



Introduction to Short Cuts



awprofessional.com



prenhallprofessional.com



quepublishing.com



samspublishing.com



ciscopress.com

Contents at a Glance

About Digital Short Cuts	3
Sample: Using BusyBox	4
Sample: Understanding AJAX	8
Sample: Debugging Embedded Linux	13
Sample: Using Google Spreadsheets	16
Sample: Refactoring to Agility	21
Sample: Cheating Online Games	25
Sample: Using Microsoft Windows Media Player 11	29
For More Information	33

What People Are Saying

“I was pleased to participate in the Short Cuts program for several reasons. Technology, especially open source, moves at a lightning fast pace. Short Cuts is a way to offer content as it’s being written, often well in advance of the books’ publication date. Some topics stand alone quite well, and are useful as references to the practicing engineer. This is especially true when the topic covers a subject or technology that is widely popular among the expected readership. Such is the case with *Busybox*. As an author, it is also a very useful way to get early feedback on the content.”

—Chris Hallinan, author of, *Using BusyBox* (Digital Short Cut) ISBN: 0132335921

“I wrote my shortcut because I had worked with a number of companies that were having trouble transitioning from waterfall or adhoc development methods toward more agile methods. Those companies were intimidated by the transition because it seems like a large and very risky leap. Working with those companies, I refined a strategy for making that transition that can also help companies with agile methods continue to optimize their development processes. I wrote my shortcut in the hopes that other companies can benefit from our experiences. The shortcut format seemed very appropriate because the publishing turn around was fast and companies that need this information need it now.”

—Carol Wellington, author of *Refactoring to Agility* (Digital Short Cut) ISBN: 0321486471

“I’ve recently picked up *Rubyisms in Rails*, a Short Cut from Addison-Wesley Professional, authored by Jacob Harris. I love this format. For about 10 bucks, you can get a PDF-only copy of a 50-100 page “book”. All of the Short Cuts I’ve seen so far have been very focused, covering a reasonable topic in sufficient depth without creating a monstrous 600+ page tome. To me, Short Cuts are almost perfect — they’re timely, useful, and a great value.”

—Pat Eyler, Co-Founder of the Seattle Ruby Brigade

“I thought I wrote the book on hacking online games, I was wrong!”

—The Warcry Network, on *Cheating Online Games* (Digital Short Cut) ISBN: 0321460723

“PTG has a relationship with an authoring community like no other publishing house. Industry experts develop the leading print books for us today and PTG applies the same care to our digital products as we do to any of our printed books. Readers of Short Cuts can expect the same level of quality, accuracy, knowledge, and insight they receive from our printed books, and gain access to the information sooner, in a more portable form.”

—Paul Boger, Vice President and Publisher, Pearson Education

About Digital Short Cuts

Short Cuts are short, concise, PDF documents designed specifically for busy technical professionals. Each Short Cut is tightly focused on a specific technology or technical problem. This may be a cutting-edge new technology that shows great promise, or it may be an existing technology that has reached the “tipping point” and is about to take off. Either way, written by industry experts and best selling authors, Short Cuts get customers the technical information they need — now.

Varying from 50 to 150 pages in length, Short Cuts are developed to save readers time, show users how to solve a specific problem, or introduce a new topic.

Format

Short Cuts published by Pearson Technology Group imprints are in Mobi Pocket format via Amazon.com and in Adobe® Reader® format, which means they are optimized for online viewing. Most Short Cuts are encrypted with Adobe’s Digital Rights Management (DRM) technology. You can install documents on up to six computers registered to an individual user. These may include desktop and laptop computers and handheld devices such as PDAs as well as upgraded machines. We strongly recommend using Adobe Reader 7 or Adobe Acrobat 7 to view Digital Short Cuts.

Viewing and Navigation

Bookmarked tables of contents provide easy access to particular sections. You can also jump to a topic by clicking its entry in the Bookmarks palette, which is available from any page in the document.

Some Digital Short Cuts contain embedded files for use by other programs. Embedded files are designated by this icon:



Printing

Unlimited printing is allowed. Copying is limited to 30 pages every 30 days in order to limit piracy risks. Digital Short Cuts use a horizontal (or landscape) page layout for better viewing on computer monitors.

Copy/Paste Features

Customers who have the professional version of Adobe Acrobat® will also be able to annotate and highlight the Short Cut.

Prentice Hall Open Source Software Development Series

Using BusyBox

Christopher Hallinan

SAMPLE Using BusyBox
by Christopher Hallinan
ISBN 0132335921

11.1 Introduction to BusyBox	3
11.1.1 BusyBox is Easy	4
11.2 BusyBox Configuration	5
11.2.1 Cross-Compiling BusyBox	7
11.3 BusyBox Operation	8
11.3.1 BusyBox Init	12
11.3.2 Example rcS Initialization Script ..	16
11.3.3 BusyBox Target Installation	17
11.3.4 BusyBox Commands	20
11.4 Chapter Summary	21
11.4.1 Suggestions for Additional	22
Reading	



11.3.2 Example rcS Initialization Script

Before BusyBox spawns an interactive shell, it tries to execute commands from a script called `/etc/init.d/rcS`, as shown in Listing 11-7. It is here where your applications come to life in a BusyBox system. A simple rcS initialization script is provided in Listing 11-8.

LISTING 11-8 Simple rcS BusyBox Startup Script

```
#!/bin/sh

echo "Mounting proc"
mount -t proc /proc /proc

echo "Starting system loggers"
syslogd
klogd

echo "Configuring loopback interface"
ifconfig lo 127.0.0.1

echo "Starting inetd"
xinetd

# start a shell
busybox sh
```

This simple script is mostly self-explanatory. First, it is important to mount the `/proc` file system on its reserved mount point, `/proc`. This is because many utilities get their information from the `/proc` file system. This is explained more fully in Chapter 9. Next we launch the system loggers as early as possible, to capture any startup problems. Following the system log daemons, we configure the local loopback interface for the system. Again, a number of traditional Linux facilities assume that a loopback interface is present, and if your system has support for sockets configured, you should enable this pseudo interface. The last thing we do before starting a shell is launch the Internet superserver `xinetd`. This program sits in the background listening for network requests on any configured network interfaces. For example, to initiate a `telnet` session to the board, `xinetd` intercepts the request for `telnet` connection and spawns a `telnet` server to handle the session.

