Using ArcGIS and Python to Web-Enable USAPI Mapping System

2016 ESRI User Conference

Calvin Poulsen
National Drought Mitigation Center

USAPI

Introduction:

- The USAPI is an offshoot of the US Drought Monitor.
- It is expert driven weekly composite drought index for United States Affiliated Pacific Islands.
- It was developed to allow these islands to participate in the US drought relief programs that require USDM drought status.
- The original maps were developed by Richard Heim and he is still the only USDM author producing the USAPI.

USAPI

Continued:

- The original maps were not well balanced and had lots of extraneous information.
- I talked Richard into letting me automate the process.

The process

- The Dx levels are calculated, discussed and finalized for each island
- The Dx levels for each island is entered into the system and submitted.
- The system checks for new or changed entries and creates and emails maps for these entries.

Issues

- Richard wanted both monthly and weekly maps.
- Some islands would drop out of the based on data availability.
- Needed to be point based rather than area based.
- Needed to be accessible at anytime.

Solution

Move the entire operation to a web interface.

Adjust the MXD for all Island combinations.

Move the USAPI data to a separate database.

Add a Scheduled task that runs every 15 minutes.

The Web Interface

Time Scale

(●) Weekly ( ) Monthly

April 26, 2016 ▼

Author

Heim , Richard ▼

Status

Submit Changes

Marshall Islands

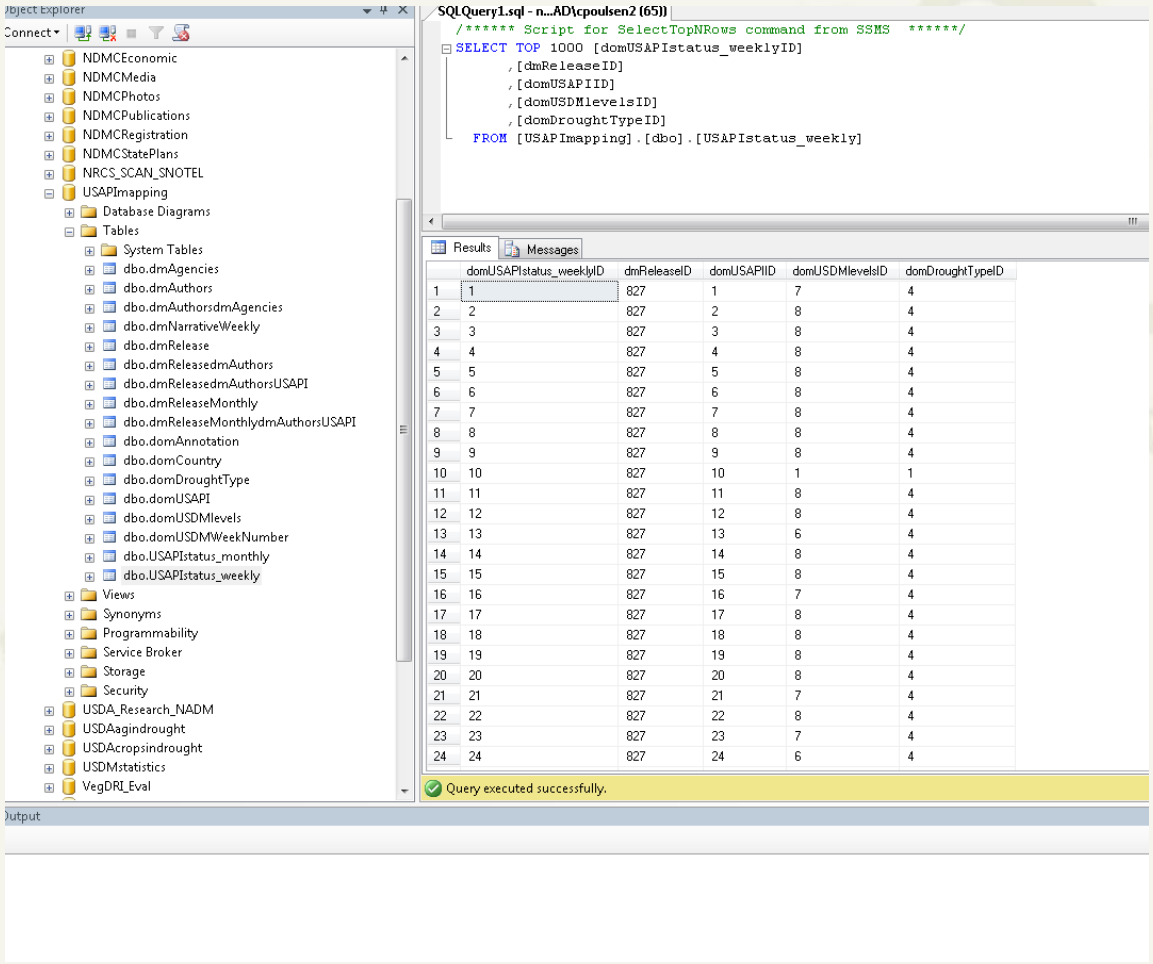
Location	Not Visible	No Data	None	D0	D1	D2	D3	D4	No Impacts	S	L	SL
Ailinglaplap	○	○	○	○	○	○	○	●	○	○	○	●
Jaluit	○	○	○	○	○	○	○	●	○	○	○	●
Kwajalein	○	○	○	○	○	○	○	●	○	○	○	●
Majuro	○	○	○	○	○	○	○	●	○	○	○	●
Mili	○	○	○	○	○	○	●	○	○	○	○	●
Utirik	○	○	○	○	○	○	○	●	○	○	○	●
Wotje	○	○	○	○	○	○	○	●	○	○	○	●

