

# Hectares BC

**DRAFT**

## How To - Data Updates / Additions

### Prepared For

Matt Austin  
Nature Trust

### Prepared By

Refractions Research Inc.  
#300 – 1207 Douglas Street  
Victoria, BC  
[www.refractions.net](http://www.refractions.net)

### Document Tracking

Version: 1  
Last Updated: May 28, 2008

  
Refractions  
RESEARCH

THE GEOSPATIAL EXPERTS



## REVISION HISTORY

---

<b>Date</b>	<b>Version</b>	<b>Author</b>	<b>Description of Changes</b>
May 28, 2008	1	Emily Gouge	First draft

# CONTENTS

---

## Table of Contents

- 1 Introduction.....1
  - 1.1 Purpose.....1
  - 1.2 Target Audience.....1
  - 1.3 References.....1
- 2 Overview.....2
  - 2.1 Data Update / Additions Process.....2
  - 2.2 File Datastore Overview.....2
- 3 Adding new Layer.....5
  - 3.1 Adding New Data Layer Steps.....5
  - 3.2 Simply By Value Rasterization.....11
  - 3.3 Simple Category Rasterization.....13
- 4 Updating an Existing Data Layer.....17
  - 4.1 Updating New Data Layer Steps.....17

# 1 INTRODUCTION

## 1.1 PURPOSE

This document is intended to provide a Hectares BC administrator with information on how to update and add data layers.

## 1.2 TARGET AUDIENCE

The intended audience for the document is the Hectares BC administrator who has knowledge about the system design and implementation.

## 1.3 REFERENCES

The following resources were used as references in the creation of this document. All of these documents can be found on the Hectares BC Wiki ([www.hectaresbc.org](http://www.hectaresbc.org)).

- Hectares BC Technical Documentation
- Hectares BC Data Update Documentation
- Hectares BC Administrator Guide
- Hectares BC Wiki and Application

## 2 OVERVIEW

Data additions and updates to the Hectares BC system are a critical part of keeping the data current and the benefits of the system at a maximum.

### 2.1 DATA UPDATE / ADDITIONS PROCESS

At a minimum the data update (or new data additions) process requires the following steps:

1. Gather the new dataset.

This involves the work necessary to gather the raw dataset and compile it into a format that can be translated into a raster file.

2. Rasterize the raw data.

This step converts the raw data into a raster file that can be loaded into the Hectares BC system. It involves both creating a conversion script and running this script. If the data is already rasterized this step is simplified to resampling the raster into the Hectares BC specifications.

3. Create/Update Metadata.

Here all metadata files for the grids, categories, and values are populated. This information is used to create the layer tree structure in the Hectares BC system and provide users with information about the data.

4. Reload Database

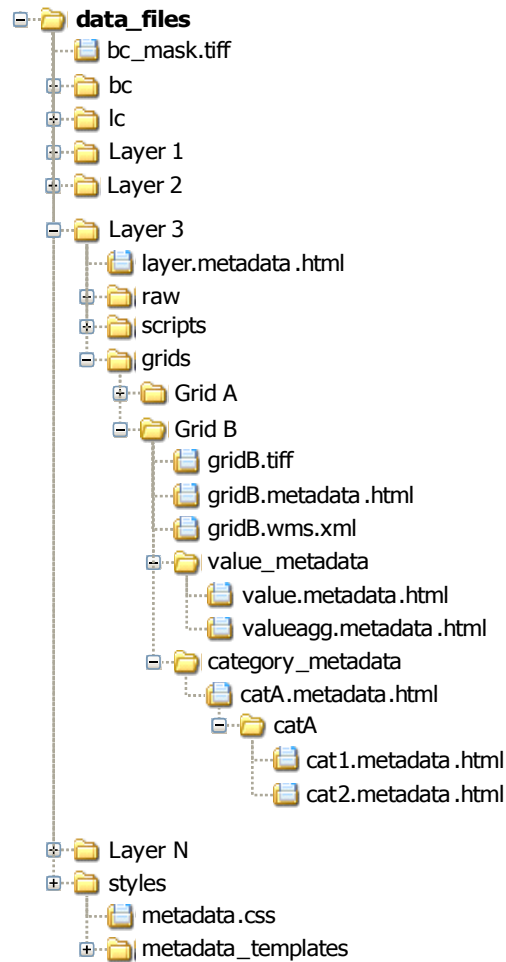
This step includes reloading the database data files, metadata, and regenerating the Mapserver map file.

