

# AUTOMATING GIS WITH PYTHON

## Spring NEARC

**Kristina Grace**

GIS Analyst

**May 22, 2012**

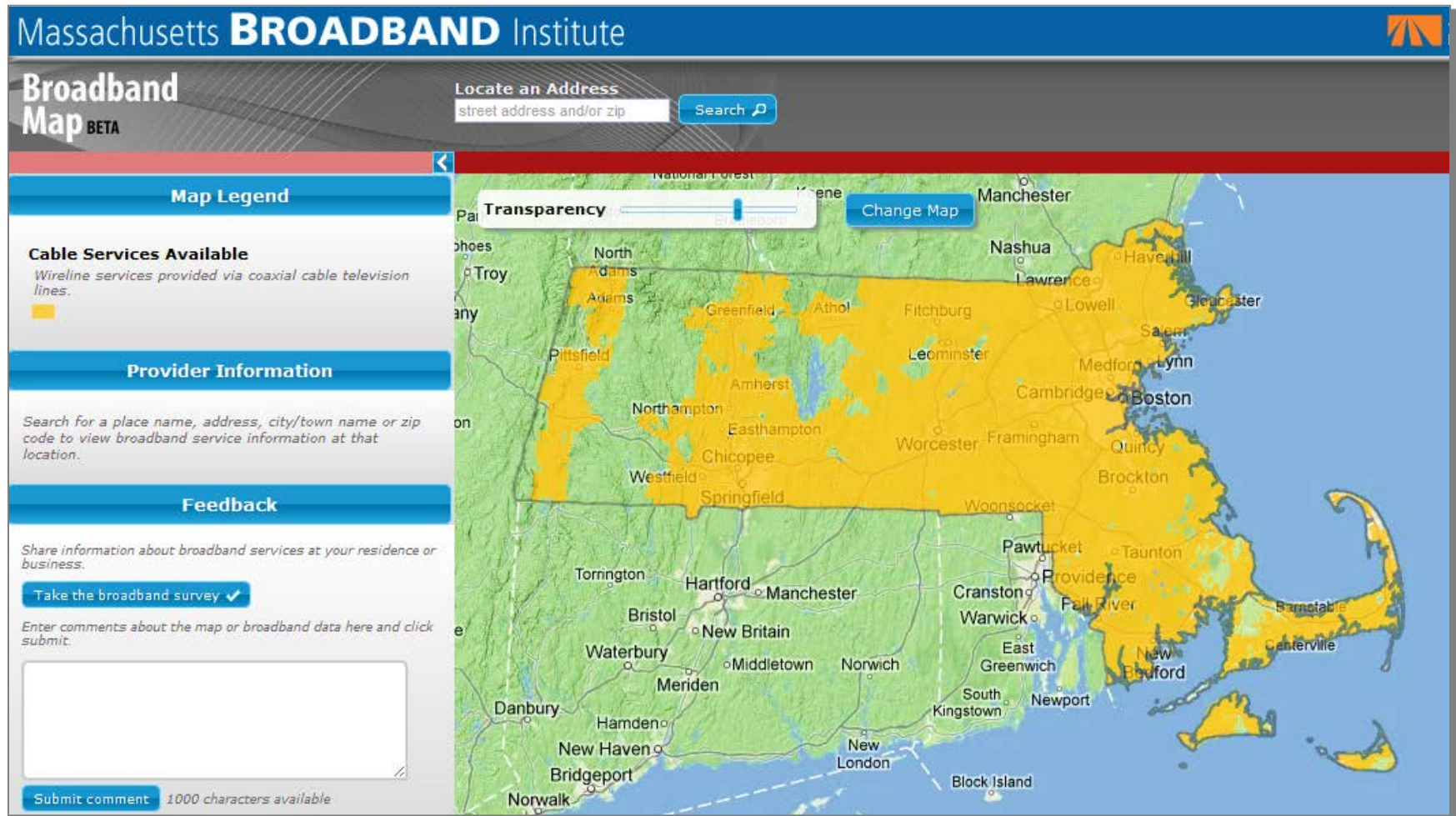


# Topics

- Broadband Availability and MassBroadband 123 maps
- Create and document tools made from Python scripts
- Script examples
  - error handling
  - tool parameters
  - check if path exists
  - where clause
  - calendar object
  - ArcToolbox: Conversion and Data Management
  - turn fields on and off in tables views
- Online build phase map

