Archaeological Consulting

Quesnel Forest District GIS Model of Archaeological Potential Revision Project, 2008.

Prepared by: Matrix Research Ltd.

Prepared for: Canadian Forest Products Ltd.

April, 2008

CONTACTS

Quesnel Forest District GIS Model of Archaeological Potential Revision Project, 2008.

Prepared for:

Canadian Forest Products Ltd.
Quesnel Division
1920 Brownmiller Road
Quesnel, B.C. V2J-6S1
Phone: (250) 992-5581
Fax: (250) 992-8520

Prepared by:

Matrix Research Ltd.
395 Johnston Avenue
Quesnel, B.C. V2J-3M6
Phone: (250) 992-6623
Fax: (250) 992-6653

March 2008

ACKNOWLEDGEMENTS

Matrix Research Ltd. would like to thank Canadian Forest Products Ltd., Quesnel Division for the opportunity to conduct this model revision. In particular, we thank Steven Day who initiated and guided us through the project.

This project was a joint effort between Matrix Research Ltd. and Spatial Mapping Ltd. Steven Monai and Hesheng Zhang of Spatial are thanked for their innovative ideas, problem solving approaches, and quick turn around times.

Many people helped to supply the various information compiled and / or consulted during this review. Nathan Strand and Steven Day from Canadian Forest Products Ltd. supplied and / or coordinated access to digital map coverages necessary for the project. Leslie Lund at the Ministry of Forests and Range provided data pertaining to the 1998 Quesnel Forest District GIS Model of Archaeological Potential. Morgan Chenier of West Fraser Mills Ltd. provided digital map data. Claire Tweeddale and Chris Niziolomski of Forest Ecosystem Solutions Ltd. provided digital map data. Mark McGirr from Integrated Land Management Bureau, Ministry of Agriculture and Lands supplied a ground-truthed and partially GPS'd map file for the Alexander Mackenzie Heritage Trail (aka Nuxalk-Carrier Grease Trail). Mark also clarified some irregularities in the Ministry of Environment stream classification files.

Al Mackie, John McMurdo, and Eric Forgeng of the Archaeology Branch, Ministry of Tourism Sport and the Arts provided current lists of heritage permit reports on file at the branch as well as assisted in transferring the reports needed for this project.

Ewan Anderson from Arcas Consulting Archaeologists Ltd. provided data on recent archaeological assessments completed by Arcas within the Quesnel Forest District.

Although many people and organizations provided the data necessary for the project, Matrix Research Ltd. alone is responsible for the accuracy of the final product.

CREDITS

Project Management

Ty Heffner M.A., R.P.C.A.

Report Author

Ty Heffner

Research

Ty Heffner
Richard Burkmar, B.A., M.Sc.
Michael Ross, B.A.

Spatial Mapping GIS Personnel

Steven Monai
Hesheng Zhang

Matrix GIS Supervisor

Deanna Windsor

Administration

Sharon Alton

TABLE OF CONTENTS

CONTACTS ..... ***i***

ACKNOWLEDGEMENTS ..... ***ii***

CREDITS ..... ***iii***

TABLE OF CONTENTS ..... ***iv***

1.0 Introduction ***5***
Map 1: Study Area Location 8

2.0 Background and Project Area ***9***

2.1 The Study Area **9**

3.0 Methodology ***10***

3.1 Data Compilation **10**
Table 1: Source Data Sets 10
Table 2: Layers Created 11

4.0 Modelling Results ***17***
Map 2: Revised GIS Model of Archaeological Potential. 18

5.0 References Cited ***19***

Appendix A: Stakeholder Workshop Summary

Appendix B: Feature Definitions, Sources, Code and Buffers

Appendix C: Revised Model Script (on CD only)

1.0 INTRODUCTION

This GIS Model of Archaeological Potential revision project was conducted at the request of Canadian Forest Products Ltd. The objective of the project was to revise the existing GIS Model of Archaeological Potential (Arcas 1998) to increase its effectiveness and make it conform to new Archaeological Overview Assessment (AOA) standards and guidelines (Archaeology Branch 2008). Data available for modeling has increased since the original GIS Model of Archaeological Potential was created in 1998. Updated data includes a limited amount of new digital base mapping information, as well as ten years of archaeological survey data. Archaeological survey data has been compiled and summarized in a recently completed archaeological data inventory project (Heffner 2007) for the Quesnel Forest District. Based on that newly compiled data, an archaeological data gap analysis was also conducted for the Quesnel Forest District (Heffner *et al.* 2007). The archaeological data gap analysis reviewed the effectiveness of the existing archaeological overview assessment model, identified gaps in the current archaeological record (site and survey), and provided recommendations for revising the current model. The current project represents the culmination of these efforts.

Standards followed during this study include those contained in the *British Columbia Archaeological Impact Assessment Guidelines* (Archaeology Branch 1998), the *British Columbia Archaeological Inventory Guidelines* (Archaeology Branch 2000), and the upcoming *Archaeological Overview Assessments as General Land Use Planning Tools: Standards and Guidelines*, released in draft form (Archaeology Branch 2008).

Definitions

This study incorporates cultural heritage resource data. According to the *Heritage Conservation Act* (Province of British Columbia 1986), cultural heritage resources include both protected and non-protected resources. This broad definition encompasses a wide variety of site categories such as archaeological and traditional use sites. An archaeological site is any geographical location that contains physical evidence of past human activity. Archaeological sites that predate 1846 AD are automatically protected by the *Heritage Conservation Act*. Examples include lithic scatters, cultural depressions, petroglyphs, and pictographs. Traditional use sites post-date 1846 AD and represent a resource category that has meaning in cultural tradition both conceptually (i.e. spiritually) and tangibly (i.e. through traditional physical activity). These resources are often representative of geographically defined areas that have traditionally been used by one or more contemporary groups of aboriginal people for one or more culturally significant activities, such as bathing pools and locations of significant events. Also included under this definition are resource gathering areas, such as post-1846 culturally modified tree (CMT) sites. For purposes here, cultural heritage resources will refer to historic sites and post-

1846 CMT sites, resources that are not automatically protected under the *Heritage Conservation Act*.

This study summarizes the results of an update and revision of the GIS Model of Archaeological Potential. This study does not address, evaluate, or comment on traditional aboriginal use of the area and should not be considered valid for that purpose.

Project Design and Scheduling

This project was designed to occur in three phases. This report represents the completion of Phase 2. The following is a brief description of each phase.

Phase 1

The objective of Phase 1 was to organize a workshop for various stakeholders in Quesnel Forest District. The purpose of the workshop was to introduce the project, describe the efforts undertaken to date, provide information on the proposed approach to the project, and obtain feedback. The workshop was held on January 22, 2008 and was well attended. The workshop summary is attached to this report as Appendix A.

Phase 2

Objectives of Phase 2 included the following: 1) update to the archaeological data inventory, 2) recreation of the original model, 3) targeted data gathering and creation of an updated data set, 4) analysis and revision of the model to increase effectiveness and efficiency, and 5) creation of a summary report and accompanying maps of the proposed new model for circulation to the stakeholder groups.

Phase 3

During this phase, tentatively scheduled for May 2008, a final stakeholder workshop will be held to review the proposed new model and obtain any feedback that is forthcoming. Any necessary refinements will be made to the model. Project deliverables will be finalized and distributed to stakeholders.

Project Deliverables

Deliverables resulting from Phase 2 of this study include the following: 1) this draft report, which provides a summary of the project as well as documentation of the methods employed; 2) an updated archaeological data inventory; and 3) a draft revised GIS Model of Archaeological Potential for Quesnel Forest District. Digital map files and an electronic copy of this report in PDF format are contained on the DVD located in the back cover of this report.

Use of Archaeological Data

Please note that archaeological site location information is contained in the archaeological site database. The Archaeology Branch has authority over access to this information in accordance with the Heritage Conservation Act. This information is provided to archaeologists, development proponents, First Nations, and other concerned parties on a need to know basis. The site information provided in these files is in agreement with the Third Party Access section of the Heritage Register Data Request Form. Under the terms of the data request submitted to the Archaeology Branch for this project, the archaeological site information can be used by Canadian Forest Products Ltd., other forest licensees, First Nations, and the Ministry of Forests and Range but cannot be distributed to any other third parties without the written permission of the Archaeology Branch. Copyright of digital site information belongs exclusively to the Province of British Columbia.

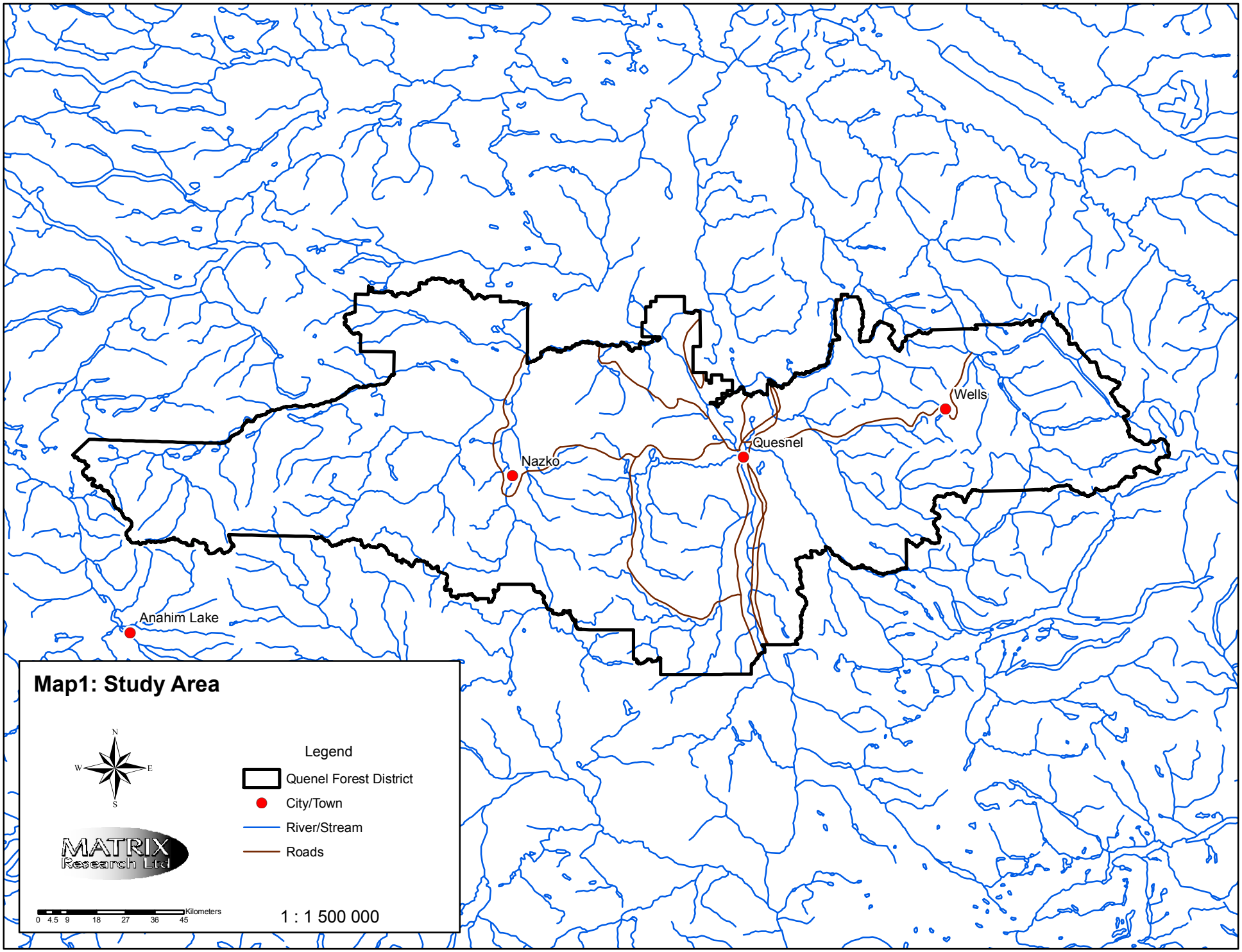
The database was compiled from archaeological site forms and permit reports for planning purposes and to serve as an overview of archaeological site information. It is not intended to amend or replace management recommendations provided in the original permit reports. References are provided throughout the database to ensure easy access to the results and recommendations of individual assessments.

Data Currency

It is important to note that the archaeological data summarized during this project changes through time as more assessments are conducted and sites are located or updated. The utility of a database of this type is enhanced by periodic updates as new data become available.

Archaeological site data contained in this database is current as of March 10, 2008 (the date the heritage data was downloaded from the Remote Access to Archaeological Data (RAAD) website). We also included data for archaeological sites recorded by Matrix Research Ltd. for which site forms had been submitted after March 10, 2007. These sites were included so that the data set would be as complete as possible.

Archaeological assessment data is complete up to and including the 2007 permit year. At the time of this project, not all 2007 permit reports had been submitted to the Archaeology Branch. Data on cultural heritage resource sites is as current as the permit reports from which the data were compiled. This project was undertaken concurrently with a trail inventory within the Quesnel Forest District. Consequently, heritage trail research is current to March 10, 2008. All maps containing trails that were obtained during the archival research component of the 2007 archaeological data inventory project (Heffner 2007) have been mapped and are included in the trail layer.


Wells
Quesnel
Nazko
Anahim Lake
Map1: Study Area
N
W
E
S
MATRIX Research Ltd
Legend
Quenel Forest District
City/Town
River/Stream
Roads
0 4.5 9 18 27 36 45
Kilometers
1 : 1 500 000

2.0 BACKGROUND AND PROJECT AREA

2.1 The Study Area

The study area encompasses the entire Quesnel Forest District (Map 1). The Quesnel Forest District represents a land base of approximately 2,077,233 hectares. A description of the biophysical characteristics (past and present) and a summary of the ethnographic cultures of the study area can be found in the report of the *Archaeological Data Gap Analysis of the Quesnel Forest District, 2007* (Heffner *et al.* 2007).

3.0 METHODOLOGY

3.1 Data Compilation

For a project of this nature it is essential to have the most complete and up-to-date information possible. Considerable efforts were taken at the beginning of this project to research and obtain the best possible data. The two categories of data included digital map data, such as stream classifications and digital elevation models, and archaeological data, including the locations and attributes of archaeological sites, assessment areas, and trails.

Digital Map Data

Digital map data was available from a number of sources and details are provided in Table 1. The source datasets were each manually loaded into file geodatabases for use in the geoprocessing scripts that were developed. The processing scripts were implemented in the Python language and perform the tasks of buffering the various source features to the prescribed distances, rasterizing the buffers, and storing the rasters in a mapsheet-tiled directory structure, one directory per BCGS 1:20,000 mapsheet tile. The specific data format used to store the rasterized tiles is the ArcInfo GRID format.

Table 1: Source Data Sets

Source Dataset Name [Format]	Dataset Provider	Brief Dataset Description	Date Obtained
LAT_Q [DEM lattice]	Canfor Quesnel (Nathan Strand)	Digital Elevation Model, 10-metre resolution	2008/01/28
TRIM [mapsheet-tiled coverages]	LRDW	TRIM	2008/01/30
ERC_ECOSEC [shapefile]	LRDW	Province-wide Ecosection polygons (a.k.a. "big-BEC")	2008/01/28
1998 arch. model dataset [raster GRID]	MoFR (Leslie Lund)	1998 working model dataset, containing both input and output layers	2007/02
TLKS_RCA [coverage]	LRDW	Quesnel FD lake polygons	2008/01/28
Arcas_Lithic_Procurement_Zone [shapefile]	Matrix Research (Deanna Windsor)	Lithic procurement zone polygons	2008/03/26
VRI [shapefiles]	LRDW	VRI in Quesnel FD mapsheets, except those within West Fraser's TFL area	2007
West Fraser TFL VRI [shapefiles]	West Fraser (via Canfor, Nathan Strand)	VRI in West Fraser's TFL area	2008/02/01
Archaeological_Sites_no_CMTs [shapefile]	Matrix Research (Deanna Windsor)	Known archaeological sites	2008/04/04
PEM_BEC [coverage]	Forest Ecosystem Solutions (Claire Tweeddale)	BEC zones/subzones in Quesnel FD	2008/01/28
Recorded_Trails_2008 [shapefile]	Matrix Research (Deanna Windsor)	Heritage Trails	2008/03/26
UWR_CL [coverage]	LRDW	Ungulate Winter Ranges (for Mule Deer) in Quesnel FD	2008/01/28

WHA_CL [coverage]	LRDW	Wildlife Habitat Areas in Quesnel FD	2008/01/28
Wetlands [shapefile]	Canfor Quesnel (Nathan Strand)	Wetlands polygons in Quesnel FD	2008/02/14

From these data, the following layers were created (Table 2). These layers correspond to the layers used by the 1998 GIS Model of Archaeological Potential. One additional layer was created. The layer 'streamS4' was separated from the existing 'streamfish' layer to allow more flexibility in refining the logical statements.

Table 2: Layers Created

Model Data Layer Name	Source Dataset Name(s) [Format(s)]	Brief Description of the Feature Creation or Selection Process
SLOPE	LAT_Q [DEM lattice]	generate slope class polygons from the lattice
CLIFFS	TRIM [mapsheet-tiled coverages]	extract features coded as cliffs
ECO	ERC_ECOSEC [shapefile]	clip dataset to Quesnel FD and use all polygons
ESKER	TRIM [mapsheet-tiled coverages]	extract features coded as eskers
FALLS	TRIM [mapsheet-tiled coverages]	extract features coded as falls
GLACIALAKE	1998 arch. model dataset [raster GRID]	extract GLACIALAKE from 1998 dataset and use as-is
LAKELARGE	TLKS_RCA [coverage]	query polygons of area > 250 ha
LAKEMED	TLKS_RCA [coverage]	query polygons of area btw 100 and 250 ha
LAKESMALL	TLKS_RCA [coverage]	query polygons of area btw 5 and 99.9 ha
LAKEVS	TLKS_RCA [coverage]	query polygons of area < 5 ha
LITHICZONE	Arcas_Lithic_Procurement_Zone [shapefile]	use all polygons
MELTLARGE	1998 arch. model dataset [raster GRID]	extract MELTLARGE from 1998 dataset and use as-is
MELTSMALL	1998 arch. model dataset [raster GRID]	extract MELTSMALL from 1998 dataset and use as-is
OLDGROWTH	VRI [shapefiles]; West Fraser TFL VRI [shapefiles]	query polygons for pine/cedar species and ageclass 8+
OPENRANGE	VRI [shapefiles]; West Fraser TFL VRI [shapefiles]	query polygons for open range
RAPIDS	TRIM [mapsheet-tiled coverages]	extract features coded as rapids
SITEPLY	Archaeological_Sites_no_CMTs [shapefile]	query polygons larger than a threshold size
SITEPNT	Archaeological_Sites_no_CMTs [shapefile]	query polygons smaller than a threshold size
STREAMFISH	Shapefiles from West Fraser and Canfor	fish-bearing double and single line streams, excluding salmon stream network
STREAMOTHR	Shapefiles from West Fraser and Canfor	Streams excluding salmon-bearing, fish-bearing and S4 (non salmon-bearing) streams
STREAMS4	Shapefiles from West Fraser and Canfor	S4 streams excluding salmon-bearing streams
STREAMSALM	Shapefiles from West Fraser and Canfor	salmon-bearing double and single line streams (as per FISS vector coverage) plus all tributaries (s1-

		s4, 0-8% grade) to these streams
SUBZONE	PEM_BEC [coverage]	use all polygons (BEC Zones and Subzones)
TRAILGOOD	Recorded_Trails_2008 [shapefile]	query features with confidence='High'
TRAILMOD	Recorded_Trails_2008 [shapefile]	query features with confidence='Moderate'
TRAILPOOR	Recorded_Trails_2008 [shapefile]	query features with confidence='Low'
UNGULATE	UWR_CL [coverage]; WHA_CL [coverage]	use all UWR_CL polygons for Mule Deer; query Caribou polygons from WHA_CL
WETLARGE	Wetlands [shapefile]	query polygons of area > 5 ha
WETSMALL	Wetlands [shapefile]	query polygons of area < 5 ha
WHITEBARK	VRI [shapefiles]; West Fraser TFL VRI [shapefiles]	query polygons for whitebark pine
ZONE	PEM_BEC [coverage]	dissolve polygons on BEC Zone

Model Script

The script of the 1998 GIS model of archaeological potential was converted from Arc Macro Language (AML) to the Python language, so that it would run most conveniently in the modern ArcGIS software environment. [Python is now the preferred geoprocessing language for ArcGIS; AML is deprecated by ESRI and may eventually be phased out entirely.] The model takes the code values of the mapsheet-tiled, rasterized data layers as its input, and produces as its output a set of archaeological potential ratings for each of the model's Site Types. The model also calculates the overall archaeological potential for each location as the maximum of all the Site Type ratings at that location. The model's outputs are stored in the same attribute tables as the input data layers. Finally, after the model has run, additional Python scripts convert the model's output from raster cells to vector polygons, and then they calculate the Kvamme Gain Statistic. The revised model script is attached to this report as Appendix C.

Archaeological Data

For details on the archaeological data used in this project, readers are referred to the report for the recently completed archaeological data inventory of the Quesnel Forest District (Heffner 2007) for information on the sources consulted and the methodology utilized while compiling the archaeological information that serves as the basis for this project. The 2007 data inventory was updated during this project to include the most up to date information possible. The updated data inventory includes 3337 assessment areas, 1338 previously recorded archaeological sites, 458 cultural heritage resource sites, and 227 heritage trails.

Logical Statements

The 1998 GIS Model of Archaeological Potential uses a deductive approach to assign levels of archaeological potential to various portions of the landscape. It lists a series of traditional aboriginal land use activities, determines the archaeological site type correlates, and

summarizes the environmental variables most favourable to the traditional activity and site type. A series of logical statements is used to assign archaeological potential to areas that exhibit favourable environmental variables.

