

Barcodesoft Webservice for

Bar Code Generating

User's Manual

Table of Contents

1.	Overview.....	1
1.1.	XML and HTTP	1
1.2.	SOAP	1
1.3.	WSDL	1
1.4.	XML Schema	2
1.5.	Trial Version Limitation	2
2.	Using Barcodesoft Webservice	3
2.1	Code39	4
2.2	Code39Ext	5
2.3	Codabar.....	6
2.4	I25	7
2.5	Code93	8
2.6	Code128A	9
2.7	Code128B	10
2.8	Code128C	11
2.9	GS1128	12
2.10	Code25	13
2.11	Code11	14
2.12	Telepen	15
2.13	MSI	16
2.14	UPCA	17
2.15	UPCE	18
2.16	EAN13	19
2.17	EAN8	20
2.18	Bookland	21
2.19	Datamatrix	22
2.20	QRCode	23
2.21	PDF417	24
2.22	Aztec	26
3.	XML Schema	
3.1	BcsImageFormat	27
3.2	BcsOrientation	27
3.3	DataMatrixFormat	27
3.4	QrcodeErrorLvl	27
3.5	QrcodeFormat	28
3.6	AztecTargetSize	28
3.7	AztecErrorLevel	28
4.	Generate Barcode Using Webservice	29

4.1 Generate BarCode in PHP	29
4.2 Generate Bar Code in C#	32
5. End User License Agreement.....	33
Glossary.....	35
Index.....	36

Chapter 1. Overview

Webservice is a software system designed to support interoperable machine-to-machine interaction over the internet. The programmatic interfaces made available over the web for application-to-application communication are often referred to as web services. There are many types of applications that can be considered web services but interoperability between applications is enhanced most by the use of familiar technologies such as XML and HTTP. These technologies allow applications using differing languages and platforms to interface in a familiar way.

1.1 XML and HTTP

XML and HTTP are the foundation for calling web services. A user interacts with the web service interfaces by sending XML messages over HTTP. XML and HTTP are useful for creating and sending messages because they are flexible and widely supported on many platforms and languages. This interoperability allows applications to communicate using differing languages and platforms using these common technologies.

1.2 SOAP

Because XML and HTTP are so flexible, a web service message may be represented and communicated in many different ways. Therefore it is important to have standards that are common among all platforms and languages that use a web service. Simple Object Access Protocol (SOAP) is one standard that formally sets the conventions governing the format and processing rules of XML messages and how they are used with HTTP.

1.3 WSDL

A useful technology that is associated with web services is Web Service Description Language (WSDL). WSDL formally sets the format for describing a web service using XML. By providing a Uniform Resource

Identifier (URI) to a WSDL, applications can discover what operations a particular web service provides and how an operation's messages look.

Barcodesoft webservice is described at the following URL

<http://www.barcodesoft.com/bcdgen.asmx?WSDL>

1.4 XML Schema

Data types are a very important component of web services because different data types may be used to compose SOAP messages. WSDL uses XML Schema to describe data types used in a web service.

1.5 Trial Version Limitation

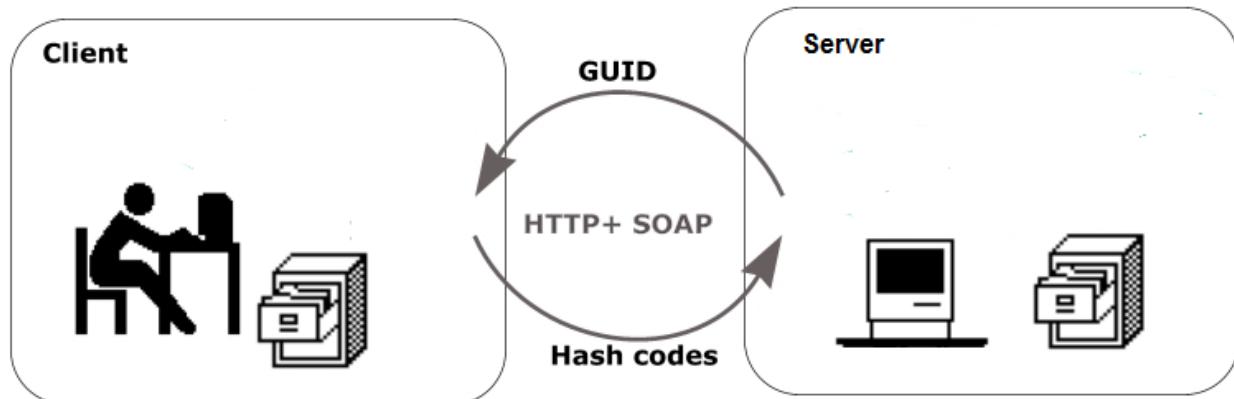
When you use Barcodesoft Webservice a license is required. You will receive a token string after getting a valid license. Without the token, a “Demo” watermark will show up in the barcode image.

Chapter 2 Using Barcodesoft Webservice

Barcodesoft webservice uses XML (eXtensible Markup Language) that follow Simple Object Access Protocol (SOAP) to stream bar code images from our web server to your computer. You can integrate published webmethods with your own client desktop application to generate barcode. You can also deploy barcode generating webservice on your own webserver.

Bar Code Generating Webservice supports most linear and 2D barcode symbologies: Code39, Code128, UPC-A, UPC-E, EAN13, EAN8, Bookland, ISBN, POSTNET, Code93, Code25, Interleaved 25, Codabar, MSI / Plessey, Data Matrix, PDF417, QR Code and Aztec Code.

Bar code image could be in one of the following formats: BMP, GIF, Jpeg, TIFF and PNG.



The usage of Barcodesoft Encoder is quite simple and straightforward. All buttons are self-intuitive.

There are 22 webmethods provided by Barcodesoft Webservice as listed below.

Code39, Code39Ext, Codabar, I25, Code93, Code128A, Code128B, Code128C, GS1128, Code25, Code11, Telepen, MSI, UPCA, UPCE, EAN13, EAN8, Bookland, Datamatrix, QRCode, PDF417 and Aztec.

2.1 Code39

Code39 accepts 9 parameters as listed below

```
<s:element minOccurs="0" maxOccurs="1" name="textToEncode" type="s:string"/>  
<s:element minOccurs="1" maxOccurs="1" name="ShowHumanReadable" type="s:boolean"/>  
<s:element minOccurs="1" maxOccurs="1" name="ShowCheckDigit" type="s:boolean"/>  
<s:element minOccurs="1" maxOccurs="1" name="ImageFormat" type="tns:BcsImageFormat"/>  
<s:element minOccurs="1" maxOccurs="1" name="Orientation" type="tns:BcsOrientation"/>  
<s:element minOccurs="1" maxOccurs="1" name="Resolution" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Width" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Height" type="s:int"/>  
<s:element minOccurs="0" maxOccurs="1" name="strToken" type="s:string"/>
```

The first parameter is **textToEncode**. This is your data to be encoded.

The second parameter is a Boolean. It tells the Webservice to show human readable text or not.

The third parameter is also a Boolean. It tells the Webservice to show check digit or not.

The fourth parameter is **ImageFormat**. It can be one of the following values: BMP, JPG, PNG, WMF, TIFF and GIF. It defines returned barcode image format.

The fifth parameter is **Orientation**. It is about returned barcode image orientation. It can be one of the following values: Original, Rotate90, Rotate180 and Rotate270.

The sixth parameter is **Resolution**. It sets returned barcode image resolution. Its value ranges between 96 and 600.

