

CorpsMap Version 2.5 Installation Guide
US Army Corps of Engineers
Engineer Research and Development Center
RS/GIS Center
72 Lyme Road
Hanover, NH 03755

September 30, 2010

CorpsMap Version 2.5 Installation Guide

Draft (\rsgisbase.crrel.usace.army.mil\u4rrcspg\Corpsmap_Documentation_Project\CorpsMap_Installation_Guide.htm)

Table of Contents

- Corpsmap Overview and Components
- Preliminary Install Checklist
- Corpsmap Installation Guide
- Post-Install Notes
- Adding a new CorpMap Map Application
- Adding layers to a CorpMap Map Application
- Corpsmap Debugging
- Corpsmap Program Flow
- Appendices
 - Appendix 1- Mapserver Parameters for CM2 Projection
 - Appendix 2 - List of CorpsMap2 (CM2) Configuration tables and column/field descriptions
 - Appendix 3 - Technology Overview
 - Appendix 4 - CorpsMap Frequently Asked Questions (FAQs)
 - Appendix 5 - CorpsMap Developer site and updates
 - Appendix 6 - Other information web-accessible
 - Appendix 7 - Apex, Web browsers
 - Appendix 8 - Future Releases, Developments and Expected Improvements Under Consideration
 - Appendix 9 - Sample Map Files

1. CorpsMap Overview and Components

CorpsMap is a web-based Geographic Information System designed to deliver geographic information to members of the Corps of Engineers.

CorpsMap contains GIS data from the Corps, other Federal agencies and other private and public organizations. Corpsmap also links to Corpwide-databases to geographic information in order bring data and information to members of the Corps of Engineers.

CorpsMap technology is used in a variety of web-based mapping applications. Some USACE divisions and districts have local installations of web mapping applications using CorpMap technology. While this guide is written for the CorpsMap UOC viewer, some features will be similar to local CorpsMap installations. However, some local versions may be customized and may look different and operate differently than the UOC viewer.

The CorpsMap USACE Operations Center (UOC) Viewer is located at: <https://corpsmap.usace.army.mil>

CorpsMap Components

Corpsmap is made up of various components:

- A. Oracle Database
- B. Apache Webserver (with the “mod_plsql” module installed, which allows one to create web applications using Oracle). See http://www.orafaq.com/wiki/Mod_plsql_FAQ for details)
- C. Mapserver, which is a free, open-source web-rendering engine for spatial data (see www.mapserver.org)
- D. Various Javascript libraries, which have been packaged together on the RS/GIS servers at <https://maps.crrel.usace.army.mil/cm2/releases/install/>, including:
 - a. OpenLayers (www.openlayers.org)
 - b. Google Maps API (<http://code.google.com/apis/maps/index.html>)
 - c. GeoExt and ExtJS
 - i. GeoExt imbeds a map with interactive navigation controls (www.geoext.org/tutorials/quickstart.html)
 - ii. ExtJS is a cross-browser javascript library for building rich internet applications (www.sencha.com/products/js)
- E. The above Javascript tools have been downloaded and condensed in to a zip file located on the RS/GIS Center’s servers here: <https://maps.crrel.usace.army.mil/cm2/releases/install/>
- F. Note: This document uses the SQL Developer application to access and modify Oracle database tables. This can be done with other Oracle front-end software packages such as Quest Toad or SQL Plus. SQL Developer is used here because it is a free tool and is fully developed and supported by Oracle. SQL Developer can be downloaded for free here: <http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html> An overview of SQL Developer is located here: <http://www.oracle.com/technetwork/developer-tools/sql-developer/overview/index.html>

2. CorpsMap Installation - Preliminary Installs and Procedures Summary

- A. **Setting up the server.** Install and configure one of the following operating systems:

- Windows 2003: typically ACE-IT will set up Windows Server 2003 or other version
- Linux

B. Setting up Oracle Database:

- Oracle: typically ACE-IT will install Oracle Version 10 or 11
- Version 11.2 is the recommended for CorpsMap, with version 10.2 being the minimum recommended version.
- Optional step: Oracle Spatial
- For Oracle 11g on Windows install procedures, see www.oracle.com/technology/obe/11gr1/install/dbinst/windbinst2.htm

C. Setting up the Apache Webserver (also known as the Oracle HTTP server or OHS) and the “mod_plsql” module.

OHS is a web server built by Oracle, based on the Apache HTTP Server. It includes the MODPLSQL module:

- Setting up Oracle HTTP Server: can download from Oracle
 - http://download.oracle.com/docs/cd/B19306_01/server.102/b14190/toc.htm
- Mod_plsql (Oracle Fusion Middleware) guide:
 - Version 10.2 http://download.oracle.com/docs/cd/B19306_01/server.102/b14337/toc.htm
 - Version 11.1.1 (11g Release 1):
http://download.oracle.com/docs/cd/E14571_01/portal.1111/e12041.pdf
- Oracle Frequently Asked Questions (FAQ's) for Oracle HTTP Server document covers:
 - Starting and Stopping the Oracle HTTP / Apache Server
 - Publishing static HTML pages on the HTTP Server
 - Executing “cgi-bin” programs from the HTTP Server
 - Password-protecting directories
 - Configuring the Oracle HTTP server using the “httpd.conf” file in \$ORACLE_HOME/Apache/Apache/conf/
 - The FAQ document is located here: http://www.orafaq.com/wiki/HTTP_Server_FAQ
 - The following virtual servers are predefined when one install the Oracle HTTP Server:
 - HTTP Server Home Page will be of the form: <http://machine-name:7777/> -- for the Apache, mod_plsql, mod_ose, mod_JServ, and JSP documentation.

- Oracle Enterprise Manager (OEM) Web_stage - <http://machine-name:3339/> for Oracle Enterprise Manager Console and applications.
- Apache Version ____
- Mod_plsql version ____
- Documentation also available from Apache Project:
 - <http://httpd.apache.org/docs/>

D. Setting up MapServer:

- Install Mapserver Version 5.6.5 recommended, minimum version 5.6.3
- Mapserver version available online: <http://mapserver.org/>

E. Setting up ArcSDE (optional)

- Setup documentation available from ESRI at: <http://webhelp.esri.com/arcgisdesktop/9.3/>
- Tight coupling between the ArcSDE version and any ESRI clients (such as ArcMap or ArcCatalog) that attempt to access ArcSDE.
 - The *client* version must be equal to or greater than ArcSDE version
 - Therefore, upgrade *clients first* then SDE.
 - Example: ArcMap 9.3 can access ArcSDE 9.2 data, but ArcMap 9.2 will not be able to access ArcSDE 9.3 data
 - More on Installing and Configuring ArcSDE for Oracle is available here: http://webhelp.esri.com/arcgisdesktop/9.3/index.cfm?TopicName=Administering_ArcSDE_for_Oracle
A more general overview is available here: http://webhelp.esri.com/arcgisdesktop/9.3/index.cfm?TopicName=An_overview_of_ArcSDE_geodata_base_administration

3. CorpsMap Install Procedure:

Once your server, Oracle HTTP Server / Apache Web Server (OHS) and Mapserver are set up, CorpsMap can be installed.

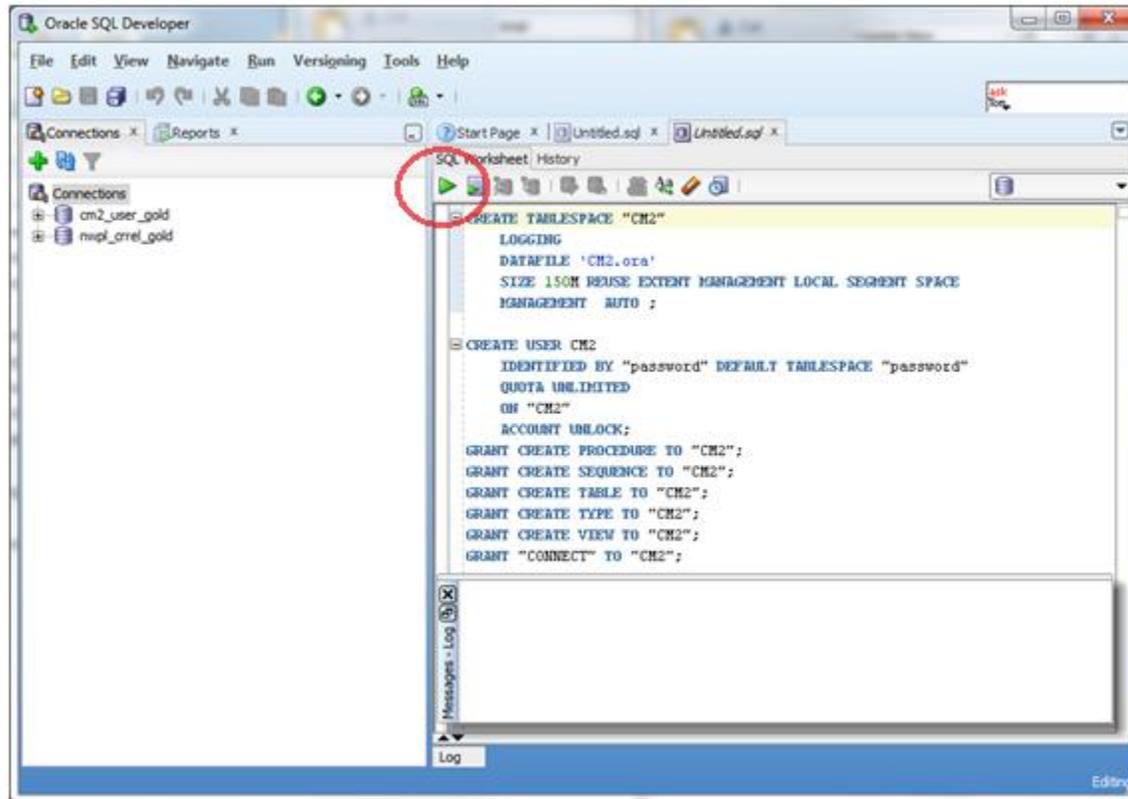
If you are installing CorpsMap for the first time, follow these steps. If you already have CorpsMap installed and are just adding a new CM application, please skip ahead to Chapter 5, “**Adding a new CM Application**”.

- A. Insure that the Oracle database (minimum Version 10.2, maximum 11.1.1) is installed and the Apache Webserver (Oracle HTTP Server) with the “mod_plsql” module is installed on your server.

- B. Navigate to <https://maps.crrel.usace.army.mil/cm2/releases/install/> and download the files **cm_js.zip** and **cm_images.zip**
1. Unzip the files **cm_js.zip** and **cm_images.zip** file to a path on your webserver, preferably a “cm2” folder under your “htdocs” webserver folder (for example, here at crrel our cm2 folder is under the webserver at \\maps.crrel.usace.army.mil\htdocs\cm2)
 - a) The path on your webserver will be referenced in the CM2_MAPCONFIG table.
 - b) Cm_js.zip contains javascript, java, html, python code as well as txt, xml, gif images and other supporting files necessary to make CM2 run
 - c) Cm_images.zip contains png, jpg, gif images
 - d) Spatial_tables.zip – Contains shapefiles for U.S. States and Zip Codes
- C. Using Notepad++ or other text editor, open the “CM2_create_tablespace_and_schema” SQL script, and edit with your information. Notepad ++ is available for free at <http://sourceforge.net/projects/notepad-plus/>

