

DYMO Label Software (DLS) Version 7.6 SDK

NOTES

1. All SDK documentation and Sample code can now be accessed from the Start menu under the **DYMO Label Software SDK** folder. By default all SDK documentation and Sample code is saved in C:\Program Files\DYMO Label\DLS_SDK.
2. This SDK requires that DLS v7.6 (or later) software be installed on the machine using the SDK functions. It also requires that a full install of DLS, including all template groups, be present. A copy of DLS is provided with all LabelWriter printers. Additionally, it can be downloaded from our web site at www.dymo.com.
3. To view the SDK Manual, you will need Adobe Acrobat Reader.

Contents

1. What's New
2. Files Included in the SDK

1. What's New

New IDymoAddIn5 interface allows you to open label files using a URL (see the SDK Manual for OpenURL() function for more details).

High and Low Level COM samples have been updated to use functions illustrating printing to a specified roll on a LabelWriter Twin Turbo.

Users who use DLL calls can now choose to suppress error messages so their application can run uninterrupted.

The SDK now supports printing on the new LabelWriter 400, Duo, Twin Turbo, ExecuLabel LM450 (LabelManager 450), ExecuLabel LP350 (LabelPoint 350), and LabelManager PC II.

This latest version of the DYMO SDK includes new .NET samples in C#, VB, ASP.NET, and C++.

2. Files Included in the SDK

Below is a short description of all files installed by the SDK:

The **Documentation Folder** contains the **DYMO Label Software SDK Manual.pdf** file. This is the reference manual for the DYMO SDK. It contains full details on all functions, properties and structures used by the SDK. It is recommended that you read this manual before you start developing with the SDK.

This folder also contains the **Windows SDK FAQ.pdf** file. This FAQ file contains answers to the most commonly asked questions concerning our SDK. Please refer to it before contacting the SDK support email.

2.1. High Level COM

The **High Level COM** folder contains sample code that uses the high-level COM interface of the SDK. The sample applications show how to open a label file, set data on label objects that are on the label file, and print the label. This is by far the most common application for the DYMO SDK.

Samples are provided in Delphi, VB, Visual C++, C#.NET, VB.NET, ASP.NET, Visual FoxPro, Access, and Power Builder. In addition, new .NET samples are provided and illustrate roll selection in the LabelWriter Twin Turbo printer.

To illustrate D1 tape printing using a LabelWriter Duo printer, we have provided a sample written in VB. You will find this in a separate folder named **Tape SDK Sample** within the **High Level COM** folder.

To illustrate the SmartPaste functionality (a patented DYMO technology), we have provided a sample written in C++ that allows you to SmartPaste data from the Clipboard or a comma delimited file.

2.2. Low Level COM

The **Low Level COM** folder contains sample code that uses the low-level COM interface in the SDK. This interface is significantly more difficult to develop and debug than the high-level COM interface, and should not be attempted without first trying the high-level COM interface. The samples show how to open a label file, create and modify label objects on the label, and render the label on the screen.

Low-level COM samples are provided in Delphi, VB, and Visual C++.

2.3. IE Scripts

The **IE Scripts** folder contains JavaScript and VBScript samples that illustrate using the high-level COM interface to print a label from Internet Explorer.

Note that these scripts are only compatible with Microsoft Internet Explorer. A separate example for printing from the Firefox browser is provided elsewhere.

2.4. XPCOM

The **XPCOM** folder contains necessary files for printing from the Firefox browser. DYMO provides XPCOM wrapper objects that provide access to the high-level COM interface of the SDK. To use these wrapper objects, you will need to install them. They are located in the **XPCOM** folder and may be installed by opening the **install.htm** file contained in the **XPCOM** folder.

2.5. Firefox Scripts

The **Firefox Scripts** folder contains a sample that illustrates label printing using the high-level COM interface through the XPCOM wrapper in Firefox.

2.6. DLL Calls

The **DLL Calls** folder contains sample code that uses the DLL calls provided by the SDK. Like the low-level COM interface, the DLL interface is significantly more difficult to develop and debug than the high-level COM interface, and should not be attempted without first trying the high-level COM interface. The sample applications illustrate using DLL calls to open a label file, create label objects on the file, and render the label on the screen. The samples are in Delphi and Visual C++.

2.7. DDE Calls

The **DDE Calls** folder contains sample code that uses the DDE interface to control DLS to print labels. The sample applications show how to open a label file, set data on the objects that are on the label, and print the label. The sample code is provided in Delphi, VB, Visual C++, and Access.

2.8. Paper Size

The **Paper Size** folder contains VB sample code that shows you how to select a paper size in a printer driver when printing using Windows API.

Note that the DDE functionality in the SDK is only provided for backwards compatibility. New applications should use the equivalent high-level COM interfaces that are also provided by the SDK.