IBM PRESS

Getting Started with IBM WebSphere sMash

Ron Lynn, Karl Bishop,
Brett King

The authors and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

© Copyright 2011 by International Business Machines Corporation. All rights reserved.

Note to U.S. Government Users: Documentation related to restricted right. Use, duplication, or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corporation.

IBM Press Program Managers: Steven M. Stansel, Ellice Uffer
Cover design: IBM Corporation
Associate Publisher: Greg Wiegand
Marketing Manager: Kourtnaye Sturgeon
Acquisitions Editor: Katherine Bull
Publicist: Heather Fox
Development Editor: Kendell Lumsden
Managing Editor: Kristy Hart
Designer: Alan Clements
Project Editor: Lori Lyons
Copy Editor: Water Crest Publishing
Indexer: Lisa Stumpf
Compositor: Nonie Ratcliff
Proofreader: Apostrophe Editing Services
Manufacturing Buyer: Dan Uhrig

Published by Pearson plc
Publishing as IBM Press

IBM Press offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:

U.S. Corporate and Government Sales
1-800-382-3419
corpsales@pearsontechgroup.com.

For sales outside the U.S., please contact:

International Sales
international@pearson.com.

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both: IBM, the IBM logo, IBM Press, Tivoli, and WebSphere. Java, JavaScript, JDK, JRE, JVM, and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both. Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both. Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both. Other company, product, or service names may be trademarks or service marks of others.

Library of Congress Cataloging-in-Publication Data

Lynn, Ron.
Getting started with IBM Websphere sMash / Ron Lynn, Karl Bishop, and Brett King.
p. cm.
Includes index.
ISBN-13: 978-0-13-701970-0
ISBN-10: 0-13-701970-X
1. Web site development. 2. WebSphere. I. Bishop, Karl. II. King, Brett, 1968- III. Title.
TK5105.8885.W43L96 2010
006.7--dc22

2010030753

All rights reserved. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, write to:

Pearson Education, Inc.
Rights and Contracts Department
501 Boylston Street, Suite 900
Boston, MA 02116
Fax (617) 671 3447

ISBN-13: 978-0-13-701970-0
ISBN-10: 0-13-701970-X

Text printed in the United States on recycled paper at Courier-Westford, Westford, Massachusetts.

First printing September 2010

Thank you to all the developers out there who are tired of complex and cumbersome development, and those who had enough desire to do something different and, dare we say it, fun. We wrote this book for you. We hope you enjoy developing in WebSphere sMash using Groovy, PHP, normal Java, and JavaScript with Dojo. None of us can stand to write classic JEE applications anymore. If that doesn't speak to the value of this product, nothing does.

Contents

Introduction

IBM® WebSphere® sMash is a platform for the rapid development and deployment web applications using popular web technologies. It quickly enables developers to go from concept to production in a fraction of the time required by traditional platforms and web application models. Developers can use the dynamic scripting languages Groovy and PHP to speed development and still have Java™ as the underlying system language for extension development. Add to this easy development of Representational State Transfer (REST) services and rich AJAX interfaces. This is a platform that excels at quickly getting Web 2.0 applications built. To speed deployment, the runtime environment is integrated, so there is no "server environment" to deploy to. Applications can be built to run standalone on any machine. This makes IBM WebSphere sMash an ideal platform for the development of situation applications.

Situational Applications

Situational applications are defined by Wikipedia as follows:

A **situational application** is "good enough" software created for a narrow group of users with a unique set of needs. The application typically (but not always) has a short life span, and is often created within the group where it is used, sometimes by the users themselves. As the requirements of a small team using the application change, the situational application often also continues to evolve to accommodate these changes. Although situational applications are specifically designed to embrace change, significant changes in requirements may lead to an abandonment of the situational application altogether—in some cases, it is just easier to develop a new one than to evolve the one in use.

Given the constantly changing business landscape, situational applications provide the ideal instrument for constant adaptation. As a platform for situational application, WebSphere sMash provides an ideal mix of features and flexibility allowing for rapid application development.

