Rajesh Lal

Creating Vista® Gadgets

Using HTML, CSS, and JavaScript with Examples in RSS, Ajax, ActiveX (COM), and Silverlight™

SAMS

Creating Vista® Gadgets: Using HTML, CSS and JavaScript with Examples in RSS, AJAX, ActiveX (COM) and Silverlight™

ISBN-13: 978-0-672-32968-5
ISBN-10: 0-672-32968-9
Library of Congress Cataloging-in-Publication Data

Lal, Rajesh.
Creating Vista gadgets : using html, css and javascript with examples in rss, ajax, activex (com) and silverlight / Rajesh Lal.
p. cm.
ISBN 978-0-672-32968-5 (pbk.)
1. Microsoft Windows (Computer file) 2. Operating systems (Computers) 3. User interfaces (Computer systems) I. Title.
QA76.76.O63L3546 2008
005.4'46--dc22
2008016280
Printed in the United States of America
First Printing April 2008

Trademarks

Warning and Disclaimer

Bulk Sales

This Book Is Safari Enabled

Associate Publisher
Greg Wiegand

Acquisitions Editor
Loretta Yates

Development Editor
Todd Brakke

Managing Editor
Patrick Kanouse

Project Editor
Jennifer Gallant

Copy Editor
Margo Catts

Indexer
Ken Johnson

Proofreader
Mike Henry

Technical Editor
Marc Clifton

Publishing Coordinator
Cindy Teeters

Book Designer
Anne Jones

Introduction

With the broadest ever worldwide release of a PC operating system, in 2007 Windows Vista opened the door to an era of gadget development. Gadgets, which reside on the Windows Vista Sidebar, are small, lightweight, and can be very useful applications. The Sidebar is a brand new platform for innovation and it gives users a unique way to interact with information.

This book is for people who want to create feature-rich and professional-looking Vista Sidebar gadgets. It's a guide for designers, developers, and anyone else who has a basic knowledge of HTML, CSS, and JavaScript and wants to leverage this new and innovative platform. It's for anyone who wants to create a gadget for his company, or for a programmer with a great idea to implement on a Sidebar gadget platform, or even for a hobbyist programmer, who wants to try his hand on a gadget platform. This book is intended to give you ideas for *what* you can do with this new platform and *how* you can do it.

What's in the Book

This book starts with a brief background on gadgets, and then gives a broad and clear view of the architecture of gadget development. Gadget design considerations are an important part of this book and they go side by side with almost all the chapters that deal with gadget development. Once you've read up on the concept and scope of gadget development, the book helps you create a gadget called MyBlog. During this process the text elaborates on the architecture, design constraints, and implementation details for the gadget and then details some standard practices applicable to all gadget development. The last section deals with more advanced gadget examples that utilize .NET, XML, XHTML, CSS, Ajax, and Microsoft Silverlight.

This book is divided into three broad sections.

Section 1: The Foundation

The four chapters in this first section give a thorough background of Sidebar gadgets. The section explains the types of gadgets, the architecture, and the technology behind the

gadget development. The "Approach to Design" chapter helps you know the difference between a merely good-looking gadget and a one that is professional, rich, and worth the space it takes up on the user's desktop. The last chapter discusses the revenue model of the gadget: what you need to know to sell your gadgets.

Section 2: Developing a Gadget

This section walks you through the standard development process of a gadget. It details the creation of the basic MyBlog Gadget, which makes use of an RSS/Atom feed. It also goes through best practices with the user interface, design guidelines, and common assumptions. The later chapters improve on the basic gadget based on standard practices and also deal with deploying and distributing a gadget.

Section 3: Advanced Samples

The section deals with advanced samples. You will be able to create advanced gadgets such as a Site Statistics Gadget, a Radio Gadget, and a YouTube Video Gadget. All the samples follow standard patterns, making it easier to switch between the features and functionality you want, when you want. You will also learn how to use ActiveX COM for creating a utility gadget with a sample .Net Most Recent Used (MRU) Gadget. The final chapter shows you how to create a gadget with Microsoft Silverlight. You will also see how, in just a few minutes, to create a Sidebar gadget with Microsoft Popfly.

If you are a relatively new gadget developer, I would suggest you to start with the first section. If you have basic background knowledge of gadgets and you just want to start with the step-by-step practical approach to gadget development, you can directly start with the second section, "Developing a Gadget." The third section, "Advanced Samples," is for people who have developed a gadget and want to go beyond the basics of gadget development. Each chapter in the third section is actually an advanced sample dealing with a particular type of gadget in a scenario of its own.

Special Features and Notations

This book is meant to be a definite, precise, and concrete guide for gadget development. By pruning redundant information and filtering and highlighting the information that is more crucial, we have tried to make it as comprehensive as possible. This book includes various features and conventions that help you get the most out of the book.

HTML, CSS, and JavaScript code blocks will be shown as follows:

```
<HTML>
Code in HTML, CSS, and JavaScript
</HTML>
```

Sample single code lines will look like this:

```
Statement one;

Statement two;
```

Other comments will also show up in the code with two backslashes

```
// Comment one
// Comment two
```

The book also uses the following boxes for important information:

NOTE

A Note includes extra information to broaden your understanding of a topic.

TIP

A Tip provides alternative, shortcuts, or insider information of the topic being discussed.

CAUTION

A Caution warns you of potential traps and pitfalls.

Supporting Website

The book has a supporting website where you can download all the codes and gadgets. The website also has blogs I have written on Sidebar gadgets and some of my personal views on gadget development. You are invited to check that site and contact me personally. You will also find errata and most updated information there.

Visit www.innovatewithgadgets.com

CHAPTER 3

An Approach to Design

IN THIS CHAPTER

- Design Considerations
- Challenges for the User Interface
- Visual Themes
- Transparent Images in the Gadget

"You know you've achieved perfection in design, not when you have nothing more to add, but when you have nothing more to take away."