The seventh parameter is **Width**. It sets returned barcode image width in pixel. When set to zero, Webservice will set the shortest possible value for barcode image width.

The eighth parameter is **Height**. It sets returned barcode image height in pixel. When set to zero, Webservice will set a height for barcode image automatically.

The ninth parameter is **strToken**. Every licensed user will have a unique Token string for calling webmethods. If you leave it blank, returned barcode image will have a "Demo" watermark.

2.2 Code39Ext

Code39Ext accepts 9 parameters as listed below

```
<s:element minOccurs="0" maxOccurs="1" name="textToEncode" type="s:string"/>  
<s:element minOccurs="1" maxOccurs="1" name="ShowHumanReadable" type="s:boolean"/>  
<s:element minOccurs="1" maxOccurs="1" name="ShowCheckDigit" type="s:boolean"/>  
<s:element minOccurs="1" maxOccurs="1" name="ImageFormat" type="tns:BcsImageFormat"/>  
<s:element minOccurs="1" maxOccurs="1" name="Orientation" type="tns:BcsOrientation"/>  
<s:element minOccurs="1" maxOccurs="1" name="Resolution" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Width" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Height" type="s:int"/>  
<s:element minOccurs="0" maxOccurs="1" name="strToken" type="s:string"/>
```

The first parameter is **textToEncode**. This is your data to be encoded.

The second parameter is a Boolean. It tells the Webservice to show human readable text or not.

The third parameter is also a Boolean. It tells the Webservice to show check digit or not.

The fourth parameter is **ImageFormat**. It can be one of the following values: BMP, JPG, PNG, WMF, TIFF and GIF. It defines returned barcode image format.

The fifth parameter is **Orientation**. It is about returned barcode image orientation. It can be one of the following values: Original, Rotate90, Rotate180 and Rotate270.

The sixth parameter is **Resolution**. It sets returned barcode image resolution. Its value ranges between 96 and 600.

The seventh parameter is **Width**. It sets returned barcode image width in pixel. When set to zero, Webservice will set the shortest possible value for barcode image width.

The eighth parameter is **Height**. It sets returned barcode image height in pixel. When set to zero, Webservice will set a height for barcode image automatically.

The ninth parameter is **strToken**. Every licensed user will have a unique Token string for calling webmethods. If you leave it blank, returned barcode image will have a "Demo" watermark.

2.3 Codabar

Codabar accepts 9 parameters as listed below

```
<s:element minOccurs="0" maxOccurs="1" name="textToEncode" type="s:string"/>  
<s:element minOccurs="1" maxOccurs="1" name="ShowHumanReadable" type="s:boolean"/>  
<s:element minOccurs="1" maxOccurs="1" name="ShowCheckDigit" type="s:boolean"/>  
<s:element minOccurs="1" maxOccurs="1" name="ImageFormat" type="tns:BcsImageFormat"/>  
<s:element minOccurs="1" maxOccurs="1" name="Orientation" type="tns:BcsOrientation"/>  
<s:element minOccurs="1" maxOccurs="1" name="Resolution" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Width" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Height" type="s:int"/>  
<s:element minOccurs="0" maxOccurs="1" name="strToken" type="s:string"/>
```

The first parameter is **textToEncode**. This is your data to be encoded.

The second parameter is a Boolean. It tells the Webservice to show human readable text or not.

The third parameter is also a Boolean. It tells the Webservice to show check digit or not.

The fourth parameter is **ImageFormat**. It can be one of the following values: BMP, JPG, PNG, WMF, TIFF and GIF. It defines returned barcode image format.

The fifth parameter is **Orientation**. It is about returned barcode image orientation. It can be one of the following values: Original, Rotate90, Rotate180 and Rotate270.

The sixth parameter is **Resolution**. It sets returned barcode image resolution. Its value ranges between 96 and 600.

The seventh parameter is **Width**. It sets returned barcode image width in pixel. When set to zero, Webservice will set the shortest possible value for barcode image width.

The eighth parameter is **Height**. It sets returned barcode image height in pixel. When set to zero, Webservice will set a height for barcode image automatically.

The ninth parameter is **strToken**. Every licensed user will have a unique Token string for calling webmethods. If you leave it blank, returned barcode image will have a "Demo" watermark.

2.4 I25

I25, or Interleaved 2of5, accepts 9 parameters as listed below

```
<s:element minOccurs="0" maxOccurs="1" name="textToEncode" type="s:string"/>  
<s:element minOccurs="1" maxOccurs="1" name="ShowHumanReadable" type="s:boolean"/>  
<s:element minOccurs="1" maxOccurs="1" name="ShowCheckDigit" type="s:boolean"/>  
<s:element minOccurs="1" maxOccurs="1" name="ImageFormat" type="tns:BcsImageFormat"/>  
<s:element minOccurs="1" maxOccurs="1" name="Orientation" type="tns:BcsOrientation"/>  
<s:element minOccurs="1" maxOccurs="1" name="Resolution" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Width" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Height" type="s:int"/>  
<s:element minOccurs="0" maxOccurs="1" name="strToken" type="s:string"/>
```

The first parameter is **textToEncode**. This is your data to be encoded.

The second parameter is a Boolean. It tells the Webservice to show human readable text or not.

The third parameter is also a Boolean. It tells the Webservice to show check digit or not.

The fourth parameter is **ImageFormat**. It can be one of the following values: BMP, JPG, PNG, WMF, TIFF and GIF. It defines returned barcode image format.

The fifth parameter is **Orientation**. It is about returned barcode image orientation. It can be one of the following values: Original, Rotate90, Rotate180 and Rotate270.

The sixth parameter is **Resolution**. It sets returned barcode image resolution. Its value ranges between 96 and 600.

The seventh parameter is **Width**. It sets returned barcode image width in pixel. When set to zero, Webservice will set the shortest possible value for barcode image width.

The eighth parameter is **Height**. It sets returned barcode image height in pixel. When set to zero, Webservice will set a height for barcode image automatically.

The ninth parameter is **strToken**. Every licensed user will have a unique Token string for calling webmethods. If you leave it blank, returned barcode image will have a "Demo" watermark.

2.5 Code93

Code93 accepts 9 parameters as listed below

```
<s:element minOccurs="0" maxOccurs="1" name="textToEncode" type="s:string"/>  
<s:element minOccurs="1" maxOccurs="1" name="ShowHumanReadable" type="s:boolean"/>  
<s:element minOccurs="1" maxOccurs="1" name="ShowCheckDigit" type="s:boolean"/>  
<s:element minOccurs="1" maxOccurs="1" name="ImageFormat" type="tns:BcsImageFormat"/>  
<s:element minOccurs="1" maxOccurs="1" name="Orientation" type="tns:BcsOrientation"/>  
<s:element minOccurs="1" maxOccurs="1" name="Resolution" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Width" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Height" type="s:int"/>  
<s:element minOccurs="0" maxOccurs="1" name="strToken" type="s:string"/>
```

The first parameter is **textToEncode**. This is your data to be encoded.

The second parameter is a Boolean. It tells the Webservice to show human readable text or not.

The third parameter is also a Boolean. It tells the Webservice to show check digit or not.

The fourth parameter is **ImageFormat**. It can be one of the following values: BMP, JPG, PNG, WMF, TIFF and GIF. It defines returned barcode image format.

The fifth parameter is **Orientation**. It is about returned barcode image orientation. It can be one of the following values: Original, Rotate90, Rotate180 and Rotate270.

