

Hectares BC

Infrastructure and Development Environment

Prepared For

Matt Austin
Project Manager
Ministry of Environment

Prepared By

Refractions Research Inc.
#300 – 1207 Douglas Street
Victoria, BC
www.refractions.net

Document Tracking

Version: 1
Last Updated: Oct. 23, 2007



Refractions
RESEARCH
THE GEOSPATIAL EXPERTS



REVISION HISTORY

Date	Version	Author	Description of Changes
Oct 23, 2007	1	Emily Gouge (Refractions Research)	First draft

CONTENTS

Table of Contents

1 Introduction.....	1
1.1 Purpose.....	1
1.2 Target Audience.....	1
1.3 References.....	1
2 Infrastructure Overview.....	2
3 Software Packages.....	3
3.1 Production Environment.....	3
3.2 Development Tools.....	3
4 Development Standards.....	5
4.1 Coding Formatting.....	5
4.2 Testing.....	5
4.3 Code Comments and JavaDocs.....	5
4.4 Document Naming.....	5
4.5 Version Control.....	5
5 System Setup.....	7
5.1 Logging System Setup.....	7
5.2 Web Server Setup.....	8
5.2.1 Database Pooling.....	8
5.2.2 Database Users.....	8
5.3 TileCache Configuration.....	9
5.4 HaBC Web Application Options.....	9
5.4.1 BaseMap Options.....	10
5.4.2 Tile Resolution.....	10
5.5 Development Environment Set-Up.....	10
5.5.1 web.xml.....	11
5.5.2 ROOT.xml.....	11
5.5.3 log4j.properties.....	12
5.5.4 Eclipse Build Paths.....	12
5.6 Trac Setup.....	13
5.6.1 Ticket Setup.....	13
5.6.2 Wiki.....	14

1 INTRODUCTION

1.1 PURPOSE

The purpose of this document is to describe the infrastructure and development environment for the Hectares BC system.

1.2 TARGET AUDIENCE

The intended audience for this document is technically proficient and interested in the tools used to develop and run the Hectares BC System. During application development and implementation, it will be used primarily by:

- Matt Austin, Project Manager
- Members of the Hectares BC Working Group
- Refractions Development Team

1.3 REFERENCES

The following resources were used as references in the creation of this system documentation:

- Hectares BC Project Charter
- Hectares BC Requirements Specification
- Hectares BC System Design Document

2 INFRASTRUCTURE OVERVIEW

The following diagram provides a high-level outline of the Hectares BC system infrastructure. The main Hectares BC application is a web based application that accesses multiple web services that support Hectares BC processing and visualization.

