Volume

1

NETSPECTIVE CORPORATION

Corporate Publications and Documentation

Getting Started With Sparx

Chapter 1

Getting Started With Sparx

Target Audience

This document is targeted at individuals, such as yourself, who have come to know of Sparx and are curious to see its capabilities in practical terms. This document is a map showing you where to look for information related to Sparx. It is also a guide that helps you get started on the road to making sparks of your own. This includes information about downloading, installing and using the free 30 day Sparx evaluation kit available from the Netspective Corp. web site as well as pointers to Sparx reference documentation available online.

Last, but certainly not least, this document includes information targeted at individuals who wish to evaluate Sparx online without downloading an evaluation kit or setting up any servers.

Discovering Sparx

The main source of information about Sparx is the Netspective Corp. web site. There you will find both technical and non-technical descriptions of how Sparx can help you achieve higher productivity by spending more time developing your application and less time developing a framework.

A few of the more important items on the Netspective web site are listed and described below. The URL at the end of each item's description will lead you to its location on the Netspective web site.

- **The Sparx Architecture**. Learn more about the design of Sparx and how it helps developers do more in less time. Find out what makes the Sparx Application Platform so different from mere "forms builders". `http://www.netspective.com/web-netspective/products`

- **Sparx Services**. Find out what Sparx can offer you as a developer or technical manager. `http://www.netspective.com/web-netspective/services`

- **Sparx Support**. These are online resources for developers who use Sparx for enterprise development or who are interested in evaluating Sparx for themselves. `http://www.netspective.com/web-netspective/support`

Evaluating Sparx

Once you have discovered what Sparx can do for you, it is time to see it in action. Netspective Corp. allows you to evaluate Sparx in two different ways. These are listed below along with the pros and cons of each.

The first (and recommended) way is by downloading a free 30 day evaluation kit. This kit contains everything you need to start getting familiar with Sparx and learning how to develop applications powered by it. The main advantage of using this method is that you will get a good hands-on look at Sparx and, by the time you are through with the documentation and tutorials, you will be well on your way to developing solid applications using Sparx. The disadvantage of this method is that before downloading and installing the Sparx evaluation kit, you need to ensure that your development environment satisfies all the pre-requisites for using the kit. These pre-requisites include setting up a J2EE application server and an Oracle database server. Further, if you will be developing applications using Sparx, a good Java IDE is strongly recommended.

The second way to evaluate Sparx is to view all the documentation online and follow along with the online versions of the applications created in the tutorials. The main advantage this method offers you is that it is quick and simple. Since there are no pre-requisites as such, you can start learning about Sparx within minutes. The disadvantage is that you will not have the ability to follow along with the tutorials and dabble in the application code yourself.

If you choose to do an online evaluation, continue on to the section titled Making Sparx Fly Online. If you choose to download and setup a Sparx evaluation kit on your PC, however, continue on to the section titled Making Sparx Fly on your PC

Making Sparx Fly Online

An online evaluation is only different from a regular evaluation in the amount of interaction you have with Sparx. While the regular evaluation will enable you to re-create all the demonstration applications that come with the evaluation kit, the online version will allow you to only see the final products. In both cases, however, you will be able to follow up with the tutorial and see how the final form of each application is reached.

The only thing you will need to keep in mind while following the tutorials is that all URLs listed in the documentation that are supposed to be pointing to the applications developed in the tutorials will be different. The general rule is that for an application named appName, the URL for its online version should look like the following URL: http://developer.netspective.com:8089/appName.

Therefore, if you are following the development of the Hello World application, instead of using your browser to access the URL mentioned in the documentation (`http://localhost:8089/hello`), you should point your web browser to the following URL: `http://developer.netspective.com:8089/hello`. Similarly, when following the development of the Sparx Collection, you should point your web browser to `http://developer.netspective.com:8089/library`. Finally, when perusing the tutorials regarding the development of Cura (the project management application), you should go to `http://developer.netspective.com:8089/cura`.

Making Sparx Fly on your PC

Pre-Requisites

Since Sparx is an application framework for J2EE application servers, a fundamental requirement to develop applications with it (and with a J2EE application server) is a Java SDK. You can obtain Sun's official Java SDK for Windows from Sun's Java web site at `http://java.sun.com/j2se/1.3/download.html`. This is a link to the Java 1.3.1_03 SDK.

