

the professional  
labeling software

[www.nicelabel.com](http://www.nicelabel.com), [info@nicelabel.com](mailto:info@nicelabel.com)



# NiceLabel WebSDK Programming Guide

Version 20080925-08

© 2008 Euro Plus d.o.o. & Niceware® International, LLC

All rights reserved.

[www.nicelabel.com](http://www.nicelabel.com)



**European Office**  
Euro Plus d.o.o.  
Ulica Lojzeta Hrovata 4c  
SI-4000 Kranj, Slovenia  
Tel.: +386 4 280 50 00  
Fax: +386 4 233 11 48  
[www.europlus.si](http://www.europlus.si)  
[info@europlus.si](mailto:info@europlus.si)



**North American Office**  
Niceware International, LLC  
10437 Innovation Drive, Suite 147  
Milwaukee, WI 53226  
Tel.: +1 414 476 6423  
Fax: +1 414 476 7955  
[www.nicewareintl.com](http://www.nicewareintl.com)  
[info@nicewareintl.com](mailto:info@nicewareintl.com)

# Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>4</b>
1.1	<b>Welcome to the NiceLabel WebSDK User's Guide .....</b>	<b>4</b>
1.2	<b>Inside the NiceLabel WebSDK Documentation.....</b>	<b>4</b>
1.3	<b>Accelerated Introduction.....</b>	<b>5</b>
1.4	<b>Conventions Used In This Guide.....</b>	<b>5</b>
1.5	<b>Product Support.....</b>	<b>6</b>
<b>2</b>	<b>WebSDK Overview .....</b>	<b>7</b>
2.1	<b>Why web printing?.....</b>	<b>7</b>
2.2	<b>Components of WebSDK .....</b>	<b>7</b>
2.3	<b>Architectural Overview .....</b>	<b>8</b>
2.4	<b>Technical Structure.....</b>	<b>9</b>
2.5	<b>Server functionality.....</b>	<b>9</b>
2.6	<b>Client functionality.....</b>	<b>10</b>
<b>3</b>	<b>Installation .....</b>	<b>11</b>
3.1	<b>System Requirements .....</b>	<b>11</b>
3.2	<b>Installation of NiceLabel WebSDK .....</b>	<b>12</b>
3.2.1	Installable Components .....	12
3.2.2	Deployment setup.....	12
3.2.3	Developer Setup .....	13
3.2.4	License Certificate .....	15
<b>4</b>	<b>Programming Reference.....</b>	<b>16</b>
4.1	<b>Exception handling.....</b>	<b>16</b>
4.2	<b>PrintServer Web Service .....</b>	<b>17</b>
4.2.1	NPSService Class .....	17
4.2.2	FileLocation Class .....	30
4.3	<b>StorageServer Web Service .....</b>	<b>33</b>
4.3.1	StorageService Class .....	33
4.4	<b>NLWAX Web Browser ActiveX and Java Control .....</b>	<b>37</b>
4.4.1	NLWAXForm Class.....	38
<b>5</b>	<b>NiceLabel WebSDK Sample Application.....</b>	<b>44</b>
5.1	<b>NPSSampleApplication .....</b>	<b>44</b>
5.1.1	SourceCode.....	44
5.2	<b>Explanation of Sample Application.....</b>	<b>44</b>
5.2.1	Integration of NLWAX.ocx .....	44
5.2.2	Server-Side Code Explanation .....	45
5.2.3	Additional JavaScript files.....	49
5.2.4	Web.Config Settings .....	49
<b>6</b>	<b>Contacts .....</b>	<b>51</b>

## Disclaimer

Euro Plus d.o.o. & Niceware® International, LLC reserve the right to make changes in specifications and other information contained in this document without prior notice, and the reader should in all cases consult Euro Plus d.o.o. & Niceware® International, LLC to determine whether any such changes have been made. The information in this publication does not represent a commitment on the part of Euro Plus d.o.o. or Niceware® International, LLC. Euro Plus d.o.o. & Niceware® International, LLC shall not be liable for technical or editorial errors or omissions contained herein; nor for incidental or consequential damages resulting from the furnishing, performance, or use of this material. This document contains proprietary information that is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced, or translated into another language without the prior written consent of Euro Plus d.o.o. & Niceware® International, LLC.

Web Addresses: <http://www.europlus.si> <http://www.nicewareintl.com>

## Trademarks

NiceLabel®, NiceLabel Pro®, NiceLabel WebSDK®, NiceLabel SDK®, and NiceDriver® are trademarks or registered trademarks of Euro Plus d.o.o in the U.S.A. and other countries. Niceware® is a registered trademark of Niceware International, LLC.

Microsoft, Visual Studio, Visual C#, Visual Basic, Windows, and the Windows logo are trademarks or registered trademarks of Microsoft Corporation.

Other brands and their products are trademarks or registered trademarks of their respective holders and should be noted as such.

# 1 Introduction

---

## 1.1 Welcome to the NiceLabel WebSDK User's Guide

NiceLabel WebSDK provides all of the tools necessary to integrate label printing with web based applications. These developer tools include Web Services, NiceLabel Print Engine, browser ActiveX and Java control, sample Web applications, and associated documentation. NiceLabel WebSDK allows developers to programmatically access NiceLabel on the server and send print jobs to visitors' printers.

NiceLabel WebSDK also provides an ActiveX or Java component which will be executed on the client computer (automatically downloaded by the Web browser). This component will transfer the print streams generated on the server to the local ports of the client computer.

## 1.2 Inside the NiceLabel WebSDK Documentation

The following is a brief overview of what to expect in each section of this document.

### Introduction - Section 1

Familiarizes you with the fundamentals of the NiceLabel WebSDK and the contents of this documentation.

This section also includes the following topics:

- **Accelerated Introduction** – Steps to get up to speed as quickly as possible.
- **Conventions Used in this Guide** - Describes conventions used throughout this document.
- **Product Support** - How to obtain additional assistance.

### WebSDK Overview - Section 2

Contains diagrams and explanations of how NiceLabel WebSDK is architected and deployed.

### Installation - Section 3

Reviews System Requirements and Installation procedures for NiceLabel WebSDK.

### Reference - Section 4

Supplies syntax, code examples, and related information for each class contained in the NiceLabel WebSDK, including the following main sections:

- **PrintServer Web Service**
- **StorageServer Web Service**
- **NLWAX Web Browser ActiveX and Java Control**

### Samples - Section 5

The Samples Section provides brief descriptions of sample web applications included in the NiceLabel WebSDK.

## 1.3 Accelerated Introduction

For an accelerated introduction to NiceLabel WebSDK, follow these steps in sequence:

- Scan the **WebSDK Overview** section to gain a solid understanding of how the different tools inter-relate.
- Review the **Installation** section.
- Become familiar with the **Programming Reference** section.
- Load the **Sample** application for a hands-on view of coding practices.

As you begin to develop your applications, continue to use the NiceLabel WebSDK documentation as a primary source of information. Utilize the various **Support** options if you encounter issues not covered in the User's Guide.

## 1.4 Conventions Used In This Guide

Every attempt has been made to ensure that this document is easy to use and understandable for all users. To ensure adequate coverage while being as concise as possible a number of conventions have been utilized throughout this document. This page describes these conventions.

### Web Service Namespaces

When establishing web references to the PrintServer and StorageServer web services, the user is required to create web reference names. These names are used within the project as the base namespaces for the web reference classes. Since these namespaces are defined by the user, but occasionally referenced by name within this document, the following naming conventions have been established:

Web Service	Namespace
PrintServer	PrintService
StorageServer	NiceStorageService

For situations where an actual namespace is required (e.g. code examples), the above namespaces will be used.

### Code Example Conventions

Many pages in the Reference section contain an Example section. Many examples use objects based on the NPSService, FileLocation, and StorageService classes and showing their instantiation in each example will only detract from the intent of that example. The following shows the instantiation of each class which should be assumed in all examples.

#### Use of the NicePrintService object

[Visual Basic]

```
Dim NicePrintService As New PrintService.NPSService
```

```
[C#]  
PrintService.NPSService NicePrintService = new PrintService.NPSService;
```

### Use of the LabelFile object

```
[Visual Basic]  
Dim LabelFile As New PrintService.FileLocation
```

```
[C#]  
PrintService.FileLocation LabelFile = new PrintService.FileLocation;
```

### Use of the LabelStore object

```
[Visual Basic]  
Dim LabelStore As New NiceStorageService.StorageService
```

```
[C#]  
StorageService.StorageService LabelStore = new NiceStorageService.StorageService;
```

## A Note on Code Examples

Most Examples demonstrate usage of a member as it could exist in an actual application. They are intended solely as examples of proper usage. They are not intended to be a part of any actual application.

## 1.5 Product Support

This section provides information on product support options and how to provide feedback about the NiceLabel WebSDK.

### NiceLabel Forums

For technical questions about the NiceLabel WebSDK, use our **NiceLabel Forums** at <http://forums.nicelabel.com/>

### NiceLabel Tutorials

For self-guided lessons covering all aspects of label design and using the NiceLabel family of products, utilize our **Tutorials** at <http://www.nicelabel.com/Learning-center/Tutorials>

### Getting Help from NiceLabel Technical Support Services

For current information regarding technical support for the NiceLabel WebSDK, visit <http://www.nicelabel.com/Support>.

## 2 WebSDK Overview

---

### 2.1 Why web printing?

- No manual software installations on workstations
- Use Web Browser as the main tool
- Easier maintenance - software updates are performed on server only – no need to update workstations
- Opening a new business model – provide (label) printing as service

#### Standard label printing process:

**1. Label design (periodically):**

Label formats are designed in standard label designers (NiceLabel Express, NiceLabel Pro) and stored on a web server. Typical users involved: label designers, system administrators  
required: NiceLabel label design knowledge; web server access

**2. Integration and setting up a label printing solution (one time):**

All software components are installed and set up properly (to a server and optionally clients), label printing engine is integrated in a 3<sup>rd</sup> party app (using web technologies like ASP.NET, web services, etc.). Typical users involved: system administrators, web application programmers  
required: web app programming knowledge to integrate label printing in HTML, ASP, ASP.NET; NiceLabel programming knowledge; (IIS) web server management control

**3. Label printing (daily):**

From a web application; typical users involved: standard web application users – daily  
required: no special knowledge or user rights; right for (automatic) web component installation (optionally)

### 2.2 Components of WebSDK

#### NicePrintServer

Web Service which is used by the custom Web Application.