—Antoine de Saint-Exupery

Design Considerations

This chapter is about gadget design and user interface. Design starts with the factors that determine the type of gadget you want to develop, the information it will display, the user interface, the usage pattern, and the behavior of a gadget.

Design includes the dimensions, the images, the text, and the "look and feel." You also decide how the gadget interacts with the user and how it interacts with the system. This chapter discusses the visual theme, how the gadget can look like a part of Windows Vista, and the overall user experience.

Before you start, keep these two things in mind:

- **Justify the space**—The Windows Vista Sidebar is neither very tall nor wide. It can have—at the most—five or six gadgets at any particular time. Thousands of other gadgets, freely available online, will compete for the same screen space. So, offering the set of features a user critically needs is an important factor. Be prepared to convince users that your gadget justifies the space.

- **Ensure overall quality**—If your users don't experience quality throughout your gadget, they may conclude there is a lack of quality everywhere. This means you need to pay attention to the quality of icons, images, text, background, and interaction. Each of these elements is equally important. A good idea and a great implementation with an average user interface cannot stand up to the competition (see Figure 3.1).

FIGURE 3.1 Two gadgets that perform the same function, one with a nice interface and the other plain and simple. Which one would you prefer?

Consider the following four factors before designing a gadget (see Figure 3.2):

- Information first
- The right user interface
- The gadget's usage pattern
- The gadget's behavior

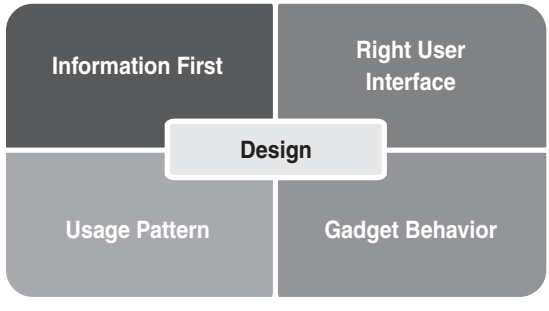

FIGURE 3.2 A gadget's design is an important aspect of development.

Putting Information First

Gadgets are meant for a single task. When designing a gadget, keep in mind that a gadget should have a small set of goals relevant to the specific task. A gadget should show only data suitable for that task and that task only. Information in the gadget window should satisfy the following rules:

- Live data
- Information for quick access
- User's choice
- Brief information for further action

Let's take a closer look at what each of these really means.

Live Data

Gadgets should display only information that changes regularly, such as live feeds, news updates, daily weather, battery status, and so on. Information that is static for more than a day makes a gadget dull.

A user looks to a gadget to see interesting things that are active. Fun gadgets are an exception but can be made more interesting if live data is also added. For example, a subset of a popular game is nice to have in a Sidebar, but it's better if there is live information, like a regularly updated scoreboard, or live updates about the game. If that is not possible, adding a capability to change the gadget's wallpaper (background image) is a good option.

Easy Access to Information

Gadgets are particularly useful at showing information for which users don't have to start an application or open a web page. A good example might be stock values, an event calendar, traffic maps, local fuel cost, and so forth. Any information that saves a user's time makes a gadget worth the space it takes.

Information Relevant to Individual Users

Gadgets are all about user choice and preference. If there are a lot of users for a particular newspaper's crossword puzzle, a gadget that taps into that theme can be an instant hit with them. User-tailored gadgets are very popular, such as a gadget for a Flickr website showing the user's own shared pictures or the user's favorite blog feed.

Information for Further Action

The gadget should show enough information for the user to decide on further action. For example, a website statistics gadget should not show each and every detail corresponding to website usage. It should show statistics for a week or a day and let users decide what further action they want to take.

Case Study: The Soapbox Video Gadget

Imagine you wanted to create a video gadget (see Figure 3.3). The task is to create a gadget for a video feed from http://soapbox.msn.com with the goal of accomplishing the following:

- Listing frequently updated videos from soapbox.msn.com in a simple and aesthetically pleasing way
- Playing a video in the flyout

A video feed normally contains thumbnails of the video, reviews, and ratings from users. A gadget with these goals gives you the following options for design decision:

- Making available the thumbnails and ratings for each video
- Adding capability to browse video by categories
- Functionality to search for videos by keyword
- Video resize option with support for Windows Media Player and Flash Player
- Easy page-wise access to list of videos

Figure 3.3 shows a preview of the Soapbox Video Gadget with all these options. More information about the gadget can be found at http://www.codeproject.com/KB/gadgets/SoapBoxGadget.aspx.

FIGURE 3.3 This fully developed gadget gives quick access to Soapbox video.

Try to make your gadget feature-rich by providing a complete set of functionality related to a specific task (see Figure 3.4). Think like a user. If the gadget shows a list of videos, the gadget should also categorize them and give easy access to all the items in the video feed.

Providing the complete set of functionality does not mean that the gadget should do everything the Soapbox video website is doing. Video sharing and review capabilities in the website may not be a part of the gadget. Complete functionality here means the gadget should give access to all the information from the video feed, which is the gadget's input. The users should be able to filter videos, sort them, search by keyword, and play videos in the way they want. In Chapter 11 we will see how to create a video gadget using YouTube Video feedback.

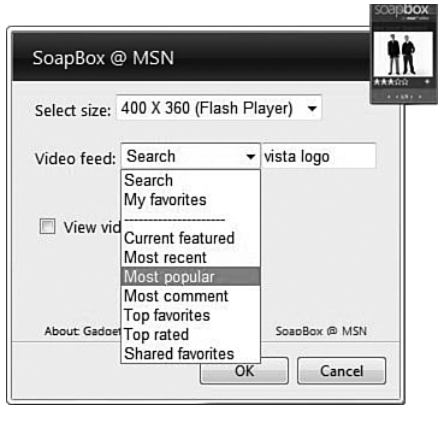

FIGURE 3.4 This gadget is able to categorize MSN Soapbox videos, giving users easy access to the one they want to see.