Alternative Java SDK packages include IBM's Java SDK. The Jikes compiler (also developed at IBM) complements a standard Java SDK with rapid compile speeds and much more informative error messages than the default compilers. You can find IBM's Java SDK at `http://www7b.boulder.ibm.com/wsdd/wspvtdevkit-info.html`. IBM's Jikes compiler for Java can be found at the following URL: `http://oss.software.ibm.com/developerworks/opensource/jikes/`.

It is important to understand that **you only need to install one Java SDK** to run the Sparx evaluation kit. This can either be the Sun Java SDK or the IBM Java SDK or any other variety that you might prefer, as long as it complies with Java 1.3.

You can also optionally install a complete Java integrated development environment (IDE) for developing Sparx applications using the evaluation kit. Installation of such a development environment is, of course, strictly a personal preference and is completely **optional**. Sparx does not require nor endorse any particular Java IDE but will work with any IDE that you might wish to use while developing applications.

Easy to use and freely available IDEs include products from IBM and Borland and many others. These include products such as IBM's VisualAge for Java and JBuilder Community Edition. Such IDEs greatly enhance the development experience by

providing numerous small aids ranging from simple syntax highlighted editors to complex class browsers. As stated earlier, the use of such an IDE is not needed but is highly recommended. The VisualAge for Java IDE can be found at the following URL: `http://www7.software.ibm.com/vad.nsf`. JBuilder CE is available at `http://www.borland.com/jbuilder/offers/`.

Additionally, if you are planning on evaluating Cura, the project management application that is bundled with the evaluation kit, you will need to install Oracle (version 8 or newer) on a server and make it available for access by the server that will have the Sparx evaluation kit installed. You can choose to install Oracle on the same server as the Sparx evaluation kit provided it has the resources to handle both an application server and a database server.

Installing the Evaluation Kit

Having made sure your development environment is capable of running a J2EE application server and having installed a Java SDK of your choice, you are now ready to download and install the Sparx evaluation kit.

Navigate to the Sparx Support page on Netspective Corp's web site and download the evaluation kit. The evaluation kit comes in the form of a Java JAR file. Save this JAR file into a directory of your choice. Now drop to a command prompt and navigate to the directory containing this JAR file. From here, installation is a matter of executing the command `java -jar sparx-eval.jar` and following the instructions in the automated installer.

