**Importing Map Shapefiles into AWIPS2 using importShapeFile[¶](https://collaborate.nws.noaa.gov/trac/siteconfig/wiki/HowToImportShapeFile" \l "ImportingMapShapefilesintoAWIPS2usingimportShapeFile" \o "Link to this section)**

**5/26/11 version 2.6**

**Overview**

The following is an overview of the steps required to import map shapefiles into AWIPS2, making them accessible from CAVE:

1. Copy the .shp, .shx and .dbf files for the map into the edex\_static/site/LLL/ shapefile directory
2. Run importShapeFile.sh
3. Create a new bundles/maps file
4. Restart CAVE

The shapefile import occurs on the Database server (dx1) and the bundle file creation occurs on the CAVE workstation. On the ADAM workstation, all of the following tasks are completed on the ADAM workstation.

The following example, using a shapefile called **localtowns**, provides more detailed steps for adding a map in AWIPS2.

**THIS SECTION FOR ADAM USERS ONLY**

In order to import shapefiles into the postGIS database, access control entries must be modified on the ADAM workstation to allow the script to complete.

1.) As user root, edit the local entry for the maps database and the host entry for the maps database to allow trusted access

su - root

cd /awips2/data

vi pg\_hba.conf

Modify the following two entries (near the top and bottom of the file):

local maps all mdf5

host maps all 127.0.0.1/xx mdf5

Replace **mdf5** with **trust** in both entries (as shown below) and save the file.

local maps all trust

host maps all 127.0.0.1/xx trust

2.) Reload the database as user awips

su - awips

pg\_ctl reload -D /awips2/data

Proceed with the following instructions

**END ADAM ONLY SECTION**

**Import Shapefile Procedure**

**Note:** Perform the following steps on the Database server (dx1) as user “awips”.

1. Create a shapefiles directory in the …/edex\_static/site/LLL/ directory. LLL is the site id (OAX).

mkdir /awips2/edex/data/utility/edex\_static/site/OAX/shapefiles

1. In the shapefiles directory created in Step 1, create a directory with the name of the desired database table name.

mkdir /awips2/edex/data/utility/edex\_static/site/OAX/shapefiles/localtowns

**Note:** Normally this directory is named after the shapefile name, but you need to ensure the table name is unique. The “AWIPS II Site Data Configuration: Step-By-Step Guide” contains a list of the AWIPS shapefiles (Table 2.3-1).

1. Copy the shapefiles into the directory created in Step 2. The import process requires at least the .shp, .shx and .dbf shapefiles be present in the directory. The files in the “…/shapefiles/localtowns” directory are as follows:
2. localtowns.shp
3. localtowns.shx
4. localtowns.dbf

**Note:** We are working on a script to import multiple shapefiles and to determine the type of shapefile. So the geometry simplification arguments will not be required on the command line. There is still time for comments for importing shapefiles. TODO: add steps for script to import multiple shapefiles.

1. To import one shapefile at a time, execute the importShapeFile.sh script with the following arguments: shapefile, scheme (mapdata), table (shapefile name), geometry simplification levels (not used for Point data), user, port, and awips root directory. Executing the command “importShapeFile.sh” without any arguments displays a usage statement on the screen, and provides a brief description of the arguments.

cd /awips2/static/database.maps/

The following are possible geometry simplification level settings represented by ${SIMPLEV} in the importShapeFiles command below and should be used as shown at this time (Find more information on parameters at: [ImportShapeFileParameters](https://collaborate.nws.noaa.gov/trac/siteconfig/wiki/ImportShapeFileParameters) ):

• Point shapefiles

export SIMPLEV=””

• Arc (Line) or Polygon shapefiles

export SIMPLEV=“0.064, 0.016, 0.004, 0.001”

As user “awips” execute the following command:

./importShapeFile.sh

/awips2/edex/data/utility/edex\_static/site/OAX/shapefiles/localtowns/localtowns.shp

mapdata localtowns “${SIMPLEV}” awips 5432 /awips2

1. Check the output to the terminal window to verify the import script ran without errors. Also, note the type of the shapefile (Point, Arc (Line), or Polygon), since you will need to know this for setting up CAVE. The following shows a screen shot of importing a shapefile containing point data. Following this link [ImportLineAndPolygonShapefiles](https://collaborate.nws.noaa.gov/trac/siteconfig/wiki/ImportLineAndPolygonShapefiles) shows examples of importing shapefiles containing Line (ARC) and Polygon data.

**Screen shot of terminal window depicting a successful import run of Point data**

[awips@localhost maps]$ cd /awips2/static/database.maps/

[awips@localhost database.maps]$ SIMPLEV=""

[awips@localhost database.maps]$ ./importShapeFile.sh

/awips2/edex/data/utility/edex\_static/site/OAX/shapefiles/localtowns/localtowns.shp

mapdata localtowns "${SIMPLEV}" awips 5432 /awips2

Importing localtowns.shp into mapdata.localtowns ...