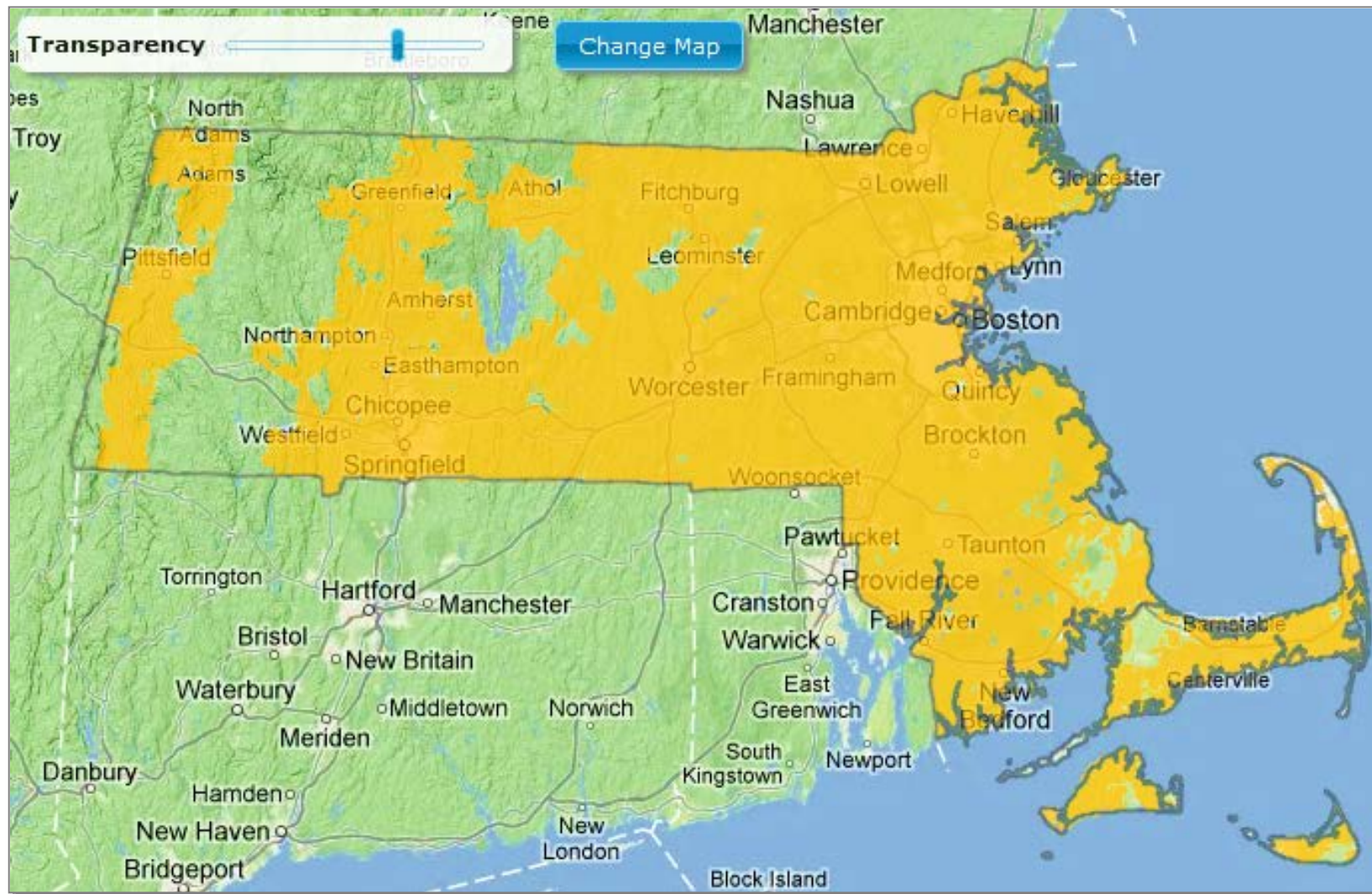
# Broadband Availability Map

<http://mapping.massbroadband.org>



Also incorporated at [www.broadbandmap.gov](http://www.broadbandmap.gov)