### 2.2 FILE DATASTORE OVERVIEW

Any data maintainer performing data updates or additions should be familiar with the Hectares BC file datastore structure. A complete description of the file datastore and related files can be found in the Technical Documentation. An overview is provided below for reference.

All data and metadata about data layers in the Hectares BC system is stored as files. This information is used to populate the Hectares BC database that drives the application user interface. The files stored with each layer includes all the raw data, rasterization scripts, raster data, and metadata files. Metadata files are stored as html files that include fixed database key information, category to raster value links, GUI names, and any other related metadata information.

The following figure outlines the folder and file structure of the Hectares BC file datastore which consists of layers, grids, and metadata files. A layer is a collection of related data files (for example “roads and linear features”), a grid is a single data file (for example “road length” or “distance to road”). Each grid becomes a single raster file and a single column in the master grid table. Data updates and additions involve updating or adding new grids.



**Figure 1: File Datastore Overview**

All raw data files associated with a layer are placed in the **raw** folder and all rasterization scripts are placed in the **scripts** folder. Each grid associated with a layer has a folder in the **grids** directory. In this **grids** folder is the raster file, the grid metadata file, the wms metadata file, and folders for category or value metadata files (these folders only need to appear if they contain valid metadata files). The **category\_metadata** folder contains all related category metadata and the **value\_metadata** folder contains all related value metadata.

In addition to general metadata information about the particular item, each metadata file has a collection of **<meta>** tags that contain database keys and other information important to how the data is displayed in the system but not visible to the users. Templates for each of the metadata files can be found in the

`/raid/habc_data_prod/data_files/styles/metadata_templates` directory. If the `<meta>` tags are not populated correctly the data may fail to load or produce unexpected results.

## 3 ADDING NEW LAYER

### 3.1 ADDING NEW DATA LAYER STEPS

The following steps outline the process for adding a new layer:

#### 1. Locate the Hectares BC File Datastore

The file data store contains the information about the data layers in the Hectares BC application including the raster files, raw files, and metadata files.

On hectare, the development file datastore is located:

```
/raid/habc_data_devel/data_files/
```

The production file datastore is located:

```
/raid/habc_data_prod/data_files/
```

#### 2. Determine which Layer to Use

This step determines if a new layer needs to be added or if the new data layer will fit into one of the existing data layers. The data layers are all the folders located in the file datastore directory. The following “layer” folders are reserved for Hectares BC system files and should not be used to store data layers:

- bc – contains information about the BC raster
- lc – contains metadata for saved and public *show me where* queries
- styles – contains style sheets, metadata templates and script templates.

If a new layers needs to be created:

- i. Determine the layer name.

The layer name should be comprised of lower case characters (a-z) and digits (0-9). It is possible however not recommended to use underscores. Layers names should be meaningful and should not change once selected.

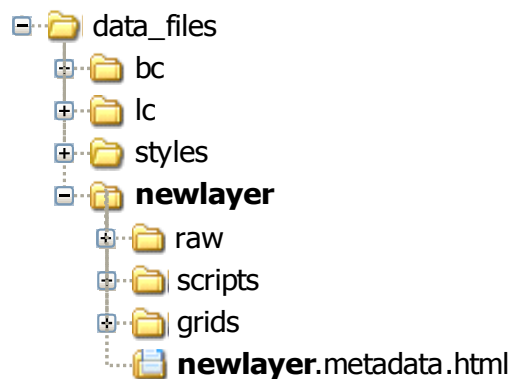
- ii. Create a new folder named with the layer name.