NOTICE: table "localtowns" does not exist, skipping

Shapefile type: Point

Postgis type: POINT[2]

psql:<stdin>:13: NOTICE:

CREATE TABLE will create implicit sequence "localtowns\_gid\_seq" for serial column "localtowns.gid"

psql:<stdin>:13: NOTICE:

CREATE TABLE / PRIMARY KEY will create implicit index "localtowns\_pkey" for table "localtowns"

addgeometrycolumn

---------------------------------------------------------

mapdata.localtowns.the\_geom SRID:4326 TYPE:POINT DIMS:2

(1 row)

[awips@localhost database.maps]$

1. [Optional] Using pgadmin3 or psql, verify the table was created in the mapdata scheme in the maps database. The table name is the name of the directory you created in Step 2, usually the shapefile name. For the example, there should be a localtowns table in the mapdata scheme of the maps database in postgres. Using psql you can execute the following commands as user “awips” to verify that the table was created:
2. psql maps
3. \d mapdata.localtowns
4. \q

**Create Bundle File Procedure**

**Note:** Perform the following steps on a workstation.

1. On a workstation, create a new map bundle file to define the menu item in the Map menu in CAVE. Change to the …/cave/etc/bundles/maps directory.
2. cd /awips2/cave/etc/bundles/maps
3. As user 'awips' create the site path for local map menus if it does not already exist.
4. su - awips
5. mkdir /awips2/edex/data/utility/cave\_static/site/XXX/bundles/maps (where XXX is your 3 letter site ID)
6. Use the CWAs.xml file as a template to create a new file for the imported map. Copy the CWAs.xml file to a file named from the shapefile (shapefile\_name.xml).

The filename does not have to be an exact match to the shapefile or table name, as that is not a requirement. Using a name similar to the shapefile name may help to associate the shapefile with the map bundle file. Since many of the files in the bundles/maps directory following the camel naming format, I am using this in the example.

cp CWAs.xml /awips2/edex/data/utility/cave\_static/site/XXX/bundles/maps/localTowns.xml

1. Edit the new xml map bundle file for the menu item to add to the Maps menu in CAVE. The steps are different for Point, or Line and Polygon shapefiles:

**Line and Polygon shapefiles** - in the resourceData element, change the value of the table element to the name of the database table from Step 6, and the mapName to the label to show in the Map menu in CAVE. If labels are required in the new map, then edit the labelField in the Capability element. Otherwise, delete the capability element with type “labelableCapability”. The value for the labelField relates to a column in the database table with the same name. For example, the value CWA for the labelField refers to the values in the cwa column of the CWA table in the mapdata scheme.

**Point shapefiles** - in the resourceData element, change the type attribute to “dbPointMapResourceData”, change the value of the table element to the name of the new database table name from Step 6, and the mapName to the label to show in the Map menu in CAVE. If labels are required in the new map, then edit the labelField in the Capability element. Otherwise, delete the capability element with type “labelableCapability”. The value for the labelField relates to a column in the database table with the same name. For example, the value CWA for the labelField refers to the values in the cwa column of the CWA table in the mapdata scheme.

For more information on configuring the resource data options, the [MapResourceDataConfigOptions](https://collaborate.nws.noaa.gov/trac/siteconfig/wiki/MapResourceDataConfigOptions) page provides more details on configuring the capabilities and resources for a map. The following screen shot shows the example map bundle file for localTowns:

**Screen shot of localTowns.xml map bundle file**

<bundle>

<displayList>

<displays xsi:type="mapRenderableDisplay" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<descriptor xsi:type="mapDescriptor">

<resource>

<loadProperties>

<capabilities>

<capability xsi:type="outlineCapability" lineStyle="SOLID" outlineOn="true"

outlineWidth="1" />

<capability xsi:type="colorableCapability" colorAsString="#9b9b9b" />

<capability xsi:type="labelableCapability" labelField="name"/>

</capabilities>

<resourceType>PLAN\_VIEW</resourceType>

</loadProperties>

<properties isSystemResource="false" isBlinking="false" isMapLayer="true"

isHoverOn="false" isVisible="true">

<pdProps maxDisplayWidth="100000000" minDisplayWidth="0" />

</properties>

<resourceData xsi:type="dbPointMapResourceData">

<table>mapdata.localtowns</table>

<mapName>Local Towns</mapName>

</resourceData>

</resource>

</descriptor>

</displays>

</displayList>

</bundle>

1. Restart CAVE to load the changes for the new map bundle file.
2. Verify the new map appears in the **Maps** menu in CAVE, and by selecting the new map from the **Maps** menu the appropriate map loads into the perspective.
3. Copy the map bundle file to all workstations requiring the imported maps.