Hectares BC

Administator Guide

Prepared For

Matt Austin
Nature Trust

Prepared By

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

Document Tracking

Version: 1
Last Updated: May 22, 2008

REVISION HISTORY

Date	Version	Author	Description of Changes
Sept 5, 2007	1	Emily Gouge Amr Alam (Refractions Research)	First draft

Table of Contents

1 INTRODUCTION

1.1 PURPOSE

The purpose of this document is to provide an overview of the Administrator Interface of the Hectares BC system. It also provides a guide for administrative tasks that need to be occasionally performed for optimal system performance.

1.2 TARGET AUDIENCE

The intended audience for the document is the Hectares BC working group members and system administrators.

1.3 REFERENCES

The following resources were used as references in the creation of this system documentation. All of these documents can be found on the Hectares BC wiki (www.hectaresbc.org).

- Hectares BC Wiki and Application

2 LOADING DATA

2.1 USING THE DATA LOADERS

There are three different loaders available:

- The Master Grid tables loader
- The Metadata loader
- The Map File generator

All three are contained in the same JAR file and will be run in a similar manner. Each can be run using the full Java command, or using the executable files provided for simplicity.

2.1.1 Properties Files

All three loaders make use of the same two properties files. These file are located in the **`load_lib`** folder and should be looked over before running the loaders.

2.1.1.1 Logging Properties File

The logging properties file must be called **`log4j.properties`**, and must reside on the classpath of the application. The log4j file can be configured using the standard log4j configuration; a sample file is shown below.

```
log4j.rootLogger=TRACE, A1
log4j.appender.A1=org.apache.log4j.ConsoleAppender
log4j.appender.A1.layout=org.apache.log4j.PatternLayout

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

2.1.1.2 HaBC Properties File

The Hectares BC properties file contains defaults required for data loading. This includes information such as the database name, the user name, and file locations. This file must reside on the classpath of the application.

The properties file contains the following information:

- file_datastore – The directory containing all the Hectares BC file application information (ex. **`/raid/habc_data_prod/`**).

- data_files_location – The location of the Hectares BC file datastore (ex. **/raid/habc_data_prod/data_files**).
- db_url – the database connection string
- db_username – The user name to use in connecting to the database.
- db_password – Password associated with the given user name.
- hkey_metadata_file_name – The name of a SQL script used in data loading.

Sample properties file:

```
#This properties file contains various settings for the Hectares BC system

#Location of File Datastore
file_datastore = /raid/habc_data_prod/
data_files_location = /raid/habc_data_prod/data_files/

#Database parameters
db_url = jdbc:postgresql://hectare:5432/habc_prod
db_username = habc_user
db_password =

#must be on classpath; only put name here
hkey_metadata_file_name = update_hkey_metadata.sql
```

2.2 LOADING MASTER GRIDS

Loading the master grid tables is fairly simple. The required TIFF files must exist in the appropriate locations and should conform to the required specifications; please refer to the HaBC Technical Documentation for the current specifications.

The master grid loader does not take any arguments; it does however require that the two properties files mentioned earlier be present.

The loader can be run using the following:

```
java -Xmx265M -cp "./load_lib/habc-
dataloader-3.9.jar:./load_lib/log4j-1.2.15.jar:./load_lib/postgresql-
jdbc-8.2-505-copy-20070719.jdbc3.jar:./load_lib"
net.refractions.habc.dataload.MasterGridsLoader
```

This needs to be run from the following location:

```
/raid/habc_data_prod/admin_tools
```

An alternative is to use the provided executable file located in that same folder and can be run as shown below. This executable file will also run the query validation immediately after the load.

```
./loadMasterGrids.txt
```

The loader simply outputs the names of the raster files that were found, and then a countdown of all the map rows that have been processed.

Note: Loading the master grids is a **destructive action**; this means that existing master grids will be dropped.

2.2.1 Validating Queries

Validating queries is required after loading the master grids. Since the master grids are dropped and re-created, there are no guarantees that data layers and their corresponding hkeys used in queries still exist. Please refer to the HaBC Technical Documentation for more information.

The validator can be run using the following:

```
java -Xmx265M -cp "./load_lib/habc-
dataloader-3.9.jar:./load_lib/log4j-1.2.15.jar:./load_lib/postgresql-
jdbc-8.2-505-copy-20070719.jdbc3.jar:./load_lib"
net.refractions.habc.dataload.QueryValidator
```

This needs to be run from the following location:

```
/raid/habc_data_prod/admin_tools
```

The validator will output a list of the invalid queries and the first failure encountered for each.

Queries that are deemed invalid are flagged as invalid and are not displayed in the *Show Me Where* or *How Much* tabs. They remain viewable in the *My HaBC* tab, but are displayed as invalid.

2.3 LOADING METADATA

The metadata loader loads the metadata files into the database. This should be run after the files are modified in any way.

Loading the metadata drops the existing database tables and recreates/repopulates them. This is a destructive action which will clear out any existing metadata in the database.

The loader will stop on any error. This will result in incomplete data if errors are met; therefore errors should be corrected and the loader re-run when they are encountered. Generally, errors are typos in the metadata files or directory structure. The loader should provide some reference to the type and location of the error.

The metadata loader does not take any arguments; it does however require that the two properties files mentioned earlier be present.

The loader can be run using the following:

```
java -Xmx265M -cp "./load_lib/habc-
dataloader-3.9.jar:./load_lib/log4j-1.2.15.jar:./load_lib/postgresql-
jdbc-8.2-505-copy-20070719.jdbc3.jar:./load_lib"
net.refractions.habc.dataload.MetadataLoader
```

This needs to be run from the following location:

```
/raid/habc_data_prod/admin_tools
```

An alternative is to use the provided executable file located in that same folder and can be run as follows:

```
./loadMetadata.txt
```

2.4 MAPFILE GENERATOR

The Mapfile Generator reads metadata information in the file data store and generates a mapfile. This mapfile is used by Mapserver to provide WMS access to the data. This supports the *Raster Data* tab and the WMS/WCS access to the data in the Hectares BC system.

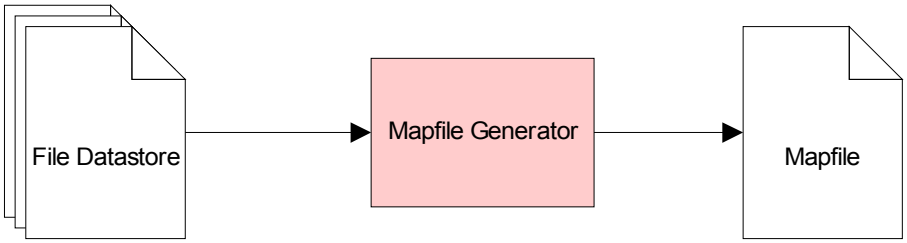

Figure 1: Mapfile Generator

2.4.1 Usage

2.4.1.1 Arguments

The Mapfile Generator takes two arguments: the location of the template mapfile, and the location to output the mapfile.