Instead of starting a shell, your own applications can be launched from this `rcS` initialization script. Listing 11-8 is a simple example of a Telnet-enabled target board running basic services such as system and kernel loggers.

11.3.3 BusyBox Target Installation

The discussion of BusyBox installation can proceed only when you understand the use and purpose of symlinks. The BusyBox makefile contains a target called `install`. Executing `make install` creates a directory structure containing the `busybox` executable and a symlink tree. This environment needs to be migrated to your target embedded system's root directory, complete with the symlink tree. The symlink tree eliminates the need to type `busybox` command for each command. Instead, to see a listing of files in a given directory, the user need only type `ls`. The symlink executes `busybox` as described previously and invokes the `ls` functionality. Review Listing 11-4 and Listing 11-5 to see the symlink tree. Note that the BusyBox build system creates links only for the functionality that you have enabled via the configuration utility.

The easiest way to populate your root file system with the necessary symlink farm is to let the BusyBox build system do it for you. Simply mount your root file system on your development workstation and pass a PREFIX to the BusyBox makefile. Listing 11-9 shows the procedure.

LISTING 11-9 Installing BusyBox on Root File System

```
$ mount -o loop bbrootfs.ext2 /mnt/remote
$ make PREFIX=/mnt/remote install
/bin/sh applets/install.sh /mnt/remote
  /mnt/remote/bin/ash -> busybox
  /mnt/remote/bin/cat -> busybox
  /mnt/remote/bin/chgrp -> busybox
  /mnt/remote/bin/chmod -> busybox
  /mnt/remote/bin/chown -> busybox
...
  /mnt/remote/usr/bin/xargs -> ../../bin/busybox
  /mnt/remote/usr/bin/yes -> ../../bin/busybox
  /mnt/remote/usr/sbin/chroot -> ../../bin/busybox
```

.....
You will probably need to make your busybox binary
setuid root to ensure all configured applets will
work properly.
.....



Prentice Hall Open Source Software Development Series

Understanding AJAX

Consuming the Sent Data with XML and JSON

Joshua Eichorn

SAMPLE

Understanding AJAX
by Joshua Eichorn
ISBN 0132337932

Introduction

3.1 Document-Centric Approaches	3
3.2 Remote Scripting	20
3.3 How to Decide on a Request Type	51
3.4 Summary	52

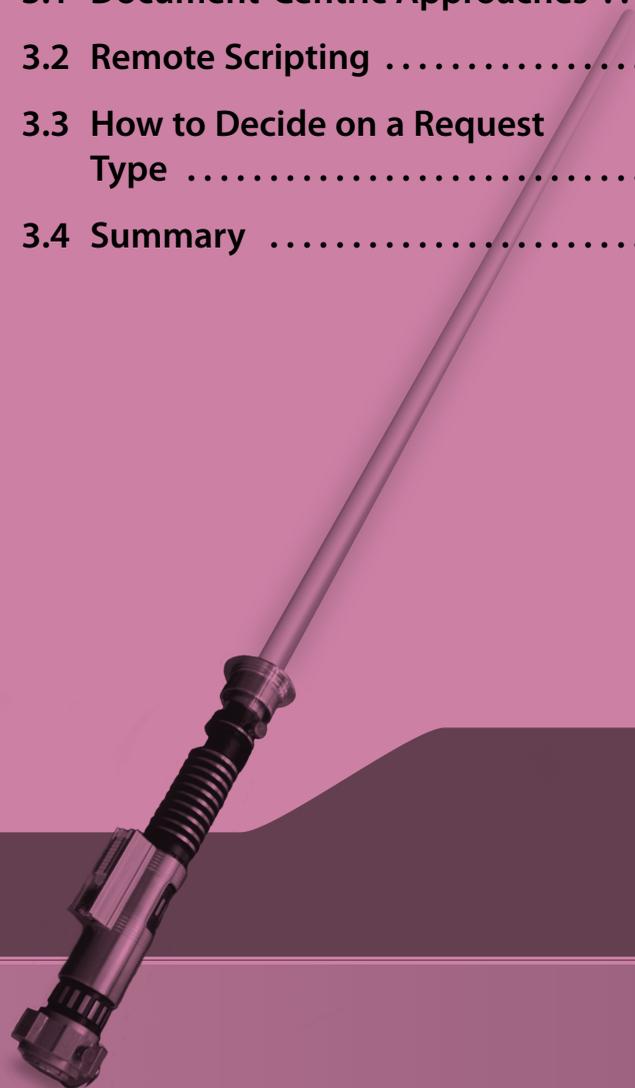
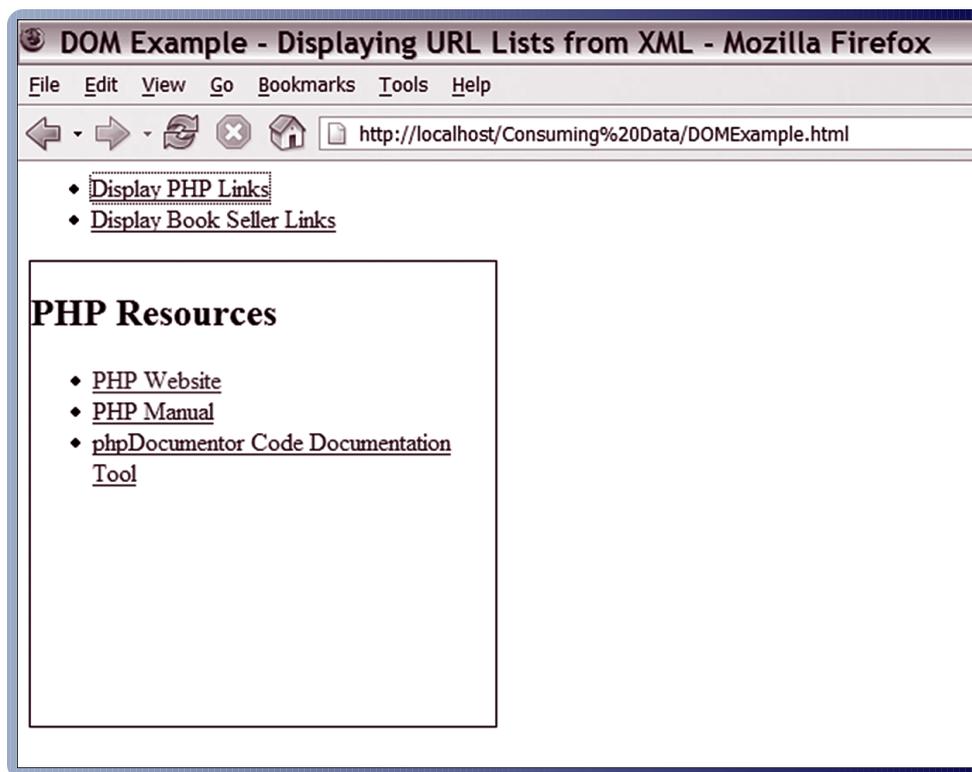