#### NiceEngine

Used to process label data and generate preview images on the server.

#### NiceDrivers

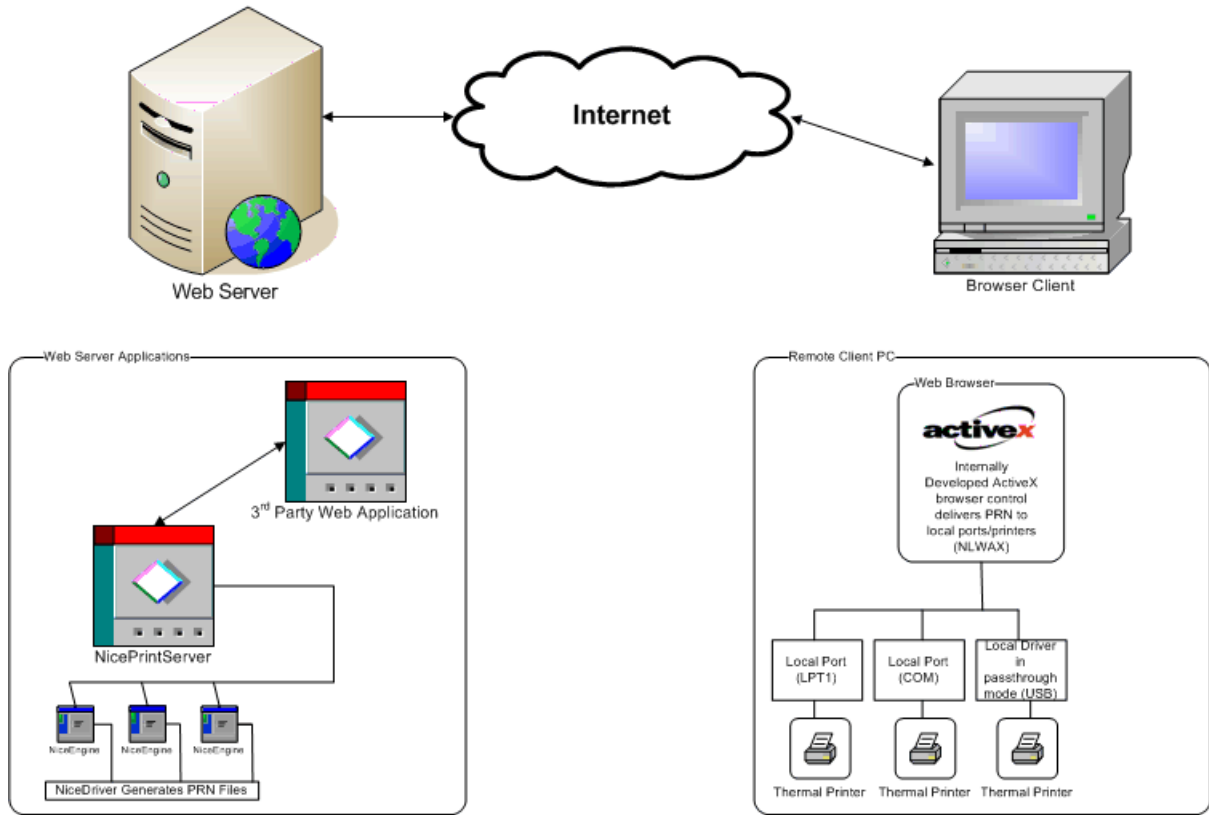
Used to create print streams on the server.

#### NLWAX (NiceLabel Web ActiveX and Java Component)

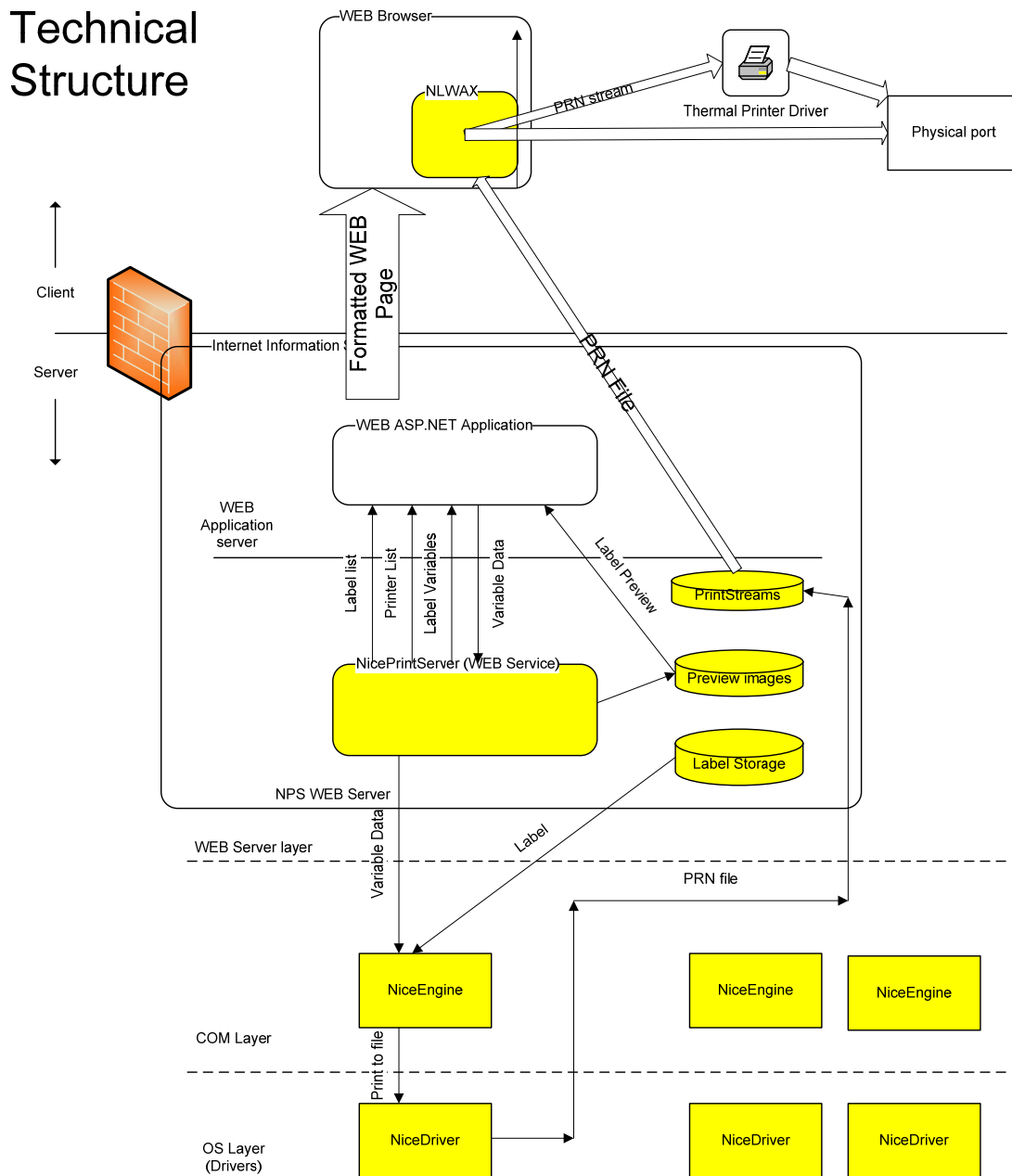
Client side ActiveX and Java component for transferring the print streams to the client physical ports.

## 2.3 Architectural Overview

### NiceLabel WebSDK General Overview



## 2.4 Technical Structure



NL WEB SDK Components

## 2.5 Server functionality

NiceLabel WebSDK is designed primarily to be used by ASP.NET applications, running on Windows operating system under IIS. NiceLabel WebSDK can be installed on the same web server

(running within IIS) as the web application, but is not required. Web application can use web service interfaces to communicate with NiceLabel WebSDK.

The following functionality is provided by the web services:

**Get List of available printers.** StorageServer web service will check the system and return list of all printers installed on the server (where NiceLabel WebSDK is running)

**Get list of available labels.** NPSService web service will return a list of labels (LBL files), that are available on the server and can be used for web printing. Labels can be grouped into hierarchical structure. This structure is part of the web service functionality.

**Get list of label variables.** When the user selects a label to be used, NPSService will return a dataset of variables and their properties that are available for that label.

**Create Label Preview.** When the web application has gathered the variable data from the user, it can make requests to generate the label preview (using the current variable data)

**Create Print Stream.** When the web application has gathered the variable data from the user, it can make request to generate the print stream. In this step, NiceEngine and a Windows printer driver will be used to generate the print stream. The web application is required to pass the following data to NPSService: Label name, Variable data, Printer name, and Quantity.

Preview Images and Print streams are stored in a temporary location on the web server (where NiceLabel WebSDK is running). NPSService returns the URLs of these locations to the web application which can then use the files as appropriate.

## 2.6 Client functionality

To enable clients (running only web browser) to print to their local ports (without printer driver being installed), the ActiveX and Java component (NLWAX.ocx) must be installed on the client. The custom web application must create the web page that will automatically install the component on the client computer. To control the NLWAX, the web application must include scripting code in the generated web page. This script will call NLWAX to download the PRN file (provide the URL link) and send it to the local printer. The user can have the option to select which local port will be used for printing. In addition, user will be able to select locally installed printers and print through the printer driver (required for USB printers).

Printing without printer drivers on the client computer is possible only to LPT & COM ports (optionally to network ports). For USB printing, a printer driver must be locally installed, and will be used only to pass the generated stream to the printer.

## 3 Installation

---

This section provides details for installing NiceLabel WebSDK on deployment web servers as well as development computers.

### 3.1 System Requirements

To ensure adequate performance, NiceLabel WebSDK has the following minimum and recommended system requirements for client and server applications. Please ensure that the target systems meet these requirements before starting installation process.

#### Operating System Requirements

The NiceLabel WebSDK is supported on the following platforms.

Scenario	Operating system
Server	Microsoft® Windows® Server 2003 family
Client	Microsoft® Windows® 98 Second Edition or higher

#### Additional Software Requirements

Scenario	Application	Minimum Version
Server	Internet Information Systems	6.0
	Microsoft .NET Framework	2.0
	NiceLabel® NiceDrivers	4.1
	Database Engines	As required by labels.
Client	Internet Explorer	5.01
	Mozilla based browser	1.7

#### Hardware Requirements

Scenario	Required processor	Recommended processor	Required RAM	Recommended RAM
Server	Pentium 1.6 GHz	Pentium 2.0 GHz or faster	512 MB*	1.0 GB or higher
Client	Pentium 90 MHz*	Pentium 90 MHz or faster	32 MB*	96 MB or higher

\*Or the minimum required by the operating system, whichever is higher.