During this project, the logical statements and their resulting archaeological potential output were reviewed in detail. In a past project (Heffner *et al.* 2007), a series of observations was made regarding areas where the model could be improved. Some of the major shortcomings identified are as follows:

- The model uses inappropriate buffer widths (in some cases 50 m around large lakes [e.g., Tzenzaicut Lake] vs. 250 m around small streams [e.g., tributaries of Tzenzaicut Lake] that flow into those large lakes).
- Stream and lake buffers that do not account for wet meadow margins.
- Stream buffers that do not account for escarpments or terraces located considerable distances from the streams.
- Stream buffers along small mapped streams that do not exist on the ground (most pronounced in eastern portions of Quesnel Forest District.
- Pixelation along edges of potential zones that creates millions of polygons.
- There is no way to determine attributes used by the model in assigning high potential. This produces a black box effect. Ideally, the layers should be queriable so that the basis for the potential assignment is known.
- The model has never been updated. There is a considerable amount of new data (digital map information [e.g., PEM] as well as archaeological data) available or forthcoming that would improve the model considerably.
- There are too many potential zones. For resource management purposes there should be a single high potential layer.
- The model does not incorporate negative survey results.
- The model misses many small wetlands that have demonstratedly high archaeological potential.
- The model should be thoroughly reviewed using orthophotos to prevent / correct numerous problems (missed features, misfit buffers, etc.)

Model revision efforts focused on correcting these recognized deficiencies. In some cases the problem does not lie with the model itself but with the digital data available. In other cases, it is changes that occurred to the model prior to implementation that are responsible. The following section reviews each of these issues individually.

Buffer Widths

Over the years, our observation has been that, in some circumstances, the 1998 model assigns high potential buffer widths that are disproportionate to the environmental feature being buffered. This issue was explored during the 2007 data gap analysis (Heffner *et al.* 2007). A near-analysis was conducted to determine at what distances from specific environmental features the majority of archaeological sites are located. The near-analysis confirmed the observations and the near-analysis data were used to refine buffer widths. Major changes are as follows:

- The definition of large lake was changed from >1000 ha to >250 ha. Very few lakes in Quesnel Forest District are >1000 ha in size. This change has resulted in wider buffers around the larger lakes. This change also necessitated a change to the definition of medium lake from 100 – 1000 ha to 100 – 250 ha.
- Buffer 2 of large wetlands was changed from 50 – 300 m to 50 – 100 m. This allows more flexibility in assigning high potential to large wetlands.
- The buffer around open range polygons was reduced from 200 m to 100 m. This reduces the default high potential buffer around open range.

Meadow Margins

Another observed shortcoming of the 1998 model is that buffers did not account for wet meadow margins located adjacent to other hydrological features. For example, in the case of a stream buffered at 100 m that was surrounded by a 75 m strip of wetland, only 25 m of dry ground adjacent to the wetland would be considered high potential rather than the intended 100 m. The same scenario applies with lakes that have wet meadow margins. It is important that the model account for these areas because wet stream margins represent floodplain areas where a stream has meandered back and forth over the years and wet lake margins may represent former lake levels. This problem was resolved by classifying any wetlands located adjacent to other hydrological features as though they were part of that feature.

Escarpment and Terraces

The 1998 model has no formalized means of identifying individual escarpment or terrace features located significant distances away from modern watercourses. These features are important as they are the oldest landforms and therefore the most likely to contain sites of significant antiquity. Currently, the model makes allowance for these types of landforms by identifying large glacial meltwater channels where these features are most likely to occur and increasing high potential buffer widths within those areas. This method is moderately successful and does incorporate many escarpment and terrace features. There are, however, some landforms located beyond these buffers. In order to incorporate these landforms without increasing high potential areas unnecessarily, a method needs to be devised that identifies these

features individually. We were unsuccessful at creating a solution to this problem, and this represents an area where the model can be improved in the future.

Small Streams

Another major problem with the 1998 model is the preponderance of small streams classified as fish-bearing and therefore buffered in the same manner as larger streams. Field observations demonstrate that many of those small streams are intermittent or virtually non-existent. This was seen as one of the areas where the model could be improved the most and large areas of unnecessary high potential could be removed. The problem stems partly from the quality of the digital data. Stream classifications are incomplete and in many cases small streams are classified as fish-bearing, by default, for management purposes without any empirical basis for that classification. During the 2007 gap analysis we observed that, although many sites are associated with S4 streams they are usually located in close proximity to another hydrological feature (e.g., a lake or wetland), and they are usually located within 50 m of the stream. This sets S4 streams apart from larger streams. For these reasons we separated S4 streams from the other fish-bearing streams, created appropriate buffer widths, and amended the logical statements of the model to account for the change. The logical statements were then further modified so that S4 streams were only classified as high potential and buffered under certain circumstances (e.g., within a certain distance of another feature). These changes have significantly reduced the amount of modeled high potential area without reducing the effectiveness of the model.

Pixelation

Because of the raster-based format of the model, potential is assigned to cells that measure 10 m by 10 m in dimension. When individual cells are high potential but not located adjacent to other high potential cells they show as isolated pixels. These pixels will be removed from the final model output prior to circulation.

Determining Attributes

The native file format of the 1998 model does allow a user to query individual high potential areas to determine the basis for the rating. Because of the way the model was reformatted prior to circulation, however, these features were disabled. The revised model will allow users to query individual high potential areas.

Model Updates

The current project is the first formal update and revision to the 1998 model. There have been some improvements in the map data available but, for the most part, the data sets used are

fairly similar. The biggest advance made since 1998 has been the accumulation of archaeological data (new site discoveries and survey coverage) that was summarized during the 2007 archaeological data inventory. The recently completed Predictive Ecosystem Mapping (PEM) dataset for Quesnel Forest District shows promise for incorporation into the model but scheduling limitations did not allow for it to be used during this project. It is important that the model be updated periodically as new data becomes available.

Potential Zones

The 1998 model creates four classes of archaeological potential: high, moderate-high, moderate, and low. As implemented by the District Manager, however, only the high potential layer was used for determining when archaeological assessments were conducted. Recently, the Archaeology Branch, has released new standards that require archaeological overview assessments (AOAs) to create a single high potential layer. The 1998 model has been updated and revised to meet the new specifications.

Negative Survey Results

During the 2007 archaeological data inventory all permitted archaeological assessments and corresponding survey areas were compiled into a GIS database. The archaeological data contained in that database was used as a critical source of information during this project. We did not find an easy way to integrate that data into the model but because it is in a GIS format it can easily be overlaid and cross-checked with the model output. We made extensive use of archaeological data during the current project.

Missed Wetlands

It was noted that the 1998 model does not assign high potential to some small wetland features that may deserve it. Logical statements were amended so that under certain circumstances some small wetlands were classified as high potential.

Model Review

The revised model is currently in draft form. Limited review using orthophotos has been conducted. More extensive review will be conducted prior to finalization.

4.0 MODELLING RESULTS

Kvamme's Gain Statistic

A simple measure of the effectiveness of archaeological potential models is Kvamme's Gain statistic (Brandt et al. 1992), which compares the percentage of land base occupied by a potential zone to the percentage of the archaeological site population that falls into that zone. An effective model defines a high potential zone that uses the least amount of area to contain the highest number of archaeological sites. The formula is as follows:

Kvamme's Gain = 1 – (%Area / %Sites)

The Kvamme's Gain for the high potential zone of the original 1998 AOA Model (with the original 1998 base map data but 2007 archaeological site data) would be 1 – (20.97 / 83.62) which is 0.75. This indicates that the original model is relatively effective at determining high archaeological potential. A score of 1 would be the ideal, but next to impossible to achieve, result. Low potential zones should have a negative score and the larger the negative score the more effective the model is at identifying low potential areas. It should be noted here that the 1998 AOA Model was not created as a one potential zone model. It defined four potential zones (high, moderate-high, moderate, and low), each with recommendations for varying degrees of archaeological assessment intensity. As implemented by the District Manager, however, only proposed forestry developments that overlap with the high potential zone require an archaeological assessment.

During the current project we updated the base map data and recreated the 1998 model. The recreated model run on the updated data set produced 637,096.6404 hectares of high potential and encapsulated 936 of 1327 known sites. This results in a Kvamme's Gain Statistic of 0.57, a relatively poor number. A main reason for the discrepancy is the stream classification layer from Ministry of Environment includes many more S4 streams than in the 1998 stream data. As mentioned previously this was one of the bigger challenges of the project.

After making the adjustments to the base mapping layers and logical statements in the model discussed earlier, we were able to make substantial improvements in terms of the number of sites included in the high potential. Our revised model produces 335,481.6432 hectares of high potential and encompasses 1192 of 1327 known sites. This results in a Kvamme's Gain Statistic of 0.82, a significant improvement over the original model. The revised model captures 19% more sites in 14% less area than the recreated model.

N

W E

S

Wells

Quesnel

Nazko

Legend

- Quesnel Forest District
- Archaeological Site
- Cultural Heritage Resource
- Heritage Trail
- Town/City
- Lake
- High Archaeological Potential-2008
- High Archaeological Potential-1998
- Park

0 5 10 20 30 40 50

Kilometers

Map 2: Revised GIS Model of Archaeological Potential

MATRIX Research Ltd

Map Created by Deanna Windsor April 14, 2008

5.0 REFERENCES CITED

Arcas Consulting Archeologists Ltd.
1998 *GIS Modelling of Archaeological Potential: Quesnel Forest District, 1998*. Report on file, Cariboo Forest Region, Williams Lake, B.C.

Archaeology Branch
2008 *Archaeological Overview Assessments as general Land Use Planning Tools*. Draft report. Archaeology Branch, Ministry of Small Business, Tourism, and Culture. Victoria, B.C.

2000 *British Columbia Archaeological Inventory Guidelines*. Archaeology Branch, Ministry of Small Business, Tourism, and Culture. Victoria, B.C.

1998 *British Columbia Archaeological Impact Assessment Guidelines*. Archaeology Branch, Ministry of Small Business, Tourism, and Culture. Victoria, B.C.

Brandt, R., B.J. Groenwoudt, K.L. Kvamme
1992, *An Experiment in Archaeological Site Location: Modeling in the Netherlands Using GIS Techniques*, "World Archaeology", 24, pp. 268-282.

Heffner, Ty
2007 Archaeological Data Inventory of the Quesnel Forest District, 2007. Matrix Research Ltd., Quesnel, B.C.

Heffner, Ty, *et al*.
2007 Archaeological Data Gap Analysis of the Quesnel Forest District, 2007. Matrix Research Ltd., Quesnel, B.C.

Province of British Columbia
1986 *Heritage Conservation Act*. Document Issued by the Ministry of Small Business, Tourism, and Culture, Victoria, B.C.

APPENDIX A

Stakeholder Workshop Summary

395 Johnston Avenue, Quesnel, B.C. V2J-3M6
Phone (250) 992-6623 Fax (250) 992-6653
Email ty@matrixr.ca or susie@matrixr.ca

Workshop Summary

2008 Update to the Quesnel Forest District GIS Model of Archaeological Potential Stakeholder Workshop 1

Held January 22, 2008, 8:30 am to 1:30 pm
Best Western Tower Inn, 500 Reid Street, Quesnel

Purpose of Workshop

Canadian Forest Products Ltd. is sponsoring a FIA-funded project to update the existing GIS Model of Archaeological Potential for the Quesnel Forest District. This project represents the culmination of a multi-year program that involved the compilation of archaeological site and survey information for all archaeological work that has been conducted to date within Quesnel Forest District. Last year the archaeological data inventory was completed and a gap analysis was conducted in which recommendations were made for updating the current model.

Important stakeholders were invited to the workshop to participate in the project in order to provide them with information on the project strategies and objectives and to obtain their feedback and suggestions to incorporate into the project.

Topics

Topics of discussion included the following:

- Background on the 1998 GIS Model of Archaeological Potential.
- Results of the recently completed Archaeological Data Inventory and Archaeological Data Gap Analysis.
- New GIS based information (e.g., PEM and ecogroupings) that is now available for incorporation into the model.
- Proposed strategies for updating the GIS model.
- Stakeholder feedback and suggestions.
- Discussion.

Agenda

- 8:45-9:00 Welcome and Introductions
- 9:00-9:15 Project Objectives (Canfor's FIA Program, Management)
- 9:15-9:30 Background (AOAs, MOU, modelling, etc.)
- 9:30-10:00 Current GIS Model Outline
- 10:00-10:30 Archaeological Data Inventory
- 10:30-10:45 Archaeological Data Gap Analysis
- 10:45-11:00 Coffee
- 11:00-11:30 Revising AOA Model (strategy, methods, theory)
- 11:30-12:00 New Digital Data Available
- 12:00-12:30 Lunch
- 12:30-1:00 Stakeholder Feedback, Suggestions
- 1:00-1:30 Discussion
- 1:30 Workshop closing statements

Presentation materials are included in the powerpoint files that accompany this summary.

395 Johnston Avenue, Quesnel, B.C. V2J-3M6
Phone (250) 992-6623 Fax (250) 992-6653
Email ty@matrixr.ca or susie@matrixr.ca

Attendance
The following individuals were in attendance:

NAME	AFFILIATION
Deanna Windsor	Matrix Research Ltd.
Lisa Lefebvre	MoFR, Quesnel District
Kurtis Jacques	B.C.T.S.
Geronimo Squinas	Red Bluff Indian Band
Stefan Schwartzmann	MoFR, Quesnel District
Laurell Crocker	Nazko First Nation
Alicia Hjorth	Nazko First Nation
Mike Sakakibara	West Fraser Mills, Ltd.
Rob Ballinger	West Fraser Mills, Ltd.
Hesheng Zhang	Spatial Mapping, Ltd.
Steven Monai	Spatial Mapping, Ltd.
Steven Day	Canfor-Quesnel
Ty Heffner	Matrix Research, Ltd.
Bernice Cremo	Nazko First Nation

Comments
Comments received at the end of the meeting:

Lisa Lefebvre:
- There should be changes made to update the TRIM maps used in the model, ie. Wetland layer.

Stefan Schwartzmann:
-Sites found in low potential could become a variable in the High potential model.
-Is there special training for field technicians, layout personnel, etc to look for sites in low potential areas?

Rob Ballinger:
-Revising the model too often (yearly) would have repercussions on planning.
-The model will improve as new and better data is added-ie. terrain.

General:
-Inventory data should be maintained but model revisions should not be yearly.
-Length of time between revisions to the model in future should be discussed at the March workshop.

Steven Day:
-PEM Presentation.
-PEM data could be important in the modeling process, ie. resource gathering areas.

Laurell Crocker:
-The Potential Model can broaden the use of the Traditional Use Study.

Mike Sakakibara:
-Good that you are rerunning the existing model.
-Good to spend FIA money where it is needed.
-It is beneficial to First Nations to have hard data to use for treaty negotiations.

395 Johnston Avenue, Quesnel, B.C. V2J-3M6
Phone (250) 992-6623 Fax (250) 992-6653
Email ty@matrixr.ca or susie@matrixr.ca

Kurtis Jacques:

-There is a concern about inadequate TRIM data.

Lisa Lefebvre:

-There is a need to use the newest and best data available.

Spatial Mapping:

-Will examine the existing model.
-Think that new data will improve the output.
-Will examine and fix pixelation problems.
-Some TFLs do not have updated Forest Cover-some gaps?

General:

-GIS ArcView format is general to all.
-1998 Model depended on the data available at the time.
-Should be better Slope data in PEM.
-More wetland classification now.
-PEM has reclassified BGCZones- to predict species habitation.

For Further Information

Feel free to contact Ty Heffner (Phone 250-992-6623, Fax 250-992-6653, or Email ty@matrixr.ca) with any questions about the project or stakeholder workshop. Copies of the Quesnel Forest District Archaeological Data Inventory and Archaeological Data Gap Analysis are available, by request, in PDF format. In most cases, stakeholder groups will already have copies on file.

A second workshop is planned in early March to review the updated model and obtain additional feedback prior to finalization. Invitations will be sent to everyone on the current mailing list.

2008 Update to the Quesnel Forest District GIS Model of Archaeological Potential

Stakeholder Workshop 1

Workshop Schedule

- 8:45-9:00 Welcome and Introductions
- 9:00-9:15 Project Objectives (Canfor's FIA Program, Management)
- 9:15-9:30 Background (AOAs, MOU, modelling, etc.)
- 9:30-10:00 Current GIS Model Outline
- 10:00-10:30 Archaeological Data Inventory
- 10:30-10:45 Archaeological Data Gap Analysis
- 10:45-11:00 Coffee
- 11:00-11:30 Revising AOA Model (strategy, methods, theory)
- 11:30-12:00 New Digital Data Available
- 12:00-1:00 Lunch
- 1:00-1:30 Stakeholder Feedback, Suggestions
- 1:30- Discussion
- Workshop closing statements

Definitions

- Archaeological Site
  - Physical evidence of human activity pre-1846
- Traditional Use Site
  - Locations of traditional aboriginal activities post-1846
- Historic Site
  - Locations of historic events, structures, etc.
- Heritage Trail
  - Precontact or historic travel route
- Cultural Heritage Resources
  - Collective designation

Background

- 1990's:Five Year Development Plan Reviews
  - Site Specific AOAs
- 1996: Memorandum of Understanding
  - Archaeology Branch and MoF (MoF determines when Arch assessments are required, Arch Branch maintains control of site database and permitting).
- 1997: Cultural Heritage Overview of Cariboo Forest Region
  - Environmental zones and site type variability
- 1998: GIS Model of Archaeological Potential
  - MoF contracts to consultants for a model
- 2002-2004: Regional AOAs to Address Model Shortcomings
  - West Fraser TFL 52, Weldwood Operating Areas, Tolko Bark Beetle Area D
- 2004-2007: Archaeological Data Inventory and Data Gap Analyses
  - Beginning of current process to update GIS Model

Current GIS Model - Summary

- Arcas in co-operation with Carrier Chilcotin Tribal Council
- First GIS model of archaeological potential for the region
- Classes of archaeological potential
  - High, Moderate-High, Moderate and Low
  - Recommended management actions for each class
  - District Manager implementation of the model

Current GIS Model - Research

- First Nation Consultation
  - Meetings, information sharing, protocols
- Ethnographic Research
  - Traditional activity table
- Archaeological Research
  - Archaeological Site Database
  - Previous archaeological surveys for forestry
  - Targeted archaeological inventories

Current GIS Model – Research, cont'd

- Community-based archaeological research
  - Planned interviews
- Trail Research
  - Archival maps
- Biophysical Research
  - Biogeoclimatic zones
- Palaeoenvironmental Research
  - Glacial history, climate and vegetation changes

Definitions

- Archaeological Overview Assessment (AOA)
  - Office review to determine potential, need for and scope of archaeological investigation
- Archaeological Inventory Study (AIS)
  - Field studies to locate archaeological sites in a study area
- Preliminary Field Reconnaissance (PFR)
  - Initial survey to collect data, determine potential and need for further work
- Archaeological Impact Assessment (AIA)
  - Detailed survey to identify all archaeological sites within a proposed development area and assess potential impacts

Current GIS Model - Structure

- Deductive approach
  - Traditional activities, resulting site types, favourable environmental variables (like Alexander)
  - Raster (cell) format assigning values based on likelihood of supporting traditional activities
- Map Output
  - Variable width buffers around high potential features (lakes, streams, wetlands, etc.)
- Application
  - Archaeological Assessments conducted on areas that overlap high potential buffers.

2007 Archaeological Data Inventory - History

- Weldwood 2004
- Canfor and Nazko 2005
- Canfor and Nazko 2006
- Canfor 2007

- PDF Map

2007 Archaeological Data Inventory - Approach

- Archaeological Data Review
  - Provincial site database
  - Permit reports, non-permit reports
  - Archives
- Data Compilation
  - Previously assessed areas digitized as polygons linked to attribute tables containing associated data
  - Previously recorded archaeological and cultural heritage sites digitized as polygons and heritage trails digitized as lines linked to attribute tables containing associated data

2007 Archaeological Data Inventory - Results

- Four GIS Coverages
  - Assessment Areas = 1320 (6.5% of QFD)
  - Archaeological Sites (pre-1846) = 1262
  - Cultural Heritage Sites (post-1846) = 414
  - Heritage Trails = 121
- Data Tour

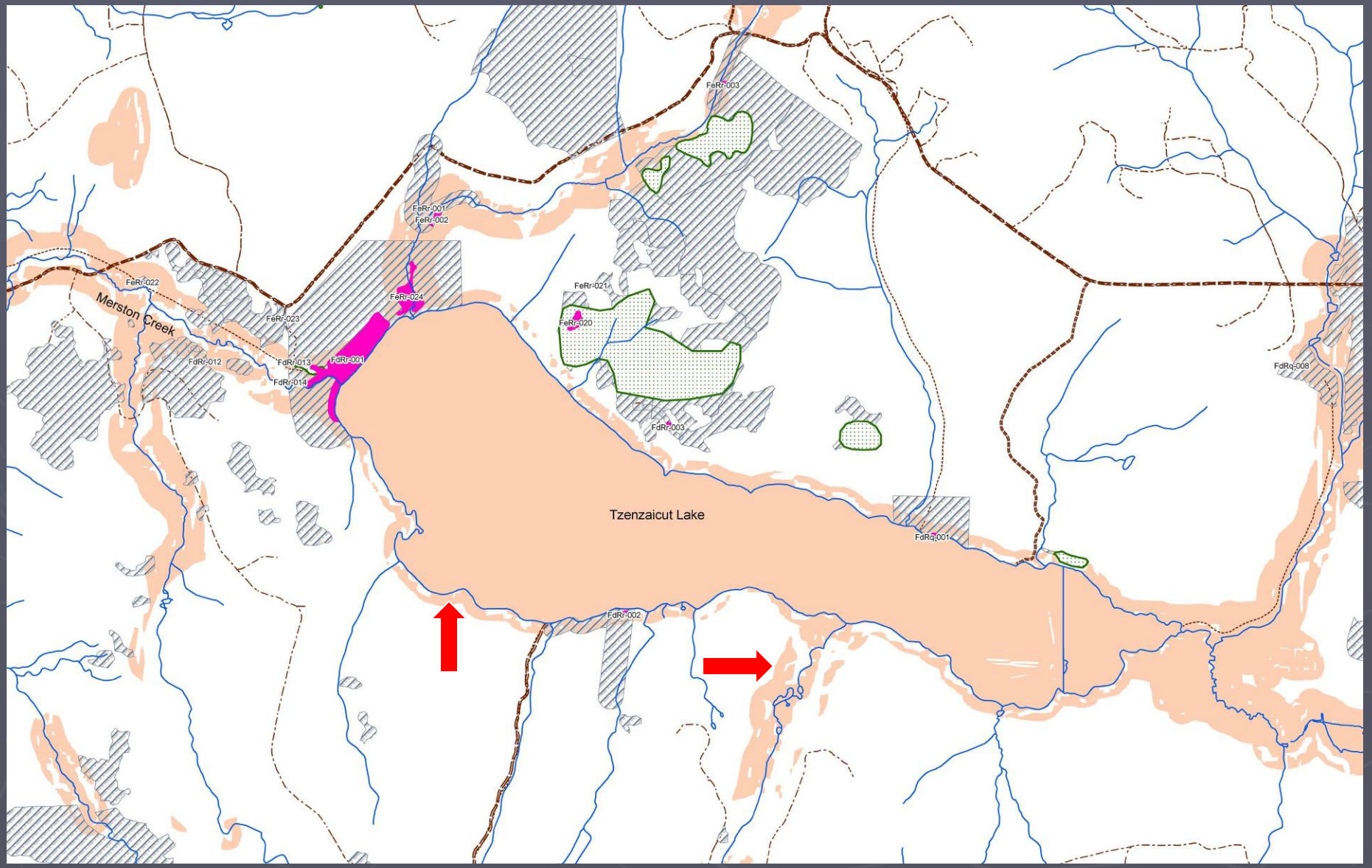
2007 Archaeological Data Inventory - Benefits

