Context

This guide provides a field overview of the Netcool ITNM GIS. It serves as an addendum to the general GIS information available via IBM knowledge Center as per https://www.ibm.com/support/knowledgecenter/en/SSSHRK_4.2.0/visualize/task/viz_configuringgeographicalviews2.html

All information is current as of ITNM 4.2 FP10, July 2020.

Introduction

Once installed, your Netcool ITNM UI components shall include the artifacts to run GIS enriched maps.

The browser is served these maps and the base OpenLayers libraries from the DASH server $JAZZSM_HOME/profile/installedApps/JazzSMNode01Cell/isc.ear/ncp_gis.war, for example /opt/IBM/netcool/IBM/JazzSM/profile/installedApps/JazzSMNode01Cell/isc.ear/ncp_gis.war

Diagram 1.0, is a high level architectural sequence flow for how GIS enrichment and GIS maps are served to the endUser.

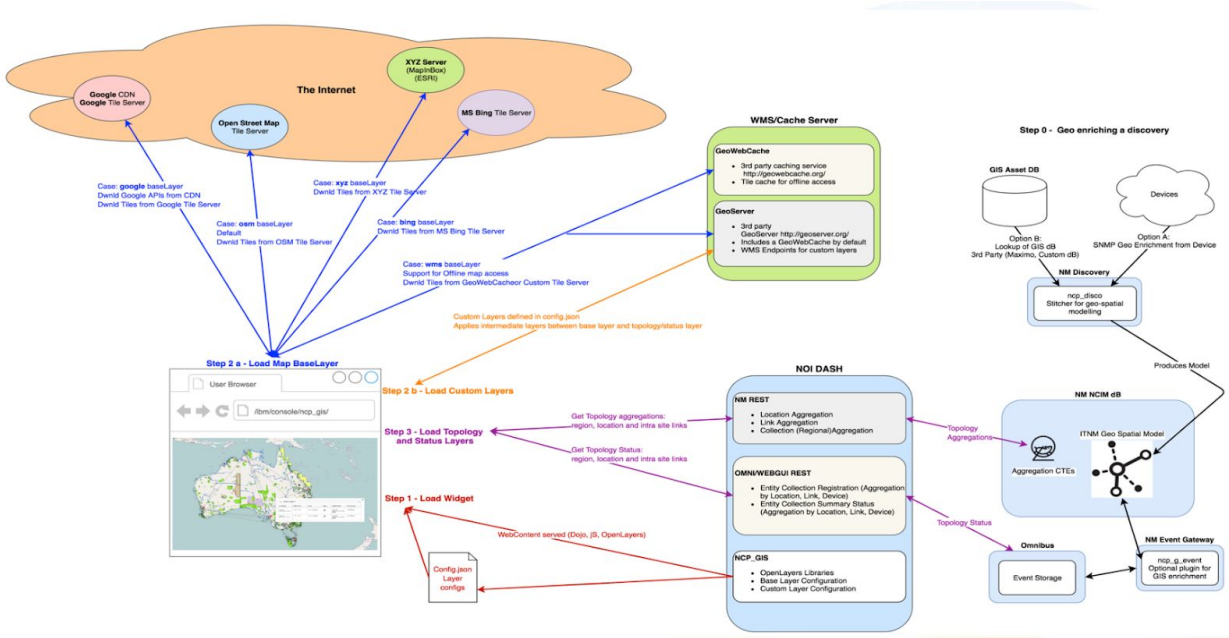

Diagram 1.0 - High Level Architecture

Diagram 1.0, shows the loading and rendering steps involved in the ncp_gis module, which are useful to understand when debugging , analyzing logs or customizing in the future.

The basic architecture is that of RESTful service which supplies geo-enriched topology to the OpenLayers JS widget, which then plots the markers (icons), polylines (links) and collects event status from Omnibus WebGUI for those topological entities.

To avoid confusion, ncp_gis is not related to nor does it re-use any of the logical maps technology in ITNM, such as Network Views, Hop Views, etc. That technology stack is provided from a 3rd party commercial licensed technology stack, known as TomSawyer. The GIS maps (ncp_gis) in ITNM are built upon OpenLayers technology, therefore the operation and customization of Network Views, should not be confused with ncp_gis, they are simply different technology stacks.