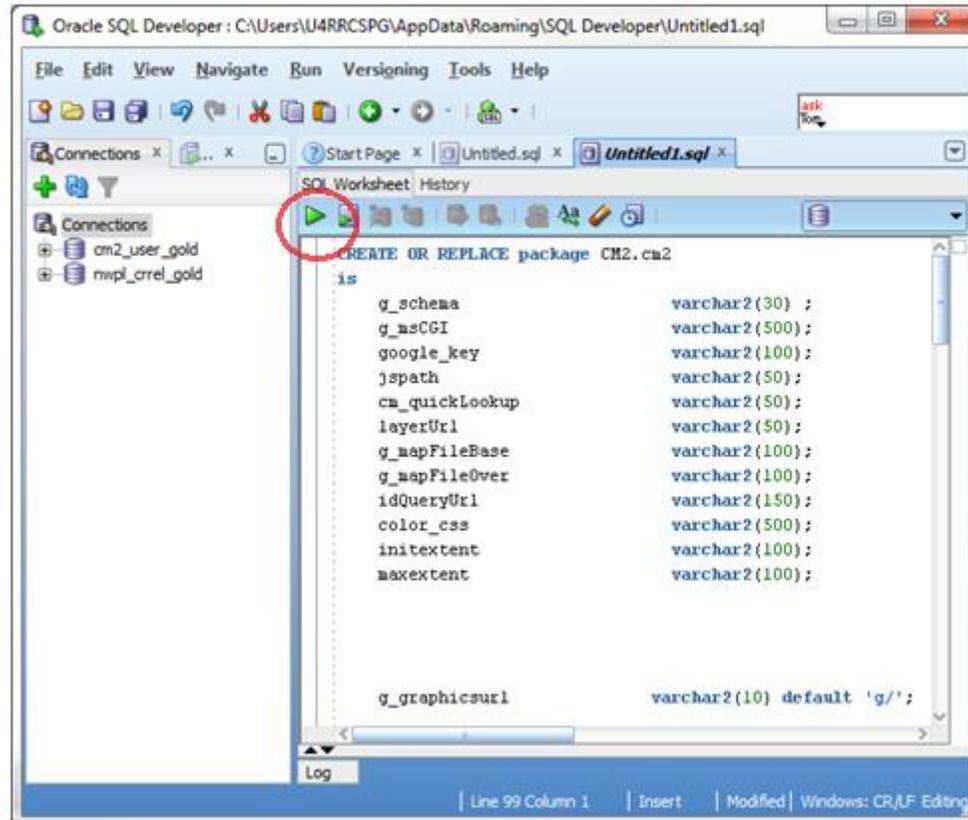
```
CREATE TABLESPACE "CM2"  
  LOGGING  
  DATAFILE 'CM2.ora'  
  SIZE 150M REUSE EXTENT MANAGEMENT LOCAL SEGMENT SPACE  
  MANAGEMENT AUTO ;  
  
CREATE USER CM2  
  IDENTIFIED BY "password" DEFAULT TABLESPACE "password"  
  QUOTA UNLIMITED  
  ON "CM2"  
  ACCOUNT UNLOCK;  
GRANT CREATE PROCEDURE TO "CM2";  
GRANT CREATE SEQUENCE TO "CM2";  
GRANT CREATE TABLE TO "CM2";  
GRANT CREATE TYPE TO "CM2";  
GRANT CREATE VIEW TO "CM2";  
GRANT "CONNECT" TO "CM2";
```

- D. Next, execute this script in Oracle, using Oracle SQL Developer, Quest Toad, Oracle SQL+ or any other Oracle Database SQL client to run this SQL script in Oracle. This will create the CM2 tablespace and schema. Oracle SQL Developer can be downloaded for free at: <http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html>

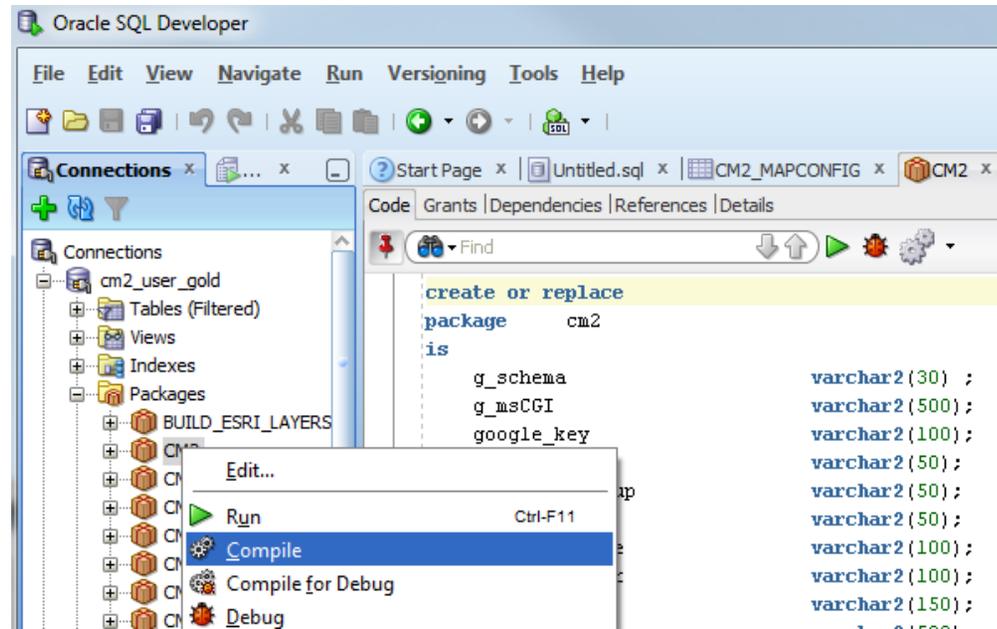


- E. Download the “CM2.DMP” export file from RSGIS server. This is export file contains everything needed to rebuild the database tables and structure. Importing it will set up the same structure in your Oracle database. Use Oracle’s “IMP” command (from the Microsoft DOS prompt or Linux prompt) to import the “CM2.DMP” export file into Oracle.
1. Example: *imp system/password FULL=y FILE=CM2_createtablespace_and_schema.dmp*
 2. *imp SYSTEM/password FULL=y FILE=dba.dmp*
- F. Navigate to <https://maps.crrel.usace.army.mil/cm2/releases/install/> and download the files **CM2.pks** and **CM2.pkb**. These two files will build the “CM2” package in your Oracle database. This package is basically a series of procedures and Oracle code that run the Oracle components of CorpsMap.

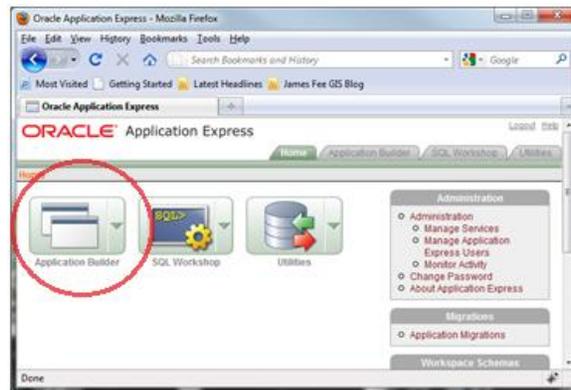
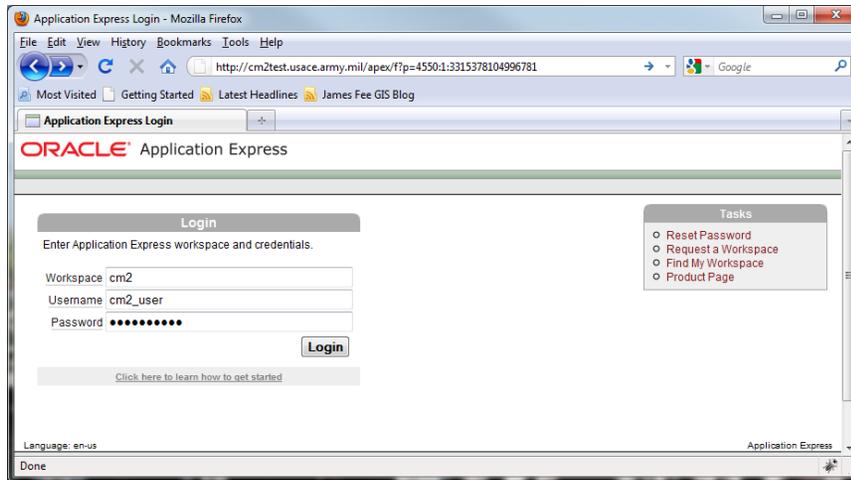
1. Using SQL Developer or Toad, copy in the “CM2.pks” text, and run the statements. This is the “package specification” for the CM2 package in Oracle. In SQL Developer, you can open a new SQL file by choosing File > New > SQL File.

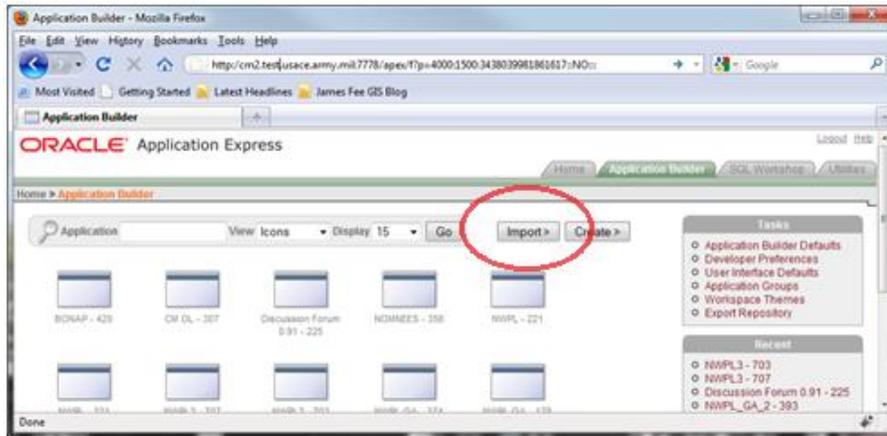


2. Next, copy in the “CM2.pkb” text and run the statements, similar to above. This is the “package body”.
3. Now, compile the package. In SQL Developer, this can be accomplished by clicking the database connection, selecting “Packages”, finding the CM2 package and then choosing “Compile”.

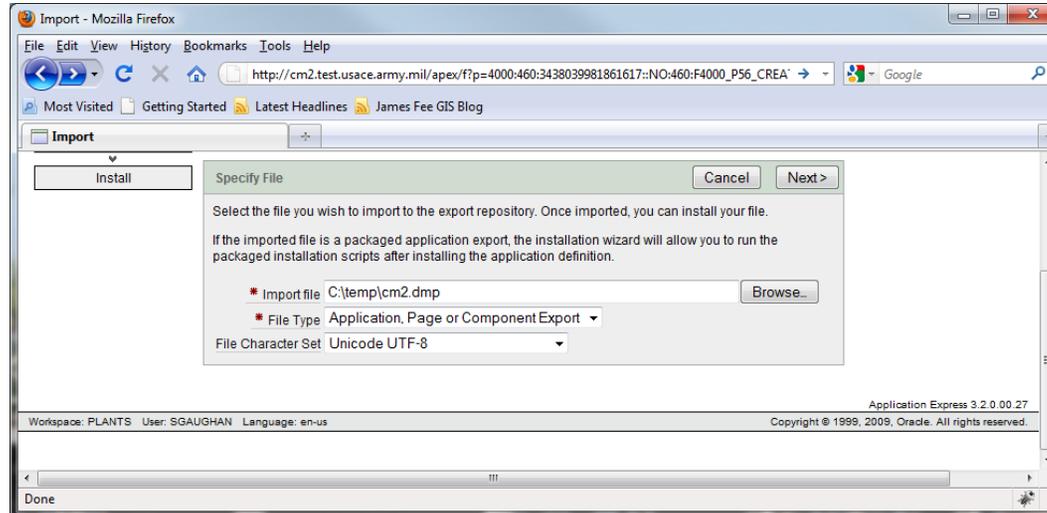


- G. Close your Oracle SQL Developer or Toad application.
- H. Navigate to your Oracle Application Express home page and log in as an APEX developer or administrator. This is typically at [https://<server name>/<Oracle Data Access Descriptor \(DAD\) >](https://<server name>/<Oracle Data Access Descriptor (DAD) >)
- I. Create an APEX Workspace (of any name) that uses the existing schema that was created in Step 2 by the "CM2_create_tablespace_and_schema" SQL script.
- J. Navigate to the APEX Workspace, and click on "Application Builder", and then on the "Import" button.



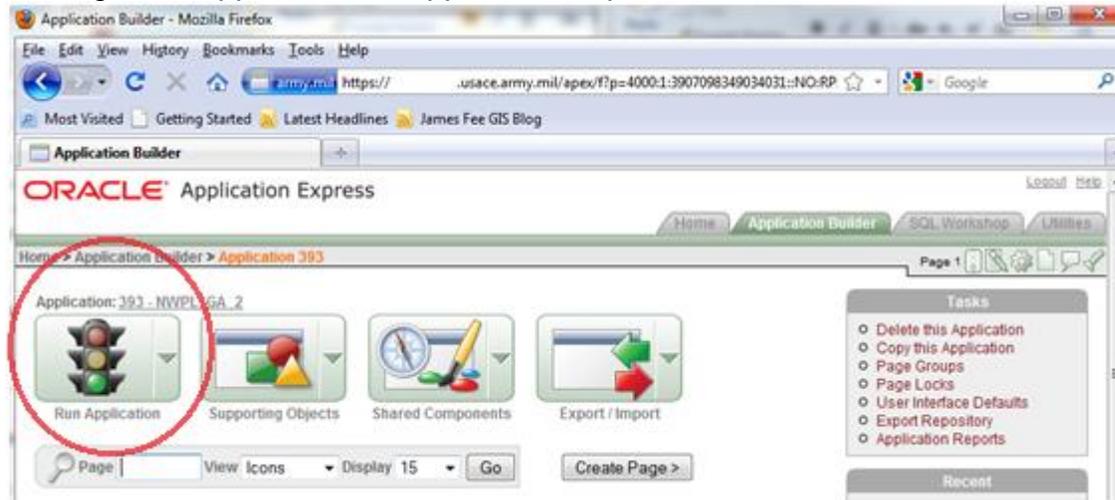


1. Browse to the “cm2_apex_search_application” SQL file and load the file into that apex workspace.
2. Set the File type to “Application, Page or Component”, and File Character Set to “Unicode UTF-8”



3. Write down the Oracle Application Express application number you choose.

- a) Application Number: _____
 - b) This number will get added to the SEARCH_URL, ID_QUERY, and EXCEL_QUERY columns, in the form of a query of the form: *f?p=379:10:0:::P10_LAYER,P10_ID:"+qlid+","+l_id* where “379” is the application number
4. Test to see if the CorpsMap Apex application is up and running by going to the chosen application or by clicking “Run Application” in Application Express:



5. After clicking “Run Application”, your new CorpsMap application should be up and running using the default map. The URL that displays after clicking “Run Application” is the URL of the application that can be sent out to users.
 6. The URL will be of the form:
https://<server_name>.usace.army.mil/<oracle_dad>/f?p=<apex_application_number> That will be followed by an Apex “session number”, for example :111:283418836914378.
 7. Note: If the application is to be proxied outside the firewall, the URL will be different.
- K. To begin adding layers to your new CorpsMap Application, see section 6 “**Adding data to be viewed in the map application**”
- L. A few notes on the CM2 database tables, which can be viewed and edited using Oracle SQL Developer or Toad:

1. In the CM2_MAPCONFIG Oracle table, any value for GLOBAL is used when the package is called with no map= parameter.
2. Values from the GLOBAL row are used in other mapnames (?MAP=XXXX) unless the value is overridden by filling out the column.
3. For example, the file path referenced in the COLOR_CSS_PATH column will be overridden for map=gtest but not for map=ctest

The screenshot shows the Oracle SQL Developer interface with the CM2_MAPCONFIG table selected. The table data is displayed in a grid view with the following columns: MAPNAME, BASEMAP_FILE, OVERLAY_FILE, and JS_PA.

