



ZimWeb

Documentation

Extend the Entity Power of Zim to the Internet

Copyright © 2003, ZIM Technologies International Inc.

Table of Contents

README	4
WELCOME	5
Contacting Us	5
Trademarks and Copyright Information	6
ABOUT ZIMWEB	8
RELEASE NOTES	9
WHAT IS ZIMWEB	14
An Introduction	14
ZimWeb Architecture	15
Features	17
Compatibility with the ZimCGI	17
Ability to handle POST and GET requests	17
Debug information	17
HTTP session and cookie parameter management	17
Access to key information about the request	18
Support for client authentication by different methods	18
Control over the response	18
Improved Zim session control	18
XML and XSLT support	18
Page template support	19
XSL-FO support	19
Parameter template can be specified for the procedure	19
Improved Zim session control	19
Ability to start ZimWeb automatically when Java servlet engine starts	19
Administration tools	20
USING ZIMWEB	21
Installation	21
The ZimWeb package	21
Example: Installing the ZimWeb on Tomcat V4.1.x	22
Reference	25
Configuring ZimWeb	26
Configuration file extensions	29
Commands issued by the Zim Server to the ZimWeb	30
Input Parameters	34
Parameter sources	34
Parameters with new or changed meanings	34

Page Templates	38
Template file format	38
Sample template customer.htm	38
Parameter XML format	38
Sample XML data	38
XML URIs	40
URIs in XML generated by the Zim database agent	40
URIs in XSLT stylesheets	40
URIs in XSL-FO	40
ZimWeb Administration	41
Web administration	41
Command line or scripted administration	41
CallURL utility	42
Creating Secure ZimWeb Applications	43
BIBLIOGRAPHY	44
XML Technologies	44
Java	44
END USER LICENCE AGREEMENT	45
SAMPLE SERVLET CONFIGURATION FILE WEB.XML	52

README

This release of the *ZimWeb Help Files* (February 2003) is created in support of the ZimWeb release. This document is deemed to be a Standard release.

To read more please see the Release Notes.

Welcome

Contacting Us

To receive updated information on ZimWeb, please visit our website at www.zim.biz. We welcome your feedback. Please send any general or specific comments to:

Mail	ZIM Technologies International Inc. 20 Colonnade Road, Suite 200 Ottawa, Ontario K2E 7M6
Phone	(613)727-1397
Fax	(613)727-9868
E-mail - Information	info@zim.biz
E-mail - Technical	zim@zim.biz



Trademarks and Copyright Information

Zim and ZimWeb are trademarks or registered trademark of ZIM Technologies International Inc.

Windows, Internet Explorer, Visual Basic, VB Script, Microsoft Access, Visual C++, ODBC, and SQL Server are registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Red Hat is a registered trademark of Red Hat Inc.

Linux is a registered trademark of Linus Torvalds.

NOVELL is a registered trademark of Novell Inc.

UNIX is a registered trademark of UNIX Systems Laboratories, Inc.

DB2/6000, DB2/2, DB2 SQL/DS, OS/400, and AIX are registered trademarks of International Business Machines Corporation.

ORACLE is a trademark of Oracle Corporation.

HP-UX is a trademark of Hewlett-Packard Company.

Java is a trademark of Sun Microsystems, Inc.

QNX is a trademark of QNX Software Systems Ltd.

The Graphics Interchange Format© is a copyright property of CompuServe Incorporated. GIF (sm) is a Service Mark property of CompuServe Incorporated.

All other product names referenced are believed to be either trademarks or registered trademarks of their respective owners.

This manual contains information that is proprietary to ZIM Technologies International Inc. It may not be reproduced by any means without the express written permission of ZIM Technologies International Inc., nor may its contents be divulged to others except as necessary for the purpose of using the product or products described herein.

Copyright © 1989 - 2003, ZIM Technologies International Inc.
All rights reserved.

Acknowledgements

This product includes software developed by the Apache Software Foundation, www.apache.org.

Notice

Every attempt has been made to ensure that the contents of this manual accurately describe Zim. However, because the software is continually being enhanced, certain differences can exist between the documentation and the software. These differences are covered by release notes and by online documentation delivered with your software. If you find any omissions or errors in this document, please contact us. Your comments will be incorporated into the next release of the manual.

Phone

(613) 727-1397

Fax

(613) 727-9868

E-mail

zim@zim.biz

Zim Developers Library

Published by:

ZIM Technologies International Inc.

Date:

January 2003

About ZimWeb

Zim is a powerful and flexible environment for developing and using all types of database applications.

Zim's entity-relationship (E-R) data model and fully integrated Object Dictionary enable progressive program development, whether the information processing system is simple or complex. Zim Integrated Server, also referred to in this document as Zim Server, provides a database server capability that can be used with Zim or other client processes. The following topics describe ZimWeb — its operation and how it can be used to create Web applications using Zim.

This user guide describes the features and functions of ZimWeb. The What is ZimWeb section provides some of the basic information that you will need to understand how to develop and set up ZimWeb applications. This is followed by a complete description of the implementation of the example application. Most Web applications using ZimWeb can be modeled after this example.

Since the example does not use all features of ZimWeb, a number of additional topics provide reference information on software installation, configuration, etc.

ZimWeb makes use of Zim Integrated Server (ZIS) to access Zim databases. Because of this you should have a good working knowledge of the Zim Integrated Server (ZIS) product in order to develop web-based Zim applications. Please refer to Zim Integrated Server for additional information.

Release Notes

ZimWeb Release Notes

- The ZimWeb Reference Platform
 - Installing the ZimWeb Reference Platform
 - Checking that ZimWeb is running correctly
 - Command Line Tools
 - About the ZimWeb Reference Platform
- Installing ZimWeb in other Servlet Container environments
 - Conflicts with different XML Parser and XSLT Processors

The ZimWeb Reference Platform

The ZimWeb installation includes not only ZimWeb, but also a Zim Integrated Server 7.1, an Example Application, a Sun Java SDK and the Apache Tomcat Servlet Container. Together these are termed the "ZimWeb Reference Platform".

The "ZimWeb Reference Platform" is available for two environments: Windows and redhat Linux.

The ZimWeb Reference Platform is straightforward to install on a Windows PC or on a redhat 7.x linux machine (of course it depends on the environment that you receive ZimWeb), and enables a Zim developer to start working with ZimWeb very quickly. All components of the software apart from ZimWeb itself and Zim Integrated Server 7.1 are available for download under open licenses of various kinds which mean that they can be redistributed.

ZimWeb is a Java Servlet, which means that it can be executed on a variety of operating systems and Servlet Containers. ZimWeb has been tested successfully under various Java versions on various versions of Apache Tomcat on various platforms.

However, if you experience problems executing ZimWeb in another environment, then it is recommended that you attempt to reproduce the problem on one of the ZimWeb Reference Platform.

Installing the ZimWeb Reference Platform

Note: You will need activations keys to install this product. Please contact us at info@zim.biz and ask for the keys. There is a file called agreement.doc that needs to be signed.

Windows Environment:

Execute the program setup.exe and follow the instructions.

redhat Linux Environment:

Execute the shell `zimweb_install.sh` and follow the instructions.

Note: Both procedures will work even if, and will not interfere with, any Java environment or Tomcat installation that is already installed.

Checking that ZimWeb is running correctly

To check that ZimWeb is running correctly, go through the following procedure:

For Windows as a ZimWeb Reference Platform:

1. Start the Zim Integrated Server 7.1 by selecting **Start | Programs | ZimWeb | Start Zim Integrated Server**.
2. Start Tomcat by selecting **Start | Programs | ZimWeb | Start Tomcat**. You should see the database agents start on the Zim Integrated Server console.
3. Browse to the Tomcat Homepage by selecting **Start | Programs | ZimWeb | Tomcat Homepage**. If you see the Tomcat Homepage then Tomcat is functioning correctly.
4. Browse to ZimWeb Administration by selecting **Start | Programs | ZimWeb | ZimWeb Administration**. When requested for the username and password, enter "zimwebadmin" and "zimweb" (the default values) respectively. The ZimWeb Administration Tool should appear, and should be indicating that the Database Connections are running.
5. Stop and Start the database connections by pressing the appropriate button on the ZimWeb Administration page. Make sure that the connections are running for the next test.
6. Browse to ZimWeb Example Application by selecting **Start | Programs | ZimWeb | ZimWeb Example Application**. The ZimWeb Example Application page should appear. Check that each example runs correctly.
7. Shutdown Tomcat by selecting **Start | Programs | ZimWeb | Shutdown Tomcat**. You should see the database agents terminated on the Zim Integrated Server console.
8. Shutdown the Zim Integrated Server by selecting **Start | Programs | ZimWeb | Zim Integrated Server Administrator**. You will see a window asking for Server Host Name and Server Port Number. Inform localhost and 5002 and press Connect. Select Server and Stop Server.

For redhat Linux 7.x as a ZimWeb Reference Platform:

1. Go to the **bin** directory located on ZimWeb installed directory (default is `/opt/zimweb/bin`)
2. Start the Zim Integrated Server 7.1 by executing **`zimserv_startup.sh`**.
3. Start Tomcat by executing **`tomcat_startup.sh`** You should see the database agents start on the `nohup.out` file.
4. Browse to the Tomcat Homepage opening the Linux Web Browser and accessing <http://localhost:8080>. If you see the Tomcat Homepage then Tomcat is functioning correctly.
5. Browse to ZimWeb Administration opening the Linux Web Browser and accessing <http://localhost:8080/ZII/servlet/ZIIAdmin> . When requested for the username and password, enter "zimwebadmin" and "zimweb" (the default

- values) respectively. The ZimWeb Administration Tool should appear, and should be indicating that the Database Connections are running.
6. Stop and Start the database connections by pressing the appropriate button on the ZimWeb Administration page. Make sure that the connections are running for the next test.
 7. Browse to ZimWeb Example Application opening the Linux Web Browser and accessing <http://localhost:8080/ZII/example/index.htm> .The ZimWeb Example Application page should appear. Check that each example runs correctly.
 8. Shutdown Tomcat by executing **tomcat_shutdown.sh**. You should see the database agents terminated on the nohup.out file.
 9. Shutdown the Zim Integrated Server by executing **zimserv_shutdown.sh**. You will see a window asking for one option. Select option 3 and after 4.

Command Line Tools

The ZimWeb Reference Platform includes the following scripts in the directory [ZIMWEB]\bin which can be used from the command line or in automated processes:

- tomcat_startup.bat - Start Tomcat.
- tomcat_shutdown.bat - Shutdown Tomcat.
- zimwebadmin.bat - Send a command to the ZimWeb Administration Tool. The syntax is zimwebadmin.bat <command> [<username> <password>]. For example, zimwebadmin.bat conn-status zimwebadmin zimweb would check the status of the database connections if the username and password are set to their default values.
- xmlproc.bat - A command line interface for XML processing. You can test XSLT, Page Templates and rendering with XSL-FO from the command line. Enter xmlproc.bat without any parameters for the syntax.