# Cable Availability Map

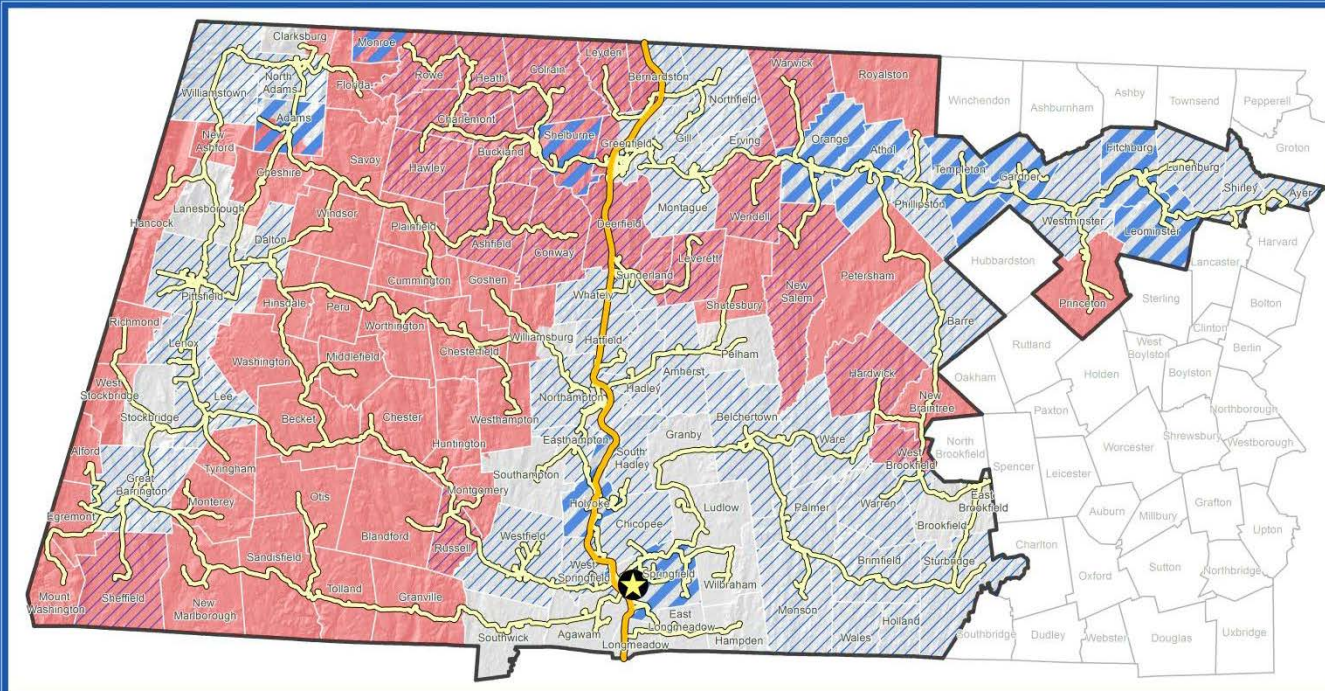




# MassBroadband 123



## MassBroadband 123 Service Area and Network Economic and Broadband Service Status



### Mass Broadband 123 Service Area Based on Federal Definitions

- Unserved or Underserved**  
Less than half of households have access to broadband and/or less than 40% of households subscribe to broadband
- Served**  
More than half of households have access to broadband and more than 40% of households subscribe to broadband

### Areas of Economic Need

- State Designation
- Federal & State Designations

State designations meet criteria for "Economic Target Areas."  
Federal designations meet criteria for "Economically Distressed Areas."

### Broadband Network

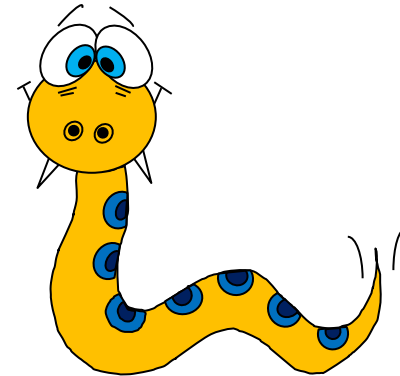
- MBI Existing Fiber Completed I-91 Installation
- MBI Fiber to be Built
- Internet Point of Presence





# Create And Document Tools

1. Create script
  - Python IDLE
  - use TEMPLATE.py
2. Add scripts to ArcToolboxes
  - add to tool or toolbox
3. Add description to tools
  - edit metadata
    - keywords
    - summary
    - parameter description



# Error Handling

- processes inside try
- error message formatting inside except

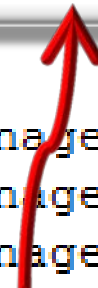
**arcpy.GetMessages(2)**

```
except:|
#
t
t
t
P
    msgs = 'GP ERRORS:\n' + arcpy.GetMessages(2)
    str(sys.exc_type) + ': ' + str(sys.exc_value) + '\n'
    arcpy.AddError(pymsg)
    msgs = 'GP ERRORS:\n' + arcpy.GetMessages(2) + '\n'
    arcpy.AddError(msgs)
```

# Delete Files And Feature Classes

**arcpy.Delete\_management(data\_element)**

arcpy.Delete\_management(CWR3LogTbl)



```
arcpy.Delete_management(CWR3LogTblView)
arcpy.Delete_management(workGDBParam + "\\\" + CWR3FiberTbl)
arcpy.Delete_management(workGDBParam + "\\\" + CWR3CAITbl)
arcpy.Delete_management(workGDBParam + "\\\" + CWR3LogTbl)
print 'intermediate tables deleted'
```




# Check If Path Exists

- Path supported by ArcGIS:  
`arcpy.Exists("")`

**if arcpy.Exists(dataset):  
    # do something**

- Path not supported by ArcGIS:  
`os.path.exists("")`

if arcpy.Exists(FC):  
    arcpy.Delete\_management(FC)



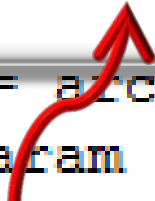
```
##Export CAI to CAI_today's date table in  
if arcpy.Exists(workGDBParam + "\\\" + CWR3  
    arcpy.Delete_management(workGDBParam +  
arcpy.TableToTable_conversion(MB123Network  
print 'imported the fiber and CAI tables'
```



# Tool Parameters

**Param = arcpy.GetParameterAsText(0)**

```
# dateChangeParam = arcpy.GetParameterAsText(3)
#
workGDBParam = arcpy.GetParameterAsText(1) # default
changeLogTblParam = arcpy.GetParameterAsText(2) # default
dateChangeParam = arcpy.GetParameterAsText(3) # default
CWR3LogTblFolderParam = arcpy.GetParameterAsText(4)
- - -
```



# Tool Parameters

Script Fiber and CAI to DBF v1.1 Properties


General Source Parameters Validation Help


Display Name	Data Type
@ master file geodata...	Workspace or Feature Dataset
work GDB	Workspace
original change log t...	Table
get logs starting wit...	Date
Output folder for D...	Folder


↑


↓

# Extract Data For Cost Analysis


 **Script Fiber and CAI to DBF v1.1**

master file geodatabase  


work GDB  


original change log table  


get logs starting with the following date

Output folder for DBF files  


## Script Fiber and CAI

This script joins GIS log sta  
by Cornell Robinson III.

To filter the log table so only  
'5/15/2012', '05/15/2012', 'M

The tool creates two new D  
The following fields from the  
ChangeDate.