```
MapfileGenerator -mapfile <mapfile loc> -template <habc mapfile template>
```

The template mapfile is used to specify all elements about the mapfile, except the layer information, which is populated by the mapfile generator from the file datastore. The template should be a valid mapfile with a line containing the text "<layers>" where the layer information should be placed. A sample template file can be generated by the mapfile generator by running with only a template argument (as shown below). This will cause the software to generate a mapfile template to the specified location.

```
MapfileGenerator -template <habc mapfile template>
```

A sample template file:

```
MAP
  EXTENT 159600 173500 1881200 1747900
  SIZE 256 256
  UNITS meters
  IMAGECOLOR 0 0 0
  FONTSET /raid/habc_data_devel/styles/fonts/fonts.lst
```

```
  WEB
    METADATA
      "wms_title" "HaBC Raw Data Access"
    END #end metadata
  END #end web

  PROJECTION
    "init=epsg:3005"
  END #end projection

  OUTPUTFORMAT
    NAME gif
    DRIVER GD/GIF
    MIMETYPE image/gif
    EXTENSION gif
    IMAGEMODE PC256
    TRANSPARENT ON
  END #output format

  OUTPUTFORMAT
    NAME GTiff
    DRIVER GDAL/GTIFF
    MIMETYPE image/gtiff
    EXTENSION gtiff
    IMAGEMODE BYTE
    TRANSPARENT OFF
  END #output format

  LEGEND
    STATUS ON
    IMAGECOLOR 255 255 255
    LABEL
      TYPE truetype
      FONT verdana
      SIZE 8
      COLOR 0 0 0
      OFFSET 0 -2
    END #label
  END #legend

<layers>

END #end mapfile
```

2.4.1.2 Output

If completed successfully the map file will be written and the user will see the following message:

```
Map file generation complete: <mapfile>
```

Otherwise the user will receive an error along with the following message:

```
Map file generation failed.
```

The mapfile generator traverses the file datastore identifying the grids that are to be added to the mapfile. This includes verifying the necessary metadata files exist. If any of the required files are missing a warning message will be displayed warning the user that this layer will not be included in the mapfile generation. The mapfile generation will continue for the other layers.

2.4.2 Styling

The styling for each layer is determined by the information in its metadata. Each grid requires a **<gridname>.wms.xml** file in the **<layer>/grids/<gridname>** directory which contains information about each value to be styled.

This xml file has the following format:

```
<gridname>
  <entry>
    <legend_entry>Alsek Ranges</legend_entry>
    <values>1</values>
    <color>
      <red>28</red>
      <green>164</green>
      <blue>6</blue>
    </color>
  </entry>
  ...
</gridname>
```

There can be multiple entry elements; one element for each legend entry. The legend_entry contains the text displayed in the legend. Each legend_entry also has associated with it, a values tag which describes the range of values for the entry and a color tag which describes the color for the range of values.

The values can contain a single number, a comma-delimited list of numbers, or a range using a hyphen. For example for a single number: **<values>1</values>**; for a list of numbers: **<values>1,3,5,7</values>**; for a range **<values>1-100</values>**. A combination of ranges and values can also be specified **<values>1-100,200,300-500</values>**.

If this file does not exist for any grid the mapfile generation will fail. At a minimum this file must exist with the **<gridname></gridname>** elements.

2.5 LOADING BC MAP SHEETS

The Zoom To Location functionality requires a database table containing map sheet names and bounding boxes for BC. The bc_mapsheets table has the following fields:

Field	Description
map_name	The name of the map sheet (Example: 92F.024)
the_geom	The geometry column containing the map sheet polygon. This is an optional column.
bounding_box	The geometry column containing the bounding box of the map sheet polygon.

Table 1: BC Map Sheets Table

A dump of the table is located in

```
/raid/habc_data_prod/admin_tools/load_lib/bc_mapsheets.sql
```

Running the SQL commands in this file will create the table and populate it with the proper content.

This data only needs to be loaded if there is a change to the definition of the mapsheets or for some reason the table is missing from the production database.

3 METADATA DATA MAINTENANCE

Metadata in the Hectares BC system is stored as HTML files in the file datastore. This information is then used to populate the database when the metadata loader is run.

Metadata can be updated by either:

- modifying the HTML metadata file and rerunning the metadata loader; or
- using the metadata editor.

The metadata editor is a web based interface that allows users to edit contents of the HTML files. Once the changes are saved the metadata editor automatically updates the database and the file datastore with the modifications made. This is done to ensure that the metadata in the database and file datastore remain synchronized so future runs of the metadata load don't overwrite modifications.

The metadata editor is protected so that only authorized users may make changes. Authorized users are those with login access to the Hectares BC Wiki.

To access the metadata window there are two options:

1. The metadata window in the *Show Me Where, How Much* and *Raster Data* tabs includes an 'Edit Metadata' link at the bottom of each metadata file. Selecting this link will open up the metadata editor with the given metadata files selected.
2. Using the following link. However, using this link is not recommended as it loads the tree structure for all metadata files in the system and is very slow. http://www.hectaresbc.org/dbforms/metadata_editor_prod.php

Option 1, navigating to the editor from a specific metadata window within the application, will bring up a simplified version which only displays a limited list of tree items (the item selected and all siblings are displayed). This makes modifying the 'Tree Order' field simpler.

3.1 METADATA USER AUTHENTICATION

The metadata editor is password protected to allow only authorized Hectares BC users to access the metadata link. Standard HTTP Authentication is used to verify passwords.

All users names and passwords are stored in the **`/raid/svn/svn.passwd`** file.

Usernames and passwords can be managed using the **`htpasswd`** command.

For further information on the **`htpasswd`** utility program see the man pages. Please note that granting access to the metadata editor also grants access to the Wiki (with no real permissions).

4 WIKI AUTHENTICATION

The Hectares BC Wiki has been setup so that it is readable by all users, but only editable by authenticated users.

To add a user to the system they must first be added to the HTTP Authentication (See Section 3.1 - Metadata User Authentication). Then the user can be added to the trac system. Use the **trac-admin** tool to give the user developer or workinggroup permissions. The trac wiki is located at **/raid/trac/habc**. To list the current users and permissions use:

```
trac-admin /raid/trac/habc permission list
```

To give a new user workinggroup permissions:

```
trac-admin /raid/trac/habc permission add <USERNAME> workinggroup
```

5 BATCH PROCESSOR

The batch processor processes requests users have sent to the queue. These requests are queries which take significant processing time to complete and thus are not available to users from the user interface.

When the batch processor runs it grabs the next job from the batch processing queue, executes the job, packages the results, places them in a web-accessible folder, and emails the recipients a notification of where they can download their results. No results are sent as email attachments. This process is repeated until all queued jobs are completed.

5.1 HABC USER

The batch processor is run under the **habc/admin321** user on the hectares box. This user requires a home directory, access to read and write the output and input directories specified in the **batchprocessor.properties** file, and permissions to run cron jobs.

Currently the batch processor cron job is set up as follows (no emails are sent):

```
0 */1 * * *    /raid/habc_data_prod/admin_tools/runCronBatchProcessor.sh >> /
raid/habc_data_prod/admin_tools/log/batchprocessor.log 2>&1
```

5.2 BATCH PROCESSOR INPUTS

In addition to the required Java jar files, the batch processor requires three properties files:

- batchprocessor.properties – contains database connection and file locations
- mailer.properties – contains emailer information
- log4j.properties – contains logging information

Samples of each of these properties files can be found in the **/raid/habc_data_prod/admin_tools/batchprocessor_lib** folder.

5.2.1 batchprocessor.properties

The batchprocessor.properties file contains various settings required for the batch processor to run. These include:

- job_processing_dir – temporary processing directory to gather results of each job before zipping them up and placing them in a web accessible location
- metadata_gen_url – the URL of the metadata generator for land characterization queries

- filedatastore_dir – the main directory of the file datastore
- output_dir – the web accessible location to put packaged job results
- gdal_translate – the path to the Gdal software
- db_host – database host
- db_port – database port
- db_database – database name
- db_username – database username to connect with
- db_password – database password
- batch_expire_time – completed batch job expiration time in hours

5.2.2 mailer.properties

The mailer.properties file contains information about the e-mail server and the web address of the results location. The information in this file is used to send users notifications that their jobs have been completed and a link to where they can download the results.

```
#Mail server
mail_server = hectare
mail_server_port = 25
mail_username =
mail_password =

#From email
from_email = noreply@hectaresbc.org

#From email name
from_email_name = HectaresBC Exporter

#Admin email
admin_email = admin@hectaresbc.org

#Email templates
email_template_file = results_email_template.txt
error_email_template_file = error_email_template.txt

#Output Web Address
output_address = http://www.hectaresbc.org/user_exports_prod/
```

5.2.3 log4j.properties

The following properties file sets up log4j to write **ERROR** (or higher) level messages to the **/raid/habc_data_prod/admin_tools/log/batchprocessor.java.log** file.

```
log4j.rootLogger=ERROR, A1
```

```
log4j.appender.A1=org.apache.log4j.FileAppender
log4j.appender.A1.layout=org.apache.log4j.PatternLayout
log4j.appender.A1.File=/raid/habc_data_prod/admin_tools/log/batchprocessor.ja
va.log

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

Further information on how to set up a log4j.properties file can be found at http://logging.apache.org/log4j/1.2/manual.html.

5.3 RUNNING THE BATCH PROCESSOR

The batch queue processor is implemented as a Java program that can be run manually or be set up to run at defined times by a **cron** scheduler.

5.3.1 Manually

The batch processor can be run manually by running the script:
/raid/habc_data_prod/admin_tools/runBatchProcessor.sh

5.3.2 Cron Job

A cron job is set up to run the **runCronBatchProcessor.sh** script and logging information to the **/raid/habc_data_prod/admin_tools/log/batchprocessor.log** file. This script differs from the **runBatchProcessor.sh** script in that it uses a lock file to ensure another version of the batch processor is not already running. Although multiple instances of the batch processor can be executed concurrently, it is not recommended for performance purposes.

Errors will be written to one of two logs:

- The Batch Processor log file (specified in the **batchprocessor_lib/log4j.properties** file). These errors are generated by the Java batch processor program.
- The cron job log file (specified in the crontab). These errors are generated by the cron job.

In addition, if a job fails the job will be classified with a status of Error in the batch queue and the error_message should provide a brief description of why it failed.

6 BATCH QUEUE CLEANER

The batch queue cleaner program removes expired jobs and all associated files from the batch queue. Batch queue jobs expire after a specified amount of time after the last accessed date, as defined in the **`batchprocessor.properties`** file.

6.1 RUNNING THE BATCH QUEUE CLEANER

The batch queue cleaner is implemented as a Java program that is run automatically as part of the batch processor. It can also be run manually.

6.1.1 Manually

The batch queue cleaner can be run manually by running the script:
`/raid/habc_data_prod/admin_tools/runBatchProcessor.sh`

7 BATCH PROCESSOR CHECKER

The batch processor checker program verifies the items in the batch queue are being processed and the batch queue processor has not crashed or become hung up on a single job. This tool has been added to the cron to run at regular intervals.

This software checks the following:

- No item has been waiting for more than **<max_time_in_queue>**.
- No item has been processing for more than **<max_processing_time>**.
- The total number of queued items does not exceed **<max_queue_size>**.
- The total number of running items does not exceed **<max_running_tasks>**.

The parameters **max_time_in_queue**, **max_processing_time**, **max_queue_size**, **max_running_tasks** are all specified in the **maintenance.properties** file.

If any of these checks generate errors then an email is sent to the system administrator (the **to_email** field in the **maintenance.properties** file).