About the ZimWeb Reference Platform

The ZimWeb Reference Platform consists of:

- File Release_Notes.htm - the ZimWeb Release Notes (this file).
- Directory j2sdk - contains the Java environment: Java2 SDK Version 1.4.1_01 for Windows or redhat Linux (it depends on the environment that you receive ZimWeb).
- Directory tomcat - contains the Servlet Container: Apache Tomcat V4.1.18 LE. In particular, note that ZimWeb itself is installed at tomcat/wepapps/ZII, and that by default the Tomcat logs for ZimWeb are in the files tomcat/logs/localhost_ZimWeb_log..
- Directory conf - contains the ZimWeb configuration file zimweb.cfg.
- Directory logs - contains the ZimWeb log file zimweb.log.<date>.txt
- Directory exampledb - A Zim database for the ZimWeb Example Application
- Directory zis7.1 - The Zim Integrated Server 7.1.
- Directory WAR - contains the ZimWeb Web Application Archive file ZII.WAR, for installing ZimWeb in other Servlet Container Environments.

The Java environment is unmodified from the original Sun distribution.

The ZimWeb Tomcat installation was produced using this procedure:

1. Start with Tomcat V4.1.18 LE.
2. Added the role "zimwebadmin", and the user "zimwebadmin" with password "zimweb" within that role, to the Tomcat configuration file [TOMCAT_ROOT]/conf/tomcat-users.xml. The original file is saved as [TOMCAT_ROOT]/conf/tomcat-users-ORIGINAL.xml.
3. Adjust the Tomcat configuration file [TOMCAT_ROOT]/conf/server.xml so that the ZimWeb application is logged separately from other applications into [TOMCAT_ROOT]/logs/localhost_ZimWeb_log.<date>.txt. The original file is saved as [TOMCAT_ROOT]/conf/server-ORIGINAL.xml.
4. Added the ZimWeb application directory at [TOMCAT_ROOT]/webapps/ZII. This includes the web portion (index page, stylesheets and page templates) of the ZimWeb Example Application.

Installing ZimWeb in other Servlet Container environments

Installing web applications into Servlet Containers is typically done using a Web Application Archive or WAR file. This is a JAR (Java Archive) image of the application root, file extension ".war" and with a name matching the application root directory name.

The file [ZIMWEB]/WAR/ZII.war is a Web Application Archive consisting of ZimWeb plus the web portion of the ZimWeb Example Application.

When installing it, you will need to adjust the ZimWeb configuration parameters config-file and log-file (in Tomcat these are found in the file [TOMCAT_ROOT]/webapps/ZII/web.xml) as appropriate.

Conflicts with different XML Parser and XSLT Processors

ZimWeb requires JAXP 1.1 XML Parsing and XSLT Processing facilities. Tomcat itself, and other typical Servlet Containers, require XML Parsing facilities. ZimWeb includes recent versions of the Apache Xerces and Xalan APIs to provide these facilities in the directory [TOMCAT_ROOT]/webapps/ZII/WEB-INF/lib/jaxp.

However, if different versions of XML Parsing and XSLT Processing facilities are available at the same time then there can be conflicts between them, leading to the Servlet Container and/or ZimWeb malfunctioning.

There are several choices to make the required facilities available and function correctly, including:

- In the default installation environment, Java 1.4.x already includes these facilities. Hence the Tomcat 4.1.x **LE** version has been selected, which does not include these, and by default these facilities are not supplied in ZimWeb either, so there is no potential for conflict.
- If installing into an environment using a Java version 1.2.x or 1.3.x then the Java environment does not include these facilities, so XML Parsing will probably be included in the Servlet Container environment e.g. in Tomcat

non-LE versions you would find these in the directory [TOMCAT_ROOT]/common/endorsed. So for Tomcat **non-LE** versions, we recommend that you replace the files in that directory (xerces.jar and xml-apis.jar) with the files in the directory [TOMCAT_ROOT]/webapps/ZII/WEB-INF/lib/jaxp.

What is ZimWeb

An Introduction

ZimWeb connects a Zim Server to the Internet, developed from the current ZimCGI. It is designed to be, for the most part, upward compatible with ZimCGI, but it has some distinct advantages, including:

- It is an all-Java servlet - you can run it on any platform with a Java Servlet Engine such as Apache's Jakarta Tomcat.
- It makes Zim web development easier with support for features like client sessions, cookies, etc.
- It supports XML and multiple client types (web browsers, cellphones and PDAs, etc.) by integrating with the Apache's Xalan-Java version 2 XSLT styling engine and Apache's Formatting Object Processor .

To help you understand why these technologies are so useful for developing Internet and other applications we recommend that you read the bibliography.

ZimWeb Architecture

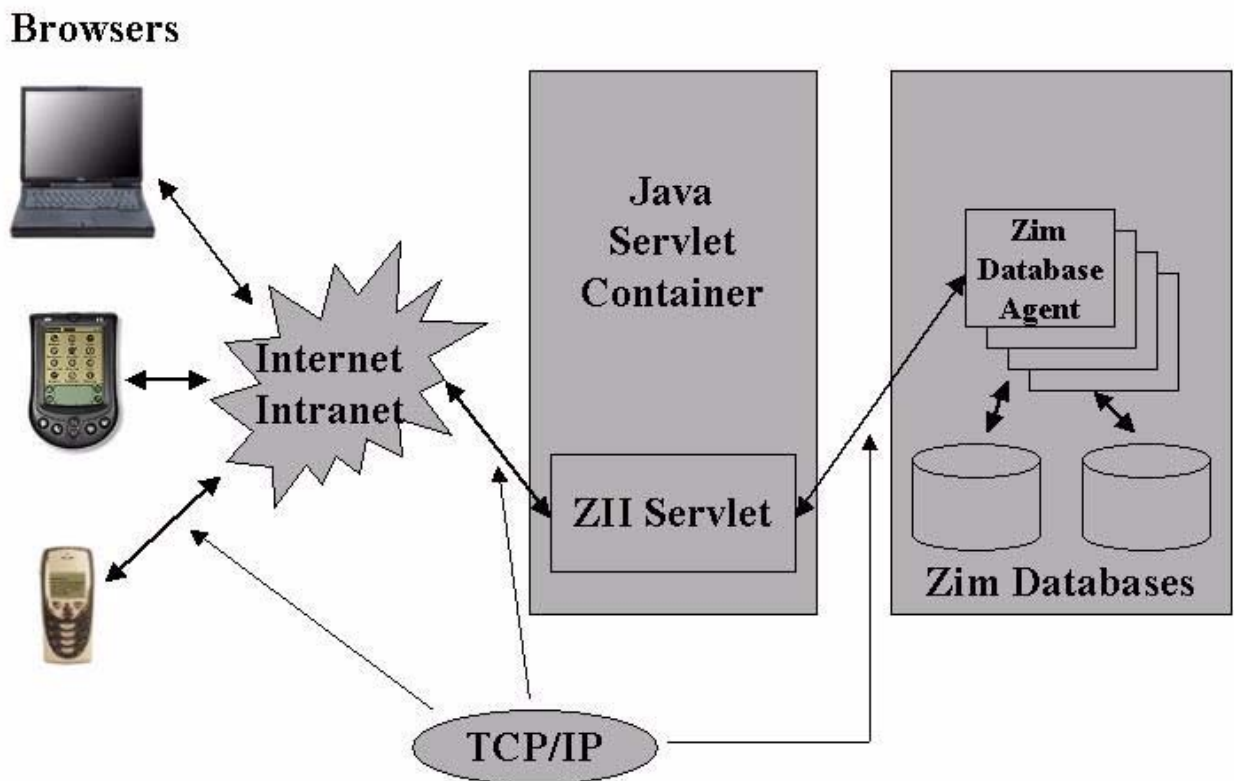
ZimWeb Architecture

In the case of Web applications implemented in Zim, a Java Servlet program called ZIIServlet, provided as part of ZimWeb, works together with Zim Integrated Server (ZIS) to process the request from the client.

In particular, request parameters (e.g. HTML form field values) are passed as parameters to a Zim program that is executed on the Zim Integrated Server. That program is responsible for processing the request and constructing the response. Although the response can be sent directly back to the client, ZimWeb includes a variety of options for processing it.

Java Servlets must be executed on a Java Servlet Container such as Apache's Jakarta Tomcat. ZimWeb communicates with Zim Integrated Server using TCP/IP. The Zim Integrated Server can be running on the same system as ZimWeb, or another system, provided that there is a network connection between the two systems.

The following diagram shows the basic components of a Web application using ZimWeb.



ZimWeb Components

Browser

Typical browsers include Netscape Navigator and Microsoft Internet Explorer. Browsers can be found on a variety of platforms including PCs, PDAs (Personal Digital Assistants) and cell phones. The browser is not supplied with ZimWeb

Java Servlet Container

Also known as a Java Servlet Engine. An environment in which Java Servlets can be executed. Tests were done with Apache's Jakarta Tomcat, which is the official reference platform for Java Servlet Containers. A Java Servlet Container can be run independently, or it can be integrated with a Web Server. The ZimWeb Reference Platform includes Tomcat, but a variety of alternatives are available.

ZIIServlet

ZIIServlet is a Java Servlet. ZIIServlet can be loaded automatically by the Java Servlet Container when the first request is received, or alternatively it can be loaded in advance so that it is ready beforehand.

Zim Database Agent

The Zim database agent is part of Zim Integrated Server. It is a process that runs on a specific machine and accesses a specific database. ZimWeb can communicate with any number of databases on any number of database machines. However, a Database Agent is always connected to a specific database.

Features

N.B. To ensure complete understanding of this topic we recommend you familiarize yourself with ZimCGI.

Compatibility with the ZimCGI

ZimWeb is, for the most part, upwards compatible with the ZimCGI, meaning that existing ZimCGI applications should not require much change to execute in the same manner with the ZimWeb.

The key difference is that **ZimWeb requires you to have a TEMPLATE parameter**- it is not optional (however see Parameter template can be specified for this procedure).

Ability to handle POST and GET requests

ZimWeb can accept parameters via either a POST request (as ZimCGI) or a GET request (with the parameters in the URL, which means you can bookmark a ZimWeb request or add a hyperlink).

Debug information

If you supply the parameter DEBUG in the request, you can see all sorts of useful details about the parameters, the http headers, etc. which can be very useful for debugging.

Since exposing the debugging information to any client may pose a security hazard, the ability to debug a request can be enabled and disabled by the application administrator.

HTTP session and cookie parameter management

N.B. When we refer to a client session, we are referring to the HTTP session, not the Zim agent session.

