



## **Creating Intelligent Templates™**

Using dynamic design techniques to print multiple outputs using a single document

# Contents

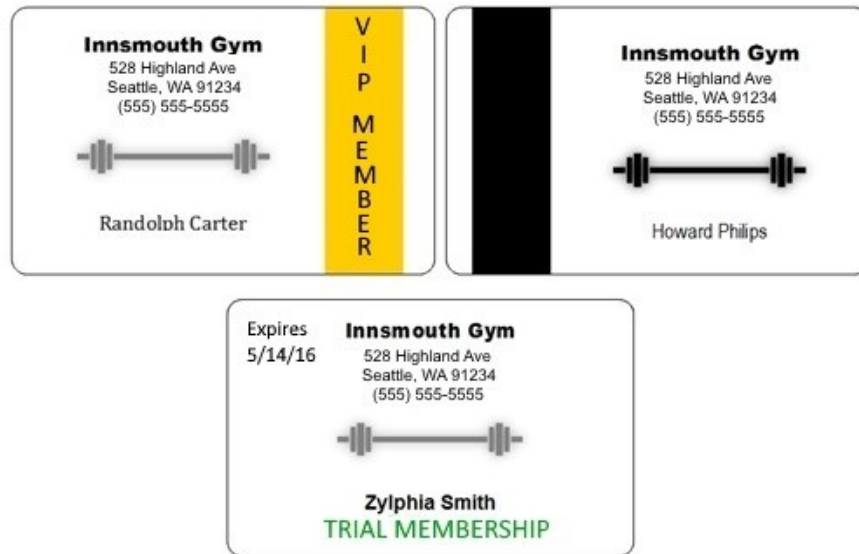
---

- Overview** ..... **3**
  - Dynamic Design Techniques ..... 3
- Conditional Printing** ..... **4**
  - About Conditional Printing ..... 4
  - Conditionally Printing Objects ..... 4
  - Conditionally Printing Layers ..... 5
  - Conditionally Printing Templates ..... 7
- Visual Basic Script** ..... **10**
  - Document-Level Events ..... 10
  - Changing Object Properties with VBScript ..... 11
  - Support for Data-Sourced VBScript ..... 12
- Auto Sizing** ..... **13**
  - Auto Sizing Text Objects ..... 13
  - Auto Sizing Barcodes ..... 14
  - Auto Sizing Picture Objects ..... 14
- Suppression** ..... **16**
- Appendix A: Task Reference** ..... **17**
- Related Documentation** ..... **18**

## Overview

As complex and variable designs become increasingly popular, so does the need for dynamic design techniques. BarTender's ability to create Intelligent Templates™ enables companies to produce a wide variety of flexible label designs without the need to create and maintain hundreds of separate documents. This allows you to vary the image, color, object position and text on each item that you print.

The following membership cards were generated with one Intelligent Template using a variety of dynamic design techniques, which will be explained in more detail throughout this white paper.



## Dynamic Design Techniques

There are a number of techniques available for creating a dynamic design:

- **Conditional Printing** allows you set "conditions" that define when objects, layers or even entire templates will be printed.
- Custom **Visual Basic Scripts** execute in response to certain document-level events. With VBScript, you can alter any object's appearance by changing its color, font, position, size and/or border.
- **Auto-Sizing** text and barcode objects forces the object to automatically fit a designated space on your template.
- **Suppression** automatically hides text on your template, based on conditions you define.

## Conditional Printing

Conditional printing allows you to define exactly when different objects, layers or templates in your document will print. For example, you can set a graphic to conditionally print under one condition, but not under any others. Or you can choose to print a template based on a value in your database. You can customize printing for as much or as little of your design as you want.

### About Conditional Printing

Conditional printing is based around conditional operators. For example, your conditions for printing might be based on whether the conditionalized item **contains** or **does not contain** certain text, numeric values, or images, or whether the conditionalized item is **less than**, **greater than**, or **equals** a set value.

You can apply conditional printing to objects, layers or templates, depending on how much of your design you want to change at print-time.

