changes in this API according to its usage.

Acquasys Pocket Investor is a trademark of Acquasys Informática Ltda.