7.1 PROPERTIES FILES

7.1.1 maintenance.properties

The batch processor checker uses the same **maintenance.properties** file as the statistics generator. As shown below, this file contains database connection information, batch queue limits, and email information necessary for the checker to run.

```
#Hectares BC Database Connection information
db_host = hectare
db_port = 5432
db_database = habc_prod
db_username = habc_batchprocessor
db_password =

#Batch Queue Time Limits  (times are specified in hours)
max_time_in_queue = 36
max_processing_time = 4
max_queue_size = 10
max_running_tasks = 2

# -- Email --
mail_server = hectare
mail_server_port = 25
mail_username =
mail_password =
```

```
from_email = noreply@hectaresbc.org
from_email_name = HectaresBC Batch Queue
email_template_file = batch_checker_email_template.txt
#multiple email addresses can be separated by ;
to_email=admin@hectaresbc.org
```

7.1.2 log4j.properties

In addition, the batch processor uses a log4j.properties file to specify what level of logging to use and where the log file should be written.

```
log4j.rootLogger=ERROR, A1
log4j.appender.A1=org.apache.log4j.FileAppender
log4j.appender.A1.layout=org.apache.log4j.PatternLayout
log4j.appender.A1.File=/raid/habc_data_prod/admin_tools/log/maintenance.java.
log

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

7.2 RUNNING THE BATCH PROCESSOR CHECKER

The batch processor checker is implemented as a Java program that can be run manually or through the **cron** scheduler.

7.2.1 Manually

The batch processor can be run manually by running the script:
/raid/habc_data_prod/admin_tools/runBatchQueueChecker.sh

7.2.2 Cron Job

A cron job is set up to run the **runBatchQueueChecker.sh** script daily (around midnight). The cron job is setup under the **habc** user (see Section 5.1 - Habc User).

Errors will be written to one of two logs:

- The maintenance log file (specified in the **maintenance_lib/log4j.properties** file). These errors are generated by the Java batch processor checker program.
- The cron job log file (specified in the crontab). These errors are generated by the cron job.

7.3 RESOLVING ISSUES

If an item has been waiting in the batch queue for more than the specified time or a particular item has been processing for more than the specified time, this could be caused by:

- a process that failed and did not release the lock file, used by the batch processor to ensure only a single copy of the batch processor is running
- a single large batch queue item that is taking a significant amount of time to compute
- many processing time expensive batch queue items

To resolve:

- Ensure that the batch processor is still running. Review the cron tab log file and ensure it contains no errors; as well, review the **`batchprocessor.java.log`** file for errors.
- If the batch processor is not running and the lock file **`batchqueue.lock`** exists, then the batch processor has failed without releasing the lock file. Delete this file and the batch queue processor should start running again.
- Review the items in the batch queue. If one item has taken a long time then you may need to stop the batch process and remove the lock (**`batchqueue.lock`**) file.

If the total number of queue items exceeds the specified maximum:

> This is a warning that there are many items in the batch queue. These items can be reviewed using the batch queue tab on the administrator interface and removed if necessary.

If the total number of running items exceeds the specified maximum:

> Currently, the system is setup so only a single batch processor task is running at once. As a result there should only ever be one running item in the batch queue. If there are more than one running item, then the batch processor has failed on one of the items. The log files should provide some insight into why this item failed. To remove the failed item, use the administrator interface and delete the batch queue item that has failed.

8 STATISTIC GENERATOR

Statistics about the Hectares BC system are stored in separate database tables. These statistics tables are updated regularly by the statistic generator. The statistic generator is a Java program that determines what items have been added, modified or used since the previous execution of the statistic generator and updates all statistics tables to reflect these changes. It has been set up as a cron process to run at fixed intervals.

8.1 STATISTIC GENERATOR INPUTS

In addition to the required Java jar files, the statistic generator requires two properties files:

- maintenance.properties – contains database connection information
- log4j.properties – contains logging information

Below is a sample maintenance.properties file:

```
#This properties file contains various settings for the maintenance module of
Hectares BC system

#Hectares BC Database Connection information
db_host = hectare
db_port = 5432
db_database = habc_devel
db_username = habc_user
db_password =
```

Below is a sample log4j.properties file

```
log4j.rootLogger=ERROR, A1
log4j.appender.A1=org.apache.log4j.FileAppender
log4j.appender.A1.layout=org.apache.log4j.PatternLayout
log4j.appender.A1.File=/raid/habc_data_prod/admin_tools/log/maintenance.java.
log

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

8.2 RUNNING THE STATISTIC GENERATOR

8.2.1 Cron Process

Similar to the batch processor, the statistic generator cron job is run under the **habc** user account. Currently the statistic generator cron job is set up to run daily (no emails are sent):

```
#Statistics Gatherer
0 0 * * * /raid/habc_data_prod/admin_tools/runMaintenanceProcessor.sh >>
/raid/habc_data_prod/admin_tools/log/maintenance.log 2>&1
```

Cron logs are written to the
/raid/habc_data_prod/admin_tools/log/maintenance.log file.

The Java program writes logs to the file specified in the log4j.properties file.

8.2.2 Manually

The statistic generator can be run manually by running the following command:

```
/raid/habc_data_prod/admin_tools/runMaintenanceProcessor.sh
```

9 ADMINISTRATOR INTERFACE

An administrator interface was developed to aid with the maintenance of the Hectares BC system. This application was designed with a similar interface as the main Hectares BC application

The Hectares BC administrator interface can be accessed from:

http://www.hectaresbc.org:22080/app/habc-sprint9/HaBCAdmin.html

9.1 LOGIN

Upon accessing the administrator interface you will be prompted to login. The login includes both a user name and password.

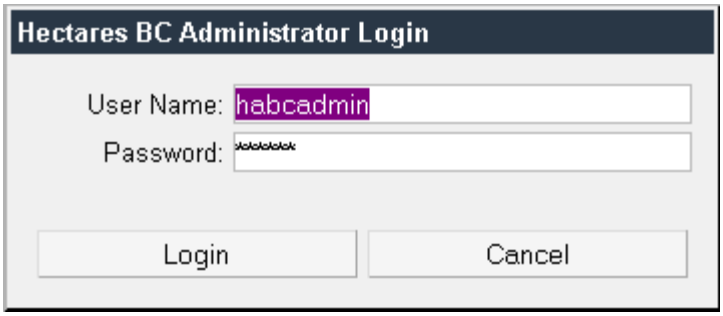

Figure 2: Hectares BC Administrator Login

The current administrator login user name/password is **habcadmin/beaver**. The password can be changed from the Login Accounts Tab once logged in.

9.2 TABS

The administrator interface has the following tabs:

Welcome	The Hectares BC Welcome tab.
HaBC Statistics	A tab containing various statistics about the Hectares BC system including which layers are most popular, which search terms are most popular, various batch queue statistics, and a link to the Google Analytics site and Webalizer page for further statistics.
Login Accounts	A tab to manage user accounts and change administrator passwords.
Batch Queue	A tab to view and manage the batch queue.
Public Query List	A tab to view and edit public 'Show Me Where' and 'How Much' queries.
Show Me Where ...	A tab to create and edit 'Show Me Where' queries.
How Much ...	A tab to create and edit 'How Much' queries.
Maintenance	A tab to perform maintenance on the Hectares BC system including clearing cached tiles and query tables.

