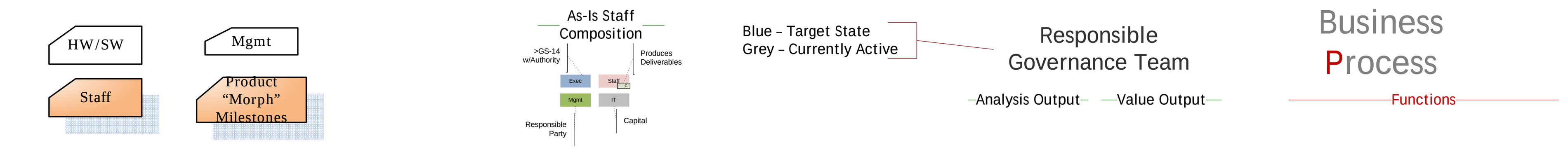
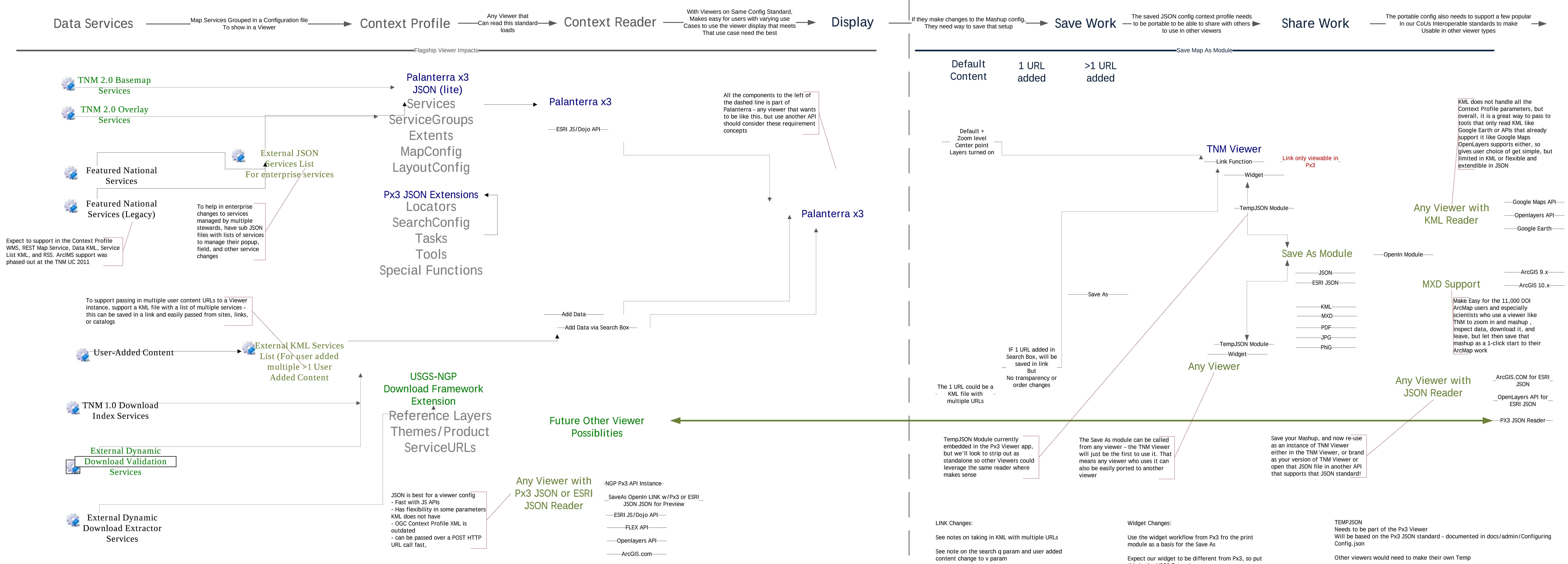
USGS NGP As-Is Delivery Information Flow :
Context Profile v2 Delivery Products & Services