MAPNAME	BASEMAP_FILE	OVERLAY_FILE	JS_PA
7 GTEST	/data/cm2_mapfiles/cm2_basemap.map	/data/cm2_mapfiles/cm2.map	(null)
8 BTEST	/data/cm2_mapfiles/cm2_basemap.map	/htdocs/amy/cm2/cm2_agg.map	(null)
9 NLD	/data/cm2_mapfiles/cm2_basemap.map	/data/cm2_mapfiles/cm2.map	(null)
10 AEDIS	/data/cm2_mapfiles/cm2_basemap.map	/htdocs/aedis/aedis.map	(null)

M. A few notes on the Reference URL's for CM2:

1. [http\(s\)://web_server_name:\(port\)/dad_path/schema_name.cm2.map?=XXXXX](http(s)://web_server_name:(port)/dad_path/schema_name.cm2.map?=XXXXX) (for specific mapname) or
2. [http\(s\)://web_server_name:\(port\)/dad_path/cm2.cm2.map](http(s)://web_server_name:(port)/dad_path/cm2.cm2.map) for default map.
3. dad_path = database access descriptor, which is a set of instructions for the Oracle HTTP server and the mod_plsql extension module. The configuration file "dads.confand" is typically located on your machine under the ORACLE_HOME directory, ORACLE_HOME\Apache\modplsql\conf in windows, or ORACLE_HOME/Apache/modplsql/conf in Linux.

N. A few reference notes for Full URL Parameters in CM2:

1. p_layers: Comma delimited list of checked layers - Overrides cookie values
2. p_maptypes: (view/edit) Start CM2 in View mode (no edit tools) or in Edit mode (full Editing Toolset)

3. p_unique_id: PK value of geometry to retrieve/save when editing/creating geometry
4. p_editlayer: layername passed to geometry store package to allow editing of multiple layers
5. p_geom_api: url to geometry storage package
6. p_geom_api_get: url to geometry retrieval package
7. p_role_id: passed to p_geom_api for role based editing
8. p_basemap: (geh/gem/ges/gec/cm) meaning hybrid/streets/satellite/terrain/corpsmap. overrides cookie based basemap
9. map: selects values row in cm2_mapconfig table.

4. Post-Install Instructions:

The map URL will be of the form:

[https://<your web server name>.usace.army.mil/<oracle data access descriptor \(dad\)> / cm2.cm25.map?map=<map name in CM_MapConfig table>](https://<your web server name>.usace.army.mil/<oracle data access descriptor (dad)> / cm2.cm25.map?map=<map name in CM_MapConfig table>)

The default map will be shown if no map name is specified.

5. Adding a new CorpsMap Application:

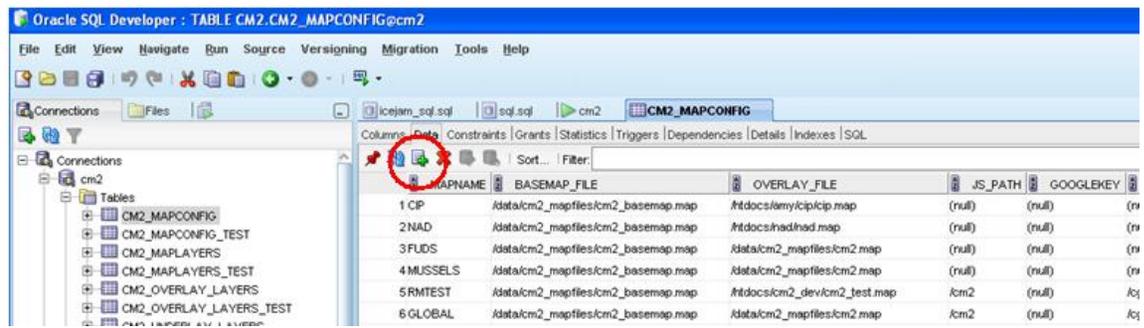
- A. The Oracle table CM2_MAPCONFIG contains rows of data where each row represents a separate map viewer.
 1. Example: CRREL hosts numerous CorpsMap viewers, such as the Corps-wide UOC viewer, the National Levee Database viewer, etc
- B. Setting up the "GLOBAL" map in the CM2_MAPCONFIG table
 1. The CM2_MAPCONFIG table contains a row with the map name "GLOBAL". This row is the "default" map, and any future maps can be based on this map. Setting up this row first will expedite future map viewer creation.
- C. To set up the "GLOBAL" row, fill out the following rows:
 - MAPNAME = GLOBAL (this row is the name identifying a map viewer)
 - BASEMAP_FILE = path to your standard MapServer Map file. See sample mapfile in Appendix ___??
Example: /data/cm2_mapfiles/cm2_basemap.map
 - OVERLAY_FILE = path to your overlay MapServer Map file. Example: /data/cm2_mapfiles/cm2_overlay.map.
Layers are controlled via folder "tree" in the viewer

- JS_PATH=subdirectory to javascript location. Typically /cm2. This should be the folder containing the MapFish 1.0 directory
- GOOGLEKEY – can leave blank. The Google maps key for your specific server URL. Get from <http://code.google.com/apis/maps>
- MSCGI - path to MapServer CGI binary
- LAYER_URL - URL to procedure generating the layers XML. Usually CM2.CM2.LAYERS?p_mapname=XXXXX
- CM_SCHEMA - the schema that the CM2 package is located in
- ID_QUERY - URL to invoke for an ID based query. Usually:
f?p=APEX_APP_NUMBER:10:0::::P10_LAYER,P10_ID:"+qlid+", "+l_id
- QLOOKUP - deprecated – unused
- COLOR_CSS_PATH - path to an alternative CSS file. Used to "skin" CM2 for different looks. See extjs.com for more skins
- INIT_EXTENT - Comma delimited extent (LL). Used for initial map view if someone doesn't override with local cookie or URL parameter
- MAX_EXTENT - Command delimited extent (LL). The extent to go to when clicking the Max Extent button in CM2 (Left most icon in toolbar).
- REDRAW_LAYER_CHANGE - (Y/empty) - if Y, a change to the layer tree (layer checked) won't draw until Redraw button on Layer Toolbar clicked
- SEARCH_URL - URL to apex search page. Usually: f?p=APEX_APP_NUMBER:5
- MAPTITLE - (html) Sets the Title in the Upper Left-hand corner to this value and on initial load. Can contain graphics/html
- MAPTITLE_TT - Sets the Browser window title bar
- EXCEL_QUERY - URL to apex Excel export page. Export is based on query for that layer.
 - Usually: f?p=APEX_APP_ID:11:0::::P11_LAYER,P11_ID:"+qlid+", "+rowSelection
 - EDIT_COMPLETE_URL - Url to invoke on completion of a geometry save.
 - SSL - (Y/empty) Y for running on https, empty for http. Sets Cookie types

- INFOLAYER_AUTOSSET - (Y/empty) Y to have the InfoLayer set whenever a layer is turned on (that is a queryable layer) in the layer tree.
- LOCAL_JS – (Y/y or null) – If Y then no external APIs are used (eg Google Maps API). This disables all Google Maps API feature such as Street View, Overview Map and Traffic. The Google BaseMaps are replaced by OpenStreetMap.
- CM2_NORTH_REGION – clob – a javascript north region block,
- CM2_NORTH_DIV – clob – an html div block associated with the content

D. To add a new map viewer application:

1. Add a new row to CM2_MAPCONFIG table using SQL Developer, TOAD, or other Oracle Database SQL Client interface. To add a row in SQL Developer, first navigate to the CM2_MAPCONFIG table in the left-hand pane, then in the right-hand pane, click the “Data” tab , and finally click the green “+” button  on the top bar to add a row:



This screen shot shows a table in Oracle SQL Developer with six rows, which means there are six different map viewers in the CM2_MAPCONFIG table. The red circle is around the “Add Row” button.

2. Add new row, with the following fields:
 - a) NAME = <your map name> Convention is all capital letters, and name **must be unique**.
 - b) BASEMAP_FILE =

- c) OVERLAY_FILE= the path here should point to your MapServer “map” file. At CRREL, the map files are typically stored in \\maps.crrel.usace.army.mil\data\cm2_mapfiles , of the format \\<webserver name>\data\cm2_mapfiles. This is preferable to placing map files in “htdocs” folders, which may be exposed outside the firewall, and which may include passwords and connection strings to databases.
- d) Provided that the “GLOBAL” row has been filled out properly (see steps above), all other columns can be left blank. By doing so, any null value from the table in the calls will be re-directed from this new row to the value in the “GLOBAL” row.

E. The map should now be able to be seen through your web server. The format of the URL for the map will be:

- 1. http:// <your web server> / <oracle data access descriptor (dad)> / cm2.cm25.map?map= <value of “mapname” field in CM2_CONFIG table>
- 2. Example: <https://rsgis.crrel.usace.army.mil/apex/cm2.cm25.map?map=UOC> where rsgis.crrel.usace.army.mil is webserver or proxy server, “apex” is the oracle data access descriptor or “dad”, and the name of the map in the CM2_CONFIG table is “UOC”.

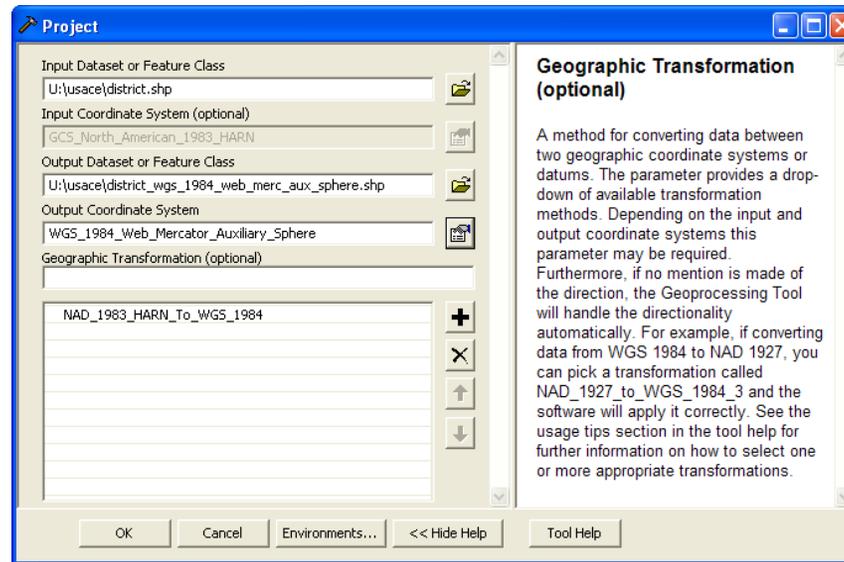
6. Adding data to be viewed in the map application

A. Here is a typical flow for adding data to a particular map application:

- 1. Project data in to the correct projection
- 2. Load Data (SDE, SDO, etc)
- 3. Create an entry in the MapServer Mapfile to link to the data and to symbolize and stylize the data
- 4. Enable the map application to place the data in the folder tree or table of contents.