## 3.2 Installation of NiceLabel WebSDK

The NiceLabel WebSDK Installation utility is used to install WebSDK on a development computer or on the deployment server. The Setup procedure has two modes of operation to choose from at the beginning of installation:

- Deployment setup
- Developer setup

\* Note: *Client Setup* is not an installation option as the NLWAX control is the only component that is required on a client (“web site visitor”) computer. This component will automatically be installed by the browser the first time the page is loaded.

Both Deployment and Developer installations require the resources of the Microsoft .NET Framework 2.0. If this framework is not currently available it will automatically be installed.

Installation will automatically set the ASP.NET version for both virtual directories to 2.0.



### 3.2.1 Installable Components

- NicePrintServer Web Services
  - PrintServer web service
  - StorageServer web service
  - Sample Labels
- NiceEngine
- Sample Web application
- Source code for sample application
- Documentation

### 3.2.2 Deployment setup

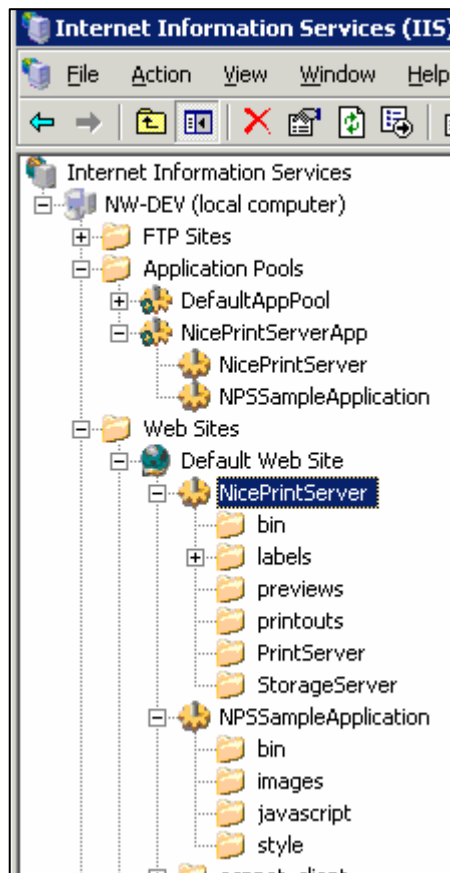
Deployment setup is the default mode in the wizard which will install the following items to the server:

- NicePrintServer Web Services (consists of PrintServer and StorageServer web services)
- NiceEngine

User needs to select the target folder for installation. In this folder all the files for both web services will be installed. NiceEngine is installed under the operating system’s Shared Files folder (default location is C:\Program Files\Common Files\EuroPlus Shared).

Several modifications will be made within IIS to enable full functionality of WebSDK.

- **Creation of the NicePrintServer Virtual Directory**  
The NicePrintServer Virtual Directory will be created under the Default Web Site and contains both the PrintServer and StorageServer web services. The installer will set NicePrintServer to use ASP.NET version 2.0.
- **Creation of the NicePrintServerApp Application Pool**  
To allow the PrintServer web services full access to NiceEngine it is required to have NicePrintServer run within a custom Application Pool. The NicePrintServerApp Application Pool is created for this purpose and NicePrintServer is automatically set to use this.
- **Creation of Nice Print Server User Account**  
The NPS\_USER user account will be created and automatically added to the IIS\_WPG group. The NicePrintServerApp Application Pool will run under this account.



When running WebSDK setup in Deployment mode, there will be no programs added to the Start Menu. The only change is in Add/Remove Software where a new entry will be created for **NiceLabel WebSDK**.

### Silent setup

WebSDK setup can be executed in Silent mode. This can be performed by running the installation program with /s parameter:

```
NiceLabel-WebSDK5.exe /s
```

When running setup in silent mode, no user intervention is required (no windows are opened). The setup will be automatically executed in Deployment mode. This mode of setup can be used for embedding the installation within another web application setup.

### 3.2.3 Developer Setup

Developer setup mode is intended for installing WebSDK on the Developer's computer. Setup will prompt the user to agree with the license agreement and a readme file will be displayed.

In addition to the components of the Deployment setup, the following features can optionally be installed:

- Sample Web Application
- Source Code for Sample Web Application



If the Sample Web Application is selected to be installed it will be created a new Virtual folder is in IIS named NPSSampleApplication.

This application can be used to check operation of the web services, to perform web printing, etc.

## NiceLabel WebSDK Sample

### Current Printing Options

Label: AIAG B-10.lbl  
 Printer: ZEBRA LP 2844-Z  
 Port: LPT1:  
 Print Quantity:  Print Label

### Label Preview

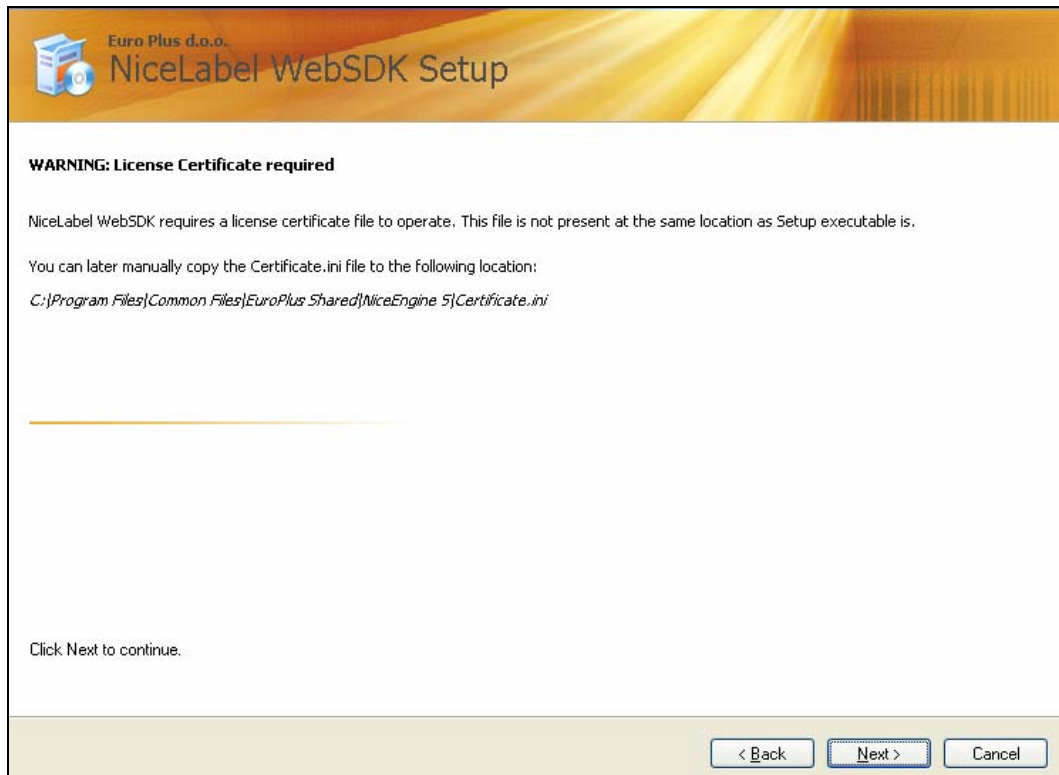
Available Printers		Labels	
Citizen CLP 7401	Group 1	AIAG B-10.lbl	
HP LaserJet 2200 Se	Group 2	AIAG B10_6x9_PrimaryMetalsTag.lbl	
Printronix SL5304R N		CIRCUIT_CITY.lbl	
SATO CL408e		Data.lbl	
ZEBRA LP 2844-Z		form - big.lbl	
ZEBRA R110Xi 200D		MIL_STD_129PGenericCargo~LaserPr	
ZEBRA R2844-Z		NUTRITION_FACTS.lbl	
ZEBRA R4Mplus			
ZEBRA TLP 2824			
ZEBRA Z4Mplus 203			

### Label Variables

Name	Prompt	Format	Length	Fixed Length	Required	Value	
LotNumber	Enter supplier Lot Number:	Numeric	14	<input type="checkbox"/>	<input type="checkbox"/>	12345678900987	<a href="#">Edit</a>
EQuantity	Enter quantity:	Numeric	6	<input type="checkbox"/>	<input type="checkbox"/>	600000	<a href="#">Edit</a>
SupplierPN	Enter supplier Part No.:	All	14	<input type="checkbox"/>	<input type="checkbox"/>	2223U765432988	<a href="#">Edit</a>
Supplier	Enter supplier:	All	8	<input type="checkbox"/>	<input type="checkbox"/>	K98-0999	<a href="#">Edit</a>
PONumber	Enter P.O. Number	All	8	<input type="checkbox"/>	<input type="checkbox"/>	AQ123455	<a href="#">Edit</a>
SerIncrement	Enter increment step:	Numeric	9	<input type="checkbox"/>	<input type="checkbox"/>	1	<a href="#">Edit</a>
PartNo	Enter part number:	All	19	<input type="checkbox"/>	<input type="checkbox"/>	BARCODE SOFTWARE	<a href="#">Edit</a>
Serial	Enter starting serial number:	Numeric	9	<input type="checkbox"/>	<input type="checkbox"/>	000000001	<a href="#">Edit</a>

### 3.2.4 License Certificate

WebSDK and SDK (PrintEngine) require a license certificate file to operate. Setup utility will check the location of original setup file to find the Certificate.ini file. If this file is not found the following warning screen will be displayed before installation begins:



If the Certificate.ini file is found, it will be automatically copied to the proper location.

## 4 Programming Reference

---

This section contains reference documentation of the public components that constitute the NiceLabel WebSDK.

### PrintServer Web Service

Documents the public members available in the PrintServer Web Service.

### StorageServer Web Service

Documents the public members available in the StorageServer Web Service.

### NLWAX Web Browser ActiveX and Java Control

Documents the public members exposed by the NLWAX.OCX.

## 4.1 Exception handling

In case of errors during operation of NiceLabel WebSDK Web Services, the exception is thrown to the calling application. The exception caught by the client is always SoapException.