Rapid Application Development

The use of dynamic scripting languages is well acknowledged to improve software developer productivity. Groovy and PHP both have many build-in and reusable components and allow developers to create function with fewer lines of code. The use of dynamic scripting languages along with the integrated runtime environment also speeds the develop-compile-test loop by removing the need for compilation. Developers need to make changes only to their scripts, and IBM WebSphere sMash detects and executes the changed code. This sort of rapid development means that applications can be continuously tested and available to testers or end users even while an application is still being developed, extended, or evolved. Applications can also be

rapidly changed as the situation dictates. This type of continuous application adaptation fits the constantly changing business landscape perfectly.

IBM WebSphere sMash Development Process

IBM WebSphere sMash has been developed using a community-driven commercial development process. This means that most of the development efforts are done in the open and are transparent to all parties through a community-based website. The community website hosts the IBM WebSphere sMash incubator project, Project Zero. Project Zero is the technology for which the IBM WebSphere sMash product is derived. IBM encourages everyone to participate by contributing to the open discussions around the product development, features, functions, and bugs. The community website provides tools to facilitate participation. The tools include forums, a wiki, a bug-reporting tool, a developers blog, and access to the source repository. You can even download the latest stable and experimental versions of the product. This begs the question: Which version should I use?

Available IBM WebSphere sMash Offerings

IBM WebSphere sMash is available in four different offerings. The offerings range from fully supported with an IBM commercial license and extended features to bleeding-edge nightly builds of the latest and greatest code. The four offerings are summed up in Table 1.1.

Table 1.1 IBM WebSphere sMash Offerings

Offering	Features	Availability
IBM WebSphere sMash	Stable, production version of WebSphere sMash with a Standard IBM Commercial license.	Available for purchase from www.ibm.com/software/webservers/smash/
IBM Reliable Transport Extension for WebSphere sMash	Stable, production version of WebSphere sMash with a Standard IBM Commercial license. This version extends WebSphere sMash with features for reliable messaging and communication.	Available for purchase from www.ibm.com/software/webservers/smash/
IBM WebSphere sMash Development Edition	Stable, community version of WebSphere sMash with a license that allows for development and limited deployment.	Available for free from www.projectzero.org/download/
Project Zero	Experimental version of WebSphere sMash with features not yet available in the stable versions.	Available for free from www.projectzero.org/download/latest.php

From this list of offerings, it should be fairly easy to match up what level of investment you're looking for. Most developers start with the IBM WebSphere sMash Development Edition unless they're looking for the cutting-edge enhancements or would like to peek into the future. When the developers have established that WebSphere sMash is their preferred platform for development, the business would then invest in the production-ready and supported versions.

What Is Covered in This Book?

This book covers a wide range of topics of interest to software developers. The intent of this book is to supplement and expand upon the information available on the projectzero.org website and in the product documentation. Each chapter includes concrete examples and demonstrations of how to use the technology presented. You can find much of the code from this book at the IBM Press Books website: ibmpressbooks.com/sMash. We open the book by discussing installation and development environments. From our own experience as software engineers, we see that these are important aspects for the adoption of any new development environment. After we learn about installation and the available development environments, we move into a chapter that outlines a complete application. This chapter is intended to stand by itself and give you a quick introduction to the complete WebSphere sMash development cycle. After that, we dive into the details of many different aspects of WebSphere sMash. This book is by no means comprehensive. We've included what we believe to be important aspects to get you going and give hints into where there's more to be uncovered. We hope you enjoy this book and find it to be useful in learning WebSphere sMash.

This page intentionally left blank

CHAPTER 1

Installing the IBM WebSphere sMash CLI

In the Introduction, we talked about the four different IBM WebSphere sMash offerings. For the purposes of this book, any version should be appropriate. So, feel free to download the latest Project Zero build or the latest IBM WebSphere sMash Developer Edition. Before we do that, though, let's prepare our Java development environment.

First Things First: Java Development Environment

Prior to installing the sMash CLI, you must ensure that you have a functional Java SE Development Kit (JDK™) version 5 or 6 installed. It is important to remember that you cannot use the Java SE Runtime Environment (JRE™) to develop with sMash. The AppBuilder development environment requires libraries found only in the JDK and not in the JRE. This is a common problem with first-time users. You may choose to use either the JDK from Sun or IBM. They can be found in the locations in Listing 1.1.