ZimWeb allows you to control and access client session and cookie parameters.

For example, your ZimWeb application could set the client session parameter PASSWORD to indicate a password that the user had supplied when they logged in. From that point, the parameter PASSWORD, if referenced in the TEMPLATE, would be supplied with that value, until the user exited their browser, the session timed out (which the application can control), or your application cleared the parameter or invalidated the session (i.e. they logged out from your Zim application). Client session parameters do not cause the Zim agent session to remain open, so your Zim agents remain available between processing client requests.

Also, client session parameters are managed by the ZimWeb, and are not sent to the browser, which can be good from a security standpoint.

Similarly, your application could set a cookie parameter USERNAME to the user's name if, say, they check a "Remember me" box. Your application will then receive the USERNAME parameter, which can be a reference in the TEMPLATE, until the cookie expires (which the application can control) or the parameter is cleared. In addition, ZimWeb allows you to specify that particular parameters which only comes from the HTTP session or cookies, by prefixing their names with the session or

cookie respectively.

Access to key information about the request

ZimWeb exposes key information about the request as parameters which can be included in your TEMPLATE, these include:

- The PathInfo of the request is placed in the PATHINFO parameter. For example, say the ZimWeb servlet (ZII) is at the URL `http:www.mycorp.com/ZII/servlet/ZII` - if it receives a request for a URL relative to that e.g. `http:www.mycorp.com/ZII/servlet/ZII/one/two`, then the PATHINFO parameter would be set to `/one/two`.
- The URL of the request entered by the client (including any Pathinfo, but excluding any parameters) is placed in the REQUESTURL parameter.
- If the client HTTP session is being tracked through a cookie (which it must be for HTTP sessions to function correctly), the parameter SESSIONFROMCOOKIE is set to 1, otherwise 0.

Support for client authentication by different methods

ZimWeb provides support for client authentication in various different ways:

- If the client has been authenticated with the web server or servlet engine, their user name is placed in the AUTHUSERNAME parameter, and the authentication method (usually BASIC) is placed in the AUTHTYPE parameter.
- The Zim application can force the client to authenticate itself using the request-authentication:<realm> command, and discover the (unauthenticated) user name and password of the client through the CLIENTUSERNAME and CLIENTPASSWORD parameters.
- The Zim application can record the client's authentication status in HTTP session parameters.

Control over the response

ZimWeb gives the Zim application control over the response sent back to the client, including the ability to:

- Redirect the client to a particular URL.
- Set how long the response can be cached by the client.
- Set the HTTP status or error code and message.
- Set arbitrary HTTP headers.

Improved Zim session control

ZimWeb includes options to improve the security of, and to simplify the use of, persistent Zim sessions:

- Like the ZimCGI, it has improved security by preventing clients from starting sessions when the Zim application is not expecting this - this could be used to deny service by committing all the Zim database agents.
- Zim sessions can be placed under the exclusive control of the Zim application through the start-zim-session, end-zim-session and cancel-zim-session commands, coupled with disabling the input parameters previously used to control Zim sessions.
- New or rewritten applications can manage Zim sessions through the HTTP session, eliminating the need to pass the SESSIONID parameter back and forth.

XML and XSLT support

If your Zim application generates XML output, then it can request that an XSLT style sheet is applied to that XML to generate what is sent to the client (web browser,

cellphone, etc.).

Also, the web request can indicate that a particular style sheet is required, overriding the style sheet specified by the Zim application. It can even request that no style sheet is used, returning instead the raw XML.

XSLT stylesheets are automatically compiled and cached in memory for maximum performance. The web-based administration tool allows you to see which XSLT stylesheets are cached, and also enables you to clear the cache, forcing the stylesheets to be reloaded, which is useful if you have changed the stylesheets.

There is no upper limit on the XSLT stylesheet cache size - if you run out of memory through caching stylesheets then you haven't got enough memory to run your application efficiently.

Page template support

Some pages, particularly those designed by graphic artists etc., may not be convenient to turn into XSLT stylesheets (see XML and XSLT support).

ZimWeb includes page template support. A page template is just a regular page (HTML etc.), with parameter placeholders to indicate where each parameter to be supplied by the Zim application is to be placed. This template can be designed easily with your normal web page design tool.

The Zim application can then supply the parameters in an XML format. These parameters are combined with the template, with the result page being delivered to the client.

Just as with the XSLT processing, the templates are compiled into memory for maximum performance, and the request can specify an alternative, or even no page template.

XSL-FO support

ZimWeb includes XSL-FO support - this allows the Zim application to render XSL-FO (Formatting Objects) - usually the result of XSLT processing, though it can be from a page template or even the unprocessed Zim agent output - into Adobe Portable Document Format (PDF) or Rich Text Format (RTF), so the application can deliver typeset output directly to the client's browser.

Parameter template can be specified for the procedure

When specifying which procedures can be run for a given connection, you can also specify the parameter template, eliminating the need to supply a TEMPLATE parameter with the request.

Improved Zim session control

The Zim application can directly control the termination of a Zim session - i.e. whether a Zim database agent is dedicated to a particular client, or is released when a request has been processed. This is very useful in preventing "denial-of-service" to the Zim web application by holding open every available Zim database agent.

Of course, the ZII's HTTP session and cookie parameter management eliminates much of the need to hold open the Zim session.

Ability to start ZimWeb automatically when Java servlet engine starts

By setting "load-on-startup" for the ZimWeb servlet configuration (the file web.xml if

you are using Tomcat), ZimWeb will automatically start, and create all the Zim agent sessions, when the Java servlet engine starts.

Note: If the Zim Server is not available, the ZimWeb will start anyway, and you can start the Zim agent sessions when the Zim Server does become available with the Web-based administration tool.

Administration tools

ZimWeb has a web-based administration tool, which enables you to:

- Monitor, start and stop the Zim agent sessions e.g. if you want to prevent access to the Zim application for maintenance purposes.
- Enable or disable client request debugging.
- See the value of the startup parameters.
- Monitor and clear the XSLT style sheet and page template caches (e.g. if you update your style sheets or page templates, they will not be read if the style sheet or page template is already cached in memory.)

In addition, ZimWeb includes a utility which enables you to perform these functions from the command line or automatically in scripts, scheduled tasks, etc.

Using ZimWeb

Installation

Installation

"Please refer to the ZimWeb Release Notes for instructions on installing ZimWeb."

Here are some additional details about installing ZimWeb.

The ZimWeb package

ZimWeb is packaged as a WAR (Web Application Archive) file containing the following files:

Package	Contents
example/index.htm	Home page for the ZimWeb Example Application.
WEB-INF/example/template/*.htm	Page templates for the ZimWeb Example Application.
WEB-INF/example/xslt/*.xsl	XSLT style sheets for the ZimWeb Example Application.
WEB-INF/web.xml	Configuration for the ZII servlets.
WEB-INF/lib/avalon-framework-xxx.jar	Apache Avalon Framework (required by Apache FOP).
WEB-INF/lib/batik-xxx.jar	Apache Batik SVG processor (required by Apache FOP).
WEB-INF/lib/fop-xxx.jar	Apache FOP XSL-FO processor.
WEB-INF/lib/jimi-xxx.jar	Sun JIMI (required by Apache FOP).
WEB-INF/lib/jfor-xxx.jar	JFOR XSL-FO processor for RTF.
WEB-INF/lib/logkit-xxx.jar	Apache Logkit (required by Apache FOP).
WEB-INF/lib/zii.jar	ZimWeb itself.
WEB-INF/lib/jaxp/xalan-xxx.jar	Apache Xalan XSLT processor (in case this is not part of your Java or servlet engine environment).
WEB-INF/lib/jaxp/xercesImpl-xxx.jar	Apache Xerces XML parser (in case this is not part of your Java or servlet engine environment)..
WEB-INF/lib/jaxp/xml-apis-xxx.jar	XML APIs for XML parsing and XSLT processing from Apache Xalan (in case this is not part of your Java or servlet engine environment)..

The ZimWeb file zii.jar contains classes in the following Java packages:

File	Description
zim.CGIServer	The original ZimCGI, with a few

	modifications
zim.jdbc	Zim JDBC Driver
biz.zim.zii	The main ZimWeb classes
biz.zim.util	Utility classes for ZimWeb

There are two servlets in ZimWeb which you can invoke:

Servlet	Description
biz.zim.zii.ZIIServlet	<p>The ZimWeb itself. This is the servlet that you invoke to service a request to a Zim web application instead of the zimcgic.exe (or whatever) of the ZimCGI. You will probably want to alias to something shorter. The ZimWeb sample configuration file web.xml for Tomcat shortens it to ZII, so if, say, you installed it with Tomcat's default parameters (servicing port number 8080), you would invoke it with a URL like <code>http://<hostname>:8080/ZII/servlet/ZII.</code></p>
biz.zim.zii.ZIIAdminServlet	<p>The ZimWeb Administration servlet allows you to administer ZimWeb, including checking its configuration parameters, starting and stopping Zim Server agent connections, and checking and clearing the XSLT style sheet cache. Read more about ZimWeb administration. N.B. In a production environment, you want to secure access to this servlet - you can shut down your Zim web application with a single click! To find out more information on this topic please read below or check out your Java Servlet Engine.</p>

The ZimWeb jar file also contains the ZimCGI Server - equivalent to the zimcgis.jar file. You may wish to use the zii.jar for both purposes to avoid problems with multiple versions on your Java CLASSPATH.

Example: Installing the ZimWeb on Tomcat V4.1.x

Before installing ZimWeb, make sure you have a working Tomcat installation, consisting of, say, Java 1.4.x plus Tomcat V4.1.x LE. N.B. You choose a Tomcat LE version with Java 1.4.x because Java 1.4 includes XML processing facilities which would otherwise be duplicated in Tomcat.

One way of installing the ZII on Tomcat V4.1.x is as follows:

1. Place the ZimWeb WAR package file ZII.war in the directory [TOMCAT_ROOT]/webapps.
2. Remove any directory [TOMCAT_ROOT]/webapps/ZII if it already exists e.g. if you have installed a previous version of ZimWeb.