- Archaeological Research
  - Cultural resource management
  - Academic interests
- Development Planning
  - Tracking assessment investments
  - Resource management
- First Nations Interests
  - Traditional use studies
  - Development referrals
  - Treaty and land use planning

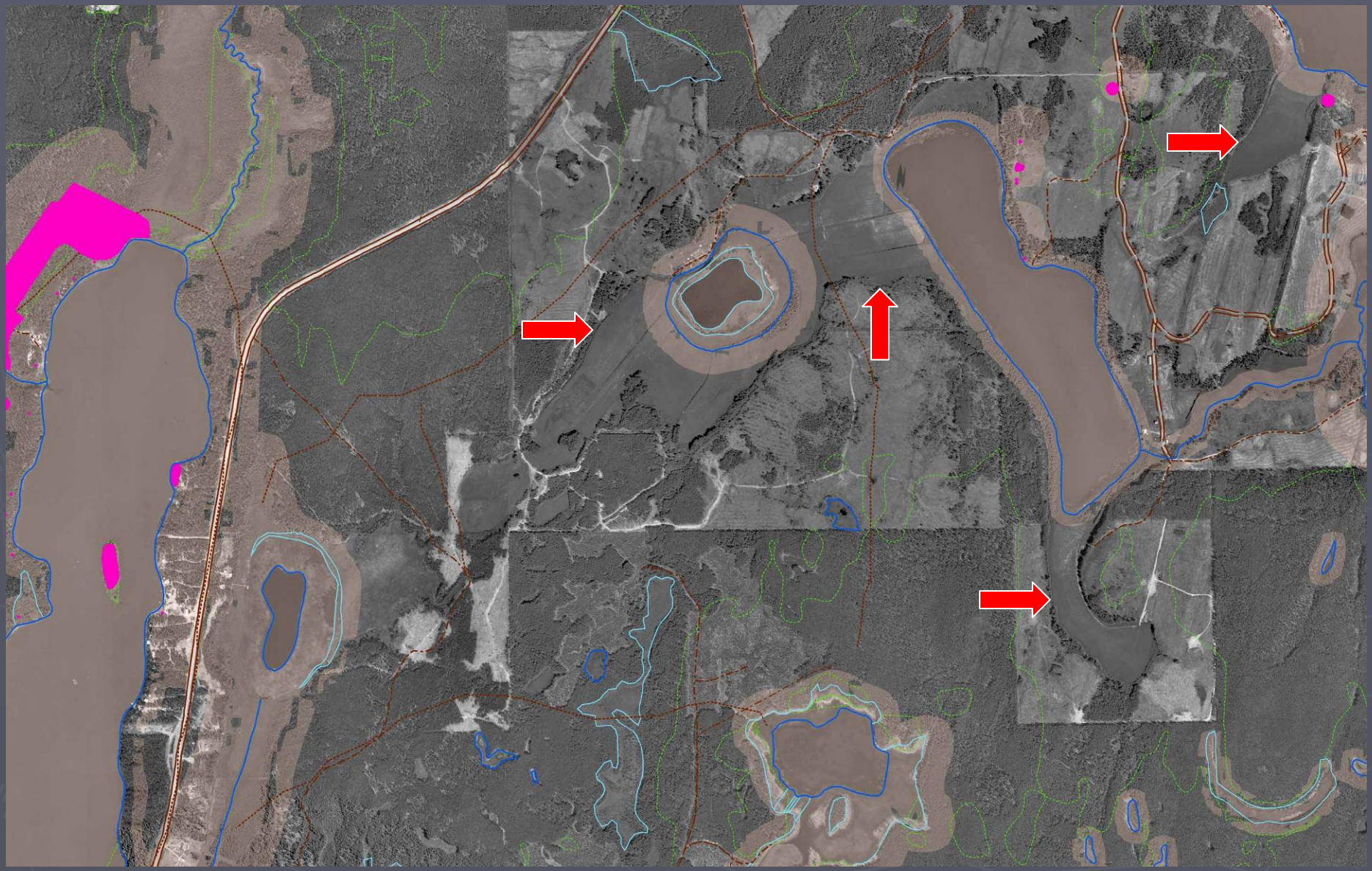
2007 Archaeological Data Gap Analysis - Objectives

- Characterize existing archaeological site inventory
- Review effectiveness of current AOA Model
- Identify gaps in current archaeological record that would affect future modelling efforts
- Tour report, tables