# Export Feature Classes

Extract Data For Cost Analysis Script

```
arcpy.FeatureClassToFeatureClass_conversion  
(FC, out_path, out_name)
```

```
arcpy.FeatureClassToFeatureClass_conversion  
(MB123NetworkDSPParam + '\\CAI',  
workGDBParam, CWR3CAIFC)
```

```
arcpy.MakeTableView_management(workGDBParam + "\\\" +  
arcpy.FeatureClassToFeatureClass_conversion(MB123N  
arcpy.MakeTableView_management(workGDBParam + "\\\"
```

# Where Clause

- String variable
- Field name needs to be surrounded by quotes
- Used in
  - `arcpy.SelectLayerByAttribute_management()`
  - `arcpy.MakeTableView_management()`

**`whereClause = ' "fieldName" >= ' + variable`**

```
if int(dateList[1]) < 10:
```

`whereClause = ' "DateChange" >= ' + dateChange`

```
else:
```

```
    day = str(dateList[1])
```

```
    dateChange = 'date \' + year + '-' + month + '-' + day + '\'
```

```
    arcpy.AddMessage('dateChange: ' + dateChange)
```

```
    whereClause = ' "DateChange" >= ' + dateChange # query that wil
```



# Where Clause

## Extract Data For Cost Analysis Script

```
where_clause = "InclnBuild <> 'N' and InclnBuild <> 'W'"
```

```
dateChange = 'date \' + year + '-' + month + '-' + day + \''  
whereClause = ' "DateChange" >= ' + dateChange
```



# Turn Fields On And Off

## Extract Data For Cost Analysis Script

**fieldinfo.addField (field\_name, new\_field\_name,  
visible, split\_rule)**

```
fieldinfo.addField(field.name, field.name, 'HIDDEN', '')
```

```
for field in fields:
    if field.name == 'DateReques':
        fieldinfo.addField(field.name, field.name, 'HIDDEN', '')
    elif field.name == 'DateChange':
        fieldinfo.addField(field.name, field.name, 'VISIBLE', '')
```



# Join Tables And Feature Classes

## Extract Data For Cost Analysis Script

```
arcpy.AddJoin_management(in_layer_or_view,  
in_field, join_table, join_field, {join_type})
```

```
arcpy.AddJoin_management(CWR3LogTblView,  
"ID",workGDBParam + "\\ " + CWR3CAITbl,  
"CAI_ID",'KEEP_COMMON')
```


```
arcpy.RemoveJoin_management(CWR3LogTblView, CWR3CAITbl)  
print 'join removed'  
# Inner-join the CAI table to the log table  
arcpy.AddJoin_management(CWR3LogTblView, "ID",workGDBParam  
print 'CAI join done'
```

# Add Fields To Feature Classes

## Extract Data For Cost Analysis Script

```
arcpy.AddField_management(table, field_name,  
{field_type})
```

```
arcpy.AddField_management(workGDBParam + "\\\" +  
CWR3LogTblJoinFiber, "Cost", "DOUBLE")
```



```
arcpy.AddField_management(workGDBParam + "\\\" + CWR3LogTblJoinFiber,  
arcpy.AddField_management(workGDBParam + "\\\" + CWR3LogTblJoinFiber,  
arcpy.AddField_management(workGDBParam + "\\\" + CWR3LogTblJoinFiber,  
arcpy.AddField_management(workGDBParam + "\\\" + CWR3LogTblJoinFiber,  
print 'fields added to the log and fiber table'
```



# Export file gdb tables to DBF

Extract Data For Cost Analysis Script

**arcpy.CopyRows\_management(in\_rows, out\_table)**

arcpy.CopyRows\_management(table, file.dbf)

```
if os.path.exists(CWR3LogTblFolderParam + "\\\" + CWR3LogTblJoinCAI_DBF):  
    print CWR3LogTblFolderParam + "\\\" + CWR3LogTblJoinCAI_DBF + ' already exists'  
    arcpy.AddMessage(CWR3LogTblFolderParam + "\\\" + CWR3LogTblJoinCAI_DBF + ' already exists'  
    arcpy.Delete_management(CWR3LogTblFolderParam + "\\\" + CWR3LogTblJoinCAI_DBF  
    arcpy.CopyRows_management(workGDBParam + "\\\" + CWR3LogTblJoinCAI, CWR3LogTblJoinCAI_DBF)
```