Detailed information about the cause of the error is stored in the Details member of the SoapException. This member contains XML document with the following structure:

```
<detail>
  <Error xmlns=" http://www.nicelabel.com/PrintServer">
    <ErrorType>PrintServerException</ErrorType>
    <ErrorID>1</ErrorID>
    <ErrorMessage>Message</ErrorMessage>
  </Error>
</detail>
```

Application can use the Error detailed attributes for further diagnostics.

### Exception types

Exception type	Description
License Exception	An error related to the licensing is thrown. Error IDs can be: 1 – License not found 2 – License not valid 3 – License expired
PrintEngine Exception	There was a problem using the PrintEngine. The ErrorID parameter holds the ErrorID returned from NiceEngine.
PrintServer Exception	General error with PrintServer WebService. ErrorID is always -1.
StorageServer Exception	General error with StorageServer WebService. ErrorID is always -1.

## 4.2 PrintServer Web Service

PrintServer Web Service contains two base classes for handling label printing and storage

### Classes

Class	Description
NPSService	NPSService class provides main label printing functionality
FileLocation	Provides functionality to manage label files in hierarchical groups.

### 4.2.1 NPSService Class

NPSService class provides main label printing functionality, including the following key functions:

- Retrieve the list of available printers on the server
- Retrieve the list of variables for specific label
- Create preview image for specific label with variable data
- Create print stream for specific label with variable data

### Web Reference

NPSService.asmx

### Signature

[Visual Basic]  
 Public Class **NPSService**  
 Inherits System.Web.Services.Protocols.SoapHttpClientProtocol

[C#]  
 public class **NPSService** : System.Web.Services.Protocols.SoapHttpClientProtocol

### Example

The following example demonstrates instantiating a new object based on the NPSService class.

```
[Visual Basic]
Public NicePrintService As PrintService.NPSService
Sub Application_Start(ByVal sender As Object, ByVal e As EventArgs)
    ' Create new instance of NPSService on application startup
    NicePrintService = New PrintService.NPSService
End Sub
```

```
[C#]
PrintService.NPSService NicePrintService;
void Application_Start(object sender, EventArgs e)
{
    // Create new instance of NPSService on application startup
    NicePrintService = new PrintService.NPSService();
}
```

### 4.2.1.1 NPSService Members

#### Public Constructors

Constructors	Description
NPSService Constructor	Initializes a new instance of the <b>NPSService</b> class.

[Visual Basic]  
Public Sub **New**()

[C#]  
public **NPSService**()

#### Public Methods

Public Methods	Description
CleanUpPrintEngine	Releases all instances of PrintEngine running on server.
CreatePrintFile	Creates a print stream file for the label.
CreatePdfFile	Creates a PDF file for the label
GetPrinterList	Gets all the names of the printers installed on the server.
GetVariables	Returns a DataSet which contains all variables for the label
SetPrintEngineCount	Sets up the number of print engines.
TestEngineConnection	Tests the connection between PrintServer Web service and PrintEngine.
UpdatePreviewFile	Creates a label preview image for the label.

#### 4.2.1.1.1 CleanUpPrintEngine Method

This method releases all instances of PrintEngine running on server. It is recommended to call this method from the Application\_End event of the web application. This should be called only when the calling application is the only one using that service. Do not call this method if there are other applications using NPSService Web Service since it can interfere with their use of PrintEngine. Please see SetPrintEngineCount method for information on using multiple instances of PrintEngine.

#### Member of

PrintService.NPSService

#### Signature

[Visual Basic]  
Public Sub **CleanUpPrintEngine**()

[C#]  
public void **CleanUpPrintEngine**()

## Parameters

Parameter	Data Type	Description
none	N/A	N/A

## Return Value

Data Type	Description
None/void	N/A

## Exceptions

Exception Types	Condition
PrintServerException	Failed cleaning up PrintEngines.

## Example

The following example demonstrates calling the `CleanUpPrintEngine` method from a web application's `Application_End` event. (Please review *Conventions* for notes on the use of the `NicePrintService` object.)

```
[Visual Basic]
Sub Application_End(ByVal sender As Object, ByVal e As EventArgs)
    ' Clean up all instances of NiceEngine on application shutdown
    NicePrintService.CleanUpPrintEngine()
End Sub
```

```
[C#]
void Application_End(object sender, EventArgs e)
{
    // Clean up all instances of NiceEngine on application shutdown
    NicePrintService.CleanUpPrintEngine();
}
```

### 4.2.1.1.2 CreatePrintFile Method

This Method is creating a print stream file for the label identified by the `LabelFile` parameter. The print stream is generated for the printer specified in the `Printer` parameter.

The quantity for the print job must be passed in the `Quantity` parameter. Value for this parameter can include also NiceLabel special values (UNLIMITED, VARIABLE).

The name of the print stream file will also include a `.prn` file. This is used to allow caller application to identify the name of the print stream file uniquely to avoid conflicts when multiple users are using the web service.

The Variable data is provided in the parameter `dsVariables`, which is using the same structure as the return value of the `GetVariables` method.

## Member of

PrintService.NPSService

## Signature

[Visual Basic]  
 Public Function **CreatePrintFile**(ByVal *Printer* As String, \_  
   ByVal *LabelFile* As PrintService.FileLocation, \_  
   ByVal *Quantity* As String, \_  
   ByVal *PrnFile* As String, \_  
   ByVal *dsVariables* As System.Data.DataSet) As String  
 [C#]

public string **CreatePrintFile**(string *Printer*,  
 PrintServer.FileLocation *LabelFile*,  
 string *Quantity*,  
 string *PrnFile*,  
 System.Data.DataSet *dsVariables*)

## Parameters

Parameter	Data Type	Description
<i>Printer</i>	System.String	Name of Printer.
<i>LabelFile</i>	PrintService.FileLocation	Holds label's relative path and file name.
<i>Quantity</i>	System.String	Number of labels to be printed. Also accepts special values (UNLIMITED, VARIABLE).
<i>PrnFile</i>	System.String	Unique name for generated print file.
<i>dsVariables</i>	System.Data.DataSet	Dataset that holds a table of label variables.

## Return Value

Data Type	Description
System.String	Relative path and name of generated print stream file

The return value of the method is relative path and name of the generated print stream file. The calling application must use the web service base address (for example "http://machinename/NicePrintServer/") as the prefix and add the return value. This will point to the location of generated print stream file. Calling application must then take care to download the print stream file and send it to the printer.

## Exceptions

Exception Types	Condition
PrintServerException	An unexpected exception occurs during creation of print file.
	Provided label file location is not valid.
PrintEngineException	Engine can not be started.

	Label file can not be opened by the engine.
	Label file can not be printed.
	Maximum count of engine's is already opened and in use. Server is busy.

## Example

The following function demonstrates calling the CreatePrintFile method with exception handling.

[Visual Basic]

```
Public Function RetrievePrinterCodeFile(ByVal PrinterName As String, _
    ByVal UserFile As PrintService.FileLocation, _
    ByVal Quantity As String, _
    ByVal FileName As String, _
    ByVal Vars As System.Data.DataSet) As String

    Dim PrinterCode As String
    Try
        PrinterCode = NicePrintService.CreatePrintFile(PrinterName, UserFile, Quantity, FileName, Vars)
        Return PrinterCode
    Catch ex As Exception
        PrinterCode = "CreatePrintFile() error: " & ex.Message
        Return PrinterCode
    End Try

End Function
```

[C#]

```
public string RetrievePrinterCodeFile(string PrinterName, PrintService.FileLocation UserFile, string
Quantity, string FileName, System.Data.DataSet Vars)
{
    string PrinterCode;
    try
    {
        PrinterCode = NicePrintService.CreatePrintFile(PrinterName, UserFile, Quantity, FileName, Vars);
        return PrinterCode;
    }
    catch (Exception ex)
    {
        PrinterCode = "CreatePrintFile() error: " + ex.Message;
        return PrinterCode;
    }
}
```

### 4.2.1.1.3 CreatePdfFile Method

This Method creates a PDF for the label identified by the LabelFile parameter. The PDF file is created using a PostScript printer driver installed on the server.

By default, WebSDK expects the printer driver "Apple Color LW 12/660 PS" to be installed (this is a standard printer driver that comes with the MS Windows OS). If you wish to use a different printer driver, edit the file "C:\Program Files\Common Files\EuroPlus Shared\NPSWinServer.config". There you will find a node with the name "PostScriptPrinterName", which you should change to reflect the printer that WebSDK will use. If the specified postscript printer is not installed, the CreatePdfFile call will throw an exception.

The quantity for the print job must be passed in the Quantity parameter. Value for this parameter can also include special NiceLabel values (UNLIMITED, VARIABLE).

The Variable data is provided in the parameter *dsVariables*, which is using the same structure as the return value of the *GetVariables* method.

## Member of

PrintService.NPSService

## Signature

[Visual Basic]

```
Public Function CreatePdfFile(ByVal LabelFile As PrintService.FileLocation, _
    ByVal Quantity As String, _
    ByVal PrnFile As String, _
    ByVal dsVariables As System.Data.DataSet) As String
```

[C#]

```
public string CreatePdfFile(PrintServer.FileLocation LabelFile,
    string Quantity,
    string PrnFile,
    System.Data.DataSet dsVariables)
```

## Parameters

Parameter	Data Type	Description
<i>LabelFile</i>	PrintService.FileLocation	Holds label's relative path and file name.
<i>Quantity</i>	System.String	Number of labels to be printed. Also accepts special values (UNLIMITED, VARIABLE).
<i>PrnFile</i>	System.String	Unique name for generated PDF file.
<i>dsVariables</i>	System.Data.DataSet	Dataset that holds a table of label variables.

## Return Value

Data Type	Description
System.String	Relative path and name of generated print stream file

The return value of the method is relative path and name of the generated PDF file. The calling application must use the web service base address (for example "http://machinename/NicePrintServer/") as the prefix and add the return value. This will point to the location of generated PDF file.

## Exceptions

Exception Types	Condition
PrintServerException	An unexpected exception occurs during creation of print file.
	Provided label file location is not valid.
	PostScript printer driver is not installed.

PrintEngineException	Engine can not be started.
	Label file can not be opened by the engine.
	Label file can not be printed.
	Maximum count of engine's is already opened and in use. Server is busy.

#### 4.2.1.1.4 GetPrinterList Method

GetPrinterList method returns an array of all the names of the printers installed on the server.

#### Member of

PrintService.NPSService

#### Signature

[Visual Basic]  
Public Function **GetPrinterList()** As String()

[C#]  
public string[] **GetPrinterList()**

#### Parameters

Parameter	Data Type	Description
none	N/A	N/A

#### Return Value

Data Type	Description
System.String	An array of strings containing all printer names.