FIGURE 3-2

Updating an HTML page by consuming XML documents using the DOM



3.1.3 Consuming XML Using XSLT

eXtensible Stylesheet Language Transformations (XSLT) is another popular way to take a DOM document and generate new output. The advantage it has over DOM is that the transformation and the data are in an XML file. XSLT has been used by many successful AJAX applications, such as Google Maps, but it does have a number of drawbacks. XSLT browser support is uneven, and

even when two browsers, such as Internet Explorer 6 and Firefox 1.0, support the same main features, the application programming interfaces (APIs) for controlling the transformations from JavaScript are completely different. This difference is large enough that you can't just write a simple wrapper like you can for XMLHttpRequest. If you need cross-browser support, you'll need to rely on a library like Sarissa instead. (The Sarissa library is explained in detail in Chapter 8, "Libraries Used in Part II: Sarissa, Scriptaculous.")

XSLT can also be problematic simply due to its complexity. Not only will you need to learn how to write the XSLT style sheets that drive the actual transformation, but you'll also need to learn XPath, which is used to refer to XML nodes inside the style sheet. Because XSLT is a World Wide Web Consortium standard, there are tools and documentation out there to help, but in many cases, the added effort required over a DOM approach isn't worth the effort.

Although the purpose of this book isn't to teach you how to write an XSLT style, I will explain the basics of the one used. Listing 3-5 replaces the JavaScript DOM code in Listing 3-4 with an XSLT transformation. The same XML data files (PhpResources.xml and Booksellers.xml) that are used in the DOM example are used here. The Mozilla XSLT API is used in these examples because it's easier to understand, and all you have to do to make it work in IE is include the Sarissa library.

LISTING 3-5 XSLTExample.html

```
1 <html>
2 <head>
3 <title>XSLT Example - Displaying URL Lists from XML</title>
4 <script type="text/javascript" src="HttpClient.js"></script>
5 <script type="text/javascript" src="sarissa/sarissa.js"></script>
```

LISTING 3-5 Continued

```
6 <script type="text/javascript">
7 var client = new HttpClient();
8 client.isAsync = true;
9 var xsltProcessor = new XSLTProcessor();
10
11 function setup() {
12     // set up an xsltprocessor and import the stylesheet
13     client.callback = function() {
14         var xslRef = client.xmlhttp.responseXML;
15         xsltProcessor.importStylesheet(xslRef);
16     }
17     client.makeRequest('Resources.xsl');
18 }
19
20 function displayResources(url) {
21     client.callback = function() {
22         var newDom = client.xmlhttp.responseXML;
23         var output = xsltProcessor.transformToDocument(newDom);
24         document.getElementById('target').innerHTML = "";
25         document.getElementById('target').appendChild(
26             document.importNode(output.firstChild,true));
27     }
28
29     client.makeRequest(url,null);
30 }
```

LISTING 3-5 Continued

```
31 </script>
32 </head>
33 <body onload="setup()">
34
35 <ul>
36     <li><a href="javascript:displayResources('PhpResources.xml')">
37         Display PHP Links</a></li>
38     <li><a href="javascript:displayResources('BookSellers.xml')">
39         Display Book Seller Links</a></li>
40 </ul>
41
42 <div style="position: absolute; width:100px; height:20px;
43     top: 5px; right: 5px; display:none"
44     id="HttpClientStatus">Loading ...</div>
45
46 <div id="target" style="width:300px; height:300px;
47     border: solid 1px black"> </div>
48 </body>
49 </html>
```

The first 18 lines cover the basic setup; we include our `HttpClient XMLHttpRequest` wrapper and the Sarissa XML compatibility library. On line 7, we create an `HttpClient` instance; this will be used to load both the style sheet and the XML files we're going to transform; on line 8, we set `isAsync` to `true` because we will be making only asynchronous requests. On line 9, we create a new

shortcut

Prentice Hall Open Source Software Development Series

Debugging Embedded Linux

Christopher Hallinan

SAMPLE

Debugging Embedded Linux
by Christopher Hallinan
ISBN 0131580132

 PRENTICE HALL
PEARSON EDUCATION

PRENHALLPROFESSIONAL.COM

KERNEL DEBUGGING TECHNIQUES

14.1 Challenges to Kernel Debugging	4
14.2 Using KGDB for Kernel Debugging	6
14.3 Debugging the Linux Kernel	14
14.4 Hardware-Assisted Debugging	53
14.5 When It Doesn't Boot	63
14.6 Chapter Summary	69

DEBUGGING EMBEDDED LINUX APPLICATIONS

15.1 Target Debugging	72
15.2 Remote (Cross) Debugging	73
15.3 Debugging with Shared Libraries	80
15.4 Debugging Multiple Tasks	87
15.5 Additional Remote Debug Options	97
15.6 Chapter Summary	99



SECTION 14.2

Using KGDB for Kernel Debugging

14.2.2 Target Boot with KGDB Support

After your kernel is built with KGDB support, it must be enabled. Unfortunately, the method to enable it is not yet uniform across all architectures and implementations. In general, KGDB is enabled by passing a command-line switch to the kernel via the kernel command line. If KGDB support is compiled into the kernel but not enabled via a command-line switch, it does nothing. When KGDB is enabled, the kernel stops at a KGDB-enabled breakpoint very early in the boot cycle to allow you to connect to the target using `gdb`. Figure 14-3 shows the logic for generating an initial breakpoint when KGDB is enabled.