The sixth parameter is **Resolution**. It sets returned barcode image resolution. Its value ranges between 96 and 600.

The seventh parameter is **Width**. It sets returned barcode image width in pixel. When set to zero, Webservice will set the shortest possible value for barcode image width.

The eighth parameter is **Height**. It sets returned barcode image height in pixel. When set to zero, Webservice will set a height for barcode image automatically.

The ninth parameter is **strToken**. Every licensed user will have a unique Token string for calling webmethods. If you leave it blank, returned barcode image will have a "Demo" watermark.

2.6 Code128A

Code128A accepts 9 parameters as listed below

```
<s:element minOccurs="0" maxOccurs="1" name="textToEncode" type="s:string"/>  
<s:element minOccurs="1" maxOccurs="1" name="ShowHumanReadable" type="s:boolean"/>  
<s:element minOccurs="1" maxOccurs="1" name="ShowCheckDigit" type="s:boolean"/>  
<s:element minOccurs="1" maxOccurs="1" name="ImageFormat" type="tns:BcsImageFormat"/>  
<s:element minOccurs="1" maxOccurs="1" name="Orientation" type="tns:BcsOrientation"/>  
<s:element minOccurs="1" maxOccurs="1" name="Resolution" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Width" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Height" type="s:int"/>  
<s:element minOccurs="0" maxOccurs="1" name="strToken" type="s:string"/>
```

The first parameter is **textToEncode**. This is your data to be encoded. Please be reminded that Code128A accepts uppercase alpha and control characters.

The second parameter is a Boolean. It tells the Webservice to show human readable text or not.

The third parameter is also a Boolean. It tells the Webservice to show check digit or not.

The fourth parameter is **ImageFormat**. It can be one of the following values: BMP, JPG, PNG, WMF, TIFF and GIF. It defines returned barcode image format.

The fifth parameter is **Orientation**. It is about returned barcode image orientation. It can be one of the following values: Original, Rotate90, Rotate180 and Rotate270.

The sixth parameter is **Resolution**. It sets returned barcode image resolution. Its value ranges between 96 and 600.

The seventh parameter is **Width**. It sets returned barcode image width in pixel. When set to zero, Webservice will set the shortest possible value for barcode image width.

The eighth parameter is **Height**. It sets returned barcode image height in pixel. When set to zero, Webservice will set a height for barcode image automatically.

The ninth parameter is **strToken**. Every licensed user will have a unique Token string for calling webmethods. If you leave it blank, returned barcode image will have a "Demo" watermark.

2.7 Code128B

Code128B accepts 9 parameters as listed below

```
<s:element minOccurs="0" maxOccurs="1" name="textToEncode" type="s:string"/>  
<s:element minOccurs="1" maxOccurs="1" name="ShowHumanReadable" type="s:boolean"/>  
<s:element minOccurs="1" maxOccurs="1" name="ShowCheckDigit" type="s:boolean"/>  
<s:element minOccurs="1" maxOccurs="1" name="ImageFormat" type="tns:BcsImageFormat"/>  
<s:element minOccurs="1" maxOccurs="1" name="Orientation" type="tns:BcsOrientation"/>  
<s:element minOccurs="1" maxOccurs="1" name="Resolution" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Width" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Height" type="s:int"/>  
<s:element minOccurs="0" maxOccurs="1" name="strToken" type="s:string"/>
```

The first parameter is **textToEncode**. This is your data to be encoded. Please be aware that Code128B accepts both uppercase and lowercase alpha.

The second parameter is a Boolean. It tells the Webservice to show human readable text or not.

The third parameter is also a Boolean. It tells the Webservice to show check digit or not.

The fourth parameter is **ImageFormat**. It can be one of the following values: BMP, JPG, PNG, WMF, TIFF and GIF. It defines returned barcode image format.

The fifth parameter is **Orientation**. It is about returned barcode image orientation. It can be one of the following values: Original, Rotate90, Rotate180 and Rotate270.

The sixth parameter is **Resolution**. It sets returned barcode image resolution. Its value ranges between 96 and 600.

The seventh parameter is **Width**. It sets returned barcode image width in pixel. When set to zero, Webservice will set the shortest possible value for barcode image width.

The eighth parameter is **Height**. It sets returned barcode image height in pixel. When set to zero, Webservice will set a height for barcode image automatically.

The ninth parameter is **strToken**. Every licensed user will have a unique Token string for calling webmethods. If you leave it blank, returned barcode image will have a "Demo" watermark.

2.8 Code128C

Code128C accepts 9 parameters as listed below

```
<s:element minOccurs="0" maxOccurs="1" name="textToEncode" type="s:string"/>  
<s:element minOccurs="1" maxOccurs="1" name="ShowHumanReadable" type="s:boolean"/>  
<s:element minOccurs="1" maxOccurs="1" name="ShowCheckDigit" type="s:boolean"/>  
<s:element minOccurs="1" maxOccurs="1" name="ImageFormat" type="tns:BcsImageFormat"/>  
<s:element minOccurs="1" maxOccurs="1" name="Orientation" type="tns:BcsOrientation"/>  
<s:element minOccurs="1" maxOccurs="1" name="Resolution" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Width" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Height" type="s:int"/>  
<s:element minOccurs="0" maxOccurs="1" name="strToken" type="s:string"/>
```

The first parameter is **textToEncode**. This is your data to be encoded. Please be aware that Code128C accepts numeric-only data.

The second parameter is a Boolean. It tells the Webservice to show human readable text or not.

The third parameter is also a Boolean. It tells the Webservice to show check digit or not.

The fourth parameter is **ImageFormat**. It can be one of the following values: BMP, JPG, PNG, WMF, TIFF and GIF. It defines returned barcode image format.

The fifth parameter is **Orientation**. It is about returned barcode image orientation. It can be one of the following values: Original, Rotate90, Rotate180 and Rotate270.

The sixth parameter is **Resolution**. It sets returned barcode image resolution. Its value ranges between 96 and 600.

The seventh parameter is **Width**. It sets returned barcode image width in pixel. When set to zero, Webservice will set the shortest possible value for barcode image width.

The eighth parameter is **Height**. It sets returned barcode image height in pixel. When set to zero, Webservice will set a height for barcode image automatically.

The ninth parameter is **strToken**. Every licensed user will have a unique Token string for calling webmethods. If you leave it blank, returned barcode image will have a "Demo" watermark.

2.9 GS1128

GS1128 accepts 9 parameters as listed below

```
<s:element minOccurs="0" maxOccurs="1" name="textToEncode" type="s:string"/>  
<s:element minOccurs="1" maxOccurs="1" name="ShowHumanReadable" type="s:boolean"/>  
<s:element minOccurs="1" maxOccurs="1" name="ShowCheckDigit" type="s:boolean"/>  
<s:element minOccurs="1" maxOccurs="1" name="ImageFormat" type="tns:BcsImageFormat"/>  
<s:element minOccurs="1" maxOccurs="1" name="Orientation" type="tns:BcsOrientation"/>  
<s:element minOccurs="1" maxOccurs="1" name="Resolution" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Width" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Height" type="s:int"/>  
<s:element minOccurs="0" maxOccurs="1" name="strToken" type="s:string"/>
```

The first parameter is **textToEncode**. This is your data to be encoded. Please specify your Application Identifiers with brackets.

The second parameter is a Boolean. It tells the Webservice to show human readable text or not.