Refer to the checklist of guidelines for gadget development and compare that to this gadget. What live information does the gadget display?

- Frequently updated soapbox feed.
- Currently featured videos.
- Recent videos.

Does the gadget offer quick access to the information it provides?

- Ability to click on the thumbnail to play the video in the flyout.
- Ability to browse through the video list.

Does the gadget give the user a choice?

- Settings page gives an option to customize the list of videos.
- Users can select the media player and video size.

Does the gadget provide information that lets the user decide on further action?

- Gadget gives information about the video's ratings, a thumbnail image, and the title in the main window for users to decide to watch the video.
- Gadget gives Previous and Next options for browsing the videos in the feed.

Constructing the Right User Interface

Gadgets are a visual experience and the right user interface makes all the difference. Here are four pointers for creating one:

- Keep the gadget simple and aesthetically pleasing.
- Show only relevant information.
- Make use of visuals such as icons and images, rather than text.
- Be sure that the gadget is not too obtrusive.

Simplicity and Aesthetics

A gadget should look simple and aesthetically pleasing. Take a look at the gadget samples provided in Figure 3.5.

Clearly the objects in the first pair are both simpler and more aesthetically pleasing than the obnoxious clock and difficult-to-read note. If a gadget is going to be part of the desktop, it is something the user will look at every day; make sure it is as clear and aesthetic. The choice of color and fonts can also make a difference.

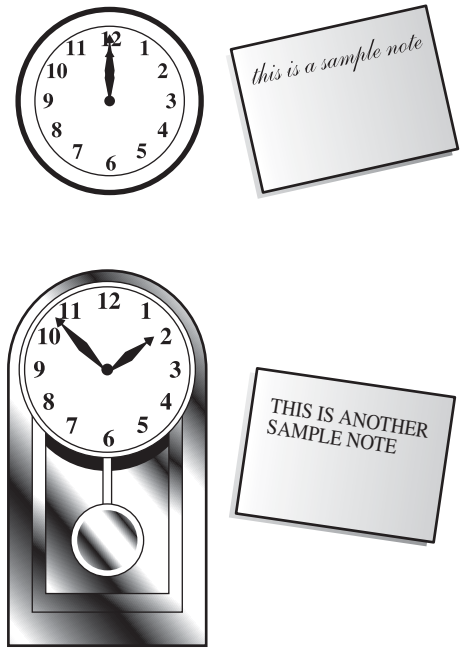

FIGURE 3.5 Choice of clear image and aesthetically pleasing fonts can make a lot of difference in gadget design.

Show Only Relevant Information

The maximum width of a gadget is 130 pixels. That's not a lot of room to work with, especially if there's a lot of information you want to convey. It can be done, however. Just look at the RSS feed and Calendar Gadgets shown in Figure 3.6.

FIGURE 3.6 The RSS feed Gadget shows the title partially in the gadget window, and the complete title is shown as the tooltip.

So, what is it exactly that makes these well-designed gadgets? For one, titles should generally be only as long as the space allows. With some gadgets, like this RSS news feeder, that

isn't possible. But in this case, tooltips are applied to good effect. And, although you can't see this clearly from the black-and-white photos in this book, the fonts have different colors to highlight the title.

The RSS Reader Gadget doesn't try to do too much. It displays only four records at a time, and the Calendar Gadget displays just the current month or the current day, based on the user's choice.

Figure 3.7 shows two more gadgets that are designed to accomplish the same goals, but suffer from an extremely poor design.

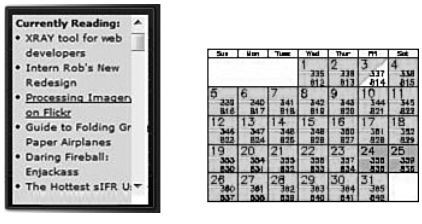

FIGURE 3.7 The gadget with scrollbars and the overfilled calendar both try to squeeze too much information into the small space.

The presence of a scrollbar in a gadget is unacceptable. It's far better to use paging because a scrollbar can further reduce the already small space a gadget provides. Providing paging functionality to browse multiple items with previous and next options and page numbers can remove the clutter from the gadget screen. Both the Blog and the Calendar Gadgets have far too much information cramped into a small space. That's the kind of design mistakes you should strive to avoid when designing your own gadgets.

Make Use of Visuals

Make use of icons, images, and signs as much as possible; they give visual clues of the functionality (see Figure 3.8). For example, a Weather Gadget can use pictures of clouds, the sun, and rain instead of corresponding text to depict different weather condition. The proper use of images makes the gadget more user friendly.

Two gadgets are compared in Figure 3.8. Both gadgets show the same information, using different designs. However, the designs in the top row have visuals that give a rich experience to the user. When designing a gadget, check for the following:

- Can any information displayed in the gadget can be replaced by visuals?
- Is the gadget too plain or does it lack design?

Visual themes are covered in more detail in later section.