KGDB requires a serial port for connection to the host.³ The first step in setting up KGDB is to enable a serial port very early in the boot process. In many architectures, the hardware UART must be mapped into kernel memory before access. After the address range is mapped, the serial port is initialized. Debug trap handlers are installed to allow processor

³ Notwithstanding the comments made earlier about KGDB over Ethernet.

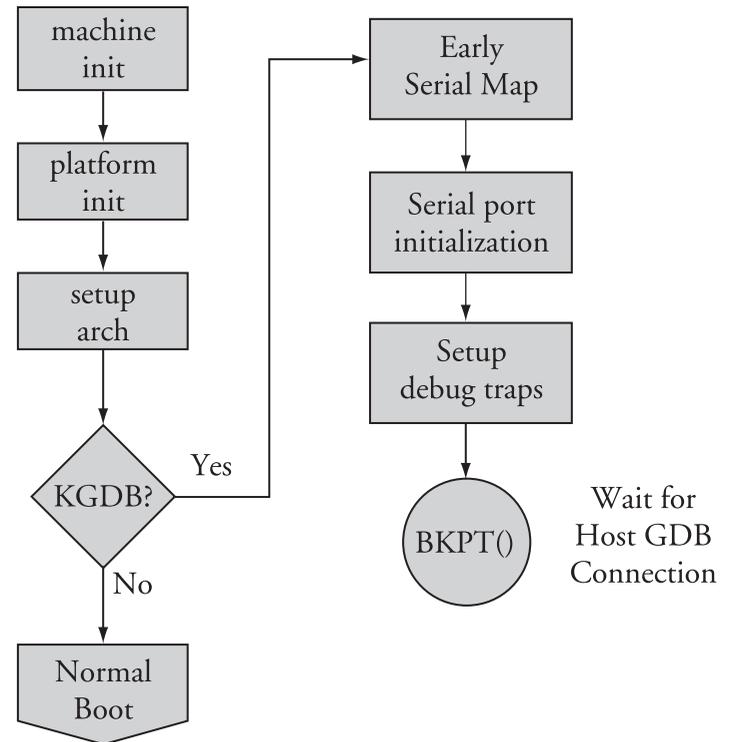


FIGURE 14-3
 KGDB logic

SECTION 14.2

Using KGDB for Kernel Debugging

mapped, the serial port is initialized. Debug trap handlers are installed to allow processor exceptions to trap into the debugger.

Listing 14-1 displays the terminal output when booting with KGDB enabled. This example is based on the AMCC 440EP Evaluation Kit (Yosemite board), which ships with the U-Boot bootloader.

LISTING 14-1 Booting with KGDB Enabled Using U-Boot

```
=> sete bootargs console=ttyS1,115200 root=/dev/nfs rw ip=dhcp gdb
=> bootm 200000
## Booting image at 00200000 ...
   Image Name:   Linux-2.6.13
   Image Type:   PowerPC Linux Kernel Image (gzip compressed)
   Data Size:    1064790 Bytes = 1 MB
   Load Address: 00000000
   Entry Point:  00000000
   Verifying Checksum ... OK
   Uncompressing Kernel Image ... OK
#T0440:c000ae5c;01:c0205fa0;#d9    <<< See text
```

Most of the boot sequence is familiar from our coverage of U-Boot in Chapter 7, “Bootloaders.” This kernel boot sequence has two unique features: the command-line parameter to enable KGDB and the odd-looking text string after the kernel is uncompressed.

Recall from Chapter 7 that the kernel command line is defined by the U-Boot `bootargs` environment variable. Notice that we have added the `gdb` parameter, which instructs the kernel to force an early breakpoint and wait for the host debugger (your `cross-gdb`) to connect.

USING Google™ Spreadsheets

Michael Miller

SAMPLE

Using Google Spreadsheets
by Michael Miller
ISBN 0768668336

Introduction

Introducing Google Spreadsheets	4
Migrating from Excel to Google Spreadsheets	12
Navigating the Google Spreadsheets Workspace	12
Creating, Opening, and Saving Google Spreadsheets	21
Entering and Editing Data	26
Working with Ranges	27
Sorting Data	29
Inserting, Deleting, Moving, and Copying Data	30
Changing Column Width and Row Height	33
Formatting Google Spreadsheets	34
Entering and Editing Formulas	39
Using Functions	42
Working with Multiple Sheets	64
Printing Google Spreadsheets	65
Sharing and Collaborating with Google Spreadsheets	65
A Final Word	70
Index	71
Command Reference	73

Printing Google Spreadsheets

When you're finished creating your spreadsheet, you might want to print a hard copy. This is a fairly easy, if not overly intuitive, task, as there's no quick and easy "print" button or function. Instead, here's what you need to do:

1. Click the File button and select Get HTML.
2. This opens a new browser window that contains just the spreadsheet, no other buttons or controls, as shown in Figure 40. It's this window that you want to print.
3. Switch to the new browser window, then click the Print button in your web browser.

Sales by Region					
	Q1	Q2	Q3	Q4	FULL YEAR
North	\$2,000	\$2,100	\$2,200	\$2,400	\$8,700
East	\$2,500	\$2,600	\$2,500	\$2,700	\$10,300
West	\$3,000	\$2,800	\$2,900	\$3,200	\$11,900
South	\$2,200	\$2,100	\$2,000	\$1,900	\$8,200
TOTAL	\$9,700	\$9,600	\$9,600	\$10,200	\$39,100

Figure 40 Use the Get HTML command to open a new browser window for printing.

The contents of the new browser window—that is, your spreadsheet data—will now print.

note

Google Spreadsheets doesn't let you automatically print multiple sheets in a spreadsheet file; it only prints one sheet at a time. You must switch to each sheet separately, and then go through the File, Get HTML procedure.

Sharing and Collaborating with Google Spreadsheets

The truly unique feature of Google Spreadsheets is the capability to share a spreadsheet with others—either for viewing or for collaborative editing. The only hitch to this process is that anyone you wish to share with must have their own Google Account to access the Google Spreadsheets site. That said, you can easily invite another users to create his or own new Google Account.

note

At present, there is no limit to the number of people with whom you can share a spreadsheet. The only requirement is that anyone you invite to share your spreadsheet have their own Google Account.