B. Getting your data in to the right projection

- 1. Using ArcCatalog > Data Management Tools > Feature > Project
 - i. Output Coordinate System (Browse) > Projected > World > WGS 1984 Web Mercator (Auxiliary Sphere).prj



C. **Loading data** in to Oracle via ArcCatalog

- i. Data should be in spherical Mercator, the projection of Google Maps base layers such as terrain. In ArcGIS, this projection is “WGS 1984 Web Mercator (Auxiliary Sphere).prj”

D. **Editing the mapfile** to display the layer. Mapfiles can be edited using most text editors. A few good ones are Notepad ++ and Ultraedit. There is also a plug-in for Notepad available at:

<http://gianpiero.campanella.googlepages.com/mapfile-highlighting-Notepad-1.3.zip>

E. Any overlay mapfile should have the following information in it:

1. MAXSIZE 8000 – this tells MapServer to go as big as 8000x8000 pixels.
2. The projection of the mapfile should be in Spherical Mercator. This matches the projection used by Google Maps layers and ESRI ArcGISOnline layers that are used as base layers (Bing and Yahoo maps use this projection as well).

PROJECTION

```
"+proj=merc +a=6378137 +b=6378137 +lat_ts=0.0 +lon_0=0.0 +x_0=0.0 +y_0=0 +k=1.0 +units=m
+nadgrids=@null +wktext +no_defs"
```

END

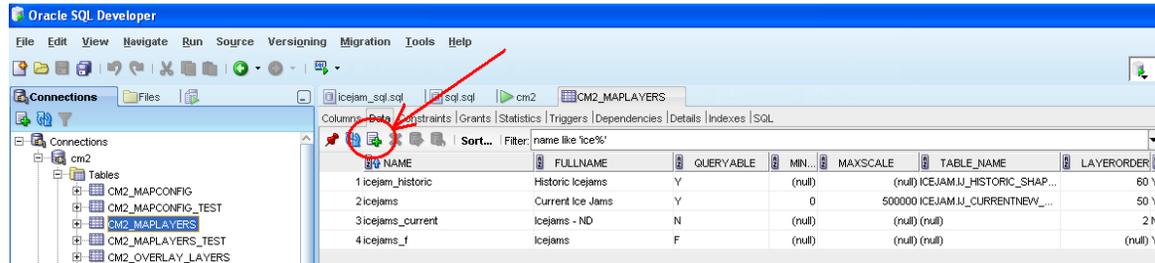
3. Google Maps, Bing Maps and All ArcGIS Online map services use the “Web Mercator auxiliary sphere projection, WKID 102100, 256x256 pixel tiles with refined scales. This simplifies overlays and mash-ups)
4. The scale options in CM2 are static because they are based on map caches from other map services like Google Maps and ESRI Online map services. The scale choices may look odd, but they are basically a global scale beginning at 1:110,936,068 and divided in half for each zoom step. Draw time is faster as a result of using cached zoom scales. Google Terrain tiles only go down to 1:13,542. Google Streets go down farther.
5. Define the Map Extents, in Meters, to match the Spherical Mercator projection map extents that are also used by Google and ESRI Online maps:
[EXTENT -20037508 -20037508 20037508 20037508.34](#)
6. Adding a new layer to the map file. Paste the code in to the map file. Indentation is important:

```
### START layers
LAYER
  NAME redriver_mar182009
  TYPE RASTER
  DATA "/fire/share/for_mikesmith/Flood09_imagery/lstat5_p30r27_18mar09.img"
  STATUS off
  PROJECTION
    "proj=utm"
    "zone=14"
    "ellps=GRS80"
    "datum=NAD83"
    "units=m"
    "no_defs"
  END
  MAXSCALE 4000000
  OFFSITE 0 0 0
  PROCESSING "BANDS=4,2,1"
  PROCESSING "SCALE=AUTO"
  #PROCESSING "DITHER=YES"
END
```

7. NOTE: Name of layer in mapfile (and subsequently in the CM2_MAPLAYERS table) should be unique across *all* map viewers, otherwise issues may arise.
8. Exception to unique names: Layers may be given the same name in a mapserver map file if the intent is to have all of the layers turned on simultaneously by checking the composite layer name in the table of contents.
9. Fill out MINSCALE and MAXSCALE appropriately in the mapserver map file.

F. Adding the map layer name to the CM2_MAPLAYERS table

1. Add a new row to the “CM2_MAPLAYERS” table. In SQL Developer, use the green + sign button to add a new row:



2. Enter the layer name exactly as it was entered in the map file in to the “NAME” field. No spaces allowed. The convention is to use all lower-case letters.
3. FULLNAME is the name shown in the table of content’s folder tree
4. Set the “QUERYABLE” column to “Y” if the layer can be queried, “N” if it cannot. The value “F” is reserved to indicate that the row is a folder, not a layer.
 - i. CM2_MAPLAYERS lists both *layers* and *folders* in order to build the table-of-contents in the map application. The “QUERYABLE” column is set to “F” for folders. For layers, it is set to either “Y” or “N”.
5. MINSCALE and MAXSCALE can be null or can be filled out. Layers will not be visible in the map and will be “grayed out” in the table of contents when the map scale is outside the bounds of these two fields.
6. TABLE_NAME is the name of the table to be queried.
7. The “LAYERORDER” column is *relative*. This means that the lower layer-order number will be drawn first, relative to all other layers in the same folder. The folder that the layers fall under is defined in the CM2_MAPLAYERS table in the “PARENT” folder.
8. The “LAYERORDER” convention is to go by tens: first layer = 10, second layer =20, third layer =30. By doing so, there is “room” to insert another layer if necessary. For instance, if you want the “current_icejams” layer to be displayed above the “historical_icejams” layer, set “current_icejams” layerorder to 10 and “historical_icejams to be 20.
9. The “ACTIVE” column should be set to Y, if the layer is to be displayed in *any* CM2 map application
10. The “PRIVATE” column is no longer used.
11. In the “PARENT” column, indicate the folder that the layer should be placed in. Folders can be “nested” inside one another, but layers will always have a folder as a parent. To place a layer at the “root” level rather than in a folder, set the PARENT field to **root**

- i. NOTE: a child layer or folder with no parent will not display in the layer tree. The PARENT field must be filled out, and it must be either set to **root** or to a valid folder elsewhere in the CM2_MAPLAYERS table
12. The "COLUMNS" field indicates what columns should be displayed when a layer is queried. Separate the columns using a comma. This field can contain any valid SQL for columns, and can also be used to construct hyperlinks for columns returned.
- i. To have the columns have a "nice" column heading, enclose the column heading in double quotes.
 - ii. Constructing [Hyperlinks](#) in the "COLUMNS" list: Note that using a standard SQL concatenation and single quotes, you can construct hyperlinks as a column:
 1. Example 1, using a field name to create a hyperlink in a column. This example uses the field name "loc_name". Here is the entire entry in the "COLUMNS" field: `loc_name
"Location", '<a
href="http://maps.crrel.usace.army.mil:7778/arrk/nap/'||video||'"
target="_new">Video Link' "Video"`
 2. Example 2: using an APEX variable to create a hyperlink in a column. Here is the entire entry in the "COLUMNS" field: `initcap(project_name) "Project Name",
initcap(organization) "District", congressional_appropriation
"Appropriation", authorization "Authorization", primary_function
"Primary Purpose", '<a
href="https://maps.crrel.usace.army.mil/apex/f?p=386:202:::::P202_PRO
JECT_ID:'|| project_id ||'" target="_new">Details' "Details"`
 3. Generic example: `name "Name", initcap(project_manager) "PM", '<a
href="someUrlHere?id='||ID_COLUMN||'"
target="_new">'||description||'' "Hyperlink"`
 4. Note also that the "COLUMN" field can contain multiple columns, one or more of which can be a hyperlink. This example will return the columns Project Name, District, Details, with "Details" as a hyperlink: `initcap(project_name) "Project Name",
initcap(organization) "District", '<a
href="https://maps.crrel.usace.army.mil/apex/f?p=386:202:::::P202_PRO
JECT_ID:'|| project_id ||'" target="_new">Details' "Details"`
13. The "POPUP" column OPUP - This is the information displayed when the initial query is done and used for the Quick Query display. Its has a very specific format. It consists of 4 values, comma delimited. They are Value1, Value2, Longitude, Latitude. Value1 and value2 can be any column or sql value. Value1 will be

displayed with the column heading NAME and Value2 will have the column heading DESC (for Description). Longitude and Latitude should be numeric (-180/+180/-90/+90) values. If derived from spatial column, use must use an alias (sql object access rules). The value is predefined as "d".

i. Examples:

1. Point spatial column (named shape): d.shape.sdo_point.x, d.shape.sdo_point.y
2. Line spatial column (named geom): sdo_geom.sdo_pointonsurface(d.geom,0.1).sdo_point.x, sdo_geom.sdo_pointonsurface(d.geom,0.1).sdo_point.y
3. Polygon spatial column (named shape): sdo_geom.sdo_centroid(d.shape,0.1).sdo_point.x, sdo_geom.sdo_centroid(d.shape,0.1).sdo_point.y
4. Complete entry: initcap(installation), description, d.shape.sdo_point.x, d.shape.sdo_point.y

14.ID_COLUMN: this is the unique identifier for the table (column name)

15.MAPNAME: leave blank, no longer used

16.DESCRPTION: any text that will displayed when the mouse is "hovered" over the layer in the layer tree

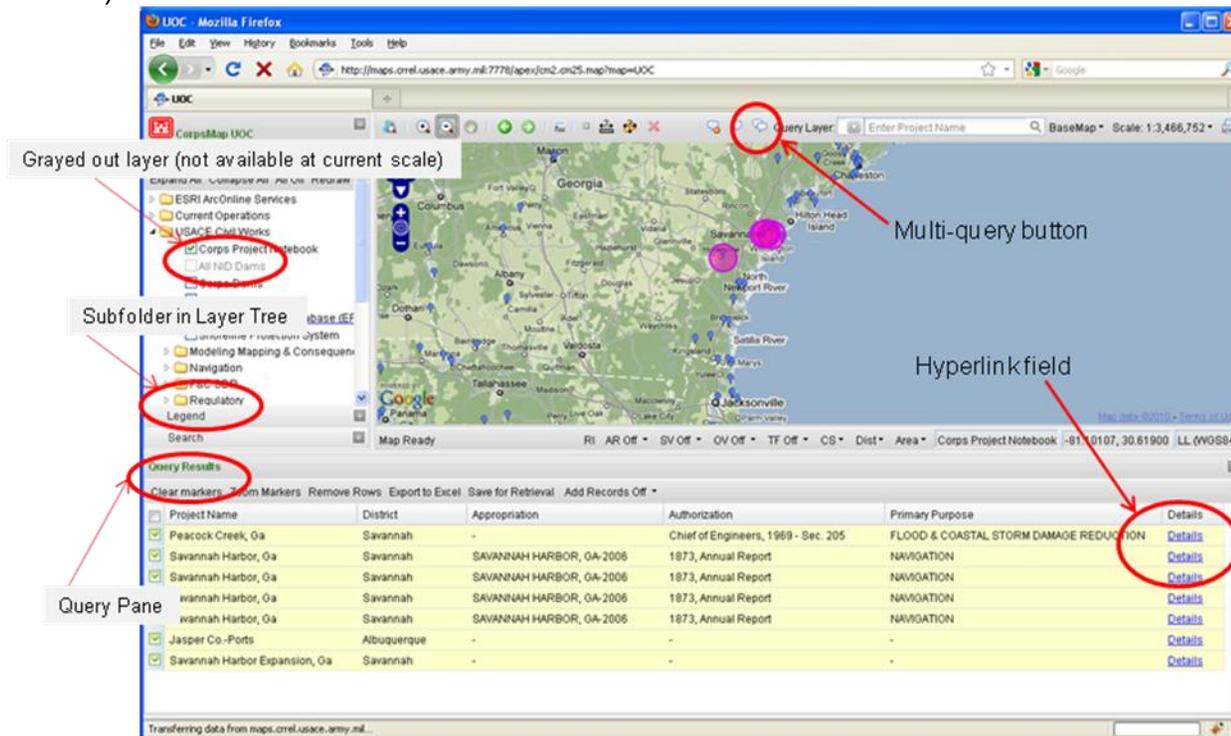
17.INC_MAPNAME AND EXC_MAPNAME:

- i. INC_MAPNAME indicates all maps that should **include** this layer, listed by map name from the "CM2_MAPCONFIG" table.
- ii. EXC_MAPNAME indicates all maps that should **exclude** this layer, listed by map name from the "CM2_MAPCONFIG" table.
- iii. Typically, if a layer is only used in one or a few map application, use the INC_MAPNAME field. Otherwise use the "EXC_MAPNAME". The CM2 application will search **both** columns in order to figure out whether a layer should be displayed or not.

G. Defining a new folder

1. Using SQL Developer, Toad or other Oracle Database SQL client software, create a row in CM2_MAPLAYERS table and give it a unique "NAME".
 - i. The convention with folders is to follow the name with _f to indicate that it is a folder, not a map layer. Example is icejam_f, which is the folder that contains two icejam layers inside it.
2. FULLNAME is the name shown in the table of content's folder tree
3. Set the "QUERYABLE" field to "F" for "Folder".
 - i. CM2_MAPLAYERS lists both *layers* and *folders* in order to build the table-of-contents in the map application. The "QUERYABLE" column is set to "F" for folders. For layers, it is set to either "Y" or "N".
4. Set the "ACTIVE" field value to Y if the folder is going to be used in **any** map application. Use the "INC_MAPNAME" and "EXC_MAPNAME" to specify which map application should display the folder.