### Enabling Conditional Printing

Conditional printing is enabled from the **When to Print** dialog in BarTender. This lets you define a conditional statement based on a database field or named data source value to determine when the object, template or layer will print. For more information, refer to the following topics in the BarTender help system:

- [Overview of Conditional Printing](#)
- [Building Conditional Expressions](#)

### Conditionally Printing Objects

When your design changes only a single object between multiple items, you should consider setting up conditional printing for that one object. This lets you print (or not print) one object based on a condition that you define. Conditionally printing an *object* allows you to define specific conditions under which an object will appear on your printed item. This technique is useful when you need different objects to appear in variations of the same template.

Any object can be conditionalized to print when you want it to, including:

- Barcodes
- Encoder objects
- Text objects
- Line objects
- Shape objects
- Pictures

## Example

In the example below, the document is connected to a database containing a list of conference attendees and their statuses. Using the **When to Print** dialog, the bottom text object is set to conditionally print based on whether or not the attendee has a status of "VIP". Note that the position, formatting and visibility of all other objects on the name tag remain the same, regardless of whether or not the status is printed.



## Conditionally Printing Layers

When you have a complex design where multiple objects are the same between all designs, but a number of others change, you should consider using conditional layers. A *layer* is an object or group of objects that occupy a particular plane, and which can be stacked on top of each other.

### How Layers Work in BarTender

In BarTender, layers may contain one or more text objects, images, barcodes, shapes and/or lines included on the same plane. If you need to dynamically print multiple objects on your template at different times, you can put objects on different layers, and conditionalize the layers to print when certain conditions are met. You use each layer's **When to Print** dialog to determine when that layer will print, just as you would when conditionalizing an object.

## Example

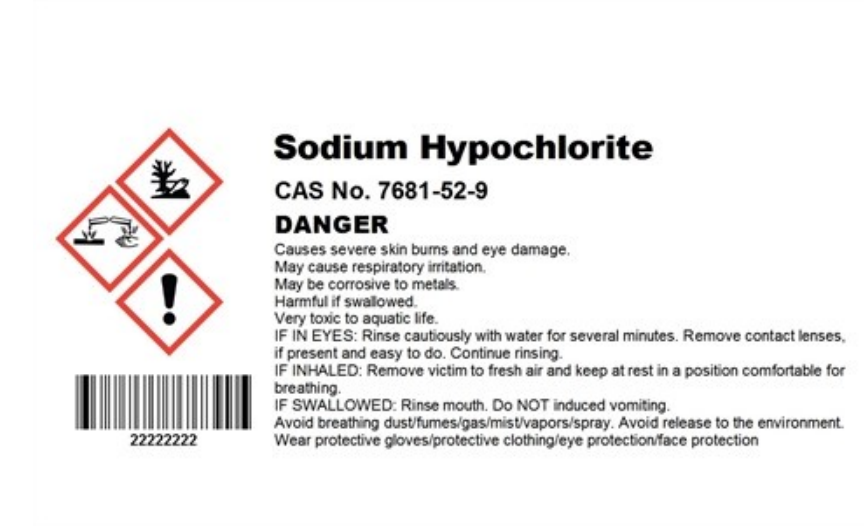
A base layer will contain all of the objects that remain the same on all of your designs:



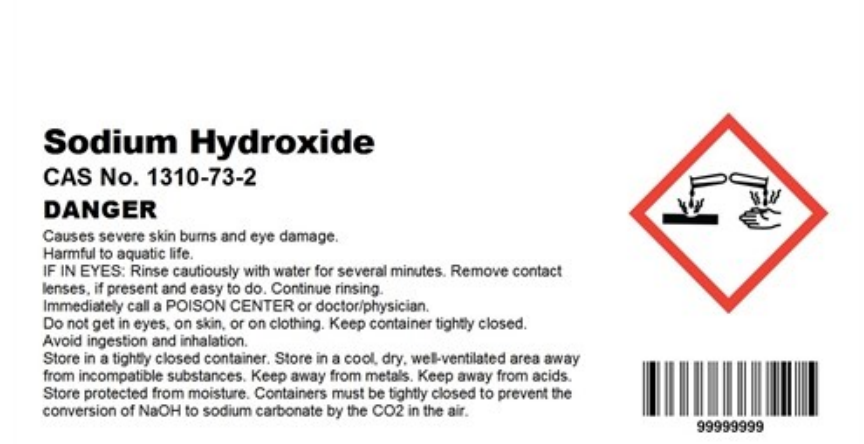


Chlor-Alkali Division  
54300 South 226th Street  
Kent, WA 98030  
www.xyzchem.com  
+1 360 555 1234