# Data Archiving Tool

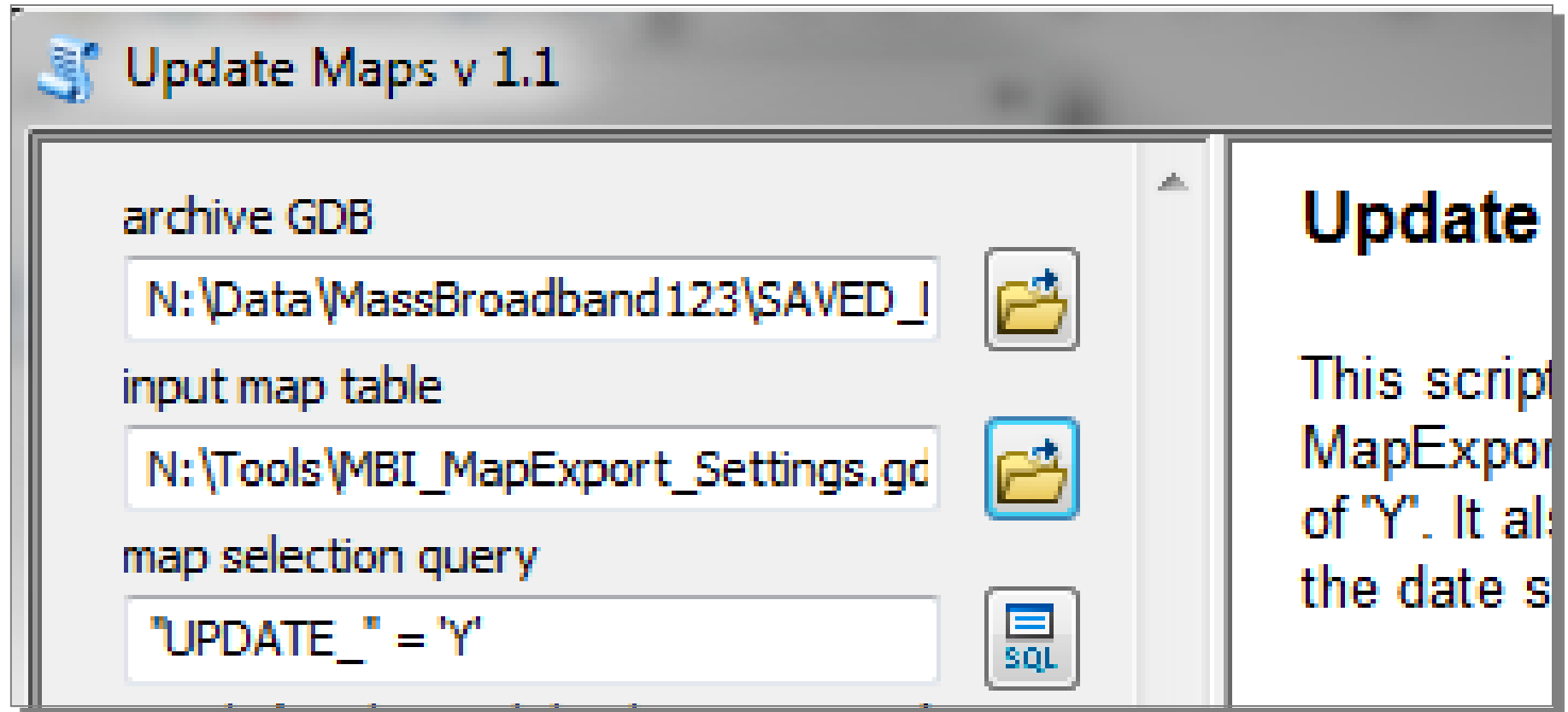
- Set up archive file gdb ahead of time
- Set today's date
- Add today's date to the output feature class when using Feature Class to Feature Class conversion tool

```
now = datetime.datetime.now() # Set today's date
```

```
'CAI_' + now.strftime('%Y%m%d'))
```



# Update Maps Tool



**Update Maps v 1.1**

archive GDB  
N:\Data\MassBroadband123\SAVED\_I

input map table  
N:\Tools\MBI\_MapExport\_Settings.gd

map selection query  
"UPDATE\_" = 'Y'

**Update**

This script  
MapExport  
of 'Y'. It al  
the date s

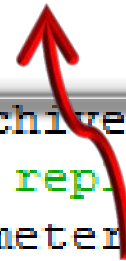


# Replace Data Source

## Update Maps Script

```
layer_name.replaceDataSource(gdb,  
"FILEGDB_WORKSPACE", new source feature class)
```

```
# change  
for CAI  
    if m  
        CAILayer.replaceDataSource(archiveGDBParam,  
        arcpy.AddMessage('data source replaced i  
    elif arcpy.Exists(archiveGDBParameter + "\\'  
        CAILayer.replaceDataSource(archiveGDBPa  
        arcpy.AddMessage('data source replaced
```



# Update Map Elements

Element Name

MapDate

MapDate

Cancel

Apply



0 2.5 5 10 15 Miles

Map Date: 26 April 2012

ARRA Funded Project





# Change Date On Map

## Update Maps Script

```
mapDate =  
arcpy.mapping.ListLayoutElements(map_document,  
{element_type}, {wildcard})[0]
```

```
mapDate = arcpy.mapping.ListLayoutElements(mxd,  
"TEXT_ELEMENT", "MapDate")[0]
```

```
mapDate = arcpy.mapping.ListLayoutElements(mxd, "TEXT_ELEMENT",  
if monthParam == '':  
    newMapDate = 'Map Date: ' + now.strftime('%d') + ' ' + now.
```





# calendar Object

## Update Maps Script

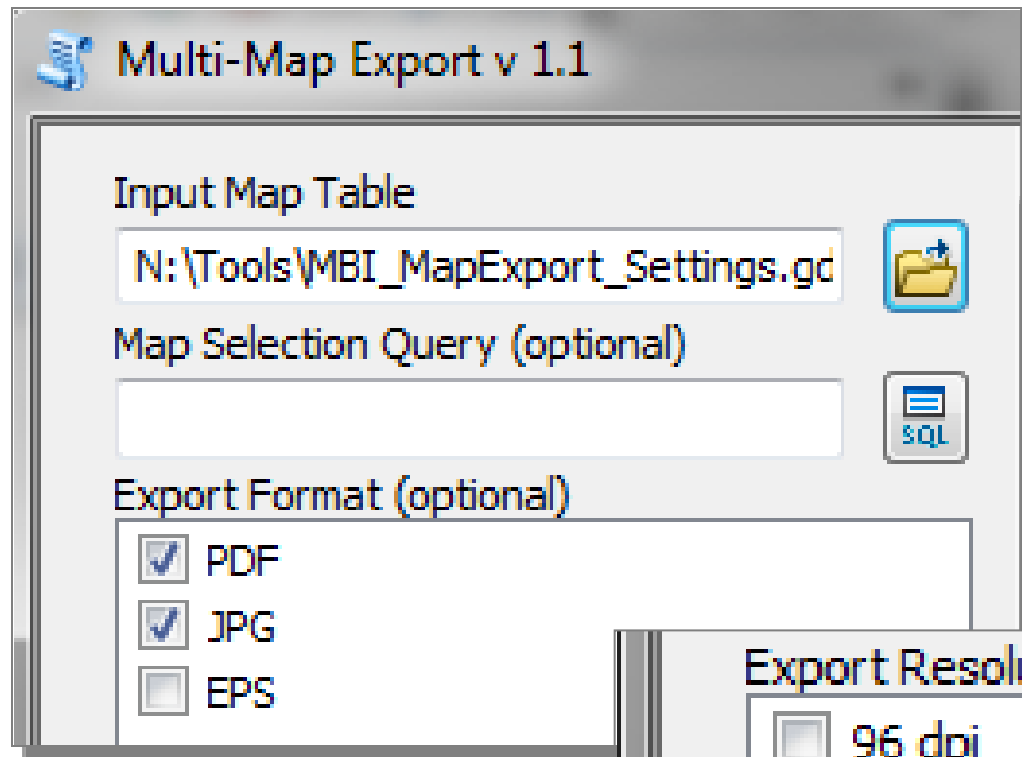
- methods
  - month\_name[]
  - day\_name[]