5. In the “PARENT” column, indicate the folder that the sub-folder should be placed in. Folders can be “nested” inside one another, but layers will always have a folder as a parent. To place a folder at the “root” level rather than in another folder, set the PARENT field to **root**
 - i. NOTE: A child folder or layer with no parent will not display in the layer tree. The PARENT field must be filled out, and it must be either set to **root** or to a valid folder elsewhere in the CM2_MAPLAYERS table
 6. Use the “INC_MAPNAME” and “EXC_MAPNAME” to specify which map application should display the folder.
 7. All other columns can be left blank for folders.
- H. Layers, Queries, Hyperlinks in the CM2 interface (See also the CorpsMap Users Guide for more information on using the application)



I. Layer Troubleshooting:

1. If a layer is not showing up, check the following:
 - i. Does the layer have a valid entry in the Mapserver "map" file?
 - ii. Does the layer have a valid entry in the CM2_MAPLAYERS table?
 - iii. Do the names match exactly in the map file and the CM2_MAPLAYERS table? Case sensitive?
 - iv. Does the layer have a valid "PARENT" folder? If not, temporarily set the PARENT column for the layer to **root** to see if the layer shows up.
 - v. Do all the child folders have valid parent folders?
 - vi. Does the Mapserver "Map" file have valid MINSCALE and MAXSCALE values?
 - vii. Is the layer projected in to spherical Mercator, so that it matches the underlying layers from Google and ESRI?
 - viii. Is the map application working? Can you see all the other layers that are supposed to be in the application?

J. Projection Information – ArcGIS

1. To project a GIS data layer to match Google Maps layers and ESRI ArcgisOnline layers, use ArcToolbox > Data Management > Projections
2. **Recommended Projection:** "WGS 1984 Web Mercator (Auxiliary Sphere).prj"
3. This uses the standard WGS84. Instead the map projection is a new implementation of Mercator which only has sphere-based math. Thus, if the projected coordsys includes a spheroid (ellipsoid-based) geographic coordsys, it's converted to a sphere.
4. Alternate: "WGS 1984 Web Mercator.prj"
5. This one uses a custom geographic coordinate system based on a sphere. It should have been called something like 'WGS 1984 (Major Auxiliary Sphere) Web Mercator'. Because it's a custom GCS, it needs a separate geographic/datum transformation to convert between it and the standard WGS84.
6. NOTE: Both versions give exactly the same results. And this is the same definition used by Microsoft Virtual Earth and Google Maps (Google Earth uses a different projection).

7. CM2 Debugging/Tips and Other Info:

- A. In order to see the calls that CM2 is making in the background. There are several methods to debugging CM2, starting with the easiest methods:
 1. Use Firebug / Firefox - www.getfirebug.com
 2. Use Chrome and choose View->Developer->Developer tools
 3. Use IE7/8, you need some external programs to see ajax callouts

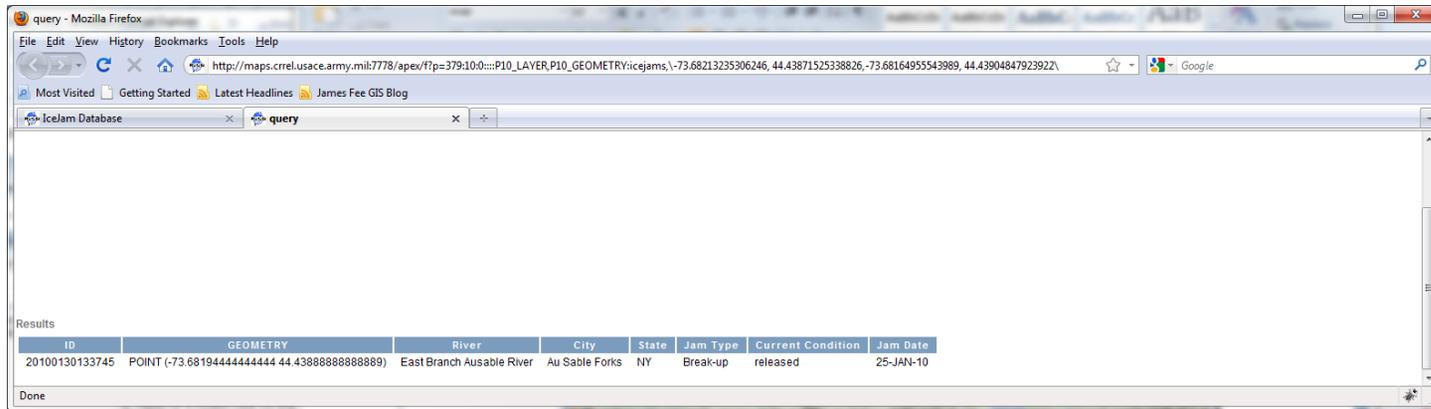
- i. simpler one: <http://projects.nikhilk.net/WebDevHelper>
 - ii. the more comprehensive one is : <http://ajax.dynatrace.com/pages/download/download.aspx>
 - iii. Both of these install and allow http logging to see external callouts that CM2 is doing.
- B. Map drawing errors in CM2 v2.5 now display a dialog box with the error returned from MapServer. Occasionally, after an error, and its correction, CM2 still sees an error and returns the image from MapServer as text. If this happens, just reload the page.
- C. If CM2 is not loading, examine the external file loads (see 1) and make sure that all the Javascript libraries are loading. URLs in red are returning 404/not found errors
- D. CM2 needs to be granted execute to the user that "owns" the DAD that it is executed under
- E. To query a table, the CM2 user needs a grant select on that table
- F. For restoration of previous view/extents/layers etc, CM2 relies on user cookies.
- G. If this is not working properly, make sure your SSL column in the CM2_MAPCONFIG is set properly. It needs to be Y when running under https so the cookies can be marked as secure cookies.
- H. If a query is not returning a response, it is generating an error somewhere.
 - 1. The "Net" tab of Firebug is the best place to start looking. To enable the "Net" tab in Firebug, open up Firefox, navigate to your application, click the bug icon in the lower right corner. Place the cursor just next to the "Net" tab, and choose "Enable" from the pop-up menu.

Using "Firebug" with CorpsMap within the "Firefox" web browser.

The screenshot shows a Mozilla Firefox browser window with the address bar displaying `http://maps.crrl.usace.army.mil:7778/apex/cm2.cm25.map?map=icejam`. The main content area shows a map of the Lake Champlain region with various layers and a Firebug console at the bottom. The console displays a list of network requests, including a highlighted URL for a map data request.

Request	Status	Response Size	Response Time	Source
<code>GET /mapserv5_nldb?map=/data/cm2</code>	200 OK	13.4 KB	100ms	maps.crrl.usace.army.mil:7778
<code>GET shadow.png</code>	200 OK	311 B	106ms	extjs.cacheify.net
<code>GET shadow-lr.png</code>	200 OK	135 B	109ms	extjs.cacheify.net
<code>GET shadow-c.png</code>	200 OK	118 B	104ms	extjs.cacheify.net
<code>GET /mapserv5_nldb?map=/data/cm2</code>	200 OK	10.3 KB	78ms	mt0.google.com
<code>GET /mapserv5_nldb?map=/data/cm2</code>	200 OK	11.7 KB	69ms	mt0.google.com
<code>GET /mapserv5_nldb?map=/data/cm2</code>	200 OK	13.6 KB	68ms	mt0.google.com
<code>GET /mapserv5_nldb?map=/data/cm2</code>	200 OK	14.3 KB	246ms	mt0.google.com

By right-clicking on a URL in Firebug, you can copy the URL location and paste it in to a new browser window, to see exactly what the URL is returning:



- I. By default, errors will show as a 404 not found. This is a general Apache error message that is displayed when a sql/plsql error occurs. You can see the error in the Apache error logs or you can change the DAD to display the error on the web page. To do this, edit the DAD as follows:
 1. Add a new Configuration Parameter as PlsqlErrorStyle ModplsqlStyle, if it is not present already
 2. Restart Apache in order to implement this change
 3. Back in Firefox, using the “Net” tab of Firebug, the error message will be shown on the webpage by right-clicking on the “geojson2” call
- J. For Oracle 11g, you need to enable network callouts for Google Local and Google Address Searching as follows (log in as sys dba to run this code):

```

BEGIN
  DBMS_NETWORK_ACL_ADMIN.create_acl (
    acl      => 'google.xml',
    description => 'Enable Google ACL functionality',
    principal => 'CM2', -- change this if you dont use CM2 as your schema
    is_grant  => TRUE,
    privilege => 'connect',
    start_date => SYSTIMESTAMP,
    end_date  => NULL);

  DBMS_NETWORK_ACL_ADMIN.assign_acl (

```

```

acl      => 'google.xml',
host     => '*.googleapis.com');

DBMS_NETWORK_ACL_ADMIN.assign_acl (
acl      => 'google.xml',
host     => '*.google.com');
COMMIT;
END;

```

8. CorpsMap components:

1. Oracle Database
2. Apache Webserver (with the “mod_plsql” module installed, which allows one to create web applications using Oracle). See http://www.orafaq.com/wiki/Mod_plsql_FAQ for details)
3. Mapserver, which is a free, open-source web-rendering engine for spatial data (see www.mapserver.org)
4. Javascript libraries:
 - a. OpenLayers (see www.openlayers.org)
 - b. Google Maps API (see <http://code.google.com/apis/maps/index.html>)
 - c. GeoExt and ExtJS
 - i. GeoExt imbeds a map with interactive navigation controls (see www.geoext.org/tutorials/quickstart.html)
 - ii. ExtJS is a cross-browser javascript library for building rich internet applications (see www.sencha.com/products/js)

9. CorpsMap’s program flow can be described as follows:

1. Web request is sent to Apache and Apache’s “mod_plsql” module
2. The CM2(5) Oracle PL/SQL package is invoked with parameters, dynamic javascript and static javascript sent to the browser
3. The initial map request is then sent to MapServer
4. Google Maps API request

Once the information is loaded in to the browser:

1. Map Draw requests such as panning and zooming go simultaneously to:
 - a. Google Maps API
 - b. Mapserver
 - c. Underlay / Overlay layers

2. Map query requests are:
 - a. Sent to Oracle CM2 package's "geojson" procedure

If a problem occurs, there are various strategies for debugging:

1. A "404 error" is the Apache Webserver indicating the page has generated an error.
 - a. Error can be viewed in the "dads.conf" file by changing the default "errorstyle"
 - i. PLSQLErrorStyle modplsqlstyle
2. Use the "Firefox" web browser with the "Firebug" plug-in and view "Net Connections":
 - a. Open "geojson2" procedure requests in a new tab to see error
 - b. Missing "grants" are common errors
 - c. To fix, tables need "grant select to CM2"
3. For errors like "PLS-00201:identifier 'DBMS_NETWORK_ACL_ADMIN' must be declared", there may be execute privileges missing. Try issuing the following command in Oracle using SQL Plus, SQL Developer, Toad or any other Oracle SQL Client:
 - a. Grant execute on dbms_network_acl_admin to cm2;

Appendices

Appendix 1

Mapserver Parameters for CM2 Projection – this projection should be defined in every Mapserver "map" file that is to be used with CM2. The projection is a WGS 1984 spherical Mercator projection used by Google Maps, ESRI ArcGIS Online layers and other worldwide mapping layers.

PROJECTION

```
"+proj=merc +a=6378137 +b=6378137 +lat_ts=0.0 +lon_0=0.0 +x_0=0.0 +y_0=0 +k=1.0 +units=m  
+nadgrids=@null +wktext +over +no_defs"  
END
```

UNITS meters

EXTENT -20037508 -20037508 20037508 20037508.34

Appendix 2 – CorpsMap2 Configuration tables

Corpsmap2 Configuration Tables (see next chapter for all columns and descriptions)

- CM2_MAPCONFIG – This table contains a list of all the map applications. For example, here at CRREL, we host numerous CorpsMap applications, including the main “UOC” viewer, the National Levee Database Viewer. Each map application has a unique name in the “MAPNAME” field. The “BASEMAP_FILE” and “OVERLAY_FILE” point to the MapServer “Map” files that list the layers to bring in to map.