Listing 1.1 Locations of Supported JDKs

```
IBM: http://www.ibm.com/developerworks/java/jdk/
Sun: http://java.sun.com/javase/downloads/index.jsp
```

After you have downloaded a JDK, install it per the instructions provided for your operating system. The following steps show the configuration using the Linux® command line. For Windows® users, you need to open a command prompt and alter the statements appropriately. When installed, you can verify your JDK installation by running the commands in either Listing 1.2 or Listing 1.3.

Listing 1.2 Linux Java Version Check

```
$ java -version
java version "1.6.0"
Java(TM) SE Runtime Environment (build pxi3260sr3-20081106_07(SR3))
IBM J9 VM (build 2.4, J2RE 1.6.0 IBM J9 2.4 Linux x86-32 jvmxi3260-
20081105_25433 (JIT enabled, AOT enabled)
J9VM - 20081105_025433_lHdSMr
JIT  - r9_20081031_1330
GC   - 20081027_AB)
JCL  - 20081106_01
```

Listing 1.3 Windows Java Version Check

```
C:\>java -version
java version "1.6.0"
Java(TM) SE Runtime Environment (build pwi3260sr2-20080818_01(SR2))
IBM J9 VM (build 2.4, J2RE 1.6.0 IBM J9 2.4 Windows XP x86-32
jvmwi3260-20080816
_22093 (JIT enabled, AOT enabled)
J9VM - 20080816_022093_lHdSMr
JIT  - r9_20080721_1330ifx2
GC   - 20080724_AA)
JCL  - 20080808_02
```

The output shown in the listings indicates that I am running the IBM JDK version 1.6.0 on a Linux x86 platform. If running the Sun JDK, a similar output will be shown. The important thing is that you get JDK version information back and not some random error. Now that we have the JDK installed and have ensured that it is working, we can move on to downloading and installing WebSphere sMash.

Installing the Command-Line Interface

The installation for IBM WebSphere sMash is very easy. At a high level, you simply need to download the zip file and unzip it, set up a proxy server if needed, and test the installation. IBM WebSphere sMash is highly modular. It uses the ivy dependency manager (see http://ant.apache.org/ivy/) to manage dependencies for each application. Every application has tremendous flexibility to include only the modules the application needs. Developers also have the ability to create reusable modules for other applications to use.

For the purposes of this book, it is fine to download the latest stable WebSphere sMash Development Edition driver or the experimental Project Zero driver. You can download either from the Project Zero website (see Figure 1.1).

Figure 1.1 Project Zero website

The website, projectzero.org, will change over time, but you can find the download link to retrieve the zip file from the site. Remember, it is http://projectzero.org.

After you have the zip file on your computer, you need to unzip it using whatever extraction tool you have available. Unzip the contents into a directory of your choosing. The zip file has a base of zero, which you want to retain. In these samples, I am using a top-level directory called development located under my home directory. In the following, replace the location shown with your preferred directory:

```
unzip ~/Downloads/zero_<version>.zip -d ~/development/
```

Of course, if you're using a different operating system, your unzip command and paths will look different. After the files are extracted, you should end up with a directory structure similar to that shown in Figure 1.2 under your target directory.

The final step to finish the installation procedure consists of setting up your environment to include the location to the JRE and the Project Zero binaries. This will be accomplished by manipulating the PATH environment variable appropriately. Placing these commands in your normal login environment ensures that everything is ready to go the next time you log in. This is an optional step, but will save you the hassle of having to specify full paths when running the Zero command line. Examples of these paths for Linux are shown in Listing 1.4.

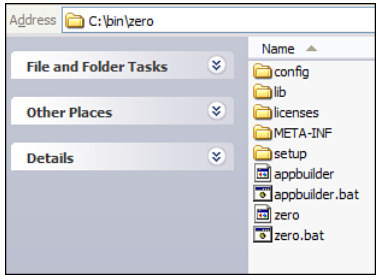

Figure 1.2 Zero directory structure

Listing 1.4 Example Linux Paths

```
export JAVA_HOME=/usr/lib/jvm/java-1.5.0-ibm-1.5.0.8/jre
export ZERO_HOME=~/development/zero
export PATH=$JAVA_HOME/bin:$ZERO_HOME:$PATH
```

For Windows users, you can set the environment variables by accessing the Control Panel > System > Advanced tab > Environmental Variables. Then, in either the user or system areas, add/edit the following values, as shown in Listing 1.5. Be sure to adjust the directory paths to match your actual environment.

