

## Transferring Local Maps from Operational AWIPS to WES-2 Bridge

These local maps are an important part of your situational awareness besides being integral to calculations of FFMP and being used by your WarnGen templates. So to have as much operational representativeness as possible with your WES-2 Bridge, you need to keep it up to date with the local maps from your operational AWIPS system.

The process is to

- (1) determine the list of local maps
- (2) extract or dump the maps and corresponding metadata from the database
- (3) transfer the resulting files to the WES-2 Bridge machine by placing them into the proper location with the expected naming convention.

These local maps came from actual shapefiles (.shp, .shx, and .dbf files) that were imported into the maps PostgreSQL database by config\_awips2.sh. These actual shapefiles are stored in edex\_static/site/<site>/shapefiles in the Utility Tree, but are not actually used by the AWIPS-2 software (it uses the database version instead). Therefore, there is no need to transfer the actual shapefiles to WES-2 Bridge, because they are not used, and they can consume significant amounts of disk space.

### Step 1. Determine the list of local maps.

To determine the list of local maps as compared to the default maps, you need to examine the list of default maps. Below is a report of the set of 35 default maps as of Build 14.3.1. This list was generated by performing a “\dt” command using psql.

*Note: Throughout this exercise, the red text is the output from the computer, and the black text is what you would type.*

#### List of relations

Schema	Name	Type	Owner
mapdata	airport	table	awips
mapdata	allrivers	table	awips
mapdata	artcc	table	awips
mapdata	basins	table	awips
mapdata	canada	table	awips
mapdata	city	table	awips
mapdata	county	table	awips
mapdata	customlocations	table	awips
mapdata	cwa	table	awips
mapdata	firewxaor	table	awips
mapdata	firewxzones	table	awips
mapdata	fix	table	awips
mapdata	highaltitude	table	awips

```

mapdata | highsea          | table | awips
mapdata | highway              | table | awips
mapdata | hsa                  | table | awips
mapdata | interstate           | table | awips
mapdata | isc                  | table | awips
mapdata | lake                 | table | awips
mapdata | latlon10            | table | awips
mapdata | lowaltitude         | table | awips
mapdata | majorrivers         | table | awips
mapdata | map_version         | table | awips
mapdata | marinesites         | table | awips
mapdata | marinezones         | table | awips
mapdata | mexico              | table | awips
mapdata | navaid              | table | awips
mapdata | offshore            | table | awips
mapdata | railroad            | table | awips
mapdata | rfc                 | table | awips
mapdata | specialuse          | table | awips
mapdata | states              | table | awips
mapdata | timezones           | table | awips
mapdata | world               | table | awips
mapdata | zone                | table | awips
(35 rows)

```

You will need to use psql on dx1 to determine the set of maps you want to transfer. You need to look at the maps database and to look at the mapdata schema within the maps database. The non-baseline maps in the resulting output from psql's "/dt" command are highlighted below in this example from the LOT WFO.

```

dx1-lot$ psql maps -U awips
psql (9.2.4)
Type "help" for help.

```

```

maps=# set search_path=mapdata;
SET
maps=# \dt

```

```

                List of relations
 Schema | Name | Type | Owner
-----+-----+-----+-----
mapdata | airport | table | awips
mapdata | airportdiagrams | table | awips
mapdata | airport hubs | table | awips
mapdata | airportmap | table | awips
mapdata | allrivers | table | awips
mapdata | arrivalgates | table | awips
mapdata | artcc | table | awips
mapdata | basins | table | awips
mapdata | c90_boundary | table | awips

```

mapdata	canada	table   awips
mapdata	chicago_key_locations	table   awips
mapdata	city	table   awips
mapdata	county	table   awips
mapdata	customlocations	table   awips
mapdata	cwa	table   awips
mapdata	cwsu_vor	table   awips
mapdata	departuregates	table   awips
mapdata	ffmp_basins	table   awips
mapdata	ffmp_streams	table   awips
mapdata	firewxaor	table   awips
mapdata	firewxzones	table   awips
mapdata	fix	table   awips
mapdata	golfcourses	table   awips
mapdata	highaltitude	table   awips
mapdata	highsea	table   awips
mapdata	highway	table   awips
mapdata	hsa	table   awips
mapdata	hydrohighhazarddams	table   awips
mapdata	hydrositespecific	table   awips
mapdata	i190_mile_markers	table   awips
mapdata	i290_mile_markers	table   awips
mapdata	i290mm	table   awips
mapdata	i294mm	table   awips
mapdata	i355_mile_markers	table   awips
mapdata	i355mm	table   awips
mapdata	i39_mile_markers	table   awips
mapdata	i39mm	table   awips
mapdata	i55_mile_markers	table   awips
mapdata	i55mm	table   awips
mapdata	i57_mile_markers	table   awips
mapdata	i57mm	table   awips
mapdata	i80_mile_markers	table   awips
mapdata	i88_mile_markers	table   awips
mapdata	i90_mile_markers	table   awips
mapdata	i94_mile_markers	table   awips
mapdata	il180mm	table   awips
mapdata	il188mm	table   awips
mapdata	il190mm	table   awips
mapdata	il194mm	table   awips
mapdata	il_mile_markers	table   awips
mapdata	in65mm	table   awips
mapdata	in80mm	table   awips
mapdata	in90mm	table   awips
mapdata	in94mm	table   awips
mapdata	in_mile_markers	table   awips
mapdata	inmm_final2	table   awips
mapdata	int	table   awips