The ncp_gis content is served within a DASH widget, but it can also be viewed as an independent view, once a DASH authentication is present within the browser session from the URL, https://server:port/ibm/console/ncp_gis/NetKitMapWidget.jsp. This URL we call the direct URL for naming convention.

Logs and Debugging

Logging

The logging in ncp_gis is in 2 parts, browser/client logs and server-side logs.

Client Logs

Browser logs can be viewed from the Developer Tools Console of the WebBrowser.
These logs cover the loading of OpenLayers libraries, any of the rendering of icons or links and the requests for topology and/or status.
In most browsers, the developer tools are fully loaded with Console lOgs, Source Analysis, Network tracing and Performance toolkits. Within Netcool development, we tend to use both Google Chrome and Firefox for development, debugging and analysis.

The following is an example of the Developer Tools Console logs for ncp_gis from the direct URL

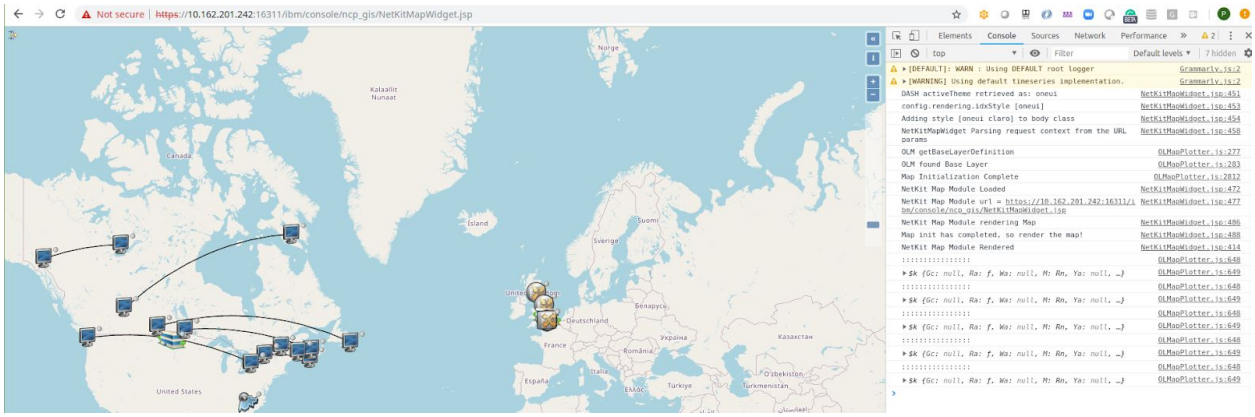

Diagram 2.0 - Web Console of ncp_gis from Google Chrome

These logs are useful when collecting information about missing icons, failures to load the maps etc. The logs themselves can be saved for transmission to IBM Level Support, by right clicking on the console log and selecting “save as”.

Server Logs

As shown in the Diagram 1.0 - High Level Architecture , Step 3, invokes the RESTFul Services to load topology (aggregated or non-aggregated based on the regional aggregation used during discovery). These RESTFul services are contained within the ncp_rest module and their respective logs are location on the DASH JazzSM server at the NCP UI install path /opt/IBM/netcool/gui/precision_gui/profile/logs/tnm

The server side logs simply contain the log of the REST invocations and the underpinning dB queries, known as CTEs (Common Table Expressions) along with some info on their performance.

It is often best to attempt to track performance or result based analysis front the client based logs rather than the server-side logs in ncp_gis, due to the size and volume of data that is being returned. This topic is covered in the next section.

Debugging

RESTFul Invocations

Possibly, the most common of debug scenarios is the no-topology or slow-topology rendering scenario. The best method for analysing this scenario is to look at the RESTFul invocation from the browser via the Developer Tools -> Network toolkit.
Each topology RESTFul service has a set of instrumentation data inlined in the REST JSON response. There are 2 RESTFul Services invoked on ITNM:

Topology Locations - Returns the location data, gis coordinates, iconography, regional structures, etc. It's served from the RESTFul Service Interface:
https://server:port/ibm/console/nm_rest/topology/locations/?aggToRegion=true

Topology Connections - Returns a set of edges for the topology, indication which topology markers should be connected to each other and describes the layers, density of connections etc. It's served from the RESTFul Service Interface :
https://server:port/ibm/console/nm_rest/topology/connections/collectionIds

In the Developer Toolkit -> Network tab, filter the requisition using the term "locations" as shown on the Diagram 2.1.

