Overview

The DYMO Label Framework is a new set of APIs that allow easy integration of printing functionality. The Framework is an evolutionary step for the SDK currently provided by DYMO. The Framework provides three sets of APIs: .NET, COM, and JavaScript.

Differences from the DLS SDK

Below are the major differences between the DLS SDK and the Framework.

No separation for High-Level and Low-Level APIs

In the DLS SDK there are two APIs: high-level and low-level. The high-level API provides the most frequently used features, while the low-level API provides the most advanced features. Unfortunately, these APIs are not compatible and cannot be used together in one project. The Framework combines these into one API. The most frequently used features are still easily available and more advanced features can be used when needed.

Universal API for LabelWriter and LabelManager/Tape printers

In the DLS SDK there are two completely different APIs for printing to LabelWriter and LabelManager/Tape printers. Again, in the Framework they are unified into one API.

No Hidden State/Dependencies on DYMO Label Executable

The DLS SDK has dependencies on the state of the DYMO Label application itself. For example, last opened label, last used printer, a list of recently used templates. It is not used for most applications and can produce very subtle bugs. The Framework does not have this hidden state. When the Framework is loaded there are not any “open” labels, there is no selected printer, etc. The application is responsible for loading/opening a label if needed. If some information, e.g. last used printer needs to be preserved between application sessions, it can be done in any way suitable for the application.

Broad JavaScript support

In the DLS SDK, the support for JavaScript is limited. Though it is possible to call the SDK from Internet Explorer, Firefox on Windows, and Safari on Mac, it is somewhat cumbersome. The user must install an extension/plugin, and JavaScript code is different for each browser. The Framework provides a cross-platform, cross-browser JavaScript library that provides the same simple API for any supported browser. The currently supported browsers are Firefox, Safari, Chrome, and Opera on Windows and Mac, as well as Internet Explorer 6,7,8,9 beta on Windows.

Getting Started

Prerequisites

In order to compile and run applications that use the Framework, DYMO Label v.8 software 8.3.0.xxx must be installed on a client machine. This will install the printer drivers and all necessary libraries needed to run the Framework. DYMO Label v.8 can be downloaded from the Support area of the DYMO Web site at www.dymo.com.

Samples

The easiest way to start using the Framework is to download the DLS SDK which contains many samples for using the Framework. There are samples for all APIs provided by the Framework. See C++ folder for COM API samples, dotNET folder for .NET API samples, and JavaScript folder for JavaScript API samples.

API Reference

Comprehensive MSDN like reference documentation is available from the Developer's area of the DYMO Web site at www.dymo.com.

Blog

More samples and other information can be found on the developer’s blog http://developers.dymo.com/

API Overview

Below is a brief overview of the three APIs the Framework provides.

.NET API

The .NET API is exposed throughout the DYMO.Label.Framework.dll assembly. In your project, select “Add Reference” and then select the assembly from the list of available assemblies. After that, the types defined by the assembly will be available for usage. The root namespace is DYMO.Label.Framework. The typical use-case would be “opening” a label using Label.Open() or Label.OpenXml() methods, then using returned ILabel object to manipulate label content and to print. The sample below loads a label from a file, sets the content of an Address object, and prints it on “DYMO LabelWriter 450 Turbo” printer.

```
// obtain a reference to ILabel by loading a label from a file
ILabel label = Label.Open("MyLabel.label");
// put address data into the first address object
if (label.AddressObjectCount > 0)
    label.SetAddressText(0, "DYMO\n828 San Pablo Ave STE 101\nAlbany CA 94706");
// print it
label.Print("DYMO LabelWriter 450 Turbo");
```

For complete details, see the .NET API.

JavaScript API

The JavaScript API provides methods similar to the .NET API, though currently no “low-level” functionality is provided. A web page should reference DYMO.Label.Framework.js before calling any Framework functions. The example above looks as follows:

```
$.get('http://myserver.com/MyLabel.label', function(labelXml)
{
    var label = dymo.label.framework.openLabelXml(labelXml);
    if (label.getAddressObjectCount() > 0)
        label.setAddressText(0, "DYMO\n828 San Pablo Ave\nAlbany CA 94706");
    label.print("DYMO LabelWriter 450 Turbo");
});
```

For complete details, see the JavaScript API.

COM API

The COM API is very similar to the .NET API. The major differences are as follows:

- Static methods and most constructors are available as methods/factory methods of the “DYMO.Label.Framework” COM-class
- Overloaded methods are available as methods with different names.

The above example look as follows:

```
using namespace DYMO_Label_Framework;
IFrameworkPtr pFramework;
pFramework.CreateInstance(__uuidof(Framework));
ILabelPtr pLabel = pFramework->OpenLabel(L"MyLabel.label");
pLabel->SetAddressText(0, L"DYMO\n828 San Pablo Ave STE 101\nAlbany CA 94706");
pFramework->PrintLabel(L"DYMO LabelWriter 450 Turbo", L"", pLabel->SaveToXml(),
L"");
```