In case of emergency, call CHEMTREC day or night 800 424 9300



Each subsequent layer will contain the objects that will differ on each of your objects. Each layer can contain one or more objects:



**Sodium Hypochlorite**  
CAS No. 7681-52-9  
**DANGER**  
Causes severe skin burns and eye damage.  
May cause respiratory irritation.  
May be corrosive to metals.  
Harmful if swallowed.  
Very toxic to aquatic life.  
IF IN EYES: Rinse cautiously with water for several minutes. Remove contact lenses, if present and easy to do. Continue rinsing.  
IF INHALED: Remove victim to fresh air and keep at rest in a position comfortable for breathing.  
IF SWALLOWED: Rinse mouth. Do NOT induce vomiting.  
Avoid breathing dust/fumes/gas/mist/vapors/spray. Avoid release to the environment.  
Wear protective gloves/protective clothing/eye protection/face protection



**Sodium Hydroxide**  
CAS No. 1310-73-2  
**DANGER**  
Causes severe skin burns and eye damage.  
Harmful to aquatic life.  
IF IN EYES: Rinse cautiously with water for several minutes. Remove contact lenses, if present and easy to do. Continue rinsing.  
Immediately call a POISON CENTER or doctor/physician.  
Do not get in eyes, on skin, or on clothing. Keep container tightly closed.  
Avoid ingestion and inhalation.  
Store in a tightly closed container. Store in a cool, dry, well-ventilated area away from incompatible substances. Keep away from metals. Keep away from acids.  
Store protected from moisture. Containers must be tightly closed to prevent the conversion of NaOH to sodium carbonate by the CO<sub>2</sub> in the air.



Once you conditionalize each of the subsequent layers using the **When to Print** dialog, it will print on top of the base layer based on the conditions that you set. The final print output for the example above would be:



Chlor-Alkali Division  
54300 South 226th Street  
Kent, WA 98030  
www.xyzchem.com  
+1 360 555 1234



## Sodium Hypochlorite

CAS No. 7681-52-9

### DANGER

Causes severe skin burns and eye damage.  
May cause respiratory irritation.  
May be corrosive to metals.  
Harmful if swallowed.  
Very toxic to aquatic life.  
IF IN EYES: Rinse cautiously with water for several minutes. Remove contact lenses, if present and easy to do. Continue rinsing.  
IF INHALED: Remove victim to fresh air and keep at rest in a position comfortable for breathing.  
IF SWALLOWED: Rinse mouth. Do NOT induce vomiting.  
Avoid breathing dust/fumes/gas/mist/vapors/spray. Avoid release to the environment.  
Wear protective gloves/protective clothing/eye protection/face protection



22222222

In case of emergency, call CHEMTREC day or night 800 424 9300



Chlor-Alkali Division  
54300 South 226th Street  
Kent, WA 98030  
www.xyzchem.com  
+1 360 555 1234

## Sodium Hydroxide

CAS No. 1310-73-2

### DANGER

Causes severe skin burns and eye damage.  
Harmful to aquatic life.  
IF IN EYES: Rinse cautiously with water for several minutes. Remove contact lenses, if present and easy to do. Continue rinsing.  
Immediately call a POISON CENTER or doctor/physician.  
Do not get in eyes, on skin, or on clothing. Keep container tightly closed.  
Avoid ingestion and inhalation.  
Store in a tightly closed container. Store in a cool, dry, well-ventilated area away from incompatible substances. Keep away from metals. Keep away from acids.  
Store protected from moisture. Containers must be tightly closed to prevent the conversion of NaOH to sodium carbonate by the CO<sub>2</sub> in the air.