Table 2: Administrator Interface Tab List

9.3 HABC STATISTICS TAB

This tab contains the following statistics:

Layer Statistics	Statistics related to the usage of data layers.
Batch Queue Statistics	Statistics related to the batch queue.
Search Statistics	Statistics related to the search function.
Google Analytics	Link to the Google Analytics site.
Webalizer	Link to the Webalizer page for the Hectares BC system.

To open a particular statistics section click on the header of that section. If opened, the section will be closed when the header is clicked on.

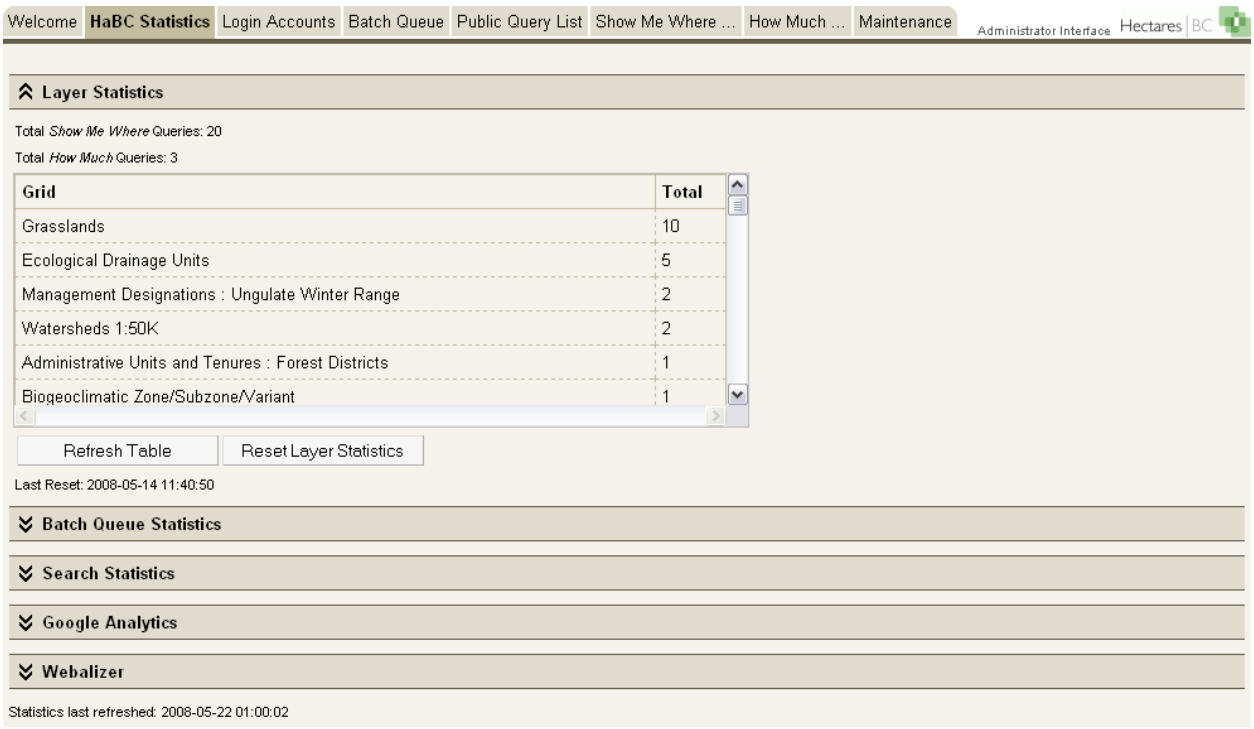

Figure 3: Statistics Tab

9.3.1 How Statistics Are Computed

Statistics are stored in separate database tables. These statistics tables are updated regularly by the statistic generator cron process. This process determines what items have been added, modified or used since the previous execution of the statistic generator and updates all statistics tables to reflect these changes. As a result statistics are updated on a fixed schedule, not each time an action is performed against the Hectares BC system. For further information on the statistic generator cron process, see Section 8 - Statistic Generator.

The last time statistics were refreshed is shown at the bottom of the HaBC Statistics tab.

By "Resetting" any of the statistics the values in the statistics tables are reset to 0. The date the statistics were last reset at appears below the "Reset" button.

9.3.2 Layer Statistics

The layer statistics include:

- The total number of *Show Me Where* and *How Much* queries asked of the system.
- The number of times each Grid has been used in these queries. If the same grid is used multiple times in a single query it is only counted once.

There are two buttons related to the layer statistics:

- Refresh Table – Reloads the layer statistics from the database. A change in the statistics will only be seen if the statistic generator has been run since the table was last loaded.
- Reset Layer Statistics – Resets all the current layer statistics to 0.

9.3.3 Batch Queue Statistics

The batch queue statistics include various items related to the batch queue. These include:

- Total Items Submitted
- Total '*Show Me Where*' Queries Submitted
- Total '*How Much*' Queries Submitted
- Total Raster Data Requests Submitted
- Average Processing Time (hours) – for all types of queries
- Average '*Show Me Where*' Processing Time
- Average '*How Much*' Processing Time
- Average Raster Data Processing Time
- Average Time In Queue – time between the submitted date and the start of processing
- Average Items Submitted Per Day

There are two buttons related to the batch queue statistics:

- Refresh Table – Reloads the batch queue statistics from the database. A change in the statistics will only be seen if the statistic generator has been run since the table was last loaded.
- Reset Batch Statistics – Resets all batch queue statistics to 0.

9.3.4 Search Statistics

The search statistics include:

- total number of searches performed
- a list of search terms used and the number of times they were used

There are two buttons related to the search statistics:

- Refresh Table – Reloads the search statistics from the database. A change in the statistics will only be seen if the statistic generator has been run since the table was last loaded.
- Reset Batch Statistics – Resets the search count to 0 and removes all search terms from the statistics tables.

9.3.5 Google Analytics

Google Analytics is a free service offered by Google that generates detailed statistics about visitors to a web site. This includes information about the number of visits to the web site, the types of browsers being used, and other client related information.

This section contains a link to the Google Analytics page and the login information for the Hectares BC account.