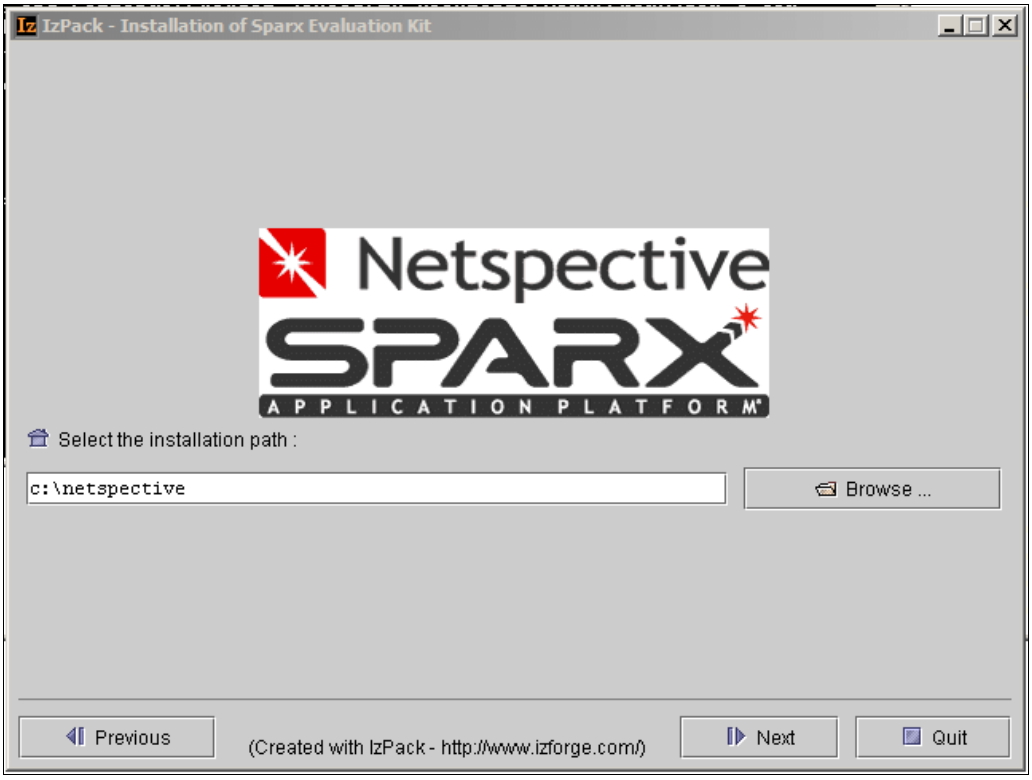

Figure 1: Choosing A Directory

The automated installer will prompt you twice: once to get the path where you want to install the evaluation kit and the second time to get the port you want to use for the application server that comes as part of the evaluation kit. The path you give in the first answer will contain the Resin application server[1], the Sparx framework, and two sample applications. The fully qualified name of the directory where the Sparx framework is installed (under the installation directory) will be known throughout all Sparx documentation as your Sparx Home. It is highly recommended that you set the environment variable SPARX_HOME to point to this path so that the tools supplied with Sparx do not have to make intelligent (and possibly wrong) guesses at the value of your Sparx Home.

[1] The Resin Application Server (`http://www.caucho.com`) is not a part of Sparx. It is free for non-commercial use. You must license Resin for commercial use if you wish to deploy it in a production environment. For a free alternative to Resin, please visit `http://jakarta.apache.org/tomcat/index.html` for the Tomcat Application Server.

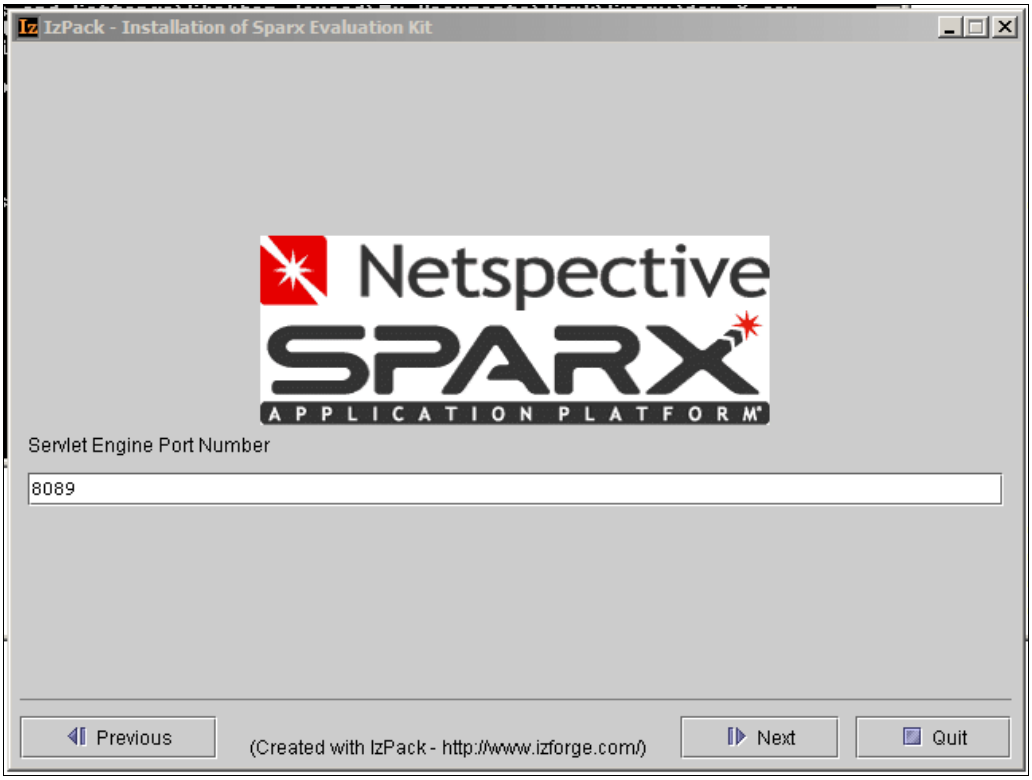

Figure 2: Choosing A Port

The port number you gave in the second answer will be the port you will use whenever you want to use or test an application running on that installation of the Sparx evaluation kit. Thus, if you leave the port number as the default 8089, you can test that the application server is up and running by opening up a web browser and accessing the URL `http://localhost:8089/`.