#### Exceptions

Exception Types	Condition
PrintServerException	Printer list can not be retrieved.

**Note:** If the Windows Print Spooler Service is not in Started state this method will not be able to retrieve printer list.

#### Example

The following example retrieves the list of printers and binds the returned array to a dropdown list.

**Note:** This example assumes the existence of a DropDownList control named ddlPrinterList.

```

[Visual Basic]
Public Sub BindPrinterList()
    Dim PrinterNames() As String
    ' Retrieve list of server side printers
    PrinterNames = NicePrintService.GetPrinterList
    ' Set array of printer names as the data source for dropdown control
    ddlPrinterList.DataSource = PrinterNames
    ' Bind data to control
    ddlPrinterList.DataBind()
End Sub

[C#]
public void BindPrinterList()
{
    PrintService.NPSService NicePrintService = new PrintService.NPSService;
    string[] PrinterNames;
    // Retrieve list of server side printers
    PrinterNames = NicePrintService.GetPrinterList;
    // Set array of printer names as the data source for dropdown control
    ddlPrinterList.DataSource = PrinterNames;
    // Bind data to control
    ddlPrinterList.DataBind();
}

```

#### 4.2.1.1.5 GetVariables Method

GetVariables returns a DataSet (.NET Framework DataSet class), which contains all variables for the label, identified by the LabelFile parameters. The dataset contains a single Table with following fields

Field Name	Field Description
Name	Name of the variable.
Value	Value of the variable.
Length	Maximum length of the variable.
FixedLength	Boolean, true when variable requires fixed length data.
IsMultiline	Boolean, true when variable supports multi-line text.
LineLength	Length of single line, when variable is multi-line
LineCount	Maximum number of lines, when variable is multi-line.
FormatID	Id of the variable format. Check NiceLabel documentation for more info.
Prompt	The prompt text for prompted variable
ValueRequired	Boolean, true when variable requires a value.

#### Member of

PrintService.NPSService

#### Signature

[Visual Basic]

Public Function **GetVariables**(ByVal *LabelFile* As PrintService.FileLocation) As System.Data.DataSet

[C#]  
public System.Data.DataSet **GetVariables**(PrintServer.FileLocation *LabelFile*)

## Parameters

Parameter	Data Type	Description
<i>LabelFile</i>	PrintService.FileLocation	Holds label's relative path and file name.

## Return Value

Data Type	Description
System.Data.DataSet	Dataset that holds a table of label variables.

## Exceptions

Exception Types	Condition
PrintServerException	An unexpected exception occurs during variable retrieval.
	Provided label file location is not valid.
PrintEngineException	Engine can not be started.
	Label file can not be opened by the engine.
	Label file can not be printed.
	Maximum count of engine's is already opened and in use. Server is busy.

## Example

The following example retrieves the dataset of variables for a specific label.

```
[Visual Basic]
Public Function GetVars(ByVal LabelFile As PrintService.FileLocation) As System.Data.DataSet
    Dim LabelVars As New System.Data.DataSet
    ' Retrieve table of variables from label
    LabelVars = NicePrintService.GetVariables(LabelFile)
    ' Return retrieved dataset
    Return LabelVars
End Function
```

```
[C#]
public System.Data.DataSet GetVars(PrintService.FileLocation LabelFile)
```

```

{
    System.Data.DataSet LabelVars = new System.Data.DataSet();
    // Retrieve table of variables from label
    LabelVars = NicePrintService.GetVariables(LabelFile);
    // Return retrieved dataset
    return LabelVars;
}
    
```

### 4.2.1.1.6 SetPrintEngineCount Method

This method is used to set up the number of print engines that will be used by the NPSService for concurrent printing/generation of print streams. It is recommended to call this method in the Application\_Start event, in case the application is the only one using the WebSDK. Please see CleanUpPrintEngine method for information on cleaning up PrintEngine instances.

#### Member of

PrintService.NPSService

#### Signature

[Visual Basic]  
 Public Sub **SetPrintEngineCount**(ByVal *engineCount* As Integer)

[C#]  
 public void **SetPrintEngineCount**(int *engineCount*)

#### Parameters

Parameter	Data Type	Description
<i>engineCount</i>	System.Int32	Number of simultaneous instances of PrintEngine to allow.

#### Return Value

Data Type	Description
None/void	N/A

#### Exceptions

Exception Types	Condition
PrintEngineException	An error occurred with PrintEngine.

#### Example

The following example demonstrates calling the SetPrintEngineCount method from a web application's Application\_Start event. (Please review *Conventions* for notes on the use of the NicePrintService object.)

```

[Visual Basic]
Sub Application_Start(ByVal sender As Object, ByVal e As EventArgs)
    ' Create new instance of NPSService on application startup
    NicePrintService = New PrintService.NPSService
    ' Set number of PrintEngines to 10
    NicePrintService.SetPrintEngineCount(10)
End Sub

[C#]
void Application_Start(object sender, EventArgs e)
{
    // Create new instance of NPSService on application startup
    NicePrintService = new PrintService.NPSService();
    // Set number of PrintEngines to 10
    NicePrintService.SetPrintEngineCount(10);
}

```

#### 4.2.1.1.7 TestEngineConnection Method

This method can be called to test the connection between PrintServer web service and PrintEngine

#### Member of

PrintService.NPSService

#### Signature

[Visual Basic]  
Public Function **TestEngineConnection()** as String

[C#]  
public string **TestEngineConnection()**

#### Parameters

Parameter	Data Type	Description
none	N/A	N/A

#### Return Value

Data Type	Description
System.String	Return "0" if the test is OK, "-1" if the Engine's TestConnection method fails, or a message of the exception that might be thrown during the test.

#### Exceptions

Exception Types	Condition
-----------------	-----------

PrintEngineException	Failed connecting with PrintEngine.
----------------------	-------------------------------------

#### 4.2.1.1.8 UpdatePreviewFile Method

This method creates a label preview image for the label, identified by the LabelFile parameter.

The name of the preview image will have the TargetPath prepended. This is used to allow caller application to identify the name of the preview image uniquely to avoid conflicts when multiple users are using the web service.

The Variable data is provided in the parameter dsVariables, which is using the same structure as the return value of the GetVariables method.

PreviewWidth and PreviewHeight define the size of the preview image generated.

LabelKind defines the part of the label design which needs to be shown in the generated image.

LabelKind values	Meaning
1	Main label
2	Header label
3	Tail label

LabelSide defines which side of the label to show when double side printing is used.

LabelSide values	Meaning
0	Front Side
1	Back Side

When HasBorder parameter is set to true, the preview image will include a border around the label. If false, border is not generated.

#### Member of

PrintService.NPSService

#### Signature

[Visual Basic]  
 Public Function **UpdatePreviewFile**(ByVal *Printer* As String, \_  
     ByVal *LabelFile* As PrintService.FileLocation, \_  
     ByVal *TargetPath* As String, \_  
     ByVal *dsVariables* As System.Data.DataSet, \_  
     ByVal *PreviewWidth* As System.Int32, \_  
     ByVal *PreviewHeight* As System.Int32, \_  
     ByVal *LabelKind* As System.Int32, \_  
     ByVal *LabelSide* As System.Int32, \_  
     ByVal *HasBorder* As Boolean) As String

[C#]  
 public string **UpdatePreviewFile**(string *Printer*,  
     PrintServer.FileLocation *LabelFile*,

```
string TargetPath,
System.Data.DataSet dsVariables,
System.Int32 PreviewWidth,
System.Int32 PreviewHeight,
System.Int32 LabelKind,
System.Int32 LabelSide,
bool HasBorder)
```

## Parameters

Parameter Name	Data Type	Description
<i>Printer</i>	System.String	Name of Printer.
<i>LabelFile</i>	PrintService.FileLocation	Holds label's relative path and file name.
<i>TargetPath</i>	System.String	Unique value added to file name.
<i>dsVariables</i>	System.Data.DataSet	Dataset that holds a table of label variables.
<i>PreviewWidth</i>	System.Int32	Width of generated preview.
<i>PreviewHeight</i>	System.Int32	Height of generated preview.
<i>LabelKind</i>	System.Int32	Main, Header, or Tail Label
<i>LabelSide</i>	System.Int32	Front or Back
<i>HasBorder</i>	System.Boolean	Determines if a border is added around preview image.

## Return Value

Data Type	Description
System.String	Relative path and name of generated print stream file

The return value of the method is relative path and name of the generated preview image. The calling application must use the web Service base address (for example "http://machinename/NicePrintServer/") as the prefix and add the return value. This will point to the location of generated preview image.

## Exceptions

Exception Types	Condition
PrintServerException	Provided label file location is not valid.
PrintEngineException	Engine can not be started.
	Label file can not be opened by the engine.
	Label preview can not be generated.
	Maximum count of engine's is already opened and in use. Server is busy.

## Example

The following example shows a user function which calls UpdatePreviewFile to generate an image file on the server and then loads the generated image file into an Image control. **Note:** This Example assumes the existence of an Image control named *imgPreview*.

[Visual Basic]

```
Public Sub DisplayPreviewImage(ByVal PrinterName As String, _
    ByVal LabelFile As FileLocation, _
    ByVal SessionName As String, _
    ByVal LabelVars As Dataset) As String

    Dim PreviewFileName As String
    PreviewFileName = NicePrintService.UpdatePreviewFile(PrinterName, LabelFile, _
        SessionName, LabelVars, 400, 600, 1, 0, True)

    imgPreview.ImageUrl = "\\SampleSession\" & PreviewFileName
    imgPreview.AlternateText = PreviewFileName
End Sub
```

[C#]

```
public string DisplayPreviewImage(string PrinterName, PrintService.FileLocation LabelFile,
    string SessionName, System.Data.DataSet LabelVars)
{
    string PreviewFileName;
    PreviewFileName = NicePrintService.UpdatePreviewFile(PrinterName, LabelFile,
        SessionName, LabelVars, 400, 600, 1, 0, true);

    imgPreview.ImageUrl = "\\SampleSession\" + PreviewFileName;
    imgPreview.AlternateText = PreviewFileName;
}
```

## 4.2.2 FileLocation Class

FileLocation is a custom object that manages the current label's LabelGroup and LabelFileName properties.