3. Start Tomcat. This automatically unpacks the contents of the file ZII.war into the directory [TOMCAT_ROOT]/webapps/ZII. It most likely won't connect successfully to any Zim database because the ZimWeb configuration file requires adjustment.
4. Stop Tomcat.
5. Adjust the ZimWeb configuration file [TOMCAT_ROOT]/webapps/ZII/web.xml as appropriate. In particular, you should ensure that the config-file and log-file parameters are set correctly. Note also that it secures the ZimWeb Administration servlet so that it can only be accessed by users in the role zimwebadmin - see the next step for how to add such a user.
6. Adjust the Tomcat configuration file [TOMCAT_ROOT]/conf/tomcat-users.xml to add a user in the role ziiadmin that can administer ZimWeb e.g. to add a user zimwebadmin with password zimweb you could do this by adding lines such as these into it:

```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>
  ...
  <role rolename="zimwebadmin"/>
  ...
  <user username="zimwebadmin" password="zimweb"
roles="zimwebadmin"/>
  ...
</tomcat-users>
```

7. Adjust the Tomcat configuration file [TOMCAT_ROOT]/conf/server.xml so that the ZimWeb application is logged separately from other applications into [TOMCAT_ROOT]/logs/localhost_ZII_log.<date>.txt. You can do this by adding the following context:

```
...
<Host ...>
  ...
  <!-- ZII Context -->
  <Context path="/ZII" docBase="ZII" debug="0"
    reloadable="true" crossContext="true">
    <Logger className="org.apache.catalina.logger.FileLogger"
      prefix="localhost_ZimWeb_log." suffix=".txt"
      timestamp="true"/>
  </Context>
  ...
</Host>
...
```

8. Important! You must ensure that ZimWeb's environment includes XML parsing and XSLT facilities, and that there are no conflicts amongst the XML processing facilities which may be supplied with your Java and Tomcat environments and ZimWeb. For example:-
 - If you are running a Tomcat LE version using Java 1.4 or higher, then the Java environment already includes XML parsing and XSLT facilities, and no action is required.
 - If you are running Tomcat non-LE version using a Java version prior to 1.4, then you will have to copy the files supplied with ZimWeb in the WEB-INF/lib/jaxp directory (xalan-xxx.jar, xercesImpl-xxx.jar and xml-apis-

xxx.jar) into the [TOMCAT_ROOT]/common/endorsed directory, replacing any versions of those files which came with your version of Tomcat.

9. Restart Tomcat. It will automatically start the Zim database connections if it is configured to do so.

The above should get you started. You will probably want to configure Tomcat further - particularly by adjusting the file [TOMCAT_ROOT]/conf/server.xml - before installing into a production environment.

N.B. Make sure that you do not have the zimcgis.jar file from the ZimCGI in your Java CLASSPATH, otherwise they may interfere with one another.

Tomcat can be configured to run both standalone, or in conjunction with other web servers like Apache.

If after reading the above you wish to install ZimWeb in another manner, or on another Java Servlet engine, and you run into difficulties we recommend you read and understand the documentation for your Java Servlet Engine.

Reference

This is a description of each of the parameters and commands that are used to control the ZimWeb. It is organised by the area where the given feature can be configured or used:

- Release Notes
- Configuring ZimWeb
- Commands issued by the Zim Server to ZimWeb
- ZimWeb Input Parameters
- ZimWeb Page Templates
- XML URIs
- ZimWeb Administration
- Creating Secure ZimWeb Applications.

Configuring ZimWeb

If you are using Tomcat, and have installed the ZimWeb in the manner described in the installation instructions, then the servlet configuration (servlet initialisation parameters) will be in the web.xml file in the [TOMCAT_ROOT]/webapps/ZII/WEB-INF directory. If you are using a different servlet engine, then you will have to consult its documentation to see how to configure it.

Here are the configuration parameters, organised by function type:

Automatic servlet loading	
load-on-startup <i>(Tomcat configuration parameter)</i>	This controls whether the ZII servlet is loaded automatically when the Tomcat Java servlet engine starts. This means that the servlet is loaded and ready to run when the first client request arrives. Read the Tomcat documentation for more details on this subject. <i>N.B. This is not an initialisation parameter for the ZII servlet - it is a configuration parameter for the servlet. If you are using a different servlet engine then consult its documentation.</i>
Automatically starting database connections	
auto-start	This controls whether the ZII will automatically start the Zim Server agent sessions when the ZII servlet is loaded. If this equals yes (the default), then they are; if it is no, then they are not. If the ZII receives a client request, and the agents are not started, then the client will receive an error indicating that the database connection have not been started.
Configuration and log files	
config-file	This gives the file name of the Zim Server agent configuration file - the same zimcgis.cfg file that the ZimCGI uses. Consult the ZimCGI documentation and Configuration file extensions for more details on this subject.
log-file	This gives the file name of the log file that the ZII write - equivalent to the zimcgisrv.log file that the ZimCGI generates. Consult the ZimCGI documentation for more for more details on this subject.
verbose	This controls whether certain aspects of the logging features of the ZII are more verbose than usual (and necessary!).

	<p>If it is yes, then the logging is more verbose. If it is no (the default), then your logs are not so verbose.</p> <p>Consult the ZimCGI documentation for more for more details on this subject.</p>
Controlling XSLT processing	
use-xslt-output-type	<p>This controls how the ZII sets the mime content type of the response, if the output is passed through the XSLT processor and there is no content-type: command.</p> <p>If it is no (the default), then the content type is assumed to be text/html.</p> <p>If it is yes, then the XSLT processor will extract the content type from the <xsl:output ... /> clause in the stylesheet.</p> <p>If you set this to yes and you do not have an <xsl:output ... /> clause in the XSLT stylesheets, then the XSLT processor will usually assume the content type is XML (text/xml), which may not be what you want.</p> <p>I don't think there is a good reason for you <i>not</i> to put an <xsl:output ... /> clause in all of your XSLT stylesheets.</p> <p>N.B. for more details on this subject please see documentation on XSLT.</p>
Session and cookie management	
default-session-timeout	<p>This is the default session timeout in seconds.</p> <p>If not specifed, ZII will default to a 1 hour timeout.</p> <p>This can be overridden for a particular session by using the session-timeout: command.</p>
default-cookie-timeout	<p>This is the default cookie expiry time in seconds.</p> <p>If not specifed, ZII will default to a 1 year expiry.</p> <p>This can be overridden for a particular cookie by using the cookie-timeout: command.</p>
Persistent Zim sessions	
zim-session-security	<p>This controls the security of persistent Zim sessions; the various options are:-</p> <ul style="list-style-type: none"> • open - insecure, as with the previous ZimCGI - the client can initiate a Zim session through the SESSIONID parameter without restriction.

	<ul style="list-style-type: none"> • disable-start - ignore requests to <i>start</i> a persistent Zim session in the input parameters (i.e. a blank SESSIONID parameter). <i>You can start the persistent Zim session with the start-zim-session command instead.</i> • restrict-start - ignore requests to <i>start</i> a persistent Zim session in the input parameters (i.e. a blank SESSIONID parameter) if the parameter TEMPLATE does not include the SESSIONID parameter. <i>This is the default setting, which should work with existing ZimCGI applications. This should be combined with specifying all the parameter templates in the "Security" declarations in the configuration file, of course.</i> • restrict-continue - prevent <i>starting</i> or <i>continuing</i> a persistent Zim session in the input parameters (i.e. a SESSIONID parameter - blank or not - without also an ENDESSION or CANCELSESSION parameter) if the parameter TEMPLATE does not include the SESSIONID parameter. • http-session - the input parameters SESSIONID, ENDESSION and CANCELSESSION are ignored; instead, ZimWeb stores the SESSIONID automatically in the HTTP session; and it is the exclusive responsibility of the Zim application to start and stop persistent Zim sessions through the start-zim-session, end-zim-session and cancel-zim-session commands; also, it is <i>not</i> necessary to have the SESSIONID in the parameter TEMPLATE, nor output the SESSIONID into a hidden field. • disable - the input parameters SESSIONID, ENDESSION and CANCELSESSION are ignored.
Debugging	

allow-debug	<p>This is the default state of the client debug capability.</p> <p>If yes then the client debug capability will initially be enabled when the ZII is started; if no (the default value) then it will initially be disabled.</p>
-------------	--

Configuration file extensions

Specified parameter templates	
Security	<p>With the ZimCGI, you can specify the Zim procedures that may be executed in a list following the Security tag(s) for each connection.</p> <p>In addition, the ZII allows you to specify the parameter template in parentheses after the procedure name.</p> <p>e.g. Security MyProc(CC) specifies that MyProc is an allowed procedure, and that the parameter template is (CC), eliminating the need to specify a TEMPLATE parameter.</p> <p>If a parameter template is specified in the configuration file, this will override any TEMPLATE parameter in the request.</p>

Commands issued by the Zim Server to the ZimWeb

Just like the ZimCGI, output from the Zim Server agent is sent to ZimWeb and then to the client (web browser, cellphone, PDA, etc.). However, there are various commands that can be used to control how the Zim application output is processed, and how ZimWeb behaves.

ZimWeb expects that the Zim application will output a ZimWeb command, or series of commands (one per line), a blank line, and then the actual content.

If you just want to send HTML to a web browser, then the beginning of your application's output should be:

```
content-type:text/html  
  
<html>  
etc...
```

N.B. That blank line between the content-type: line and the start of the content is very important! line and Commands are not case sensitive. In case you are wondering about the content-type line, it is not send directly to the web browser.

Here are the commands, organized by function type:

Content processing	
content-type:<mime>	Sets the response content type to <mime> . You do not need to set this if you are applying an XSLT stylesheet with the apply-xslt: command and the ZimWeb is configured to use the output type specified in the stylesheet's output method. If this is specified, and a stylesheet is being applied, then the content type specified will override the stylesheet's output method. If the ZimWeb cannot determine the content type, it will assume it is text/html, except if the client requests no styling (parameter STYLE=none), then, if there is no content-type: command, the content will be assumed to be XML (mime type text/xml), since you can only style XML. <i>If that all sounds rather complicated, it's just trying to express explicitly some fairly reasonable and benign assumptions.</i>
apply-xslt:<stylesheet>	Apply XSLT stylesheet <stylesheet> to the Zim agent output. The <stylesheet> parameter is a file path relative to the base path of the ZII application. If you have installed the ZimWeb in the manner described in the installation instructions then specifying the <stylesheet> as /styles/cool.xsl will refer to the file [TOMCAT_ROOT]/zii/styles/cool.xsl. The <stylesheet> parameter can be overridden by the