Below is the Target Flow Diagram for NGP Context Profile for how can be opened in or saved for use in other viewers. This particular diagram is directly attributable to the USGS National Geospatial Program business context associated with Delivery Services, but considers very directly how can be done in other viewers/programs. The associated description provides the relevant detail for the business context associated with the information flow. Project History, background, and drivers are offered in the lower left of the diagram. This ultimately captures the process and some data requirements for being implemented in an application architecture, but this does not include the application nor technical architecture

The diagram is accompanied by a annotations across the flow that provides a description of steps, process description and additional notes for the business context associated with the information flows. Steps provide a logical ordering of the processes that comprise the business context. Process descriptions provide detail of the activities related to the information flows described in the diagram.





Px3 JSON

Physical Save the Temp JSON
Allow for user download
Temp file accessible Good for 10 minutes
Expecting many NGP Instances coming in FY12 and FY13
Expecting USGS, DOI, and other Partner externally hosted instances to come in FY13 and FY14

Initially, v18 config schema will be your input into TempJSON that your add changes to and generate as output. V19 WILL CHANGE!!! V19 will have the base based ESRI JSON and include Px3 JSON extensions (like service, service groups, intial extent… ALL will change to the ESRI REST Open Specification) – BUT the Px3 JSON extensions should remain the same or is irrelevant anyhow

ESRI JSON

Physical Save the Temp JSON
Without the Px3 extensions as documented on left
Just strip them out
This should be readable by ArcGIS.com and the AGS REST SPEC
Some Transforms may need to occur – if so, this would add time
Ultimately, this would be the JSON based for a standard if OGC, OpenLayers, AGS, and other USGS/DOI viewers were to adopt a JSON standard
In future, the Px3 Module would need a ESRI JSON detector and reader
NGP EA discussing with ArcGIS.com for how to come up with ESRI JSON Spec updates

Sciencebase.gov Early FY12

ArcGIS.com REST SPEC is published

LINK w/JSON
(Preview)

This is the Open In module where a user can select to preview in Px3, Google Maps, or OpenLayers

Open in Px3 using Temp file
Create a quick ESRI JSON Reader for Google Maps
Create a quick ESRI JSON Reader for OpenLayers

KML

Likely need to reorder the Services in ServiceGroups by KML Read order
May need to research when the services are read to be turned on to assure KML order is OK
Should read in annotations
Need to make a basemap group and the overlays groups - Look at what each API supports as far as showing the basemap as a group in Table of Contents or as buttons
Consider how the metadata popups work in KML to show the metadata URL, legend URL
Do not look to harvest the metadata or legend URL, lete the KML API handle that onthefly
KML should be readable by Google Maps KML API
KML should be readable by Google Earth KML API
KML should be readable by Px3 KML API via Search/Add Data – if reading in for Px3, only the user added content will be read using the same code for passing in a multiple URL
Currently the KML output in Google Earth and Px3 has view issues
Reference the Px3 documentation on the KML API support for Px3 and Google Maps - http://viewer.nationalmap.gov/help/KML%20Support%20in%20Palanterra%20X3.htm

LINK Changes:

See notes on taking in KML with multiple URLs

See note on the search q param and user added content change to v param

Consider how to pass the metadata attributes fomr a user added content – pretty name, metadata URL, SOAP URL, transparency, order

If not easy, don't do, add FAQ that is user needs this, they need to do a Save As. Repeat, this is low priority for link, and the main purpose of the Save As

Widget Changes:

Use the widget workflow from Px3 fro the print module as a basis for the Save As

Expect our widget to be different from Px3, so put this in the USGS Extensions

Have the widget options be driven by the default.json SaveAsConfig parameters on which options are turned on (JPG, PNG, PDF, KML, JSON, MXD, AGS.com JSON, etc.). Currently this is a pdfconfig in the extract.json