The third parameter is also a Boolean. It tells the Webservice to show check digit or not.

The fourth parameter is **ImageFormat**. It can be one of the following values: BMP, JPG, PNG, WMF, TIFF and GIF. It defines returned barcode image format.

The fifth parameter is **Orientation**. It is about returned barcode image orientation. It can be one of the following values: Original, Rotate90, Rotate180 and Rotate270.

The sixth parameter is **Resolution**. It sets returned barcode image resolution. Its value ranges between 96 and 600.

The seventh parameter is **Width**. It sets returned barcode image width in pixel. When set to zero, Webservice will set the shortest possible value for barcode image width.

The eighth parameter is **Height**. It sets returned barcode image height in pixel. When set to zero, Webservice will set a height for barcode image automatically.

The ninth parameter is **strToken**. Every licensed user will have a unique Token string for calling webmethods. If you leave it blank, returned barcode image will have a "Demo" watermark.

2.10 Code25

Code25 accepts 9 parameters as listed below

```
<s:element minOccurs="0" maxOccurs="1" name="textToEncode" type="s:string"/>  
<s:element minOccurs="1" maxOccurs="1" name="ShowHumanReadable" type="s:boolean"/>  
<s:element minOccurs="1" maxOccurs="1" name="ShowCheckDigit" type="s:boolean"/>  
<s:element minOccurs="1" maxOccurs="1" name="ImageFormat" type="tns:BcsImageFormat"/>  
<s:element minOccurs="1" maxOccurs="1" name="Orientation" type="tns:BcsOrientation"/>  
<s:element minOccurs="1" maxOccurs="1" name="Resolution" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Width" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Height" type="s:int"/>  
<s:element minOccurs="0" maxOccurs="1" name="strToken" type="s:string"/>
```

The first parameter is **textToEncode**. This is your data to be encoded. It is numeric-only.

The second parameter is a Boolean. It tells the Webservice to show human readable text or not.

The third parameter is also a Boolean. It tells the Webservice to show check digit or not.

The fourth parameter is **ImageFormat**. It can be one of the following values: BMP, JPG, PNG, WMF, TIFF and GIF. It defines returned barcode image format.

The fifth parameter is **Orientation**. It is about returned barcode image orientation. It can be one of the following values: Original, Rotate90, Rotate180 and Rotate270.

The sixth parameter is **Resolution**. It sets returned barcode image resolution. Its value ranges between 96 and 600.

The seventh parameter is **Width**. It sets returned barcode image width in pixel. When set to zero, Webservice will set the shortest possible value for barcode image width.

The eighth parameter is **Height**. It sets returned barcode image height in pixel. When set to zero, Webservice will set a height for barcode image automatically.

The ninth parameter is **strToken**. Every licensed user will have a unique Token string for calling webmethods. If you leave it blank, returned barcode image will have a "Demo" watermark.

2.11 Code11

Code11 accepts 9 parameters as listed below

```
<s:element minOccurs="0" maxOccurs="1" name="textToEncode" type="s:string"/>  
<s:element minOccurs="1" maxOccurs="1" name="ShowHumanReadable" type="s:boolean"/>  
<s:element minOccurs="1" maxOccurs="1" name="ShowCheckDigit" type="s:boolean"/>  
<s:element minOccurs="1" maxOccurs="1" name="ImageFormat" type="tns:BcsImageFormat"/>  
<s:element minOccurs="1" maxOccurs="1" name="Orientation" type="tns:BcsOrientation"/>  
<s:element minOccurs="1" maxOccurs="1" name="Resolution" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Width" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Height" type="s:int"/>  
<s:element minOccurs="0" maxOccurs="1" name="strToken" type="s:string"/>
```

The first parameter is **textToEncode**. This is your data to be encoded.

The second parameter is a Boolean. It tells the Webservice to show human readable text or not.

The third parameter is also a Boolean. It tells the Webservice to show check digit or not.

The fourth parameter is **ImageFormat**. It can be one of the following values: BMP, JPG, PNG, WMF, TIFF and GIF. It defines returned barcode image format.

The fifth parameter is **Orientation**. It is about returned barcode image orientation. It can be one of the following values: Original, Rotate90, Rotate180 and Rotate270.

The sixth parameter is **Resolution**. It sets returned barcode image resolution. Its value ranges between 96 and 600.

The seventh parameter is **Width**. It sets returned barcode image width in pixel. When set to zero, Webservice will set the shortest possible value for barcode image width.

The eighth parameter is **Height**. It sets returned barcode image height in pixel. When set to zero, Webservice will set a height for barcode image automatically.

The ninth parameter is **strToken**. Every licensed user will have a unique Token string for calling webmethods. If you leave it blank, returned barcode image will have a "Demo" watermark.

2.12 Telepen

Telepen accepts 9 parameters as listed below

```
<s:element minOccurs="0" maxOccurs="1" name="textToEncode" type="s:string"/>  
<s:element minOccurs="1" maxOccurs="1" name="ShowHumanReadable" type="s:boolean"/>  
<s:element minOccurs="1" maxOccurs="1" name="ShowCheckDigit" type="s:boolean"/>  
<s:element minOccurs="1" maxOccurs="1" name="ImageFormat" type="tns:BcsImageFormat"/>  
<s:element minOccurs="1" maxOccurs="1" name="Orientation" type="tns:BcsOrientation"/>  
<s:element minOccurs="1" maxOccurs="1" name="Resolution" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Width" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Height" type="s:int"/>  
<s:element minOccurs="0" maxOccurs="1" name="strToken" type="s:string"/>
```

The first parameter is **textToEncode**. This is your data to be encoded.

The second parameter is a Boolean. It tells the Webservice to show human readable text or not.

The third parameter is also a Boolean. It tells the Webservice to show check digit or not.

The fourth parameter is **ImageFormat**. It can be one of the following values: BMP, JPG, PNG, WMF, TIFF and GIF. It defines returned barcode image format.

The fifth parameter is **Orientation**. It is about returned barcode image orientation. It can be one of the following values: Original, Rotate90, Rotate180 and Rotate270.

The sixth parameter is **Resolution**. It sets returned barcode image resolution. Its value ranges between 96 and 600.

The seventh parameter is **Width**. It sets returned barcode image width in pixel. When set to zero, Webservice will set the shortest possible value for barcode image width.

The eighth parameter is **Height**. It sets returned barcode image height in pixel. When set to zero, Webservice will set a height for barcode image automatically.

The ninth parameter is **strToken**. Every licensed user will have a unique Token string for calling webmethods. If you leave it blank, returned barcode image will have a "Demo" watermark.

2.13 MSI

MSI, or Plessey, accepts 9 parameters as listed below

```
<s:element minOccurs="0" maxOccurs="1" name="textToEncode" type="s:string"/>  
<s:element minOccurs="1" maxOccurs="1" name="ShowHumanReadable" type="s:boolean"/>  
<s:element minOccurs="1" maxOccurs="1" name="ShowCheckDigit" type="s:boolean"/>  
<s:element minOccurs="1" maxOccurs="1" name="ImageFormat" type="tns:BcsImageFormat"/>  
<s:element minOccurs="1" maxOccurs="1" name="Orientation" type="tns:BcsOrientation"/>  
<s:element minOccurs="1" maxOccurs="1" name="Resolution" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Width" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Height" type="s:int"/>  
<s:element minOccurs="0" maxOccurs="1" name="strToken" type="s:string"/>
```

The first parameter is **textToEncode**. This is your data to be encoded.