The screenshot shows the Oracle SQL Developer interface with the CM2_MAPCONFIG table selected. The table data is as follows:

MAPNAME	BASEMAP_FILE	OVERLAY_FILE	JS_PATH	GOOGLEKEY
1 CIP	/data/cm2_mapfiles/cm2_basemap.map	/htdocs/amy/cip/cip.map	(null)	(null)
2 NAD	/data/cm2_mapfiles/cm2_basemap.map	/htdocs/had/had.map	(null)	(null)
3 FLDS	/data/cm2_mapfiles/cm2_basemap.map	/data/cm2_mapfiles/cm2.map	(null)	(null)
4 MUSSELS	/data/cm2_mapfiles/cm2_basemap.map	/data/cm2_mapfiles/cm2.map	(null)	(null)
5 RMTEST	/data/cm2_mapfiles/cm2_basemap.map	/htdocs/cm2_dev/cm2_test.map	/cm2	(null)
6 GLOBAL	/data/cm2_mapfiles/cm2_basemap.map	/data/cm2_mapfiles/cm2.map	/cm2	(null)
7 GTEST	/data/cm2_mapfiles/cm2_basemap.map	/data/cm2_mapfiles/cm2.map	(null)	(null)
8 BTEST	/data/cm2_mapfiles/cm2_basemap.map	/htdocs/amy/cm2/cm2_agg.map	(null)	(null)
9 NLD	/data/cm2_mapfiles/cm2_basemap.map	/data/cm2_mapfiles/cm2.map	(null)	(null)
10 AEDIS	/data/cm2_mapfiles/cm2_basemap.map	/htdocs/aedis/aedis.map	(null)	(null)
11 UOC	/data/cm2_mapfiles/cm2_radar.map	/data/cm2_mapfiles/cm2_uoc.map	/cm2	(null)

- CM2_MAPLAYERS – This table contains a list of map layers, as well as what maps they will appear in, in the INC_MAPNAME column, and which maps to exclude the layers from, in the EXC_MAP. There is a field for a map layer description, as well as field for a link to the metadata for the layer (a URL)

Oracle SQL Developer : TABLE CM2.CM2_MAPLAYERS@cm2

File Edit View Navigate Run Source Versioning Migration Tools Help

Connections Files

icejam_sql.sql | sql.sql | cm2 | CM2_MAPLAYERS

Columns Data Constraints Grants Statistics Triggers Dependencies Details Indexes SQL

Sort... Filter:

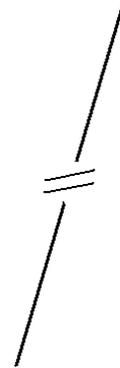
NAME	FULLNAME	QUERYABLE
1 AK_SedimentaryBasins_Project	Sedimentary Basins	N
2 activations	Activations	F
3 actstorms	Hurricane/Tropical Storms	F
4 aedis_ann_dep	Annual Depth	F
5 aedis_ann_load	Annual Load	F
6 aedis_base	Base Layers	F
7 aedis_data	Data Stations	F
8 aedis_mmp	Monthly Mean Precipitation	F
9 aedis_mmt	Monthly Mean Temperature	F
10 aedis_other	Other Layers	F
11 aedis_snow	Snow Data	F
12 ahpsfcst	NOAA Streamgages Forec...	F
13 ahpsfcst_action	Near Flood Stage	Y
14 ahpsfcst_major	Major Flooding	Y

Connections

cm2

Tables

- CM2_MAPCONFIG
- CM2_MAPCONFIG_TEST
- CM2_MAPLAYERS
- CM2_MAPLAYERS_TEST
- CM2_OVERLAY_LAYERS
- CM2_OVERLAY_LAYERS_TEST
- CM2_UNDERLAY_LAYERS
- DR\$COUNTY_NAME_CIDX\$I
- DR\$COUNTY_NAME_CIDX\$K
- DR\$COUNTY_NAME_CIDX\$N
- DR\$COUNTY_NAME_CIDX\$R
- DR\$MAPNAME_CIDX\$I
- DR\$MAPNAME_CIDX\$K
- DR\$MAPNAME_CIDX\$N
- DR\$MAPNAME_CIDX\$R
- MDRT 1625DD\$



ACTIVE	LAYERORDER	MINZOOM	MAXZOOM	LAYERTYPE	URL
Y	10	(null)	(null)	OpenLayers.Layer.Radar	http://radblast.wunderground.com/cgi-bin/radar/WUNIDS_composite
Y	500	(null)	(null)	F	(null)
Y	10	(null)	(null)	OpenLayers.Layer.Radar	http://modelmaps.wunderground.com/php/ndfd.php
Y	20	(null)	(null)	OpenLayers.Layer.Radar	http://modelmaps.wunderground.com/php/ndfd.php
Y	15	(null)	(null)	OpenLayers.Layer.ArcGIS93Rest2	http://zmgsc.cr.usgs.gov/ArcGIS/rest/services/nhss_weat/MapServer/export
N	20	(null)	(null)	OpenLayers.Layer.ArcGIS93Rest2	http://ags.pdc.org/rest/services/GHIN/PDC_Active_Hazards/MapServer/export
N	20	(null)	(null)	OpenLayers.Layer.Vector	(null)
N	20	(null)	(null)	F	(null)
N	10	(null)	(null)	OpenLayers.Layer.ArcGIS93Rest2	http://155.80.100.57/ArcGIS/rest/services/LRD/LRD_Flex_02022010/MapServer/export
Y	100	(null)	216673	OpenLayers.Layer.ArcGIS93Rest2	http://watersgeo.epa.gov/ARCGIS/REST/services/OWPROGRAM/STORET_NAD83/MapServer/expor
Y	30	(null)	(null)	OpenLayers.Layer.Radar	http://modelmaps.wunderground.com/php/ndfd.php
Y	40	(null)	(null)	OpenLayers.Layer.Radar	http://modelmaps.wunderground.com/php/ndfd.php
N	40	(null)	(null)	OpenLayers.Layer.ArcGIS93Rest2	http://events.arcgisonline.com/ArcGIS/rest/services/Gulf_Coast_ESI/MapServer/export

- CM2_UNDERLAY_LAYERS Similar to the CM2_MAPLAYERS table
 - **NOTE:** “NAME” column should not duplicate any names from the other “MAPLAYER” tables
- SDO_CS_SRS_LK table contains look-up values between names of coordinate systems and their Spatial Reference Identification or “SRID”

The screenshot shows the Oracle SQL Developer interface with the SDO_CS_SRS_LK table selected. The table structure and data are as follows:

CS_NAME	SRID	AUTH_NAME	SHORT_LIST	CS_TYPE	EPSG
37 NAD83 / Virginia North	32146 (null)		(null) State Plane ...	(null)	(null)
38 NAD83 / Washington North	32148 (null)		(null) State Plane ...	(null)	(null)
39 NAD83 / West Virginia South	32151 (null)		(null) State Plane ...	(null)	(null)
40 NAD83 / Wisconsin Central	32153 (null)		(null) State Plane ...	(null)	(null)
41 NAD83 / Wyoming East	32155 (null)		(null) State Plane ...	(null)	(null)
42 NAD83 / Wyoming West Central	32157 (null)		(null) State Plane ...	(null)	(null)
43 NAD83 / Indiana East (ftUS)	2965 National Geodetic Survey	http://...	(null) State Plane ...	(null)	(null)
44 NAD83 / Wisconsin North (ftUS)	2287 National Geodetic Survey	http://...	(null) State Plane ...	(null)	(null)
45 NAD83 / Michigan South (ft)	2253 National Geodetic Survey	http://...	(null) State Plane ...	(null)	(null)
46 NAD83 / Florida North (ftUS)	2238 National Geodetic Survey	http://...	(null) State Plane ...	(null)	(null)
47 NAD83 / Arizona West (ft)	2224 National Geodetic Survey	http://...	(null) State Plane ...	(null)	(null)

List of Corpsmap2 Configuration Tables and their Columns

CM2_MAPCONFIG Columns:

- MAPNAME - Identifies a Map configuration
- BASEMAP_FILE - Mapserver based path to the mapfile used when CorpsMap basemap is selected
- OVERLAY_FILE - Mapserver based path to mapfile used for display. Layers controlled via Tree
- JS_PATH - the browser path to the Javascript files and images. This should be the folder containing the MapFish-1.0 directory
- GOOGLEKEY - the Google maps key for your specific server URL. Get from <http://code.google.com/apis/maps>
- MSCGI - path to MapServer CGI binary
- LAYER_URL - URL to procedure generating the layers XML. Usually CM2.CM2.LAYERS?p_mapname=XXXXX
- CM_SCHEMA - the schema that the CM2 package is located in
- ID_QUERY - URL to invoke for an ID based query. Usually:
f?p=APEX_APP_NUMBER:10:0:::P10_LAYER,P10_ID:"+qlid+", "+l_id
- QLOOKUP - deprecated – unused
- COLOR_CSS_PATH - path to an alternative CSS file. Used to "skin" CM2 for different looks. See extjs.com for more skins
- INIT_EXTENT - Comma delimited extent (LL). Used for initial map view if someone doesn't override with local cookie or URL parameter
- MAX_EXTENT - Command delimited extent (LL). The extent to go to when clicking the Max Extent button in CM2 (Left most icon in toolbar).
- REDRAW_LAYER_CHANGE - (Y/empty) - if Y, a change to the layer tree (layer checked) won't draw until Redraw button on Layer Toolbar clicked
- SEARCH_URL - URL to apex search page. Usually: f?p=APEX_APP_NUMBER:5
- MAPTITLE - (html) Sets the Title in the Upper Left-hand corner to this value and on initial load. Can contain graphics/html
- MAPTITLE_TT - Sets the Browser window title bar
- EXCEL_QUERY - URL to apex Excel export page. Export is based on query for that layer.
 - Usually: f?p=APEX_APP_ID:11:0:::P11_LAYER,P11_ID:"+qlid+", "+rowSelection

- EDIT_COMPLETE_URL - Url to invoke on completion of a geometry save.
- SSL - (Y/empty) Y for running on https, empty for http. Sets Cookie types
- INFOLAYER_AUTOSET - (Y/empty) Y to have the InfoLayer set whenever a layer is turned on (that is a queryable layer) in the layer tree.
- LOCAL_JS – (Y/y or null) – If Y then no external APIs are used (eg Google Maps API). This disables all Google Maps API feature such as Street View, Overview Map and Traffic. The Google BaseMaps are replaced by OpenStreetMap.
- CM2_NORTH_REGION – clob – a javascript north region block, see <http://www.extjs.com/deploy/dev/docs/?class=Ext.Panel>.
 - Example:

```
{ region: "north",
                        id: "banner",
                        collapsible: true,
                        collapsed: false,
                        autoHeight: true,
                        layout: "fit",
                        border:false,
                        header:false,
                        contentEl: "cm2NorthDiv" }
```
- CM2_NORTH_DIV – clob – an html div block associated with the contentEl named region

CM2_MAPLAYERS Columns:

Note that the first entry in the layers table is called root and has an empty parent column. All layers at the main tree level should have a parent value of root. root should have a QUERYABLE value of F and an ACTIVE value of Y.

- NAME - MapServer layer name of the layer. Note: For folders this can be any name
- FULLNAME - This is the display Name in the Layer Tree
- QUERYABLE - (Y/N/F/empty) - Y for a querable layer (entry in InfoLayer list), F for a Folder, N or null for non-querable layer
- MINSCALE - The minimum scale at which to grey out the entry in the tree. Should match Mapserver Layer minscale

- MAXSCALE - The maximum scale at which to grey out the entry in the tree. Should match Mapserver Layer maxscale
- TABLE_NAME - Used for queries - The schema.table or schema.view to pull the queried values from.
- LAYERORDER - The numerical order in which to sort the layer. Note that this is in relation to the folder that it is in.
- ACTIVE - (Y/empty) - Y for the Layer to show in the Tree
- WMS - (Y/empty) - Y to add RSS icon. Used to indicate data is pulled from external source
- PRIVATE - not used
- PARENT - the name (column 1) of the folder that contains this layer.
- COLUMNS - the comma delimited columns (all valid SQL can be used) that should display when a layer is queried (a detail query ing the bottom panel) or exported to Excel. To have the columns have a "nice" column heading, enclose the column heading in double quotes. Note that using standard sql concatenation and single quotes, you can construct hyperlinks as a column. Example: name "Name", initcap(project_manager) "PM", ''||description||'' "Hyperlink"
- POPUP - This is the information displayed when the initial query is done and used for the Quick Query display. Its has a very specific format. It consists of 4 values, comma delimited. They are Value1, Value2, Longitude, Latitude. Value1 and value2 can be any column or sql value. Value1 will be displayed with the column heading NAME and Value2 will have the column heading DESC (for Description). Longitude and Latitude should be numeric (-180/+180/-90/+90) values. If derived from spatial column, use must use an alias (sql object access rules). The value is predefined as "d".
 - Examples:
 - Point spatial column (named shape): d.shape.sdo_point.x, d.shape.sdo_point.y
 - Line spatial column (named geom): sdo_geom.sdo_pointonsurface(d.geom,0.1).sdo_point.x, sdo_geom.sdo_pointonsurface(d.geom,0.1).sdo_point.y
 - Polygon spatial column (named shape): sdo_geom.sdo_centroid(d.shape,0.1).sdo_point.x, sdo_geom.sdo_centroid(d.shape,0.1).sdo_point.y
 - Complete entry: initcap(installation), description, d.shape.sdo_point.x, d.shape.sdo_point.y
- ID_COL: this is the unique identifier for the table (column name).
- MAPNAME - deprecated - no longer used
- DESCRIPTION - This is text (can be html) that is displayed when "hovered" over the layer in the Layer Tree.
- METADATA_URL - This is a URL to a page containing more info (ideally metadata of some sort) for this particular layer. The layer will be a clickable hyperlink if this is filled out