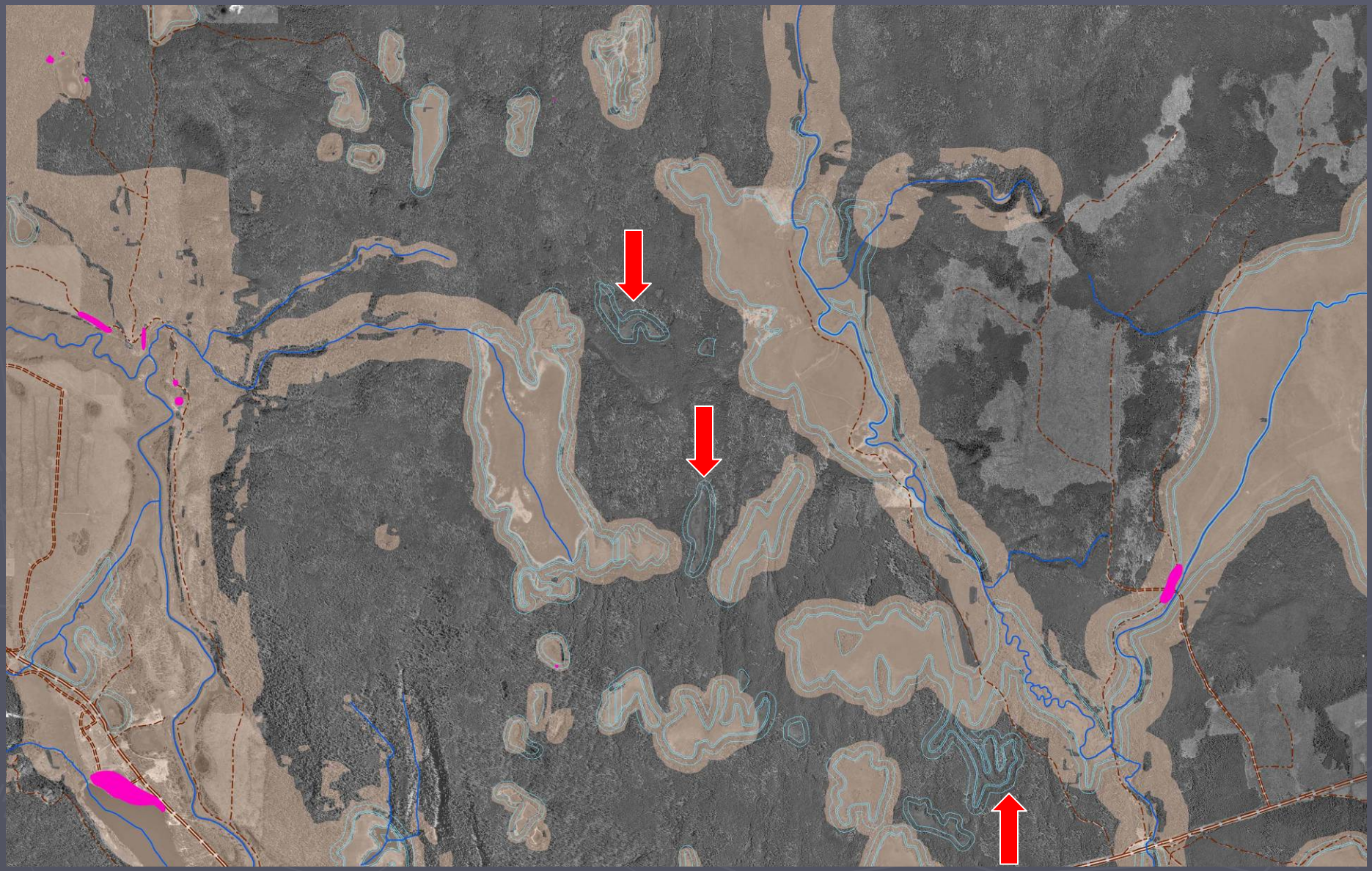
Misfit Buffer Widths: Streams vs. Lakes



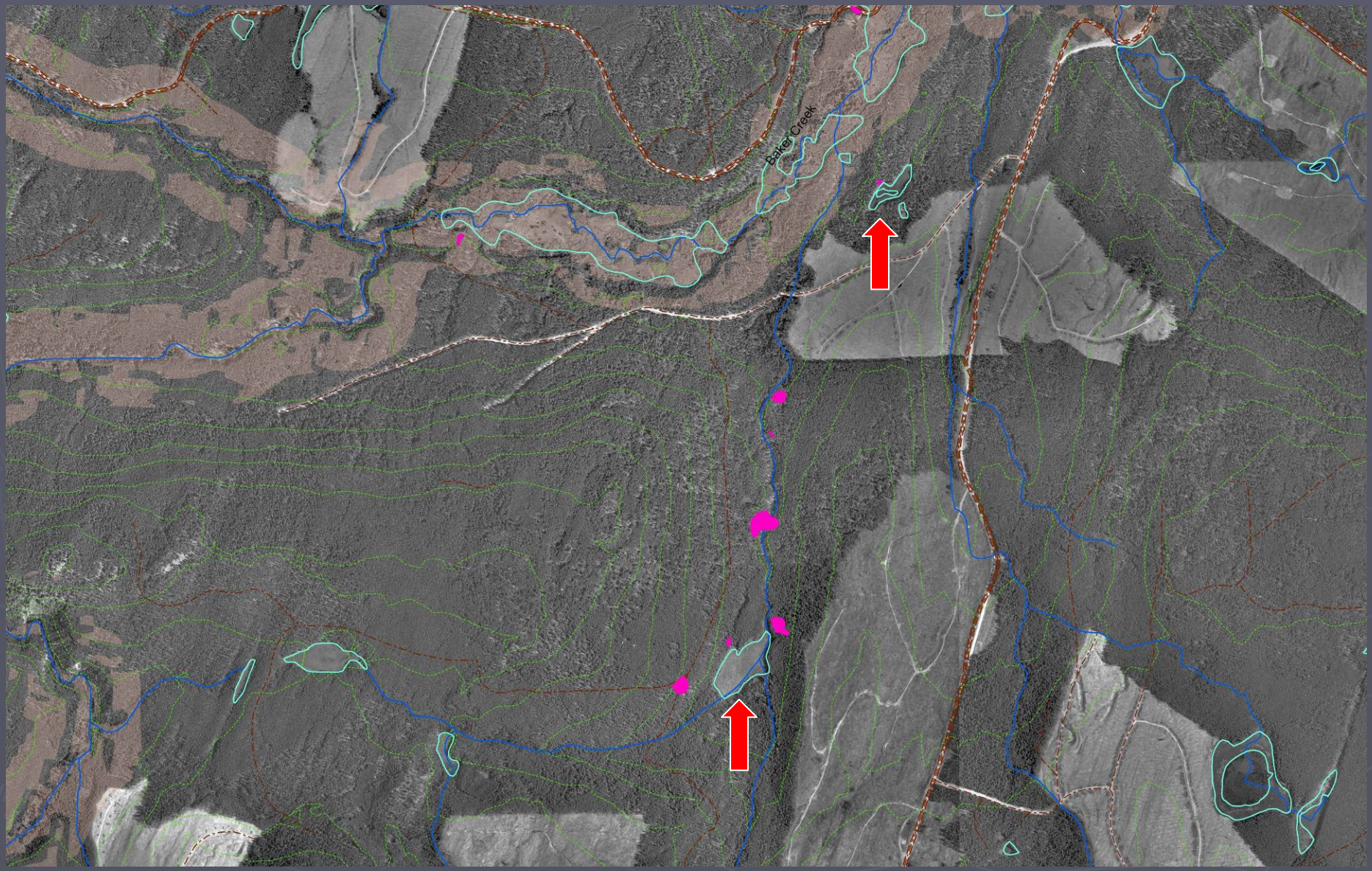
Missed Wetland Margins



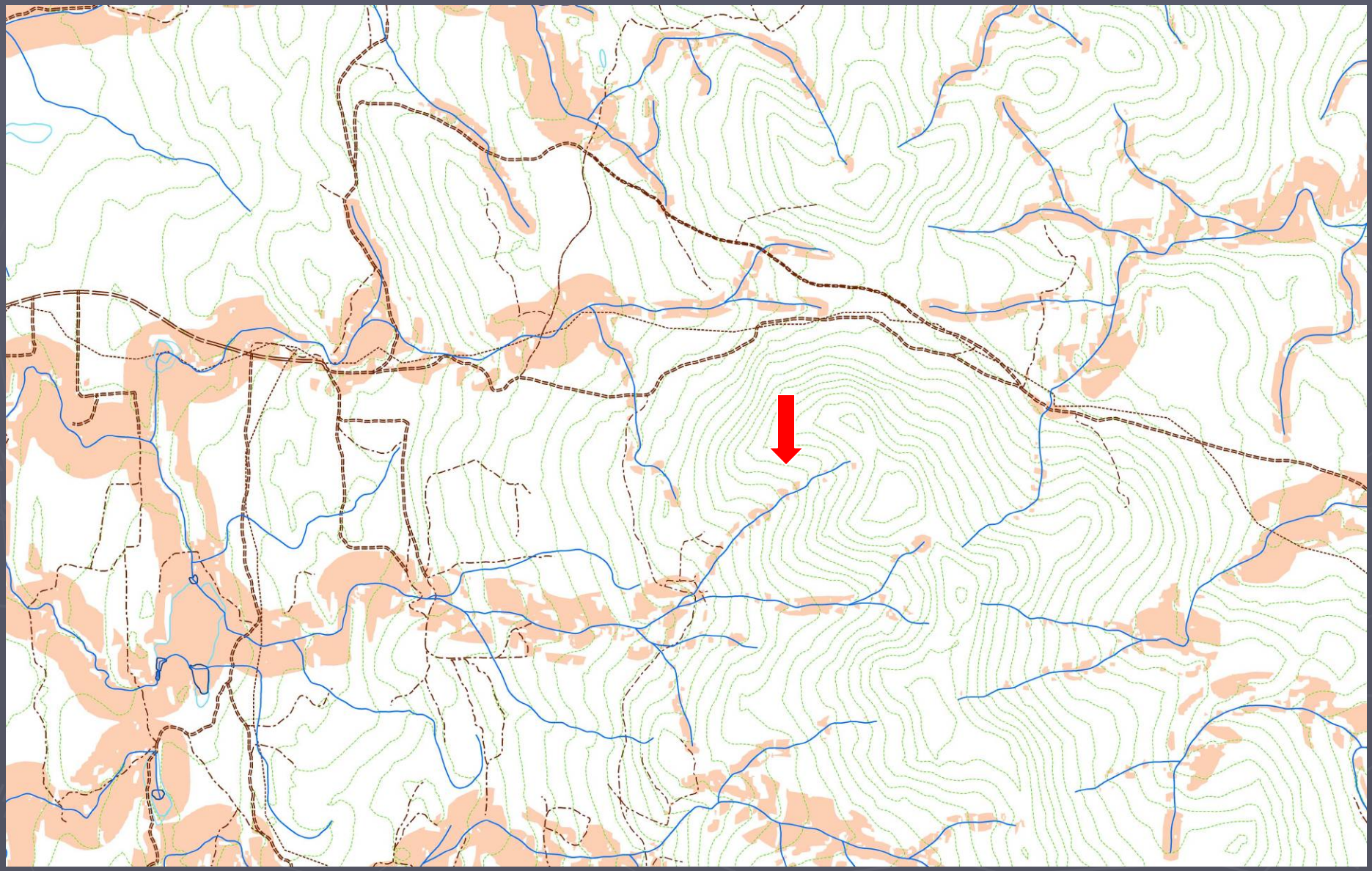
Small Wetlands Missed



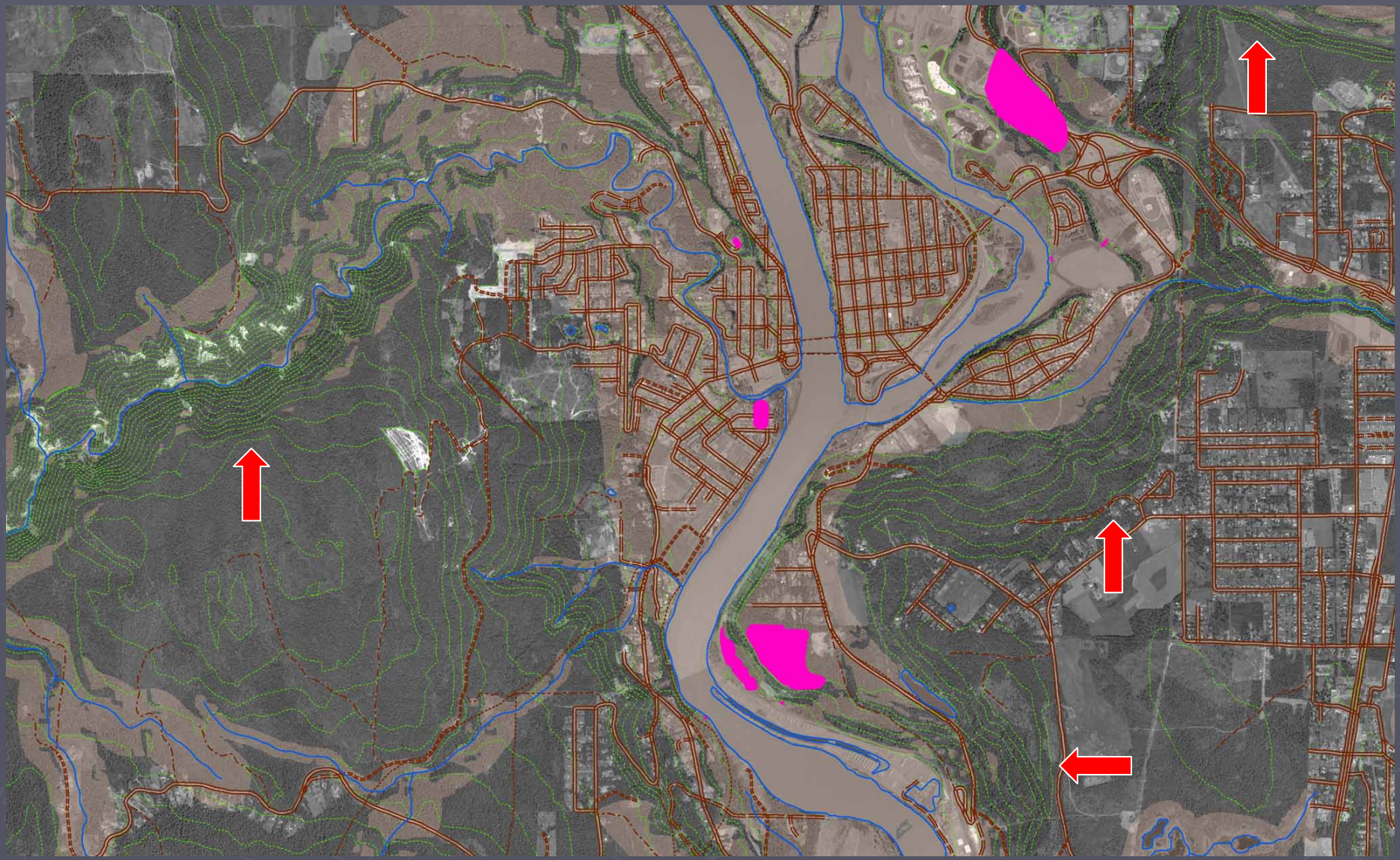
Small Wetlands Missed



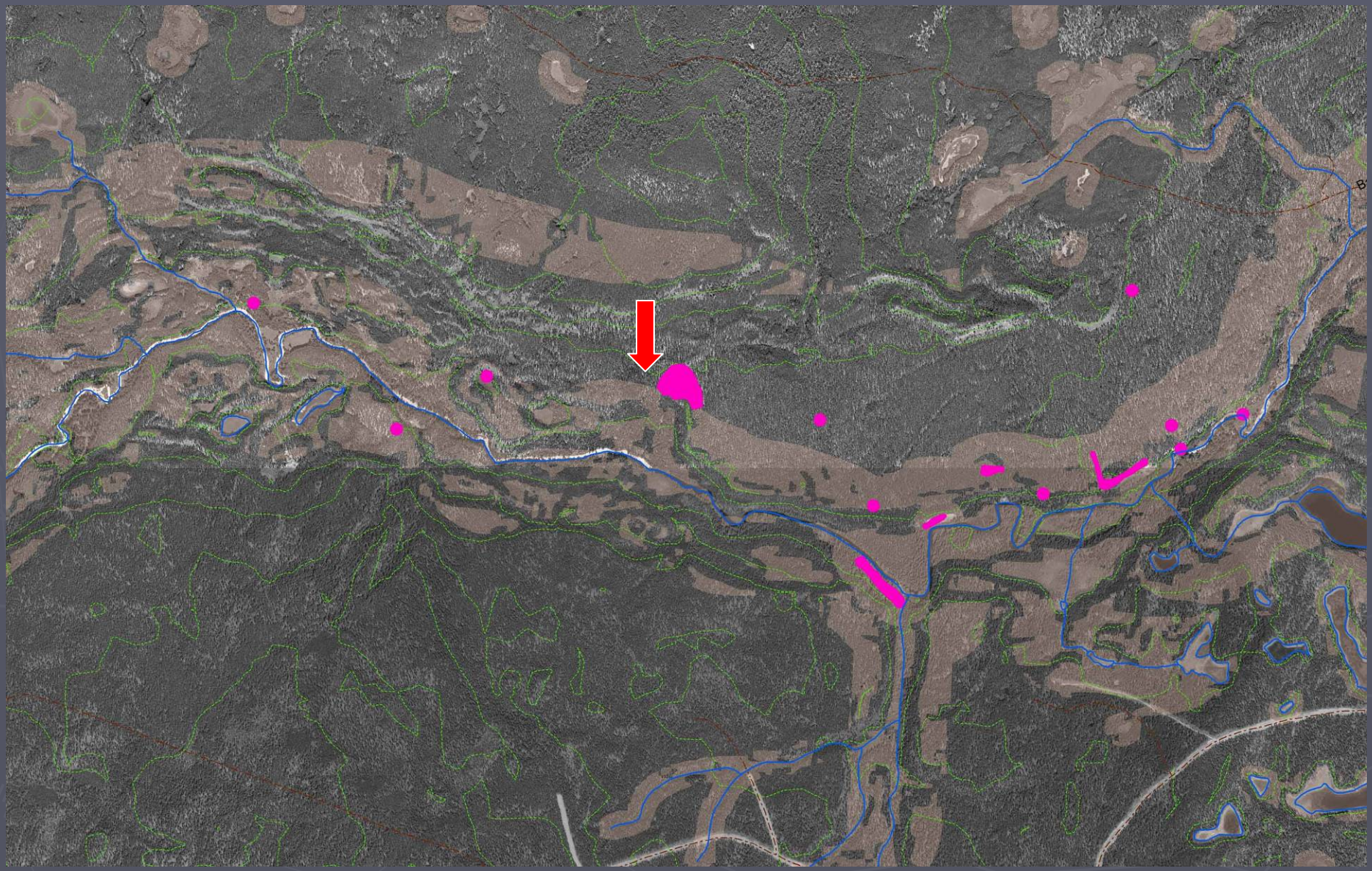
Small Misclassified Streams, Pixelation



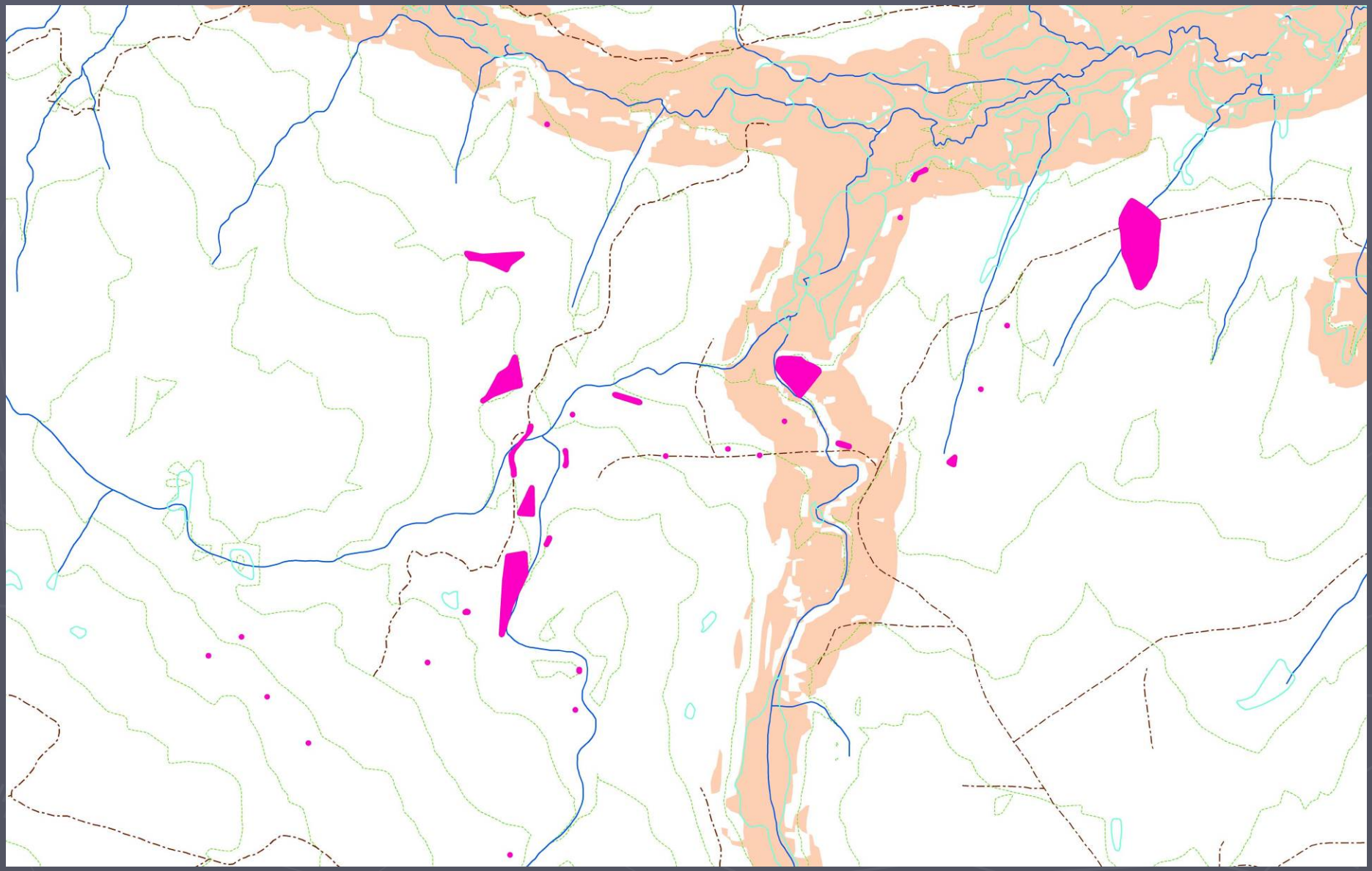
Escarpments Above Streams Missed



Escarpments Above Streams Missed



Unknown Variables: Lithic Sources



Revising AOA Model-Standards

- Archaeology Branch AOA Standards
  - Results-based, operational, effective, efficient
    - Single high potential polygon
  - Model major site types separately
    - Permanent habitation, temporary habitation or subsistence, human remains, wet sites, quarry sites, rock art sites, trails, pre-1846 CMTs
  - Kvamme gain statistic: 1 - (% area / % known sites)
    - Must achieve gain of 0.90 for major site types
    - Must achieve gain of 0.85 for entire model
  - Standardized output of GIS data
    - Available on RAAD website

Revising AOA Model-Approach

- Recreate 1998 GIS Potential Model
- Rerun model using updated versions of original mapping information
- Add new mapping information and query layers for correlation
- Iterative approach
- Draft model to review at stakeholder workshop
- Revise and finalize new model
- Implementation

APPENDIX B

Feature Definitions, Sources, Code and Buffers

Table ?: Quesnel Forest District AOAs Feature Definitions, Sources, Codes and Buffers							
Feature Category	**Feature Class**	**Definition**	**FP Code Equivalent**	**Source GIS Coverage**	**Feature Code**	**Buffer/ Polygon Code**	**Buffer/Polygon Criteria**
Streams* (fish values, based on gradients/ magnitudes)	Salmon	All current salmon-bearing double and single line streams (as per FISS vector coverage) plus all tributaries (s1-s4, 0-8% grade) to these streams.	S1 to S4 fish streams with high anadromous salmon probability	Shapefiles from West Fraser and Canfor	streamsalm	1	1st buffer=0-50 m
						2	2nd buffer=50-300 m
						3	3rd buffer=300-1000 m
						4	4th buffer=1000-2500 m
	High Fish Values	All current fish-bearing streams double and single line streams, excluding salmon stream network(high fish values)	stream classes S1 to S3 except salmon streams		streamfish	1	1st buffer=0-50 m
						2	2nd buffer=50-200 m
						3	3rd buffer=200-500 m
						4	4th buffer=500-700 m
	Low/ No Fish Values	Streams excluding salmon-bearing, fish-bearing streams and S4 (non salmon-bearing) streams	All other streams (S5, S6, NCD, S7).		streamothr	1	1st buffer=0-50 m
						2	2nd buffer=50-300 m
	s4 streams	S4 streams excluding salmon-bearing streams	S4 streams excluding salmon-bearing S4 streams		streamS4	1	1st buffer=0-50 m
						2	2nd buffer=50-100 m
						3	3rd buffer=100-300 m
Lakes** (excludes indefinite / intermittent lakes, reservoirs)	Large	>250 ha (high resource value)	L1 lakes > 250 ha	CWB_Lakes from LRDW	lakelarge	1	1st buffer=0-50 m
						2	2nd buffer=50-200 m
						3	3rd buffer=200-500 m
						4	4th buffer=500-700 m
	Medium	100-250 ha (moderate-high resource value)	L1 lakes between 100 and 250 ha		lakemed	1	1st buffer=0-50 m
						2	2nd buffer=50-100 m
						3	3rd buffer=100-300 m
						4	4th buffer=300-500 m
	Small	5-99.9 ha (moderate-low resource value)	L1 lakes <100ha		lakesmall	1	1st buffer=0-50 m
						2	2nd buffer=50-300 m
	Very Small	<5 ha (low resource value)	L2,L3,L4 lakes		lakevs	1	1st buffer=0-50 m

Table ?: Quesnel Forest District AOAs Feature Definitions, Sources, Codes and Buffers Continued							
Feature Category	**Feature Class**	**Definition**	**FP Code Equivalent**	**Source GIS Coverage**	**Feature Code**	**Buffer/ Polygon Code**	**Buffer/Polygon Criteria**
Wetland (includes indefinite, intermittent lakes, all meadows, all types of wetlands: swamps, marshes, etc.)	Large wetlands	>5 ha	W1, W5 wetlands	Shapefile from CANFOR	wetlarge	1	1st buffer=0-50 m
						2	2nd buffer=50-100 m
	Small wetlands	<5 ha	W2, W3, W4 wetlands		wetsmall	1	1st buffer=0-50 m
						2	2nd buffer=50-300 m
Slope	slope	slope (quintic)	n/a	DEM (ESRI lattice) from Canfor	slope	1	<15%
						2	15-30%
						3	30-45%
						4	45-60%
						5	>60%
Landforms (glacial and other)	Eskers	eskers (from TRIM)	n/a	TRIM	esker	1	buffer=0-50 m
	Large Glacial	major meltwater channels (terracing)	n/a	converted from raster files of 1998 model	meltlarge	1	polygon
	small Glacial	minor melwater channels (ravines), all three classes	n/a	converted from raster files of 1998 model	meltsmall	1	buffer=0-200 m
	Glacial Lakes	glacial lakes	n/a	converted from raster files of 1998 model	glacialake	1	polygon+buffer (0-100 m)
	Scarps	scarps and cliffs	n/a	TRIM	cliffs	1	buffer=0-100 m

Table ?: Quesnel Forest District AOAs Feature Definitions, Sources, Codes and Buffers Continued							
Feature Category	**Feature Class**	**Definition**	**FP Code Equivalent**	**Source GIS Coverage**	**Feature Code**	**Buffer/ Polygon Code**	**Buffer/Polygon Criteria**
Landforms (glacial and other)	Waterfalls	falls and cascades(lines and polygons)	n/a	TRIM	falls	1	buffer=0-100 m
	Rapids	rapids(lines and polygons)	n/a	TRIM	rapids	1	buffer=0-100 m
	Lithic Zone	lithic procurement area based on quarries, bedrock, and drainage	n/a	shapefile from Matrix	lithiczone	1	polygon
Ungulate	caribou core range	moderate and high caribou capability	n/a	shapefile from FES	ungulate	1	caribou polygon
	mule deer range	mule deer critical winter range	n/a	shapefile from FES	ungulate	2	mule deer polygon
VRI	old growth lodgepole	lodgepole pine stands of age class 8 or greater	PI 8+	VRI from LRDW and WestFraser	oldgrowth	1	PI 8+ polygon
	old growth cedar	cedar stands of age class 8 or greater	C 8+	VRI from LRDW and WestFraser	oldgrowth	2	C 8+ polygon
	whitebark pine	all stands with whitebark pine present (representing subalpine)	Pa	VRI from LRDW and WestFraser	whitebark	1	Pa polygon+buffer (0-300 m)
	open range	all open range polygons	OR	VRI from LRDW	openrange	1	openrange polygon + buffer (0-100 m)
Ecosection	BV	Bowron Valley		shapefile from LRDW	eco	1	BOV
	CB	Cariboo Basin				2	CAB
	CM	Cariboo Mountains				3	CAM
	CAP	Cariboo Plateau				4	CAP
	CCR	Central Chilcotin Ranges				5	CCR
	CHP	Chilcotin Plateau				6	CHP
	NU	Nazko Upland				8	NAU
	QH	Quesnel Highlands				10	QUH

Table ?: Quesnel Forest District AOAs Feature Definitions, Sources, Codes and Buffers Continued							
Feature Category	**Feature Class**	**Definition**	**FP Code Equivalent**	**Source GIS Coverage**	**Feature Code**	**Buffer/ Polygon Code**	**Buffer/Polygon Criteria**
Ecosection	QL	Quesnel Lowlands		shapefile from LRDW	eco	11	QUL
	WCU	Western Chilcotin Uplands				13	WCU
Biogeoclimatic Zone and Subzone	ATp	Alpine Tundra		shapefile from CDT Core Decision Technologies Inc	zone, subzone	1, 1	AT p polygon
	ATun					1, 1	AT un polygon
	ESSFwcp					1, 1	ESSF wcp polygon
	ESSFxvp					1, 1	ESSF xvp polygon
	IMA					1, 1	IMA polygon
	IDFxm	Interior Douglas Fir very dry mild				4, 4	IDFxm polygon
	IDFdk3	Interior Douglas Fir dry cool, Chicotin variant 3				4, 5	IDFdk3 polygon
	IDFdk4	Interior Douglas Fir dry cool, Chicotin variant 4				4, 5	IDFdk4 polygon
	ICHmk3	Interior Cedar-Hemlock moist cool				5, 7	ICHmk3
	ICHwk4	Interior Cedar-Hemlock wet cool varient 4				5, 9	ICHwk4 polygon
	ESSFmv1	Fenglemann Spruce-Subalpine Fir moist very cold variant 1				6, 10	ESSFmv1 polygon
	ESSFwc3	Englemann Spruce Subalpine Fir wet cold Cariboo variant				6, 11	ESSFwc3 polygon
	ESSF wcw					6, 11	ESSF wcw polygon
	ESSFwk1	Englemann Spruce Subalpine Fir wet cool Cariboo variant				6, 12	ESSFwk1 polygon
	ESSFxv	Englemann Spruce Subalpine Fir very dry very cold				6, 13	ESSFxv polygon
	ESSFxv1					6, 13	ESSF xv1 polygon
	MSxv	Montane Spruce very dry very cold				8, 15	MSxv polygon
	SBPSdc	Sub-Boreal Pine Spruce dry cold				9, 17	SBPSdc polygon
	SBSdk					9, 17	SBS polygon

Table ?: Quesnel Forest District AOAs Feature Definitions, Sources, Codes and Buffers Continued							
Feature Category	**Feature Class**	**Definition**	**FP Code Equivalent**	**Source GIS Coverage**	**Feature Code**	**Buffer/ Polygon Code**	**Buffer/Polygon Criteria**
Biogeoclimatic Zone and Subzone	SBPSmc	Sub-Boreal Pine Spruce moist cold		shapefile from CDT Core Decision Technologies Inc	zone, subzone	9, 18	SBPSmc polygon
	SBPSmk	Sub-Boreal Pine Spruce moist cool				9, 19	SBPSmk polygon
	SBPSxc	Sub-Boreal Pine Spruce very dry cold				9, 20	SBPSxc polygon
	SBSdw1	Sub-Boreal Spruce dry warm Horsefly variant				10, 21	SBSdw1
	SBSdw2	Sub-Boreal Spruce dry warm Blackwater variant				10, 22	SBSdw2 polygon
	SBSmc1	Sub-Boreal Spruce moist cold Babine variant1				10, 23	SBSmc1 polygon
	SBSmc2	Sub-Boreal Spruce moist cold Babine variant 2				10, 23	SBSmc2 polygon
	SBSmc3	Sub-Boreal Spruce moist cold Babine variant3				10, 23	SBSmc3 polygon
	SBSmh	Sub-Boreal Spruce moist hot				10, 24	SBSmh polygon
	IDF dw					10, 24	IDF dw polygon
	SBSmw	Sub-Boreal Spruce moist warm				10, 25	SBSmw polygon
	SBSwk1	Sub-Boreal Spruce wet cool Willow variant				10, 26	SBSwk1 polygon
Sites	Polygon sites	recorded archaeological sites >100 m in size	n/a	shapefile from Matrix	sitepply	1	polygon+buffer (0-100 m)
						2	buffer (100-250 m)
	Point sites	recorded archaeological sites <100 m in size	n/a		sitepoint	1	polygon+buffer (0-100 m)
						2	buffer(100-250 m)
Trails	definite	level of route confidence=good	n/a	shapefile from Matrix	trailgood	1	buffer=200 m
	probable	level of route confidence=moderate	n/a		trailmod	1	buffer=350 m
	possible	level of route confidence=poor	n/a		trailpoor	1	buffer=350 m

Note: *Streams are the combination of original streams and stream-associated wetlands. **Lakes are the combination of original lakes and lake-associated wetlands.