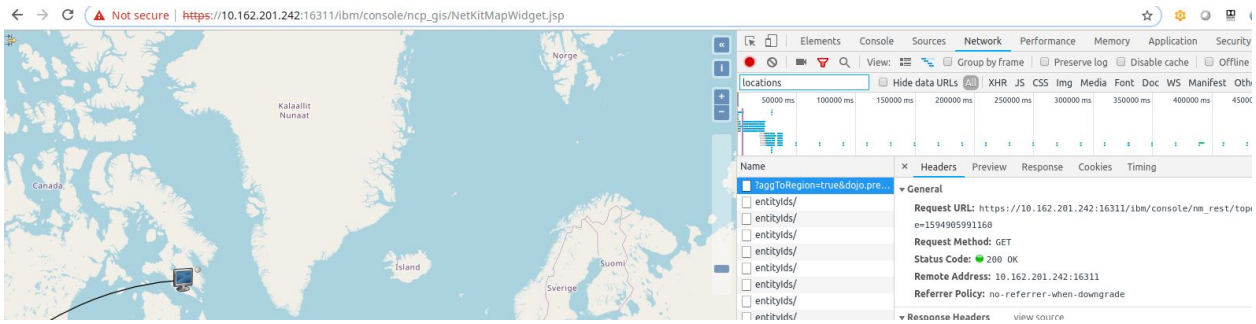

Diagram 2.1 - Filtering Network Traffic by Locations RESTFul Service invocation

This will reduce the profile data to the applicable calls. Select 1 of the calls, the first is the overall call to load all scoped topology. The timing tab will show how long the RESTFul Service call has taken from the browsers perspective: see Diagram 2.2 below.

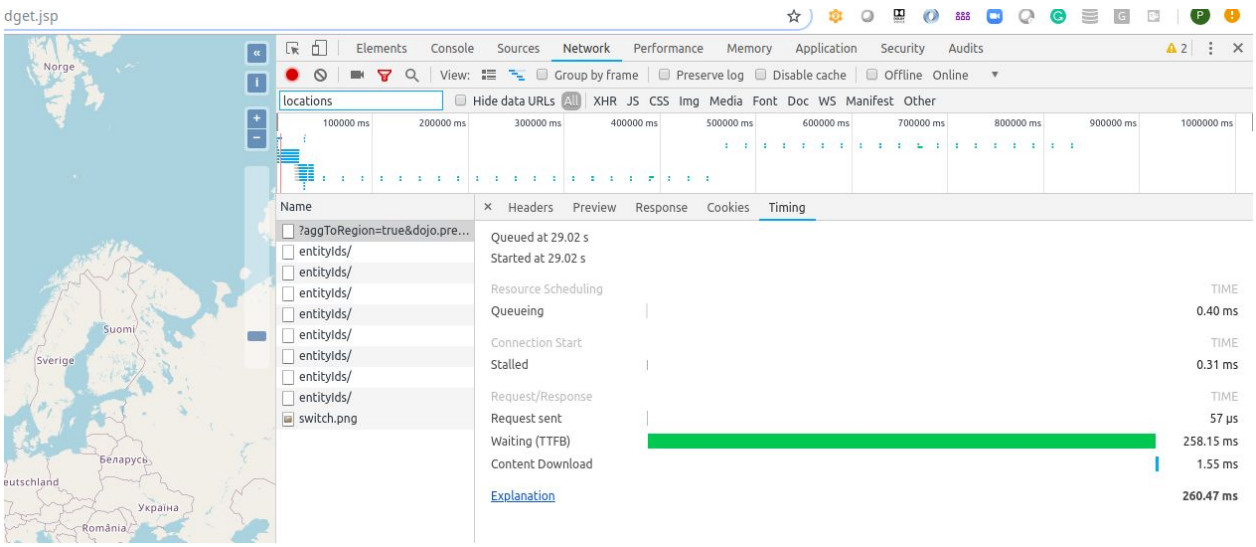

Diagram 2.2 - Network TIming of Locations RESTFul Interface invocation

Further instrumentation data is available in each of our RESTFul Service Interface JSON responses that should be used when analyzing performance. This data can be seen in the Response tab. The data herein is the JSON response payload, which carries the dB query times and payload sizing for the retrieved topology as can be seen in Diagram 2.3.