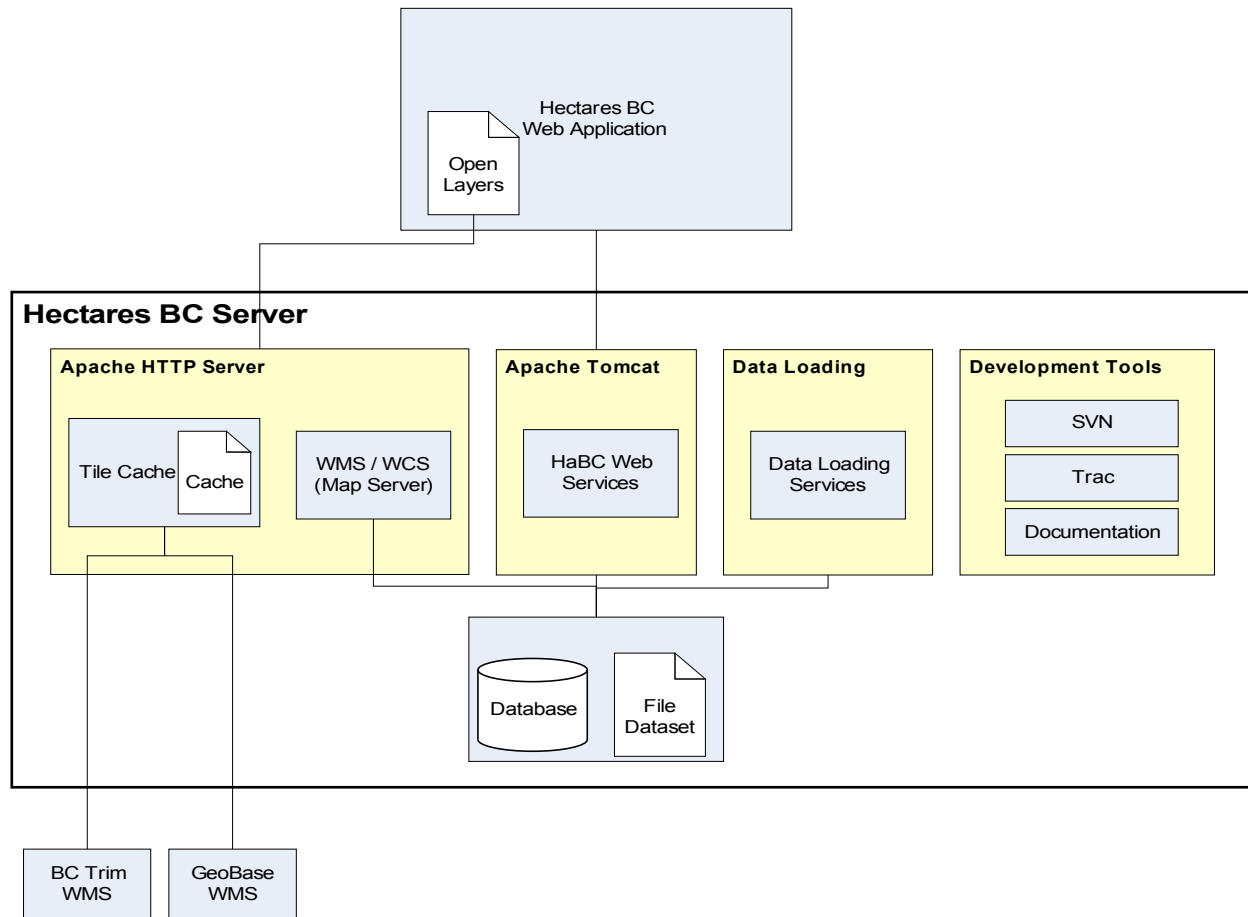


Figure 1: Hectares BC Infrastructure

All tools required for development and production systems will be installed on a single self contained Hectares BC server. This includes tools such as web servers, the database, svn repositories, and the trac system (wiki and issue tracking) as well as all documentation, data, and scripts.

3 SOFTWARE PACKAGES

3.1 PRODUCTION ENVIRONMENT

The following list outlines the software packages used in the production environment:

- **CentOS 5** – Linux operating system (Red Hat Enterprise 5 compatible).
- **Apache Tomcat** – A web container used for running the HaBC application.
- **Apache HTTP Server** – A web server for supporting Tile Cache.
- **OpenLayers** – A Javascript library adding dynamic maps to any web page.
- **TileCache & Python** – A Python based WMS/TMS service with a caching mechanism.
- **PostGIS / PostgreSQL** – A spatial enabled database management system.
- **Perl** – Scripting language to be used for data conversions.
- **GRASS** – An open source GIS software suite that will be used to provide some vector to raster conversions.
- **GDAL / OGR** – Raster / Vector conversion library used for loading and manipulating raster and vector files.
- **MapServer** – Open source tool for rendering spatial data maps.

3.2 DEVELOPMENT TOOLS

The following tools are used in the development of the Hectares BC system:

- **Eclipse** – Eclipse is an integrated development environment, providing an environment for the coding, testing, and versioning of the software developed.
- **SVN (Subversion)** – Subversion is an open-source revision control system used.
- **Maven2** – A build manager for Java projects.
- **Google™ Web Toolkit (GWT)** – An open source Java software development framework for writing AJAX applications using the Java programming language. GWT development is completed in Java; the GWT toolkit compiles the Java classes into JavaScript, which is required to run the web application.

- **GWT Drag and Drop** – An extension to the GWT toolkit that supports the implementation of drag and drop.
- **OpenLayers GWT** – A wrapper written around the OpenLayers Javascript library to integrate with the GWT toolkit.
- **JUnit** – A Java based testing framework for writing and running test cases.
- **Log4J** – An open source Java library to allow the addition of logging to an application.
- **Java V 1.4**– Programming language used in the development of HaBC. Java Version 1.4 is being used because currently this is all the GWT supports.
- **WinCHM** – A tool for authoring help pages.
- **Trac** – A wiki and issue tracking system for software development projects.

4 DEVELOPMENT STANDARDS

4.1 CODING FORMATTING

All Java source code following the *Eclipse [built-in]* formatting standards.

All Java source files have the following copyright information located at the top of the file.

```
/*
 * Copyright 2007 Nature Trust of British Columbia
 */
```

All Java packages start with the package name `org.hectaresbc`.

4.2 TESTING

Where appropriate, automated JUnit test cases have been written. All test cases are added to the `test` folder.

In addition, User Story level test scripts are compiled and performed for each User Story. These are compiled into a single TestScript document that is iterated through at the end of each development cycle. These tests focus on the user interface.