APPENDIX C

Revised Model Script
(on CD only)

```
class ArchPotModelError(StandardError):
    """Exception class for the ArchPotModel
    """
    pass


class AbstractSiteType:
    """Ancestor class of the archaeological site type classes.

    This abstract class is the ancestor class of all the archaeological
    site type classes.
    """
    def __init__(self, calculate_moderate_potentials=True):
        # Constant: Default arch potential value when no exprns match
        self.DEFAULT_POTENTIAL = 0

        # Constant: Error value for arch potential
        self.ERROR_POTENTIAL = -1

        # If False, then do not calculate potentials 1, 2, or 3.
        self.calculate_moderate_potentials = calculate_moderate_potentials

    def siteType(self):
        """Return the name of the Site Type implemented by the class.

        The default implementation returns the portion of the class
        name that follows the prefix "SiteType".
        """
        prefix = "SiteType"
        class_name = self.__class__.__name__
        prefix_idx = class_name.rindex(prefix)
        return class_name[prefix_idx + len(prefix):]

    def potential(self, data):
        """Return the arch. potential for the given data attributes.

        This method returns the archaeological potential of an area
        that is represented by the given attributes. The return value
        is an integer in [0, 1, 2, 3, 4], or -1 to indicate an error.

        If self.calculate_moderate_potentials is False, then the return
        value is an integer in [0, 4], or -1 to indicate an error.
        """
        raise ArchPotModelError(
            "This method must be overridden by a subclass")


class SiteType1(AbstractSiteType):
    """TYP1: Fishing Station - salmon
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.zone <> 1 and data.zone <> 6 and
                   data.subzone <> 15 ) and
                 ( data.streamsalm == 1 ) ):
                ret = 2

        if ( ( data.zone <> 1 and data.zone <> 6 and
               data.subzone <> 15 ) and
             ( data.streamsalm == 1 ) and
             ( data.lakelarge == 1 or data.lakemed == 1 ) ):
```

```
            ret = 4

        if ( ( data.zone <> 1 and data.zone <> 6 and
               data.subzone <> 15 ) and
             ( data.streamsalm == 1 and data.rapids == 1 ) ):
            ret = 4

        return ret


class SiteType2(AbstractSiteType):
    """TYP2: Cache - salmon
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.slope == 2 ) and
                 ( data.zone <> 1 and data.zone <> 6 and
                   data.subzone <> 15 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamsalm == 3 ) and
                 ( data.streamfish == 1 or data.streamfish == 2 or
                   data.streamothr == 1 or data.lakelarge == 1 or
                   data.lakelarge == 2 or data.lakemed == 1 or
                   data.lakemed == 2 or data.lakesmall == 1 or
                   data.wetlarge == 1 ) ):
                ret = 2

            if ( ( data.slope == 1 or data.slope == 2 ) and
                 ( data.zone <> 1 and data.zone <> 6 and
                   data.subzone <> 15 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamsalm == 3 or data.streamsalm == 4 ) ):
                ret = 2

            if ( ( data.slope == 1 or data.slope == 2 ) and
                 ( data.zone <> 1 and data.zone <> 6 and
                   data.subzone <> 15 ) and
                 ( data.streamsalm == 3 or data.streamsalm == 4 ) and
                 ( data.meltlarge == 1 or data.meltsmall == 1 ) ):
                ret = 3

            if ( ( data.slope == 2 ) and
                 ( data.zone <> 1 and data.zone <> 6 and
                   data.subzone <> 15 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamsalm == 3 or data.streamsalm == 4 ) and
                 ( data.streamfish == 1 or data.streamfish == 2 or
                   data.streamothr == 1 or data.lakelarge == 1 or
                   data.lakelarge == 2 or data.lakemed == 1 or
                   data.lakemed == 2 or data.lakesmall == 1 or
                   data.wetlarge == 1 or data.wetsmall == 1 ) ):
                ret = 3

            if ( ( data.slope == 2 ) and
                 ( data.zone <> 1 and data.zone <> 6 and
                   data.subzone <> 15 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 ) and
                 ( data.meltlarge == 1 ) ):
                ret = 3

            if ( ( data.slope == 1 ) and
```

```
                ( data.zone <> 1 and data.zone <> 6 and
                  data.subzone <> 15 ) and
                ( data.streamsalm == 3 ) ):
                ret = 3

        if ( ( data.slope == 1 ) and
             ( data.zone <> 1 and data.zone <> 6 and
               data.subzone <> 15 ) and
             ( data.streamsalm == 1 or data.streamsalm == 2 ) ):
            ret = 4

        if ( ( data.slope == 1 ) and
             ( data.zone <> 1 and data.zone <> 6 and
               data.subzone <> 15 ) and
             ( data.streamsalm == 1 or data.streamsalm == 2 or
               data.streamsalm == 3 ) and
             ( data.streamfish == 1 or data.streamfish == 2 or
               data.streamS4 == 1 or
               data.lakelarge == 1 or
               data.lakelarge == 2 or data.lakemed == 1 or
               data.lakemed == 2 or data.lakesmall == 1 or
               data.wetlarge == 1 ) ):
            ret = 4

        if ( ( data.slope == 1 or data.slope == 2 ) and
             ( data.zone <> 1 and data.zone <> 6 and
               data.subzone <> 15 ) and
             ( data.streamsalm == 1 or data.streamsalm == 2 ) and
             ( data.meltlarge == 1 ) ):
            ret = 4

        return ret


class SiteType3(AbstractSiteType):
    """TYP3: Base camp - salmon
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.slope == 2 ) and
                 ( data.zone <> 1 and data.zone <> 6 and
                   data.subzone <> 15 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamsalm == 3 ) and
                 ( data.streamfish == 1 or data.streamfish == 2 or
                   data.streamothr == 1 or data.lakelarge == 1 or
                   data.lakelarge == 2 or data.lakemed == 1 or
                   data.lakemed == 2 or data.lakesmall == 1 or
                   data.wetlarge == 1 ) ):
                ret = 2

            if ( ( data.slope == 2 ) and
                 ( data.zone <> 1 and data.zone <> 6 and
                   data.subzone <> 15 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 ) and
                 ( data.meltlarge == 1 ) ):
                ret = 3

            if ( ( data.slope == 1 ) and
                 ( data.zone <> 1 and data.zone <> 6 and
                   data.subzone <> 15 ) and
```

```
             ( data.streamsalm == 3 ) ):
            ret = 3

        if ( ( data.slope == 1 ) and
             ( data.zone <> 1 and data.zone <> 6 and
               data.subzone <> 15 ) and
             ( data.streamsalm == 1 or data.streamsalm == 2 ) ):
            ret = 4

        if ( ( data.slope == 1 ) and
             ( data.zone <> 1 and data.zone <> 6 and
               data.subzone <> 15 ) and
             ( data.streamsalm == 1 or data.streamsalm == 2 or
               data.streamsalm == 3 ) and
             ( data.streamfish == 1 or data.streamfish == 2 or
               data.streamS4 == 1 or
               data.lakelarge == 1 or
               data.lakelarge == 2 or data.lakemed == 1 or
               data.lakemed == 2 or data.lakesmall == 1 or
               data.wetlarge == 1 ) ):
            ret = 4

        return ret


class SiteType4(AbstractSiteType):
    """TYP4: Fishing Station - trout, sucker
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.zone <> 1 and data.zone <> 6 ) and
                 ( data.streamsalm == 1 or data.streamfish == 1 ) ):
                ret = 2

            if ( ( data.zone <> 1 and data.zone <> 6 ) and
                 ( data.streamsalm == 1 or data.streamfish == 1 ) and
                 ( data.streamothr == 1 ) ):
                ret = 3

        if ( ( data.zone <> 1 and data.zone <> 6 ) and
             ( data.streamsalm == 1 or data.streamfish == 1 or
               data.streamS4 == 1 ) and
             ( data.lakelarge == 1 or data.lakemed == 1 or
               data.lakesmall == 1 ) ):
            ret = 4

        if ( ( data.zone <> 1 and data.zone <> 6 ) and
             ( data.streamsalm == 1 ) and
             ( data.streamfish == 1 ) ):
            ret = 4

        return ret


class SiteType5(AbstractSiteType):
    """TYP5: Base Camp - trout, sucker, etc.
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
```

```
            if ( ( data.slope == 1 ) and
                 ( data.zone <> 1 and data.zone <> 6 ) and
                 ( data.lakelarge == 1 or data.lakelarge == 2 or
                   data.lakemed == 1 or data.lakemed == 2 or
                   data.lakesmall == 1 ) ):
                ret = 2

            if ( ( data.slope == 1 ) and
                 ( data.zone <> 1 and data.zone <> 6 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamfish == 1 or data.streamfish == 2 ) ):
                ret = 2

            if ( ( data.slope == 1 ) and
                 ( data.zone <> 1 and data.zone <> 6 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamsalm == 3 ) and
                 ( data.streamfish == 1 or data.streamfish == 2 or
                   data.streamfish == 3 ) and
                 ( data.lakelarge == 3 or data.lakemed == 3 or
                   data.lakesmall == 2 ) ):
                ret = 3

        if ( ( data.slope == 1 ) and
             ( data.zone <> 1 and data.zone <> 6 ) and
             ( data.streamsalm == 1 or data.streamsalm == 2 or
               data.streamsalm == 3 or data.streamfish == 1 or
               data.streamfish == 2 or data.streamfish == 3 or
               data.streamS4 == 1 ) and
              ( data.lakelarge == 1 or data.lakelarge == 2 or
               data.lakemed == 1 or data.lakemed == 2 or
               data.lakesmall == 1 or data.lakevs == 1 ) ):
            ret = 4

        return ret


class SiteType6(AbstractSiteType):
    """TYP6: Cache - trout, sucker, etc.
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.slope == 1 ) and
                 ( data.zone <> 1 and data.zone <> 6 ) and
                 ( data.lakelarge == 1 or data.lakelarge == 2 or
                   data.lakemed == 1 or data.lakemed == 2 or
                   data.lakesmall == 1 ) ):
                ret = 2

            if ( ( data.slope == 1 ) and
                 ( data.zone <> 1 and data.zone <> 6 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamfish == 1 or data.streamfish == 2 ) ):
                ret = 2

            if ( ( data.slope == 1 or data.slope == 2 ) and
                 ( data.zone <> 1 and data.zone <> 6 ) and
                 ( data.streamsalm == 3 or data.streamfish == 3 or
                   data.lakelarge == 3 or data.lakemed == 3 or
                   data.lakesmall == 2 ) ):
                ret = 2
```

```
            if ( ( data.slope == 1 or data.slope == 2 ) and
                 ( data.zone <> 1 and data.zone <> 6 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamfish == 1 or data.streamfish == 2 ) and
                 ( data.lakelarge == 1 or data.lakelarge == 2 or
                   data.lakemed == 1 or data.lakemed == 2 or
                   data.lakesmall == 1 ) ):
                ret = 3

            if ( ( data.slope == 1 ) and
                 ( data.zone <> 1 and data.zone <> 6 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamsalm == 3 ) and
                 ( data.streamfish == 1 or data.streamfish == 2 or
                   data.streamfish == 3 ) and
                 ( data.lakelarge == 3 or data.lakemed == 3 or
                   data.lakesmall == 2 ) ):
                ret = 3

        if ( ( data.slope == 1 ) and
             ( data.zone <> 1 and data.zone <> 6 ) and
             ( data.streamsalm == 1 or data.streamsalm == 2 or
               data.streamsalm == 3 or data.streamfish == 1 or
               data.streamfish == 2 or data.streamfish == 3 or
               data.streamS4 == 1 ) and
             ( data.lakelarge == 1 or data.lakelarge == 2 or
               data.lakemed == 1 or data.lakemed == 2 or
               data.lakesmall == 1 or data.lakevs == 1 ) ):
            ret = 4

        return ret


class SiteType7(AbstractSiteType):
    """TYP7: Fishing station - kokanee
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.slope == 3 or data.slope == 4 ) and
                 ( data.lakelarge == 1 or data.lakemed == 1 ) ):
                ret = 2

            if ( ( data.slope == 1 or data.slope == 2 ) and
                 ( data.lakelarge == 1 or data.lakemed == 1 ) ):
                ret = 3

        return ret


class SiteType8(AbstractSiteType):
    """TYP8: Base Camp - kokanee
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.slope == 1 ) and
                 ( data.lakelarge == 3 or data.lakemed == 3 ) ):
                ret = 3
```

```
        if ( ( data.slope == 1 or data.slope == 2 ) and
             ( data.lakelarge == 1 or data.lakelarge == 2 or
               data.lakemed == 1 or data.lakemed == 2 or
               data.lakesmall == 1 ) ):
            ret = 4

        return ret


class SiteType9(AbstractSiteType):
    """TYP9: Cache - kokanee
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.slope == 1 or data.slope == 2 ) and
                 ( data.lakelarge == 1 or data.lakelarge == 2 or
                   data.lakelarge == 3 or data.lakemed == 1 or
                   data.lakemed == 2 or data.lakemed == 3 ) ):
                ret = 2

            if ( ( data.slope == 1 ) and
                 ( data.lakelarge == 3 or data.lakemed == 3 ) ):
                ret = 3

        if ( ( data.slope == 1 or data.slope == 2 ) and
             ( data.lakelarge == 1 or data.lakelarge == 2 or
               data.lakemed == 1 or data.lakemed == 2 or
               data.lakesmall == 1 ) ):
            ret = 4

        return ret


class SiteType10(AbstractSiteType):
    """TYP10: Fishing Station - whitefish
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.slope == 3 ) and ( data.lakelarge == 1 ) ):
                ret = 2

            if ( ( data.slope == 1 or data.slope == 2 ) and
                 ( data.lakelarge == 1 ) ):
                ret = 3

        return ret


class SiteType11(AbstractSiteType):
    """TYP11: Base Camp - whitefish
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.slope == 1 ) and ( data.lakelarge == 3 ) ):
                ret = 3

        if ( ( data.slope == 1 ) and
```

```
             ( data.lakelarge == 1 or data.lakelarge == 2 ) ):
            ret = 4

        return ret


class SiteType12(AbstractSiteType):
    """TYP12: Cache - whitefish
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.slope == 3 ) and ( data.lakelarge == 1 ) ):
                ret = 2

            if ( ( data.slope == 1 or data.slope == 2 ) and
                 ( data.lakelarge == 1 or data.lakelarge == 2 or
                   data.lakelarge == 3 ) ):
                ret = 2

            if ( ( data.slope == 1 or data.slope == 2 ) and
                 ( data.lakelarge == 1 ) ):
                ret = 3

            if ( ( data.slope == 1 ) and ( data.lakelarge == 3 ) ):
                ret = 3

        if ( ( data.slope == 1 ) and
             ( data.lakelarge == 1 or data.lakelarge == 2 ) ):
            ret = 4

        return ret


class SiteType13(AbstractSiteType):
    """TYP13: Base Camp - hunting mule deer
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.slope == 2 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamfish == 1 or data.streamfish == 2 or
                   data.streamothr == 1 or data.lakelarge == 1 or
                   data.lakelarge == 2 or data.lakemed == 1 or
                   data.lakemed == 2 or data.wetlarge == 1 ) and
                 ( data.ungulate == 2 ) ):
                ret = 3

            if ( ( data.slope == 1 ) and
                 ( data.zone == 4 or data.subzone == 21 or
                   data.subzone == 22 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamfish == 1 or data.streamfish == 2 or
                   data.streamothr == 1 or data.lakelarge == 1 or
                   data.lakelarge == 2 or data.lakemed == 1 or
                   data.lakemed == 2 or data.wetlarge == 1 ) and
                 ( data.ungulate <> 2 ) ):
                ret = 3

            if ( ( data.slope == 2 ) and
```

```
                ( data.streamfish == 1 or data.streamfish == 2 or
                  data.streamothr == 1 or data.lakelarge == 1 or
                  data.lakelarge == 2 or data.lakemed == 1 or
                  data.lakemed == 2 or data.wetlarge == 1 ) and
                ( data.whitebark == 1 ) ):
                ret = 3

        if ( ( data.slope == 1 ) and
             ( data.streamsalm == 1 or data.streamsalm == 2 or
               data.streamfish == 1 or data.streamfish == 2 or
               data.streamS4 == 1 or
               data.lakelarge == 1 or
               data.lakelarge == 2 or data.lakemed == 1 or
               data.lakemed == 2 or data.wetlarge == 1 ) and
             ( data.ungulate == 2 ) ):
            ret = 4

        if ( ( data.slope == 1 ) and
             ( data.streamfish == 1 or data.streamfish == 2 or
               data.streamS4 == 1 or
               data.lakelarge == 1 or
               data.lakelarge == 2 or data.lakemed == 1 or
               data.lakemed == 2 or data.wetlarge == 1 ) and
             ( data.whitebark == 1 ) ):
            ret = 4

        return ret


class SiteType14(AbstractSiteType):
    """TYP14: Kill/butcher - hunting mule deer
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.zone == 2 or data.zone == 4 or
                   data.zone == 6 ) ):
                ret = 1

            if ( ( data.slope == 1 ) and
                 ( data.zone == 2 or data.zone == 4 ) and
                 ( data.openrange == 1 ) ):
                ret = 2

            if ( ( data.slope == 1 ) and ( data.ungulate == 2 ) ):
                ret = 2

            if ( ( data.slope == 1 ) and
                 ( data.zone == 4 or data.subzone == 21 or
                   data.subzone == 22 ) and
                 ( data.streamsalm == 3 or data.streamsalm == 4 or
                   data.streamfish == 3 or data.streamfish == 4 or
                   data.streamothr == 2 or data.lakelarge == 3 or
                   data.lakelarge == 4 or data.lakemed == 3 or
                   data.lakemed == 4 or data.lakesmall == 2 or
                   data.wetlarge == 2 ) and ( data.ungulate <> 2 ) ):
                ret = 2

            if ( ( data.slope == 1 or data.slope == 2 ) and
                 ( data.streamsalm == 3 or data.streamsalm == 4 or
                   data.streamfish == 3 or data.streamfish == 4 or
                   data.streamothr == 2 or data.lakelarge == 3 or
```

```
          data.lakelarge == 4 or data.lakemed == 3 or
          data.lakemed == 4 or data.lakesmall == 2 or
          data.wetlarge == 2 ) and
        ( data.ungulate == 2 ) ):
      ret = 2

  if ( ( data.slope == 1 ) and
       ( data.streamsalm == 3 or data.streamfish == 3 or
         data.streamothr == 2 or data.lakelarge == 3 or
         data.lakemed == 3 or data.lakesmall == 2 or
         data.wetlarge == 2 ) and
       ( data.ungulate == 2 ) ):
      ret = 3

  if ( ( data.slope == 1 ) and
       ( data.zone == 2 or data.zone == 4 or
         data.zone == 6 ) and
       ( data.streamsalm == 3 or data.streamfish == 3 or
         data.streamothr == 2 or data.lakelarge == 3 or
         data.lakemed == 3 or data.lakesmall == 2 or
         data.lakevs == 2 or data.wetlarge == 2 or
         data.wetsmall == 1 ) and
       ( data.openrange == 1 ) ):
      ret = 3

  if ( ( data.slope == 2 ) and
       ( data.streamsalm == 1 or data.streamsalm == 2 or
         data.streamfish == 1 or data.streamfish == 2 or
         data.streamothr == 1 or data.lakelarge == 1 or
         data.lakelarge == 2 or data.lakemed == 1 or
         data.lakemed == 2 or data.wetlarge == 1 ) and
       ( data.ungulate == 2 ) ):
      ret = 3

  if ( ( data.slope == 1 ) and
       ( data.zone == 4 or data.subzone == 21 or
         data.subzone == 22 ) and
       ( data.streamsalm == 1 or data.streamsalm == 2 or
         data.streamfish == 1 or data.streamfish == 2 or
         data.streamothr == 1 or data.lakelarge == 1 or
         data.lakelarge == 2 or data.lakemed == 1 or
         data.lakemed == 2 or data.wetlarge == 1 ) and
       ( data.ungulate <> 2 ) ):
      ret = 3

  if ( ( data.slope == 2 ) and
       ( data.streamfish == 1 or data.streamfish == 2 or
         data.streamothr == 1 or data.lakelarge == 1 or
         data.lakelarge == 2 or data.lakemed == 1 or
         data.lakemed == 2 or data.wetlarge == 1 ) and
       ( data.whitebark == 1 ) ):
      ret = 3

if ( ( data.slope == 1 ) and
     ( data.zone == 2 or data.zone == 4 ) and
     ( data.streamsalm == 1 or data.streamsalm == 2 or
       data.streamfish == 1 or data.streamfish == 2 or
       data.streamS4 == 1 or
       data.lakelarge == 1 or
       data.lakelarge == 2 or data.lakemed == 1 or
       data.lakemed == 2 or data.lakesmall == 1 or
       data.lakevs == 1 or data.wetlarge == 1 ) and
     ( data.openrange == 1 ) ):
```

```
            ret = 4

        if ( ( data.slope == 1 ) and
             ( data.streamsalm == 1 or data.streamsalm == 2 or
               data.streamfish == 1 or data.streamfish == 2 or
               data.streamS4 == 1 or
               data.lakelarge == 1 or
               data.lakelarge == 2 or data.lakemed == 1 or
               data.lakemed == 2 or data.wetlarge == 1 ) and
             ( data.ungulate == 2 ) ):
            ret = 4

        if ( ( data.slope == 1 ) and
             ( data.streamfish == 1 or data.streamfish == 2 or
               data.streamS4 == 1 or
               data.lakelarge == 1 or
               data.lakelarge == 2 or data.lakemed == 1 or
               data.lakemed == 2 or data.wetlarge == 1 ) and
             ( data.whitebark == 1 ) ):
            ret = 4

        return ret


class SiteType15(AbstractSiteType):
    """TYP15: Game Overlook - hunting mule deer
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.slope == 1 ) and
                 ( data.zone == 2 or data.zone == 4 ) and
                 ( data.openrange == 1 ) ):
                ret = 2

            if ( ( data.slope == 1 ) and ( data.ungulate == 2 ) ):
                ret = 2

            if ( ( data.slope == 1 ) and
                 ( data.zone == 4 or data.subzone == 21 or
                   data.subzone == 22 ) and
                 ( data.streamsalm == 3 or data.streamsalm == 4 or
                   data.streamfish == 3 or data.streamfish == 4 or
                   data.streamothr == 2 or data.lakelarge == 3 or
                   data.lakelarge == 4 or data.lakemed == 3 or
                   data.lakemed == 4 or data.lakesmall == 2 or
                   data.wetlarge == 2 ) and ( data.ungulate <> 2 ) ):
                ret = 2

            if ( ( data.slope == 1 or data.slope == 2 ) and
                 ( data.streamsalm == 3 or data.streamsalm == 4 or
                   data.streamfish == 3 or data.streamfish == 4 or
                   data.streamothr == 2 or data.lakelarge == 3 or
                   data.lakelarge == 4 or data.lakemed == 3 or
                   data.lakemed == 4 or data.lakesmall == 2 or
                   data.wetlarge == 2 ) and
                 ( data.ungulate == 2 ) ):
                ret = 2

            if ( ( data.slope == 1 ) and
                 ( data.zone == 2 or data.zone == 4 or
                   data.zone == 6 ) and
```

```
        ( data.esker == 1 or data.meltsmall == 1 or