- QQ_COL = stands for quick query column. The Box on the top bar is the quick query box. It now changes with the InfoLayer. The column name here is what is queried when someone types a value in the quick query box for the selected infolayer. What is displayed is the first 2 columns of the POPUP column. The column should have a function based index for maximum performance on the lower(column_name).
 - Example: create index index_name on table_name(lower(column_name));
- QQ_TEXT = The text to place in the QuickQuery box when the associated infolayer is selected
- Table_srid = set the spatial srid (oracle value) of the table to be queried (usually 8265, 8307 or null). Its used to calculate the tolerance for the click on the map of queries
- FOLDER_CHECK = Adds a check box at the folder level. When checked, all layers below that level are turned on.
- INC_MAPFILE - a comma delimited list of mapnames to include this layer
- EXC_MAPFILE - a comma delimited list of mapnames to exclude this layer
- LAYEREXT – Allows you to put in a comma delimited (lat/lon decimal degrees) extent (minx, miny, maxx, maxy format). If this column is filled out, you can right click on this layer in the layer tree and the map will zoom to the layer extent. Useful for layers with small extents.
- SHAPECOL – if the geometry column of your query table (entered into TABLE_NAME) is not SHAPE, you can put the geometry column name here (example: GEOM). If null, SHAPE is used.
- POPUP_SHAPE – The new multi-field popups now show the existing shape geometry highlighted when clicked on the map. However, some geometries may be too complex or too time consuming to display (typically polygons with high number of vertices). In these cases, you can put an alternative geometry function in this column and it will be used instead. Example: SDO_GEOM.SDO_CONVEXHULL(shape,0.1). This will highlight the convex hull of the feature in question. You could also use something like SDO_GEOM.SDO_CENTROID(shape,0.1) and get a point return.
- QQ_COL_RETURN – with the new multi-field popups, the POPUP column is no longer used. Therefore, a new column was needed for the 1-2 return columns of the Quick Query bar. The two columns are row1 and row 2 that are returned from a quick query search. Note that neither of these columns needs to be the same as the QQ_COL (which is the searched column) but it can be. If you want this the same as your CM2 2.0 setup, populate with the first 2 columns of the POPUP column
- USE_OTEXT_QQ – (Y/y or null) switches the QQ lookup to use an Oracle Text index on the QQ_COL. The Oracle Text index must exist on this column. The QQ query lookup function uses INSTR function on this QQ_COL. This is a non indexed function but is suitable for most layers (<10K records). For large searches, create an Oracle Text index on the column to search and set this column to Y.

SDO_CS_SRS_LK Table Columns

- CS_NAME - The text name of the projection. This is what will appear in the menus
- SRID - This is the Oracle Spatial Reference ID. Must be a valid value in MDSYS.SDO_CS_SRS
- AUTH_NAME - (NOT USED)
- SHORT_LIST - (1/0/null) - Put 1 in here for the Coordinate system to appear on the Search List and the Reproject List. Note that NAD27 projections do NOT get datum re-projections in the on-the-fly re-projections on the bottom toolbar. The click to get Point tool does do datum re-projection.
- EPSG – this value needs to be the EPSG code of the projection in question. This is used for calculating the on-the-fly display projection of coordinates. To look up EPSG codes for projections, the website <http://www.spatialreference.org> is very useful. In most cases (State Plane and UTM projections), the Oracle SRID = the EPSG code.

CM2_OVERLAY_LAYERS Table Columns

- NAME – same as maplayers table. A simple unique name (among all the layers of the tree. This means you do not want duplicates of CM2_MAPLAYERS or the UNDERLAY or OVERLAY entries)
- FULLNAME – how it appears in the tree. Note that you can use full html syntax here
- PARENT – the parent entry to this one. Must match either an existing NAME entry or the root entry. Used for the tree hierarchy.
- ACTIVE – Y/y to display in the tree. Anything else (or null) will disable
- LAYERORDER – a numeric value used to sort the entries. Note that the sorting only happens at the current tree level. Changing the LAYERORDER will not move an entry above its PARENT
- MINZOOM – The minimum zoom level (1-21) of this layer. Used similar to MAXSCALE but based on Google/OpenLayer Zoom Levels. Used to grey out/disable the tree entry
- MAXZOOM – The maximum zoom level (1-21) of this layer. Used similar to MINSSCALE but based on Google/OpenLayer Zoom Levels. Used to grey out/disable the tree entry
- LAYERTYPE – An OpenLayers layer type (<http://dev.openlayers.org/releases/OpenLayers-2.8/doc/apidocs/files/OpenLayers-js.html>). Example: OpenLayers.Layer.XYZ for an XYZ based layer
- URL – if the LAYERTYPE takes a URL parameter, put it here. Note that for VECTOR layers, since these return data rather than an image, the source URL need to be in the local domain due to javascript security requirements. Can be a relative URL.

- PARAMS – The parameters for the layer in key value pair javascript object format. This must be wrapped in {} (braces). If the layer type doesn't take params, leave null. If it does and you have no parameters, put in empty braces, eg {}.
- OPTIONS – the options for the layer in key value pair javascript object format. This must be wrapped in {} (braces). Note that you will want to include at a minimum {visibility: false, isBaseLayer: false} for proper tree control
- DESCRIPTION - This is text (can be html) that is displayed when "hovered" over the layer in the Layer Tree.
- METADATA_URL - This is a URL to a page containing more info (ideally metadata of some sort) for this particular layer. The layer will be a clickable hyperlink if this is filled out.
- INC_MAPFILE - a comma delimited list of map names to include this layer
- EXC_MAPFILE - a comma delimited list of map names to exclude this layer
- FOLDER_CHECK – Adds a check box at the folder level. When checked, all layers below that level are turned on.
- AUTOUPDATE – An interval value (in minutes) to trigger a layer reload
- TREE_EXTRA – extra HTML text to place in the tree after the FULLNAME
- LEGEND_URL – a URL to load a image (must be a supported browser image type (GIF/JPG/PNG) to display in the legend panel.

CM2_UNDERLAY_LAYERS Table Columns

Same as CM2_OVERLAY_LAYERS except for

- OPACITY – Used to set the opacity for UNDERLAY layers. For an unknown reason, opacity settings in the OPTIONS column wouldn't be set for UNDERLAY layers. May become deprecated in the future as this problem gets solved.

URL Parameters

- p_mapExt: Comma delimited Extents - Overrides previous value from cookie
- p_layers: Comma delimited list of checked layers - Overrides cookie values. In combination with P_UNIQUE_ID and P_MAPTYPE=VIEW will zoom and highlight and show a popup for the selected ID
- p_maptypes: (view/edit) Start CM2 in View mode (no edit tools) or in Edit mode (full Editing Toolset) . Default value is VIEW if not specified
- p_unique_id: PK value of geometry to retrieve/save when editing/creating geometry. In combination with P_UNIQUE_ID and P_MAPTYPE=VIEW will zoom and highlight and show a popup for the selected ID
- p_editlayer: layername passed to geometry store package to allow editing of multiple layers

- p_geom_api: url to geometry storage package (usually cm2_geom_util.saveShape)
- p_geom_api_get: url to geometry retrieval package (usually cm2_geom_util.getShape)
- p_role_id: passed to p_geom_api for role based editing
- p_basemap: (geh/gem/ges/gec/cm/topo) meaning google hybrid/streets/satellite/terrain/corpsmap/usgs topo. overrides cookie based basemap
- p_complete_url: allows override the mapconfig parameter of the URL to go after saving a geometry
- p_search_session: id of the search_session_table to retrieve results into the multi-query panel
- map: selects values row in cm2_mapconfig table

Appendix 3 - Technology Overview

CorpsMap components:

1. Oracle Database
2. Apache Webserver with the “mod_plsql” module, which allows creation of web applications using Oracle
3. MapServer – an open-source Web rendering engine
4. Javascript libraries, including:
 - a. OpenLayers
 - b. Google Maps API
 - c. GeoExt
 - d. ExtJS

CorpsMap Program Flow:

1. A web request is sent to Apache and the “mod_plsql” module
2. The “CM2” Oracle PL/SQL package is invoked with parameters, dynamic javascript and static javascript is sent to the browser
3. The initial map request is sent to MapServer
4. The Google maps API request is sent
5. Requests are processed and loaded in to the browser
6. Map draw requests such as panning and zooming go simultaneously to:
 - a. Google Maps API
 - b. MapServer
 - c. Underlay/Overlay layers (other web mapping services such as: _____??)
7. Map Query requests (identify tools, etc) are sent to:
 - a. the “CM2” Oracle PL/SQL package’s “geojson” procedure

CorpsMap Program Flow Troubleshooting:

1. A “404” Error indicates that the Apache webserver has a page error. Change the default errorstyle to see error in the dads.conf file
 - a. PLSQLErrorStyle modplsqlstyle
2. Use the “Firefox” browser with “Firebug” extension installed, and choose “View Net Connections” in Firebug to:
 - a. Open “geojson2” requests in a new tab, in order to see error
 - b. Missing grants are common errors
 - c. Tables need grant select to CM2 user.

Appendix 4 – Frequently Asked Questions

1. What map projection does CorpsMap use?
 - A. CorpMap uses a spherical Mercator map projection, the same projection as Google Maps, Bing Maps and ESRI ArcGIS Online Resource Center maps
2. Where is the Oracle Apache Web Server (Oracle HTTP Server or OHS) Installed?
 - A. Oracle apache web server is installed on your web server. Typically in folder: oracle\product\10.1.3\ohs
3. Where is MapServer Installed?
 - A. Mapserver is installed inside the directory tree of your OHS (Oracle HTTP Server). It is in the cgi-bin folder. There will be a mapserv.exe as well as a set of DLLs
4. How can I tell if the Oracle Apache Webserver is running or not?
 - A. By and large, the webserver is either running or not. If it responds at all, it is running. **To Do how do I test to see if it responds?** A bigger issue is whether it can communicate with the database. It does this via a “mod_plsql” configuration file containing the “Data Access Descriptors”. The file is called “dads.conf” and is located in the “ohs/modplsql/conf” folder.
5. Does Apache request the map from MapServer or does the initial web request from the user initiate the generation of the map through MapServer?
 - A. Technically, the user requests the map from the web server, the web server passes the request to MapServer and MapServer will initiate 1..N connections and web callouts (for WMS services) to get the data, render it based on the “mapfile” parameters and then return an image back to the web server which returns it to the user.

The CorpsMap application initiates these requests, and the requests can be viewed using the “Firebug” extension in the Firefox web browser, which can be downloaded here: www.getfirebug.com

6. How does CorpsMap use the Google Maps API?
 - A. CorpsMap uses the Google Maps API (not the Google Earth API) in the following way: when you call the URL for CorpsMap, it runs the Oracle PL/SQL stored procedure “CM2” (or cm25) and this creates javascript and HTML that is returned to the user’s browser, which in turn contacts Google and loads the Google Maps API using our enterprise key. The enterprise key is our permission to use the Google Maps background in the application
7. Where are the JavaScript libraries located?
 - A. The JavaScript libraries are present in your ohs/htdocs/cm2 directory. These are accessed by the client and provide a lot of the base functionality of CorpsMap. The JavaScript that I have written usually uses the JavaScript libraries to accomplish a variety of tasks (for example, displaying the Popup table of column results). The basic JavaScript libraries know how to create the popup, my code tells what data to populate, how it should be sized, where it should go, etc.
8. What is the “wrapper” around the CorpMap application?
 - A. The wrapper is an Oracle Apex application that was developed to implement the multiple tabs. All it does is add multiple tables on the top right and several dropdown boxes (for quick zooming) and a bottom pane that calls CorpsMap. CorpsMap can be called with or without the wrapper. One could create and many wrappers as they wish. They are all independent.
9. Where do all the map requests go to?
 - A. Any request that goes to your web server, goes to Apache, but not all requests that CorpsMap makes goes to your web server. The Google Maps calls go from the browser directly to Google. Underlay and overlay layer calls go directly from the browser to the server URL listed in those tables (which is why the weather radar can work independent of whether MapServer is running). The definition of the under/overlay layers are stored in the database but once the client browser has the page loaded, it can then make direct calls to those servers. This is why that all under/overlay layers need to be in the Google Maps projection (spherical Mercator). Since the calls are being make directly from the browser, the data cannot be re-

projected on the fly and needs to overlay directly on the Google Maps base maps. Calls to data that go through MapServer can be in a variety of projections and MapServer will re-project that data on the fly.

10. Is SDE used in CorpsMap?

A. SDE is not used directly in the CorpsMap web viewer, but it is part of the CorpsMap server deployment since it provides direct access to the same data as the web viewer to ESRI Clients. The goal of CorpsMap is to provide seamless access to one set of data to as many clients as possible, regardless of whether the dataset is requested in a web viewer such as the CorpsMap viewer or an ESRI client such as ArcGIS.