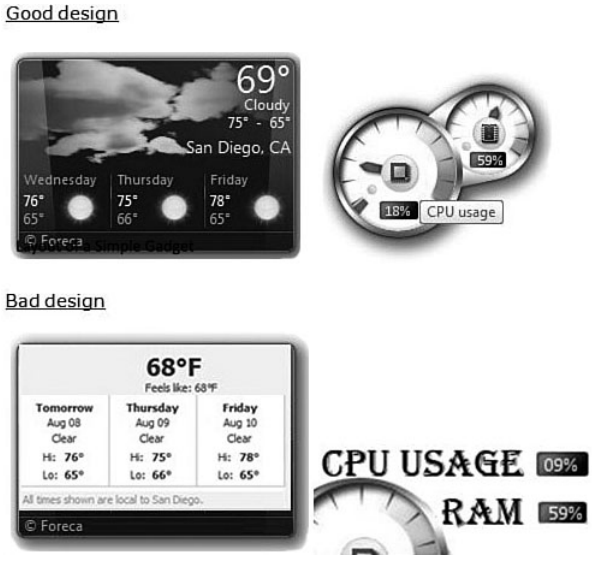

FIGURE 3.8 Pictures of clouds and sun in the Weather Gadget depict the weather. Computer usage in the form of a CPU meter is more intuitive than plain text.

Not Too Obtrusive

The design of the gadget should not be too obtrusive. The use of buttons and user controls should be avoided at all costs. For example, using a Previous and Next button can make the gadget look ugly. Instead, use proper images and show them when the mouse is moved over the gadget.

Figure 3.9 compares a Picture Slideshow Gadget with two different designs. The lower gadget shows buttons to browse either the previous image or the next one. The buttons are too obtrusive for a good design. The upper images are examples of a good design. The gadget's default view (upper left) is without any Previous or Next buttons. When the mouse is moved over the gadget, previous and next images are shown.

The Picture Slideshow Gadget shows the action buttons (images) on mouse hover. A good mouseover effect (refer to the upper-right image of Figure 3.9) with proper visuals makes a good design. Note that these are not buttons but images that are aligned with the gadget theme.

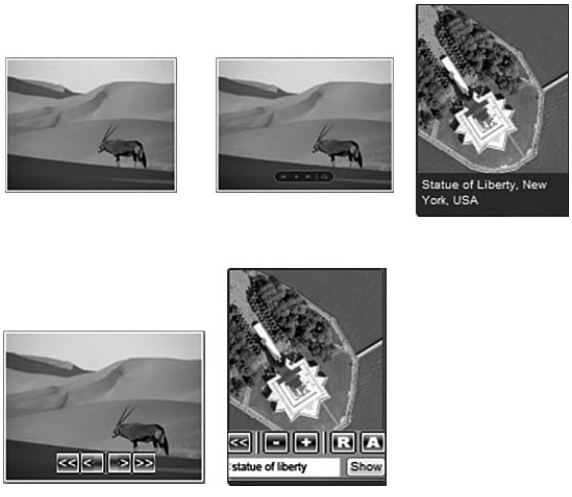

FIGURE 3.9 The always visible Previous and Next buttons are used to browse images in the Picture Slideshow Gadget (lower image), but they are too obtrusive for the small space.

Usage Patterns

The type of gadget you want to create also has an impact on its design. As discussed previously, there are four broad classifications:

- Information gadget
- Application gadget
- Utility gadget
- Fun gadget

These classifications all have common user interface guidelines, but each has its own specific design pattern that needs to be considered during development.

Information Gadget

Information gadgets collect data from multiple sources and are time sensitive. An information gadget normally uses RSS feeds that contain 10 or more items. To display them all in the gadget proper, page number and previous/next options should be available. The use of such options is also referred to as *paging*. These gadgets refresh their data regularly, so they should not be visually distracting or obtrusive (see Figure 3.10). As stated previously, there should be no scrollbars.

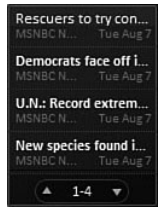

FIGURE 3.10 This example of an RSS Reader Gadget has paging options 1–4 and no distracting images. It reflects good gadget design.

Application Gadget

These gadgets depend on other applications or products for their data and act as a side product or a quick tool for data visualization. These gadgets should be designed with the main product or application in mind. The visual theme should go along with the original application (see Figure 3.11).

FIGURE 3.11 An application gadget mimics the user interface of the original application.

This is an example of a Microsoft Office Recent Documents gadget. The gadget shows the recently used Microsoft Office documents. The corresponding logo of Microsoft Office and icons for Microsoft Word, Excel, and PowerPoint make the gadget look rich and pleasing.

CAUTION

Please note that the gadget shown in Figure 3.11 is developed at Microsoft and it uses icons and images that are copyrighted by Microsoft and should not be used in publicly distributed gadgets without permission. Please check the "Use of Microsoft Copyright content" at https://www.microsoft.com/about/legal/permissions/default.mspx.

Utility Gadget

Utility gadgets provide quick information or shortcuts to frequently accessed tools and features. There should be no gimmicks in a utility gadget. The size should be appropriate: It should be the smallest of all other types of gadgets and should correspond to the feature it provides (see Figure 3.12).

FIGURE 3.12 A Battery Monitor Gadget displays percentage and time remaining in appropriate size.

This example shows a utility gadget that indicates the amount of battery life remaining in a mobile PC. The information it provides is the percentage of battery remaining and the time left. The background is an image of a battery with a percentage filled with color. This is a good design. There are no bells and whistles, but the size is appropriate and the design is intuitive.

Fun Gadgets

Fun gadgets are more distracting than other types of gadgets. As a result, users are likely to change them more frequently. Their purpose is to entertain or provide some fun activity to the user. If you are making a fun gadget, you must have a strong understanding of your target users and the gadget should have some dynamic features to keep the user interested for a longer period of time. It should also look visually pleasing (see Figure 3.13).

A visually pleasing experience comes with proper use of colors along with neat and clean images. Keep these quick tips in mind:

- Do not use too many colors; try to stick with two or three colors.
- Do not use more than one bright color in your gadget window.
- Check for good contrast colors or shades of the same color.

TIP

To get a better idea on different colors and contrast effects check http://www.colorsontheweb.com/.

FIGURE 3.13 This Minesweeper Gadget has a visually pleasing interface.