          data.glacialake == 1 or data.cliffs == 1 ) and
        ( data.ungulate <> 2 or data.whitebark <> 1 ) ):
      ret = 2

  if ( ( data.slope == 1 ) and ( data.meltsmall == 1 ) and
        ( data.ungulate == 2 or data.whitebark == 1 ) ):
      ret = 2

  if ( ( data.slope == 1 ) and
        ( data.esker == 1 or data.glacialake == 1 or
          data.cliffs == 1 ) and
        ( data.ungulate == 2 or data.whitebark == 1 ) ):
      ret = 3

  if ( ( data.slope == 1 ) and
        ( data.streamsalm == 3 or data.streamfish == 3 or
          data.streamothr == 2 or data.lakelarge == 3 or
          data.lakemed == 3 or data.lakesmall == 2 or
          data.wetlarge == 2 ) and
        ( data.ungulate == 2 ) ):
      ret = 3

  if ( ( data.slope == 1 ) and
        ( data.zone == 2 or data.zone == 4 or
          data.zone == 6 ) and
        ( data.streamsalm == 3 or data.streamfish == 3 or
          data.streamothr == 2 or data.lakelarge == 3 or
          data.lakemed == 3 or data.lakesmall == 2 or
          data.lakevs == 2 or data.wetlarge == 2 or
          data.wetsmall == 1 ) and
        ( data.openrange == 1 ) ):
      ret = 3

  if ( ( data.slope == 2 ) and
        ( data.streamsalm == 1 or data.streamsalm == 2 or
          data.streamfish == 1 or data.streamfish == 2 or
          data.streamothr == 1 or data.lakelarge == 1 or
          data.lakelarge == 2 or data.lakemed == 1 or
          data.lakemed == 2 or data.wetlarge == 1 ) and
        ( data.ungulate == 2 ) ):
      ret = 3

  if ( ( data.slope == 1 ) and
        ( data.zone == 4 or data.subzone == 21 or
          data.subzone == 22 ) and
        ( data.streamsalm == 1 or data.streamsalm == 2 or
          data.streamfish == 1 or data.streamfish == 2 or
          data.streamothr == 1 or data.lakelarge == 1 or
          data.lakelarge == 2 or data.lakemed == 1 or
          data.lakemed == 2 or data.wetlarge == 1 ) and
        ( data.ungulate <> 2 ) ):
      ret = 3

  if ( ( data.slope == 2 ) and
        ( data.streamfish == 1 or data.streamfish == 2 or
          data.streamothr == 1 or data.lakelarge == 1 or
          data.lakelarge == 2 or data.lakemed == 1 or
          data.lakemed == 2 or data.wetlarge == 1 ) and
        ( data.whitebark == 1 ) ):
      ret = 3

if ( ( data.slope == 1 ) and
```

```
             ( data.zone == 2 or data.zone == 4 ) and
             ( data.streamsalm == 1 or data.streamsalm == 2 or
               data.streamfish == 1 or data.streamfish == 2 or
               data.streamS4 == 1 or
               data.lakelarge == 1 or
               data.lakelarge == 2 or data.lakemed == 1 or
               data.lakemed == 2 or data.lakesmall == 1 or
               data.lakevs == 1 or data.wetlarge == 1 ) and
             ( data.openrange == 1 ) ):
            ret = 4

        if ( ( data.slope == 1 ) and
             ( data.streamsalm == 1 or data.streamsalm == 2 or
               data.streamfish == 1 or data.streamfish == 2 or
               data.streamS4 == 1 or
               data.lakelarge == 1 or
               data.lakelarge == 2 or data.lakemed == 1 or
               data.lakemed == 2 or data.wetlarge == 1 ) and
             ( data.ungulate == 2 ) ):
            ret = 4

        if ( ( data.slope == 1 ) and
             ( data.streamfish == 1 or data.streamfish == 2 or
               data.streamS4 == 1 or
               data.lakelarge == 1 or
               data.lakelarge == 2 or data.lakemed == 1 or
               data.lakemed == 2 or data.wetlarge == 1 ) and
             ( data.whitebark == 1 ) ):
            ret = 4

        return ret


class SiteType16(AbstractSiteType):
    """TYP16: Hunting blind - hunting mule deer
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.slope == 1 or data.slope == 2 ) and
                 ( data.streamfish == 1 or data.streamfish == 2 or
                   data.streamfish == 3 or data.streamothr == 1 or
                   data.streamothr == 2 ) and
                 ( data.lakesmall == 1 or data.lakesmall == 2 or
                   data.lakevs == 1 ) and ( data.whitebark == 1 ) ):
                ret = 3

        return ret


class SiteType17(AbstractSiteType):
    """TYP17: Base Camp - hunting caribou
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.slope == 2 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamfish == 1 or data.streamfish == 2 or
                   data.lakelarge == 1 or data.lakelarge == 2 or
                   data.lakemed == 1 or data.lakemed == 2 or
```

```
                    data.lakesmall == 1 or data.wetlarge == 1 ) and
                  ( data.ungulate == 1 ) ):
                ret = 3

            if ( ( data.slope == 1 ) and
                 ( data.zone == 1 or data.subzone == 11 or
                   data.subzone == 13 or data.subzone == 15 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamfish == 1 or data.streamfish == 2 or
                   data.lakelarge == 1 or data.lakelarge == 2 or
                   data.lakemed == 1 or data.lakemed == 2 or
                   data.lakesmall == 1 or data.wetlarge == 1 ) ):
                ret = 3

        if ( ( data.slope == 1 ) and
             ( data.streamsalm == 1 or data.streamsalm == 2 or
               data.streamfish == 1 or data.streamfish == 2 or
               data.lakelarge == 1 or data.lakelarge == 2 or
               data.lakemed == 1 or data.lakemed == 2 or
               data.lakesmall == 1 or data.wetlarge == 1 ) and
             ( data.ungulate == 1 ) ):
            ret = 4

        return ret


class SiteType18(AbstractSiteType):
    """TYP18: Hunting blind - hunting caribou
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.slope == 1 ) and ( data.ungulate == 1 ) ):
                ret = 2

            if ( ( data.slope == 2 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamfish == 1 or data.streamfish == 2 or
                   data.lakelarge == 1 or data.lakelarge == 2 or
                   data.lakemed == 1 or data.lakemed == 2 or
                   data.lakesmall == 1 or data.wetlarge == 1 ) and
                 ( data.ungulate == 1 ) ):
                ret = 3

            if ( ( data.slope == 1 ) and
                 ( data.zone == 1 or data.subzone == 11 or
                   data.subzone == 13 or data.subzone == 15 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamfish == 1 or data.streamfish == 2 or
                   data.lakelarge == 1 or data.lakelarge == 2 or
                   data.lakemed == 1 or data.lakemed == 2 or
                   data.lakesmall == 1 or data.wetlarge == 1 ) ):
                ret = 3

            if ( ( data.slope == 1 ) and
                 ( data.streamsalm == 3 or data.streamsalm == 4 or
                   data.streamfish == 3 or data.streamfish == 4 or
                   data.streamothr == 1 or data.streamothr == 2 or
                   data.lakelarge == 3 or data.lakelarge == 4 or
                   data.lakemed == 3 or data.lakemed == 4 or
                   data.lakesmall == 2 or data.wetlarge == 2 or
                   data.wetsmall == 1 ) and ( data.ungulate == 1 ) ):
```

```
            ret = 3

        if ( ( data.slope == 1 ) and
             ( data.streamsalm == 1 or data.streamsalm == 2 or
               data.streamfish == 1 or
               data.lakelarge == 1 or data.lakelarge == 2 or
               data.lakemed == 1 or data.lakemed == 2 or
               data.lakesmall == 1 or data.wetlarge == 1 ) and
             ( data.ungulate == 1 ) ):
            ret = 4

        return ret


class SiteType19(AbstractSiteType):
    """TYP19: Kill/butcher - hunting caribou
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.zone == 1 or data.zone == 6 or data.zone == 8 ) ):
                ret = 1

            if ( ( data.slope == 1 ) and ( data.ungulate == 1 ) ):
                ret = 2

            if ( ( data.slope == 2 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamfish == 1 or data.streamfish == 2 or
                   data.lakelarge == 1 or data.lakelarge == 2 or
                   data.lakemed == 1 or data.lakemed == 2 or
                   data.lakesmall == 1 or data.wetlarge == 1 ) and
                 ( data.ungulate == 1 ) ):
                ret = 3

            if ( ( data.slope == 1 ) and
                 ( data.zone == 1 or data.subzone == 11 or
                   data.subzone == 13 or data.subzone == 15 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamfish == 1 or data.streamfish == 2 or
                   data.lakelarge == 1 or data.lakelarge == 2 or
                   data.lakemed == 1 or data.lakemed == 2 or
                   data.lakesmall == 1 or data.wetlarge == 1 ) ):
                ret = 3

            if ( ( data.slope == 1 ) and
                 ( data.streamsalm == 3 or data.streamsalm == 4 or
                   data.streamfish == 3 or data.streamfish == 4 or
                   data.streamothr == 1 or data.streamothr == 2 or
                   data.lakelarge == 3 or data.lakelarge == 4 or
                   data.lakemed == 3 or data.lakemed == 4 or
                   data.lakesmall == 2 or data.wetlarge == 2 or
                   data.wetsmall == 1 ) and ( data.ungulate == 1 ) ):
                ret = 3

        if ( ( data.slope == 1 ) and
             ( data.streamsalm == 1 or data.streamsalm == 2 or
               data.streamfish == 1 or
               data.lakelarge == 1 or data.lakelarge == 2 or
               data.lakemed == 1 or data.lakemed == 2 or
               data.lakesmall == 1 or data.wetlarge == 1 ) and
             ( data.ungulate == 1 ) ):
```

```
            ret = 4

        return ret


class SiteType20(AbstractSiteType):
    """TYP20: Cache - hunting caribou
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.slope == 1 ) and ( data.ungulate == 1 ) ):
                ret = 2

            if ( ( data.slope == 2 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamfish == 1 or data.streamfish == 2 or
                   data.lakelarge == 1 or data.lakelarge == 2 or
                   data.lakemed == 1 or data.lakemed == 2 or
                   data.lakesmall == 1 or data.wetlarge == 1 ) and
                 ( data.ungulate == 1 ) ):
                ret = 3

            if ( ( data.slope == 1 ) and
                 ( data.zone == 1 or data.subzone == 11 or
                   data.subzone == 13 or data.subzone == 15 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamfish == 1 or data.streamfish == 2 or
                   data.lakelarge == 1 or data.lakelarge == 2 or
                   data.lakemed == 1 or data.lakemed == 2 or
                   data.lakesmall == 1 or data.wetlarge == 1 ) ):
                ret = 3

            if ( ( data.slope == 1 ) and
                 ( data.streamsalm == 3 or data.streamsalm == 4 or
                   data.streamfish == 3 or data.streamfish == 4 or
                   data.streamothr == 1 or data.streamothr == 2 or
                   data.lakelarge == 3 or data.lakelarge == 4 or
                   data.lakemed == 3 or data.lakemed == 4 or
                   data.lakesmall == 2 or data.wetlarge == 2 or
                   data.wetsmall == 1 ) and ( data.ungulate == 1 ) ):
                ret = 3

        if ( ( data.slope == 1 ) and
             ( data.streamsalm == 1 or data.streamsalm == 2 or
               data.streamfish == 1 or
               data.lakelarge == 1 or data.lakelarge == 2 or
               data.lakemed == 1 or data.lakemed == 2 or
               data.lakesmall == 1 or data.wetlarge == 1 ) and
             ( data.ungulate == 1 ) ):
            ret = 4

        return ret


class SiteType21(AbstractSiteType):
    """TYP21: Base camp - hunting elk ( moose )
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
```

```
            if ( ( data.slope == 2 ) and
                 ( data.zone == 4 or data.zone == 5 or
                   data.zone == 9 or data.zone == 10 ) and
                 ( data.subzone <> 23 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamfish == 1 or data.streamfish == 2 or
                   data.lakelarge == 1 or data.lakelarge == 2 or
                   data.lakemed == 1 or data.lakemed == 2 or
                   data.lakesmall == 1 or data.wetlarge == 1 ) ):
                ret = 3

        if ( ( data.slope == 1 ) and
             ( data.zone == 4 or data.zone == 5 or
               data.zone == 9 or data.subzone == 21 or
               data.subzone == 22 or data.subzone == 24 ) and
             ( data.streamsalm == 1 or data.streamsalm == 2 or
               data.streamfish == 1 or data.streamfish == 2 or
               data.lakelarge == 1 or data.lakelarge == 2 or
               data.lakemed == 1 or data.lakemed == 2 or
               data.lakesmall == 1 or data.lakevs == 1 or
               data.wetlarge == 1 ) ):
            ret = 4

        if ( ( data.slope == 1 ) and
             ( data.zone == 4 or data.zone == 8 or
               data.zone == 9 or data.zone == 10 ) and
             ( data.streamfish == 1 or data.streamfish == 2 or
               data.streamS4 == 1 or data.streamS4 == 2 ) and
             ( data.lakelarge == 1 or data.lakelarge == 2 or
               data.lakelarge == 3 or data.lakelarge == 4 or
               data.lakemed == 1 or data.lakemed == 2 or
               data.lakemed == 3 or data.lakesmall == 1 or
               data.lakesmall == 2 or
               data.lakevs == 1 or data.wetlarge == 1 or
               data.wetlarge == 2 or data.wetsmall == 1 ) ):
            ret = 4

        return ret


class SiteType22(AbstractSiteType):
    """TYP22: Kill/butcher - hunting elk ( moose )
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.zone == 4 or data.zone == 5 or
                   data.zone == 9 or data.zone == 10 ) ):
                ret = 1

            if ( ( data.slope == 1 ) and
                 ( data.zone == 2 or data.zone == 4 or
                   data.zone == 9 ) and
                 ( data.openrange == 1 ) ):
                ret = 2

            if ( ( data.slope == 1 or data.slope == 2 ) and
                 ( data.zone == 4 or data.zone == 5 or
                   data.zone == 9 or data.zone == 10 ) and
                 ( data.streamsalm == 3 or data.streamsalm == 4 or
                   data.streamfish == 3 or data.streamfish == 4 or
                   data.streamothr == 1 or data.streamothr == 2 or
```

```
            data.lakelarge == 3 or data.lakelarge == 4 or
            data.lakemed == 3 or data.lakemed == 4 or
            data.lakesmall == 2 or data.wetlarge == 2 ) ):
        ret = 2

    if ( ( data.slope == 2 ) and
         ( data.zone == 4 or data.zone == 5 or
           data.zone == 9 or data.zone == 10 ) and
         ( data.subzone <> 23 ) and
         ( data.streamothr == 1 or data.lakesmall == 2 or
           data.lakevs == 1 or data.wetlarge == 2 or
           data.wetsmall == 1 ) ):
        ret = 2

    if ( ( data.slope == 1 ) and
         ( data.zone == 4 or data.zone == 5 or
           data.zone == 9 or data.zone == 10 ) and
         ( data.subzone <> 23 ) and
         ( data.streamothr == 1 or data.lakesmall == 2 or
           data.lakevs == 1 or data.wetlarge == 2 or
           data.wetsmall == 1 ) ):
        ret = 3

    if ( ( data.slope == 2 ) and
         ( data.zone == 4 or data.zone == 5 or
           data.zone == 9 or data.zone == 10 ) and
         ( data.subzone <> 23 ) and
         ( data.streamsalm == 1 or data.streamsalm == 2 or
           data.streamfish == 1 or data.streamfish == 2 or
           data.lakelarge == 1 or data.lakelarge == 2 or
           data.lakemed == 1 or data.lakemed == 2 or
           data.lakesmall == 1 or data.wetlarge == 1 ) ):
        ret = 3

if ( ( data.slope == 1 ) and
     ( data.zone == 4 or data.zone == 5 or
       data.zone == 9 or data.subzone == 21 or
       data.subzone == 22 or data.subzone == 24 ) and
     ( data.streamsalm == 1 or data.streamsalm == 2 or
       data.streamfish == 1 or data.streamfish == 2 or
       data.lakelarge == 1 or data.lakelarge == 2 or
       data.lakemed == 1 or data.lakemed == 2 or
       data.lakesmall == 1 or data.lakevs == 1 or
       data.wetlarge == 1 ) ):
    ret = 4

if ( ( data.slope == 1 ) and
     ( data.zone == 4 or data.zone == 8 or
       data.zone == 9 or data.zone == 10 ) and
     ( data.streamfish == 1 or data.streamfish == 2 or
       data.streamS4 == 1 or data.streamS4 == 2 ) and
     ( data.lakelarge == 1 or data.lakelarge == 2 or
       data.lakelarge == 3 or data.lakelarge == 4 or
       data.lakemed == 1 or data.lakemed == 2 or
       data.lakemed == 3 or data.lakesmall == 1 or
       data.lakesmall == 2 or
       data.lakevs == 1 or data.wetlarge == 1 or
       data.wetlarge == 2 or data.wetsmall == 1 ) ):
    ret = 4

return ret
```

```
class SiteType23(AbstractSiteType):
    """TYP23: Game overlook - hunting elk ( moose )
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.slope == 1 ) and
                 ( data.zone == 2 or data.zone == 4 or
                   data.zone == 9 ) and
                 ( data.openrange == 1 ) ):
                ret = 2

            if ( ( data.slope == 1 ) and
                 ( data.zone == 4 or data.zone == 5 or
                   data.zone == 9 or data.zone == 10 ) and
                 ( data.esker == 1 or data.meltsmall == 1 or
                   data.glacialake == 1 or data.cliffs == 1 ) ):
                ret = 2

            if ( ( data.slope == 1 or data.slope == 2 ) and
                 ( data.zone == 4 or data.zone == 5 or
                   data.zone == 9 or data.zone == 10 ) and
                 ( data.streamsalm == 3 or data.streamsalm == 4 or
                   data.streamfish == 3 or data.streamfish == 4 or
                   data.streamothr == 1 or data.streamothr == 2 or
                   data.lakelarge == 3 or data.lakelarge == 4 or
                   data.lakemed == 3 or data.lakemed == 4 or
                   data.lakesmall == 2 or data.wetlarge == 2 ) ):
                ret = 2

            if ( ( data.slope == 2 ) and
                 ( data.zone == 4 or data.zone == 5 or
                   data.zone == 9 or data.zone == 10 ) and
                 ( data.subzone <> 23 ) and
                 ( data.streamothr == 1 or data.lakesmall == 2 or
                   data.lakevs == 1 or data.wetlarge == 2 or
                   data.wetsmall == 1 ) ):
                ret = 2

            if ( ( data.slope == 1 ) and
                 ( data.zone == 4 or data.zone == 5 or
                   data.zone == 9 or data.zone == 10 ) and
                 ( data.subzone <> 23 ) and
                 ( data.streamsalm == 3 or data.streamsalm == 4 or
                   data.streamfish == 3 or data.streamfish == 4 or
                   data.streamothr == 2 or data.lakelarge == 3 or
                   data.lakelarge == 4 or data.lakemed == 3 or
                   data.lakemed == 4 ) and
                 ( data.esker == 1 or data.meltsmall == 1 or
                   data.glacialake == 1 or data.cliffs == 1 ) ):
                ret = 2

            if ( ( data.slope == 1 ) and
                 ( data.zone == 4 or data.zone == 5 or
                   data.zone == 9 or data.zone == 10 ) and
                 ( data.subzone <> 23 ) and
                 ( data.streamothr == 1 or data.lakesmall == 2 or
                   data.lakevs == 1 or data.wetlarge == 2 or
                   data.wetsmall == 1 ) ):
                ret = 3

            if ( ( data.slope == 2 ) and
```

```
                ( data.zone == 4 or data.zone == 5 or
                  data.zone == 9 or data.zone == 10 ) and
                ( data.subzone <> 23 ) and
                ( data.streamsalm == 1 or data.streamsalm == 2 or
                  data.streamfish == 1 or data.streamfish == 2 or
                  data.lakelarge == 1 or data.lakelarge == 2 or
                  data.lakemed == 1 or data.lakemed == 2 or
                  data.lakesmall == 1 or data.wetlarge == 1 ) ):
              ret = 3

        if ( ( data.slope == 1 ) and
             ( data.zone == 4 or data.zone == 5 or
               data.zone == 9 or data.subzone == 21 or
               data.subzone == 22 or data.subzone == 24 ) and
             ( data.streamsalm == 1 or data.streamsalm == 2 or
               data.streamfish == 1 or data.streamfish == 2 or
               data.lakelarge == 1 or data.lakelarge == 2 or
               data.lakemed == 1 or data.lakemed == 2 or
               data.lakesmall == 1 or data.lakevs == 1 or
               data.wetlarge == 1 ) ):
            ret = 4

        if ( ( data.slope == 1 ) and
             ( data.zone == 4 or data.zone == 8 or
               data.zone == 9 or data.zone == 10 ) and
             ( data.streamfish == 1 or data.streamfish == 2 or
               data.streamS4 == 1 or data.streamS4 == 2 ) and
             ( data.lakelarge == 1 or data.lakelarge == 2 or
               data.lakelarge == 3 or data.lakelarge == 4 or
               data.lakemed == 1 or data.lakemed == 2 or
               data.lakemed == 3 or data.lakesmall == 1 or
               data.lakesmall == 2 or
               data.lakevs == 1 or data.wetlarge == 1 or
               data.wetlarge == 2 or data.wetsmall == 1 ) ):
            ret = 4

        return ret


class SiteType24a(AbstractSiteType):
    """TYP24a: Base Camp - hunting mountain sheep, goats
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( data.slope == 1 and
                 ( data.zone == 1 or data.subzone == 11 or
                   data.subzone == 13 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamfish == 1 or data.streamfish == 2 ) and
                 ( data.lakemed == 1 or data.lakemed == 2 or
                   data.lakesmall == 1 or data.lakevs == 1 ) ):
                ret = 3

        return ret


class SiteType24b(AbstractSiteType):
    """TYP24b: Hunting blind - hunting mountain sheep, goats
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL
```

```
        if self.calculate_moderate_potentials:
            if ( ( data.slope == 1 or data.slope == 2 ) and
                 ( data.zone == 1 or data.subzone == 11 or
                   data.subzone == 13 ) ):
                ret = 2

            if ( data.slope == 1 and
                 ( data.zone == 1 or data.subzone == 11 or
                   data.subzone == 13 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamfish == 1 or data.streamfish == 2 ) and
                 ( data.lakemed == 1 or data.lakemed == 2 or
                   data.lakesmall == 1 or data.lakevs == 1 ) ):
                ret = 3

        return ret


class SiteType25(AbstractSiteType):
    """TYP25: Kill/butcher - hunting mountain sheep, goats
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.zone == 1 or data.subzone == 11 or
                   data.subzone == 13 ) ):
                ret = 1

            if ( ( data.slope == 1 or data.slope == 2 ) and
                 ( data.zone == 1 or data.subzone == 11 or
                   data.subzone == 13 ) ):
                ret = 2

            if ( data.slope == 1 and
                 ( data.zone == 1 or data.subzone == 11 or
                   data.subzone == 13 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamfish == 1 or data.streamfish == 2 ) and
                 ( data.lakemed == 1 or data.lakemed == 2 or
                   data.lakesmall == 1 or data.lakevs == 1 ) ):
                ret = 3

        return ret


class SiteType27(AbstractSiteType):
    """TYP27: Kill/butcher - hunting bears
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.zone == 1 or data.zone == 6 or
                   data.zone == 8 ) ):
                ret = 1

            if ( ( data.zone == 4 or data.zone == 5 or
                   data.zone == 9 or data.zone == 10 ) ):
                ret = 1

            if ( ( data.zone == 1 or data.subzone == 11 or
```

```
                   data.subzone == 13 ) ):
                ret = 1

        return ret


class SiteType28(AbstractSiteType):