Sharing a Spreadsheet for Viewing

We'll start with the process of sharing a spreadsheet for viewing only. This lets other users view your spreadsheet but doesn't give them access to add or edit data. (It's a read-only process.)

To share a spreadsheet for viewing, follow these steps:

1. From a saved spreadsheet, click the Show Sharing Options link; this opens the sharing pane, shown in Figure 41.
2. In the Invite People to View box, enter the email addresses of the people you want to share the spreadsheet. (Separate multiple addresses with commas.)
3. Click the Invite People button.
4. This opens an Invitation window, like the one shown in Figure 42. Enter a personal message if you like, and then click the Send Invitation button.

Your recipients now receive an invitation via email. The invitation contains a link to the spreadsheet; clicking this link opens the spreadsheet in a new browser window.

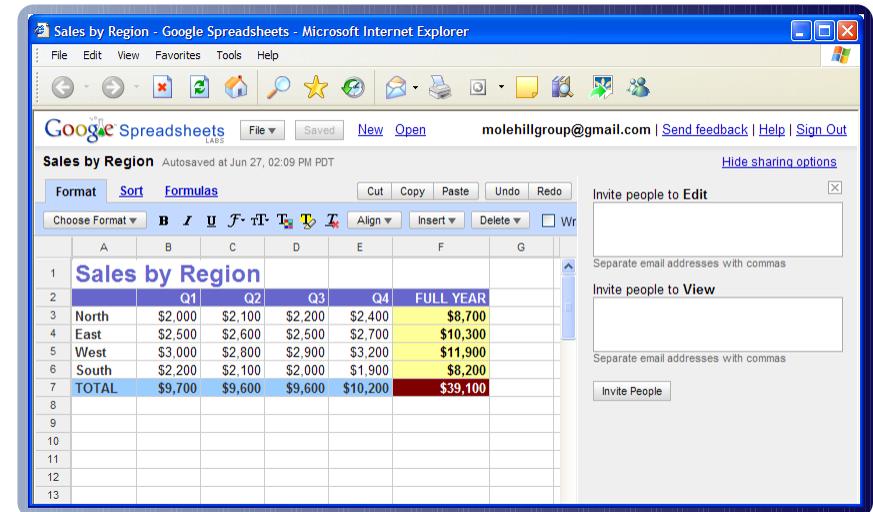


Figure 41 Getting ready to share a spreadsheet.

Anyone viewing your spreadsheet can not only navigate around the entire file (including multiple sheets within the spreadsheet file), but also save that file to their personal Google Spreadsheets online storage area or as an XLS-format file to their own PC. While viewing, other users will see the current work in progress; as soon as you press Enter when editing a cell, the edited contents will appear on the other users' screens. (They can't see the in-progress contents of the cell while you're editing the cell, however.)

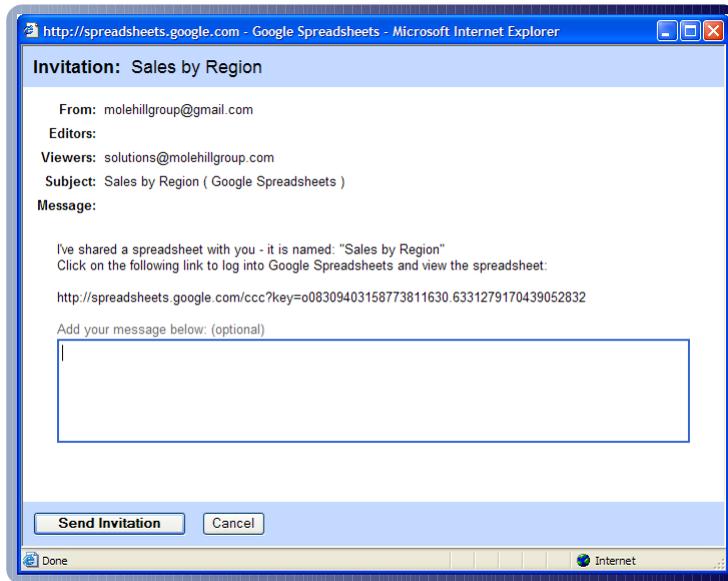


Figure 42 *Inviting another user to share a spreadsheet.*

note

Anyone you invite to view a spreadsheet can, in turn, invite other users to also view the spreadsheet.

Sharing a Spreadsheet for Collaboration

When you invite someone to view a spreadsheet, that's all they can do; they can navigate around the spreadsheet, but they can't add to or edit any of the data within the spreadsheet. If you want

to collaborate with others on a spreadsheet, you must explicitly open the spreadsheet for collaboration.

caution

Google permits more than one user at a time to make changes to an open spreadsheet; the spreadsheet isn't "locked" when the first user starts editing. This can create havoc if both users try to make changes to the same data or aren't aware of the other changes being made. For this reason, you should always use caution while collaboratively editing a spreadsheet.

To share a spreadsheet for collaboration, follow these steps:

1. From a saved spreadsheet, click the Show Sharing Options link; this opens the sharing pane.
2. In the Invite People to Edit box, enter the email addresses of the people with whom you want to collaborate. (Separate multiple addresses with commas.)
3. Click the Invite People button.
4. This opens an Invitation window. Enter a personal message if you like, and then click the Send Invitation button.

Your recipients now receive an invitation via email. The invitation contains a link to the spreadsheet; clicking this link opens the spreadsheet in a new browser window. Once accessed, the

Sharing and Collaborating with Google Spreadsheets

other users can now edit the spreadsheet, in real time. In fact, multiple users can edit the spreadsheet at the same time; each person's edits appear on all the other users' screens as soon as they press the Enter key when editing a cell.

note

Anyone you invite to edit a spreadsheet can, in turn, invite other users to also edit the spreadsheet.

Chatting with Spreadsheet Viewers

As you can see in Figure 43, the view spreadsheet window that other users see includes a chat pane. This lets your viewers chat with you, in real time, while they're viewing your spreadsheet.

To view the chat pane in your own spreadsheet window, simply click the Chat With link at the top right of the window. You can close the chat pane at any time by clicking the resulting Hide Chat link.

While chatting, you enter your comments in the bottom text box. When you press Enter, your comments are sent to the other users and appear in the main text box in the chat pane.