The second parameter is a Boolean. It tells the Webservice to show human readable text or not.

The third parameter is also a Boolean. It tells the Webservice to show check digit or not.

The fourth parameter is **ImageFormat**. It can be one of the following values: BMP, JPG, PNG, WMF, TIFF and GIF. It defines returned barcode image format.

The fifth parameter is **Orientation**. It is about returned barcode image orientation. It can be one of the following values: Original, Rotate90, Rotate180 and Rotate270.

The sixth parameter is **Resolution**. It sets returned barcode image resolution. Its value ranges between 96 and 600.

The seventh parameter is **Width**. It sets returned barcode image width in pixel. When set to zero, Webservice will set the shortest possible value for barcode image width.

The eighth parameter is **Height**. It sets returned barcode image height in pixel. When set to zero, Webservice will set a height for barcode image automatically.

The ninth parameter is **strToken**. Every licensed user will have a unique Token string for calling webmethods. If you leave it blank, returned barcode image will have a "Demo" watermark.

2.14 UPCA

UPCA accepts 8 parameters as listed below

```
<s:element minOccurs="0" maxOccurs="1" name="textToEncode" type="s:string"/>  
<s:element minOccurs="0" maxOccurs="1" name="strAddon" type="s:string"/>  
<s:element minOccurs="1" maxOccurs="1" name="ImageFormat" type="tns:BcsImageFormat"/>  
<s:element minOccurs="1" maxOccurs="1" name="Orientation" type="tns:BcsOrientation"/>  
<s:element minOccurs="1" maxOccurs="1" name="Resolution" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Width" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Height" type="s:int"/>  
<s:element minOccurs="0" maxOccurs="1" name="strToken" type="s:string"/>
```

The first parameter is **textToEncode**. This is your data to be encoded.

The second parameter is **strAddon**. It sets the addon of a UPC barcode. It can be either 2-digits or 5-digits.

The third parameter is **ImageFormat**. It can be one of the following values: BMP, JPG, PNG, WMF, TIFF and GIF. It defines returned barcode image format.

The fourth parameter is **Orientation**. It is about returned barcode image orientation. It can be one of the following values: Original, Rotate90, Rotate180 and Rotate270.

The fifth parameter is **Resolution**. It sets returned barcode image resolution. Its value ranges between 96 and 600.

The sixth parameter is **Width**. It sets returned barcode image width in pixel. When set to zero, Webservice will set the barcode image width to be 115 pixels.

The seventh parameter is **Height**. It sets returned barcode image height in pixel. When set to zero, Webservice will set the height for barcode image to be 100 pixels.

The eighth parameter is **strToken**. Every licensed user will have a unique Token string for calling webmethods. If you leave it blank, returned barcode image will have a "Demo" watermark.

2.15 UPCE

UPCE accepts 8 parameters as listed below

```
<s:element minOccurs="0" maxOccurs="1" name="textToEncode" type="s:string"/>  
<s:element minOccurs="0" maxOccurs="1" name="strAddon" type="s:string"/>  
<s:element minOccurs="1" maxOccurs="1" name="ImageFormat" type="tns:BcsImageFormat"/>  
<s:element minOccurs="1" maxOccurs="1" name="Orientation" type="tns:BcsOrientation"/>  
<s:element minOccurs="1" maxOccurs="1" name="Resolution" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Width" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Height" type="s:int"/>  
<s:element minOccurs="0" maxOccurs="1" name="strToken" type="s:string"/>
```

The first parameter is **textToEncode**. This is your data to be encoded.

The second parameter is **strAddon**. It sets the addon of a UPC barcode. It can be either 2-digits or 5-digits.

The third parameter is **ImageFormat**. It can be one of the following values: BMP, JPG, PNG, WMF, TIFF and GIF. It defines returned barcode image format.

The fourth parameter is **Orientation**. It is about returned barcode image orientation. It can be one of the following values: Original, Rotate90, Rotate180 and Rotate270.

The fifth parameter is **Resolution**. It sets returned barcode image resolution. Its value ranges between 96 and 600.

The sixth parameter is **Width**. It sets returned barcode image width in pixel. When set to zero, Webservice will set the barcode image width to be 115 pixels.

The seventh parameter is **Height**. It sets returned barcode image height in pixel. When set to zero, Webservice will set the height for barcode image to be 100 pixels.

The eighth parameter is **strToken**. Every licensed user will have a unique Token string for calling webmethods. If you leave it blank, returned barcode image will have a "Demo" watermark.

2.16 EAN13

EAN13 accepts 8 parameters as listed below

```
<s:element minOccurs="0" maxOccurs="1" name="textToEncode" type="s:string"/>  
<s:element minOccurs="0" maxOccurs="1" name="strAddon" type="s:string"/>  
<s:element minOccurs="1" maxOccurs="1" name="ImageFormat" type="tns:BcsImageFormat"/>  
<s:element minOccurs="1" maxOccurs="1" name="Orientation" type="tns:BcsOrientation"/>  
<s:element minOccurs="1" maxOccurs="1" name="Resolution" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Width" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Height" type="s:int"/>  
<s:element minOccurs="0" maxOccurs="1" name="strToken" type="s:string"/>
```

The first parameter is **textToEncode**. This is your data to be encoded.

The second parameter is **strAddon**. It sets the addon of a UPC barcode. It can be either 2-digits or 5-digits.

The third parameter is **ImageFormat**. It can be one of the following values: BMP, JPG, PNG, WMF, TIFF and GIF. It defines returned barcode image format.

The fourth parameter is **Orientation**. It is about returned barcode image orientation. It can be one of the following values: Original, Rotate90, Rotate180 and Rotate270.

The fifth parameter is **Resolution**. It sets returned barcode image resolution. Its value ranges between 96 and 600.

The sixth parameter is **Width**. It sets returned barcode image width in pixel. When set to zero, Webservice will set the barcode image width to be 115 pixels.

The seventh parameter is **Height**. It sets returned barcode image height in pixel. When set to zero, Webservice will set the height for barcode image to be 100 pixels.

The eighth parameter is **strToken**. Every licensed user will have a unique Token string for calling webmethods. If you leave it blank, returned barcode image will have a "Demo" watermark.

2.17 EAN8

EAN8 accepts 8 parameters as listed below

```
<s:element minOccurs="0" maxOccurs="1" name="textToEncode" type="s:string"/>  
<s:element minOccurs="0" maxOccurs="1" name="strAddon" type="s:string"/>  
<s:element minOccurs="1" maxOccurs="1" name="ImageFormat" type="tns:BcsImageFormat"/>  
<s:element minOccurs="1" maxOccurs="1" name="Orientation" type="tns:BcsOrientation"/>  
<s:element minOccurs="1" maxOccurs="1" name="Resolution" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Width" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Height" type="s:int"/>  
<s:element minOccurs="0" maxOccurs="1" name="strToken" type="s:string"/>
```

The first parameter is **textToEncode**. This is your data to be encoded.

The second parameter is **strAddon**. It sets the addon of a UPC barcode. It can be either 2-digits or 5-digits.

The third parameter is **ImageFormat**. It can be one of the following values: BMP, JPG, PNG, WMF, TIFF and GIF. It defines returned barcode image format.

The fourth parameter is **Orientation**. It is about returned barcode image orientation. It can be one of the following values: Original, Rotate90, Rotate180 and Rotate270.

The fifth parameter is **Resolution**. It sets returned barcode image resolution. Its value ranges between 96 and 600.