### Web Reference

NPSService.asmx

### Signature

[Visual Basic]

```
Public Class FileLocation
    Inherits System.Object
```

[C#]

```
public class FileLocation : System.Object
```

### Example

The following example demonstrates instantiating a new object based on the NPSService class.

[Visual Basic]

```
Public LabelFile As PrintService.FileLocation
Sub Application_Start(ByVal sender As Object, ByVal e As EventArgs)
    ' Create new instance of FileLocation on application startup
    LabelFile = New PrintService.FileLocation
End Sub
```

[C#]

```

PrintService.FileLocation LabelFile;
void Application_Start(object sender, EventArgs e)
{
    // Create new instance of FileLocation on application startup
    LabelFile = new PrintService.FileLocation();
}

```

#### 4.2.2.1 FileLocation Members

##### Public Constructors

Constructors	Description
FileLocation Constructor	Initializes a new instance of the <b>FileLocation</b> class.

[Visual Basic]  
Public Sub **New**()

[C#]  
public **FileLocation**()

##### Public Properties

Property	Description
LabelFileName	Gets or sets the current label file.
LabelGroup	Gets or sets the current label directory.

##### 4.2.2.1.1 LabelFileName Property

LabelFileName sets or returns the name of the current label file. This value does not include the file path (see LabelGroup property for handling that value).

##### Member of

PrintService.FileLocation

##### Signature

[Visual Basic]  
Public Property **LabelFileName**() As String

[C#]  
public string **LabelFileName** { set; get; }

##### Parameters

Parameter	Data Type	Description
none	N/A	N/A

##### Property Value

Data Type	Description
System.String	The name of the current label file (no path prefix).

## Exceptions

Exception Type	Condition
PrintServerException	Failed getting/setting label file name.

## Example

The following example demonstrates setting the label file name. (Please review *Conventions* for notes on the use of the LabelFile object.)

```
[Visual Basic]
LabelFile.LabelFileName = "Sample.lbl"
```

```
[C#]
LabelFile.LabelFileName = "Sample.lbl";
```

### 4.2.2.1.2 LabelGroup Property

LabelGroup sets or returns the path of the current label file. This value does not include the file name (see LabelGroup property for handling that value).

## Member of

PrintService.FileLocation

## Signature

```
[Visual Basic]
Public Property LabelGroup() As String
```

```
[C#]
public string LabelGroup { set; get; }
```

## Parameters

Parameter	Data Type	Description
none	N/A	N/A

## Property Value

Data Type	Description
System.String	The folder path of the current label.

## Exceptions

Exception Types	Condition
PrintServerException	Failed getting/setting label group.

## Example

The following example demonstrates setting the label file path and name. (Please review *Conventions* for notes on the use of the NicePrintService object.)

```
[Visual Basic]
LabelFile.LabelGroup = "LabelFiles\SampleLabels"
```

```
[C#]
LabelFile.LabelGroup = "LabelFiles\\SampleLabels";
```

## 4.3 StorageServer Web Service

The **StorageServer** Web Service contains classes that provide functionality to retrieve information about label files stored on the server. The labels can be organized in hierarchical structure (folders).

The service provides methods for:

- Retrieve the list of label workspaces (folders)
- Retrieve the list of labels for selected workspace

### Classes

Class	Description
StorageService	Maintains organization of Labels and LabelGroups

### 4.3.1 StorageService Class

Labels are grouped in a hierarchical structure on the server. The structure is the same as the file system folder structure. If no label grouping is used, the parameter can be empty. If labels are grouped into multiple groups, the parameter can define groups and subgroups.

Example of label storage structure:

- Group1
  - SubGroup1
  - SubGroup2
- Group2
  - SubGroup1
  - SubGroup2

An empty value can be passed for the LabelGroup parameter when label files do not belong to a group. When passing value "Group1" for LabelGroup, label file names in Group1 are returned. If passing "Group2/SubGroup4", label file names in SubGroup4 of Group2 will be returned.

## Web Reference

StorageService.asmx

## Signature

```
[Visual Basic]
Public Class StorageService
    Inherits System.Web.Services.Protocols.SoapHttpClientProtocol
```

```
[C#]
public class StorageService : System.Web.Services.Protocols.SoapHttpClientProtocol
```

## Example

The following example demonstrates instantiating a new object based on the StorageService class.

```
[Visual Basic]
Public LabelStore As NiceStorageService.StorageService
Sub Application_Start(ByVal sender As Object, ByVal e As EventArgs)
    ' Create new instance of NiceStorageService on application startup
    LabelStore = New NiceStorageService.StorageService
End Sub
```

```
[C#]
NiceStorageService.StorageService LabelStore;
void Application_Start(object sender, EventArgs e)
{
    // Create new instance of NiceStorageService on application
    // startup
    LabelStore = new NiceStorageService.StorageService ();
}
```

### 4.3.1.1 StorageService Members

#### Public Constructors

Constructors	Description
StorageService Constructor	Initializes a new instance of the <b>StorageService</b> class.

```
[Visual Basic]
Public Sub New()
```

```
[C#]
public StorageService()
```

## Public Methods

Method	Description
GetLabelGroups	Returns a list of workgroups (structured folders).
GetLabelList	Returns a list of Labels in a specific workgroup.

### 4.3.1.1.1 GetLabelGroups Method

This method returns an array of relative paths to all workgroups. Each workgroup can contain a collection of

The following list exemplifies a collection of possible return values:

- Shipping
- Shipping\Retailers
- Shipping\Retailers\North America
- Shipping\Retailers\Europe
- Shipping\Distribution Centers
- Shipping\Distribution Centers\North America
- Shipping\Distribution Centers\Europe
- Shipping\Others
- Products
- Products\Product ABC
- Product s\Product XYZ

### Member of

NiceStorageService.StorageService

### Signature

[Visual Basic]  
Public Function **GetLabelGroups()** As **String()**

[C#]  
public string[] **GetLabelGroups()**

### Parameters

Parameter	Data Type	Description
none	N/A	N/A

### Return Value

Data Type	Description
System.String	String array of relative paths of workspaces.

### Exceptions

Exception Types	Condition
StorageServerException	Failed to retrieve workspaces.

## Example

The following example demonstrates retrieving the collection of workspaces.

```
[Visual Basic]
Dim NiceLabelGroups as String()
NiceLabelGroups = NiceStorageService.GetLabelGroups
```

```
[C#]
String[] NiceLabelGroups;
NiceLabelGroups = NiceStorageService.GetLabelGroups();
```

### 4.3.1.1.2 GetLabelList Method

The GetLabelList method returns an array of label files in the current workgroup.

## Member of

NiceStorageService.StorageService

## Signature

```
[Visual Basic]
Public Function GetLabelList(ByVal labelGroup As String, ByVal browseSubGroups As Boolean) As String()
```

```
[C#]
public string[] GetLabelList(string labelGroup, bool browseSubGroups)
```

## Parameters

Parameter	Data Type	Description
labelGroup	String/string	The name of the workspace to retrieve label list from.
browseSubGroup	Boolean/bool	True/False value determining whether or not labels from sub-groups should be returned.

## Return Value

Data Type	Description
System.String	String array of label file names in specified of workspace.

## Exceptions

Exception Types	Condition
StorageServerException	Failed retrieving label list.

## Example

The following example demonstrates retrieving the label files from a specified workspace.

```
[Visual Basic]
Dim NiceLabelLists as String()
NiceLabelList = NiceStorageService.GetLabelList("Shipping\Retailers\North America", False)
```

```
[C#]
String[] NiceLabelList;
NiceLabelList = NiceStorageService.GetLabelList("Shipping\\Retailers\\North America", false);
```

## 4.4 NLWAX Web Browser ActiveX and Java Control

NiceLabel WebSDK ActiveX and Java component (NLWAX) is used for client side printing. This component is automatically installed by the web browser, and is a required part of the web based printing application.

NLWAX handles transferring the print files generated by the NicePrintServer web services to the local printers. It can be incorporated into a visual component on the page, or used as a runtime component called from the web browser.

When the page is to be loaded in the browser, the local computer will be checked if NLWAX is already installed. If it is not present, or locally installed version is older than the one required by the page, the component will be installed.

NLWAX Component is signed by Euro Plus digital signature; user must set their browser security settings to allow installation of signed ActiveX and Java components.

The **NLWAX** Web Browser ActiveX and Java Control provides accessibility to client-side resources. This control communicates directly with the NicePrintServer web service and the local printer drivers and ports.

This control provides the following functionality:

- Sends print stream to client's printer ports
- Retrieves list of client's installed printers
- Retrieves list of printer ports associated with the client's installed printers.

## Classes

Class	Description
NLWAXForm	Required to gain access to client-side printers and ports.

## 4.4.1 NLWAXForm Class

NLWAXForm class gains access to client-side resources to enable printing on local printers from the server.

Since all methods of NLWAXForm need access to client resources, they have to be called from client-side scripting. All code examples for NLWAXForm Class will demonstrate using client-side JavaScript.

### Reference

NLWAX Library

### Example Markup

```
<object classid="clsid:ECBB0B73-FDF4-4DBD-B7C9-8450C2D9FB66"
  codebase="/NLWAX.ocx#version=1,0,0,34"
  width="1"
  height="1"
  id="NLWAXForm"
  name="NLWAXForm">
</object>
```

#### 4.4.1.1 NLWAXForm Members

##### Public Methods

Method	Description
GetPortList	Retrieves a list of ports on client PC.
GetPrinterList	Retrieves a list of printers on client PC.
SendToPort	Sends print stream to local port.
SendToPrinter	Sends print stream to local printer.

##### Public Properties

Property	Description
BaseURL	Contains base URL of NicePrintServer.

#### 4.4.1.1.1 GetPortList Method

Returns an XML structured string containing an unordered list of all available ports on client PC. The following demonstrates the structure of the return value:

```
<Ports>
  <Port>
    <PortName>COM1:</ PortName >
  </Port>
  <Portr>
    <PortName>COM2:</ PortName >
  </Port>
  ...
</Ports>
```

**Member of**

NLWAX. NLWAXFormClass

**Signature**

[Visual Basic]  
Function GetPortList() As String

[C#]  
public virtual new [System.String](#) GetPortList ( )

**Parameters**

Parameter	Data Type	Description
none	N/A	N/A

**Return Value**

Data Type	Description
System.String	XML structured string containing port names

**Example**

This example demonstrates calling the GetPortList method.