NIDIS

University of Nebraska Lincoln

National Drought Mitigation Center

The Database



NIDIS

University of Nebraska Lincoln

National Drought Mitigation Center

Automation part 1

```
import pyodbc
import sys
# Import Custom Modules
sys.path.append("\\\\ndmc-005\\Production\\USAPI\\scripts\\IndividualShps")
sys.path.append("\\\\ndmc-005\\Production\\PYTHON\\email")
## NOTES
##  This peroidically checks the USAPI database for all mapdates that need to be processed
##  It wil run every 15 minutes 24hrs a day 7 days a week. It will email the resulting maps to R Heim


SqlConnection1 = 'DRIVER={SQL Server Native Client 10.0};SERVER=ndmc-002;DATABASE=USAPImapping;UID=NDMCdataWriter;PWD=password'
SqlConnection2 = 'DRIVER={SQL Server Native Client 11.0};SERVER=ndmc-002;DATABASE=USAPImapping;UID=NDMCdataWriter;PWD=password'

#Make DB Connection
try:
    cnxn = pyodbc.connect(SqlConnection2)
except:
    cnxn = pyodbc.connect(SqlConnection1)
#cnxn = pyodbc.connect(SqlConnection)
cursor = cnxn.cursor()

send_to = ['cpoulsen2@unl.edu','richard.heim@noaa.gov']
message_to = "Richard,\n\n"
message_b_p2 = "   Please contact me if you have any questions.\n\n"
message_from = "Chris Poulsen\n"
message_sig1 = "National Drought Mitigation Center\n"
message_sig2 = "GIS Manager\n"
message_sig3 = "(402) 472-8828\n"
message_sig4 = "cpoulsen2@unl.edu\n"
message_sig_hf = "-----------------------------------\n"

needmappingWeek = cursor.execute("SELECT releaseDate FROM USAPImapping.dbo.dmRelease WHERE (needMaps = 1)").fetchall()
needmappingMonth = cursor.execute("SELECT releaseDate FROM USAPImapping.dbo.dmReleaseMonthly WHERE (needMaps = 1)").fetchall()

if len(needmappingWeek)>0:
    theWattachments = []
    theWdates = []
    theRelease = ""
    wcount = 0
    ## Map each outstanding date
    import CreateUSAPISpecificWeek
    print "There are Weekly maps to process"
    for week in needmappingWeek:
        thedate = week.releaseDate
        dyear = thedate.year
        dmonth = thedate.month
        dday = thedate.day
        theWdate = str(dyear)+"/"+str(dmonth)+"/"+str(dday)
        thepaths = CreateUSAPISpecificWeek.mappit(dyear,dmonth,dday)
        ## Append Attachments
```