The sixth parameter is **Width**. It sets returned barcode image width in pixel. When set to zero, Webservice will set the barcode image width to be 115 pixels.

The seventh parameter is **Height**. It sets returned barcode image height in pixel. When set to zero, Webservice will set the height for barcode image to be 100 pixels.

The eighth parameter is **strToken**. Every licensed user will have a unique Token string for calling webmethods. If you leave it blank, returned barcode image will have a "Demo" watermark.

2.18 Bookland

Bookland accepts 8 parameters as listed below

```
<s:element minOccurs="0" maxOccurs="1" name="textToEncode" type="s:string"/>  
<s:element minOccurs="0" maxOccurs="1" name="strAddon" type="s:string"/>  
<s:element minOccurs="1" maxOccurs="1" name="ImageFormat" type="tns:BcsImageFormat"/>  
<s:element minOccurs="1" maxOccurs="1" name="Orientation" type="tns:BcsOrientation"/>  
<s:element minOccurs="1" maxOccurs="1" name="Resolution" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Width" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Height" type="s:int"/>  
<s:element minOccurs="0" maxOccurs="1" name="strToken" type="s:string"/>
```

The first parameter is **textToEncode**. This is your data to be encoded.

The second parameter is **strAddon**. It sets the addon of a UPC barcode. It can be either 2-digits or 5-digits.

The third parameter is **ImageFormat**. It can be one of the following values: BMP, JPG, PNG, WMF, TIFF and GIF. It defines returned barcode image format.

The fourth parameter is **Orientation**. It is about returned barcode image orientation. It can be one of the following values: Original, Rotate90, Rotate180 and Rotate270.

The fifth parameter is **Resolution**. It sets returned barcode image resolution. Its value ranges between 96 and 600.

The sixth parameter is **Width**. It sets returned barcode image width in pixel. When set to zero, Webservice will set the barcode image width to be 115 pixels.

The seventh parameter is **Height**. It sets returned barcode image height in pixel. When set to zero, Webservice will set the height for barcode image to be 100 pixels.

The eighth parameter is **strToken**. Every licensed user will have a unique Token string for calling webmethods. If you leave it blank, returned barcode image will have a "Demo" watermark.

2.19 Datamatrix

Datamatrix accepts 8 parameters as listed below

```
<s:element minOccurs="0" maxOccurs="1" name="textToEncode" type="s:string"/>  
<s:element minOccurs="1" maxOccurs="1" name="Format" type="tns:DataMatrixFormat"/>  
<s:element minOccurs="1" maxOccurs="1" name="ImageFormat" type="tns:BcsImageFormat"/>  
<s:element minOccurs="1" maxOccurs="1" name="Orientation" type="tns:BcsOrientation"/>  
<s:element minOccurs="1" maxOccurs="1" name="Resolution" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Width" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Height" type="s:int"/>  
<s:element minOccurs="0" maxOccurs="1" name="strToken" type="s:string"/>
```

The first parameter is **textToEncode**. This is your data to be encoded.

The second parameter is **DataMatrixFormat**. It sets barcode to be one of the 30 predefined data matrix format. Set it to zero for automatic selection.

The third parameter is **ImageFormat**. It can be one of the following values: BMP, JPG, PNG, WMF, TIFF and GIF. It defines returned barcode image format.

The fourth parameter is **Orientation**. It is about returned barcode image orientation. It can be one of the following values: Original, Rotate90, Rotate180 and Rotate270.

The fifth parameter is **Resolution**. It sets returned barcode image resolution. Its value ranges between 96 and 600.

The sixth parameter is **Width**. It sets returned barcode image width in pixel. When set to zero, Webservice will set the smallest barcode image width in pixel.

The seventh parameter is **Height**. It sets returned barcode image height in pixel. When set to zero, Webservice will set the smallest barcode image height in pixel.

The eighth parameter is **strToken**. Every licensed user will have a unique Token string for calling webmethods. If you leave it blank, returned barcode image will have a "Demo" watermark.

2.20 QRCode

QRCode accepts 8 parameters as listed below

```
<s:element minOccurs="0" maxOccurs="1" name="textToEncode" type="s:string"/>  
<s:element minOccurs="1" maxOccurs="1" name="EccLevel" type="tns:QrcodeErrorLvl"/>  
<s:element minOccurs="1" maxOccurs="1" name="QRCodeFormat" type="tns:QrcodeFormat"/>  
<s:element minOccurs="1" maxOccurs="1" name="ImageFormat" type="tns:BcsImageFormat"/>  
<s:element minOccurs="1" maxOccurs="1" name="Orientation" type="tns:BcsOrientation"/>  
<s:element minOccurs="1" maxOccurs="1" name="Resolution" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Size" type="s:int"/>  
<s:element minOccurs="0" maxOccurs="1" name="strToken" type="s:string"/>
```

The first parameter is **textToEncode**. This is your data to be encoded.

The second parameter is **EccLevel**. This is the error correction level. It can be one of the following values: L07, M15, Q25, H30.

The third parameter is **QRCodeFormat**. It sets barcode to be one of the 40 predefined QRCode formats. Set it to zero for automatic selection.

The fourth parameter is **ImageFormat**. It can be one of the following values: BMP, JPG, PNG, WMF, TIFF and GIF. It defines returned barcode image format.

The fifth parameter is **Orientation**. It is about returned barcode image orientation. It can be one of the following values: Original, Rotate90, Rotate180 and Rotate270.

The sixth parameter is **Resolution**. It sets returned barcode image resolution. Its value ranges between 96 and 600.

The seventh parameter is **Size**. It sets returned barcode image size in pixel. When set to zero, Webservice will set the smallest barcode image size in pixel.

The eighth parameter is **strToken**. Every licensed user will have a unique Token string for calling webmethods. If you leave it blank, returned barcode image will have a "Demo" watermark.

2.21 PDF417

PDF417 accepts 12 parameters as listed below

```
<s:element minOccurs="0" maxOccurs="1" name="textToEncode" type="s:string"/>  
<s:element minOccurs="1" maxOccurs="1" name="ImageFormat" type="tns:BcsImageFormat"/>  
<s:element minOccurs="1" maxOccurs="1" name="Orientation" type="tns:BcsOrientation"/>  
<s:element minOccurs="1" maxOccurs="1" name="Resolution" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="MaxRow" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="FixedColumn" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Security" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="CompactMode" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Truncated" type="s:boolean"/>  
<s:element minOccurs="1" maxOccurs="1" name="Width" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="Height" type="s:int"/>  
<s:element minOccurs="0" maxOccurs="1" name="strToken" type="s:string"/>
```

The first parameter is textToEncode. This is your data to be encoded.

The second parameter is ImageFormat. It can be one of the following values: BMP, JPG, PNG, WMF, TIFF and GIF. It defines returned barcode image format.

The third parameter is Orientation. It is about returned barcode image orientation. It can be one of the following values: Original, Rotate90, Rotate180 and Rotate270.

The fourth parameter is Resolution. It sets returned barcode image resolution. Its value ranges between 96 and 600.

The fifth parameter is MaxRows. It sets maximum rows of returned PDF417 barcode.

The sixth parameter is FixedColumns. It sets maximum columns of returned PDF417 barcode.

The seventh parameter is Security. It sets the security level of returned PDF417 barcode. Its values ranges between 0 and 9. The higher level, the larger barcode size. Set to zero for automatic selection.