**calendar.month\_name[int(variable)]**

**str(calendar.month\_name[int(monthParam)])**


```
.List
ate: ' + now.strftime('%d') + ' ' + now.strftime('%B') + ' ' + now.
ate: ' + dayParam + ' ' + str(calendar.month_name[int(monthParam)])
te
'
ate changed')
```

# Multi-Map Export Tool




Multi-Map Export v 1.1

Input Map Table

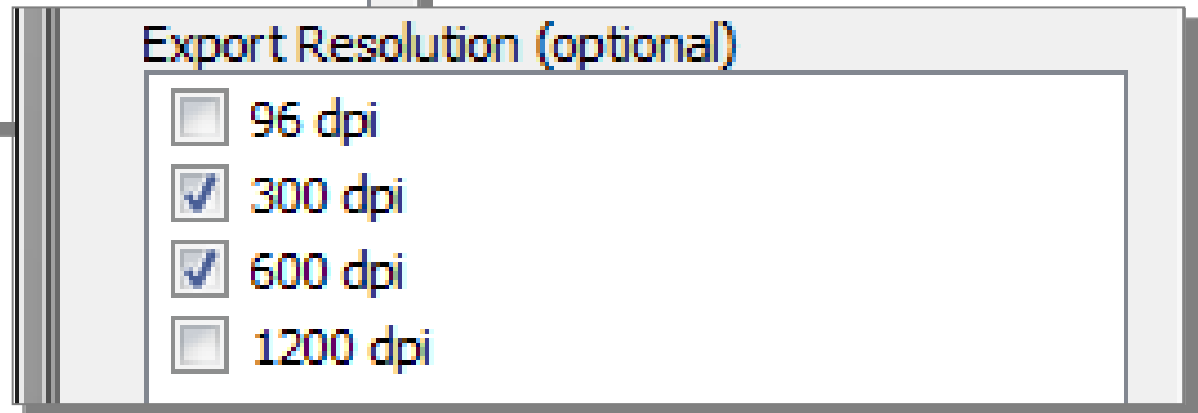
N:\Tools\MBI\_MapExport\_Settings.gd 

Map Selection Query (optional)



Export Format (optional)