99999999

In case of emergency, call CHEMTREC day or night 800 424 9300

For more information about layers, refer to the following topics in the BarTender help system:

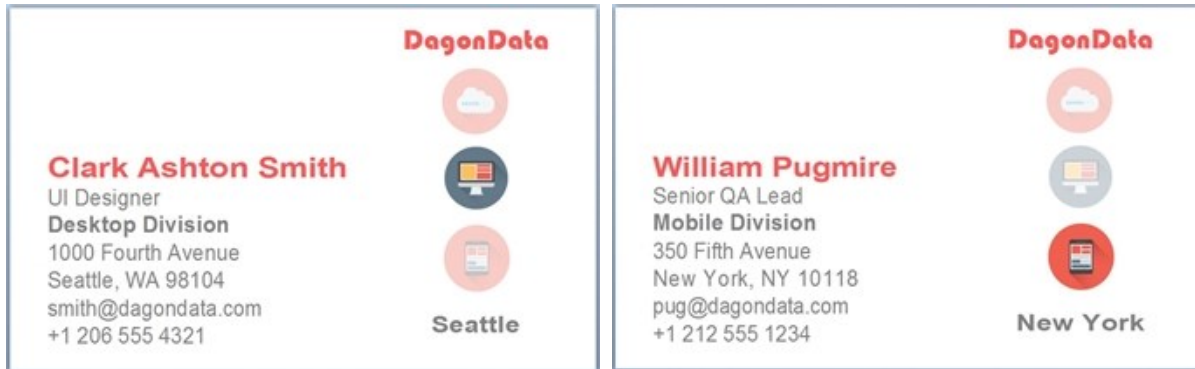
- [Overview of Layers](#)
- [Layer Properties Dialog](#)
- [Arranging Layers on a Template](#)

## Conditionally Printing Templates

When a single piece of information can radically alter the entire layout of the document you want to print, creating two or more entirely *different* templates for the same document may be the best option. You can configure templates to print using either conditional printing or template selectors.

Consider the following example: Your company has two offices, one in Seattle, another in New York. All of the employees are listed in the same database, and you need to print business cards for all of the employees. The office in Seattle has one design for their business cards, while the office in New

York has a completely different design. Each design is saved as a separate template within the same BarTender document.

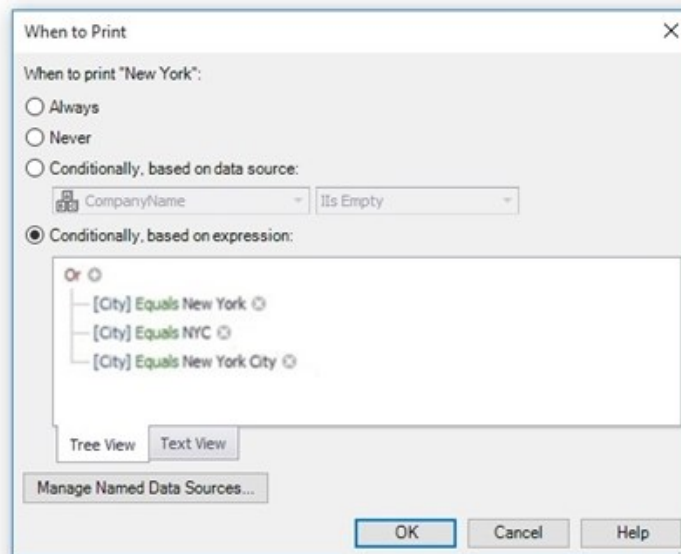


The following examples describe how to configure conditional printing for this scenario using both conditional printing and template selectors.

### Example: Conditional Printing of Templates

Conditional printing lets you define multiple conditional expressions for each template, and gives you a large amount of flexibility when defining print conditions. Consider that your database has a field called "City"; in it, and you have multiple different spellings for the same city (New York, New York City, NYC, etc.) To address the discrepancies in terminology, you can define multiple conditional statements, letting you print items for all variations of New York City.