TEMPJSON
Needs to be part of the Px3 Viewer
Will be based on the Px3 JSON standard – documented in docs/admin/Configuring Config.json

Other viewers would need to make their own Temp

JSON output and pass to Save Map which is standalone API for any viewer to call

Temp JSON needs to copy the default.json into memory then attempt to gather all user changes such as:
SOURCE JSON UPDATES:
Map Setup
New initial extent, initial level zoom, center point and Update mapconfig and layoutconfig attributes
Update which service group is initially enabled/expanded
Read in the projection from initialbaseserviceid to avoid any hardcoding for webmerc or wgs84
Service Setup
Get and updated list of what services are on (initialon) and what may be turned OFF!
Get and updated list of what layers are on and what may be turned OFF! Sometimes, some layers may not be explicitly called out, and this will need to be added if the layer is now visible on
Get and update which is the basemap group that is initially turned on

SOURCE JSON ADDS:
Read in all user added content as well as the service attributes
Read in all user added content for annotations and what is visible

Leave the Px3 additional variables alone at this time and keep in the target temp json, but likely nonPx3 viewers will ignore.

Likely this temp JSON module will need to be a USGS Extension, overriding the default Px3 which is more embedded for pure Print Module.

This code could be offered back up to Px3 team, but they may not accept.

In the target state requirements (Should be loaded into JIRA as descriptions for existing items or subtasks for newer detail)

SERVICE TYPE IN JSON CHANGES:

Having services in a default JSON that works for REST, WMS, KML, RSS in the Px3 JSON Reader.

Not all APIs will likely handle all of these, and that's fine, but we need it in ours.

Projection support depends on the viewer, but usually viewers in WebMercator will also support WGS84 calls with a reproject on the fly call

Issues with supporting WMS, KML, and RSS needs to be coordinated with the Px3 Team

JSON INPUT CHANGES:
Currently, the only JSON file that can be read are JSON files on the server and passed via p or profile param

There should be a way to pass into the URI an external JSON.

At this time, crossdomain.xml should garner what domains are allowed (like .gov, .mil, .doi.net, etc.)

Should consider if external, so we need an additional run-time lint check or a parameter where the user at design-time gets a checksum from a JSON lint and we check the checksum result which has validated it's a good JSON

=========
Support sub-JSON enterprise files from multple stewards

These should also have a checksum pre-check to assure the sub-JSONs are validated

This code would go back to Px3 as an offering, though they may not accept

These files could also be on external URLs, based on crossdomain OK.
============
Download Framework also needs to support Sub-JSONs, checksum, and external URLs

KML INPUT CHANGES:

Current, we can take a KML with data, but not a KML with list of services, which can be added to user added content.

Services should be OK to be in a KML

This is likely code that would go back to Px3.

The search box and add data should be able to handle this. Currently 2 separate code modules, but same code

Should handle the 4326 to Web Mercator conversions

The Tree in the User Added content should start with the KML file, then the children are the services and layers are harvested on read in after loading from add data/search box

SEARCH BOX AND ADD DATA CHANGES:

We need to support saving the first URL under v param and the last search in the link under q param.

Currently can do one or the other because q is used for both.

As well, possibly see if Multiple userr added content URLs can be added this way in the V param.

This code should be offered back to Px3

Also Search and Add data should be able to handle the same 4326 to webmerc conversion

This conversion should be for not just WMSs, but REST, KML, and RSS

Look to add to add data and search the fuzzy pattern logic to lookup the SOAP URL and metadata URL based on known patterns for REST, WMS, etc. – try quickly and if works, update that JSON param, otherwise, skip

We expect catalogs like sciencebase.gov, data.gov, and others to use this as a preview technique for passing in a mashup context with 1 URL or 1 URL for a KML file with multiple URLs. Already have beta tested this in sciencebase.gov and a Biological GAP/Land Cover Inventory Catalog of Parks.