Automation part 2

```
if len(needmappingWeek)>0:
    theWattachments = []
    theWdates = []
    theRelease = ""
    wcount = 0
    ## Map each outstanding date
    import CreateUSAPISpecificWeek
    print "There are Weekly maps to process"
    for week in needmappingWeek:
        thedate = week.releaseDate
        dyear = thedate.year
        dmonth = thedate.month
        dday = thedate.day
        theWdate = str(dyear)+"/"+str(dmonth)+"/"+str(dday)
        thepaths = CreateUSAPISpecificWeek.mappit(dyear,dmonth,dday)
        ## Append Attachments
        theWattachments.append(thepaths[0])
        theWattachments.append(thepaths[1])
        ## Append Dates
        theWdates.append(theWdate)
    ## Set the text and attachments for the email
    if len(theWdates)==1:
        theRelease = theWdates[0]
    else:
        for wDate in theWdates:
            if wcount==0:
                theRelease = wDate+", "
                wcount=1
            else:
                theRelease = theRelease+wDate+", "
    ## Set the email message specifics
    subject = "USAPI weekly maps for "+theRelease+" are complete."
    message_b_p1 = "    The USDM for "+theRelease+" complete and the pdf and png files are attached."
    message_signature = message_from+message_sig_hf+message_sig1+message_sig2+message_sig3+message_sig4+message_sig_hf
    message = message_to+message_b_p1+message_b_p2+message_signature
    ## Use the custom module to send the email
    import ndmc_email
    ndmc_email.send(send_to, subject, message, theWattachments)

    for dtw in theWdates:
        #print "UPDATE USAPImapping.dbo.dmRelease SET [needMaps] = 0 WHERE ([releaseDate] = '"+dtw+"')"
        cursor.execute("UPDATE USAPImapping.dbo.dmRelease SET [needMaps] = 0 WHERE ([releaseDate] = '"+dtw+"')")
        cursor.commit()
```

NIDIS

University of Nebraska Lincoln

National Drought Mitigation Center

Automation part 3

```
if len(needmappingMonth)>0:
    theMattachments = []
    theMdates = []
    theRelease = ""
    mcount = 0
    ## Map each outstanding date
    import CreateUSAPISpecificMonth
    print "There are Monthly maps to process"
    for month in needmappingMonth:
        thedate = month.releaseDate
        dyear = thedate.year
        dmonth = thedate.month
        dday = thedate.day
        thepaths = CreateUSAPISpecificMonth.mappit(dyear,dmonth,dday)
        theMdate = str(dyear)+"/"+str(dmonth)+"/"+str(dday)
        ## Append Attachments
        theMattachments.append(thepaths[0])
        theMattachments.append(thepaths[1])
        ## Append Dates
        theMdates.append(theMdate)
    ## Set the text and attachments for the email
    if len(theMdates)==1:
        theRelease = theMdates[0]
    else:
        for mDate in theMdates:
            if mcount==0:
                theRelease = mDate+", "
                mcount=1
            else:
                theRelease = theRelease+mDate+", "

    ## Set the email message specifics
    subject = "USAPI monthly maps for "+theRelease+" are complete."
    message_b_p1 = "    The USDM for "+theRelease+" complete and the pdf and png files are attached."
    message_signature = message_from+message_sig_hf+message_sig1+message_sig2+message_sig3+message_sig4+message_sig_hf
    message = message_to+message_b_p1+message_b_p2+message_signature
    ## Use the custom module to send the email
    import ndmc_email
    ndmc_email.send(send_to, subject, message, theMattachments)

    for dtm in theMdates:
        #print "UPDATE USAPImapping.dbo.dmReleaseMonthly SET [needMaps] = 0 WHERE ([releaseDate] = '"+dtm+"')"
        cursor.execute("UPDATE USAPImapping.dbo.dmReleaseMonthly SET [needMaps] = 0 WHERE ([releaseDate] = '"+dtm+"')")
        cursor.commit()
```