The example shown here is of a Minesweeper Gadget. It uses only a few colors: gray, black and red. It makes good use of contrast between red and black and the color also reflects the personality of the game.

What Gadgets Are Not Meant For

So far we have discussed what the different usage patterns are and what a gadget is meant for. This section gives you an idea what a gadget is not meant for. Keep this set of rules in mind while designing a gadget:

- Gadgets are not meant as a substitute for full applications such as email or instant messaging.
- Gadgets should not be designed as time-sensitive applications.
- Gadgets should contain no direct advertisements.

A gadget is not meant for notification purposes. Notice of new emails or instant messages should not be a gadget's purpose. Applications such as instant messaging and email notification require more robust applications such as MSN Messenger and Microsoft Outlook. A gadget is lightweight and is designed to supplement these applications rather than compete with them.

A gadget is also not meant for notification applications that need immediate attention because gadgets are not executables running in the user's desktop. They reside in the Sidebar and the user might have closed or hidden the Sidebar to avoid distraction. A gadget should not be a crucial application.

Gadgets can be used for advertisement purposes but are not meant to include banner advertisements or text ads. The small size of gadgets does not allow them. Check the next chapter to get details on the gadget's business model.

Gadget Behavior

The behavior relates to the way the gadget interacts with the user. How a gadget should react in particular circumstances, what a gadget is meant for, and what it is not meant for—all this decides the gadget's behavior.

You need to consider the following to ensure proper gadget behavior:

- Gadget configuration
- Refreshing a gadget
- Errors, information, and warnings
- Service not available information

Gadget Configuration

Most gadgets have an optional settings page that can be used to configure the gadget. You access it by clicking on the Option menu in the right-click context menu or by clicking the settings icon in the top-right corner of the gadget (see Figure 3.14).

Gadget configuration is the only option a user has to customize the gadget. It is a single administration page per instance of the gadget. Any configuration change in the settings page is applied to only that instance of the gadget. Choose wisely what changes a user

would like to have in the gadget, based on its functionality. For example, if the gadget allows, give a resize option to a mini version of the gadget with absolute essentials. This gives the user the opportunity to add more gadgets to the Sidebar, which means that your gadget has a better chance of being utilized.

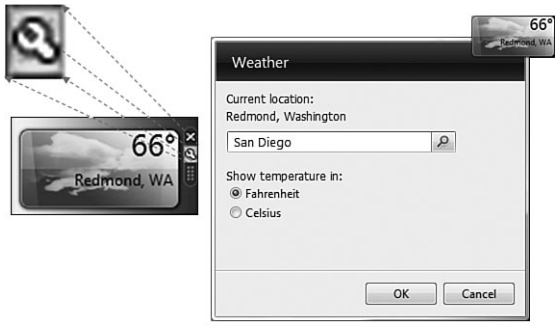

FIGURE 3.14 A settings page of a Weather Gadget enables the user to select a location, as well as choose Fahrenheit or Celsius to display temperature.

A settings page gives the user an option to customize the gadget according to choice. A configuration page gives more flexibility and freedom to the user and so increases a gadget's usability.

Refreshing a Gadget

If your gadget displays live data, you might need to refresh it regularly or on user demand. Keep in mind that if a user has a slow Internet connection, it can take some time for a gadget to reload. Using a loading image in the waiting screen with the message `Getting data...` or `loading...` is recommended (see Figures 3.15).

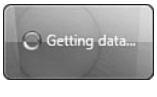

FIGURE 3.15 A Weather Gadget displays a Getting Data screen, along with a Vista busy cursor (image on the left) to display the status.

Errors, Information, and Warnings

The gadget sometimes needs to display status to the user. For example, an RSS Feed Gadget needs to tell the user when the feed is not available, or if there is no Internet connection available to fetch data from the remote server. These messages can also be custom error messages, warnings, or other information. All these status messages should be shown as inline text with standard 16×16 icons, like the ones shown in Figure 3.16, for the type of message. Pop-up dialog boxes are not allowed in a gadget.

FIGURE 3.16 Standard icons for errors, information, and warnings need to be used for the corresponding status.

A gadget is an HTML file with scripts, so make sure you have handled all the possible errors in the gadget. If there is an unexpected error or warning, display it in the same way as the Getting Data screen shown in Figure 3.15, or display the Service Not Available screen. If the error inside a gadget is not handled properly, a default runtime error message is displayed along with the line number.

JavaScript error messages, like the one shown in Figure 3.17, are annoying. One of these messages and the user will lose all faith in the gadget and won't hesitate to remove or even uninstall it. A good practice is to encapsulate each JavaScript function inside an error handler, a try-catch block.

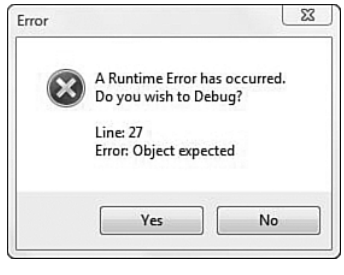

FIGURE 3.17 Typical JavaScript error messages, like this one, are too vague to be useful to the end user.

A try-catch block is a piece of code that ensures the execution of the catch block if any error occurs inside the try block. Here is the example in JavaScript. The code ensures that the user doesn't get a default JavaScript error dialog:

```
try
 {
  // your script code here
 }
 catch (e)
 {
   //set and display your inline error message and icon here
 }
```

Service Not Available Information

Service Not Available is a default screen that is used in most of the common scenarios. Use the Service Not Available screen where appropriate. For example, if the gadget is in a mobile laptop and the laptop disconnects from the Internet, show a screen with the information icon like the one shown in Figure 3.18.

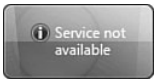

FIGURE 3.18 A Weather Gadget shows a Service Not Available screen when there is no Internet connection, along with the information icon.