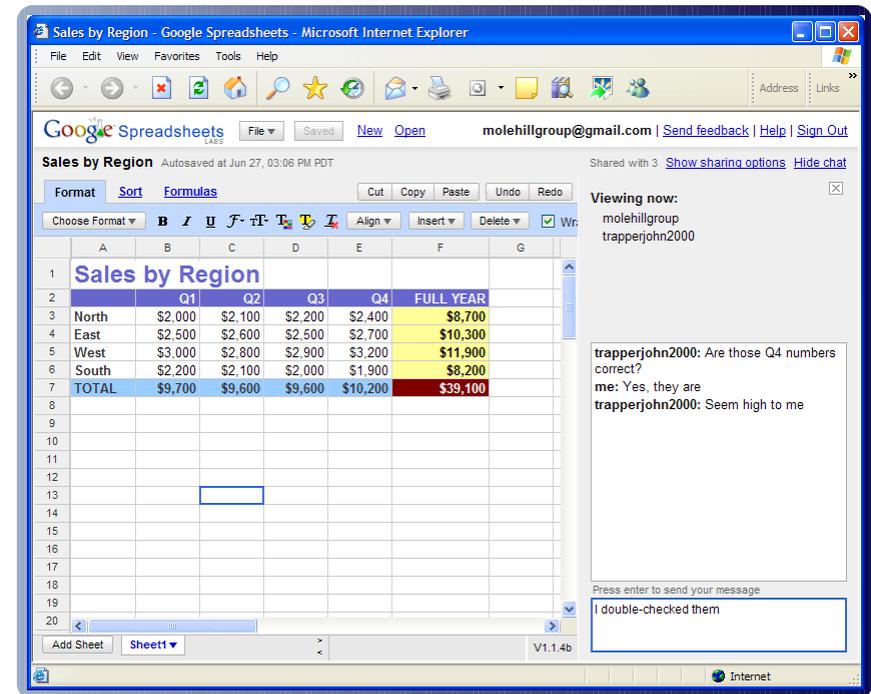


Figure 43 *Chatting with other viewers/editors of your spreadsheet.*

Rescinding Access

If, at any time, you want to block a user from viewing or editing your spreadsheet, here's what to do:

shortcut

Refactoring to Agility

Carol A. Wellington

SAMPLE

Refactoring to Agility
by Carol A. Wellington
ISBN 0321486471

 **Addison-Wesley**
Pearson Education

AWPROFESSIONAL.COM

Contents

Introduction	4
What Is Agility?	22
Phase 1—Getting to Fixed-length Development Iterations	33
Phase 2—Measuring the Process	51
Phase 3—Refactoring the Process	70
Process Innovations by Type	79
Process Smells	95



What Is Agility?

2.1 Agility Is Not Binary

Currently, processes are divided into two distinct categories: agile and plan-driven. Each has a community, a variety of practices, and a list of successful applications of those practices. With a few notable exceptions, there is very little overlap between these categories. This gives the appearance that agility is a binary attribute. However, agility is not an all-or-nothing affair that we can achieve through a single change in the development process.

Instead of asking, “Are we agile?” and trying to initiate a single, dramatic process change that will “achieve” agility, a more appropriate perspective focuses on “To what degree are we agile?” Within its community, agility has been defined as “the ability to both create and respond to change.”[5] We can answer this question by measuring the length of time between a

requirement change and when we deliver the updated functionality to a customer. This length of time is known in Lean Software Development as **cycle time**.

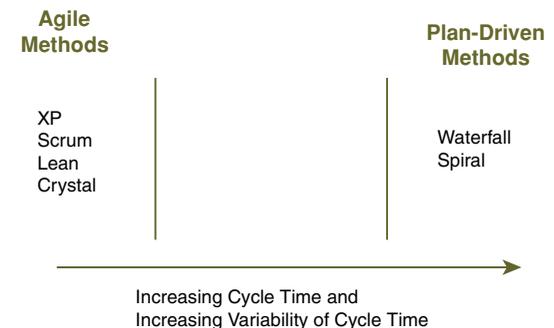


FIGURE 2.1 Apparent distinction between agile and plan-driven methods

Agile methodologies have been designed to have very short cycle times when compared with plan-driven methodologies. In general, agile methods advocate releasing new functionality in fixed-length iterations, and the iteration length defines their cycle time. This

is the most significant difference between agile and plan-driven methods: agile methods have a fixed cycle time that is relatively short, which lets the team adjust to changing customer needs rapidly.

Plan-driven (and ad hoc) methodologies define the length of time between releases by the functionality they want to be included in the release. As a result, releases are not of a fixed length, so they do not have a consistent cycle time. However, as a starting point, every process has a (possibly variable) cycle time and therefore some (possibly low) degree of agility.

With this perspective, agility is not a binary attribute and does not mandate a dramatic change in your process today. The current process has some degree of agility (though it may be slow and unstable) and defines a starting point on the path to improved agility. The challenge is to maximize the strengths of the current process and to make changes to that process in a disciplined manner with specific goals in mind.

2.2 How Much Agility Is Realistic Today?

Having acknowledged that there are degrees of agility, the relevant questions change significantly. Instead of asking, “Are we agile?” the question becomes, “To what degree should we be agile?” Increasing an organization’s agility requires an investment that may be non-trivial and brings a certain amount of risk. The cost of change must be weighed against the benefits, including any reduction in development risks, to ensure the long-term viability of the organization. It is only beneficial to pursue agility in cases where increasing agility increases the capabilities of the organization. These decisions will require that we define what is needed for long-term viability and specify how to measure our progress toward that goal.

Fully embracing an agile methodology requires changing the way engineers allocate their time and how they approach all software development activities.

The management mechanisms and the techniques and strategies for developing software differ dramatically from plan-driven mechanisms. Therefore, effectively using an agile methodology requires new skills for the engineers, project managers, and the organization as a whole. This means that we also need to consider our organization's ability to achieve agility given its current skill set. So we need to ask, "To what degree can we be agile?"

2.3 What Do We Need to React to with Agility?

In general, the "changes" that agility refers to have been changes in requirements. Agile processes have been designed for situations when requirements are either not well known or quickly changing. Usually, "agile" refers to a process, but in a broader sense, agility denotes the ability to deliver functionality in a

timely manner and the ability to adjust to externally imposed change. In these terms, "change" usually means changing requirements, but it can also mean changes in things like technology requirements, assigned personnel, or management.