4.3 CREDITS & CHANGE LOG

A CREDITS file will be produced that lists the people and organizations that have contributed to the Hectares BC system.

A CHANGELOG will be produced that tracks the features modified and bugs fixed between versions.

4.4 CODE COMMENTS AND JAVADOCS

Standard JavaDocs will be generated from the source code; therefore each class, method, and constant should be documented in a manner consistent with JavaDoc formatting.

4.5 DOCUMENT NAMING

All documents written for Hectares BC will follow the following naming convention:

HaBC_*documentname*_revN.odt

4.6 VERSION CONTROL

Subversion (SVN) is used as the version control system. Any software or scripts written for this project will exist in this repository. The repository will have the following structure:

<code>./trunk/</code>	The main repository containing the current version of the software.
<code>./trunk/apps</code>	For the Hectares BC application related code.
<code>./trunk/lib</code>	For libraries developed to support the Hectares BC application development.
<code>./trunk/database</code>	For any scripts or tools used for creating database information.
<code>./trunk/scratchpad</code>	For testing.
<code>./tags/</code>	For tags. A tag will be created at the end of each sprint. Tags are named with the sprint number and date (SprintN.MM-DD-YYYY).
<code>./branches/</code>	

5 SYSTEM SETUP

5.1 LOGGING SYSTEM SETUP

The Hectares BC system uses the Apache Log4J library for logging.

The logging system is configured through a `log4j.properties` file placed in the web application directory (`HaBC/WEB-INF/classes/log4j.properties`). This file controls what logging levels are recorded, where the logging information is written to, and what information is logged. For detailed information on the setup of the `log4j.properties` file see the Log4j documentation: <http://logging.apache.org/log4j/1.2/manual.html>

The default setup is to log levels DEBUG and higher; to a `habc_errors.log` file.

```
log4j.rootLogger=DEBUG, A1
log4j.appender.A1=org.apache.log4j.FileAppender
log4j.appender.A1.layout=org.apache.log4j.PatternLayout
log4j.appender.A1.File=habc_errors.log

# Print the date in ISO 8601 format
log4j.appender.A1.layout.ConversionPattern=%d [%t] %-5p %c - %m%n
```

The system makes use of the following log levels:

Level	Description
FATAL	A error has occurred that prevents the system from continuing in any way.
ERROR	A error has occurred, however the system can continue.
WARN	An unexpected state is reached.
INFO	A particular high level task has been started.
DEBUG	Used by developers for debugging tasks.
TRACE	Contains detailed information about what the system is doing, including all database queries processed.

Table 1: Logging Levels

5.2 WEB SERVER SETUP

5.2.1 Database Pooling

Database connections are managed using the Tomcat 6 JNDI `InitialContext` implementation. The database resources are referenced in the `web.xml` file and defined in the application specific configuration.

The web.xml requires the following resource references:

```
<resource-ref>
  <description>HaBC Postgresql Database Connection</description>
  <res-ref-name>habcdb</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

The application specific configuration for our application is provided through the `./META-INF/context.xml` file. For Tomcat6 this file contains the following information. For other versions of Tomcat refer to the Tomcat documentation for setting up the database resources.

```
<Context docBase="habc" path="" reloadable="true">
  <Resource name="habcdb" auth="Container"
    type="javax.sql.DataSource" driverClassName="org.postgresql.Driver"
    url="jdbc:postgresql://<server>:<port>/<databasename>"
    username="habc_user" password="" maxActive="2" maxIdle="2"
    maxWait="-1"/>
</Context>
```

In addition, the `postgresql-x.x.jar` file must be placed in the `Tomcat/lib` folder.

5.2.2 Database Users

To support the HaBC Application the following database user accounts are required:

- **habc_user** – Used for all connections through the web interface to the database. At a minimum this user requires select access to all metadata and data tables, and select, update, delete access to persistent tables.

5.3 TILECACHE CONFIGURATION

TileCache requires a configuration file that contains information about all the layers available through the TileCache interface. This file, `tilecache.cfg`, contains the following information. Note that this configuration may change slightly as basemap layers are tweaked to improve the user experience.