Sample code would look like this:

```
If (CheckInternetConnection)
 {
  // your functionality code here
 }
 else
 {
   //set and display your inline "Service not available" message and icon here
 }
```


Challenges for the User Interface

A gadget looks quite simple, but developing a rich and effective gadget takes much more than HTML and JavaScript code. The design of the gadget almost always decides the fate of the gadget. Therefore, while developing gadgets an important factor to keep in mind is size limitations.

So far, we have approached gadget design in a methodical way. If you take care of information first and consider the usage pattern, you can develop a good user interface and ensure the behavior of the gadget. It's time to give boundaries to the gadget. Note that by *boundaries*, I mean the actual dimensions of all the interfaces of the gadget.

A gadget should not be treated as a web page. For example, the Zodiac Sign Gadget in Figure 3.19 outgrows the width of the Sidebar and has scrollbars. This is not a good design. The size of the gadget window should not be larger than the Sidebar when docked. A larger size is desirable only in the undocked mode.

FIGURE 3.19 A Zodiac Sign Gadget with width greater than that of the Sidebar looks cumbersome in the desktop.

Knowing the standard dimensions to use when designing specific gadget types is essential for you to have success in gadget development.

Standard Dimensions

All the interfaces in a gadget are HTML pages, each with recommended and allowed dimensions. These dimensions are set in the corresponding CSS files. There are different recommended dimensions for the different types of pages you might use. These page types include

- Gadget page when docked
- Undocked gadget
- Settings page
- Flyout page

Each of these types is documented in more detail in the following sections. Table 3.1 summarizes the recommended and maximum dimensions for each page type. These dimensions are recommended by Microsoft. More information on guidelines can be found at http://msdn2.microsoft.com/en-us/library/aa511443.aspx.

TABLE 3.1 The Maximum Dimensions for Each Page Type

Type of Page	Recommended Width×Height	Maximum Width×Height	Minimum Width×Height
Gadget main page when docked	130 pixels×150 pixels 5 pixels transparent border (2 left/3 right)	130×200	130 pixels×57 pixels
Gadget page when undocked (floating)	250 pixels×180 pixels	400 pixels×400 pixels	130 pixels×57 pixels
Settings page	280 pixels×180 pixels	300 pixels×400 pixels	Variable
Flyout page	320 pixels×240 pixels	400 pixels×400 pixels	Variable

Gadget Page When Docked

A gadget width when docked should be not more than 130 pixels. The Sidebar width is 150 pixels. The image used for the gadget's background needs to have two pixels on the left and three pixels on the right side for the gadget's drop-shadow effect. These five pixels are also sometimes kept transparent (see Figure 3.20).

The minimum height for a gadget is 57 pixels, which accommodates the Settings and Drag icons in the upper-right corner. The maximum height for a docked gadget is 200 pixels. Remember that a gadget that utilizes space efficiently is more likely to be used.

The recommended size for a docked gadget is 130×150 pixels.

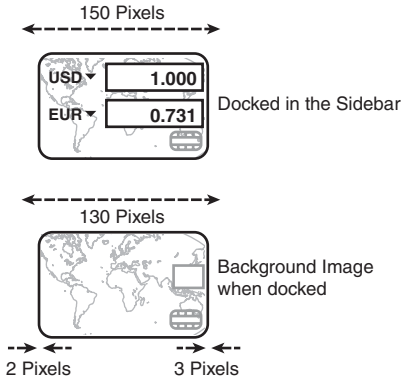

FIGURE 3.20 The Currency Gadget docked size is 130 pixels wide and 83 pixels high.

Undocked Gadget

An undocked gadget size should differ according to the gadget's functionality. If the gadget needs more real estate when undocked to show more information, the size changes accordingly. The maximum size is 400×400 pixels, and the recommended size is 250×180 (see Figure 3.21).

FIGURE 3.21 The undocked size for the Currency Gadget is 254×196 pixels.

Settings Page

The size of the settings page, like the one shown in Figure 3.22, depends on the amount of customization given to the user. For example, a Weather Gadget just needs to have an option for city and for temperature unit. The maximum allowed size for a settings page is 300×400. The recommended size is 278×180.

Flyout Page

The flyout window is meant for extra information and is optional. The maximum size allowed for a flyout page is 400×400 pixels. The recommended size is 320×240.

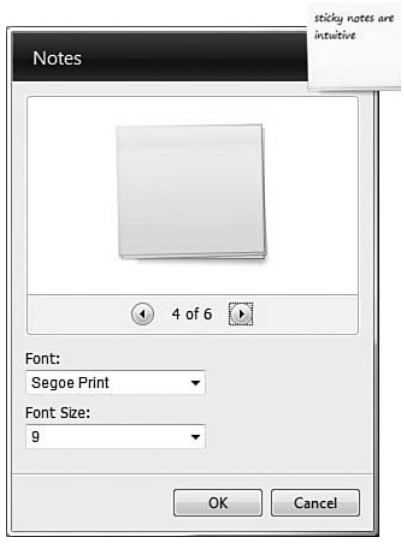

FIGURE 3.22 The settings page for the Notes Gadget is 278×310.

Other Interface Guidelines

The following is a list of other interface guidelines that should be considered:

- Use Windows styles for text, cursors, icons, and so on. Because gadgets reside within the Vista operating system, the images, text, and icons used should go along with the Vista theme.
- Use a similar style for the gadget's docked and undocked states. The style used for fonts, colors, and sizes should not be completely different between docked and undocked state.
- A gadget should be self-explanatory, with no need for help files. A gadget is a lightweight application and should be simple to understand and use.
- After installation, the gadget should start working. Initial gadget configuration makes a gadget look cumbersome. A gadget should come with working default settings. For example, a Weather Gadget shows the weather of a default city as soon as it is installed. The user is able to customize it later. Asking the user for a configuration as soon the gadget is installed is not the norm and should be avoided.
- An option to resize the gadget in its docked state is absolutely essential in a good design. A gadget that comes with an essential mini version in the docked state becomes an instant favorite of users.
- Flyouts should be used for additional information, but keep in mind that they automatically hide when the focus is in any other window. Flyouts cannot be used to display live data updates.