mapdata	interstate	table   awips
mapdata	interstatecwa	table   awips
mapdata	interstates	table   awips
mapdata	isc	table   awips
mapdata	keylocations	table   awips
mapdata	lake	table   awips
mapdata	latlon10	table   awips
mapdata	lowaltitude	table   awips
mapdata	majorhighways	table   awips
mapdata	majorhighwayslabel	table   awips
mapdata	majorrivers	table   awips
mapdata	map_version	table   awips
mapdata	marinesites	table   awips
mapdata	marinezones	table   awips
mapdata	mexico	table   awips
mapdata	navaid	table   awips
mapdata	oemczones	table   awips
mapdata	offshore	table   awips
mapdata	ordarrivalgates	table   awips
mapdata	orddeparturegates	table   awips
mapdata	railroad	table   awips
mapdata	rfc	table   awips
mapdata	rivers	table   awips
mapdata	roc_wind_points	table   awips
mapdata	specialuse	table   awips
mapdata	statehighways	table   awips
mapdata	statehighwayslabel	table   awips
mapdata	states	table   awips
mapdata	tafbuffer	table   awips
mapdata	timezones	table   awips
mapdata	warnngen_dss	table   awips
mapdata	warnngenloc	table   awips
mapdata	webcities	table   awips
mapdata	wind_farm_outline_nssl	table   awips
mapdata	windfarms	table   awips
mapdata	world	table   awips
mapdata	zau_sectors	table   awips
mapdata	zau_sectors_labeled	table   awips
mapdata	zone	table   awips

maps=# \q

## Step 2. Extract the maps and metadata from the database.

The maps themselves are stored in individual tables within the mapdata schema, but there is also a corresponding metadata record in the map\_version table.

Once you identify the tables that contain the local maps, then you will use the `pg_dump` command to save a copy of the table to disk. The reason we use this approach is to ensure an exact copy of the database is transferred to the WES. It is also faster in WES-2 Bridge to restore a file from `pg_dump` than it is to import the raw shapefiles every time they might be needed. (In addition, WES-2 Bridge does not use `config_awips2.sh`.)

The `pg_dump` command allows multiple tables to be placed into one resulting dump file, with either multiple `-t` switches (`-t mapdata.statehighways -t mapdata.statehighwayslabel`) or using a wildcard for the table names (`-t 'mapdata.state*'`). You can also use wildcards for convenience in typing or for logical grouping of multiple tables together.

We also need to get corresponding records from the `map_version` table and put them in a `.CSV` file, using `psql`. These records contain the name of the shapefile, the database table name and the time when the table was updated with that particular shapefile. WES-2 Bridge uses this information to keep track of which tables are loaded and the validity of the tables (this is critically important for FFMP). WES-2 Bridge imposes a naming convention on these files. The filename must begin with the WFO ID and an underscore (e.g., `LOT_`) and it needs to end with `"_map.dump"` or `"_map.csv"`. The `.dump` and `.csv` filenames need to match. Examples of generating these files for the LOT WFO follow.

We can use wildcards for the table names in `psql` (`%`) to match the ones we used in `pg_dump` (`*`). There are some special cases, which are outlined in this online documentation:

<http://www.postgresql.org/docs/9.4/static/app-pgdump.html#PG-DUMP-EXAMPLES>

Using `pg_dump`:

```
dx1-lot$ pg_dump -Fc -t 'mapdata.airport[dhm]*' maps -f /data/fxa/LOT_airportstuff_map.dump -U awips
```

```
dx1-lot$ pg_dump -Fc -t 'mapdata.arrival*' maps -f /data/fxa/LOT_arrival_map.dump -U awips
```

```
dx1-lot$ pg_dump -Fc -t 'mapdata.c90*' maps -f /data/fxa/LOT_c90_boundary_map.dump -U awips
```

```
dx1-lot$ pg_dump -Fc -t 'mapdata.chi*' maps -f /data/fxa/LOT_chicago_key_locations_map.dump -U awips
```

```
dx1-lot$ pg_dump -Fc -t 'mapdata.cwsu*' maps -f /data/fxa/LOT_cwsu_map.dump -U awips
```

```
dx1-lot$ pg_dump -Fc -t 'mapdata.dep*' maps -f /data/fxa/LOT_departure_map.dump -U awips
```

```
dx1-lot$ pg_dump -Fc -t 'mapdata.ffmp*' maps -f /data/fxa/LOT_ffmp_map.dump -U awips
dx1-lot$ pg_dump -Fc -t 'mapdata.golf*' maps -f /data/fxa/LOT_golf_map.dump -U awips
dx1-lot$ pg_dump -Fc -t 'mapdata.hydro*' maps -f /data/fxa/LOT_hydro_map.dump -U
awips
dx1-lot$ pg_dump -Fc -t 'mapdata.i*mile*' maps -f /data/fxa/LOT_i_mile_map.dump -U
awips
dx1-lot$ pg_dump -Fc -t 'mapdata.i*mm*' maps -f /data/fxa/LOT_i_mm_map.dump -U awips
dx1-lot$ pg_dump -Fc -t 'mapdata.interstatecwa*' maps -f
/data/fxa/LOT_interstatecwa_map.dump -U awips
dx1-lot$ pg_dump -Fc -t 'mapdata.int' -t 'mapdata.interstates' maps -f
/data/fxa/LOT_interstates_map.dump -U awips
dx1-lot$ pg_dump -Fc -t 'mapdata.key*' maps -f /data/fxa/LOT_key_map.dump -U awips
dx1-lot$ pg_dump -Fc -t 'mapdata.major*' maps -f /data/fxa/LOT_major_map.dump -U
awips
dx1-lot$ pg_dump -Fc -t 'mapdata.oemczones*' maps -f /data/fxa/LOT_oemczones_map.dump
-U awips
dx1-lot$ pg_dump -Fc -t 'mapdata.ord*' maps -f /data/fxa/LOT_ord_map.dump -U awips
dx1-lot$ pg_dump -Fc -t 'mapdata.roc_wind*' maps -f /data/fxa/LOT_roc_wind_map.dump -
U awips
dx1-lot$ pg_dump -Fc -t 'mapdata.state*' maps -f /data/fxa/LOT_state_map.dump -U
awips
dx1-lot$ pg_dump -Fc -t 'mapdata.tafbuffer*' maps -f /data/fxa/LOT_tafbuffer_map.dump
-U awips
dx1-lot$ pg_dump -Fc -t 'mapdata.warngen_dss*' maps -f
/data/fxa/LOT_warngen_dss_map.dump -U awips
dx1-lot$ pg_dump -Fc -t 'mapdata.warngenloc*' maps -f
/data/fxa/LOT_warngenloc_map.dump -U awips
dx1-lot$ pg_dump -Fc -t 'mapdata.webcities*' maps -f /data/fxa/LOT_webcities_map.dump
-U awips
dx1-lot$ pg_dump -Fc -t 'mapdata.wind*' maps -f /data/fxa/LOT_windfarms_map.dump -U
awips
dx1-lot$ pg_dump -Fc -t 'mapdata.zau*' maps -f /data/fxa/LOT_zau_map.dump -U awips
```

Using psql to obtain the metadata (.CSV) files:

```
dx1-lot$ psql maps -U awips
```

```
psql (9.2.4)
```

```
Type "help" for help.
```

```
maps=# set search_path=mapdata;
```

```
SET
```

```
maps=# copy (select * from mapdata.map_version where table_name similar to  
'airport[dhm]%) to '/data/fxa/LOT_airportstuff_map.csv' CSV HEADER;
```

NOTE: The wildcard for table names in SQL is a percent sign (%) rather than an asterisk like with the pg\_dump commands above.

```
COPY 3
```

```
maps=# copy (select * from mapdata.map_version where table_name similar to  
'arrival%') to '/data/fxa/LOT_arrival_map.csv' CSV HEADER;
```

```
COPY 1
```

```
maps=# copy (select * from mapdata.map_version where table_name like 'c90%') to  
'/data/fxa/LOT_c90_boundary_map.csv' CSV HEADER;
```

```
COPY 1
```

```
maps=# copy (select * from mapdata.map_version where table_name like 'chi%') to  
'/data/fxa/LOT_chicago_key_locations_map.csv' CSV HEADER;
```

```
COPY 1
```

```
maps=# copy (select * from mapdata.map_version where table_name like 'cwsu%') to  
'/data/fxa/LOT_cwsu_map.csv' CSV HEADER;
```

```
COPY 1
```

```
maps=# copy (select * from mapdata.map_version where table_name like 'dep%') to  
'/data/fxa/LOT_departure_map.csv' CSV HEADER;
```

```
COPY 1
```

```
maps=# copy (select * from mapdata.map_version where table_name similar to 'ffmp%')  
to '/data/fxa/LOT_ffmp_map.csv' CSV HEADER;
```

```
COPY 2
```

```
maps=# copy (select * from mapdata.map_version where table_name similar to 'golf%')  
to '/data/fxa/LOT_golf_map.csv' CSV HEADER;
```