    """TYP28: Kill/butcher - hunting marmots
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.zone == 1 or data.subzone == 11 or
                   data.subzone == 13 ) ):
                ret = 1

        return ret


class SiteType29(AbstractSiteType):
    """TYP29: Kill/Butcher - beaver/muskrat/otter
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.zone <> 1 and data.zone <> 6 ) and
                 ( data.streamsalm == 1 or data.streamfish == 1 ) ):
                ret = 2

            if ( ( data.zone <> 1 and data.zone <> 6 ) and
                 ( data.streamsalm == 1 or data.streamfish == 1 ) and
                 ( data.streamothr == 1 ) ):
                ret = 3

        if ( ( data.zone <> 1 and data.zone <> 6 ) and
             ( data.streamsalm == 1 or data.streamfish == 1 or
               data.streamS4 == 1) and
             ( data.lakelarge == 1 or data.lakemed == 1 or
               data.lakesmall == 1 ) ):
            ret = 4

        if ( ( data.zone <> 1 and data.zone <> 6 ) and
             ( data.streamsalm == 1 ) and
             ( data.streamfish == 1 ) ):
            ret = 4

        return ret


class SiteType30(AbstractSiteType):
    """TYP30: Kill/butcher - hunting snowshoe hares
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.zone == 1 or data.zone == 6 or
                   data.zone == 8 ) ):
                ret = 1

            if ( ( data.zone == 4 or data.zone == 5 or
```

```
                   data.zone == 9 or data.zone == 10 ) ):
                ret = 1

        return ret


class SiteType32(AbstractSiteType):
    """TYP32: Kill/butcher - hunting grouse
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.zone == 1 or data.zone == 6 or
                   data.zone == 8 ) ):
                ret = 1

            if ( ( data.zone == 4 or data.zone == 5 or
                   data.zone == 9 or data.zone == 10 ) ):
                ret = 1

        return ret


class SiteType33(AbstractSiteType):
    """TYP33: Base camp - hunting waterfowl
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( data.slope == 1 and data.falls == 1 ):
                ret = 3

            if ( ( data.slope == 1 ) and
                 ( data.lakelarge == 1 or data.lakelarge == 2 or
                   data.lakemed == 1 or data.lakemed == 2 or
                   data.lakesmall == 1 or data.wetlarge == 1 ) ):
                ret = 3

        return ret


class SiteType34(AbstractSiteType):
    """TYP34: Kill/butcher - hunting waterfowl
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.slope == 2 ) and
                 ( data.lakelarge == 1 or data.lakelarge == 2 or
                   data.lakemed == 1 or data.lakemed == 2 or
                   data.lakesmall == 1 or data.wetlarge == 1 ) ):
                ret = 2

            if ( ( data.slope == 1 ) and
                 ( data.lakelarge == 1 or data.lakelarge == 2 or
                   data.lakemed == 1 or data.lakemed == 2 or
                   data.lakesmall == 1 or data.wetlarge == 1 ) ):
                ret = 3

        if ( ( data.slope == 1 or data.slope == 2 ) and
```

```
             ( data.streamsalm == 1 or data.streamfish == 1 ) and
             ( data.falls == 1 ) ):
            ret = 4

        return ret


class SiteType35(AbstractSiteType):
    """TYP35: Hunting blind - hunting waterfowl
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.slope == 2 ) and
                 ( data.lakelarge == 1 or data.lakelarge == 2 or
                   data.lakemed == 1 or data.lakemed == 2 or
                   data.lakesmall == 1 or data.wetlarge == 1 ) ):
                ret = 2

            if ( ( data.slope == 1 ) and
                 ( data.lakelarge == 1 or data.lakelarge == 2 or
                   data.lakemed == 1 or data.lakemed == 2 or
                   data.lakesmall == 1 or data.wetlarge == 1 ) ):
                ret = 3

        if ( ( data.slope == 1 or data.slope == 2 ) and
             ( data.streamsalm == 1 or data.streamfish == 1 ) and
             ( data.falls == 1 ) ):
            ret = 4

        return ret


class SiteType36(AbstractSiteType):
    """TYP36 Base camp - Plant gathering - spring beauty, avalanche lily
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.slope == 2 ) and
                 ( data.streamfish == 1 or data.streamfish == 2 or
                   data.lakemed == 1 or data.lakemed == 2 or
                   data.lakesmall == 1 or data.lakevs == 1 or
                   data.wetlarge == 1 or data.wetsmall == 1 ) and
                 ( data.whitebark == 1 ) ):
                ret = 3

            if ( ( data.slope == 1 ) and
                 ( data.zone == 1 or data.zone == 6 ) and
                 ( data.streamfish == 1 or data.streamfish == 2 ) and
                 ( data.lakemed == 1 or data.lakemed == 2 or
                   data.lakesmall == 1 or data.lakevs == 1 or
                   data.wetlarge == 1 ) and
                 ( data.whitebark <> 1 ) ):
                ret = 3

        if ( ( data.slope == 1 ) and
             ( data.streamfish == 1 or data.streamfish == 2 or
               data.streamS4 == 1 or
               data.lakemed == 1 or data.lakemed == 2 or
               data.lakesmall == 1 or data.lakevs == 1 or
```

```
            data.wetlarge == 1 or data.wetsmall == 1 ) and
          ( data.whitebark == 1 ) ):
            ret = 4

        return ret


class SiteType37(AbstractSiteType):
    """TYP37 Processing - Plant gathering - spring beauty, avalanche lily
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.zone == 1 or data.zone == 6 ) ):
                ret = 1

            if ( ( data.slope == 3 ) and
                 ( data.zone == 1 or data.zone == 6 ) and
                 ( data.streamfish == 1 or data.streamfish == 2 or
                   data.streamfish == 3 or data.lakemed == 1 or
                   data.lakemed == 2 or data.lakemed == 3 or
                   data.lakesmall == 1 or data.lakesmall == 2 or
                   data.lakevs == 1 or data.wetlarge == 1 or
                   data.wetlarge == 2 or data.wetsmall == 1 or
                   data.wetsmall == 2 ) and
                 ( data.whitebark <> 1 ) ):
                ret = 2

            if ( ( data.slope == 1 or data.slope == 2 ) and
                 ( data.zone == 1 or data.zone == 6 ) and
                 ( data.streamsalm == 3 or data.streamfish == 2 or
                   data.lakemed == 3 or data.lakesmall == 2 or
                   data.lakevs == 1 or data.wetlarge == 2 or
                   data.wetsmall == 1 ) and
                 ( data.whitebark <> 1 ) ):
                ret = 2

            if ( ( data.slope == 3 ) and
                 ( data.streamfish == 1 or data.streamfish == 2 or
                   data.streamfish == 3 or data.lakemed == 1 or
                   data.lakemed == 2 or data.lakemed == 3 or
                   data.lakesmall == 1 or data.lakesmall == 2 or
                   data.lakevs == 1 or data.wetlarge == 1 or
                   data.wetlarge == 2 or data.wetsmall == 1 or
                   data.wetsmall == 2 ) and
                 ( data.whitebark == 1 ) ):
                ret = 3

            if ( ( data.slope == 2 ) and
                 ( data.streamfish == 3 or data.lakemed == 3 or
                   data.lakesmall == 2 or data.lakevs == 1 or
                   data.wetlarge == 2 or data.wetsmall == 2 ) and
                 ( data.whitebark == 1 ) ):
                ret = 3

            if ( ( data.slope == 2 ) and
                 ( data.streamfish == 1 or data.streamfish == 2 or
                   data.lakemed == 1 or data.lakemed == 2 or
                   data.lakesmall == 1 or data.lakevs == 1 or
                   data.wetlarge == 1 or data.wetsmall == 1 ) and
                 ( data.whitebark == 1 ) ):
                ret = 3
```

```
            if ( ( data.slope == 1 ) and
                 ( data.zone == 1 or data.zone == 6 ) and
                 ( data.streamfish == 1 or data.streamfish == 2 ) and
                 ( data.lakemed == 1 or data.lakemed == 2 or
                   data.lakesmall == 1 or data.lakevs == 1 or
                   data.wetlarge == 1 ) and
                  ( data.whitebark <> 1 ) ):
                ret = 3

        if ( ( data.slope == 1 ) and
             ( data.streamfish == 1 or data.streamfish == 2 or
               data.streamS4 == 1 or
               data.lakemed == 1 or data.lakemed == 2 or
               data.lakesmall == 1 or data.lakevs == 1 or
               data.wetlarge == 1 or data.wetsmall == 1 ) and
             ( data.whitebark == 1 ) ):
            ret = 4

        return ret


class SiteType38(AbstractSiteType):
    """TYP38 Cache - Plant gathering - spring beauty, avalanche lily
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.zone == 1 or data.zone == 6 ) ):
                ret = 1

            if ( ( data.slope == 3 ) and
                 ( data.zone == 1 or data.zone == 6 ) and
                 ( data.streamfish == 1 or data.streamfish == 2 or
                   data.streamfish == 3 or data.lakemed == 1 or
                   data.lakemed == 2 or data.lakemed == 3 or
                   data.lakesmall == 1 or data.lakesmall == 2 or
                   data.lakevs == 1 or data.wetlarge == 1 or
                   data.wetlarge == 2 or data.wetsmall == 1 or
                   data.wetsmall == 2 ) and
                  ( data.whitebark <> 1 ) ):
                ret = 2

            if ( ( data.slope == 1 or data.slope == 2 ) and
                 ( data.zone == 1 or data.zone == 6 ) and
                 ( data.streamsalm == 3 or data.streamfish == 2 or
                   data.lakemed == 3 or data.lakesmall == 2 or
                   data.lakevs == 1 or data.wetlarge == 2 or
                   data.wetsmall == 1 ) and
                  ( data.whitebark <> 1 ) ):
                ret = 2

            if ( ( data.slope == 3 ) and
                 ( data.streamfish == 1 or data.streamfish == 2 or
                   data.streamfish == 3 or data.lakemed == 1 or
                   data.lakemed == 2 or data.lakemed == 3 or
                   data.lakesmall == 1 or data.lakesmall == 2 or
                   data.lakevs == 1 or data.wetlarge == 1 or
                   data.wetlarge == 2 or data.wetsmall == 1 or
                   data.wetsmall == 2 ) and
                  ( data.whitebark == 1 ) ):
                ret = 3
```

```
            if ( ( data.slope == 2 ) and
                 ( data.streamfish == 3 or data.lakemed == 3 or
                   data.lakesmall == 2 or data.lakevs == 1 or
                   data.wetlarge == 2 or data.wetsmall == 2 ) and
                 ( data.whitebark == 1 ) ):
                ret = 3

            if ( ( data.slope == 2 ) and
                 ( data.streamfish == 1 or data.streamfish == 2 or
                   data.lakemed == 1 or data.lakemed == 2 or
                   data.lakesmall == 1 or data.lakevs == 1 or
                   data.wetlarge == 1 or data.wetsmall == 1 ) and
                 ( data.whitebark == 1 ) ):
                ret = 3

            if ( ( data.slope == 1 ) and
                 ( data.zone == 1 or data.zone == 6 ) and
                 ( data.streamfish == 1 or data.streamfish == 2 ) and
                 ( data.lakemed == 1 or data.lakemed == 2 or
                   data.lakesmall == 1 or data.lakevs == 1 or
                   data.wetlarge == 1 ) and
                 ( data.whitebark <> 1 ) ):
                ret = 3

        if ( ( data.slope == 1 ) and
             ( data.streamfish == 1 or data.streamfish == 2 or
               data.streamS4 == 1 or
               data.lakemed == 1 or data.lakemed == 2 or
               data.lakesmall == 1 or data.lakevs == 1 or
               data.wetlarge == 1 or data.wetsmall == 1 ) and
             ( data.whitebark == 1 ) ):
            ret = 4

        return ret


class SiteType39(AbstractSiteType):
    """TYP39: Base Camp - balsamroot, nodding onion etc.
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.slope == 2 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamfish == 1 or data.streamfish == 2 or
                   data.streamothr == 1 or data.lakelarge == 1 or
                   data.lakelarge == 2 or data.lakemed == 1 or
                   data.lakemed == 2 or data.wetlarge == 1 ) and
                 ( data.ungulate == 2 ) ):
                ret = 3

            if ( ( data.slope == 1 ) and
                 ( data.zone == 4 or data.subzone == 21 or data.subzone == 22 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamfish == 1 or data.streamfish == 2 or
                   data.streamothr == 1 or data.lakelarge == 1 or
                   data.lakelarge == 2 or data.lakemed == 1 or
                   data.lakemed == 2 or data.wetlarge == 1 ) and
                 ( data.ungulate <> 2 ) ):
                ret = 3
```

```
            if ( ( data.slope == 2 ) and
                 ( data.streamfish == 1 or data.streamfish == 2 or
                   data.streamothr == 1 or data.lakelarge == 1 or
                   data.lakelarge == 2 or data.lakemed == 1 or
                   data.lakemed == 2 or data.wetlarge == 1 ) and
                 ( data.whitebark == 1 ) ):
                ret = 3

        if ( ( data.slope == 1 ) and
             ( data.streamsalm == 1 or data.streamsalm == 2 or
               data.streamfish == 1 or data.streamfish == 2 or
               data.streamS4 == 1 or
               data.lakelarge == 1 or
               data.lakelarge == 2 or data.lakemed == 1 or
               data.lakemed == 2 or data.wetlarge == 1 ) and
             ( data.ungulate == 2 ) ):
            ret = 4

        if ( ( data.slope == 1 ) and
             ( data.streamfish == 1 or data.streamfish == 2 or
               data.streamS4 == 1 or
               data.lakelarge == 1 or
               data.lakelarge == 2 or data.lakemed == 1 or
               data.lakemed == 2 or data.wetlarge == 1 ) and
             ( data.whitebark == 1 ) ):
            ret = 4

        return ret


class SiteType40(AbstractSiteType):
    """TYP40: Processing - Plant gathering - balsamroot, nodding onion etc.
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.zone == 4 ) ):
                ret = 1

            if ( data.slope == 1 and data.zone == 4 and
                 ( data.streamsalm == 4 or data.streamfish == 4 or
                   data.lakelarge == 4 or data.lakemed == 4 ) ):
                ret = 2

            if ( data.slope == 1 and data.zone == 4 and
                 ( data.streamsalm == 3 or data.streamfish == 3 or
                   data.lakelarge == 3 or data.lakemed == 3 or
                   data.lakesmall == 2 or data.wetlarge == 2 or
                   data.wetsmall == 2 ) ):
                ret = 3

        if ( data.slope == 1 and data.zone == 4 and
             ( data.streamsalm == 1 or data.streamsalm == 2 or
               data.streamfish == 1 or data.streamfish == 2 or
               data.streamS4 == 1 or
               data.lakelarge == 1 or
               data.lakelarge == 2 or data.lakemed == 1 or
               data.lakemed == 2 or data.lakesmall == 1 or
               data.lakevs == 1 or data.wetlarge == 1 or
               data.wetsmall == 1 ) ):
            ret = 4
```

```
        return ret


class SiteType41(AbstractSiteType):
    """TYP41: Cache - Plant gathering - balsamroot, nodding onion etc.
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.zone == 4 ) ):
                ret = 1

            if ( data.slope == 1 and data.zone == 4 and
                 ( data.streamsalm == 4 or data.streamfish == 4 or
                   data.lakelarge == 4 or data.lakemed == 4 ) ):
                ret = 2

            if ( data.slope == 1 and data.zone == 4 and
                 ( data.streamsalm == 3 or data.streamfish == 3 or
                   data.lakelarge == 3 or data.lakemed == 3 or
                   data.lakesmall == 2 or data.wetlarge == 2 or
                   data.wetsmall == 2 ) ):
                ret = 3

        if ( data.slope == 1 and data.zone == 4 and
             ( data.streamsalm == 1 or data.streamsalm == 2 or
               data.streamfish == 1 or data.streamfish == 2 or
               data.streamS4 == 1 or
               data.lakelarge == 1 or
               data.lakelarge == 2 or data.lakemed == 1 or
               data.lakemed == 2 or data.lakesmall == 1 or
               data.lakevs == 1 or data.wetlarge == 1 or
               data.wetsmall == 1 ) ):
            ret = 4

        return ret


class SiteType42(AbstractSiteType):
    """TYP42 Processing - Plant gathering - huckleberries
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.zone == 1 or data.zone == 6 ) ):
                ret = 1

        return ret


class SiteType43(AbstractSiteType):
    """TYP43: Processing - Plant gathering - saskatoons, soapberries, blueberries
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.zone == 4 or data.zone == 5 or
                   data.zone == 9 or data.zone == 10 ) ):
                ret = 1
```

```
        return ret


class SiteType44(AbstractSiteType):
    """TYP44: Processing - Plant gathering - kinnikinnick berries
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.zone == 4 or data.zone == 5 or
                   data.zone == 9 or data.zone == 10 ) ):
                ret = 1

        return ret


class SiteType45(AbstractSiteType):
    """TYP45: Processing - Plant gathering - tree lichen
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.zone == 1 or data.zone == 6 or
                   data.zone == 8 ) ):
                ret = 1

        return ret


class SiteType46(AbstractSiteType):
    """TYP46 Processing - Plant gathering - whitebark pine nuts
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.zone == 1 or data.zone == 6 ) ):
                ret = 1

        return ret


class SiteType47(AbstractSiteType):
    """TYP47 Cache - Plant gathering - whitebark pine nuts
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.zone == 1 or data.zone == 6 ) ):
                ret = 1

        return ret


class SiteType48(AbstractSiteType):
    """TYP48: Culturally Modified tree - Plant gathering - lodgepole pine.
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL
```

```
        if self.calculate_moderate_potentials:
            if ( ( data.zone == 8 or data.zone == 9 or
                   data.zone == 10 ) ):
                ret = 1

            if ( ( data.slope == 1 or data.slope == 2 ) and
                 ( data.streamsalm == 4 or data.streamfish == 4 or
                   data.lakelarge == 4 or data.lakemed == 4 ) and
                 ( data.oldgrowth == 1 ) ):
                ret = 2

            if ( ( data.slope == 3 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamfish == 1 or data.streamfish == 2 or
                   data.lakelarge == 1 or data.lakelarge == 2 or
                   data.lakemed == 1 or data.lakemed == 2 or
                   data.lakesmall == 1 or data.wetlarge == 1 ) and
                 ( data.oldgrowth == 1 ) ):
                ret = 2

            if ( ( data.slope == 1 or data.slope == 2 ) and
                 ( data.streamsalm == 3 or data.streamfish == 3 or
                   data.lakelarge == 3 or data.lakemed == 3 or
                   data.wetlarge == 2 ) and
                 ( data.oldgrowth == 1 ) ):
                ret = 3

            if ( ( data.slope == 1 ) and ( data.trailgood == 1 ) ):
                ret = 3

        if ( ( data.slope == 1 or data.slope == 2 ) and
             ( data.streamsalm == 1 or data.streamsalm == 2 or
               data.streamfish == 1 or data.streamfish == 2 or
               data.streamS4 == 1 or data.streamS4 == 2 or
               data.lakelarge == 1 or data.lakelarge == 2 or
               data.lakemed == 1 or data.lakemed == 2 or
               data.lakesmall == 1 or data.wetlarge == 1 ) and
             ( data.oldgrowth == 1 ) ):
            ret = 4

        return ret


class SiteType49(AbstractSiteType):
    """TYP49: Cooking - Plant gathering - lodgepole pine
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.zone == 8 or data.zone == 9 or data.zone == 10 ) ):
                ret = 1

            if ( ( data.slope == 1 or data.slope == 2 ) and
                 ( data.streamsalm == 4 or data.streamfish == 4 or
                   data.lakelarge == 4 or data.lakemed == 4 ) and
                 ( data.oldgrowth == 1 ) ):
                ret = 2

            if ( ( data.slope == 3 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamfish == 1 or data.streamfish == 2 or
                   data.lakelarge == 1 or data.lakelarge == 2 or
```

```
                    data.lakemed == 1 or data.lakemed == 2 or
                    data.lakesmall == 1 or data.wetlarge == 1 ) and
                  ( data.oldgrowth == 1 ) ):
                ret = 2

            if ( ( data.slope == 1 or data.slope == 2 ) and
                 ( data.streamsalm == 3 or data.streamfish == 3 or
                   data.lakelarge == 3 or data.lakemed == 3 or
                   data.wetlarge == 2 ) and
                  ( data.oldgrowth == 1 ) ):
                ret = 3

        if ( ( data.slope == 1 or data.slope == 2 ) and
             ( data.streamsalm == 1 or data.streamsalm == 2 or
               data.streamfish == 1 or data.streamfish == 2 or
               data.streamS4 == 1 or data.streamS4 == 2 or
               data.lakelarge == 1 or data.lakelarge == 2 or
               data.lakemed == 1 or data.lakemed == 2 or
               data.lakesmall == 1 or data.wetlarge == 1 ) and
              ( data.oldgrowth == 1 ) ):
            ret = 4

        return ret


class SiteType51(AbstractSiteType):
    """TYP51: Culturally Modified tree - Plant gathering - cedar
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.zone == 3 or data.zone == 5 or
                   data.zone == 7 ) ):
                ret = 1

            if ( ( data.slope == 1 or data.slope == 2 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamsalm == 3 or data.streamfish == 1 or
                   data.streamfish == 2 or data.streamfish == 3 or
                   data.lakelarge == 1 or data.lakelarge == 2 or
                   data.lakelarge == 3 or data.lakemed == 1 or
                   data.lakemed == 2 or data.lakemed == 3 or
                   data.wetlarge == 1 ) and
                  ( data.oldgrowth == 2 ) ):
                ret = 3

        return ret


class SiteType52(AbstractSiteType):
    """TYP52: Habitation - pithouse
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.slope == 1 or data.slope == 2 ) and
                 ( data.zone <> 1 and data.zone <> 6 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 ) ):
                ret = 2

            if ( ( data.slope == 1 ) and
```

```
          ( data.zone <> 1 and data.zone <> 6 ) and
          ( data.streamfish == 1 or data.streamfish == 2 or
            data.lakelarge == 1 or data.lakelarge == 2 or
            data.lakemed == 1 or data.lakemed == 2 ) ):
        ret = 2

    if ( ( data.slope == 1 ) and
         ( data.zone <> 1 and data.zone <> 6 ) and
         ( data.streamfish == 1 or data.streamfish == 3 or
           data.streamfish == 4 ) and
         ( data.lakelarge == 3 or data.lakelarge == 4 or
           data.lakemed == 3 or data.lakemed == 4 ) ):
        ret = 3

    if ( ( data.slope == 2 ) and
         ( data.zone <> 1 and data.zone <> 6 ) and
         ( data.streamsalm == 1 or data.streamsalm == 2 or
           data.streamsalm == 3 ) and
         ( data.meltlarge == 1 ) ):
        ret = 3

    if ( ( data.slope == 1 ) and
         ( data.zone == 2 or data.zone == 4 or
           data.zone == 9 or data.zone == 10 ) and
         ( data.streamfish == 1 or data.streamfish == 2 or
           data.streamfish == 3 ) and
         ( data.meltlarge == 1 ) ):
        ret = 3

    if ( ( data.slope == 1 ) and