- ☒ PDF
- ☒ JPG
- ☐ EPS

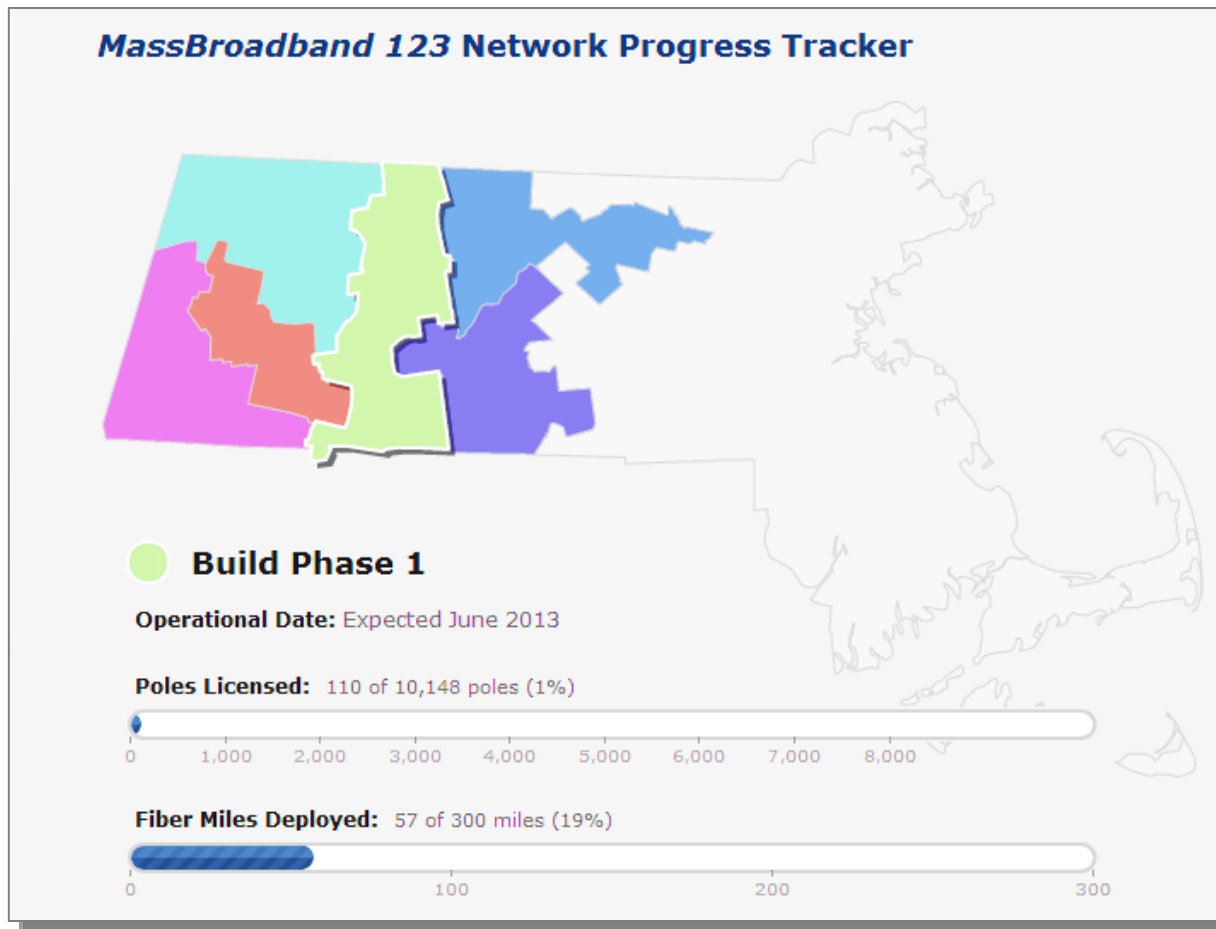


Export Resolution (optional)

- ☐ 96 dpi
- ☒ 300 dpi
- ☒ 600 dpi
- ☐ 1200 dpi

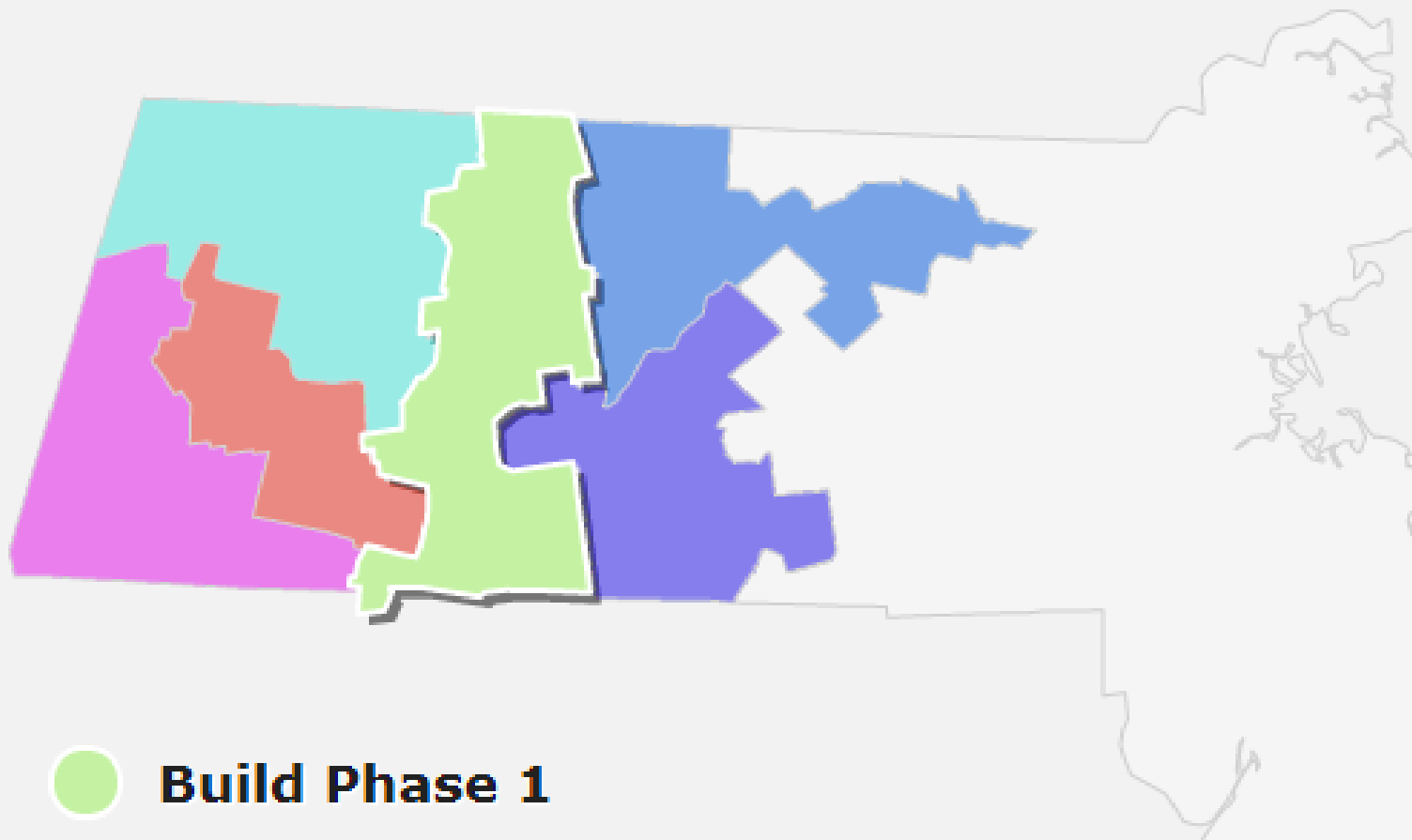
# Online Build Phase Map

[www.massbroadband.org](http://www.massbroadband.org)