Px3 Baseline:

Expect to code against Px3 v18

Likely we will look at replacing the Print Module with the New Save as. Module

Move the current v17 print module as the PDF, PNG, JPG base for the new save as module

When v18 new print comes out, consider what fixes would benefit to bring over

Consider reusing and copying as much code as makes sense

The save as module should be standalone with the exception of the widget which should go under the USGS module extension area.

Save As should be offered back up to NGA, but not expected to be adopted

Also, as noted, any changes to search box, add data, KML Input, external json, multiple sub-json support should be considered to offer back to Px3

		tnmdev	tnmbeta		nationalmap.gov
Source *RDj		Happy Path	Functional	Non-Functional/Load	Live
	Widget		8/15/2011		EOFY
	Refactoring to SaveAs Module		EOFY		EOFY
	TempJSON Generator		8/15/2011		EOFY
	TempJSON w/Annotations				
	Refactor TempJSON Generator				
Px3 Reader* With Px3 v18 Team and FA					
	Multi-URL KML				
	Sub JSON				
	External JSON URL (Xdomain)				EOFY
	Search/Add Data Updates				
	Px3 v18 Code Review/Merge		Date?		EOFY?
Target *RDj					
	Px3 JSON		8/15/2011		EOFY
	JSON Lite				
	KML		8/15/2011		EOFY
	MXD				
	PDF		8/15/2011		EOFY
	JPG		8/15/2011		EOFY
	PNG		8/15/2011		EOFY
Open In *RDj plus CDI Viewer Teams					
	Preview JSON Px3				EOFY
	Px3				
	Gmaps				
	Gearth				
	Flex API (WIM)				
	Open Layers (Reused CIDA)				

MXD
Investigate using TempJSON Java output as input to MXS using Jython (Java Python bridge) or similar. Otherwise, need a 2nd tempJSON generator in Px3
Needs to handle generating LYR files to be embeded in MXD
Generatign LYR files currently only tested for REST, need to do from WMS, RSS, KML
Once LYR files generated – using Java, then pass the tempJSON and LYR files to ArcPy to build MXD
Use CDI code to build MXD as base, and refactor
Need to handle reordering like PDF, and KML

PDF
Refactor code into a SaveAs Module
Separate the TempJSON source generator leaving that in the Px3 and PDF part of SaveAs Module
Need to take old USGS Print module code, examine, and bring support to handl the Px3 bugs such as WMS, KML, RSS support as well as the byte type setup for handling different image types
Fixed Transparency in the USGS Module already
Fix Order in the USGS Module
Consider when Px3 v18 comes out, what code should/could be brought in
PDF Layer Control in Future

JPG
Refactor code into a SaveAs Module like PDF
Leave Alone until Px3 v18

PNG
Refactor code into a SaveAs Module like PDF
Leave Alone until Px3 v18

Frqwh{w#surilh#edfnjurxqg=

Wkh#XVJ V#surylghv#p dq|#qdwlrqdo#hjlrqdo#dqg#orfdo#gdwdvhwv#iru#grz qordg/#vwuhdp lqj# lqwhjudwlrq#vxfk#dv#Z I V2Z FV/#dqg#dqdo|vlv#Xulp dwho|/#p rvw#gdwdvhwv#duh#suhvhqwhg#iru# ylvxdd}dwlrq#lq#Èylhz huvŠ#z klfk#dvlf#qdyljdwlrq#dqg#lqwhudfwlrq#iru#lqvshfwlrq#dqg#hyhq# dj kwz hljkw#Z heJ LV#dnh#z he#vhuylfh#ixqfwlrqv#dqqrwdwlrqv#hwf1#P dq|#ylhz huv#µ gliihuhqw# DSLv/#fdhqw#sxursrvhv#dqg#qlfkh#ixqfwlrqv#µ duh#lqyhvwhg#lq#dw#XVJ V#dqg#GRL#dqg#wkh# z krdh#I hghudo#J ryhuqp hqw1