         ( data.zone <> 1 and data.zone <> 6 ) and
         ( data.streamsalm == 1 or data.streamsalm == 2 or
           data.streamsalm == 3 or data.streamsalm == 4 ) and
         ( data.streamfish == 3 or data.streamfish == 4 or
           data.streamothr == 2 or data.lakelarge == 3 or
           data.lakelarge == 4 or data.lakemed == 3 or
           data.lakemed == 4 or data.lakesmall == 2 or
           data.wetlarge == 2 or data.wetsmall == 2 ) ):
        ret = 3

    if ( ( data.slope == 1 ) and
         ( data.zone <> 1 and data.zone <> 6 ) and
         ( data.streamsalm == 4 ) and
         ( data.meltlarge == 1 ) ):
        ret = 3

if ( ( data.slope == 1 ) and
     ( data.zone == 2 or data.zone == 4 or
       data.zone == 9 or data.subzone == 21 or
       data.subzone == 22 or data.subzone == 24 ) and
     ( data.streamfish == 1 or data.streamfish == 2 ) and
     ( data.streamS4 == 1 or
       data.wetlarge == 1 or
       data.openrange == 1 ) ):
    ret = 4

if ( ( data.slope == 1 ) and
     ( data.zone <> 1 and data.zone <> 6 ) and
     ( data.streamsalm == 1 or data.streamsalm == 2 or
       data.streamsalm == 3 or data.lakelarge == 1 or
       data.lakelarge == 2 or
       data.lakemed == 1 or data.lakemed == 2 ) and
     ( data.meltlarge == 1 ) ):
```

```
            ret = 4

        if ( ( data.slope == 1 ) and
             ( data.zone <> 1 and data.zone <> 6 ) and
             ( data.streamfish == 1 or data.streamfish == 2 or
               data.streamfish == 3 ) and
             ( data.lakelarge == 1 or data.lakelarge == 2 or
               data.lakemed == 1 or data.lakemed == 2 or
               data.lakesmall == 1 ) ):
            ret = 4

        if ( ( data.slope == 1 ) and
             ( data.zone <> 1 and data.zone <> 6 ) and
             ( data.streamsalm == 1 or data.streamsalm == 2 or
               data.streamsalm == 3 or data.streamsalm == 4 ) and
             ( data.streamfish == 1 or data.streamfish == 2 or
               data.streamS4 == 1 or
               data.lakelarge == 1 or
               data.lakelarge == 2 or data.lakemed == 1 or
               data.lakemed == 2 or data.lakesmall == 1 or
               data.wetlarge == 1 or data.wetsmall == 1 ) ):
            ret = 4

        return ret


class SiteType53(AbstractSiteType):
    """TYP53: Habitation - rectangular plank house
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.slope == 2 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamfish == 1 or data.streamfish == 2 or
                   data.lakelarge == 1 or data.lakelarge == 2 or
                   data.lakemed == 1 or data.lakemed == 2 or
                   data.lakesmall == 1 ) ):
                ret = 2

            if ( ( data.slope == 1 ) and
                 ( data.streamsalm == 3 or data.streamfish == 3 or
                   data.lakelarge == 3 or data.lakemed == 3 or
                   data.lakesmall == 1 ) ):
                ret = 2

            if ( ( data.slope == 2 ) and
                 ( data.streamfish == 3 or data.streamfish == 4 ) and
                 ( data.meltlarge == 1 ) ):
                ret = 2

            if ( ( data.slope == 1 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamfish == 1 or data.streamfish == 2 or
                   data.lakelarge == 1 or data.lakelarge == 2 or
                   data.lakemed == 1 or data.lakemed == 2 or
                   data.lakesmall == 2 or data.lakevs == 1 ) ):
                ret = 3

            if ( ( data.slope == 1 ) and
                 ( data.streamfish == 3 or data.streamfish == 4 or
                   data.lakelarge == 3 or data.lakemed == 3 or
```

```
          data.lakesmall == 2 ) and
        ( data.meltlarge == 1 ) ):
       ret = 3

    if ( ( data.slope == 2 ) and
         ( data.streamfish == 1 or data.streamfish == 2 ) and
         ( data.meltlarge == 1 ) ):
       ret = 3

    if ( ( data.slope == 1 ) and
         ( data.streamfish == 3 ) and
         ( data.streamothr == 1 or data.streamothr == 2 or
           data.wetlarge == 1 or data.wetlarge == 2 or
           data.wetsmall == 1 or data.wetsmall == 2 or
           data.openrange == 1 ) ):
       ret = 3

    if ( ( data.slope == 1 ) and
         ( data.zone <> 1 and data.zone <> 6 ) and
         ( data.streamsalm == 1 or data.streamsalm == 2 or
           data.streamsalm == 3 or data.streamsalm == 4 ) and
         ( data.streamfish == 3 or data.streamfish == 4 or
           data.streamothr == 2 or data.lakelarge == 3 or
           data.lakelarge == 4 or data.lakemed == 3 or
           data.lakemed == 4 or data.lakesmall == 2 or
           data.wetlarge == 2 or data.wetsmall == 2 ) ):
       ret = 3

    if ( ( data.slope == 1 ) and
         ( data.zone <> 1 and data.zone <> 6 ) and
         ( data.streamsalm == 4 ) and
         ( data.meltlarge == 1 ) ):
       ret = 3

if ( ( data.slope == 1 ) and
     ( data.zone <> 1 and data.zone <> 6 ) and
     ( data.streamsalm == 1 or data.streamsalm == 2 or
       data.streamsalm == 3 or data.lakelarge == 1 or
       data.lakelarge == 2 or
       data.lakemed == 1 or data.lakemed == 2 ) and
      ( data.meltlarge == 1 ) ):
    ret = 4

if ( ( data.slope == 1 ) and
     ( data.zone <> 1 and data.zone <> 6 ) and
     ( data.streamfish == 1 or data.streamfish == 2 or
       data.streamfish == 3 ) and
      ( data.lakelarge == 1 or data.lakelarge == 2 or
       data.lakemed == 1 or data.lakemed == 2 or
       data.lakesmall == 1 ) ):
    ret = 4

if ( ( data.slope == 1 ) and
     ( data.zone <> 1 and data.zone <> 6 ) and
     ( data.streamsalm == 1 or data.streamsalm == 2 or
       data.streamsalm == 3 or data.streamsalm == 4 ) and
      ( data.streamfish == 1 or data.streamfish == 2 or
       data.streamS4 == 1 or
       data.lakelarge == 1 or
       data.lakelarge == 2 or data.lakemed == 1 or
       data.lakemed == 2 or data.lakesmall == 1 or
       data.wetlarge == 1 or data.wetsmall == 1 ) ):
    ret = 4
```

```
        if ( ( data.slope == 1 ) and
             ( data.streamfish == 1 or data.streamfish == 2 or
               data.streamS4 == 1 or
               data.lakelarge == 1 or data.lakelarge == 2 or
               data.lakemed == 1 or data.lakemed == 2 or
               data.lakesmall == 1 ) and
             ( data.meltlarge == 1 ) ):
            ret = 4

        if ( ( data.slope == 1 ) and
             ( data.zone <> 1 and data.zone <> 5 and
               data.zone <> 6 and data.zone <> 8 ) and
             ( data.streamfish == 1 or data.lakelarge == 1 or
               data.lakemed == 1 or data.lakesmall == 1 ) ):
            ret = 4

        if ( ( data.slope == 1 ) and
             ( data.zone <> 1 and data.zone <> 6 ) and
             ( data.streamfish == 1 or data.streamfish == 2 ) and
             ( data.streamS4 == 1 or
               data.streamothr == 1 or
               data.wetlarge == 1 or
               data.wetsmall == 1 or
               data.openrange == 1 ) ):
            ret = 4

        return ret


class SiteType54(AbstractSiteType):
    """TYP54: Habitation - base camp
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.slope == 2 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamfish == 1 or data.streamfish == 2 or
                   data.lakelarge == 1 or data.lakelarge == 2 or
                   data.lakemed == 1 or data.lakemed == 2 or
                   data.lakesmall == 1 ) ):
                ret = 2

            if ( ( data.slope == 1 ) and
                 ( data.streamsalm == 3 or data.streamfish == 3 or
                   data.lakelarge == 3 or data.lakemed == 3 or
                   data.lakesmall == 1 ) ):
                ret = 2

            if ( ( data.slope == 2 ) and
                 ( data.streamfish == 3 or data.streamfish == 4 ) and
                 ( data.meltlarge == 1 ) ):
                ret = 2

            if ( ( data.slope == 1 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamfish == 1 or data.streamfish == 2 or
                   data.lakelarge == 1 or data.lakelarge == 2 or
                   data.lakemed == 1 or data.lakemed == 2 or
                   data.lakesmall == 2 or data.lakevs == 1 ) ):
                ret = 3
```

```
    if ( ( data.slope == 1 ) and
         ( data.streamfish == 3 or data.streamfish == 4 or
           data.lakelarge == 3 or data.lakemed == 3 or
           data.lakesmall == 2 ) and
         ( data.meltlarge == 1 ) ):
        ret = 3

    if ( ( data.slope == 2 ) and
         ( data.streamfish == 1 or data.streamfish == 2 ) and
         ( data.meltlarge == 1 ) ):
        ret = 3

    if ( ( data.slope == 1 ) and
         ( data.streamfish == 3 ) and
         ( data.streamothr == 1 or data.streamothr == 2 or
           data.wetlarge == 1 or data.wetlarge == 2 or
           data.wetsmall == 1 or data.wetsmall == 2 or
           data.openrange == 1 ) ):
        ret = 3

    if ( ( data.slope == 1 ) and
         ( data.zone <> 1 and data.zone <> 6 ) and
         ( data.streamsalm == 1 or data.streamsalm == 2 or
           data.streamsalm == 3 or data.streamsalm == 4 ) and
         ( data.streamfish == 3 or data.streamfish == 4 or
           data.streamothr == 2 or data.lakelarge == 3 or
           data.lakelarge == 4 or data.lakemed == 3 or
           data.lakemed == 4 or data.lakesmall == 2 or
           data.wetlarge == 2 or data.wetsmall == 2 ) ):
        ret = 3

    if ( ( data.slope == 1 ) and
         ( data.zone <> 1 and data.zone <> 6 ) and
         ( data.streamsalm == 4 ) and
         ( data.meltlarge == 1 ) ):
        ret = 3

if ( ( data.slope == 1 ) and
     ( data.zone <> 1 and data.zone <> 6 ) and
     ( data.streamsalm == 1 or data.streamsalm == 2 or
       data.streamsalm == 3 or data.lakelarge == 1 or
       data.lakelarge == 2 or
       data.lakemed == 1 or data.lakemed == 2 ) and
     ( data.meltlarge == 1 ) ):
    ret = 4

if ( ( data.slope == 1 ) and
     ( data.zone <> 1 and data.zone <> 6 ) and
     ( data.streamfish == 1 or data.streamfish == 2 or
       data.streamfish == 3 ) and
     ( data.lakelarge == 1 or data.lakelarge == 2 or
       data.lakemed == 1 or data.lakemed == 2 or
       data.lakesmall == 1 ) ):
    ret = 4

if ( ( data.slope == 1 ) and
     ( data.zone <> 1 and data.zone <> 6 ) and
     ( data.streamsalm == 1 or data.streamsalm == 2 or
       data.streamsalm == 3 or data.streamsalm == 4 ) and
     ( data.streamfish == 1 or data.streamfish == 2 or
       data.streamS4 == 1 or
       data.lakelarge == 1 or
```

```
              data.lakelarge == 2 or data.lakemed == 1 or
              data.lakemed == 2 or data.lakesmall == 1 or
              data.wetlarge == 1 or data.wetsmall == 1 ) ):
            ret = 4

        if ( ( data.slope == 1 ) and
             ( data.streamfish == 1 or data.streamfish == 2 or
               data.streamS4 == 1 or
               data.lakelarge == 1 or data.lakelarge == 2 or
               data.lakemed == 1 or data.lakemed == 2 or
               data.lakesmall == 1 ) and
             ( data.meltlarge == 1 ) ):
            ret = 4

        if ( ( data.slope == 1 ) and
             ( data.zone <> 1 and data.zone <> 5 and
               data.zone <> 6 and data.zone <> 8 ) and
             ( data.streamfish == 1 or data.lakelarge == 1 or
               data.lakemed == 1 or data.lakesmall == 1 ) ):
            ret = 4

        if ( ( data.slope == 1 ) and
             ( data.zone <> 1 and data.zone <> 6 ) and
             ( data.streamfish == 1 or data.streamfish == 2 ) and
             ( data.streamS4 == 1 or
               data.streamothr == 1 or
               data.wetlarge == 1 or
               data.wetsmall == 1 or
               data.openrange == 1 ) ):
            ret = 4

        return ret


class SiteType55(AbstractSiteType):
    """TYP55: Habitation - rock shelter
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.cliffs == 1 ) ):
                ret = 2

            if ( ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamsalm == 3 or data.streamfish == 1 or
                   data.streamfish == 2 or data.streamfish == 3 or
                   data.lakelarge == 1 or data.lakelarge == 2 or
                   data.lakelarge == 3 or data.lakemed == 1 or
                   data.lakemed == 2 or data.lakemed == 3 ) and
                 ( data.cliffs == 1 ) ):
                ret = 3

        return ret


class SiteType56(AbstractSiteType):
    """TYP56: Trails - also see mapped trail coverage
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
```

```
            if ( ( data.streamfish == 1 ) ):
                ret = 3

        return ret


class SiteType57(AbstractSiteType):
    """TYP57: Transit Camp
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.slope == 2 ) and
                 ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamfish == 1 or data.streamfish == 2 or
                   data.streamothr == 1 or data.lakelarge == 1 or
                   data.lakemed == 1 or data.lakemed == 2 or
                   data.lakesmall == 1 or data.lakevs == 1 or
                   data.wetlarge == 1 or data.wetsmall == 1 ) and
                 ( data.trailgood == 1 ) ):
                ret = 3

        if ( ( data.slope == 1 ) and
             ( data.streamsalm == 1 or data.streamsalm == 2 or
               data.streamfish == 1 or data.streamfish == 2 or
               data.streamS4 == 1 or
               data.streamothr == 1 or data.lakelarge == 1 or
               data.lakelarge == 2 or data.lakemed == 1 or
               data.lakemed == 2 or data.lakesmall == 1 or
               data.lakevs == 1 or data.wetlarge == 1 or
               data.wetsmall == 1 ) and
             ( data.trailgood == 1 ) ):
            ret = 4

        return ret


class SiteType58(AbstractSiteType):
    """TYP58: Rock Quarrying - Quarry
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.cliffs == 1 ) ):
                ret = 2

            if ( ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamfish == 1 or data.streamfish == 2 or
                   data.streamothr == 1 ) and
                 ( data.lithiczone == 1 ) ):
                ret = 3

            if ( ( data.cliffs == 1 ) and
                 ( data.lithiczone == 1 ) ):
                ret = 3

        return ret


class SiteType59(AbstractSiteType):
    """TYP59: Rock Quarrying - lithic workshop
```

```
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.slope == 1 or data.slope == 2 or
                   data.slope == 3 ) and
                 ( data.streamsalm == 4 or data.streamfish == 4 ) and
                 ( data.lithiczone == 1 ) ):
                ret = 2

            if ( ( data.slope == 1 ) and
                 ( data.lithiczone == 1 ) ):
                ret = 2

            if ( ( data.slope == 1 or data.slope == 2 ) and
                 ( data.streamsalm == 3 or data.streamfish == 3 or
                   data.streamothr == 2 ) and
                 ( data.lithiczone == 1 ) ):
                ret = 3

        if ( ( data.slope == 1 or data.slope == 2 ) and
             ( data.streamsalm == 1 or data.streamsalm == 2 or
               data.streamfish == 1 or data.streamfish == 2 or
               data.streamS4 == 1 or
               data.streamothr == 1 ) and
             ( data.lithiczone == 1 ) ):
            ret = 4

        return ret


class SiteType60(AbstractSiteType):
    """TYP60: Rock Quarrying - Base camp
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            pass

        if ( ( data.slope == 1 ) and
             ( data.streamsalm == 1 or data.streamsalm == 2 or
               data.streamfish == 1 or data.streamfish == 2 or
               data.streamS4 == 1 or
               data.streamothr == 1 ) and
             ( data.lithiczone == 1 ) ):
            ret = 4

        return ret


class SiteType61(AbstractSiteType):
    """TYP61: Ceremonial - Rock art
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.streamsalm == 1 or data.streamsalm == 2 or
                   data.streamsalm == 3 or data.streamsalm == 4 or
                   data.streamfish == 1 or data.streamfish == 2 or
                   data.streamfish == 3 or data.streamfish == 4 or
```

```
                    data.lakelarge == 1 or data.lakelarge == 2 or
                    data.lakelarge == 3 or data.lakelarge == 4 or
                    data.lakemed == 1 or data.lakemed == 2 or
                    data.lakemed == 3 or data.lakemed == 4 ) and
                  ( data.cliffs == 1 ) ):
                ret = 3

            if ( ( data.streamsalm == 1 or data.streamsalm == 2 ) and
                 ( data.oldgrowth == 1 ) ):
                ret = 3

        return ret


class SiteTypeSITPOT(AbstractSiteType):
    """SITPOT: recorded sites
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.slope == 1 or data.slope == 2 ) and
                 ( data.sitepnt == 2 or data.siteply == 2 ) ):
                ret = 3

        if ( ( data.sitepnt == 1 or data.siteply == 1 ) ):
            ret = 4

        return ret


class SiteTypeTRAPOT(AbstractSiteType):
    """TRAPOT: known trails
    """
    def potential(self, data):
        ret = self.DEFAULT_POTENTIAL

        if self.calculate_moderate_potentials:
            if ( ( data.trailmod == 1 or data.trailpoor == 1 ) ):
                ret = 2

            if ( data.trailgood == 1 ):
                ret = 3

        return ret


class ArchPotModel:
    """Archaeological Potential Model
    """
    def __init__(self,
                 preload_package_types=True,
                 calculate_moderate_potentials=True
                 ):
        self._sitetype_objects = {}
        self.calculate_moderate_potentials = calculate_moderate_potentials

        if preload_package_types:
            self.loadAllPackageSiteTypes()

        self.site_categories = {
            'ALL': {
                'pot_field':  'pot',
```

```
            'list_field': 'hi_types',
            'site_types': [
                '1','2','3','4','5','6','7','8','9','10',
                '11','12','13','14','15','16','17','18',
                '19','20','21','22','23','24a','24b','25',
                '27','28','29','30','32','33','34','35',
                '36','37','38','39','40','41','42','43',
                '44','45','46','47','52','53','54','55',
                '56','57','58','59','60','61','SITPOT',
                'TRAPOT',
                ],
            },
        'PERMANENT HABITATION': {
            'pot_field':  'pot_prmhab',
            'list_field': 'hi_prmhab',
            'site_types': [
                '52','53','54','55',
                ],
            },
        'TEMPORARY HABITATION': {
            'pot_field':  'pot_tmphab',
            'list_field': 'hi_tmphab',
            'site_types': [
                '1','2','3','4','5','6','7','8','9','10',
                '11','12','13','14','15','16','17','18',
                '19','20','21','22','23','24a','24b','25',
                '27','28','29','30','32','33','34','35',
                '36','37','38','39','40','41','42','43',
                '44','45','46','47','57',
                ],
            },
        'HUMAN REMAINS': {
            'pot_field':  'pot_humrem',
            'list_field': 'hi_humrem',
            'site_types': [
                '1','2','3','52','53','54',
                ],
            },
        'CMTS': {
            'pot_field':  'pot_cmts',
            'list_field': 'hi_cmts',
            'site_types': [
                '48','49','51',
                ],
            },
        }

def loadAllPackageSiteTypes(self):
    global_values = globals().values()
    for entity in global_values:
        if ( (type(entity).__name__ == 'classobj') and
             issubclass(entity, AbstractSiteType) and
             (entity is not AbstractSiteType)
            ):
            self.addSiteType(entity)

def addSiteType(self, classobj):
    if type(classobj).__name__ != 'classobj':
        raise ArchPotModelError('Argument must be a classobj')

    if not issubclass(classobj, AbstractSiteType):
        raise ArchPotModelError(
            classobj.__name__ + 'is not a subclass of AbstractSiteType')
```

```
    if classobj is AbstractSiteType:
        raise ArchPotModelError(
            'AbstractSiteType cannot explicitly implement a SiteType')

    site_type_obj = classobj(self.calculate_moderate_potentials)
    site_type_name = site_type_obj.siteType()

    if site_type_name not in self._sitetype_objects.keys():
        self._sitetype_objects[site_type_name] = site_type_obj
    else:
        existing_obj = self._sitetype_objects[site_type_name]
        existing_class = existing_obj.__class__

        if ( issubclass(classobj, existing_class) and
             (classobj is not existing_class)
            ):
            self._sitetype_objects[site_type_name] = site_type_obj
        else:
            raise ArchPotModelError(
                'Duplicate implementation for SiteType "' +
                site_type_name + '"')

def siteTypesList(self):
    """Returns a sorted list of the Site Type names
    """
    zero_padded_site_types = map(lambda item: item.rjust(2, '0'),
                                 self._sitetype_objects.keys())
    return map(lambda item: item.lstrip('0'),
               sorted(zero_padded_site_types))

def siteTypeToFieldName(self, site_type):
    """Returns the database field name for the given site type
    """
    if site_type[0].isdigit():
        return 'typ' + site_type
    else:
        return site_type

def fieldNameToSiteType(self, field_name):
    """Returns the site type name for the given database field name
    """
    if field_name[:3].lower() == 'typ':
        return field_name[3:]
    else:
        return field_name

def siteTypeFieldsList(self):
    """Returns a list of Site Type field names
    """
    return map(lambda item: self.siteTypeToFieldName(item),
               self.siteTypesList())

def potentials(self, data):
    """Returns a dict of arch. pot. values, one for each SiteType
    """
    ret = dict()
    for sitetype in self._sitetype_objects.keys():
        sitetype_object = self._sitetype_objects[sitetype]
        potential = sitetype_object.potential(data)
        ret[sitetype] = potential

    return ret
```

```
def archPotential(self, potentials, site_category='ALL'):
    """Returns the arch. pot represented by the given potentials dict

    site_category determines which site types contribute to the arch
    potential to be returned.

    The minimum allowed arch pot is 1, not 0, so if none of the
    given site category's site types has a non-zero potential, this
    method returns 1 anyway.
    """
    category_uc = site_category.upper()
    site_types = self.site_categories[category_uc]['site_types']
    return max([potentials[stype] for stype in site_types] + [1])

def hiPotSiteTypes(self, potentials, site_category='ALL'):
    """Returns a list of site types of high pot. for the site category

    site_category determines which site types contribute to the arch
    potential to be returned.
    """
    category_uc = site_category.upper()
    site_types = self.site_categories[category_uc]['site_types']
    return [st for st in site_types if potentials[st] == 4]

def archPotentialFieldName(self, site_category='ALL'):
    """Returns the database field name used to store arch potential

    Each site_category has its own arch potential database field.
    """
    category_uc = site_category.upper()
    return self.site_categories[category_uc]['pot_field']

def hiPotSiteTypesFieldName(self, site_category='ALL'):
    """Returns the database field name storing the high pot site types

    Each site_category has its own site-types-list database field.
    """
    category_uc = site_category.upper()
    return self.site_categories[site_category]['list_field']

def siteCategoriesList(self):
    """Returns a list of the site type categories
    """
    return self.site_categories.keys()
```

MATRIX
Research Ltd