Listing 1.5 Example Windows Paths

```
JAVA_HOME    C:\Program Files\Java\jdk1.6.0_16\jre
ZERO_HOME    C:\development\zero
PATH         %JAVA_HOME%\bin;%ZERO_HOME%\bin;%PATH%
```

This completes the basic installation of IBM WebSphere sMash. The next section discusses setting up proxy support. However, if your connection to external websites doesn't require you to use a proxy, skip the proxy section and continue to the section on testing your installation.

Activating HTTP(S) Proxy Support

The command-line interface (CLI) retrieves modules from remote repositories. For the command-line interface to do this, Java needs access the Internet. If you use a proxy server to access external websites, you need to set up the proxy server for IBM WebSphere sMash. If you don't know if you use a proxy server, skip this section and try testing your WebSphere sMash installation. If you receive an unknown host error message, you probably need to come back to this section and set up your proxy server. The easiest way to do this is to edit the zero script file for Linux or zero.bat file for Windows in the installation directory.

Open the script file or the batch file that is appropriate for your platform in your favorite text editor. At the top of the file, there are comments that direct you to uncomment particular lines and update the values (see Listing 1.6 for Linux and Listing 1.7 for Windows).

Listing 1.6 Linux Zero Script Fragment Demonstrating Proxy Configuration

```
# proxy support -
#    uncomment the following line( and the export ZERO_OPTS) and
#    update proxy values
# ZERO_OPTS="$ZERO_OPTS
       -Dhttp.proxyHost=myProxyHost
       -Dhttp.proxyPort=myProxyPort
       -Dhttps.proxyHost=myProxyHost
       -Dhttps.proxyPort=myProxyPort"
# export ZERO_OPTS
```

For Linux, uncomment by removing the # sign at the start of the ZERO_OPTS line and the start of the export line. Set the proxy variables to the appropriate values, and your proxy is set up.

Listing 1.7 Windows Zero Batch File Fragment Demonstrating Proxy Configuration

```
rem proxy support -
rem   uncomment the following line and
rem   update values to add proxy settings.
rem set ZERO_OPTS=%ZERO_OPTS%
-Dhttp.proxyHost=myProxyHost
-Dhttp.proxyPort=myProxyPort
-Dhttps.proxyHost=myProxyHost
-Dhttps.proxyPort=myProxyPort
```

For Windows, uncomment by removing the rem at the start of the set ZERO_OPTS. Set the proxy variables to the appropriate values, and your proxy is set up.

Now that your proxy is set, we can continue on and sanity test our IBM WebSphere sMash installation.

Test Your IBM WebSphere sMash Installation

The quickest way to verify that your IBM WebSphere sMash installation is installed correctly is to run a version check. To do this, open a command line and change directories to the zero installation directory. In this directory, there is a script for Linux named zero and a batch file for Windows named zero.bat. On the command line, type zero version. If your installation has been done properly, you should see output similar to Listing 1.8.

Listing 1.8 Output from the "Zero Version" Command on Windows

```
C:\temp\zero>zero version
CWPZT0901I: The following module(s) are not currently in the local
  repository:
        zero:zero.cli.tasks:[1.0.0.0,)
CWPZT0902I: Trying to locate the module(s) using one of the config-
  ured remote repositories
CWPZT0545I: Retrieving zero.cli.tasks-1.1.1.1.30731.zip from host
http://www.projectzero.org/sMash/1.1.x/repo/base
CWPZT0545I: Retrieving zero.kernel-1.1.1.1.30725.zip from host
http://www.projectzero.org/sMash/1.1.x/repo/base
CWPZT0600I: Command resolve was successful
Command-line Version:  1.1.1.1.30719 20090908 2152
Command-line Information:
        Name: zero.cli
        Version: 1.1.1.1.30719
        Location: C:\temp\zero
        Modulegroup: stable

        Dependencies:
                zero:zero.cli.tasks:1.1.1.1.30731 (userhome)
                zero:zero.kernel:1.1.1.1.30725 (userhome)

The java command path is
C:\bin\ibm-java-sdk-60-win-i386\jre\bin\java.exe

java version "1.6.0"
Java(TM) SE Runtime Environment (build pwi3260sr2-20080818_01(SR2))
IBM J9 VM (build 2.4, J2RE 1.6.0 IBM J9 2.4 Windows XP x86-32
jvmwi3260-20080816_22093 (JIT enabled, AOT enabled)
J9VM - 20080816_022093_lHdSMr
JIT  - r9_20080721_1330ifx2
GC   - 20080724_AA)
JCL  - 20080808_02
```