- iii. Inside *layername* folder:

- Create three subfolders: **grids**, **raw**, and **scripts**
- Create the layer metadata file: **layername.metadata.html**

- Populate the layer metadata (**layername.metadata.html**) file. A template for populating the layer metadata files can be found in **./styles/metadata\_templates/layer.metadata.html**. Below is a brief description of the **<meta>** tags that need to be filled out. In addition the body of the metadata file should be filled out with an overview of what the layer contains. In general it should not contain the details about the layer being added (date created etc) as this information will be placed in the grid metadata.

Field	Description
FIXEDLAYERNAME	This is the layer name selected above.
GUIORDER	This is the order the layer should appear in the tree. This is relative to all the other layers.
GUILAYERNAME	This is the name of the layer as the users will see it.



**Figure 2: New Layer File Structure**

If your layer fits within an existing layer, view the layer metadata file **layername.metadata.html** and ensure no updates are necessary to that file to accommodate the new layer.

### 3. Create the grid folder.

The grid folder is the folder in which the raster data and metadata will be placed. The grid folder must exist in the **layername/grids** directory.

To add a new grid folder:

- i. Determine the grid name.

The grid name should be comprised of all lower case characters (a-z) and digits (0-9). It is possible however not recommended to use underscores. Grid names should be meaningful and should not change once selected.

- ii. Create a folder **layername/grids/gridname** for the new grid in the **grids** directory.

#### 4. Collect the raw data

The raw data should be compiled into a single zip file and placed in the **layername/raw/** directory. This zip file should contain a metadata file that identifies the source of the raw data and date of collection.

Complex data layers may require custom storage formats.

#### 5. Rasterize the data and generate the metadata files.

This step requires the creation and running of a rasterization script to convert the raw data to the rasterized format. This script should also create all necessary metadata files.

Sections 3.2 (Simply By Value Rasterization) and 3.3 (Simple Category Rasterization) contain the steps for performing simple by value and category rasterizations.

For more complex rasterizations custom scripts will need to be generated to perform the rasterization. Any scripts used in the rasterization of the data should be stored in the **layername/scripts** folder. If another grid has a similar format to the data being rasterized it may be possible use its rasterization script as a template.

#### 6. Populate the metadata files.

The rasterization step will rasterize the data and generate all the necessary metadata files. However the rasterization step will not populate the metadata files with meaningful names. This step involves editing each of the generated metadata files to ensure all the necessary metadata information is added to these files.

The sections below identify the metadata files which need to be populated. At a minimum the grid metadata and wms xml metadata need to be populated. The value and category metadata files should also be reviewed and updated if they exist.

##### Grid Metadata

The grid metadata contains all information related to the raw data, including the source, date created, date loaded and description. The grid metadata file will be called **layername/grids/gridname/gridname.metadata.html**.

If the grid metadata files was not created by the rasterization step a template can be found in **./styles/metadata\_templates/grid.metadata.html**.

The grid metadata file includes the following **<meta>** tags that need to be populated for correct loading and display of the data in the Hectares BC application. Some of these (fixed

layer names, fixed grid names, scale factor) should be set by the rasterization script and should not need editing.

Field	Description
FIXEDLAYERNAME	The layer name associated with the grid (this will be the same as the name of the layer folder the grid is contained within).
FIXEDGRIDNAME	This is the name of the grid (the name selected for the grid folder).
GUIGRIDNAME	This is a version of the grid name that will be displayed to the user in the user interface.
GUIORDER	This is the order the grid should appear in the tree. This is relative to other grids in the same layer.
DESCRIPTION	This should contain a very brief description of the grid.
SCALE_FACTOR	The scale factor that should be applied to the data. For example if temperate values are provided as integers representing 10 <sup>th</sup> of a degree (104 = 10.4 degrees) the scale factor should be set to 0.1.
INDEXED	If the column should be indexed in the master grid tables or not. Generally this should be set to "No"

In addition the body of the grid metadata files should be updated to include the name of the dataset, the creation date, the load date, a description of the grid, the source of the grid and any other related information. A header for each of these sections is provided in the template.