NIDIS

University of Nebraska Lincoln

National Drought Mitigation Center

Mapping part 1

```
# ---------------------------------------------------------------------------
# CreateUSAPICurrentWeek.py
# Created on: 2015-02-25
# Description: Runs all of the mapping python code for the current USAPI USDM
# ---------------------------------------------------------------------------

# Import arcpy module
import ...
# Import Custom Modules
#sys.path.append("\\\\seca\\f\\OperationalProcessing\\PYTHON\\usdm_mapping")
import CreateUSAPIShp
import CreateUSAPIMap
#sys.path.append("\\\\seca\\f\\OperationalProcessing\\PYTHON\\usdm_processing")
#import usdm_daylight

SqlConnection1 = 'DRIVER={SQL Server Native Client 10.0};SERVER=ndmc-002;DATABASE=USAPImapping;UID=NDMCdataWriter;PWD=password'
SqlConnection2 = 'DRIVER={SQL Server Native Client 11.0};SERVER=ndmc-002;DATABASE=USAPImapping;UID=NDMCdataWriter;PWD=password'

#Make DB Connection
try:
    pyodbc.connect(SqlConnection2)
    SqlConnection = SqlConnection2
except:
    pyodbc.connect(SqlConnection1)
    SqlConnection = SqlConnection1
#Make DB Connection
cnxn = pyodbc.connect(SqlConnection)
cursor = cnxn.cursor()

FirstAuthorAffil = ""
FirstAuthorName = ""

#Date (YYYY, M, D)
#cursor.execute("select MAX(releaseDate) As relDate from dmRelease where isActive = 'true'")
cursor.execute("select MAX(releaseDate) As relDate from dmRelease where isUSAPIweek = 1")
maprows = cursor.fetchone()
mapdate = maprows.relDate
print mapdate
# This will set the daylight status
#theDays = usdm_daylight.dates(mapdate.year)
#if theDays[0] < mapdate < theDays[1]:
#    theDSTstatus = "EDT"
#else:
```

Mapping part 2

```
# Global Variables
InputDir = "\\\\ndmc-005\\Production\\USAPI\\data\\weekly"
MappingDir = "\\\\ndmc-005\\Production\\USAPI"
OutputDir = "\\\\ndmc-005\\Production\\USAPI\\data\\weekly"
BaseDir = "\\\\ndmc-005\\Production\\USAPI"
FileDate =  str(mapdate.year) + str("%02d" % (mapdate.month,))  + str("%02d" % (mapdate.day,))
ReleaseDate = str("%02d" % (mapdate.month,)) + "/" + str("%02d" % (mapdate.day,)) + "/" + str(mapdate.year)

PdfPath = "pdf"
PngPath = "png"
JpgPath = "jpg"
SvgPath = "svg"
TifPath = "tif"
EmfPath = "emf"
AiPath = "ai"
ShpPath = "shp"

#Check to see if folders exist and create if necessary
a = os.path.dirname(OutputDir + "\\" + PdfPath + "\\" + FileDate + "\\")
if not os.path.exists(a):
    os.makedirs(a)

b = os.path.dirname(OutputDir + "\\" + PngPath + "\\" + FileDate + "\\")
if not os.path.exists(b):
    os.makedirs(b)

c = os.path.dirname(OutputDir + "\\" + ShpPath + "\\" + FileDate + "\\")
if not os.path.exists(c):
    os.makedirs(c)

#Get the date from db
cursor.execute("select * from dmRelease where releaseDate = '" + ReleaseDate + "'")
weekrows = cursor.fetchone()

# Get Date Values
#mapdate = weekrows.releaseDate
rel_id = weekrows.dmReleaseID

#Get the Annontation ID
ann_id = str(weekrows.domAnnotationID)
```