Starting Resin

Once the installer exits, you need to ensure that the Resin application server is started before attempting to access any Sparx application. There are two main ways of starting Resin. The first is as a foreground application that you manually start and leave running while you are working with Sparx. The second is as a Windows service or a UNIX daemon.

You can start Resin in the foreground by opening up My Computer under Windows and navigating to the evaluation kit's installation directory (`c:\netspective` by default). Navigate to the `resin\bin` directory under this installation directory and double-click on the executable named `httpd.exe`. This should start Resin in foreground mode.

If you do not want the trouble of manually starting Resin every time you want to work with Sparx, you can start it as a Windows service (or UNIX daemon) by following the instructions on Resin's home page at `http://www.caucho.com`.

Starting the Fire

With a successful installation of the evaluation kit under your belt, you can move on to more important matters such as playing with Sparx. You can get an idea of the kind of applications that can be developed with Sparx by taking a look at the sample applications that come with the evaluation kit.

You should be able to access the first application – a very simple Hello World application – by opening a web browser to the following URL: `http://localhost:8089/hello`. This is, of course, assuming you chose 8089 as the port number for your application server. If this is not the case, you should substitute 8089 with the port number you chose while installing the evaluation kit. Further, if you opted for an online evaluation, you can see the Hello World application by pointing your web browser to the following URL: `http://developer.netspective.com:8089/hello`.

The second application, the Sparx Collection, is a more extensive one and is a model of a library of books from which you can add, edit or delete any number of books. This can be accessed at the URL `http://localhost:8089/library`. For online evaluators, the Sparx Collection can also be found at the following URL: `http://developer.netspective.com:8089/library`.

Using the Source

If you opted for a downloadable evaluation kit, you already have the source code for all these applications and can easily open all the files up to get a better understanding of the application. However, if you opted for an online evaluation, you are still in luck. You can review the source code for each application by going to the application's ACE (Application Component Explorer) and choosing "Application" from the Documents menu. The first time you enter the application documentation section, you should see a list of all the directories that exist in the application's root directory. You can click on any one to navigate to it and view the list of files and sub-directories inside it. If you click on any XML, JSP or Java source file, you should also be able to see the source for those files directly from the browser.