Wkh#vroxwlrq#lv#qrw#È4#ylhz hu#ru#È4#ylhz hu#DSLŠ0vhh#wkh#ÈYlhz hu#H{ sorvlrq#Frqxqgurp Š# wr#wkh#uljkw#Z h#duh#vxfn#lq#d#p xowlsch#ylhz hu#hqylurqp hqw#z h#fdqqrw#hfrp p hqg#ihz # DSLv/#dqg#hqvwuf#wr#vkhu#dw#ehvw#Wkh#surehp #z lwk#wklv#lv#wkdw#z khq#vrp hrqh#jrhv#wr#d#qhz # ylhz hu#wkh|#grqÑw#doz d|v#kdyh#wkh#vdp h#edfngursx#edvhp ds#ru#frp p rq#dvvhw#ri#yhuvdo|v#dqg# p rvw#ylhz huv#grqÑw#kdyh#fdsdeldw|#wr#dgg#vhuylfhv#lqwr#wkh#ylhz hu#h{fhsw#wkfn#fdhqw#dqg# Wkh#Qdwlrqdo#P ds#Hvvhqwldo|#wkhuh#lv#qr#z d|#iru#d#xvhu#wr#mxp s#edvlq#ehwz hhq#ylhz huv1

Lq#wkh#hdu|#5333v/#RJ F#vsrqvruhg#wkh#Z he#P ds#Frqwh{w#Surilch#{ P O#vwdqgdug1#Wkh#ghd# z dv#kdyh#d#vwdqgdug#frqilj xudwlrq#ilch#z lwk#wkh#qlwldo#ylhz #hqwv#iru#edvhp ds/#XUOv#iru# dhj hqgv#p hwdgdwd#dqg#z he#p ds#z klfk#p ds#z huh#wxuqhg#rq#e|#ghidxow#z kdw#vkhu#yhuvdo|# Z P Vv#duh#ydlodedh#lq#d#slfnh#dvv#z kdw#lv#wkh#ghidxow#vwdwh#ruhu#wudqvsduhqf|#dqg# jurxslqj#Ii#rx#p dgh#d#p ds#wxuqhg#d|hu#rq2rii/#fkdqjhg#ruhu#wudqvsduhqf|#/#rrp hg# sdq/#|rx#frxog#wkhq#vdyh#|rxu#qhz #p dsxsŠ#dv#w#fdohg#qrz #dv#lq#RJ F#Frqwh{w#Surilch# iru#dwhu#rq#wkdw#vlwh/#ru#ii#qrqvkhu#vlwh#vxssruwhg#uhdglqj #wkdw#vwdqgdug/#frxog#mxvw#orp s#lw# lqwr#dqrwkhu#ylhz hu1

Frqwh{w#surilch#edfnjurxqg#frqwlqxhg,=

J uhdw#frqfhsw#exw#glgqÑw#wdnh#rii#Rq#*Jhrsvdwldo#Rqh0Vwrs*#dqg#vsruwv#vkhuv#rq#d#dujh#vfdoh#surgxfhg# vkh#ilch#Vlqfh#wkhq/#doo#ylhz huv#vxssruw#wkh#vxsvhw#wr#p dnh#wkhlu#rz q#frqwh{w#dnh#Rs hqOd|huv#HVUL#MV# ru#I dh{#DSL/#Jrrjoh#P ds#Elqj#P ds#dqg#p rvw#dohuq#dwlyh#ylhz huv#Mdydvfulsw#Rebhfw# Qrwdwlrq,#dv#wkh#frqwh{w#surilch#dqj xdjh#ri#fkrlfh1#Exw#p rvw#ghyhorshuv#duh#fkrrvlqj#wkhlu#rz q# frqwh{w#surilch#vwdqgdug1#Wklv#lv#yhu|#vslfdo#iru#ghyhorshuv#wr#gr#z lwkrxw#d#vwdqgdug#wr#uhihu#wr1