# Online Build Phase Map

## *MassBroadband 123 Network Progress Tracker*





# Build Phase Tracker



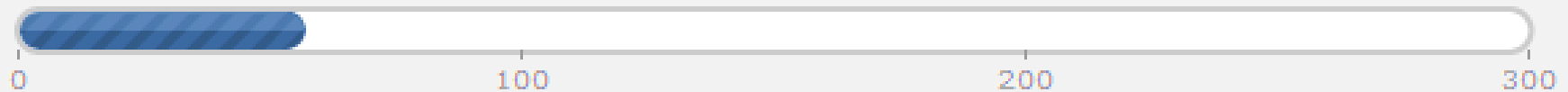
## Build Phase 1

**Operational Date:** Expected June 2013

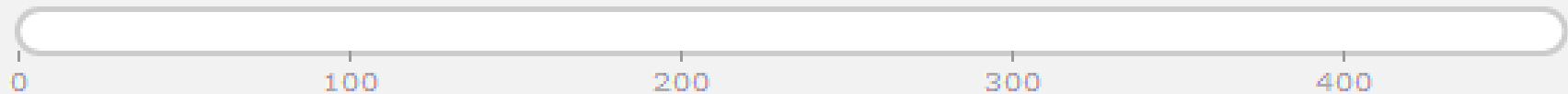
**Poles Licensed:** 110 of 10,148 poles (1%)




**Fiber Miles Deployed:** 57 of 300 miles (19%)



**CAIs Connected:** 0 of 466 CAIs (0%)




# Progress Statistics Tool

 Build Phase Statistics


feature dataset

N:\Data\MassBroadband123.gdb\Net



output folder

N:\Data\MassBroadband123\MBI\_We



map document

N:\Workspaces\IN\_PROGRESS\Network\_Tra

data frame

Layers

Run statistics on the following feature classes:

☒ CAI

☒ Poles

☒ Fiber

## Build Phase Statistics

**WARNING:** The script will create a new file with today's date.

This script creates MB123 network statistics. It calculates the number of CAI and the miles of fiber (planned and installed) using Spatial Join and other network analysis tools.



# Statistics Analysis

## Online Build Phase Tracker Script

```
arcpy.Statistics_analysis(input_data,output_table,  
[[field,"COUNT"]], case_field)
```

```
i arcpy.Statistics_analysis(asset,buildPhaseSJ +  
"_stats",[[assetPrimaryKey,"COUNT"]], BuildPhaseField)  
arcpy.AddMessage(buildPhaseSJ + "_stats" + ".del  
arcpy.Statistics_analysis(asset,buildPhaseSJ + "_sta  
arcpy.AddMessage("Calculated statistics for " + asse
```





# Statistics Analysis Output

## Online Build Phase Tracker Script

G4\$Phase	FREQUENCY	SUM_LenMiles
Build Phase 1	1078	309.6468340
Build Phase 2	407	179.9728937
Build Phase 3	416	141.7151424
Build Phase 5	266	101.8440182
Build Phase 6	450	215.4502940
Build Phase 7	288	122.1585009




# SearchCursor Method

## Online Build Phase Tracker Script

```
rows = arcpy.SearchCursor(table_name)
```

```
rows = arcpy.SearchCursor(buildPhaseSJ + "_stats")
```



```
rows = arcpy.SearchCursor(buildPhaseSJ + "_stats")  
buildPhaseFieldStats = arcpy.ListFields(buildPhase  
statsField = arcpy.ListFields(buildPhaseSJ + "_sta  
arcpy.AddMessage("list field variables defined")
```



# getValue of Row

## Online Build Phase Tracker Script

**row.getValue(field.name)**

```
for row in rows:
```

```
    if row.getValue(myField.name) == "Build Phase 1":
```

```
        i bp1 = row.getValue(statsField.name)
```

```
    elif row.getValue(buildPhaseFieldStats.name) == "Build Phase 2":  
        bp2 = row.getValue(statsField.name)
```

```
    elif row.getValue(buildPhaseFieldStats.name) == "Build Phase 3":  
        bp3 = row.getValue(statsField.name)
```




# Replace Text in File

## Online Build Phase Tracker Script

**`text.replace(placeholder,new_number)`**

```
fileread = open(JSFileTEMPLATE, 'r')  
filetext = fileread.read()  
filetext = filetext.replace("BP1CCT",str(bp5))
```

```
filetext = filetext.replace("BP2CCT",str(bp2))  
filetext = filetext.replace("BP3CCT",str(bp3))  
filetext = filetext.replace("BP5CCT",str(bp5))
```





# Review

- Met Python
- Learned how to create and document tools from Python scripts
- Saw Python script examples, such as defining variables and parameters, handling errors, exporting datasets, adding fields, getting field values, summarizing data and using extra modules such as calendar and os
- Saw Output example online



# Resources

[www.massbroadband.org](http://www.massbroadband.org)

[www.broadbandmap.gov](http://www.broadbandmap.gov)

[www.geospatialtraining.com](http://www.geospatialtraining.com)

<http://docs.python.org/library>

???

E-mail: [grace@masstech.org](mailto:grace@masstech.org)