From the [When to Print](#) dialog, define your conditional statements, telling the document to print when it encounters "New York", "NYC", or "New York City" in the "City" field of the database.



Repeat the procedure for the Seattle template, telling the template to conditionally print when the "City" field contains the values "Seattle" or "SEA".



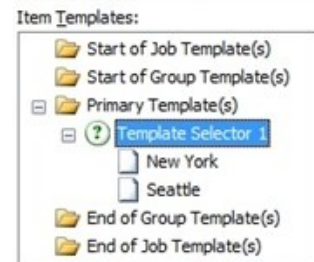
With this design, if the value BarTender reads from the database is "New York", "NYC", or "New York City", the New York template will print. If the database value is "Seattle" or "SEA", then the Seattle template will print.

For more information, refer to the following BarTender help topics:

- [Page Setup Dialog](#)
- [When to Print Dialog](#)

### Example: Using Template Selectors

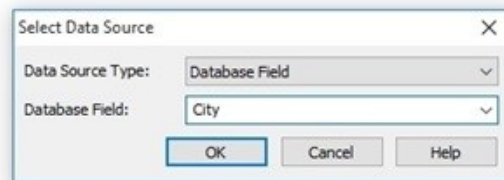
Template selectors are a simple way to choose between templates when you know all the *exact values* that could show up in your data source. When a template selector is in use, it will determine which template to print based upon the content of the data source; only when the data source in question matches the exact name of the template will that template print. Let's say that the only values for "City" that exist in the database are "Seattle" and "New York", and your templates have matching names.



From the **Templates** tab of the **Page Setup** dialog, add a new template selector. Drag both the Seattle and New York templates to the template selector.

Specify the value you want to match the template name. In our business card example, we will select the database field for "City".

With this design, if the value BarTender reads from the database is "Seattle," then the Seattle template will print. If the database value is "New York," then the New York template will print. Note that the data source value must match the template name *exactly* for the template selector to function.



For more information about template selectors, refer to the [Using Template Selectors](#) topic in the BarTender help system.

## Visual Basic Script

Visual Basic Script (VBScript) is a very flexible language that you can use to programmatically modify objects at different times throughout the document design and print process.

You can use VBScript to change object properties such as size, location and color, in combination with any other variable(s) you have the ability to script. For example, you could change an object's color based on the time of day, or change an object's size based on the color of another object. VBScript is perhaps the most dynamic template design tool of all, because with the appropriate programming skills, you can accomplish almost anything you want within your BarTender document.

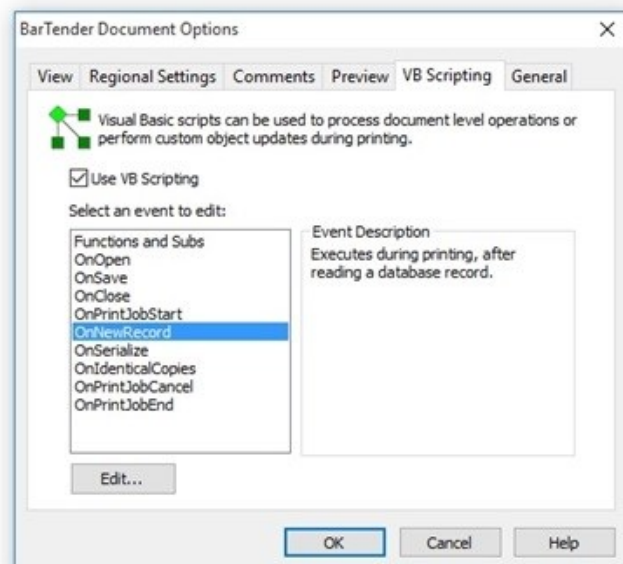
**NOTE:** Although Visual Basic scripts can be complex enough to be considered "programs," there are times when just a single line of VBScript can prove quite useful. Still, scripting is typically a technique for "power users" and not ideal for computer novices. Whatever your computer skills, you should consider whether BarTender doesn't already offer a solution to your challenge before embarking upon a script-based solution. For more information, refer to the [Visual Basic Scripting](#) book in the BarTender help system.