This listing came from a Windows-based machine, so the output will look slightly different than what you might see on a Linux machine. Also, the versions and paths will be appropriate for the version of IBM WebSphere sMash you downloaded, the version of Java you're using, and the paths you have selected for installation.

You should notice at the top of the output that IBM WebSphere sMash automatically retrieves certain modules from the remote repository. Namely, it retrieves the CLI tasks and the zero kernel modules. These modules are necessary for IBM WebSphere sMash to run and, if successfully downloaded, indicate that you have connectivity to the remote repository.

If you receive errors during the download of these modules, you need to fix those errors before moving on. An easy test to see if you have connectivity to the repository is to take the URL listed in the output in Listing 1.8 and plug it into your web browser. In Listing 1.8, the URL we'd use is the following:

```
http://www.projectzero.org/sMash/1.1.x/repo/base
```

This URL will be different depending on the version of IBM WebSphere sMash you are using. If you have access to the URL from your web browser, but see errors using the zero version check, you probably need to go back to the proxy section and set up a proxy server for IBM WebSphere sMash to use.

When the zero version check works, you can get started using the command-line interface.

Getting Started with the Command-Line Interface (CLI)

In the next chapter, we look at various development environments, including the command-line interface (CLI). Let's take a quick look at the CLI now just to get acquainted with it. As we saw, the CLI is a module retrieved from a remote module repository. There are many commands available, but don't let this scare you. The CLI is very helpful. To get a list of available commands, type the following:

```
zero help
```

The results of typing this into the command line can be seen in Listing 1.9.

Listing 1.9 Results of the "Zero Help" Command

```
Usage:  zero [main-opts] <task> [task-args]
main-opts:
  -d            enable minimal logging
  -v            enable all logging
  -l=<file>     use the given file to output log to

compile         Compiles all Java source files under the module's
/java           directory.
create          Creates a new module in the current directory.
help            Prints the documentation for the given command.
modulegroup     Manage modulegroups.
package         Packages the module as a zip file.
```

```
publish          Publish the module to your local repository.
repository       Manage repositories.
resolve          Determine the module's dependencies.
rollback         Reverts the effects of the last resolve or update.
search           Finds and prints all matching modules in the reposi-
                 tory.
switch           Switch the module group.
update           Resolves a module to its latest dependencies.
version          Displays version information
```

As you can see, there are a bunch of available commands. Each command may have arguments that it can be given, but again the command line is very helpful. You need only to type the following to get help on a particular command:

```
zero help <command>
```

For example, if we use zero help search, we get the results shown in Listing 1.10.

Listing 1.10 Results of "Zero Help Search" in the CLI

```
Usage:

zero search <org:module[:revision]> [-remote] [-json]

The command uses the current module group to find in the local re-
pository for a module that matches the given <org:module:revision>
value and prints its search results. If no module name is given, the
command will print information about all available modules. You can
also use the -remote option to tell the command to search the remote
repository instead of the local one.

If the -json option is used the report is formatted as a JSON docu-
ment.

The return codes for this command are:
0 - success
1 - command failed
```

The search command enables you to search for modules in the repository. There are commands to create new IBM WebSphere sMash applications, start and stop applications, package applications, and many more. Spend some time familiarizing yourself with the commands, and we'll take another look at the CLI in the next chapter.

Conclusion

In this chapter, we learned how to install IBM WebSphere sMash and how to test that installation. We also started working with the command-line interface (CLI) and became familiar with the available commands. In the next chapter, we're going to take a look at three different development environments that you can use to develop IBM WebSphere sMash applications. The first is a web browser-based environment called AppBuilder that is built into IBM WebSphere sMash. The second is an Eclipse-based development environment. Finally, we'll come back to the CLI and take a look at what else it can do.

Index

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