9.3.6 Webalizer

Webalizer is an application that parses Apache web logs and provides detailed statistics about file usage. This is useful for tracking network bandwidth usage and optimization.

This section contains a link to the Webalizer page. Access to the Webalizer page requires Trac authentication.

9.4 LOGIN ACCOUNTS TAB

This tab contains a list of all user login accounts created on the system, including the associated name, organization, created date and last logged in date. The accounts are sorted alphabetically.

To delete user accounts, click on the check mark next to the accounts to delete and click the 'Delete' button. When an account is deleted all *Show Me Where*, *How Much*, and Batch Queue items the user has generated will also be deleted.

To refresh the account list, click the 'Refresh Table' button.

In addition to user logins, this tab also includes the administrator accounts. To change the password associated with an administrator account, click on the 'Change Password' function. A dialog box where you can enter the new password will be displayed.

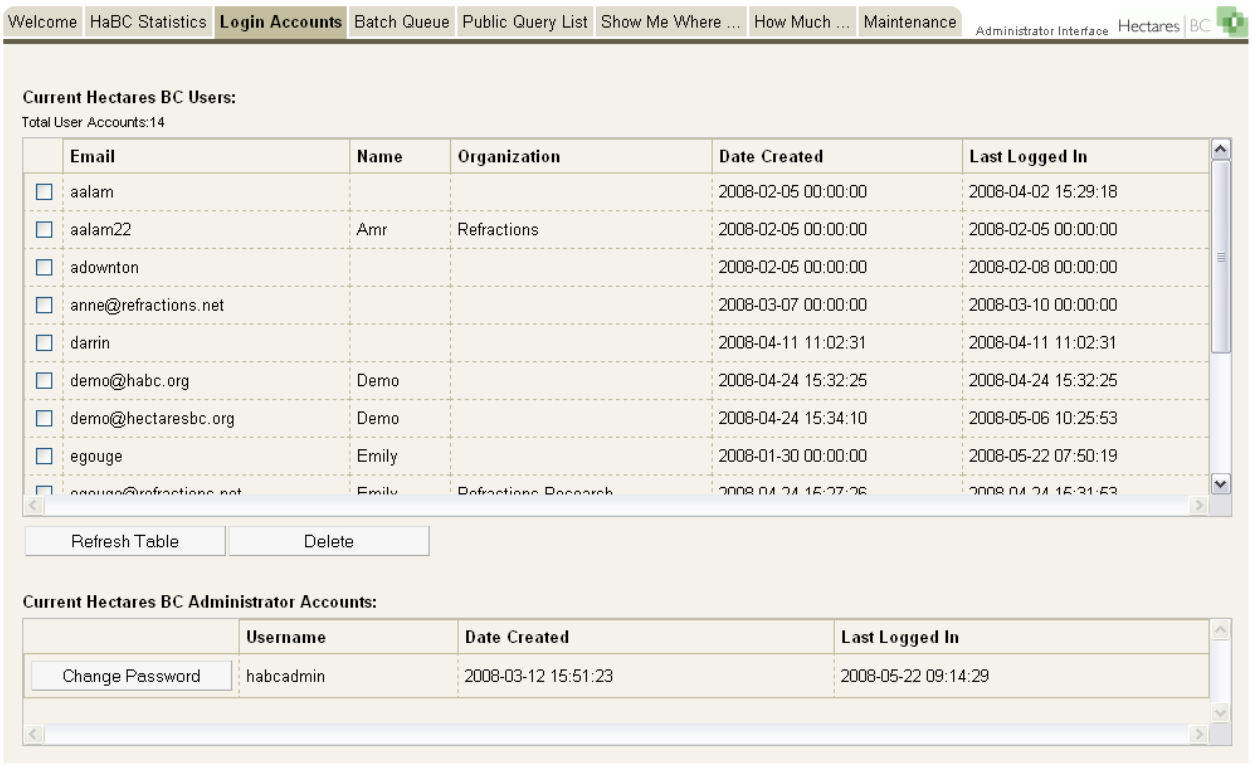

Figure 4: Login Accounts Tab

9.5 BATCH QUEUE TAB

The batch queue tab allows an administrator to view and remove items from the batch queue. This tab contains a summary about the current state of the batch queue (the number of items queues, running, error, done, deleted) and a list of all batch items.

When a user deletes a batch queue item it is flagged as 'Deleted', however it is not removed from the batch queue table until deleted by the administrator or it expires and is removed by the batch queue cleaner. For more information on the batch queue cleaner, see Section 6 - Batch Queue Cleaner.

To delete batch queue items click on the check boxes next to the items to delete and click the 'Delete Selected' button. The check boxes below the batch queue table can be used to easily select all, deleted, done, error, or queued items.

To refresh the batch queue table click the 'Refresh Table' button.

Although the administrator can delete 'Running' batch items if the batch processor is still working on completing the job the batch processor will not be terminated. The job will continue to complete, however it will not appear in the batch queue after completion. Generally 'Running' items should not be deleted unless they have failed and were not marked as 'Error'.