For a complete listing of object properties you can dynamically change with VBScript, refer to the [Object Properties Dialog](#) topic in the BarTender help system.

## Document-Level Events

Modifying a BarTender document's design and objects is *only* available in document-level event scripts. There are a number of document-level events that you can control with VBScript, including when you open or close a document, execute a print job, read a database record and much more. To write scripts for these events, open the [Visual Basic Script Editor](#).

For a complete listing of document-level events, refer to the [Script Pane](#) topic in the BarTender help system.



## Changing Object Properties with VBScript

Let's look at some of examples of changing object properties with VBScript.

### Example 1

A common use case is to change an object's color in response to the data coming from a database.

For this example, let's say you want to make an object's text color gold, silver or bronze, depending on membership level. You would enter script similar to this one in the **Editor** (center) pane of the **Visual Basic Script Editor**:

```
Set Notice = Objects("Text 1")
If (Field("MembershipType") = "Gold") Then
    Notice.TextColor = btColor.Gold
Else
    If (Field("MembershipType") = "Silver") Then
        Notice.TextColor = btColor.Silver
    Else
        If (Field("MembershipType") = "Bronze") Then
            Notice.TextColor = btColor.DarkGoldenrod
        Else
            Notice.TextColor = btColor.Black
        End If
    End If
End If
```

### Example 2

Alternatively, your database may have a column that contains the name of a color. In this case, you can use VBScript to set the color of an object, reading the color directly from the database. Your script might look something like this:

```
ReferenceField("ConditionColor")
Objects("Box 1").FillColor = Eval(Field("ConditionColor"))
```

Then, the "ConditionColor" field in your database could contain values like "BtColor.Red" or "BtColor.FromRGB(255,0,0)".

### Example 3

You can use VBScript to change the properties of or switch between linked objects as well as embedded objects. In this example, you will use VBScript to switch between two linked image objects based on conditional values:

```
ReferenceField("Image")
    Set ImageObject = Objects("Picture 1")
If Field("Image") = "Black" Then
    ImageObject.PicturePath = "C:\vbscriptimagepath\images\BarTender_Black.png"
Else
    ImageObject.PicturePath = "C:\vbscriptimagepath\images\BarTender_RGB.png"
End If
```

#### Example 4

You can also hide or show objects based on the data that is read in from a database or named data source. You can either use an object's PrintVisibility property, or you can simply move it on or off the design area, using its X and Y properties. For instance, if you wanted a line to appear only if a database field called "Location" equals "Seattle," your code would look something like this:

```
Set Notice = Objects("Line 1")
If (Field("Location") = ("Seattle")) Then
    PrintVisibility = True
Else
    PrintVisibility = False
End If
```

#### ***Support for Data-Sourced VBScript***

You can feed VBScript into BarTender from external sources, such as a database or a program that is controlling BarTender. For example, you could have a field in a database contain VBScript code. Then, when BarTender reads the data record that contains that field, VBScript code in BarTender can use the VB "Execute" statement or the "Eval" function to execute the imported VBScript.

**NOTE:** This is an advanced technique, and should be used only by those with extensive VBScript experience, or with the help of a BarTender reseller or other consultant.

## Auto Sizing

"Auto sizing" refers to BarTender's ability to automatically adjust the size of objects on your template based on the data contained in the object. By enabling auto sizing for text objects, you allow BarTender to change the font size and/or scale of text objects or human readable text to fit the available space on the template. For example, auto sizing will allow you to print a 3-line address in the same template field as a two-line address by shrinking the text.

For barcodes, BarTender will change the X dimension of the symbol to fit a defined space.