[JavaScript]  
var WAX = document.getElementById('NLWAXForm');  
var portList = WAX.GetPortList();

**4.4.1.1.2 GetPrinterList Method**

Returns an XML structured string containing an unordered list of all installed printers on the client PC along with its associated port. The following demonstrates the structure of the return value:

```
<Printers>
  <Printer>
    <DeviceName>Printer1Name</DeviceName>
    <Port>LPT1:</Port>
  </Printer>
  <Printer>
    <DeviceName>Printer2Name</Printer>
    <Port>LPT2:</Port>
  </Printer>
  ...
</Printers>
```

**Member of**

NLWAX. NLWAXFormClass

**Signature**

[Visual Basic]  
Function GetPrinterList() As String

```
[C#]
public virtual new System.String GetPrinterList ( )
```

## Parameters

Parameter	Data Type	Description
None	N/A	N/A

## Return Value

Data Type	Description
System.String	XML structured string containing print names and associated ports.

## Example

The following example demonstrates retrieving the label files from a specified workspace.

```
[JavaScript]
var WAX = document.getElementById('NLWAXForm');
var printerList = WAX.GetPrinterList();
```

### 4.4.1.1.3 SendToPort Method

SendToPort method is used to send the print stream file specified by the PrintFile parameter to the Port, identified by the PortName parameter.

PrintFile is the relative file to the web service base URL, which must be set by the BaseURL property.

The NLWAX component will download the print stream file and send it to the port.

When IPPort number is greater than 0, the PortName represents a TCP/IP address. In this case TCP/IP connection to the PortName:IPPort is made and print stream is sent to this destination.

When NLWAX is sending files to specific port, it first checks the installed printers, if there is any printer already connected to the port. If so, it is using that driver in passthrough mode to transfer file to the printer (ports are locked, when a printer driver is using it).

In the printer is installed, the LabelName parameter is used to identify the print job in the print spooler.

If there is no printer used on the specified port, NLWAX will send the print file directly to the port.

## Member of

NLWAX. NLWAXFormClass

## Signature

```
[Visual Basic]
Sub SendToPort(LabelName As String, PrintFile As String, PortName As String, IPPort As Long)
```

```
[C#]
public virtual new void SendToPort ( System.String LabelName , System.String PrintFile ,
System.String PortName , System.Int32 IPPort )
```

## Parameters

Parameter	Data Type	Description
LabelName	String/string	File name of label to print.
PrintFile	String/string	Print stream file name previously generated by the NicePrintService.CreatePrintFile method
PortName	String/string	Port name as it appears on client PC.
IPort	Long/long	Port number.

## Return Value

Data Type	Description
None/void	N/A

## Example

The following example demonstrates sending a print file to the LPT1 port.

```
[JavaScript]
var WAX = document.getElementById('NLWAXForm');
WAX.SendToPort('Test.lbl', 'printouts/123Test.prn', 'LPT1:', 0);
```

### 4.4.1.1.4 SendToPrinter Method

The SendToPrinter method is used to send the print stream file, specified by the PrintFile parameter to the printer, identified by the PrinterName parameter.

PrintFile is the relative file from the web service base URL, which must be set by the BaseURL property.

Component will download the print stream file, and send it to the printer in passthrough mode (directly to the spooler).

The LabelName parameter will be used to identify the print job in the print spooler.

## Member of

NLWAX.NLWAXFormClass

## Signature

```
[Visual Basic]
Sub SendToPrinter(LabelName As String, PrintFile As String, PrinterName As String)
```

```
[C#]
public virtual new void SendToPrinter ( System.String LabelName , System.String PrintFile ,
System.String PrinterName )
```

## Parameters

Parameter	Data Type	Description
LabelName	String/string	File name of label to print.
PrintFile	String/string	Print stream file name previously generated by the NicePrintService.CreatePrintFile method
PrinterName	String/string	Printer name as it appears on client PC.

## Return Value

Data Type	Description
None/void	N/A

## Example

The following example demonstrates retrieving the label files from a specified workspace.

```
[JavaScript]
var WAX = document.getElementById("NLWAXForm");
WAX.SendToPrinter('Test.lib', 'printouts/123Test.prn', 'ZEBRA LP 2844-Z');
```

### 4.4.1.1.5 BaseURL Property

This property is used to set the base URL for NPSService. NLWAX is using this property later to properly download the print files and to verify the connection with NPSService correctly.

## Member of

NLWAX. NLWAXFormClass

## Signature

[Visual Basic]  
Property BaseURL As String

[C#]  
public virtual new string **BaseURL** [ get, set ]

## Parameters

Parameter	Data Type	Description
none	N/A	N/A

## Property Value

Data Type	Description
System.String	Gets or sets the BaseURL value.

## Example

The following example demonstrates passing the base URL of your installation of NiceLabel WebSDK to NLWAX.

```
[JavaScript]
var WAX = document.getElementById('NLWAXForm');
WAX.BaseURL = 'http://WebSDKHost/NicePrintServer/';
```

### 4.4.1.1.6 Example of NLWAX Web Browser Control

```
<script type="text/javascript">
  if (BrowserDetect.browser == 'Explorer')
  {
    //<!-- NLWAX ActiveX control -->
    document.write(
      "<object classid=\"clsid:ECBB0B73-FDF4-4DBD-B7C9-8450C2D9FB66\" "
        + "codebase=\"./NLWAX.ocx#version=1,0,0,34\" "
        + "width=\"1\" "
        + "height=\"1\" "
        + "id=\"NLWAXForm\" "
        + "name=\"NLWAXForm\">"
      + "</object>");
  }else{
    document.write(
      "<applet name=\"NLWAXForm\" "
        + "code=\"nlwax.UserInterface\" "
        + "WIDTH=0 HEIGHT=0 "
        + "ARCHIVE=\"./nlwax.jar\">"
      + "</applet>");
  }
</script>
```

# 5 NiceLabel WebSDK Sample Application

---

NiceLabel WebSDK provides a sample application which demonstrates how to perform various common tasks. This sample includes solution, project files and complete source code. By default, this application is installed in the <Program Files>\Euro Plus\NiceLabel WebSDK\SampleApplication\ folder. It will also be installed as a web application under the Default Web Site in IIS. You can access it using this link: <http://localhost/NPSSampleApplication/>.

## 5.1 NPSSampleApplication

A Visual Studio 2005 web application that demonstrates using web service calls, ASP.NET server controls, HTML elements, and client-side JavaScript to achieve client-side printing.

### 5.1.1 SourceCode

Contains complete source code for the WebSDKSampleApp solution.

## 5.2 Explanation of Sample Application

### 5.2.1 Integration of NLWAX.ocx

The NLWAX ActiveX and Java Control is implemented in the NLWAX.ascx Web User Control. It is not required to incorporate NLWAX.ocx in this way, but is implemented as such based on the design consideration of compartmentalizing the needed controls, server-side methods, and client-side methods in one object. Whether NLWAX.ocx is implemented as part of a User Control or is used directly on the web page the following HTML Object tag is required:

```
<object classid="clsid:ECBB0B73-FDF4-4DBD-B7C9-8450C2D9FB66"  
  codebase="./NLWAX.ocx#version=1,0,0,34"  
  width="1"  
  height="1"  
  id="NLWAXForm"  
  name="NLWAXForm">  
</object>
```

The classid and codebase properties must maintain the values as shown above. The width, height, id, and name properties can be modified as needed.

#### 5.2.1.1 Usage of Controls in NLWAX.ascx Web User Control

<asp:HiddenField> - A number of ASP.Net HiddenField controls are utilized to pass user data between server-side and client-side code.

<Object> – As explained in the Integration of NLWAX.ocx section, this control is a required component that handles client-side printing.

<Script> – This control includes the NLWAXfunctions.js external JavaScript file.

## Explanation of NLWAXFunctions.js

NLWAXFunctions.js is an external JavaScript file that contains client-side scripts which call NLWAX.ocx to retrieve client information (e.g. lists of installed printers and available ports on the client's machine) as well as passing the print stream to the client's printer.

The main method performs four functions:

**LoadPrinterList** – This calls NLWAX.ocx's GetPrinterList method. The return value is then stored in the hdfPrinterList HiddenField making it accessible from server-side code.

**LoadPortList** - This calls NLWAX.ocx's GetPortList method. The return value is then stored in the hdfPortList HiddenField making it accessible from server-side code.

**Sets NLWAX.ocx's BaseUrl property** – This must be set for the component to successfully verify web service connection and get the full URL name to the print file. The server-side code retrieves this value from the Web.Config file and populates hdfBaseUrl. This script then takes that value and passes it to NLWAX.ocx.

**Check if printing has been requested** – When the value of hdfIsPrinting is 'true'

When the user clicks on the 'Print' button (<asp:button id='btnPrint'> in default.aspx) a series of events occur:

- The btnPrint.Click event passes the print quantity to the public Print function in NLWAX.ascx.
- NLWAX.ascx calls the CreatePrintFile method of NLWAX.ocx passing the appropriate .
- NLWAX.ocx generates the print stream, writes it to a temporary file on the server, and returns the location and name of this file.
- NLWAX.ascx stores the returned file name in a HiddenField and sets the value of hdfIsPrinting to 'true'
- At this point the JavaScript recognizes the hdfIsPrinting is true and calls either the SendToPrinter or SendToPort method of NLWAX.ocx.
- SendToPrinter utilizes the installed driver on the client PC and sends the data to the associated port.
- SendToPort sends the print data to the port.

## 5.2.2 Server-Side Code Explanation

### 5.2.2.1 Default.aspx.vb

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
    If Not Page.IsPostBack Then
        NLWAX1.BaseURL = ConfigurationManager.AppSettings("PrintService.BaseURL")
        Passes the BaseURL value from web.config to NLWAX.ascx. This value is used by
        NLWAX.ocx for retrieving preview image and printer code files from the server.

        NLWAX1.SessionName = Now.Ticks.ToString
        Generates a unique SessionName and passes that to NLWAX.ascx. This value is
        used to differentiate between multiple concurrent sessions.

        PopulateLabelTree(NLWAX1.LabelGroups)
```

Populates a `TreeView` control with label groups and labels retrieved from the `StorageService` web service.

```
PopulateDriversTree(NLWAX1.Printers)
```

Populates a `TreeView` control with a list of Drivers available on the server. This list is retrieved from the `NicePrintService` web service.

```
End If
End Sub
```

```
Protected Sub btnPrint_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    NLWAX1.Print(CInt(txtQuant.Text))
```

Initiates printing sequence. Please see *Explanation of NLWAXFunctions.js* above for a detailed explanation of this process.