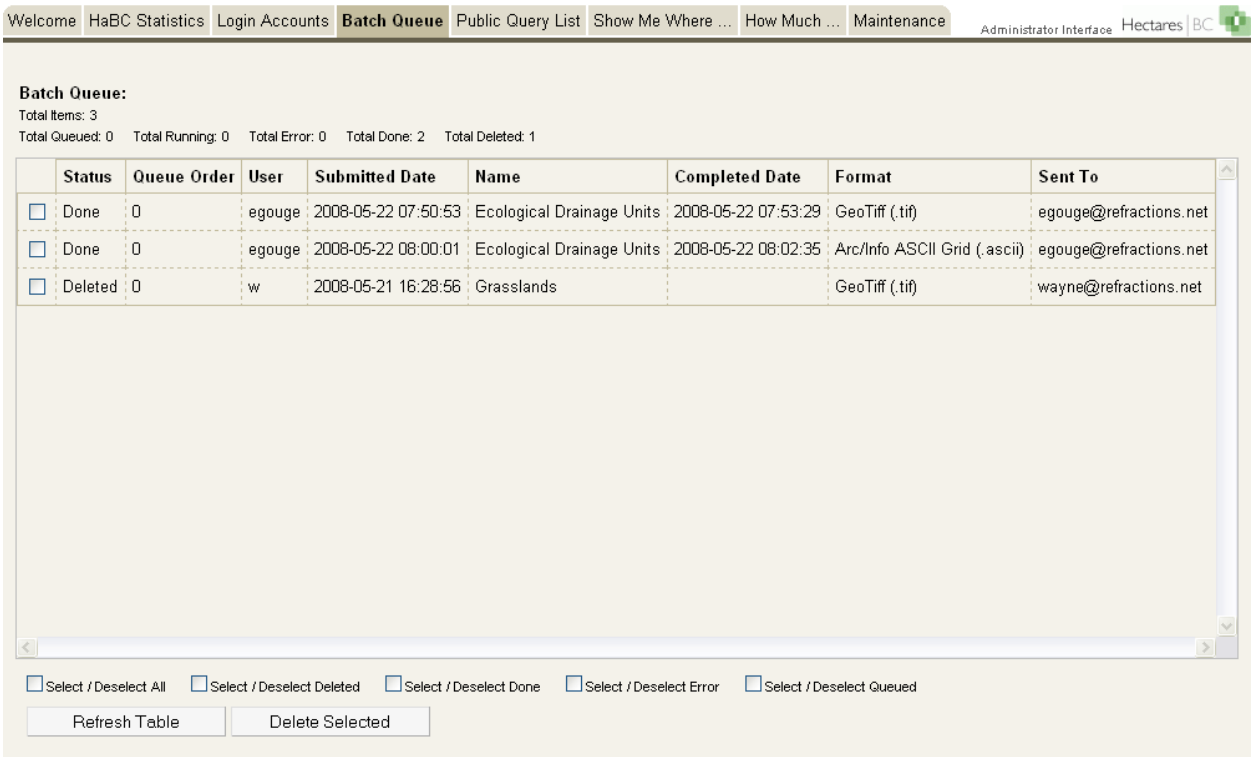

Figure 5: Batch Queue Tab

9.6 PUBLIC QUERY LIST TAB

This tab contains a list of all public *Show Me Where* and *How Much* queries. These queries can be removed or modified from this tab.

To delete queries click on the check boxes next to the queries to delete and press the 'Delete' button.

To refresh the list of queries click the 'Refresh Table' button.

To edit a query, click the 'Edit' button next to the query to edit. This will automatically take you to the corresponding *Show Me Where* or *How Much* tab with the query selected.

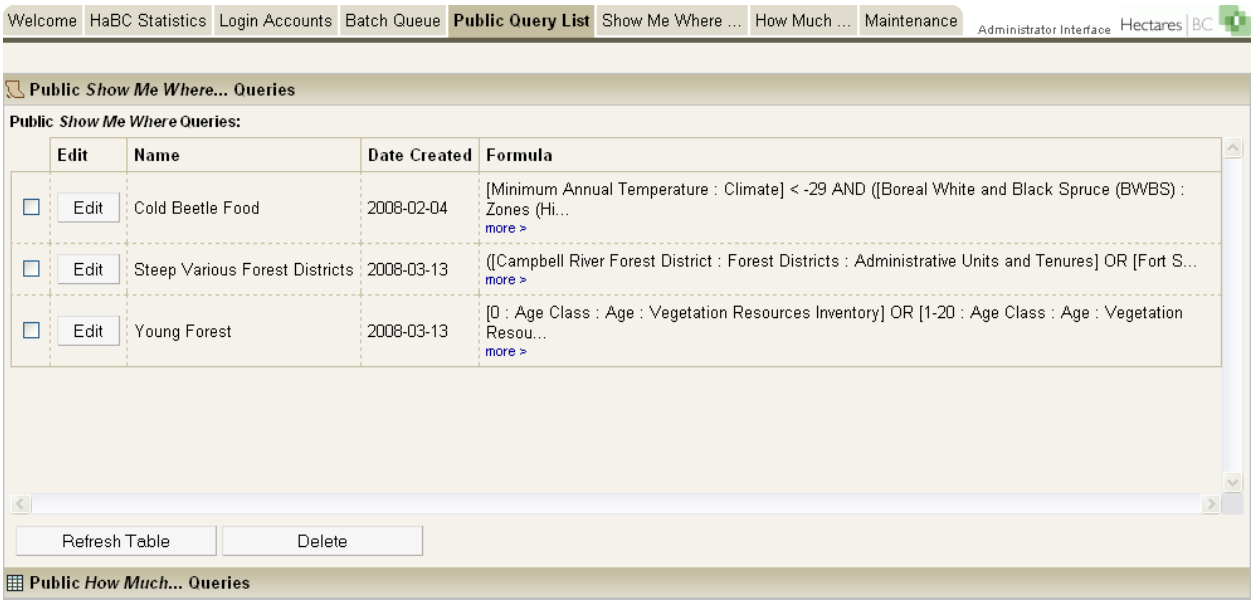

Figure 6: Public Query List Tab

9.7 *SHOW ME WHERE* TAB

This tab is the same as the *Show Me Where* tab in the Hectares BC user application with a few exceptions:

- Only public queries show up in the drop down list.
- In the layer tree under 'Saved "Show Me Where" Queries' both Public Queries and User Queries are shown, but if a new public query is added only the User Queries section will be updated with the new query (until the application is reloaded).
- The query list in the 'Public Query List' tab is not automatically refreshed. If you add a new query or update an existing query you will have to manually refresh the table (using the 'Refresh Table' button) to view the updated list.

9.8 *HOW MUCH* TAB

This tab is the same as the *How Much* tab in the Hectares BC user application with a few exceptions:

- Only public queries show up in the drop down list.
- In the layer tree under 'Saved "Show Me Where" Queries' both Public Queries and User Queries are shown, but if a new public *Show Me Where* query is added only the User Queries section will be updated with the new query (until the application is reloaded).

- The query list in the 'Public Query List' tab is not automatically refreshed. If you add a new query or update an existing query you will have to manually refresh the table (using the 'Refresh Table' button) to view the updated list.

9.9 MAINTENANCE TAB

This tab provides for general maintenance of the Hectares BC system.