NIDIS

University of Nebraska Lincoln

National Drought Mitigation Center

Mapping part 3

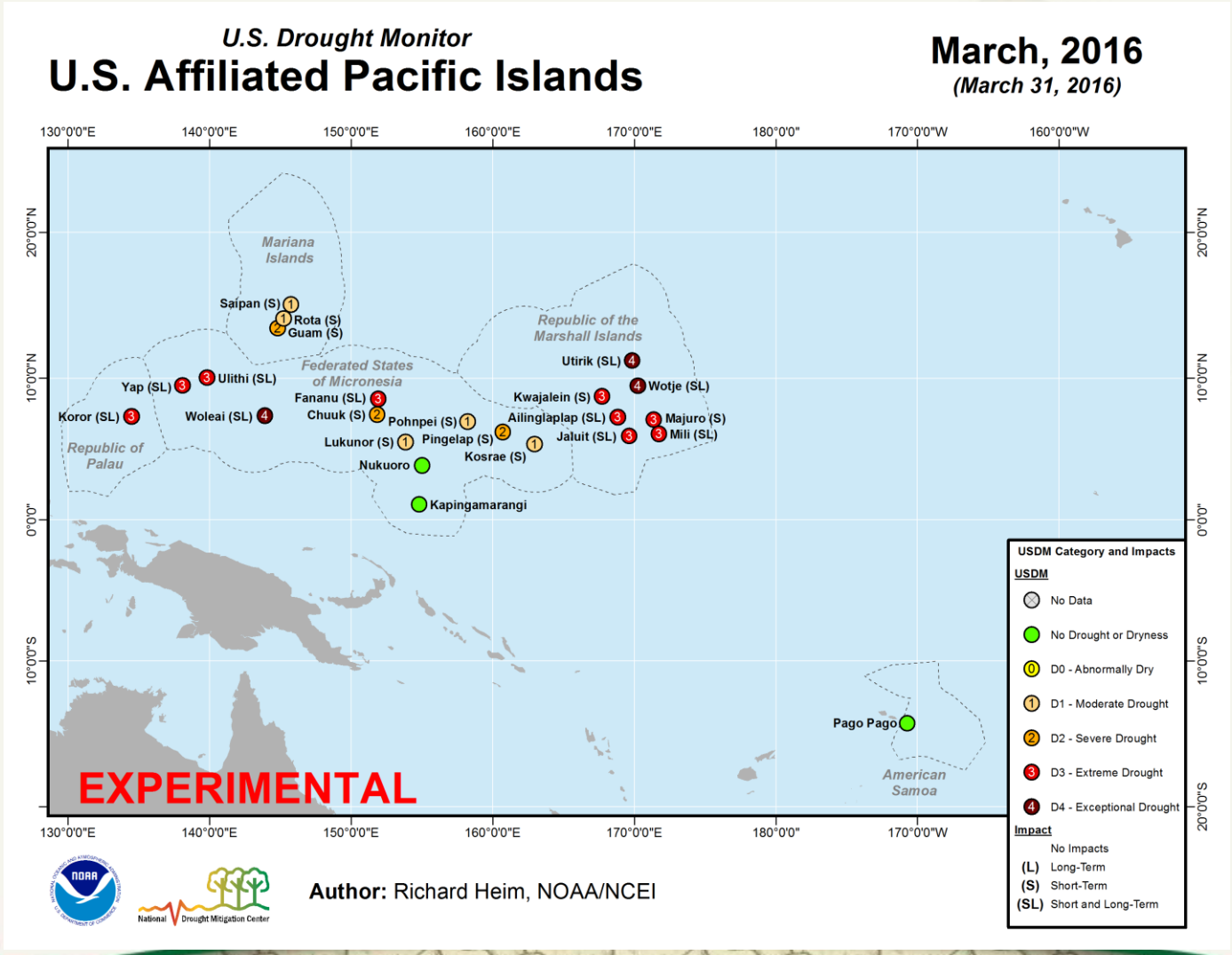
```
#Get Map Release Date (Check for released on Wed.)
if (weekrows.releasedWednesday == 1):
    releasedate = weekrows.releaseDate + datetime.timedelta(days=1)
else:
    releasedate = weekrows.releaseDate + datetime.timedelta(days=2)

cursor.execute("select * from dmReleasedmAuthorsUSAPI where dmReleaseID = "+str(weekrows.dmReleaseID))
a_rows = cursor.fetchall()

for row in a_rows:
    if (row.authorOrder == 3):
        ThirdAuthorID = row.dmAuthorsID
    elif (row.authorOrder == 2):
        SecondAuthorID = row.dmAuthorsID
    else:
        FirstAuthorID = row.dmAuthorsID


if (FirstAuthorID > 0):
    cursor.execute("select first, last from dmAuthors where dmAuthorsID = " + str(FirstAuthorID))
    fa_rows = cursor.fetchone()
    FirstAuthorName = fa_rows.first + " " + fa_rows.last
    FirstAuthorName = FirstAuthorName.strip()
    cursor.execute("select fullName, acronym from dmAgencies join dmAuthorsdmAgencies on dmAgencies.dmAgenciesID = dmAuthorsdmAgencies.dmAgenciesID where dmAuthorsID = " + str(FirstAuthorID)+ "order by agencyOrder asc")
    aa_rows = cursor.fetchall()
    if (len(aa_rows) > 1):
        for row in aa_rows:
            FirstAuthorAffil = FirstAuthorAffil + "/" + row.acronym
        FirstAuthorAffil = FirstAuthorAffil[1:]
    else:
        for row in aa_rows:
            FirstAuthorAffil = row.fullName

# Close the DB Connection
cnxn.close()

cweek = "true"

##if (not os.path.isfile(OutputDir + "\\" + ShpPath + "\\" + FileDate + "\\" + FileDate + "_usapi.shp")):
##    CreateUSAPIShp.create(OutputDir,mapdate,FileDate,ShpPath,rel_id,"w")
##else:
##    print "Shapefile already exists"
CreateUSAPIShp.create(OutputDir,mapdate,FileDate,ShpPath,rel_id,"w",SqlConnection)
CreateUSAPIMap.mapping(InputDir, ShpPath, OutputDir, MappingDir, PdfPath, PngPath, mapdate, cweek,"w", FirstAuthorName, FirstAuthorAffil)
```