11. Can ArcGIS Server layers be used in CorpsMap?

A. ArcGIS Server works similarly to MapServer although not all features of CorpsMap Web viewer are supported at this time. You can display data layers from ArcGIS server (9.3 or later). In version 2.6 of CM2, query (clicking at a location and seeing the attribute data) will be supported. Version 2.6 should be available sometime after the ESRI conference.

12. How do I edit MapServer Map Files?

A. To edit MapServer Map Files, use Notepad, Notepad ++ or other text editor. Format is very important for the map file to work. MSCompanion is a free editor available at <http://code.google.com/p/mscompanion/> but does require MapServer components in order to run. Notepad ++ is available for free at <http://sourceforge.net/projects/notepad-plus/>

13. How do I exclude a layer from the map and Table of Contents?

A. To exclude from the GLOBAL (default) Table of Contents, put GLOBAL in the EXC_MAPNAME column. Alternatively, if you only want it included in a few (or one map), then use the INC_MAPNAME and it will only show in those map instances

Appendix 5 – CorpsMap Developer site and updates

- Updates to CorpsMap are downloadable from <https://software.forge.mil/sf/projects/corpsmap>
 - This site is CAC-enabled
 - After creating an account on software.forge.mil, request access to the “corpsmap” workspace
 - If you have trouble accessing the workspace, contact Mike Smith at CRREL for permission to the site

Appendix 6 – Other information web-accessible

- For information to be accessible for any web application (such as CorpsMap) then it must reside under the “htdocs” directory
- Examples are metadata files, the help file, and the help videos
- It is possible to use a redirection to simplify the URL to the eGIS web map. It is called **index.html** and is located in the htdocs directory.

Appendix 7 – Apex, Web browsers

- **Apex** stands for Oracle Application Express. Apex is essentially a wrapper around the GIS functionality of the CM2 application. The CM2 web map is called and delivered via Apex.
- **Browser Notes:** Supported browsers include: Firefox 3.6, Internet Explorer 7 & 8, Google Chrome. Firefox and Chrome tend to be about 50% faster than IE8.
- There are differences in how the web map portal performs some functions within various web browsers. Some of these differences are described below.
 - Internet Explorer: Measurement tools (Area and Distance)- The second mouse click to end the measurement often does not ‘take’. These tools work better within FireFox or Google Chrome.
 - FireFox: Some web pages or files that are accessed from our internal file system (not the web) do not display correctly. Metadata files stored locally is the best example of this. Google Chrome and Internet Explorer work better.

Appendix 8 - Future Developments under consideration:

- ArcGIS Server query support (popup, multiquery and quick query). Standalone tool to insert/update a tree structure from ArcGIS server service to CM2 overlay/underlay trees.
- PDF print output support. Multipage printing, multiple layouts and templates. Will require a Java Servlet for PDF generation (OC4J/Tomcat).

- Updates to CorpsMap will be made accessible from <https://software.forge.mil/sf/projects/corpsmap>

Appendix 9 – Sample Map File

Everything below this line represents a sample Mapserver “map” file (cm2_basemap_example.map)

```
MAP
NAME cmol
STATUS ON
SIZE 700 500
TRANSPARENT off
SYMBOLSET /htdocs/symbols/symbol_orm.sym
FONTSET /htdocs/fonts/fonts.list

MAXSIZE 4096
DEBUG on

IMAGECOLOR 179 198 212 #1 2 14
IMAGETYPE PNG
DATAPATTERN ".*"

OUTPUTFORMAT
NAME "AGGA"
DRIVER "AGG/PNG"
IMAGEMODE "RGB"
END

OUTPUTFORMAT
NAME "PNG"
DRIVER "AGG/PNG"
IMAGEMODE "RGB"
END

PROJECTION
# "proj=aea"
```

```
# "lon_0=-96"
# "lat_0=23"
# "lat_1=29.5"
# "lat_2=45.5"
# "x_0=0"
# "y_0=0"
"init=epsg:900913"
#PROJECTION
#      "+proj=merc +a=6378137 +b=6378137 +lat_ts=0.0 +lon_0=0.0 +x_0=0.0 +y_0=0 +k=1.0 +units=m
+nadgrids=@null +wktext +no_defs"
END
#" +proj=merc +a=6378137 +b=6378137 +lat_ts=0.0 +lon_0=0.0 +x_0=0.0 +y_0=0 +k=1.0 +units=m
+nadgrids=@null +no_defs"

#END
UNITS meters
#UNITS dd
#EXTENT -180 -90 180 90
EXTENT -20037508 -20037508 20037508 20037508.34

WEB
TEMPLATE "corpsmap.html"
IMAGEPATH "/htdocs/tmp/"
IMAGEURL "/tmp/"
METADATA
    "wms_title" "ORMGIS"
    "wms_srs" "EPSG:4269 EPSG:4326"
    "wms_onlineresource" "https:// <server name> /cgi-bin/wms/orm?"
    "wms_abstract" "Basemap for ORM"
    ###
        "wfs_title" "ORMGIS"
    "wfs_srs" "EPSG:4269"
    "wfs_onlineresource" "https:// <server name> /cgi-bin/wfs/orm?"
```

```
"wfs_abstract" "Basemap for ORM"  
END  
END
```

```
LEGEND  
IMAGECOLOR 255 255 255  
KEYSIZE 18 12  
LABEL  
TYPE TRUETYPE  
FONT "tahoma"  
SIZE 8  
COLOR 0 0 0  
END  
STATUS on  
END
```

```
SCALEBAR  
STYLE 0  
IMAGECOLOR 255 255 255  
LABEL  
FONT "tahoma"  
COLOR 0 0 0  
SIZE tiny  
BUFFER 3  
BACKGROUNDCOLOR 255 255 255  
END  
SIZE 150 3  
COLOR 255 255 255  
BACKGROUNDCOLOR 0 0 0  
OUTLINECOLOR 0 0 0  
UNITS miles  
INTERVALS 4  
STATUS embed
```

```
TRANSPARENT true
END
```

```
###
### BASE TEMPLATE LAYERS (NOW SHAPEFILES)
###
```

```
LAYER
  NAME cntry01
  GROUP "cntry"
  DATA '/fire/world/esri/country'
  # CONNECTIONTYPE oraclespatial
  # CONNECTION 'user/password@database'
  # DATA "SHAPE from ESRI_WORLD.COUNTRY using SRID 8265"
  # PROCESSING "CLOSE_CONNECTION=DEFER"
  PROJECTION
    "proj=latlong"
    "ellps=GRS80"
  END
  METADATA
    "wms_title" "Country Boundaries"
    "wms_srs" "EPSG:4269 EPSG:4326"
    "wms_abstract" "ESRI World-Wide Layer: Country Boundaries (cntry02)"
    "wms_extent" "-177.47 13.43 -64.71 72.55"
  END
  TYPE polygon
  STATUS default
  MINSCALE 70000000
  CLASS
    SYMBOL 0
    COLOR 165 215 165
    OUTLINECOLOR 125 159 125
  END
```

END

LAYER

NAME admin01

GROUP "admin"

DATA '<server name>/world/esri/country'

CONNECTIONTYPE oraclespatial

CONNECTION 'user/password@database'

DATA "SHAPE from ESRI_WORLD.ADMIN using SRID 8265"

PROCESSING "CLOSE_CONNECTION=DEFER"

PROJECTION

"proj=latlong"

"ellps=GRS80"

END

METADATA

"wms_title" "Administrative Boundaries"

"wms_srs" "EPSG:4269 EPSG:4326"

"wms_abstract" "ESRI World-Wide Layer: Administrative Boundaries (admin02)"

"wms_extent" "-177.47 13.43 -64.71 72.55"

END

TYPE polygon

STATUS default

MAXSCALE 70000000

CLASSITEM "GMI_CNTRY"

CLASS

EXPRESSION "USA"

SYMBOL 'dummy'

END

CLASS

EXPRESSION /\

SYMBOL 0

COLOR 165 215 165

OUTLINECOLOR 125 159 125

```
END
END
###
END #-> end mapfile
```

Appendix 10 – Adding data via Oracle “Materialized View”

Example:

Retrieve the following code from the software.forge.mil site (you’ll need a CAC card and Internet Explorer to get to the site. Log in, and navigate to the “CorpsMap” Project.

Go to the “File Releases” tab, and click [CPN_Netlink.sql](#) to download the code.

SoftwareForge : View Release - Windows Internet Explorer

https://software.forge.mil/sf/frs.do/viewRelease/projects.corpsmap/frs.data_feeds_for_cm2.base_release

software Forge

This page contains dynamic content -- Highest classification is UNCLASSIFIED//FOUO

HELP LOG IN

PROJECTS SEARCH ENTERPRISE TOOLS SUPPORT TRAINING FAQ

Project: CorpsMap Jump to ID: rel1349 GO

Project Home Tracker Documents Tasks Source Code Discussions Reports File Releases Wiki

File Releases > Data Feeds for CM2 > Materialized Views > View Release

Release: rel1349

Name: Materialized Views Status: Active
Description: Maturity: General Availability

Files Reported Tracker Artifacts Fixed Tracker Artifacts Associations

File Name	Size	Created On	Last Modified On
<input type="checkbox"/> earthquakes.sql	8.21 KB	04/12/2010 9:45 AM	08/16/2010 5:49 PM
<input type="checkbox"/> Fema_Q3.txt	1.73 KB	04/12/2010 9:46 AM	04/12/2010 9:46 AM
<input type="checkbox"/> FWS_Critical_Habitat.map	514 bytes	04/12/2010 9:47 AM	04/12/2010 9:47 AM
<input type="checkbox"/> nationwide_fires_rss.sql	2.75 KB	04/12/2010 9:48 AM	04/12/2010 9:48 AM
<input type="checkbox"/> NHD.map	3.98 KB	04/12/2010 9:49 AM	04/12/2010 9:49 AM
<input type="checkbox"/> Drought_Monitoring.map	3.31 KB	04/12/2010 9:59 AM	04/12/2010 9:59 AM
<input type="checkbox"/> marine_traffic_realtime.sql	7.87 KB	04/12/2010 10:02 AM	04/12/2010 10:02 AM
<input type="checkbox"/> noaa_tide_gages.sql	5.38 KB	04/12/2010 10:11 AM	04/12/2010 10:11 AM
<input type="checkbox"/> P2Data_MView.sql	6.53 KB	04/12/2010 10:16 AM	04/12/2010 10:16 AM
<input type="checkbox"/> VB_Sat.sql	734 bytes	04/12/2010 10:18 AM	04/12/2010 10:18 AM
<input type="checkbox"/> CPN_Netlink.sql	694 bytes	05/05/2010 3:05 PM	05/05/2010 3:05 PM
<input type="checkbox"/> COE Bridge Database MV.sql	5.67 KB	10/28/2010 12:49 PM	11/06/2010 11:35 AM

Download Release

Return

Done Internet 100%

CREATE DATABASE LINK "CPN.LOCAL"
CONNECT TO MS

```
IDENTIFIED BY ms
USING '(DESCRIPTION=(ADDRESS_LIST=(ADDRESS = (PROTOCOL = TCP)(HOST = goldldb.crrel.usace.army.mil)(PORT =
1521))) (CONNECT_DATA = (SID = goldldb)))';
```

```
create materialized view cpn_project_list
refresh complete on demand
as
select * from cpnp.cpn_project_list@cpn.local;
```

```
insert into user_sdo_geom_metadata values (
    'CPN_PROJECT_LIST','SHAPE',
    sdo_dim_array(
        sdo_dim_element('longitude',-180,180,0.0005),
        sdo_dim_element('latitude',-90,90,0.0005)),
    8265);
commit;
```

```
CREATE INDEX CPN_PROJECT_SIDX ON CPN_PROJECT_LIST
(SHAPE)
INDEXTYPE IS MDSYS.SPATIAL_INDEX
NOPARALLEL;
```

Using SQL Developer

Exporting a table (known as "unload" in Oracle)

Oracle SQL Developer > highlight table in list, click Tools > Unload Data

- Choose connection
- Browse under save as and choose xls as the format

Unload Wizard - Step 1 of 5

Source/Destination

Source/Destination
Types to Unload
Specify Objects
Specify Data
Unload Summary

Connection: gold1db_cm2_basic_connection

Unload DDL

Pretty Print Include Dependents Include Grants Storage
 Terminator Include BYTE keyword Include Drops
 Show Schema Add Force to Views Cascade Drops

Unload Data

Format: xls Header

Data Worksheet Name:

Query Worksheet Name:

Save As: Single File C:\temp\exp_cm2_overlay.xls Browse...

Encoding: default

Proceed to summary.

Help < Back Next > Finish Cancel

Unload Wizard - Step 3 of 4

Specify Data

Source/Destination
Types to Unload
Specify Data
Unload Summary

Name: %

CM2_OVERLAY_LAYERS

Database Object	Columns	Where
CM2.CM2_OVERLAY_L...	*	