A sample grid metadata file:

```
<html>
<head>
<link rel="stylesheet" type="text/css" href="../../styles/metadata.css"/>
<meta name="fixed_layer_name" content="linearfeat"></meta>
<meta name="fixed_grid_name" content="railway"></meta>
<meta name="gui_grid_name" content="Railway Lines"></meta>
<meta name="gui_order" content="3"></meta>
<meta name="description" content="Railway line length per hectare."></meta>
<meta name="scale_factor" content="1"></meta>
<meta name="indexed" content="no"></meta>
</head>
<body>
  <h1>Railway Lines</h1>

  <h2>Creation Date</h2>
  <p>1996</p>

  <h2>Description</h2>
```

```

<p>Railline transportation features extracted from TRIM.</p>

<h2>Rights</h2>
<p>This material is owned by the Government of British Columbia and
protected by copyright law. It may not be reproduced or redistributed without
the prior written permission of the Province of British Columbia.</p>

<h2>Creator</h2>
<p>Integrated Land Management Bureau (<a
href="http://ilmbwww.gov.bc.ca/">ILMB</a>) </p>

<h2>Publisher</h2>
<p>Integrated Land Management Bureau (<a
href="http://ilmbwww.gov.bc.ca/">ILMB</a>) </p>

<h2>Source</h2>
<p><a href="http://ilmbwww.gov.bc.ca/bmgs/pba/trim/">TRIM Program</a></p>

<h2>Load Date</h2>
<p>2008-03-20</p>

<h2>Refresh Period</h2>
<p>Unknown</p>

<h2>Coverage</h2>
<p>BC</p>
</body>
</html>

```

## WMS Metadata

The wms metadata file, */layername/grids/gridname/gridname.wms.xml*, contains the information necessary to color the categories of data for display in the WMS Service (and raster data tab). For categorical data it provides a color for each category, for value data it categorized the data and provides associated colors.

The wms metadata will be generated by the rasterization script; however will likely require customization for the given data layer. Generally for category data it will create one color for each category, for value data it will create a single category.

The format of this metadata files is as follows:

```

<landscape>
  <entry>
    <legend_entry>CATEGORY NAME</legend_entry>
    <values>CATEGORY VALUES</values>
    <color>
      <red>RED</red>
      <green>GREEN</green>
      <blue>BLUE</blue>
    </color>
  </entry>
  ...
</landscape>

```

The CATEGORY NAME is the name of the category as it will be displayed on the legend. The CATEGORY VALUES are the values in the raster grid associated with the given category. Commas delimit single values and colons delimit ranges. For example “1,5,7” references values 1 or 5 or 7 and “1:10” references values 1 through 10 inclusive. Notations can be combined as in “1,6:8,10” which references values 1, or 6 through 8, or 10. The RED, GREEN, and BLUE values specify the color to associate with the category.

### Value Metadata

All value metadata will be placed in the **layername/grids/gridname/value\_metadata/** folder. The value metadata will contain a single value metadata file (used in the Show Me Where tab) and multiple value aggregation metadata files (used in the How Much Tab).

The value metadata file will be called

**layername/grids/gridname/value\_metadata/gridname.metadata.html.**

The aggregation metadata files will be called

**layername/grids/gridname/value\_metadata/gridnameagg.metadata.html**

These files should be generated by the rasterization step, however if it is necessary to create further files, templates for these files can be found at

**styles/metadata\_templates/value.metadata.html** and  
**styles/metadata\_templates/valueagg.metadata.html.**

These files will have the following **<meta>** tags that need to be populated correctly in order for the data to be loaded and displayed correctly. Some of these fields (fixed layer, fixed grid, and fixed value names) should be populated by the rasterization script and should not need modifying.