	<p>input parameter STYLE.</p> <p>The ZimWeb automatically compiles and caches XSLT stylesheets in memory for maximum performance.</p> <p><i>N.B. The ZII does not accept a URI for the stylesheet as running a stylesheet not under your control could be a security hazard.</i></p>
<p>apply-page-template: <form-template></p>	<p>Apply the page template <page-template> to the Zim agent output.</p> <p>The <page-template> parameter is a file path relative to the base path of the ZimWeb application, just like the <stylesheet> parameter above.</p> <p>The <page-template> parameter can be overridden by the input parameter PAGETEMPLATE.</p> <p>The ZimWeb automatically compiles and caches page templates in memory for maximum performance.</p> <p>Read more about using page templates.</p> <hr/> <p><i>N.B. The ZimWeb does not accept a URI for the form template as this could be a security hazard.</i></p>
<p>apply-xslfo: <renderer></p>	<p>Treat the Zim agent output, or the result of the other content processing (typically XSLT, though it could be a page template), as XSL Formatting Objects and render it with the specified <renderer>.</p> <p>The two rendering options currently available are PDF (Adobe Portable Document Format), for which you would specify apply-xslfo:pdf, or alternatively RTF (Rich Text Format), for which you would specify apply-xslfo:rtf.</p> <p>Note that rendering to PDF is much more accurate than rendering to RTF.</p>
<p>HTTP session parameter management</p>	
<p>set-session-param: <name> = <value></p>	<p>Sets the session parameter <name> to the value <value>.</p> <p>If the parameter <name> already exists, then its value is changed to <value>.</p> <p>Once set, the parameter will be available for inclusion in the TEMPLATE parameter.</p> <p>Session parameters remain set until they are cleared, or the session times out or is invalidated.</p> <p>Session parameters are not sent to the client - they are managed by the ZII servlet.</p> <p><i>N.B. Cookies should be enabled on the client's browser in order that http session parameters function correctly.</i></p> <p><i>If the same parameter exists in several places then the form parameters have priority over the session parameters, and these have priority over the cookie parameters.</i></p>
<p>clear-session-param: <name></p>	<p>Clears the session parameter <name>.</p>
<p>invalidate-session</p>	<p>Terminates the client session and starts a new one. This will cause all session parameters to be discarded.</p>

session-timeout: <timeout>	<p>Sets the session timeout to <timeout> seconds.</p> <p>If not specified, the session timeout defaults to the value specified in the Configuring ZimWeb.</p> <p><i>N.B. It only changes the timeout for this session, not any other session, and it does not change the default session timeout.</i></p>
Cookie parameter management	
set-cookie-parm: <name>=<value>	<p>Sets the cookie parameter <name> to the value <value>.</p> <p>If the parameter <name> already exists, then its value is changed to <value>.</p> <p>Once set, the parameter will be available for inclusion in the TEMPLATE parameter.</p> <p>Cookie parameters remain set until they are cleared, or the cookie reaches its expiry time.</p> <p>To work, using cookie parameters requires that the client has cookies enabled.</p> <p><i>N.B. Cookies should be enabled on the client's browser in order that cookie parameters function correctly.</i></p> <p><i>If the same parameter exists in several places then the form parameters have priority over the session parameters, and these have priority over the cookie parameters.</i></p>
clear-cookie-parm: <name>	<p>Clears the cookie parameter <name>.</p> <p><i>N.B. This will not clear the JSESSIONID cookie which is required for session management.</i></p>
clear-all-cookies	<p>Clears all the cookies set by the ZimWeb.</p> <p><i>N.B. This will not clear the JSESSIONID cookie which is required for session management.</i></p>
cookie-timeout: <timeout>	<p>Sets the cookie timeout to <timeout> seconds, as opposed to the default cookie timeout, for cookies set after it in the ZimWeb commands.</p> <p>A negative timeout indicates that the cookie will expire when the user exits the client. However, I would suggest you use a session parameter instead in that case.</p> <p><i>N.B. It does not change the default cookie timeout.</i></p>
Persistent Zim session management	
start-zim-session	<p>Start a persistent Zim session. Key points to note about this are:-</p> <ul style="list-style-type: none"> • This is the Zim application requesting that the Zim session is to persist, not a request by the client. • If the persistent Zim session is being managed by a SESSIONID parameter in the form parameters then the Zim procedure must include the SESSIONID parameter in its TEMPLATE in order that it can continue using that Zim session. • Unlike the ZimCGI, <i>all</i> Zim database agent requests have a SESSIONID; but of course Zim

	sessions that are not to persist are ended at the end of the request in which they were started.
end-zim-session	Ends the current persistent Zim session.
cancel-zim-session	Ends the current persistent Zim session without sending any response to client.
Status and error codes	
set-status: <status-code>	Set an HTTP status code (without a message - sending a message with a status code is deprecated) to the client.
send-error: <status-code>[:<message>]	Send an HTTP error code, with or without a message, to the client.
Authentication	
request-authentication: <realm>	Send a BASIC authentication request for a given realm to the client.
Miscellaneous	
redirect: <URL>	Redirect the client to a new URL.
set-cache-time: <seconds>	Set the caching time for a response in seconds (normally the responses indicate that they cannot be cached).
set-header: <name>=<value>	Set a string HTTP header to a given value.
set-int-header: <name>=<value>	Set an integer HTTP header to a given value.
set-date-header: <name>=<value>	Set a date HTTP header to a given value. N.B. Dates are in milliseconds since the epoch (midnight on January 1, 1970).

Input Parameters

Input parameters are those parameters supplied as input to the Zim application. Parameters tend to work in the same way as the ZimCGI, except that:

- The TEMPLATE parameter is required, except where a default parameter template has been specified for a procedure.
- There are more parameters with reserved meanings - namely PATHINFO, STYLE, PAGETEMPLATE and RENDER.
- Parameters can come from HTTP sessions and cookies as well as form input.
- Form parameters can come from both POST and GET requests.

N.B. Parameter names are not case sensitive.

Parameter sources

Input parameters can come from three different sources:

- Form parameters (i.e. POST/GET).
- Session parameters.
- Cookie parameters.

You **must** specify the parameters to be passed to your Zim application in the TEMPLATE parameter, or the default parameter template for the procedure. Normally, if the same parameter exists in several places then the form parameters have priority over the session parameters, and these have priority over the cookie parameters.

However, if a parameter has a name beginning with the prefix session. e.g. session.IsLoggedIn then it can only come from the HTTP session parameters. Similarly, if a parameter has a name beginning with the prefix cookie. e.g. cookie.UserName then it can only come from the cookie parameters.

Parameters with new or changed meanings

Parameters with new or changed meanings, compared with the ZimCGI, are:

ZimCGI parameters which have been modified	
TEMPLATE	The meaning and format of this parameter has not been changed, but it is required (unlike the ZimCGI), unless the procedure has a default parameter template.
DEBUG	The meaning and format of this parameter has not been changed - if you set the parameter DEBUG to any value in the request parameters then the debugging information will be sent to the client. However, the debugging information is a lot more comprehensive, including the raw output of the Zim database agent. <hr/> Note: For security reasons, any

	<p>commands in the Zim database agent output are <i>not</i> processed in a DEBUG request.</p> <p>Note also that a request for debugging information from a client will be ignored unless <u>debug capability has been enabled.</u></p>
Request information parameters	
PATHINFO	<p>The PathInfo of the request is placed in the PATHINFO parameter, which you can place in your TEMPLATE.</p> <p>For example, say the ZimWeb servlet (ZII) is at the URL http:www.mycorp.com/ZII/servlet/ZII - if it receives a request for a URL relative to that e.g. http:www.mycorp.com/ZII/servlet/ZII/one/two, then the PATHINFO parameter would be set to /one/two.</p>
REQUESTURL	<p>The URL the client entered; N.B. this <i>includes</i> any PathInfo, but <i>excludes</i> any parameters.</p>
SESSIONFROMCOOKIE	<p>Set to 1 if the client HTTP session is being tracked through a cookie (e.g. the JSESSIONID cookie in Tomcat), otherwise 0.</p> <hr/> <p>N.B. This will be set on the second and subsequent calls to the servlet - the session is not set up the first time a client invokes the servlet. Of course, HTTP session tracking through cookies requires <u>that cookies be enabled on the browser.</u></p>
SESSIONFROMURL	<p>Set to 1 if the client http session is being tracked through URL rewriting, otherwise 0.</p> <hr/> <p>N.B. Session tracking does not currently work effectively through URL rewriting - instead you should track sessions by <u>enabling cookies on the browser.</u></p>
Authentication parameters	
AUTHUSERNAME	<p>The user authenticated by the web server.</p>
AUHTYPE	<p>The authentication method used by the web server - BASIC etc.</p>
CLIENTUSERNAME	<p>The (not necessarily authenticated) user name the client is claiming.</p>
CLIENTPASSWORD	<p>The (not necessarily authenticated) password the client is claiming.</p>
XSLT styling parameters	
STYLE	<p>This parameter indicates that a particular</p>

	<p>XSLT stylesheet is to be applied to the output of the Zim Server agent.</p> <p>If the Zim Server agent output includes an <code>apply-xslt:</code> command, then the stylesheet specified in the <code>STYLE</code> parameter overrides the that specified by the <code>apply-xslt:</code> command.</p> <p>The <code>STYLE</code> parameter specifies a file path relative to the base path of the ZimWeb application. If you have installed the ZimWeb in the manner described in the installation instructions then specifying <code>STYLE</code> as <code>/styles/cool.xsl</code> will refer to the file <code>[TOMCAT_ROOT]/ZII/styles/cool.xsl</code>.</p> <p>If you specify <code>STYLE=NONE</code> then the Zim agent output (presumably XML) will be returned as-is, without the application of any XSLT or form template processing.</p> <hr/> <p>Note: specifying a <code>STYLE</code> automatically suppresses any XSL-FO rendering specified by the Zim Server agent, unless the <code>RENDER</code> parameter is also supplied.</p>
Page template parameters	
<p>PAGETEMPLATE</p>	<p>This parameter indicates that a particular page template is to be applied to the output of the Zim Server agent.</p> <p>If the Zim Server agent output includes an <code>apply-page-template:</code> command, then the page template specified in the <code>PAGETEMPLATE</code> parameter overrides the that specified by the <code>apply-page-template:</code> command.</p> <p>The <code>PAGETEMPLATE</code> parameter specifies a file path relative to the base path of the ZimWeb application. If you have installed the ZimWeb in the manner described in the installation instructions then specifying <code>PAGETEMPLATE</code> as <code>/styles/fancy.html</code> will refer to the file <code>[TOMCAT_ROOT]/ZII/styles/fancy.html</code>.</p> <p>Read more about ZII page templates.</p> <p>If you specify <code>PAGETEMPLATE=NONE</code> then the Zim agent output (presumably XML) will be returned as-is, without the application of any XSLT or form template processing.</p> <hr/> <p>Note: specifying a <code>PAGETEMPLATE</code> automatically suppresses any XSL-FO rendering specified by the Zim Server agent, unless the <code>RENDER</code> parameter is also supplied.</p>

XSL Formatting Objects parameters

RENDER

This parameter indicates that the Zim agent output, or the result of the other content processing (typically XSLT), is to be treated as XSL Formatting Objects and rendered according to the RENDER parameter.

The two rendering options currently available are PDF (Adobe Portable Document Format), for which you would specify the value PDF, or alternatively RTF (Rich Text Format), for which you would specify the value RTF..

Note: that rendering to PDF is much more accurate than rendering to RTF.