In fact, an organization demonstrates agility, or a lack thereof, in response to a wide variety of changes—not just changes in requirements. For example, a plan-driven organization with a solid training and mentoring program may be quite agile when dealing with changes in personnel, but it is unlikely that it will be agile in response to changes in the customer's needs. Similarly, an XP organization may be quite agile when the customer's needs change, but the level of automation and uniqueness of their process may make them less agile in dealing with changes in personnel. For these reasons, we must ask, "What kind of agility do we need?"



Table of Contents

What This Shortcut Will Cover	3
A Brief History of Cheating	4
Defeating Piracy by Going Online	5
Or Not.....	6
The Lawyers Have Landed Bearing EULAs ...	7
The Rise of MMORPGs	9
The WoW Warden Is Watching	9
Cheating Is Quick and Easy	13
Grinding Is Boring and Dull	13
Farming Makes Things Easy	14
Virtual-World Economics	25
Farming Hurts the Virtual Economy	27
Games as Reality	28
Cracking Down on Farming	29
Online Game, Real-World Cheating	30
Defeating Cheaters and Crossing the Line ...	31
The Governor Watches the Watcher	32
Where Do You Stand?	43
The Tip of the Online Gaming Iceberg	44

Cheating Online Games

Gary McGraw and Greg Hoglund

SAMPLE

Cheating Online Games
by Greg Hoglund and Gary McGraw
ISBN 0321460723

That's it: a complete macro for automating the attraction and killing of monsters in *WoW*. If you run this macro, your character can accumulate experience and gold without human intervention.

Figure 2 shows a screenshot taken while the *WoW_Agro* macro is running. Notice the pile of dead monsters near the character. Note that running a macro such as this is against the rules and can cause you to be banned from the game. The *WoW* character's name shown in the code (and displayed on the screen in Figure 2), Xanier, is no secret. Blizzard banned Hoggund's Xanier account just before the character was to reach level 60. In this case, Hoggund was a little too flagrant with his cheating. Hoggund had spent more than \$400 on game-card registered accounts by this point. All the accounts were banned for various reasons.

FIGURE 2

A screenshot from *WoW* showing what happens when the agro macro runs. Virtual bodies pile up as the macro runs all night.



Time for another pesky question: Why is farming with a macro cheating? Some argue that Blizzard made the game boring in the first place, so fair is fair. However, there is more to it than this. To understand this, we need to understand a bit more about the MMORPG economy.

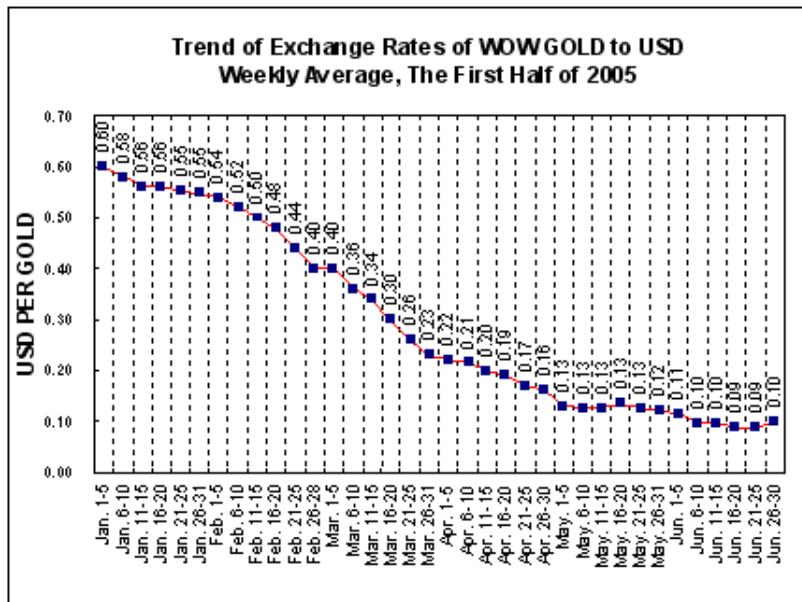
Virtual-World Economics

As previously mentioned, MMO games such as *WoW* are persistent worlds with thousands of interacting players. These games all have a form of play money that players use to buy and sell in-game equipment, items, and other virtual property. These virtual economies operate by the same complex rules as any real economy.

Most MMOs that are sizable have exchange rates between virtual gold and hard currency. The website GameUSD.com tracks exchange rates over time (<http://gameusd.com/world-of-warcraft-gold.htm>). Figure 3 shows exchange rates in the first half of 2005 for *WoW* gold to U.S. dollars.

FIGURE 3

Exchange rates between *WoW* virtual gold and U.S. dollars. This chart reproduced by permission from *WoW Gold Price Research* at GameUSD.com (<http://www.gameusd.com/>).



Academics have even turned to virtual economies as objects of research. In one early study, academic Edward Castranova determined that the gross domestic product of the MMO *EverQuest* exceeded that of many real countries, as reported in *Walrus* magazine:⁵

⁵ See *Walrus* magazine for a complete article on game economics at <http://www.walrusmagazine.com/article.pl?sid=04/05/06/1929205&mode=nested&tid=1>.

The Gross National Product of EverQuest, measured by how much wealth all the players together created in a single year inside the game ... turned out to be \$2,266 U.S. per capita. By World Bank rankings, that made EverQuest richer than India, Bulgaria, or China, and nearly as wealthy as Russia.

It was the seventy-seventh richest country in the world. And it didn't even exist.

Although Castranova's estimation of *EverQuest* GDP is probably too high, it is clear that the little worlds of online games do, in fact, create wealth out of nothing.

MMOs also suffer from economic instability and other macroeconomic problems just as real economies do. One of the earliest successful MMOs, *Ultima Online*, experienced massive inflation that led to a currency crisis in 1997. It turns out that the inflationary pressure came from an exploit that allowed players to “dupe” gold (that is, create copies of gold pieces from scratch with no work). Duping is akin to counterfeiting.

In any case, groups of academics are using MMOs for research purposes, focusing on which kind of economic model makes the most sense: centralized economy, free market, welfare state, and so on. It turns out that game designers and game minders make critical decisions regarding the economics of their games all the time (much like central bankers).