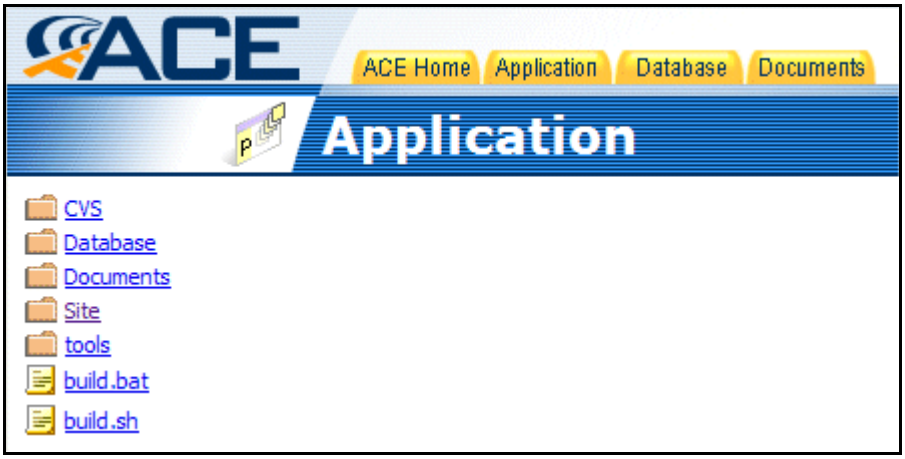

Figure 3: ACE showing directory structure of an application

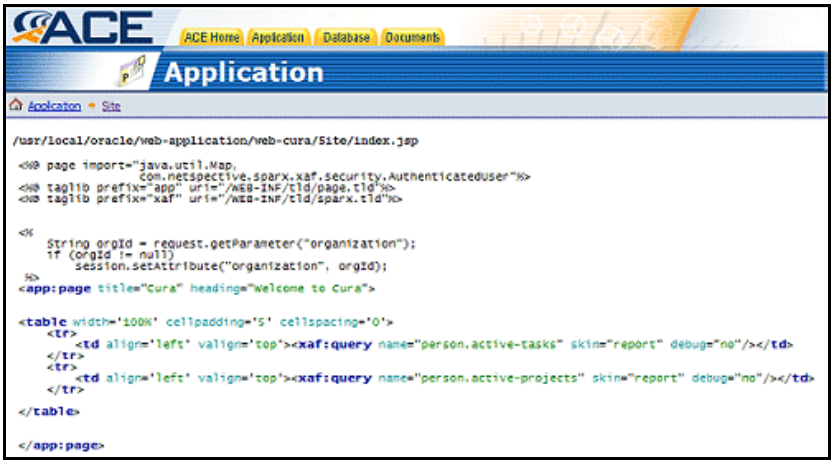

Figure 4: ACE showing how to view a JSP file

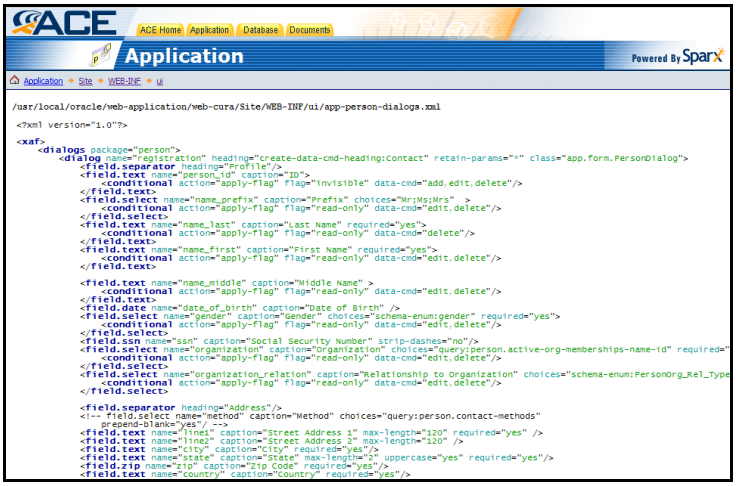

Figure 5: ACE showing how to view an XML file

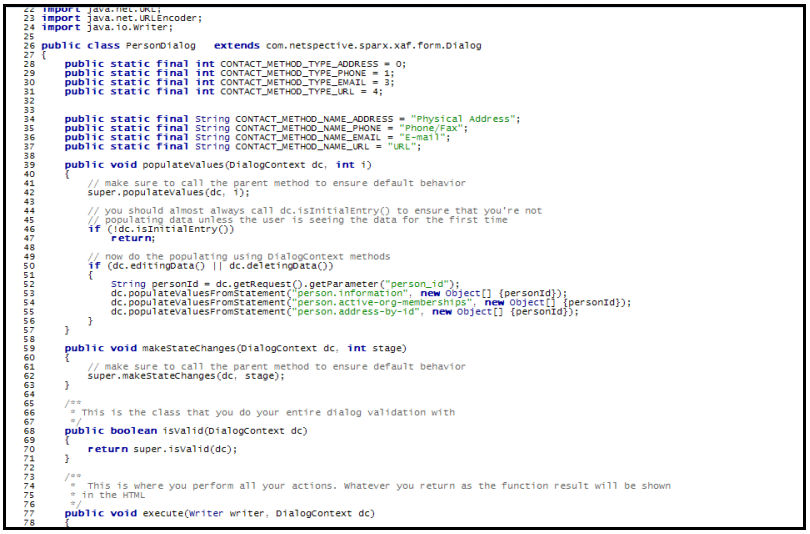

Figure 6: ACE showing how to view a Java file

As the screenshots have already shown, it is extremely easy to navigate the entire source tree for an application. Further, when viewing any of the source files of the application, they are presented in an extremely easy to use form complete with line numbers and syntax highlighting.

While you're there, you can also access the JavaDoc documentation of the complete Sparx API by choosing "Sparx" from the Documents menu in ACE. Details on how to access an application's ACE are present in the tutorials of Volumes 2 and 3.

Quenching the Thirst

Once you have seen these very elementary examples of Sparx's capabilities, you can proceed to the Sparx Tutorial. This tutorial not only shows you the basics of application development with Sparx but also takes you, step by step, through the creation of both the Hello World and the Sparx Collection applications. This includes application design, implementation and elaborate explanations. You can find the Sparx Tutorial in the `documentation` directory immediately under your Sparx Home.