The two important measures are the queryTime and the totalExecTime. Both measures are in milliseconds. The represent the time it takes to query the geo enriched topology from the ITNM NCIM dB (queryTime) and the total time it takes to query the dB and prepare the JSON payload and respond to the RESTFul client (browser). This data is useful in analyzing performance of ncp_gis and should be provided to IBM Level Support when answering tickets.


```
            "lastupdated": 1585916210000
        }
    },
    "totalExecTime": 75,
    "queryTime": 74,
    "countByDomain": [{
        "count": 4,
        "value": 1,
```


Diagram 2.3 - Query Instrumentation in the RESTFul Service Interface JSON response

In some cases no topology is returned from the RESTFul interface. This is either:

1. No geo enriched data is present in NCIM and the geo enrichment stitching process has either not been run or has failed. To debug this the ITNM admin must analyze the stitching process and determine where the stitching has failed.
2. The RESTFul Service has failed due to a failure in IBM JazzSm or when accessing the dB, in this case the Server side logs are useful as they contain trace level information of the CTE's that are run. It's possible to take these queries and manually run them against the dB with IBM Level Support assistance to diagnose the root cause of the problem.

Aside from looking at the "locations" filtered queries, the other query of importance is the "connections" filtered query which is used to determine the links/edges between topologies. The same analytical process for "locations" applies to "connections".

Missing Icons

Sometimes a map may have links, but those links may appear to be not connected to anything, this is known as the missing class icon in the topology.

The default endNode icon should be applied when a new classtype is being used in ITNM. The classtype for each topological entity is described in the JSON response of the Locations RESTFul Service. An example of that response is shown in Diagram 2.4 - ClassType of Location Markers


```
    "boundseast": -75.70000000,
    "classtype": "EndNode",
    "displaylabel": "11.11.11.14",
    "locationdescription": "166 Capital Blvd Ottawa",
    "boundswest": -75.70000000,
    "webtopdatasource": "NCO_AGG_P",
    "level": 0,
    "entityname": "166 Capital Blvd Ottawa",
    "longitude": -75.70000000,
    "deviceidlist": [4846286],
    "latitude": 45.41666700,
    "entitytype": 111
}, {
    "boundssouth": 51.45984970,
    "boundsnorth": 51.45984970,
    "mainnodeentityid": 4857933,
    "entityid": 4857933,
    "sublocationcount": 0,
    "boundseast": -0.19923449,
    "classtype": "Router",
```

Diagram 2.4 - ClassType of Location Markers

The classType of a location marker is translated to an icon within the code and is served from the list of iconography that is located within the deployed ncp_gis.war at $JAZZSM_HOME/profile/installedApps/JazzSMNode01Cell/isc.ear/ncp_gis.war/resources/common_assets/topology_icons/

The translation process of classtype to icon name is based on toLower() function. Once the icon is mapped it is served to the browser, which can be traced using the Developer Tools -> Network tab.

Customizing Icons

It is possible to remap the current icons or add new icons to the ncp_gis iconography. There are 2 simple rules to adhere to when remapping or adding a new icon for a classtype.

1. The icon must be a PNG format, in 32 * 32 pixel size. For example resources/common_assets/topology_icons/dbfarm.png: PNG image data, 31 x 32, 8-bit/color RGBA, non-interlaced
2. The name of the icon must be lowercase and match the classtype of the location topology returned from NCIM. The process of defining a classtype for a new topological entity in NCIM is described in the IBM Knowledge Center documentation.

Once an icon has been updated or added the endUser must logout, clear cookies and stored content and re login to DASh and ncp_gis to see the new icon.

