Extendable Hello World Translator

Functional Specification

# Functional Requirements

* 1. Must be implemented as a console application
  2. When executed must print “Hello World” (in English)
  3. Must allow configuring one or more translator plug-ins that translate “Hello World” to another language. The configuration file must support an application setting named “Plugins” with the value containing comma-separated list of available plugins.
  4. The translator plugins must allow obtaining the target language and translation of “Hello World” in the target language.
  5. For each configured translator plug-in, the application must print the supported target language and the translated equivalent of “Hello World” in that language.

# Implementation Details

* 1. A class library named Plugin.dll must implement
     1. An abstract translator base class named TranslatorBase with an abstract method string Translate() that returns the translation of “Hello World”.
     2. An attribute named Language with a readonly string property returning the target language.
  2. The plug-ins must be implemented as class library assemblies and put into the Plugins subfolder within the application directory
  3. The plug-in assembly can contain zero or more classes deriving from TranslatorBase (one per target language)
  4. The application must discover all classes in all configured plugin assemblies that derive from TranslatorBase class and for each discovered class instantiate an object of that class and print the result of Translate method invoked for that object.
  5. If a translator class has a LanguageAttribute applied then the application must print the target language specified by that attribute next to the printed translation of “Hello World”. Otherwise append “(n/a)” to the printed translation.

# Expected Output