Visual Themes

The visual theme determines the look and feel of the gadget. Visual theme is about images, text, fonts, colors, sizes, and styles—everything together. It is all about what users see and also what they don't see. They all work together to present a unified theme to a user. Here is the list of items to consider when developing a theme for your gadget:

- Title
- Icon
- Drag image
- Background image
- Controls
- Text and style

A gadget's visual theme starts at its icon and title, which appear in the Gadget Picker window after installation.

Title

Keep the gadget's title to two or three meaningful words, 15 letters maximum, so that it displays properly in the Gadget Picker. For example, for a Google Search Gadget, use the title "Google Search" rather than "Google.com Search bar." Figure 3.23 shows two gadgets in the Gadget Picker window. The first one has the name "Currency" and the second one is called "MyMoneyGadget," which displays as "MyMoneyGad…." You can avoid this undesirable result by keeping the title short.

Icon

A transparent icon gives a rich look. The `icon.png` file is an image file whose dimensions are 64×64. Figure 3.23 shows two versions of a gadget that compares different currencies of the world. The left image uses a meaningful title and an attractive icon. The right image, MyMoneyGadget, is not a meaningful title and the icon is very crude. This is not a good design.

FIGURE 3.23 To avoid turning off potential users from the start, gadgets should have an attractive icon and short title.

Drag Image

The drag image appears when you try to drag an image from the Gadget Picker window to the Vista Sidebar. The source of the drag image is `drag.png`. It's meant to show the screenshot of the actual working gadget in its docked size (see Figure 3.24).

FIGURE 3.24 When a clock gadget is dragged in the Gadget Picker window, it shows the `drag.png` image.

Background Image

The gadget's background image is the wallpaper for the gadget. It is normally a plain image with rounded curves. The image is selected based on the gadget's functionality. An image of a world map for a Currency Gadget, for example, is appropriate (see Figure 3.25). Multiple images are used for the gadget for different modes, like docked and undocked. The images used can also be different for different sizes of the gadget.

FIGURE 3.25 A Currency Gadget uses different images to display the gadget's background in different sizes.

When the gadget is docked, the image should have a shadow of 2 pixels on the left and 3 pixels on the right. When the image is undocked, the shadow should be more prominent—10 pixels to the left and 18 pixels to the right.

You can add the shadow to an image by using photo-editing software or the graphics protocol provided by the Vista Gadget Model. A black shadow with an opacity level 25%

can be used for the docked image for a two-three-pixel shadow, and an opacity value of 75% can be used for a 10-pixel shadow in images for undocked gadgets.

Controls

For any controls or action buttons that need to appear on the gadget, only the glyph images consistent with Windows should be used. Figure 3.26 shows a number of these glyphs with possible actions they can each represent. If the action buttons need to be shown in the gadget's main window, always try to keep them hidden by default and have them activate only on mouse hover or when the gadget is active. All these glyph images can be found in the Extras section in the book download.

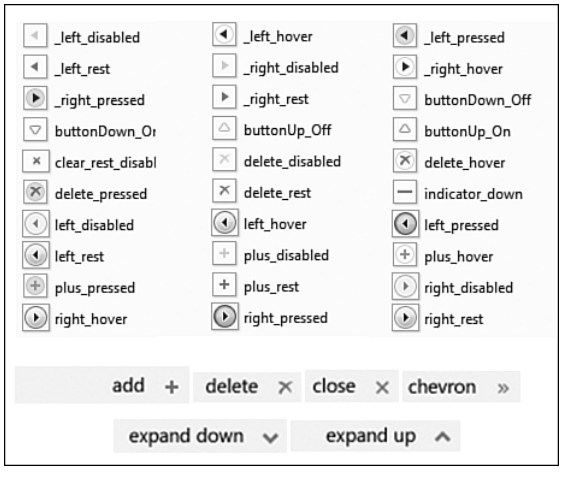

FIGURE 3.26 The glyph images that are consistent with the Windows Vista user interface should be used in the gadget.

Figure 3.27 shows an example of a gadget with controls that are hidden until the user hovers the mouse over the gadget.

FIGURE 3.27 The Notes Gadget by default (left image) does not show the action buttons. When the mouse is moved on the Notes Gadget (image on the right), the action buttons are visible.

Text and Style

The fonts and the styles used to format text in the gadget are also very important while designing a gadget. The style sheet is a language used to describe the presentation of an HTML page. A gadget can use multiple sStyle sheets for different pages of the gadget window, such as the settings page and the flyout page. The style sheet and font you should use depend on the gadget's functionality and theme, and can vary with different gadgets. Here is a link that can further help you understand HTML and CSS: http://www.w3.org/Style/Examples/011/firstcss.

Here is some of the standard style information as applied to the different pages. These styles help the gadget to display a consistent look with Windows Vista.

For the gadget window:

```
Font: Calibri or Segoe UI
Style:
body
{
    margin: 0px;
    padding: 0px;
    color: #000;
    font-family: Calibri, Tahoma, sans-serif;
}
```

For the settings window:

```
Font: Segoe UI
Style:
body
{
    padding: 0;
    margin: 0px;
    font-family: Segoe UI, Tahoma, sans-serif;
    font-size:12px;
}
```

For the flyout:

```
Font: Calibri
Style:
body
{
    margin: 0px;
    padding: 0px;
    color: #ffffff;
    font-size: 10px;
    font-family: Calibri, Tahoma, sans-serif;
}
```

NOTE

These are merely recommendations for suitable style sheet information. Users should change it according to the feature set that the gadget provides. For example, the Notes Gadget shown in Figure 3.27 uses italic fonts.