Game economies are certainly rich enough that they have spawned an entrepreneurial class of professionals. One early entrepreneur, Robert Kiblinger, was an *Ultima Online* player who went on to found UOtreasures (<http://www.uotreasures.com/>). UOtreasures is a classic middleman, buying virtual characters and items, and then later reselling them at a profit. The largest such outfit is *Internet Gaming Entertainment*, known to gamers as *IGE* (<http://www.ige.com/>). Interestingly, IGE runs its currency trading operations in Hong Kong. On their website, IGE claims, “Some experts



Your Short Cut to Knowledge

Using Microsoft® Windows Media® Player 11

Michael Miller

SAMPLE

Using Windows Media Player 11
by Michael Miller
ISBN 0768668638

CONTENTS AT A GLANCE

- Introducing Windows Media Player
- Important Changes to Windows Media Player 11
- Understanding Digital Media
- Comparing Digital Audio Formats
- Mastering the Windows Media Player Interface
- Skin Mode and the Mini Player
- Playing Music with Windows Media Player
- Managing Your Media Library
- Creating and Editing Playlists
- Burning Your Own Music CDs
- Syncing Music (and Other Files) to a Portable Audio Player
- Purchasing and Downloading Music from URGE
- Viewing DVDs, Videos, and Recorded TV Programs
- Managing and Viewing Digital Photos
- Enhancing Audio and Video Playback
- Adding More Functionality with Plug-Ins
- Configuring Windows Media Player



How Digital Media Works

In the world of computers and the Internet, all data is stored in digital format. (In a digital file, information is assembled from a series of 0 and 1 bits.) Digital is better than its counterpart (called analog) in that noise and distortion aren't introduced into the process. A digital copy, if recorded properly, can be an exact copy of the original. It's impossible to make an exact copy using analog methods.

In the audio and video world, music and movies used to be recorded and stored nondigitally. (Vinyl records and VHS videotapes are both analog formats.) However, with the advent of compact discs (CDs) and DVDs, music and movies began to be stored digitally.

When music and movies are digital, it's easy to incorporate them into the computer environment. To a computer, one digital file is pretty much the same as another. It's all the same to your PC whether you're saving or copying an MP3 music file or a Word document—they're both collections of bits and bytes.

Compatible Digital Media File Formats

Not all digital media is the same, however. Many different file formats store digital audio and video—and not all media players are compatible with all formats.

Windows Media Player 11 can read and write the majority of today's most popular digital media files. In fact, there are only a handful of media file formats in use today that WMP can't handle—chief among them are Apple's AAC and QuickTime formats, and RealNetwork's RealMedia files. Blame these incompatibilities on corporate politics; Apple's iTunes Music Player won't play back Microsoft's WMA files, and Microsoft's Windows Media Player won't play back Apple's AAC files. It's reciprocal incompatibility.

Table 1 details which digital media file formats are compatible with Windows Media Player 11.

How Digital Media Works

Table 1 Windows Media Player-Compatible Digital File Formats

Format	Type	File Extension(s)
Audio Interchange File Format	Audio	.AIF, .AIFC, .AIFF
CD Audio Track	Audio	.CDA
MPEG Audio Layer II	Audio	.MP2, .MPA
MPEG Audio Layer III	Audio	.MP3
Musical Instrument Digital Interface (MIDI)	Audio	.MID, .MIDI, .RMI
UNIX (Sun Microsystems and NeXT) Sound Files	Audio	.AU, .SND
Windows Audio	Audio	.WAV
Windows Media Audio	Audio	.WMA
Advanced Systems Format	Audio, Video	.ASF
Audio Visual Interleave	Audio, Video	.AVI
Advanced Stream Redirector	Metafile	.ASX
Windows Media Audio Redirector	Metafile	.WAX
Windows Media Redirector	Metafile	.WMX
Windows Media Video Redirector	Metafile	.WMV
M3U	Playlist	.M3U
Windows Media Playlist	Playlist	.WPL
Windows Media Player Skins	Skins	.WMZ, .WMS
Indeo Video Technology (Intel)	Video	.IVF

Comparing Digital Audio Formats

Table 1 Windows Media Player-Compatible Digital File Formats *continued*

Format	Type	File Extension(s)
Microsoft Digital Video Recording	Video	.DVR-MS
MPEG-1	Video	.MPEG, .MPG, .M1V
QuickTime (versions 2.0 and earlier)	Video	.MOV, .QT
Windows Media Video	Video	.WMV, .WM
Windows Media Download Package	Wrapper	.WMD

note

Playlist files contain lists of songs for use in assembling playlists; they don't contain the audio files themselves. Metafiles contain metadata, or information about other files. Skin files contain visual skins for Windows Media Player. And wrapper files contain a combination of audio/video files, metadata, and other data.

- QuickTime—versions 3.0 or later (.MOV, .QT)
- MPEG-4 (.MP4)

Of these noncompatible formats, the most problematic is AAC, which is the format used by Apple's iTunes Music Store and iPod portable audio player. If you download an AAC file from the iTunes Music Store, you can't play it with Windows Media Player. For that matter, if you have a WMA-format audio file in Windows Media Player, you can't play it on your iPod, or on the iTunes Music Player. As stated previously, Apple and Microsoft just don't play well together.

Incompatible File Formats

What file formats aren't supported by WMP 11? Here's a short list:

- Apple Advanced Audio Coding (.AAC)
- Free Lossless Audio Codec (FLAC)
- RealNetworks (.RA, .RM, .RAM)

Comparing Digital Audio Formats

Of all these different media file types, the ones you'll deal with most often are the digital audio formats. That's because you have



For More Information

For More Information

Pearson Technology Group is publishing new Short Cuts on a regular basis. To learn about new Digital Short Cuts, please visit www.informit.com/shortcuts.

Permission to Share

Know others who would like to view a Short Cut? Feel free to share this new product with friends and colleagues; just send the PDF along as an e-mail attachment.

STAY UP TO DATE ON NEW SHORT CUTS BY JOINING OUR SITE AND SUBSCRIBING TO THE "WHAT'S NEW" NEWSLETTER!

awprofessional.com

prenhallprofessional.com

quepublishing.com

samspublishing.com

ciscopress.com

informit.com