Field	Description
FIXEDLAYERNAME	The layer name associated with the grid (this will be the same as the name of the layer folder the grid is contained within).
FIXEDGRIDNAME	This is the name of the grid (the name selected for the grid folder).
FIXEDVALUENAME	This is the name of the value. Generally this should be the same as the fixed grid name.
GUI VALUE NAME	The name of the value as it will be displayed to the user.
GUI ORDER	The order the value is displayed to the user relative to other values for the grid.
UNITS	The units the value are measured in.
AGGREGATION (aggregation metadata files only)	The type of aggregation to apply to the metadata (sum, avg etc.). See Section 3.2 - Simply By Value Rasterization.

## Category Metadata

All category metadata will be placed in the **layername/grids/gridname/category\_metadata/** folder. All the category metadata files should be created and populated by the rasterization script with the information necessary to load the data. However, it will be necessary to update the **gui\_category\_value** metadata tag and the body of the metadata files.

If additional category metadata files are required a template can be found in **styles/metadata\_templates/category.metadata.html**.

The category metadata files have the following **<meta>** tags. With the exception of the **gui\_category\_value** these fields should be populated correctly by the rasterization script and require no modification.

Field	Description
FIXEDLAYERNAME	The layer name associated with the grid (this will be the same as the name of the layer folder the grid is contained within).
FIXEDGRIDNAME	This is the name of the grid (the name selected for the grid folder).
FIXEDCATEGORYNAME	The fixed category name.
FIXEDCATEGORYVALUE	The fixed category value.
GUI ORDER	The order the value is displayed to the user relative to other categories in the same level in the hierarchy (all the other files in the same folder).
GRID_VALUES	A valid sql statement that links the values in the raster file to this particular category.
GUI_CATEGORY_VALUE	The category value to be displayed to the user.

## 7. Reload the Database

See the Administrator Guide for information on how to reload the database.

## 3.2 SIMPLY BY VALUE RASTERIZATION

This section describes the steps necessary to rasterize a simple by value layer.

Simple by value rasterizations make the following assumptions about the input data files:

- raw data is provided as a single shapefile
- the raw data consists of polygonal or linear data

## 1. Create a rasterization script

All scripts used to convert from raw format to rasterized format should be placed in the **layername/scripts/**. To create the rasterization script:

- i. In the scripts directory create a conversion script file called **convert\_gridname.pl** (or another meaningful name).
- ii. From the **/styles/scripts\_templates/** folder copy:
  - the contents of the **convert\_byvalue\_linear.pl** script to the newly created script file
  - the entire **java** folder to the **layername/scripts/** folder.

## 2. Modify the convert\_gridname.pl file

The template conversion file contains some place holders for data layer specific parameters. These must be replaced with the valid values for the layer being rasterized.

Lines 48-58 will contain something similar to:

```
#hectares bc layer information
my $layername = "<LAYERNAME>";
my $gridname  = "<GRIDNAME>";

#input data specifications
my $rawfile   = "../raw/<RAWZIPFILE>.zip";
my $shapefile = "<SHAPFILENAME>.shp";
my $datasrid  = 3005;
my $workingdir = "./temp/";
my @aggregations = ("AGGREGATION1", "AGGREGATION2"); #sum, avg etc.
my $rasterizationtype = "<Linear | Polygonal>";
```

The **layername**, **gridname**, **rawfile**, **shapefile**, **aggregations**, and **rasterizationtype** variables all need to be updated.

- The **layername** variable should be set to the layer name selected.
- The **gridname** variable should be set to the grid name selected.
- The **rawfile** variable should be the relative reference to the raw data zip file (relative from the **scripts** directory). These files should also be placed in the **../raw/** directory.
- The **shapefile** variable should be set to the name of the shapefile (.shp) file in the raw zip file.
- The **rasterizationtype** should be set to **Linear** or **Polygonal** depending on your input data type (linear features or polygonal features).