If you specify RENDER=NONE then rendering is suppressed, and the XSL-FO is returned directly to the browser.

Page Templates

ZimWeb includes page template support. This allows you to combine parameters from Zim with the page (web page, etc.) delivered to the client.

Template file format

A page template is just a regular page (HTML etc.), with parameter placeholders to indicate where each parameter to be supplied by the Zim application is to be placed. This template can be designed easily with your normal web page design tool.

The parameters placeholders are formatted like HTML comments (e.g. so for a parameter named company you would use a parameter placeholder `<!--company-->`).

Note: the parameter and the parameter placeholder must be in the same case.

The page template processor will substitute the value of the parameter for the parameter placeholder. If the value of a parameter in a placeholder is not supplied, its value will be assumed to be blank. You can use the same parameter placeholder more than once in a given template.

Sample template customer.htm

```
<html>
  <body>
    <h1>Customer Details</h1>
    <table>
      <tr>
        <th>Customer Code</th>
        <td><!--cc--></td>
      </tr>
      <tr>
        <th>Customer Name</th>
        <td><!--company--></td>
      </tr>
    </table>
  </body>
</html>
```

Parameter XML format

The Zim application supplies the values of parameters in an XML format. The root element can have any name. The parameter names are the names of the elements which are children of the root element. The character content of each of the children of the root element is the value of the parameter.

An easy way to demonstrate this is by the following example:

Sample XML data

```
<?xml version="1.0!?">
  <customer>
    <cc>1007</cc>
    <company><![CDATA[Manning & Associates]]></company>
  </customer>
```

Note the use of the CDATA section in the above to allow the company parameter to include any characters. Alternatively the "&" character could have been replaced by its corresponding entity `&`;

XML URIs

The various XML technologies incorporated into ZimWeb may require references from one source to another (e.g. one XSLT stylesheet referencing another, or a reference to an external image in XSL-FO).

The following explains precisely how references to other sources - URIs - are interpreted in different circumstances.

URIs in XML generated by the Zim database agent

If there is a relative URI in the XML generated by the Zim database agent - e.g. a reference to a DTD (document type definition) - then this is interpreted as relative to a file `zii.xml` (which does not have to exist) in the base path of the ZimWeb application.

(e.g. If you have installed ZimWeb in the manner described in the installation instructions then specifying that a DTD is at path `dtd/mydtd.dtd` will refer to the file `[TOMCAT_ROOT]/ZII/dtd/mydtd.dtd`.)

URIs in XSLT stylesheets

If there is a relative URI in an XSLT stylesheet - (e.g. a reference to another stylesheet or document) - then this is interpreted as relative to that stylesheet. (e.g. If you have a stylesheet `[TOMCAT_ROOT]/ZII/styles/cool.xsl` that imports or includes a stylesheet `library/valueformat.xsl` then that will refer to the file `[TOMCAT_ROOT]/ZII/styles/library/valueformat.xsl`).

URIs in XSL-FO

If there is a relative URI in XSL-FO - (e.g. a reference to an image in an external file) - then this is interpreted as relative to a file `zii.xml` (which does not have to exist) in the base path of the ZII application.

e.g. If you have installed ZimWeb in the manner described in the installation instructions then specifying a reference to an image at path `images/fop.jpg` will refer to the file `[TOMCAT_ROOT]/ZII/images/fop.jpg`.

ZimWeb Administration

ZimWeb has an administration tool - the ZIIAdmin servlet - which can be used to administer ZimWeb, either from the web or from a command line or script, which can be scheduled.

Web administration

If you invoke the ZIIAdmin servlet from a web browser then you will see a page with various parts under different headings:-

Database Status
Allows you to monitor, start and stop the Zim agent sessions e.g. if you want to prevent access to the Zim application for maintenance purposes. The current status of the Zim agent sessions is indicated as "Database connections are RUNNING" or "... NOT RUNNING". If the connections are running, then an adjacent button allows you to "Shutdown Database Connections"; if they are not running, then the button allows you to "Start Database Connections".
Debug Status
Indicates whether client debugging capability is enabled or disabled - press the adjacent button to toggle the status as necessary.
Initialisation Parameters
Shows the values of the startup parameters for the ZimWeb.
XSLT Stylesheet cache
Indicates the current contents of the XSLT stylesheet cache. If you need to clear the cache - e.g. if you have changed an XSLT stylesheet which is cached into memory and thus will not be read - then click the adjacent button "Clear XSLT Stylesheet Cache".
Page Template cache
Indicates the current contents of the page template cache. If you need to clear the cache - e.g. if you have changed a page template which is cached into memory and thus will not be read - then click the adjacent button "Clear Page Template Cache".
FOP driver pool
Indicates the current number of FOP (Apache Formatting Object Processor) driver instances and the number checked out (in active use at the time that you start the web-administration tool). If the ZII receives more than one request for XSL-FO rendering at the same time then it will automatically increase the number of driver instances available. If you need to reset the FOP driver pool then click the adjacent button "Reset FOP Driver Pool".

Command line or scripted administration

The ZIIAdmin servlet can also be invoked from the command line or by a script. The servlet itself requires two parameters:-

- action - set to indicate what query or action the servlet is to perform. Permitted actions include:-
 - conn-status - return the status of the Zim database agent connections.
 - conn-start - attempt to start the Zim database agent connections.
 - conn-stop - attempt to stop the Zim database agent connections.

- debug-status - return the debug capability status.
- debug-enable - enable the debug capability.
- debug-disable - disable the debug capability.
- clear-style-cache - clear the XSLT stylesheet cache.
- clear-page-template-cache - clear the page template cache.
- reset-xslfo-driver-pool - reset the XSL/FO driver pool.
- content - set to text so that the servlet will return plain text instead of HTML.

CallURL utility

To enable the servlet to be called from the command line or a script, ZimWeb includes the CallURL utility to enable calls to a URL, with BASIC authentication details if necessary.

The full syntax for CallURL is:

```
java -cp <path>/zii.jar biz.zim.util.CallURL <URL> [ <username> <password> ]  
... to invoke the URL <URL>, optionally with BASIC authentication for user  
<username> and password <password>.
```

(e.g. To start the database connections on a local running ZimWeb installation or a default Tomcat installation, if the ZimWeb file zii.jar is in the current directory, and the ZimWeb administration servlet has security requiring the username admin and password friend, you could use the command:-

```
java -cp zii.jar http://localhost:8080/ZII/servlet/ZIIAdmin?action=conn-start&content=text admin  
friend)
```

Creating Secure ZimWeb Applications

Some key guidelines for creating secure ZimWeb applications include:-

- Secure access to the ZimWeb Administration servlet so that unprivileged users cannot access it. *The installation instructions for Tomcat* (See Installation) show how to reserve it for a given user name and password.
- Ensure that the client *DEBUG facility* is disabled by default by *setting the allow-debug configuration option to no*.
- Specify a TEMPLATE in the security configuration for all procedures - this is described in the configuration file extensions.
- Avoid or restrict using Zim sessions if at all possible, as they are openings to a denial of service by committing all the Zim database agents:
 - Use http session variables to preserve state information instead - see *parameter sources* for more details about this.
 - Use the most secure setting possible for the *zim-session-security* configuration option- preferably disable persistent Zim sessions if they are not required, or track Zim sessions in the http session. In addition, existing unmodified ZimCGI applications can also have their security improved.
- Record the authentication of a user in an http session parameter e.g. *session.AuthenticatedUser*, which is present in *all* procedure templates and checked by *all* procedures. The initial authentication can be performed by the web server or the Zim application.
- Place XSLT stylesheets and templates in a secure location that cannot be accessed directly by clients. A suitable location is under the WEB-INF directory of the application, as demonstrated by the ZimWeb example application.
- Be careful about what XML information is output by the application - remember that the client can specify *style=none* in any request to see the raw, unstyled XML data.

Bibliography

XML Technologies

The following books will be particularly useful in explaining the XML technologies that are crucial to using the ZimWeb effectively.

- XML By Example (2nd Edition) and Applied XML Solutions by Benoit Marchal - these will give you a good overview of the XML landscape.
- XSLT Programmer's Reference 2nd Edition by Michael Kay - this is *the* reference book for XSLT, though it is a bit "heavy" to get you started - XML By Example (2nd Edition) is a much gentler introduction. But he knows his stuff - he wrote the Saxon XSLT processor.
- XSLT by Doug Tidwell - he's a "technology evangelist" from IBM.
- XSLT Cookbook by Sal Mangano
- XSL-FO by Dave Pawson.

Java

If you want to learn about Java then take a look at these books.

- **General Java programming**
 - Learning Java, 2nd Edition by Patrick Niemeyer and Jonathan Knudsen - a great introduction to the Java language.
 - Java in a Nutshell, 4th Edition by David Flanagan, and Java Enterprise in a Nutshell, 2nd Edition by William Crawford, Jim Farley and David Flanagan - together, these provide a useful reference for the Java language.
- **Server-side Java programming**
 - Java Servlet Programming, 2nd Edition by Jason Hunter with William Crawford - this is a very clear exposition of server-side Java programming.
 - Professional Apache Tomcat by lots of authors. This recently-published book is really helpful in trying to get the best out of the Tomcat servlet engine.
- **XML Processing with Java**
 - Java & XML, 2nd Edition by Brett McLaughlin - this is an excellent explanation of this topic.
 - Java and XSLT by Eric M. Burke - another excellent O'Reilly book.

End User Licence Agreement

ZIM TECHNOLOGIES INTERNATIONAL INC.

IMPORTANTE: A versão em português do "License Agreement" fornecida abaixo do texto em inglês é somente para referência. Em concordando com os termos e condições da versão em português do "License Agreement" você está concordando com a versão em inglês.

Software License Agreement

IMPORTANT: BEFORE INSTALLING THE PACKAGE CONTAINING THE SOFTWARE, CAREFULLY READ THE FOLLOWING TERMS AND CONDITIONS. IF YOU DO NOT AGREE WITH THEM, PROMPTLY RETURN THE PACKAGE UNUSED. INSTALLING THE PACKAGE INDICATES THAT YOU HAVE ACCEPTED THESE TERMS AND CONDITIONS.

1. GRANT OF LICENSE:

ZIM Technologies International Inc. ("ZIM") hereby grants to you, the licensee, a limited, non-exclusive, non-transferable license to use the Software in accordance with the terms and conditions of this License Agreement. The Software consists of (a) the computer programs and related data in the form delivered to you with this License Agreement (the "Programs") and any updates or error corrections thereto and (b) the online manual and related printed documentation (the "Documentation"). The Programs are delivered on one or more magnetic disks or CD-ROMs (the "Media").

2. TERM:

This License Agreement is effective from the day you open the sealed Software package, and continues until terminated. You may terminate this License Agreement at any time by destroying the Software together with all copies in any form. ZIM may terminate this License Agreement if you fail to comply with any of its terms or conditions. You agree upon such termination to destroy the Software together with all copies in any form.