Lq#533;/#QJ D#Sdodqwhud#{6#z hqw#wkh#gluhfwlrq#dqg#qlfhd|#grfxp hqwhg#wkhlu#frqilj xudwlrq#MVRQ1# XVJ V#Wkh#Qdwlrqdo#P ds#iroorz hg#vxlw#xvlqj#wkdw#dqg#h{whqglqj#z lwk#d#grz qordg#iudp hz run#HVUL#dv#d# frusrudwlrq#fdp h#xs#z lwk#d#55<0sdj h#dsshqgl{#ri#wkh#UHVW#vshflilfdwlrq/#qrwlqj#d#frqilj xudwlrq# vfkhp d#dqg#z dv#uhdhdvhg#ghidxow#z lwk#I dh{/#HVUL#MV#DSL/#dqg#DufJ LV#frp #Lwv#vlp lodu#wr#wkh#S{6# lp sdp hqwdwlrq/#exw#wkhuh#duh#gliihuhqfhv#XVJ V#dvnhg#RshqJhr#RshqOd|huv#ii#wkh|#z rxog#eh#ks#iru#d# MVRQ#frqwh{w#surilch#vwdqgdug/#dqg#wkh|#vdlg#wkh|#z huh#lqwhuhvwhg#lq#frooderudwlqj#dv#glg#HVUL# DufJ LV#2#Uhvw#Vshf#Whdp 1

Dv#z hoo#XVJ V#Frp p xqlw|#iru#Gdwd#Lqwhjudwlrq#z klfk#eulqjv#wrjhwkhu#p rvw#ylhz hu#ghyhorshuv#dqg# p dqdjhuv#lq#I\ 44#dj uhhg#wr#slorwlqj#d#frqwh{w#surilch#vkdulqj#frqfhsw#dqg#lq#I\ 45#dj uhhg#wr# lqyhvwljdwlqj#lq#dgrswlqj#d#vlqjoh#MVRQ#vwdqgdug#iru#frusrudwh#ylhz huv#dqg#hydqjhd}lqj#dqg#vkdulqj# frgh#wr#vxssruw#ru#surp rwh#ylhz huv1

J lyhq#wklv#p rp hqwxp /#wkh#dufklwhfwxuh#surs rvdo#wkhuh0lq#lv#wkh#vwduwlqj#edvlv#iru#wkh#frqyhuvdwlrq#z lwk#d# jrdo#ri#qrw#Èelj#edqj Š#frp sdwlaw#vwdqgdug#iru#doo#sduwv#ri#wkh#frqwh{w#exw#d#frh#sduw#wkdw#v#xvhg# frp p rq#e|#doo1

Ylhz hu#H{sorvlrq#Frqxqgurp #h{dp soh

XVJ V#dorqh#kdv#433v#ri#ylhz huv#iq#ydu|lqj#gr}hqv#ri#dj df|#dqg#dqg#gr}hqv#ri#fxuuhqw# ylhz lqj #sodwirup v#gr}hqv#ri#krp hjurz q#Mdyd#ru#Mdydvfulsw#DSLv/#p xowlsdh#kdug0frghg# J rrjoh#P dsv#wrp h#Elqj#P dsv#dqg#vhyhudo#dovruw#ri#dj df|/#RshqOd|huv#ydu dwlrqv#lq# HVUL#MV2GRMR#DSL#Wkh#Qdwlrqdo#P ds#Sdodqwhud#H{whqvlrq#wr#MV#DSL#I dh{/#Vloyhuljkw# dqg#dj df|#HVUL#ylhz huv#Dv#z hoo#wkh#frp p xqlw|#ri#wklfn#fdhqw#5‡8#glp hqvlrq#ylhz huv# vxfk#dv#J rrjoh#Hduwk#dqg#wkhuh#duh#z hoo#rxhu#43/333#DufP ds#dfhqvhv#dw#GRL#dqg#d# p l{hg#dxglhqfh#ri#HVUL#DufJ LV#h{sorudhu#dqg#d#jurz lqj#lqwhuhvw#lq#DufJ LV1frp