- **@aggregations** is the list of aggregations that are available for the data layer (these will be displayed in the layer tree in the summary tab). The following values are allowed: sum, avg, min, and max. It is possible to use addition aggregations; however some database modifications are necessary. See the Technical Documentation for information on adding aggregations. A couple of examples are provided here:

```
my @aggregations = ("sum");
my @aggregations = ("sum", "min", "max");
```

Below are examples for lines 48-58:

```
#hectares bc layer information
my $layername = "linearfeat";
my $gridname = "railway";

#input data specifications
my $rawfile = "../raw/railwaylines.zip";
my $shapefile = "rails.shp";
my $datasrid = 3005;
my $workingdir = "./temp/";
my @aggregations = ("sum"); #sum, avg etc.
my $rasterizationtype = "Linear";
```

### 3. Run the rasterization script

The rasterization script is designed to run on a relative folder structure so ensure that the script is run from within the **scripts** directory.

```
> cd /raid/habc_data_devel/data_files/layername/scripts/
> perl convert_gridname.pl
```

Depending on the complexity of the data layers running the script could take substantial time. For small datasets (railways, transmission lines) it will likely take a couple of hours. For large complex datasets (roads, stream) it may take a couple of days.

### 4. Review the output

The output from the rasterization should be reviewed to ensure the results are reasonable.

The rasterization script should create the following files/folders:

- a tiff file: *layername/grids/gridname/gridname.tiff*
- a grid metadata file: *layername/grids/gridname/gridname.metadata.html*
- a wms xml file: *layername/grids/gridname/gridname.wms.xml*
- a value\_metadata folder: *layername/grids/gridname/gridname/value\_metadata*
- a value\_metadata file: *layername/grids/gridname/value\_metadata/gridname.metadata.html*

- one aggregation file for each aggregation specified:  
*layername/grids/gridname/value\_metadata/gridnameaggregation.metadata.html*

The example script provided above would produce the following file/folder structure:

```
/linearfeat/grids/railway/  
/linearfeat/grids/railway/railway.tiff  
/linearfeat/grids/railway/railway.metadata.html  
/linearfeat/grids/railway/railway.wms.xml  
/linearfeat/grids/railway/railway/value_metadata/  
/linearfeat/grids/railway/railway/value_metadata/railway.metadata.html  
/linearfeat/grids/railway/railway/value_metadata/railwaysum.metadata.html
```

### 3.3 SIMPLE CATEGORY RASTERIZATION

This section describes the steps necessary to rasterize a simple by categorical layer.

A simple categorical rasterization makes the following assumptions:

- raw data is provided as a single shapefile
- the raw data consists of non-overlapping polygons
- the raw data contains a single set of hierarchical attributes to rasterize on (for example: zone, subzone, and variant).

#### 1. Create a rasterization script

All rasterization scripts should be placed in the *layername/scripts/* directory. To create the categorical rasterization script:

- i. In the scripts directory create a conversion script file called **convert\_gridname.pl** (or another meaningful name).
- ii. From the */styles/scripts\_templates/* folder copy the contents of the **convert\_category\_template.pl** script to the newly created script file

#### 2. Modify the convert\_gridname.pl file

The template conversion file contains some place holders for data layer specific parameters. These must be replaced with the valid values for the layer being rasterized.

Lines 60-70 of the *convert\_gridname.pl* file will contain the following variables.

```
#layer information  
my $layername = "LAYERNAME";
```

```

my $gridname = "GRIDNAME";

#raw data information
my $rawfiles = "../raw/RAWZIPFILE.zip";
my $shapefile = "SHAPEFILE.shp";
my $datasrid = 3005;

# rasterization attributes
my @attributenames = ("CATEGORY1", "CATEGORY2", "CATEGORY3");

```

These variables need to be updated to reflect the dataset being rasterized:

- The **layername** variable should be set to the layer name selected.
- The **gridname** variable should be set to the grid name selected.
- The **rawfile** variable should be the relative reference to the raw data zip file (relative from the **scripts** directory). The raw data file should always be placed in the **../raw/** directory.
- The **shapefile** variable should be set to the name of the shapefile (.shp) file in the raw zip file.
- The **@attributenames** is a list of the attributes to rasterize the data on. The attributes in this list are assumed to be hierarchical in nature with CATEGORY1 being the root category and CATEGORY2 a subcategory. As many attribute names can be specified as needed, but at least one needs to be specified. The name needs to exactly match the attribute name in the raw shapefile.