COPY 1

```
maps=# copy (select * from mapdata.map_version where table_name similar to 'hydro%')  
to '/data/fxa/LOT_hydro_map.csv' CSV HEADER;
```

COPY 2

```
maps=# copy (select * from mapdata.map_version where table_name similar to 'i%mile%')  
to '/data/fxa/LOT_i_mile_map.csv' CSV HEADER;
```

COPY 12

```
maps=# copy (select * from mapdata.map_version where table_name similar to 'i%mm%')  
to '/data/fxa/LOT_i_mm_map.csv' CSV HEADER;
```

COPY 15

```
maps=# copy (select * from mapdata.map_version where table_name similar to  
'interstatecwa') to '/data/fxa/LOT_interstatecwa_map.csv' CSV HEADER;
```

COPY 1

```
maps=# copy (select * from mapdata.map_version where table_name = 'int' or table_name  
= 'interstates') to '/data/fxa/LOT_interstates_map.csv' CSV HEADER;
```

COPY 2

```
maps=# copy (select * from mapdata.map_version where table_name like 'key%') to  
'/data/fxa/LOT_key_map.csv' CSV HEADER;
```

COPY 1

```
maps=# copy (select * from mapdata.map_version where table_name like 'major%') to  
'/data/fxa/LOT_major_map.csv' CSV HEADER;
```

COPY 3

```
maps=# copy (select * from mapdata.map_version where table_name like 'oemczones%') to  
'/data/fxa/LOT_oemczones_map.csv' CSV HEADER;
```

COPY 1

```
maps=# copy (select * from mapdata.map_version where table_name like 'ord%') to  
'/data/fxa/LOT_ord_map.csv' CSV HEADER;
```

COPY 2

```
maps=# copy (select * from mapdata.map_version where table_name like 'roc_wind%') to  
'/data/fxa/LOT_roc_wind_map.csv' CSV HEADER;
```

COPY 1



```
maps=# copy (select * from mapdata.map_version where table_name like 'state%') to
'/data/fxa/LOT_state_map.csv' CSV HEADER;
```

COPY 3

```
maps=# copy (select * from mapdata.map_version where table_name like 'tafbuffer%') to
'/data/fxa/LOT_tafbbuffer_map.csv' CSV HEADER;
```

COPY 1

```
maps=# copy (select * from mapdata.map_version where table_name like 'warngen_dss%')
to '/data/fxa/LOT_warngen_dss_map.csv' CSV HEADER;
```

COPY 1

```
maps=# copy (select * from mapdata.map_version where table_name like 'warngenloc%')
to '/data/fxa/LOT_warngenloc_map.csv' CSV HEADER;
```

COPY 1

```
maps=# copy (select * from mapdata.map_version where table_name like 'webcities%') to
'/data/fxa/LOT_webcities_map.csv' CSV HEADER;
```

COPY 1

```
maps=# copy (select * from mapdata.map_version where table_name like 'wind%') to
'/data/fxa/LOT_windfarms_map.csv' CSV HEADER;
```

COPY 2

```
maps=# copy (select * from mapdata.map_version where table_name like 'zau%') to
'/data/fxa/LOT_zau_map.csv' CSV HEADER;
```

COPY 2

```
maps=# \q
```

```
dx1-lot$
```

### Step 3. Transfer the resulting files to the WES-2 Bridge machine.

In this situation, we will place these files in the central localization store on the wes2 machine. If you were building a case to send to another office, then you would place these map files in the localization with the case. We will first ensure that the maps directory exists in the central localization, make the directory if it doesn't exist, and then transfer the files made above to that location.

```
wes2-lot$ cd /awips2/edex/data/utility
```

```
wes2-lot$ ls
```

```
cave_config  cave_static  common_static  edex_static
```

```
wes2-lot$ mkdir maps (only need to do this if the directory does not already exist)
```

```
wes2-lot$ cd /data/fxa
```

```
wes2-lot$ cp LOT_*map.dump /awips2/edex/data/utility/maps
```

```
wes2-lot$ cp LOT_*map.csv /awips2/edex/data/utility/maps
```

The WES-2 Bridge software will recognize the presence of these map files and automatically load them into the appropriate map databases when needed for a particular EDEX instance.