Transparent Images in the Gadget

The images used in a gadget window have a shadow effect and sometimes semitransparent oval shapes. These images use either a PNG or a GIF image format. The recommended images used for the gadget are in a PNG format. The next few sections describes these formats in more detail.

Alpha Transparency

Microsoft started supporting PNG icons with alpha transparency in Windows Vista. Creating an alpha transparent image is a process of combining two images: a central image laid on top of a background image to create an appearance of partial transparency. It is used to display an image that has transparent or semitransparent pixels.

Alpha transparency is used extensively in the images used for gadgets to create the shadow or oval shape effect.

Portable Network Graphics File

PNG is an abbreviation for Portable Network Graphics. It's an advanced graphics format with 48-bit color. It's one of the standard formats that is beginning to replace the use of GIF images, which are limited to 8-bit colors (256 colors). It also includes an alpha channel for showing transparency.

This is the standard format used in all the images for a gadget. A PNG file allows every pixel (dot) to have any level of transparency, from completely opaque to completely transparent and anything in between.

The PNG image format combines the best features of GIF and JPG/JPEG. It supports binary transparency along with alpha transparency, meaning that each pixel of the image can also have one of 256 different levels of transparency. PNG format produces a file with approximately the same file size as that of an equivalent GIF image, assuming that they have the same number of colors. Appendix A shows how to create a PNG image in Adobe Photoshop.

GIF File Limitations

Graphics Interchange Format is a popular bitmapped graphics file format developed by CompuServe. To date, it is the most widely used graphics format on the Web. Given that GIF files can also be transparent, you might wonder why you would not want to use them in favor of PNG. There are three reasons for not using a GIF format:

- GIF files support only 100% transparency; you cannot create a translucent shadow effect.
- GIF images are limited to 8-bit colors, so they're not very rich.
- PNG format is endorsed by the World Wide Web Consortium (W3C), an organization responsible for managing standards for the World Wide Web.

About Accessibility

Accessibility is a way of producing applications accessible to the broadest range of people. This includes people with disabilities such as poor eyesight, motor impairments, and so on, and also to people who prefer to use keyboards for fast access, people with limited bandwidth, with older computers, people using applications such as text-only browsers, screen readers, and other devices to access applications. To learn more about adding accessibility to an application, visit the Microsoft Accessibility Developer Center at http://msdn2.microsoft.com/en-us/accessibility/default.aspx.

For a Sidebar gadget, accessibility relates to theme colors, high color contrast, and HTML accessibility in general. Adding accessibility to a gadget means taking care of gadget design in terms of the following:

- Keyboard access
- HTML accessibility
- Theme colors and contrast

Keyboard Access

One of the fundamental accessibility scenarios is that some users use only their keyboards. This is also a requirement from power-users who love the speed of using the keyboard rather than a point-and-click device such as a mouse. So, specifically, the Sidebar and the gadgets themselves should be accessible from the keyboard.

The keyboard support listed in Table 3.2 is provided by the Sidebar

TABLE 3.2 Shortcut Keys to Access Sidebar Gadget

Keyboard Access	Action
Windows logo key + Spacebar	Brings all gadgets to the front and puts focus on the Windows Sidebar
Windows logo key + G	Cycles through the gadgets

After your gadget is in focus, you have to make sure all its features are accessible through the keyboard. The following lists important points to keep in mind when adding accessibility:

- Focus should be put on the gadget when it is loaded.
- Users should be able to tab through the controls and links in the gadget.

- The Enter key should act like a mouse click to push a button or follow a link and should call the `onclick` event of the control or link.
- The `onfocus` and `onfocusout` functions should be called to simulate the mouse hover effect.
- Apart from the gadget main page, a gadget's Flyout and Settings page should also be accessible from the keyboard.

The, "About Accessibility," section in Chapter 6, "Design Patterns and Standard Practices," shows you how to implement these in a gadget.

General HTML accessibility

Because all the pages used in a gadget (the main gadget window, Flyout, and Settings pages) are HTML pages, accessibility rules apply to them as well. To adhere to these rules you should attend to the following:

- Provide text alternatives for images and links. Set the `ALT` tag in the image and `Title` tag in the `Anchor` element. The alt text as well as the title act as a short description of the image that serves the same purpose as the image or the link itself. If the image is purely a decoration, then set alt="". For example, if the gadget has a weather image, then set `alt="Cloudy"` or `alt ="Partly sunny"`. Reflecting what the image conveys fulfills this requirement.
- Create content in a modular way so that the structure of the page (HTML), presentation (CSS), and behavior (JavaScript functionality) are in separate pages and the HTML page is accessible without the CSS file.

These guidelines are from the W3C Initiative Web Content Accessibility Guidelines 2.0, which can be found at http://www.w3c.org/TR/WCAG20/.

Theme Colors and Contrast

When adding accessibility to a gadget, you next need to consider the colors used in the background, images, HTML page elements, and so forth.

- Pick contrasting foreground and background colors to make your gadget information more visible. This is especially important for the color of the text against the background color. Contrast is a function of lightness, hue, and saturation. Here's a resource from the Lighthouse for the Blind, an expert in vision disabilities: http://www.lighthouse.org/accessibility/effective-color-contrast/.
- Avoid using red and green combinations because people with red-green colorblindness cannot distinguish between red and green. There are other types of colorblindness, but red-green is the most common.
- Do not use color alone to convey information.

Index

A

B

C

D

E - F

G

H

I

J - K - L

M

N

O

P

Q - R

S

T

U

V

W

X

Y - Z