Customizing Right Click Menus

The right click menu system in ncp_gis is based on the same menu system for the logical maps served using the TomSawyer technology. The menu systems are served using another RESTFul interface, known as Tools. The invocation of the right click menus can also be viewed using the Developer Tools -> Network tab, using the filter “tools” as per Diagram 4.0 - Tools Service Invocation

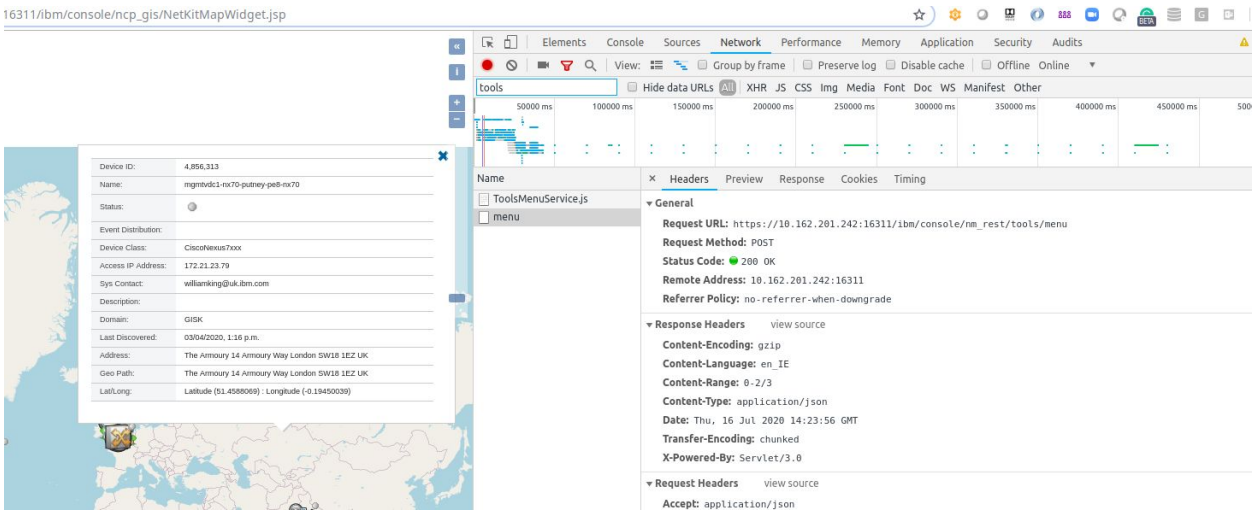

Diagram 4.0 - Tools Service Invocation

Tools are defined on the server side. There is a set of guidelines on IBM Knowledge Center at https://www.ibm.com/support/knowledgecenter/en/SSSHRK_4.2.0/resolve/concept/ops_aboutcontextmenus.html and the syntax is described at https://www.ibm.com/support/knowledgecenter/en/SSSHRK_4.2.0/admin/task/adm_xmlelementsfortools.html

An example of ncp_gis menus and tools can be seen in the base install at: /opt/IBM/netcool/gui/precision_gui/profile/etc/tnm/menus and /opt/IBM/netcool/gui/precision_gui/profile/etc/tnm/tools.

Menu's provide the id and declaration of which tools a menu will serve.

- Right click on a gis device on a map or a device in the table view of a location will serve the menu in ncp_gis_device_menu.xml
- Right click on a link on a map or a link entry on an aggregated link on locations will serve the menu in ncp_gis_link_menu.xml

There is no current support for right click on an aggregated marker such as a region or a location, only on the device or links within them.

To add new tools to the right click, follow the IBM Knowledge Center documentation.

Customizing Maps with Overlays

We support the overlay of custom maps. The feature is described on IBM Knowledge Center documentation at: https://www.ibm.com/support/knowledgecenter/en/SSSHRK_4.2.0/resolve/task/ops_geocustom.html

In general, as long as the new layer is supported by OpenLayers 4.X and matches one of the formats, it will be renderable:

- ArcGISRest
- Web Map Service (WMS)
- Web Mapping Tile Service (WMTS)

Overlay for weather could be provided by https://nowcoast.noaa.gov/help/#!section=rest-usage. A search of Google will turn up other overlays that are compatible. In general ArcGISRest overlays seem to perform best and are the most feature rich.

It should be noted that providing and configuring overlays is supported as a means of integration and IBM and it's IP partner's Wipro are not responsible for the 3rd part overlay services functionality and/or performance.

Defining specific Mapping Contexts

For more information on tailoring maps to pre-defined scopes or contexts, see the IBM Knowledge Center documentation at. https://www.ibm.com/support/knowledgecenter/en/SSSHRK_4.2.0/resolve/reference/ops_geourlparameters.html

This feature is useful if a customer wishes to define very specific maps for specific domains/views, etc. It is then possible to load that ncp_gis URL into a WebWidget in DASH to serve just that specific scoped set of geo enriched topology.