The eighth parameter is CompactMode. This values sets the Compaction Mode of PDF417. Default value is 0, it means mixed mode. 1 for text mode; 2 for number mode; 3 for binary mode.

The ninth parameter Truncated is a boolean. Its default value is false. When set to true, PDF417 barcode is truncated.

The tenth parameter is Width. It sets returned barcode image width in pixel. When set to zero, Webservice will set the smallest barcode image width in pixel.

The eleventh parameter is Height. It sets returned barcode image height in pixel. When set to zero, Webservice will set the smallest barcode image height in pixel.

The twelfth parameter is strToken. Every licensed user will have a unique Token string for calling webmethods. If you leave it blank, returned barcode image will have a “Demo” watermark.

2.22 Aztec

Aztec accepts 8 parameters as listed below

```
<s:element minOccurs="0" maxOccurs="1" name="textToEncode" type="s:string"/>  
<s:element minOccurs="1" maxOccurs="1" name="ImageFormat" type="tns:BcsImageFormat"/>  
<s:element minOccurs="1" maxOccurs="1" name="Orientation" type="tns:BcsOrientation"/>  
<s:element minOccurs="1" maxOccurs="1" name="Resolution" type="s:int"/>  
<s:element minOccurs="1" maxOccurs="1" name="nFormat" type="tns:AztecTargetSize"/>  
<s:element minOccurs="1" maxOccurs="1" name="ErrorLevel" type="tns:AztecErrorLevel"/>  
<s:element minOccurs="1" maxOccurs="1" name="Size" type="s:int"/>  
<s:element minOccurs="0" maxOccurs="1" name="strToken" type="s:string"/>  
</s:sequence>
```

The first parameter is **textToEncode**. This is your data to be encoded.

The second parameter is **ImageFormat**. It can be one of the following values: BMP, JPG, PNG, WMF, TIFF and GIF. It defines returned barcode image format.

The third parameter is **Orientation**. It is about returned barcode image orientation. It can be one of the following values: Original, Rotate90, Rotate180 and Rotate270.

The fourth parameter is **Resolution**. It sets returned barcode image resolution. Its value ranges between 96 and 600.

The sixth parameter is **Format**. It sets barcode to be one of the 36 predefined Aztec formats. Set it to zero for automatic selection.

The fifth parameter is **ErrorLevel**. This is the error correction level. Its possible values are defined in **AztecErrorLevel**.

The seventh parameter is **Size**. It sets returned barcode image size in pixel. When set to zero, Webservice will set the smallest barcode image size in pixel.

The eighth parameter is **strToken**. Every licensed user will have a unique Token string for calling webmethods. If you leave it blank, returned barcode image will have a "Demo" watermark.

Chapter 3 XML Schema

WSDL uses XML Schema to describe data types used in a web service. The following XML schemas are defined in the Webservice for webmethods to accept parameter.

3.1 BcsImageFormat

Its possible values are listed below:

BMP, JPG, PNG, WMF, TIFF, GIF

3.2 BcsOrientation

Its possible values are listed below

Original, Rotate90, Rotate180, Rotate270

3.3 DataMatrixFormat

Its possible values are listed below

AutoSelect, _10x10, _12x12, _14x14, _16x16, _18x18, _20x20, _22x22, _24x24, _26x26, _32x32, _36x36, _40x40, _44x44, _48x48, _52x52, _64x64, _72x72, _80x80, _88x88, _96x96, _104x104, _120x120, _134x134, _144x144, _8x18, _8x32, _12x26, _12x36, _16x36, _16x48

3.4 QrcodeErrorLvl

QRCode has 4 different Error Correction Levels. The higher Error Correction Level, the less decoding error chance. Its possible values are listed below

L07, M15, Q25, H30

3.5 QrcodeFormat

Its possible values are listed below:

AutoSelect, _21x21, _25x25, _29x29, _33x33, _37x37, _41x41, _45x45, _49x49, _53x53, _57x57, _61x61, _65x65, _69x69, _73x73, _77x77, _81x81, _85x85, _89x89, _93x93, _97x97, _101x101, _105x105, _109x109, _113x113, _117x117, _121x121, _125x125, _129x129, _133x133, _137x137, _141x141, _145x145, _149x149, _153x153, _157x157, _161x161, _165x165, _169x169, _173x173, _177x177.

3.6 AztecTargetSize

Its possible values are listed below:

AutoSelect, _15x15Compact, _19x19, _19x19Compact, _23x23, _23x23Compact, _27x27, _27x27Compact, _31x31, _37x37, _41x41, _45x45, _49x49, _53x53, _57x57, _61x61, _67x67, _71x71, _75x75, _79x79, _83x83, _87x87, _91x91, _95x95, _101x101, _105x105, _109x109, _113x113, _117x117, _121x121, _125x125, _131x131, _135x135, _139x139, _143x143, _147x147, _151x151.

3.7 AztecErrorLevel

Aztec Code employed Reed Solomon algorithm for error correction calculation. Barcodesoft Webservice predefined 20 error correction levels. Its default value is AztecECL23 that represents 23% Error Correction. Its possible values are listed below:

AutoSelect, AztecECL5, AztecECL10, AztecECL15, AztecECL20, AztecECL23, AztecECL25, AztecECL30, AztecECL35, AztecECL40, AztecECL45, AztecECL50, AztecECL55, AztecECL60, AztecECL65, AztecECL70, AztecECL75, AztecECL80, AztecECL85, AztecECL90.

Chapter 4 Generate Barcode Using Webservice

Barcodesoft Webservice returns a dynamically generated barcode image according to input parameters: barcode data, image size, resolution, orientation and format. You can easily integrate Barcodesoft Webservice with your own web application or desktop application.

Web Services are platform-independent and language-independent. Therefore, users can call Webmethod from different operating system and different programming language. Here are some examples in PHP and C#.

4.1 Using Webservice in PHP

PHP is a widely used server-side scripting language designed to produce dynamic Web pages. It can be embedded into HTML source code.

You can call all Webmethods of Barcodesoft Webservice using some simple PHP codes. Here are some code snippets to call Code39 and QRCode. You can modify these codes to generate barcode in other symbology.

```
<?

$symbology=$_REQUEST['symbology'];

$orientation=$_REQUEST['orientation'];

$height=$_REQUEST['height'];

$imageType=$_REQUEST['imageType'];

$width=$_REQUEST['width'];

$humanReadable=$_REQUEST['humanReadable'];

$resolution=$_REQUEST['resolution'];

$txt=$_REQUEST['edate'];



if ($symbology == 'Code39')

{

$params=array(

    'ShowHumanReadable'=>$humanReadable,

    'textToEncode' => $txt,

    'ShowCheckDigit'=>true,
```

```
'ImageFormat'=>$imageType,  
'Orientation'=>$orientation,  
'Resolution'=>$resolution,  
'Width'=>$width,  
'Height'=>$height,  
'strToken'=>'BarCodeSoft',  
);  
  
$client = new SoapClient("http://www.barcodesoft.com/bcdgen.asmx?WSDL");  
  
  
$result = $client->Code39($params);  
header("Content-type:image/png");  
echo $result->Code39Result;  
exit;  
}  
  
  
if ($symbology == 'QRCode')  
{  
  
  
$params=array(  
'ShowHumanReadable'=>$humanReadable,  
'textToEncode' => $txt,  
'EccLevel'=>'M15',  
'Size'=>0,  
'QRCodeFormat'=>'AutoSelect',  
'ShowCheckDigit'=>true,  
'ImageFormat'=>$imageType,  
'Orientation'=>$orientation,
```

```
'Resolution'=>$resolution,  
'Width'=>$width,  
'Height'=>$height,  
'strToken'=>'BarCodeSoft',  
);  
  
$client = new SoapClient("http://www.barcodesoft.com/bcdgen.asmx?WSDL");  
  
$result = $client->QRCode($params);  
  
header("Content-type:image/png");  
  
echo $result->QRCodeResult;  
  
exit;  
}  
  
?>
```

You will have to get your own token before calling Webmethods. Otherwise, returned barcode image will have “Demo” watermark.