```
[bctrim]
type=WMSLayer
url=http://openmaps.gov.bc.ca/mapserver/base?service=wms
extension=gif
srs=EPSG:3005
layers=MS
```

```

levels=8
bbox=159600, 173500,1881200,1747900
extent_type=loose
maxResolution=3200
metaTile=true

[geobase]
type=WMSLayer
url=http://wms.geobase.ca/wms-bin/cubeserv.cgi?
extension=jpeg
srs=EPSG:3005
layers=L70_B743:CEOWARE2:CCT/CCRS_OTTAWA
levels=8
bbox=159600, 173500,1881200,1747900
extent_type=loose
maxResolution=3200
metaTile=true

[bctrimdevel]
type=WMSLayer
url=http://openmaps.gov.bc.ca/mapserver/base?service=wms
extension=png
srs=EPSG:3005
layers=DBM_7H_MIL_POLITICAL_POLY_PS,DBM_7H_MIL_ROADS_LINE,BC_WATER_LINES_500M
,BC_TRANSPORT_LINES_500M,BC_PROV_BOUNDARIES_LINES_500M,BC_MAJOR_CITIES_POINTS
_500M,NTS_BC_WATER_LINES_125M,NTS_BC_TRANSPORT_LINES_125M,BC_BASEMAP_LINES_5K
M_BOUNDARY,NTS_BC_COASTLINE_LINES_125M,TRIM_EBM_WATERBODIES,TRIM_EBM_WATERCOU
RSES,TRIM_EBM_WETLANDS,TWTR_LINES,TTRN_LINES,DRA_DIGITAL_ROAD_ATLAS_LINE_SP
levels=8
bbox=159600, 173500,1881200,1747900
extent_type=loose
maxResolution=3200
metaTile=true

```

5.4 HABC WEB APPLICATION OPTIONS

The Hectares BC web application has multiple configuration options that are modified by changing the web.xml file associated with the application. The following sections outline the various options and default values.

5.4.1 BaseMap Options

The OpenLayers map on the land characterization web page contains multiple options for the base maps. These base map options are specified in the web.xml file but specify two environment entries (the defaults are shown below).

```

<env-entry>
    <env-entry-name>tilecache_url</env-entry-name>
    <env-entry-type>java.lang.String</env-entry-type>
    <env-entry-value>http://turtle/cgi-
bin/tilecache-1.9/tilecache.cgi</env-entry-value>

```

```

</env-entry>

<env-entry>
  <env-entry-name>basemaps</env-entry-name>
  <env-entry-type>java.lang.String</env-entry-type>
  <env-entry-value>BC Trim Basemap;bctrim;GeoBase Landsat 7
Satellite Images;geobase</env-entry-value>
</env-entry>

```

- tilecache_url - contains the tilecache location of the base maps
- basemaps - contains the tilecache layer names and UI layer names as a ';' delimited list (GUI name, tilecache layer name).

Currently, base map layers can only come from a single tile cache location.

5.4.2 Hectares BC Land Characterization Tile Resolution

The image tiles generated by the Tile Mapping Service for visualizing land characterizations can be provided using different resolutions of data. There are three valid options: 64, 128, and 256. The default is 64. This tile resolution is specified as an environment variable in the web.xml file, and can be changed by modifying this environment variable value.

```

<env-entry>
  <env-entry-name>lc_tile_resolution</env-entry-name>
  <env-entry-type>java.lang.Integer</env-entry-type>
  <env-entry-value>64</env-entry-value>
</env-entry>

```

5.5 DEVELOPMENT ENVIRONMENT SET-UP

GWT uses an older version of Tomcat and requires a slightly different set-up to be able to use the database pooling options of tomcat in the embedded server. The following sections outline which files need to be placed where (and their contents) in the GWT development environment.

5.5.1 web.xml

The web.xml file is placed in: `/tomcat/webapps/ROOT/WEB-INF/web.xml`

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app>

  <servlet>
    <servlet-name>shell</servlet-name>
    <servlet-class>com.google.gwt.dev.shell.GWTShellServlet</servlet-
class>
  </servlet>

```

```

<servlet-mapping>
  <servlet-name>shell</servlet-name>
  <url-pattern>/*</url-pattern>
</servlet-mapping>

<env-entry>
  <env-entry-name>tilecache_url</env-entry-name>
  <env-entry-type>java.lang.String</env-entry-type>
  <env-entry-value>http://turtle/cgi-
bin/tilecache-1.9/tilecache.cgi</env-entry-value>
</env-entry>

<env-entry>
  <env-entry-name>basemaps</env-entry-name>
  <env-entry-type>java.lang.String</env-entry-type>
  <env-entry-value>BC Trim Basemap;bctrim;GeoBase Landsat 7
Satellite Images;geobase</env-entry-value>
</env-entry>

<env-entry>
  <env-entry-name>tile_resolution</env-entry-name>
  <env-entry-type>java.lang.Integer</env-entry-type>
  <env-entry-value>64</env-entry-value>
</env-entry>

<resource-ref>
  <description>HaBC Postgresql Database Connection</description>
  <res-ref-name>habcdb</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
</web-app>