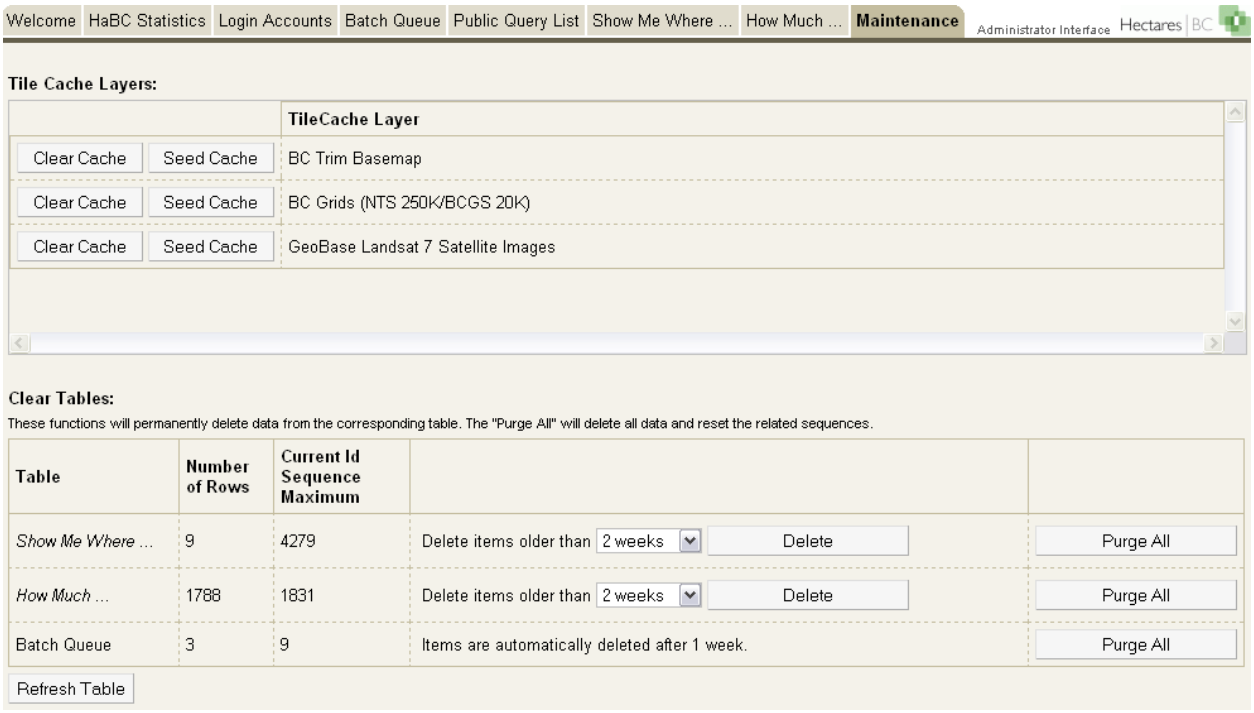

Figure 7: Maintenance Tab

9.9.1 Clearing Tile Cache

The base map layers displayed in the *Show Me Where* and *Raster Data* tabs are provided by a tile cache instance which is caching WMS requests. These tiles remain cached until explicitly removed by an administrator. To remove the cached tiles for a particular base map layer click on the 'Clear Cache' button next to the base map layer for which you wish to clear the cache.

Clearing the tile cache for a particular layer will make the Hectares BC mapping interface slower until the tiles have been re-cached.

9.9.2 Seeding Tile Cache

The tile caching software includes functionality to seed cache layers, minimizing the slowness after clearing the tile cache. To seed the cache for a particular base map layer, click on the 'Seed Cache' button next to the base map layer you wish to seed.

9.9.3 Clearing Database Table

All *Show Me Where* and *How Much* queries run in the Hectares BC system are given a unique identifier and stored in a table. This table is used for generating data layer and usage statistics. These unique ids are represented by an integer sequence. When these sequences exceed the value of an integer (2147483647) the table will require purging and the sequences require resetting. To clear the *Show Me Where* or *How Much* query tables click the 'Purge All' button. This will purge all records from the table and reset the corresponding sequence.

If there is a need to clear these tables for performance purposes, then it is also possible to delete all records older than a given time period. You can select a time period from the drop down list and use the 'Delete' button. Using this functionality will not reset the sequence values.

The number of rows and current maximum sequence value are also displayed for reference.

The batch queue can also be purged using the 'Clear Batch Queue' button. This will delete all items from the batch queue and reset all sequences.

Deleting batch queue items older than a given date is not available, as this is performed automatically on a regular basis. All batch queue items not accessed for more than 1 week old are removed automatically.

10 SYSTEM DISCLAIMER

10.1 MODIFYING

The system disclaimer text is stored in the **`SystemDisclaimer.html`** file (This file will be loaded in the same directory as the main application page – HaBC.html). This text can be modified by modifying the contents of this file. Note that this file is an html file and the html formatting needs to be maintained.

11 HECTARES BC APPLICATION HELP TAB

11.1 OVERVIEW

The Hectares BC application includes a Help tab that contains information about the Hectares BC application and related functions. This Help tab references an external web site that can be accessed directly from the following URL:

```
http://www.hectaresbc.org/app/habc-help/index2.htm
```

On the left side of this web site is a Contents pane which provides a table of contents for all the help topics. When a topic is selected the help associated with the topic is shown on the left side of the web site.

Each of the topics is stored as a separate web page. All pages related to the help system can be found here:

```
/opt/tomcat-6.0.14/webapps/habc-help/
```

11.2 UPDATING A HELP PAGE

The help system was developed using WinCHM Prov 3.37 software, however this software is not required to make modifications to the help pages. The following steps outline how to make a modification to an existing help page.

1. Locate the help page that needs to be updated.

   All help pages are contained within the **`scr/`** directory. Each page is named after the title that appears in the contents pane. For example, to edit the Raster Data Tab help page, the **`scr/Raster Data Tab.htm`** file will need to be edited. There may be a few cases where the file names differ slightly; however they should all be easily identifiable.

2. Edit and Save the help page.

   Modifications can be made in any text editor. These are html pages, therefore it is important to maintain valid html formatting.

3. At this stage the modifications should be complete and viewing the help page in a web browser should show the modifications. If the modifications are not visible it is likely because the browser has cached an old version of the file. A Ctrl-F5 refresh may be necessary to refresh the cache.

11.3 UPDATING SEARCH INDEXES

The help system uses Perlfect Search 3.37 to provide search functionality. When new help pages are added or existing pages are updated, the search indexes need to be updated. This can be done by running the following command:

```
/usr/bin/perl /raid/www/cgi-bin/perlfect/search/indexer.pl
```

An alternative is to use the provided executable file located in the **/raid/habc_data_prod/admin_tools** folder and can be run as shown below:

```
./runSearchHelpIndexer.txt
```

12 DATABASE MAINTENANCE

12.1 DATABASE USERS

The Hectares BC application requires the following database users:

- habc_user – this user is used by the web application
- habc_batchprocessor – this user is used by the batch processor