4.2 Using Webservice in C#

ASP.NET and C# language are highly promoted by Microsoft. It is becoming one of the best designed web application platform. And C# is becoming more and more popular nowadays.

To call a Webmethod in C#, you need to add reference to Barcodesoft Webservice first. This is the WSDL URL to add as reference

<http://www.barcodesoft.com/bcdgen.asmx?WSDL>

Here are some code snippets to call Code39. You can modify these codes to generate barcode in other symbology.

```
BarcodeSoft.BarCodeWebServiceSoapClient BarcodeSoap1 = new BarcodeSoft.BarCodeWebServiceSoapClient();

byte[] Barcode = BarcodeSoap1.Code39(StrToEncode.Text, true, true,
WebserviceClient.BarcodeSoft.BcsImageFormat.PNG, WebserviceClient.BarcodeSoft.BcsOrientation.Original, 400,
180, 108, "BarCodeSoft");

BitmapImage BarcodeBitmap = new BitmapImage();

BarcodeBitmap.BeginInit();

if (Barcode != null)

{

    MemoryStream BarcodeStream = new MemoryStream(Barcode);

    BarcodeBitmap.StreamSource = BarcodeStream;

    BarcodeBitmap.EndInit();

    BarcodeImage.Source = BarcodeBitmap;

}
```

You will have to get your own token before calling Webmethods. Otherwise, returned barcode image will have “Demo” watermark.

Chapter 5 End User License Agreement

By using Barcodesoft Webservice Webmethods (referred to as Software in this agreement) created by Barcodesoft, you (or you on behalf of your employer, referred to in this Agreement as 'Recipient' or 'You') are agreeing to be bound by the terms and conditions of this License Agreement. If you do not agree, do not install, copy or use the Software.

1. License

(a) Subject to Recipient's compliance with the terms and conditions of this Agreement, Barcodesoft grants to Recipient a non-exclusive, nontransferable right and license to install, copy, reproduce, display and otherwise use the Software in consideration for the license fee paid.

The Developer License allows your developer

- to rent, lease or distribute the licensed client side software bundled with your application up to 10,000 uses,
- and/or to embed the Software on one or more web pages of one domain up to 10,000 page views, provided all copyright notices of the Software are left unmodified. If use number exceeds 10,000, additional Developer License is required. Annual maintenance fee is required after one full year.

The Unlimited Developer License allows full access to the software of your developers, unlimited server deployment, and royalty-free distribution of client side Software bundled with your application within or outside your organization.

(b) Recipient may utilize the Software in the development and distribution of material in any media now or hereafter known throughout the universe but may not reverse engineer, decompile or disassemble any portion of the Software, except and only to the extent that this limitation is expressly prohibited by applicable law notwithstanding this limitation.

2. Copyright and Trademark

All title and intellectual property rights in and to the Software (including but not limited to binary code, source code, images, photographs, and text incorporated into the Software), the accompanying printed materials supplied by Barcodesoft, and any copies of the Software are owned by Barcodesoft. If this Software contains documentation that is provided only in electronic form, you may print one copy of such electronic documentation. You may not copy the printed materials accompanying the Software. All rights not expressly granted are reserved by Barcodesoft.

Notwithstanding the foregoing, Recipient shall retain ownership of any material of any nature it develops utilizing the Software, except with respect to the Software itself.

3. Termination

This Agreement is effective until terminated. This Agreement will be terminated automatically without notice from Barcodesoft if you fail to comply with any provision contained here. Upon

termination, you must destroy the written materials, the Software, and all copies of them, in part and in whole, including modified copies, if any.

This agreement will also be terminated with written notice from Recipient one week prior to annual fee charge date.

4. Representations, Warranties and Covenants

Barcodesoft hereby represents and warrants that the Software provided to Recipient therein are original works of authorship of Barcodesoft which are free and clear of any claims or liens by a third party, and represents and warrants that it has and shall maintain all right, title and interest necessary to confer the intellectual property grants described in this License Agreement. Further, Barcodesoft represents and warrants that it has the authority to enter into this License Agreement, and that Recipient's proper exercise of the rights granted herein shall not infringe upon the rights of any third party.

5. Limitation of Liability

Notwithstanding any damages that recipient might incur for any reason whatsoever, the entire liability of Barcodesoft and any of its suppliers under any provision of this agreement shall be limited to the greater of the amount actually paid by you or US \$5.00. The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails its essential purpose.

Neither Recipient, nor any of its affiliates, agents or assigns, nor the directors, officers, employees, or other representatives of any of them, is liable for any type or amount of damages arising out of or in connection with the use of the Software. This is a comprehensive limitation of liability that applies to all damages of any kind, including without limitation, compensatory, direct, indirect, consequential, incidental, special, exemplary, or punitive damages; loss of data, income, or profit; loss of or damage to property; and any and all claims of third parties.

6. Entire Agreement.

This Agreement constitutes the complete and exclusive agreement between Barcodesoft and Recipient with respect to the subject matter hereof, and supersedes all prior or contemporaneous oral or written communications, proposals, representations, understandings, or agreements not specifically incorporated herein. If any provision of this Agreement is held to be void, invalid, unenforceable or illegal, the other provisions shall continue in full force and effect.

7. Refund Policy

In the event a Recipient is not satisfied with a Barcodesoft product within 30 days of purchase, a refund may be issued. Issuing a refund is at the sole discretion of Barcodesoft. Refunds will be issued only after the Recipient has communicated with Barcodesoft to resolve issues or problems related to the original content purchased. If a refund is issued, the Recipient is responsible for deleting all files using the content and may not continue distribution of products containing the content or any part of the Software.

Glossary

Error Correction	Two-dimensional barcode employed Reed-Solomon algorithm for detecting and correcting multiple random symbol errors. The higher security level, the less decoding error chance. The tradeoff is barcode size.
Schema	Data types are a very important component of web services because different data types may be used to compose SOAP messages. WSDL uses XML Schema to describe data types used in a web service.
SOAP	Simple Object Access Protocol (SOAP) is one standard that formally sets the conventions governing the format and processing rules of XML messages and how they are used with HTTP.
Token	Barcodesoft Webservice requires a unique token for each user to call webmethod. If you leave it blank, returned barcode image will have a "Demo" watermark.
WSDL	WSDL formally sets the format for describing a web service using XML. By providing a Uniform Resource Identifier (URI) to a WSDL, applications can discover what operations a particular web service provides and how an operation's messages look. Barcodesoft webservice is described at the following URL http://www.barcodesoft.com/bcdgen.asmx?WSDL
XML	Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.

Index

E

Error Correction Level, page 35

G

PHP, Generate Barcode in PHP, page 29

C#, Generate Barcode in C#, page 32

S

XML Schema, page 1

SOAP, page 1

W

WSDL, page 1

X

XML, page 1