3. PROPRIETARY RIGHTS:

You acknowledge that the Software is the exclusive property of ZIM and/or its licensors, and that title and full ownership rights to the Software furnished under this License Agreement remain with ZIM and/or its licensors. By accepting this license, you do not become the owner of the Software, but you do have the right to use the Software as specified in this License Agreement.

4. USE OF THE SOFTWARE:

Unless you have obtained a LAN license from ZIM, you may use the Software on any computer, provided that it is used only on one computer at any one time and that the Programs are stored on the hard disk(s) of only one computer at any one time. You may not transfer the Software from one computer to another over a network or by other means of electronic communications. If you wish to use the Software on more than one computer at one time, you must either license another copy of the Software, or obtain a LAN license from ZIM. If you have obtained a LAN license, you may use the Software concurrently on the number of computers specified in your LAN license. You may make (2) copies of the Software for backup or archival

purposes. You agree not to modify, obscure, or delete any notices of ZIM's proprietary rights included in or on the Programs, Documentation, or Media, and you agree to include all such notices on any copies which you make. Unless otherwise expressly permitted, you agree not to use the software in connection with any online service, such as but not limited to any dial-up, remote access, interactive, Internet-based or other on-line service or World Wide Web site supported by one or more servers, from which you may receive compensation from subscription fees, advertising or other sources.

5. CONFIDENTIALITY:

You acknowledge and agree that the Software constitutes confidential information of, and is trade secret to, ZIM. You agree not to disclose the Software to any of your agents, independent contractors or consultants, or any of their employees, without the prior written consent of ZIM. You will take all reasonable steps to protect the Software from unauthorized reproduction, publication, disclosure or distribution. You may not use the Software to process the data of any third party. You may not modify the Software or merge the Software into any other computer programs or software. Except to the minimum extent permitted by law, you may not reverse engineer, reverse assemble or reverse compile the Software in whole or in part.

6. LIMITED WARRANTY:

ZIM warrants the enclosed Media and Documentation (the "Materials") to be free from defects in materials or workmanship, in normal use and service, for a period of ninety (90) days from the date of delivery. If a defect in the Materials should appear during this ninety (90) day period, as ZIM's sole obligation and your sole remedy, you may return the Materials to ZIM (or its authorized distributor from whom you obtained the Materials) for replacement without charge. If ZIM or such distributor is unable to satisfactorily replace the defective Materials you are entitled to receive a full refund of the license fee paid upon return of the Software. If the defect in the Materials has resulted from accident, abuse, or misapplication of the Materials, then neither ZIM nor its distributor shall have any responsibility to replace the Materials or refund the license fee paid under the terms of this limited warranty.

ZIM further warrants that the Programs will operate substantially in conformance with the Documentation for a period of ninety (90) days from the date of delivery of the Media (the "Warranty Period"). As ZIM's sole obligation and your sole remedy under this limited warranty, ZIM (or its authorized distributor from whom you obtained the Materials) shall either (i) use commercially reasonable efforts to make corrections to the Programs if they do not function in accordance with the Documentation, provided you have given ZIM written notice of non-conforming Programs during the Warranty Period, or (ii) at ZIM's option, refund the license fee upon return of the Software.

ZIM does not warrant that the contents of the Documentation or the operation of the Programs will be error free, or that ZIM will correct all Program errors, or that the Software will meet your requirements. All updates of and error corrections to the Software are provided "AS IS" without any warranty of any kind.

THESE LIMITED WARRANTIES GIVE YOU SPECIFIC LEGAL RIGHTS. YOU MAY ALSO HAVE OTHER RIGHTS WHICH VARY FROM JURISDICTION TO JURISDICTION. EXCEPT FOR THE WARRANTIES DESCRIBED HEREIN, ZIM MAKES NO EXPRESS OR IMPLIED WARRANTIES OR REPRESENTATIONS OF ANY KIND IN RESPECT OF THE SOFTWARE OR THE MEDIA, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OR

CONDITIONS OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

7. LIMITATION OF LIABILITY:

NEITHER ZIM NOR ANYONE ELSE WHO HAS BEEN INVOLVED IN THE CREATION, PRODUCTION, OR DELIVERY OF THE SOFTWARE SHALL BE LIABLE TO YOU OR ANY OTHER PARTY FOR ANY AMOUNT IN EXCESS OF THE LICENSE FEE PAID BY YOU, FOR ANY DAMAGES RESULTING FROM THE USE OR PERFORMANCE OF THE SOFTWARE OR THE MEDIA OR ARISING OUT OF ANY BREACH OF ANY WARRANTY. IN NO EVENT SHALL ZIM BE LIABLE FOR SPECIAL, INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES, INCLUDING, WITHOUT LIMITATION, DAMAGES OR LOSS TO EQUIPMENT, LOSS OF PROFITS OR REVENUE, LOSS OF GOODWILL, INCREASED EXPENSES OF OPERATION, COST OF CAPITAL, OR THE CLAIMS OF THIRD PARTIES, HOWSOEVER CAUSED, EVEN IF ZIM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Some jurisdictions do not allow the exclusion or limitation of indirect, incidental or consequential damages, so the above limitations may not apply to you.

8. MAINTENANCE AND SUPPORT:

You may purchase maintenance and support of the Software on an annual basis at ZIM's then current rates. Maintenance and support includes telephone support, correction of defects in the Software and the Documentation and updates. Updates are provided as part of new releases of the Software.

9. GOVERNING LAW:

This License Agreement is to be governed by and interpreted in accordance with the laws of the Province of Ontario, Canada, and all federal laws applicable therein, excluding principles of conflict of laws. You agree that the UN Convention on Contracts for the International Sale of Goods (Vienna, 1980) shall not apply to this License Agreement nor to any dispute or transaction arising out of this License Agreement. You agree to comply with the export laws and regulations of Canada, which place restrictions on the export and re-export of certain products and technologies, including software. You may not export the Software or the Media to countries designated by such export laws and regulations as prohibited destinations. By using the Software, you are agreeing to the foregoing and you are representing and warranting that you are not located in, under the control of, or a national or resident of any such country.

10. GENERAL:

You agree that this License Agreement is the complete and exclusive statement of the agreement between ZIM and you and that it supersedes any proposed or prior agreement, oral or written, and any other communications of the parties relating to the subject matter hereof, including any purchase order in respect of the Software. This License Agreement may be modified or supplemented only by a document signed by both ZIM and you. If any provision of this License Agreement shall be declared invalid or unenforceable, such provision shall be deemed to be deleted, and it shall not affect any other provision of this License Agreement. Any attempt to assign, transfer or sublicense your rights under this License Agreement, without the prior written consent of ZIM, shall be void and of no effect. The Software is provided with RESTRICTED RIGHTS. Use, duplication or disclosure by the U.S. GOVERNMENT is subject to the restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 and its successors [DoD procurements] or in subparagraphs (c)(1) and (2) of the

Commercial Computer Software-Restricted Rights clause at FAR 52.227-19 and its successors [civilian agency procurements] or in subparagraphs (a) and (b) of the Restricted Rights Notice in Alternative III of the Rights in Data General clause at FAR 52.227-14 and its successors [civilian agency procurements].

Contractor/manufacturer is ZIM Technologies International Inc., 20 Colonnade Road, Suite 200, Nepean, Ontario, Canada.

11. ACKNOWLEDGMENT:

YOU ACKNOWLEDGE THAT YOU HAVE READ THIS LICENSE AGREEMENT, UNDERSTAND IT AND AGREE TO BE BOUND BY ITS TERMS AND CONDITIONS.

ZIM TECHNOLOGIES INTERNATIONAL INC.

IMPORTANTE: A versão em português do "License Agreement" fornecida abaixo é somente para referência. Em concordando com os termos e condições desta versão em português do "License Agreement" você está concordando com a versão em inglês.

Contrato de Licenciamento de Uso de Software

IMPORTANTE: ANTES DE INSTALAR O SOFTWARE, LEIA COM ATENÇÃO OS TERMOS E CONDIÇÕES QUE SEGUEM. CASO VOCÊ NÃO OS ACEITE, DEVOLVA IMEDIATAMENTE O PRODUTO NÃO USADO. INSTALANDO O PRODUTO INDICA QUE VOCÊ CONCORDOU COM OS TERMOS E CONDIÇÕES CONTIDOS NESTE CONTRATO.

1. CONCESSÃO DE LICENÇA:

A ZIM Technologies International Inc. ("ZIM") concede a você, o licenciado, através deste Contrato, uma limitada, não exclusiva, não transferível licença de uso de Software em concordância com os termos e condições deste Contrato. O software consiste de: (a) os programas de computador e dados relativos a eles na forma entregue a você juntamente com este Contrato ("Programas") e qualquer atualização adicional e correção de erros; e (b) o manual on-line e a documentação relativa impressas ("Documentação"). Os programas são entregues em um ou mais discos magnéticos ou CD-ROM ("Meio").

2. TERMO:

Este Contrato entra em vigor a partir do dia que você abre a embalagem selada do Software e continua até que seja terminado. Você pode terminar este Contrato a qualquer momento destruindo o Software juntamente com todas as cópias em qualquer formato. A ZIM pode terminar este Contrato caso você desobedeça a qualquer um dos termos ou condições deste Contrato. No caso de término deste Contrato, você concorda em destruir o Software juntamente com todas as cópias em qualquer formato.

3. DIREITOS DE PROPRIEDADE:

Você reconhece que o Software é de propriedade exclusiva da ZIM e/ou de seus licenciadores, e que a titularidade e o todo o direito de posse do Software fornecido com este Contrato continuam de posse da ZIM e seus licenciadores. Ao aceitar este Contrato, você não se torna o dono do Software, você tem, entretanto, o direito de uso do Software como especificado neste Contrato.

4. USO DO SOFTWARE:

A menos que você tenha obtido, da ZIM, uma licença LAN, você poderá usar o Software em qualquer computador, desde que ele seja usado em somente um computador em um dado momento e que os programas estejam armazenados no(s) disco(s) rígido(s) de apenas um computador num dado momento. Você não pode transferir o Software de um computador para outro através de rede de computadores ou por qualquer outra forma de comunicação eletrônica. Caso você queira usar o Software em mais de um computador simultaneamente, você deve: ou licenciar outra cópia do Software; ou obter, da ZIM, uma licença LAN. Caso você tenha obtido uma licença LAN, você pode usar o Software simultaneamente no número de computadores especificado por sua licença LAN. Você pode fazer 2 (duas) cópias com o propósito de 'cópia de segurança' ou arquivo. Você concorda em não modificar, ocultar, ou remover qualquer aviso dos direitos de propriedade da ZIM incluídos nos Programas, na Documentação, ou no Meio, e você concorda em colocar este aviso em qualquer cópia que você faça. A menos que, de outra maneira expressamente permitida, você concorda em não usar o software em conexão com qualquer serviço on-line, como por exemplo, mas não limitado a, qualquer acesso discado, acesso remoto, interativo, baseado na Internet ou outro serviço on-line; ou Web site funcionando em um ou mais servidor(es), de onde você possa receber compensação através de taxa de assinatura, anúncio ou outras fontes.