NIDIS

University of Nebraska Lincoln

National Drought Mitigation Center

Current Monthly

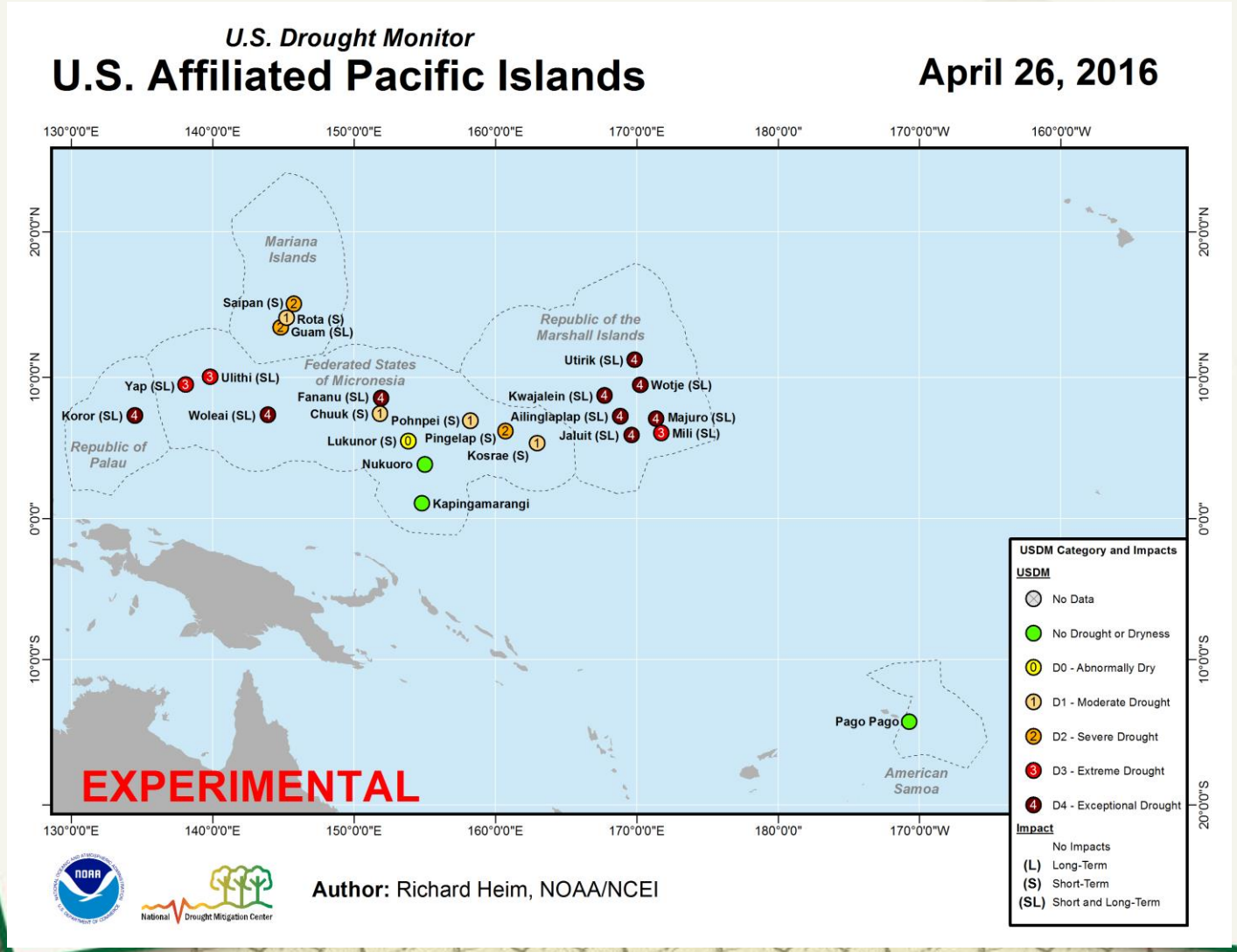


NIDIS

University of Nebraska Lincoln

National Drought Mitigation Center

Current Weekly



NIDIS

University of Nebraska Lincoln

National Drought Mitigation Center

Questions?

Chris Poulsen, GIS Manager
National Drought Mitigation Center
Email: cpoulsen2@unl.edu