```

5.5.2 ROOT.xml

The ROOT.xml file contains the database connection information (similar to what is found in the context.xml in the system environment)

```

<Context docBase="ROOT" path="" reloadable="true">

  <Resource name="habcdb" auth="Container" type="javax.sql.DataSource"/>

  <ResourceParams name="habcdb">
    <parameter>
      <name>factory</name>
      <value>org.apache.commons.dbcp.BasicDataSourceFactory</value>
    </parameter>
    <parameter>
      <name>driverClassName</name>
      <value>org.postgresql.Driver</value>
    </parameter>
    <parameter>
      <name>removeAbandoned</name>
      <value>true</value>
    </parameter>
  </ResourceParams>
</Context>

```

```

</parameter>
<parameter>
  <name>url</name>
  <value>jdbc:postgresql://turtle:1111/habc</value>
</parameter>
<parameter>
  <name>username</name>
  <value>habc_user</value>
</parameter>
<parameter>
  <name>password</name>
  <value></value>
</parameter>
<parameter>
  <name>maxActive</name>
  <value>2</value>
</parameter>
<parameter>
  <name>maxIdle</name>
  <value>2</value>
</parameter>
<parameter>
  <name>maxWait</name>
  <value>10000</value>
</parameter>
</ResourceParams>
</Context>

```

5.5.3 log4j.properties

In the development environment setup the `log4j.properties` file needs to be in the `/tomcat/ROOT/WEB-INF/classes` folder as well as `/tomcat/webapps/ROOT/WEB-INF/classes`.

5.5.4 Eclipse Build Paths

The eclipse build path can be configured using the maven command `mvn eclipse:eclipse`. Additional items need to be added to the classpath in the Eclipse environment to ensure the correct behaviour in the GWT environment. These libraries, listed below, need to be added to the *User Entries* in the *Classpath Tab* in the *Run Dialog*:

- commons-dbcp-1.2.2.jar
- commons-pool-1.3.jar
- postgresql-8.2-504.jdbc3.jar
- gwt-dev-windows.jar
- openlayers-gwt\src\main folder (from the GWT OpenLayers library)

5.6 TRAC SETUP

Trac tickets are used in Hectares BC for tracking bugs, issues, decisions, and risks.

5.6.1 Ticket Setup

The trac ticket options have been set up as follows:

Ticket Types

- Bug – For tracking bugs found in with the system during testing.
- New Feature – For tracking new features requested by stakeholders.
- Decision – For tracking decision make regarding any development or data issues.
- Risk – For tracking project risks and mitigation strategies.

Priorities:

- blocker – Development cannot continue until this issue is resolved.
- critical – The project will not meet requirements, expectations, or usability requirements if this issues is not resolved.
- major – The project will not meet all requirements if this issue is not resolved.
- minor – Certain functionalities may be difficult to use or unavailable if these issues are not resolved.
- trivial – These are issues that are “nice-to-haves” but no required to meet the project goals.

Components:

- GUI – for tickets relating to GUI functionality.
- BackEnd – for tickets relating to server side functionality
- Data – for tickets relating to the collection and processing of data and metadata
- Docs – for tickets relating to technical or user documentation
- Project – used for project level tickets such as risks and decisions

5.6.2 Wiki

Trac comes with a wiki system as well that will be used as the project wiki. This wiki will contain development information, meeting minutes, and any project related documentation.

6 SERVER SPECIFICATIONS

The Hectares BC Server is comprised of the following components.

- Quad Core Xeon Processor E53102x4MB Cache, 1.60GHz, 1066MHz FSB, PE2900
- Redundant Power Supply with Dual Cords for PowerEdge 2900
- 24GB 667MHz (12x2GB), Dual Ranked DIMMs
- 16X DVD-ROM for PowerEdge 2900
- Integrated SAS/SATA RAID 5/1: Fault Tolerant Bkplane/Mirrored Flex Bay
- PERC 5/i, Integrated Controller Card
- 1x2 Flex Bay for PowerEdge 2900
- 1.44MB Floppy Drive, Internal Black
- Two 36GB 15K RPM Serial-Attach SCSI 3Gbps 3.5-in FlexBay HardDrive
- Six 300GB 15K RPM Serial-Attach SCSI 3Gbps 3.5-in HotPlug Hard Drives
- Embedded Broadcom NetXtreme II5708 GigabitEthernet NIC
- Broadcom TCP/IP Offload EngineNot Enabled

A RAID 1 redundancy system will be applied to the operation system drives; a RAID 5 redundancy system will be applied to the data drives.