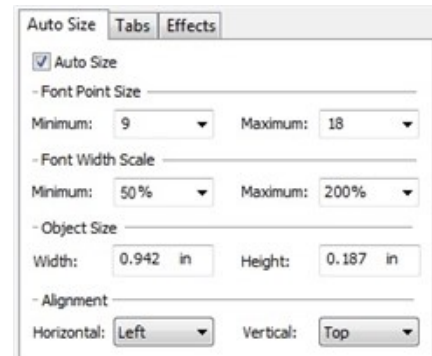
### Auto Sizing Text Objects

Auto sizing configures the size of the text to fit into the text object you have specified on the template. The text object itself stays the same size; only the text inside it is adjusted. This is accomplished by setting minimum and maximum font point sizes, determining the font width scale, and defining a specified object size.

#### Example

Suppose you are creating mailing labels. Certain fields might contain more address information than others. For instance, some customer names and company names might be longer than others, which would make the text larger than would normally fit onto your label. By auto sizing the text objects, you ensure that your text objects will always fit onto your page.

If you applied auto sizing to the "Customer Name" and "Company Name" fields of your label, the text size would adjust to fit the available space:



<b>Sam Brook</b> Beidentoupe Industries Inc. 2456 SE Willow Street Seattle WA 88888	<b>Jan W. Barleycorn</b> <b>The Dog Zone</b> 323 NW James Street Seattle WA 88889
<b>Crystal Woo</b> <b>Crystal's Candy</b> 764 53rd Place SW Seattle WA 88880	<b>Pat Persephone Marley</b> <b>Zoom Science for Kids</b> 67866 SW Calendar Street Seattle WA 88881

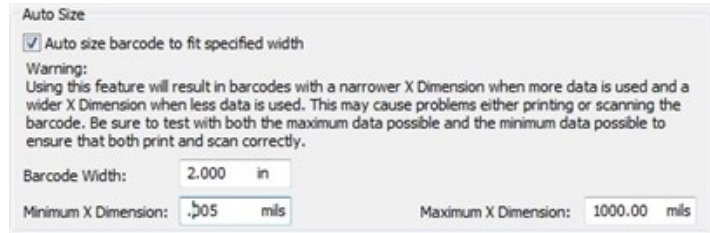
## Auto Sizing Barcodes

Auto sizing for barcodes applies both to the barcode itself and the accompanying human readable characters so that regardless of the length of the data source, your barcode will maintain a consistent size. Just as text objects adjust font size, spacing, and width in order to maintain the desired size, barcodes adjust the X dimension to maintain the desired size.

**NOTE:** The desired auto size listed is the maximum size, so the barcode might be smaller depending on the data source. Because auto sizing can affect the density (and consequently, the readability) of data, it is important to test your scanner's ability to read barcodes at both your minimum and maximum settings.

### Example

Suppose you are printing labels, and each label has a specified area where one of four barcodes will appear. The barcodes contain varying amounts of information and human readable characters, but you need them all to fit within the designated area on the template.



The results will look something like this, depending on the barcode type and the connected data source:

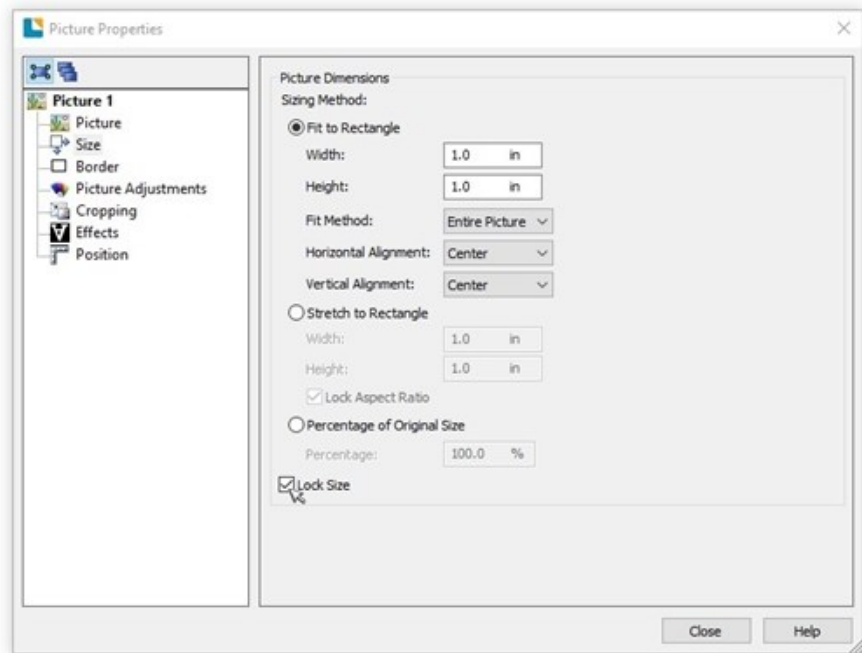


## Auto Sizing Picture Objects

You can set picture objects to automatically fit a specific area on the template. This is accomplished from the object's **Properties** dialog.

### Example

Suppose you are creating labels to mark sale items. There is a specified area on the template where the image of the current sale item will appear. You have multiple images that can appear on the label, and not all of the images are the same size. You need all images that appear on the template to be printed at the same size.



Once you have locked the size of the object, any image you source into that object will appear at the size you selected.

## Suppression

You can use BarTender's suppression feature to conditionally print data sources. Unlike the conditional printing methods discussed up to this point, data sources are suppressed, or hidden, when a condition is met.

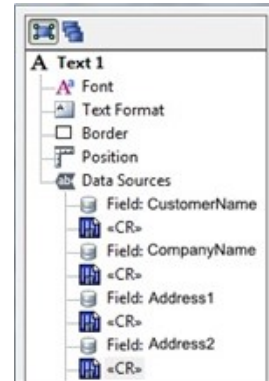
A common use is on multi-line text objects where some of the lines may not contain data. For example, some addresses contain one address line, while others contain two. When printing address labels, you may want to have BarTender automatically suppress the second address line when it is not present in the address, so it doesn't appear as a blank space.

For more information on databases, refer to the [Reading Data from Databases](#) topic in the BarTender help system.

### Example

Suppose you have a multi-line text object containing multiple data sources for an address label. The data source fields are called CustomerName, CompanyName, Address1, Address2, City, State, and Country. Sometimes there is no data in the "Address 2" field. You want to eliminate the blank line that appears on the label if the field "Address 2" is empty.

You have added carriage returns after each data source to set the desired line breaks within the text object. From the "Address 2" text object's [Properties](#) dialog, apply the **Suppression** transform to the "Address 2" carriage return to conditionally suppress the line above if it contains no data.



This will result in your address labels printing without a gap.



For more information on suppression, refer to the [Suppression Dialog](#) topic in the BarTender help system.



## Appendix A: Task Reference

Refer to the table below for some common dynamic design tasks, and the suggested method(s) for accomplishing them.

What do you want to do?	Suggested Method(s)
Show or hide a single object based on data	<a href="#">Conditional objects</a>
Show or hide multiple objects based on data	<a href="#">Conditional layers</a>
Change the color of an object based on data	<a href="#">Conditional layers</a> <a href="#">VBScript</a>
Change font size of an object based on data	<a href="#">Auto-size text</a>
Suppress (hide) the contents of a selected data source under certain conditions	<a href="#">Suppression</a>
Change the size of a barcode object based on data	<a href="#">Auto-size barcodes</a>
Change position of an object based on the size (or other qualities) of another object	<a href="#">VBScript</a>
Change part of a template design based on data	<a href="#">Conditional layers</a>
Change the entire template based on data	<a href="#">Conditional templates</a> <a href="#">Template Selectors</a>

## Related Documentation

### White Papers and Manuals

- What's New in BarTender 2016
- Printing Colors Dynamically

For downloadable versions, visit:

<http://www.seagullscientific.com/support/white-papers>

### BarTender Help System

- [Building Conditional Expressions](#)
- [Overview of Conditional Printing](#)
- [Overview of Layers](#)
- [Suppression Dialog](#)
- [Using Template Selectors](#)
- [Visual Basic Scripting](#)
- [Reading Data From Databases](#)