Wklv#h{sorvlrq#ri#ylhz huv#frxog#eh#ylhz hg#dv#d#odfn#ri#jryhuqdqfh#glvflsdqh#iru#XVJ V# iru#p dqdjlqj#vriwz duh#whfkqrorj|/#exw#dfkx#doo|#wklv#lv#dq#rssruwxqlw|#iru#xvhuv#dqg#dssolfdwlrq#z lwk# xqltxh#qlfkh#ixqfwlrqv#wr#wkdw#surmhfw#I lh#d#vshflilf#z dwhu#sxulw|#dqdo|vlv#ixqfwlrq/#d# vshflilf#elrwd#lghqwlilfdwlrq#ixqfwlrq/#hwf1#Z h#gr#qrw#qhhg#rqh#ylhz hu#dv#z hŠ#fuhdwh#d#z lvv0 dup |#nqlih#vfhqdulr1#Dv#z hoo#4#ylhz hu#whfkqrorj|#grhv#qrw#p dnh#vhqvh#dv#wr#ghyhorp|# vrp hwklqj#dv#uhxvdeoh/#z lwk#hqwhusulvh#vxssruw#dv#d#wr#ri#yhukhdg#dqg#vxvwdlq#jdlqv#ri# vxfk#wdnhv#p dq|#|hduv1#Ylhz hu#Whfkqrorjlhv#fkdqjh#hyhu|#45#wr#4;#p rqwkv/#vr#wklv# lqyhvwp hqw#z rxog#eh#kdug#wr#jhw#wkh#uhxvdeoh#sd|edfn1#Dovr/#jlyhq#wkh#idvw#wxuqdurxqg#ri# ylhz huv/#shrsoh#whqg#wr#edu#z he#ylhz hu#txlfno|1#

Dovr/#vrp h#ylhz hu#DSLv#ru#fdhqw#duh#ehwwhu#iru#gliihuhqw#vlwxdwlrqv#µ MV#DSLv#duh#jrrg# iru#jhqhudo#ylhz huv#duh#qrw#sox j0lqv#duh#hwwhuhg/#krp hjurz q#DSLv#jrrg#iru#xqltxh# vlwxdwlrqv#dv#jhqhulf#FRWV#gr#qrw#p hhw#uhtxluhp hqw/#dqg#vdfn#orrnlqj#ih#dqg# vloyhuljkw#ylhz huv#duh#juhdw#z khq#ghsor|hg#lq#vlwxdwlrqv#z khuh#sox j0lqv#duh#lvvxhv1#Dovr/# wklfn#fdhqw#duh#ehwwhu#iru#odujh#ihdwxuh#vhwv#dv#wkh|#duh#idvw#dqg#p dnh#ehwwhu#xvh#ri# p hp ru|1

Flex API example:
http://wim.usgs.gov/ColoradoRiver/ColoradoRiverMapper.html
WIM team: Jon Baier, Gary Latzke (mgr)

Open Layers Code
http://privusgs6.er.usgs.gov/GeoExtExample/
Zip in https://my.usgs.gov/jira/browse/NGTOCCSDTNM-1059

OpenIn Module
In Save As widget, have option to view in different viewers
Wee have Openlayers code that reads a stub JSON spec, need JSON Reader rewritten to handle ESRI JSON reader spec – estimated at 2 days and pass URL into URI
Google Maps – new ESRI JSON reader and pass in JSON to URL
These JSON Readers could be zip downloads as quickstarts for other USGS or whoever viewer devs to start with help spread concept of same JSON Spec
Coordinate with the USGS WIM team for a Flex API and open in