An example of values for these variables is shown below:

```

#layer information
my $layername = "bec";
my $gridname = "bec";

#raw data information
my $rawfiles = "../raw/becdata.zip";
my $shapefile = "beczones.shp";
my $datasrid = 3005;

# rasterization attributes
my @attributenames = ("zone", "subzone", "variant");

```

### 3. Run the rasterization script

The rasterization script is designed to run on a relative folder structure so ensure that the script is run from within the **scripts** directory.

```

> cd /raid/habc_data_devel/data_files/layername/scripts/
> perl convert_gridname.pl

```

Depending on the complexity of the data layers this could take substantial time. For small simple datasets it may take a few minutes, for complex datasets it could take several hours to a day.

#### 4. Review the output

The output from the rasterization should be reviewed to ensure the results are reasonable.

The rasterization script should create the following files/folders:

- a tiff file: *layername/grids/gridname/gridname.tiff*
- a grid metadata file: *layername/grids/gridname/gridname.metadata.html*
- a wms xml file: *layername/grids/gridname/gridname.wms.xml*
- a category metadata folder: *layername/grids/gridname/category\_metadata/*
- a root category metadata file:  
*layername/grids/gridname/category\_metadata/category1.metadata.html*
- a root category folder: *layername/grids/gridname/category\_metadata/category1/*
- a metadata folder for each root category value:  
*layername/grids/gridname/category\_metadata/category1/category1value1.metadata.html*  
*layername/grids/gridname/category\_metadata/category1/category1value2.metadata.html*  
...  
*layername/grids/gridname/category\_metadata/category1/category1valueN.metadata.html*
- if subcategories are defined then a folder should be created for each category value and within these the metadata files for each of the subcategories.  
*layername/grids/gridname/category\_metadata/category1/category1value1/*  
*layername/grids/gridname/category\_metadata/category1/category1value1/subvalue1.metadata.html*  
*layername/grids/gridname/category\_metadata/category1/category1value1/subvalue2.metadata.html*

## 4 UPDATING AN EXISTING DATA LAYER

### 4.1 UPDATING NEW DATA LAYER STEPS

The following steps outline the process for updating a new layer.

#### 1. Locate the Hectares BC File Data Store

The file data store contains all information about all data layers in the Hectares BC application including the raster files, raw files, and metadata files.

On hectare, the development file data store is located:

```
/raid/habc_data_devel/data_files/
```

The production file data store is located:

```
/raid/habc_data_prod/data_files/
```

#### 2. Backup Existing Data & Empty *gridname* Directory

This steps involves archiving the existing data and clearing out all files to ensure the directory is ready for rasterization. All data files and metadata should be retained for reference purposes. As well these files will be used to update the new metadata files once the data has been re-rasterized.

Backup is critical as the rasterization step will overwrite any existing files in the *gridname* directory. To backup the current data grid to be updated:

- zip up the **layername/grids/gridname/** folder
- name the zip file **gridname.ddmmyyyy.zip**
- create an archive directory in the **layername** folder (along with the **scripts**, **raw**, and **grids** folders) if one does not already exist
- place the zip file in the archive directory

Once the backup has been completed it is necessary to remove all files from the **layername/girds/gridname** folder. This will ensure that all metadata files are removed.

#### 3. Collect the Raw Data

The raw data should be compiled into a single zip file and placed in the **layername/raw/** directory. This zip file should include a metadata file that identifies the source and date collected of the raw data.

Complex data layers may require custom storage formats.