```
    btnReprint.Visible = True
```

Enables the option of reprinting.

```
End Sub
```

```
Protected Sub btnReprint_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    NLWAX1.Print(txtQuant.Text, True)
```

Initiates printing sequence flagging that this is a reprint request. A reprint simply resends a previously generated print stream to the printer.

```
End Sub
```

```
Protected Sub trvLabels_SelectedNodeChanged(ByVal sender As Object, _
    ByVal e As System.EventArgs)
```

```
    ...
    NLWAX1.LabelGroup = FolderPath
    NLWAX1.Label = trvLabels.SelectedNode.Text
```

These two statements persist the selected label name and location in session variables for later use.

```
    ...
End Sub
```

```
Protected Sub grvVars_RowUpdating(ByVal sender As Object, _
    ByVal e As System.Web.UI.WebControls.GridViewUpdateEventArgs)
    Dim ds As Data.DataSet = NLWAX1.Variables
    ds.Tables(0).Rows(e.RowIndex).Item(1) = _ CType(grvVars.Rows(e.RowIndex).FindControl("txtEditVal"),
        TextBox).Text
    ds.AcceptChanges()
    NLWAX1.Variables = ds
```

This routine retrieves the variables (as a dataset) from the label, updates the changes, and returns the dataset to update the label.

```
    ...
End Sub
```

```
Private Sub GetPreview()
    Dim url As String = ""
```

```
    ...
    url = NLWAX1.GetPreviewUrl(imgPreview.Width.Value, _
        imgPreview.Height.Value, NLWAX.LabelKind.MainLabel, NLWAX.LabelSide.Front)
```

This calls `NicePrintService's GetPreviewUrl` which generates an image of the label and returns the URL of the file.

```
    Dim errorMsg As String = url.Substring(url.LastIndexOf("/") + 1)
```

If the web service encounters an error generating the image it will return an error description in place of the file name. The error message will always start with an asterisk (\*).

```
    If errorMsg.StartsWith("**") Then
        imgPreview.AlternateText = errorMsg
        lblLabel.Text = errorMsg
```

```

        Displays error message on page.
    Else
        imgPreview.ImageUrl = url
        imgPreview.AlternateText = NLWAX1.Label
        lblLabel.Text = NLWAX1.Label
        Updates image control and label on web page to reflect new image.
    End If
    ...
End Sub

```

### 5.2.2.2 NLWAX.ascx.vb

```

#Region "Public Enums"
    Public Enum PrintType
    ...
    Public Enum LabelKind
    ...
    Public Enum LabelSide
    ...

```

These are convenience Enums used as parameters by some of the methods in this class.

```

#Region "Public Properties"
    Public ReadOnly Property IsConnected() As String
        Get
            Return NicePrintService.TestEngineConnection
            Returns the result of NicePrintService testing its connection with the Print
            Engine.
        End Get
    End Property

    Public Property BaseURL() As String
        Get
            Return Session("BaseURL")
            Gets or the BaseURL from a session variable
        End Get
        Set(ByVal value As String)
            hdfBaseURL.Value = value
            Session("BaseURL") = value
            Saves the BaseURL in a HiddenField for later use by client-side script and holds
            the value in a session variable.
        End Set
    End Property

    Public Property SessionName() As String
        Get
            Return Session("SessionName")
            Gets the SessionName held in a session variable
        End Get
        Set(ByVal value As String)
            Session("SessionName") = value
            Holds the SessionName in a session variable
        End Set
    End Property

    Public ReadOnly Property LabelGroups() As Object
        Get
            Return NPSLabelStorage.GetLabelGroups
            Returns the collection of label groups retrieved from the LabelStorage web
            service
        End Get
    End Property

    Public Property LabelGroup() As String
        Get
            Return Session("LabelGroup")
            Gets the current Label Group from a session variable
        End Get
        Set(ByVal value As String)
            Session("LabelGroup") = value

```

```

        Holds the current Label Group in a session variable
    End Set
End Property

Public ReadOnly Property Labels(ByVal LabelGroup As String, _
    ByVal Recursive As Boolean)
    Get
        Return NPSLabelStorage.GetLabelList(LabelGroup, Recursive)
        Returns a list of labels contained in the current LabelGroup from LabelStorage
    End Get
End Property

Public Property Label() As String
    Get
        Return hdfLabel.Value
        Returns name of current label held in a HiddenField
    End Get
    Set(ByVal value As String)
        hdfLabel.Value = value
        Saves the value in a HiddenField
        Session("dsVars") = Nothing
        Session("PreviewUrl") = Nothing
        Clear out any previous label data
    End Set
End Property

Private ReadOnly Property FileLocation() As PrintService.FileLocation
    Get
        __FileLocation.LabelFileName = Label
        __FileLocation.LabelGroup = LabelGroup
        Return __FileLocation
        Returns current label and location as a FileLocation object
    End Get
End Property

Public ReadOnly Property Printers() As String()
    Get
        Dim obj() As String
        obj = NicePrintService.GetPrinterList
        Array.Sort(obj)
        Return obj
        Retrieves a collection of printers from NicePrintService web service.
    End Get
End Property

Public Property Printer() As String
    Private Get
        Return hdfPrinter.Value
        Retrieves the name of the current printer from a HiddenField
    End Get
    Set(ByVal value As String)
        hdfPrinter.Value = value
        Stores the name of the current printer in a HiddenField
    End Set
End Property

Public Property Variables() As DataSet
    Get
        If Session("dsVars") Is Nothing Then
            Session("dsVars") = NicePrintService.GetVariables(FileLocation)
        End If
        Return Session("dsVars")
        Returns dataset of variables. If it is not currently stored in a session variable the
        web service is called to retrieve the dataset.
    End Get
    Set(ByVal value As DataSet)
        Session("dsVars") = value
        Session("PreviewUrl") = Nothing
    End Set
End Property

```

```

        Stores current variables dataset in a session variable and clears out any previous
        preview image.
    End Set
End Property
#End Region

#Region "Public Functions"
Public Function GetPreviewUrl(ByVal Width As Int32, ByVal Height As Int32, _
    ByVal Kind As LabelKind, ByVal Side As LabelSide) As String
    If Session("PreviewUrl") Is Nothing Then
        Dim s As String
        s = BaseURL & NicePrintService.UpdatePreviewFile(Printer, FileLocation, _
            SessionName, Variables, Width, Height, Kind, Side, True)
        Session("PreviewUrl") = s
    End If
    Return Session("PreviewUrl")
    Returns URL of preview image. If it is not currently stored in a session variable the
    web service is called to generate a new image and return the URL.
End Function

Public Function GetVarFormatName(ByVal ID As Long) As String
    Select Case ID
        Case 0
            Return "All"
        Case 1
            Return "Numeric"
        ...
    End Select
    This function returns the variable's format name based on provided ID
End Function

Public Function Print(ByVal Quantity As Int32, _
    Optional ByVal IsReprint As Boolean = False) As Boolean
    If Not IsReprint Then
        hdfFile.Value = CreatePrintFile(Quantity)
        If this is not a reprint the CreatePrintFile method is called to generate a new print
        steam.
    End If
    hdfIsPrinting.Value = "true"
    The hdfIsPrinting HiddenField is set to true. This will trigger the JavaScript to send
    the print stream to the printer.
    Return True
End Function

Private Function CreatePrintFile(ByVal Quantity As String) As String
    Dim sFile As String
    sFile = NicePrintService.CreatePrintFile(Printer, FileLocation, Quantity, SessionName, Variables)
    Return sFile
    Calls NicePrintService's CreatePrintFile method which generates a new print stream
    and returns its location.
End Function
#End Region

```

### 5.2.3 Additional JavaScript files

Global.js, printer.js, and port.js contain a variety of client-side methods. These are largely used for manipulating the user interface and storing user specific data in HiddenFields for later use.

### 5.2.4 Web.Config Settings

The web.config file contains three custom keys whose values are used by the web application. These keys are located under the <configuration><appSettings> section.

```

<add key="PrintService.BaseURL"
    value="http://localhost/NicePrintServer/">

```

The base URL of the NicePrintServer application. This value is used by NLWAX.ocx to communicate back to the NicePrintServer web service. *localhost* must be replaced with the web site name or IP address of the server hosting NicePrintServer.

```
<add key="StorageService.StorageService"  
  value="http://localhost/NicePrintServer/StorageServer/StorageService.asmx"/>  
<add key="PrintService.NPSService"  
  value="http://localhost/NicePrintServer/PrintServer/NPSService.asmx"/>
```

These two values are used by the web application to locate the NicePrintServer web services. If the web application is hosted on the same server as the webservice then *localhost* or server name can be used. If these are hosted on different servers then *localhost* must be replaced with the web site name or IP address of the server hosting NicePrintServer.

## 6 Contacts

---

### Head Office

**Euro Plus d.o.o.**

Ulica Lojzeta Hrovata 4c  
SI-4000 Kranj, Slovenia  
Tel: +386 4 280 50 00  
Fax: +386 4 233 11 48  
[www.europlus.si](http://www.europlus.si)  
[info@europlus.si](mailto:info@europlus.si)  
[sales@europlus.si](mailto:sales@europlus.si)  
[support@europlus.si](mailto:support@europlus.si)

### French Office

NiceLabel France  
CobarSoft SARL  
6 rue Salengro  
ZI hippodrome  
32 000 AUCH  
France  
Tel: +33 (0) 562 709 201  
Fax: +33 (0) 562 708 004  
[support@NiceLabel.fr](mailto:support@NiceLabel.fr)  
[www.NiceLabel.fr](http://www.NiceLabel.fr)

### North American Office

**Niceware International, LLC**

10437 Innovation Drive, Suite 147  
Milwaukee, WI 53226  
Tel: +1 414 476 6423  
Fax: +1 414 476 7955  
[www.nicewareintl.com](http://www.nicewareintl.com)  
[info@nicewareintl.com](mailto:info@nicewareintl.com)  
[sales@nicewareintl.com](mailto:sales@nicewareintl.com)  
[support@nicewareintl.com](mailto:support@nicewareintl.com)

### German Office

NiceLabel Germany GmbH  
Liebknechtstr. 29  
63179 Obertshausen  
Germany  
+49 (0)6104 405 400 Tel  
+49 (0)6104 405 4020 Fax  
  
[info@nicelabel.de](mailto:info@nicelabel.de)  
[www.nicelabel.de](http://www.nicelabel.de)