You should go through the tutorial and continue to experiment with Sparx by developing your own applications and observing the development time go from days to hours for small applications and from months to weeks for even enterprise level applications.

After your evaluation period is over, you can contact Netspective Corp. regarding licensing for commercial application development.

Appendix A

Sparx on Other J2EE Servers

Sparx is fully compliant with the J2EE specification. However, the various server vendors have custom extensions to the standard J2EE Deployment descriptors and getting your Sparx application up and running requires entries in these descriptors. Also, your Sparx application must be configured as a full J2EE application, with your Sparx web applications configured within the larger application context. The following configurations are for a "bare-bones" Sparx web application, using the web-library example app as a guide. You may, of course, need to add additional information to your descriptors to configure other aspects of you application.

WebLogic 6.1

1. In the application root directory `../web-library` create a `META-INF` directory.

2. Create the standard J2EE `application.xml` file in the `META-INF` directory as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE application PUBLIC '-//Sun Microsystems, Inc.//DTD
J2EE Application 1.2//EN'
'http://java.sun.com/j2ee/dtds/application_1_2.dtd'>
<application>
  <display-name>Web Library</display-name>
  <module>
    <web>
      <web-uri>Site</web-uri>
      <context-root>/library</context-root>
    </web>
  </module>
</application>
```

   This maps your Web Application Root (web-uri) to a URL (context-root).

3. Create an empty Weblogic `weblogic-application.xml` file in the `META-INF` directory as follows:

```
<?xml version="1.0" encoding="ISO8859_1"?>
```

```
<!DOCTYPE weblogic-application PUBLIC '-//BEA Systems,
Inc.//DTD WebLogic Application 7.0.0//EN'
'http://www.beasys.com/j2ee/dtds/weblogic-
application_1_0.dtd'>
<weblogic-application>
  <ejb>
  </ejb>

  <xml>
    <parser-factory>
    </parser-factory>

  </xml>

  <security>
  </security>

</weblogic-application>
```

4. Create the `web.xml` file in the `WEB-INF` directory as follows:

```
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems,
Inc.//DTD Web Application 2.2//EN"
"http://java.sun.com/j2ee/dtds/web-app_2_2.dtd">
<web-app>

  <display-name>Web Library Application</display-name>

  <context-param>
    <param-name>default-data-source</param-name>
    <param-value>jdbc/library</param-value>
  </context-param>

  <context-param>
    <param-name>app-exec-environment</param-name>
    <param-value>Development</param-value>
    <description>Setup Execution Environment, one of
(Production,Development, or Testing)</description>
  </context-param>

  <servlet>
    <icon>
      <small-icon></small-icon>
      <large-icon></large-icon>
    </icon>
    <servlet-name>AppComponentsExplorer</servlet-name>
    <servlet-class>
com.netspective.sparx.ace.AppComponentsExplorerServlet
</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>AppComponentsExplorer</servlet-name>
    <url-pattern>/ace/*</url-pattern>
  </servlet-mapping>


  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
  </welcome-file-list>

  <taglib>
    <taglib-uri>app</taglib-uri>
    <taglib-location>WEB-INF/tld/page.tld</taglib-
location>
```

```
  </taglib>

  <taglib>
    <taglib-uri>xaf</taglib-uri>
    <taglib-location>WEB-INF/tld/sparx.tld</taglib-
location>
  </taglib>

  <resource-ref>
    <description></description>
    <res-ref-name>jdbc/library</res-ref-name>
    <res-type>javax.sql.DataSource</res-type>
    <res-auth>Container</res-auth>
  </resource-ref>

</web-app>
```

5. Create the Weblogic `weblogic.xml` file as follows:

```
<weblogic-web-app>

  <reference-descriptor>
    <resource-description>
      <res-ref-name>jdbc/library</res-ref-name>
      <jndi-name>sparx-dataSource-hsqlPool</jndi-name>
    </resource-description>
  </reference-descriptor>
</weblogic-web-app>
```

This example assumes that a static DataSource **sparx-dataSource-hsqlPool** has been configured via the Weblogic console. Please refer to the Weblogic documentation for specifics.

NOTE: The web-library example (and Hypersonic SQL in general) must be run in client/server mode in order to function properly as a static datasource within the WebLogic server. The runServer.bat file in the `web-library/Database` directory will start the Hypersonic SQL server. The Hypersonic SQL URL for client/server access is `jdbc:hsqldb:hsql://hostname`.