#### 4. Rasterize

If the new data is in identical format to the existing raw data then the existing rasterization scripts can be used. If existing rasterization scripts are being used it is recommended to view the variables defined at the top of the rasterization scripts and ensure they are correct. These variables are the same variables described in sections 3.2 (Simply By Value Rasterization) and 3.3 (Simple Category Rasterization). It is also recommended that any comments are reviewed as for some cases the complex rasterization scripts require manual intervention or other special processing.

If the data format has changed in any way a new rasterization script will likely have to be created. This can be done by either copying the existing rasterization script and making the necessary modifications to deal with the data format changes or by following the rasterization steps outlined in sections 3.2 (Simply By Value Rasterization) and 3.3 (Simple Category Rasterization). For more complex rasterizations custom scripts may need to be developed to perform the rasterization. Any new rasterization scripts generated should be stored in the **layername/scripts** folder.

Once the rasterization script has been created it can be run. The rasterization script must be run from within the scripts directory.

```
> cd /raid/habc_data_devel/data_files/layername/scripts/  
> perl convert_gridname.pl
```

Rasterizing the data will overwrite all existing metadata so it is critical that the backup (Step 1) is performed correctly.

#### 5. Verify the Rasterization

The output from the rasterization should be reviewed to ensure the results are reasonable.

#### 6. Update Metadata Files

Rasterization will re-create all metadata files associated with the grid. This includes:

- the grid metadata file: *layername/grids/gridname/gridname.metadata.html*
- the wms xml metadata file: *layername/grids/gridname/gridname.wms.xml*
- any value metadata files in the *layername/grids/gridname/value\_metadata* folder
- any category metadata files in the *layername/grids/gridname/category\_metadata* folder

All of these files need to be repopulated with the relative information. Currently there are no tools to aid in the updating of the metadata.

### **Maintaining *fixed* Names**

The **fixed** names contained in the metadata files (**fixedlayername**, **fixedgridname**, **fixedvaluename**, **fixedcategoryname**, and **fixedcategoryvalue**) are used by the system as unique identifiers. These unique identifiers are used to build up queries and used when saving queries.

The rasterization step will generate fixed names based on the values provided in the rasterization script and the attribute values in the raw data files (for category names). These could easily change between versions. For example the BEC zone attribute in the first version may be BAFA, and in the second version it may change to Boreal Altai Fescue Alpine; this results in the fixed names changing from **bafa** to **borealaltaifescuealpine** and any queries that reference the original **bafa** to become invalid.

If these unique identifiers change then any saved queries (or batch queue items) which reference the old layers become invalid. In order to prevent queries becoming from invalid it is important to maintain these fixed variable names. These fixed names are referenced in the metadata file (inside the **meta** tags) and by the file and/or folder name. If a fixed name needs to be modified the **meta** tag needs to be updated as well as the file name and any corresponding folder names. If this is not done correctly the dataload will fail.

### **Grid Metadata**

Grid metadata can be updated by copying the contents of the achieved grid metadata file and updating and relevant dates (date created, load\_date).

### **WMS XML Metadata**

The WMS XML metadata files will have to be regenerated manually. The rasterization step will reproduce this file however if specific colors were assigned to specific categories this color information will have to be reproduced. In addition legend names may require updating.

For categorical based rasters the category values may differ between versions (bec zone bafa may reference value 2 in the version 1 raster, and value 4 in the version 2 raster) so do not simply copy over the archived wms xml metadata file otherwise the legend labels will not reference the correct values.

### **Value Metadata**

The value metadata files can be updated by copying the contents of bodies of the achieved value metadata files to the new metadata files. Modifications to GUI labels can also be made as required.

### **Category Metadata**

The category metadata files are the most complex to update. Because the link between the category and grid value will have changed (as a result of the re-rasterization) the category metadata files cannot be reused.

The archived category metadata files will require manual matching to the new category metadata files and the gui\_category\_names, and any <body> metadata information will

require manual transferring. This is also where the “fixed” names will most likely need updating. See Maintaining *fixed* Names above.

## **7. Reload the Database**

See the Administrator Guide for information on how to reload the database.