5. CONFIDENCIALIDADE:

Você reconhece e concorda que o Software constitui-se de informações confidenciais da ZIM e é segredo de negócio. Você concorda em não divulgar o Software, a menos que tenha uma permissão da ZIM por escrito, para nenhum dos seus agentes, prestadores de serviços ou consultores, ou qualquer empregado dos mesmos. Você tomará todas as medidas cabíveis para proteger o Software de reprodução, publicação, divulgação ou distribuição não-autorizada. Você não pode usar o Software para processar informações de terceiros. Você não pode modificar o Software ou incorporá-lo com qualquer outro programa de computador ou software. Exceto, pelo alcance mínimo permitido por lei, é vedada a engenharia reversa, a descompilação e desmontagem, do todo ou de parte, do Software.

6. GARANTIA LIMITADA:

A ZIM garante que o Meio e a Documentação inclusos ("Materiais") estão livre de defeitos de material e de produção, em serviço e uso normal, por um período de 90 (noventa) dias a partir da data de entrega. Na ocorrência de defeito nos Materiais, durante este período de 90 (noventa) dias, como a única obrigação da ZIM e seu único recurso, você pode retornar os Materiais para a ZIM (ou para o seu distribuidor autorizado de quem você obteve os Materiais) para substituição livre de custos. Caso a Zim ou o seu distribuidor não possa substituir, satisfatoriamente, os Materiais defeituosos, você tem direito a receber o reembolso total da licença paga, no retorno do software. Caso o defeito nos Materiais seja resultado de acidente, abuso, ou mal uso dos Materiais, aí então, nem a ZIM, nem seu distribuidor será responsável pela substituição dos Materiais ou pelo reembolso da licença paga sob os termos desta garantia limitada.

A ZIM ainda garante que os Programas estarão funcionando substancialmente em concordância com a Documentação por um período de 90 (noventa) dias a partir da data da entrega do Meio ("Período de Garantia"). Como única obrigação da ZIM e seu único recurso sob esta garantia limitada, a ZIM (ou o seu distribuidor autorizado de

quem você obteve os Materiais) pode (i) tomar medidas comerciais cabíveis para corrigir os Programas, caso eles não funcionem em concordância com a documentação, ou (ii) a critério da ZIM, reembolsar a licença paga, no retorno do Software.

A ZIM não garante que o conteúdo da Documentação ou a operação dos Programas estarão livre de error, ou que a ZIM irá corrigir todos os erros do Programa, ou que o Software irá atender aos seus (do licenciado) requisitos. Todas as atualizações e correções de erros no Software são fornecidas "no estado" sem nenhuma garantia de nenhum tipo.

ESTA LIMITAÇÃO DE GARANTIA LHE DÁ DIREITOS LEGAIS ESPECÍFICOS. VOCÊ PODE TER OUTROS DIREITOS QUE VARIEM DE JURISDIÇÃO PARA JURISDIÇÃO. EXCETO PARA AS GARANTIAS AQUI DESCRITAS, A ZIM NÃO EXPRESSA OU INDICA GARANTIAS OU REPRESENTAÇÕES DE QUALQUER TIPO, NO QUE DIZ RESPEITO AO SOFTWARE OU AO MEIO, INCLUINDO, MAS NÃO LIMITADO, ÀS GARANTIAS IMPLÍCITAS OU CONDIÇÕES DE COMERCIALIZAÇÃO OU ADAPTAÇÃO A UM PROPÓSITO PARTICULAR.

7. LIMITAÇÃO DE RESPONSABILIDADE:

NEM A ZIM NEM NINGUÉM MAIS QUE TENHA SIDO ENVOLVIDO NA CRIAÇÃO, PRODUÇÃO, OU ENTREGA DO SOFTWARE PODE SER RESPONSÁVEL PERANTE VOCÊ OU QUALQUER TERCEIRO POR QUALQUER SOMA SUPERIOR A PAGA, POR VOCÊ, PELA LICENÇA; POR QUALQUER DANO CAUSADO PELO USO OU DESEMPENHO DO SOFTWARE OU DO MEIO; OU PELO RESULTADO DE QUALQUER QUEBRA DE QUALQUER GARANTIA. EM NENHUMA HIPÓTESE A ZIM SERÁ RESPONSÁVEL POR ESPECIAL, INDIRETO, ACIDENTAL OU CONSEQÜENTE DANO, INCLUINDO SEM LIMITAÇÃO, DANOS OU PERDAS DE EQUIPAMENTOS, PERDA DE LUCRO OU RECEITA, PERDA DE BONS RELACIONAMENTOS COMERCIAIS, AUMENTO DE DESPESAS OPERACIONAIS, CUSTO DE CAPITAL, OU DE RECLAMAÇÕES DE TERCEIROS, NÃO IMPORTANDO QUAL FOI A CAUSA, MESMO QUE A ZIM TENHA SIDO NOTIFICADA DA POSSIBILIDADE DESTES DANOS. Algumas jurisdições não permitem a exclusão ou limitação de danos indiretos, acidentais ou consequentes, assim a limitação acima pode não se aplicar a você.

8. MANUTENÇÃO E SUPORTE:

Você pode adquirir manutenção e suporte para o Software anualmente pelo preço da ZIM vigente na época. Manutenção e suporte incluem: suporte via telefone; correção de defeitos no Software e na Documentação; e atualizações. Atualizações são fornecidas como parte de novos lançamentos do Software.

9. LEGISLAÇÃO REGULADORA:

Este Contrato é regido e interpretado em concordância com as leis da Província de Ontário, Canadá, e todas as leis federais aqui aplicáveis, excluído conflitos de princípios de leis. Você concorda que a Convenção da UN sobre Contatos para a Venda Internacional de Produtos (Viena, 1980) não se aplica a este Contrato, nem a qualquer disputa ou transação que surja deste Contrato. Você concorda em cumprir com as leis e regulamentações do Canada, que têm restrições na exportação e re-exportação de certos produtos e tecnologias, incluindo Software. Você não pode exportar o Software ou o Meio para países designados como destinos proibidos por estas leis e regulamentos de exportação. Usando o Software, você está concordando com o que foi previamente mencionado e você está representando e garantindo que

you are not located in, under the control of, or are a citizen or resident of any country subject to these restrictions.

10. GERAL:

You agree that this Contract is a complete and exclusive declaration between ZIM and you and that it replaces any previous contract proposed or prior, oral or written, and any other communication between the parties relative to this matter, including any purchase order with reference to the Software. This Contract may be modified or amended only through a document signed by both parties: ZIM and you. If any condition of this Contract is to be declared invalid or unenforceable, such condition shall be considered to be excluded, and this shall not affect any other condition of this Contract. Any attempt to assign, transfer, or sublicense your rights under this Contract, without the prior consent of ZIM, in writing, shall be canceled and without effect. The Software is provided with RESTRICTED RIGHTS. Use, duplication or disclosure, by the Government of the USA is subject to restrictions, as specified in, sub-paragraph (c) (1) (ii) of the 'Rights in Technical Data and Computer Software clause at DFARS 252.227-7013' and its subsequent [DoD procurements] or in sub-paragraphs (c)(1) and (2) of the 'Commercial Computer Software-Restricted Rights clause at FAR 52.227-19' and its subsequent [civilian agency procurements] or in sub-paragraphs (a) and (b) of the Restricted Rights Notice in Alternative III of the Rights in Data General clause at FAR 52.227-14' and its subsequent [civilian agency procurements]. The Manufacturer is ZIM Technologies International Inc., 20 Colonnade Road, Suite 200, Nepean, Ontario, Canada.

11. RECONHECIMENTO:

You acknowledge that you have read, understood and agreed to this Contract and its terms and conditions.

Sample Servlet Configuration File web.xml

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE web-app (View Source for full doctype...)>
- <web-app>
-   <servlet>
     <servlet-name>ZII</servlet-name>
     <servlet-class>biz.zim.zii.ZIIServlet</servlet-class>
-   <init-param>
        <param-name>auto-start</param-name>
        <param-value>yes</param-value>
     </init-param>
-   <init-param>
        <param-name>config-file</param-name>
        <param-
          value>C:\PROGRA~1\Zim\ZimWeb\conf\zimweb.cfg
        </param-value>
     </init-param>
-   <init-param>
        <param-name>log-file</param-name>
        <param-
          value>C:\PROGRA~1\Zim\ZimWeb\logs\zimweb.log
        </param-value>
     </init-param>
-   <init-param>
        <param-name>verbose</param-name>
        <param-value>no</param-value>
     </init-param>
-   <init-param>
        <param-name>use-xslt-output-type</param-name>
        <param-value>yes</param-value>
     </init-param>
-   <init-param>
        <param-name>default-session-timeout</param-name>
        <param-value>1800</param-value>
     </init-param>
-   <init-param>
        <param-name>default-cookie-timeout</param-name>
        <param-value>15552000</param-value>
     </init-param>
-   <init-param>
        <param-name>zim-session-security</param-name>
        <param-value>http-session</param-value>
     </init-param>
-   <init-param>
```

```

        <param-name>allow-debug</param-name>
        <param-value>no</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
</servlet>
- <servlet>
    <servlet-name>ZIIAdmin</servlet-name>
    <servlet-class>biz.zim.zii.ZIIAdminServlet</servlet-class>
</servlet>
- <servlet-mapping>
    <servlet-name>ZII</servlet-name>
    <url-pattern>/servlet/ZII/*</url-pattern>
</servlet-mapping>
- <servlet-mapping>
    <servlet-name>ZIIAdmin</servlet-name>
    <url-pattern>/servlet/ZIIAdmin</url-pattern>
</servlet-mapping>
- <security-constraint>
    - <web-resource-collection>
        <web-resource-name>ZimWebAdmin</web-resource-name>
        <url-pattern>/servlet/ZIIAdmin</url-pattern>
        <http-method>GET</http-method>
        <http-method>POST</http-method>
    </web-resource-collection>
    - <auth-constraint>
        <role-name>zimwebadmin</role-name>
    </auth-constraint>
</security-constraint>
- <login-config>
    <auth-method>BASIC</auth-method>
    <realm-name>ZimWeb Administration</realm-name>
</login-config>
- <security-role>
    <role-name>zimwebadmin</role-name>
</security-role>
</web-app>

```