PyramidMap Python Visualized Toolset Technical White Paper

CTRL + mouse wheel can control the zoom of the content page to achieve the best reading effect.

PyramidMap Python Visualized Toolset Technical White Paper
1: Introduction
1.1: Function overview
1.2: Function list
1.3: Contact us
1.4: Software deployment
2: Operation mode
3: Function list
4: System settings
4.1: Create GIS Server connection
4.2: Create GIS Database and Connection
4.2.1: Create geodatabase in Oracle
4.2.2: Create geodatabase in SQL server
4.2.3: Create geodatabase in PostgreSQL
4.3: Create GIS Database Connections
4.3.1: Create Oracle connection
4.3.2: Create SQL Server connection
4.3.3: Create PostgreSQL connection
4.4: GIS database application
5: Display map
6: Feature Class process
6.1: Data to feature class
6.2: Feature class to data
6.3: Append and merge features
6.3.1: Features append

6.3.2: Features merge
6.4: Feature classes migration
6.4.1: Feature class migration model
6.4.2: Feature class migration module
6.5: Standard coordinate system conversion
7: Raster processing
7.1: Raster compression
7.2: Raster tile
7.3: Raster merge
7.4: Raster conversion
7.5: Raster coordinate system transformation
7.6: Raster noData processing
7.7: Raster pyramid building
7.8: Creates raster mosaic dataset
7.9: Publish image server
8: Create mxd workflow
8.1: Vector layers mxd
8.1.1: Work model
8.1.2: Operation interface
8.2: Raster layers mxd
9: Publish map servers
9.1: Publish MapServer
9.1.1: Workflow model
9.1.2: Operation interface
9.2: Publish FeatureServer
9.2.1: Workflow model
9.2.2: Operation interface
10: GIS database
10.1: Layers List
10.2: Layer Preview
10.3: Layer Export
11: GIS server
11.1: Layers List
11.2: Layer Preview
11.3: Layer Export
12: Log query
12.1: Log category directory
12.2: Log query

1: Introduction

1.1: Function overview

The PyramidMap Python is a visualized map toolset based on Arcpy and GDAL. The input and output parameters of all tools are configured and selected through the visualization operation interface, helping you easily complete complex map processing tasks. The software adopts client / server mode, integrates GDAL, arcpy and OSR instruction sets internally, and provides a complete process of vector and image data processing, as well as a visual interface guidance system designed for this purpose. The platform supports compatibility and conversion of multi format heterogeneous map data sources such as GDB / MDB / SHP local file type, DBMS geographic database type and remote map server type, including full process processing such as feature type conversion, coordinate system conversion, map database import and export, map server

publishing, map layer segmentation, merging, compression, pixel depth conversion and pyramid construction. The PyramidMap Python visualization toolset is only supported running in a Windows environment currently , with a minimum resolution requirement of 1920 * 1200.

1.2: Function list

The PyramidMap 's functions list shown as in the table 1-1

Functions	Sub item	instruction
System settings	Manage GIS server connection	Create and save GIS server connections for publishing services.
	Manage Geodatabase	Establish or enable an Enterprise Geodatabase and save the sde connections.
	Manage Geodatabase connection	Connecting Enterprise Geodatabase and save the sde connections.
Functions	Querying and displaying on map	Querying and displaying on vector or raster layers
	Vector data toolset	Performing basic vector class map feature class management.
	Raster data toolset	Performing raster data management and processing. Processing raster dataset attributes, creating and processing raster data.
	Cartography toolset	Create and optimize map data. This includes creating features, annotations, symbols, creation of grids and graticules, and the managing data-driven pages for layout.
	General analyst and statistics toolset	Provide display, analysis, and statistical tools based on spatial continuous surfaces or maps.
	Spatial analyst and statistical toolset	Provide spatial analysis, statistics and modeling tools for vector feature data and raster data.
	Geometric network toolset	Create, manage, and run tracking tasks on spatial geometric networks, and perform various network analyses. For example, water distribution, transmission lines, gas pipelines, communication pipelines, and hydrological rivers can all be modeled and analyzed using geometric networks for resource discovery.
	Geocoding and Network Analyst toolset	Geocoding determines an accurate spatial location for features through the address descriptor, and on the contrary, it can accurately describe the spatial location. Network analysis tools can maintain traffic network models and perform network analysis on paths, nearest facility points, service areas, start destination cost matrices, vehicle delivery (VRP), and location allocation.
	Linear reference and tracking analyst	Linear reference tools can collect data on linear features such as highways, urban streets, railways, rivers, pipelines, water supply networks, and sewer networks, and perform multimodal association through attribute data and spatial location for display, query, editing, and analysis. By associating with temporal data, analyze and calculate the continuously arranged elements in the trajectory.
	3D Analyst toolset	Implement various analysis, data management, and data conversion operations on surface models and 3D vector data.
	Geodatabase toolset	Manage various geodatabases. Including data management tools for file type and relational enterprise geodatabases, as well as upgrading, managing, and maintaining on the theme.
	ArcGIS Server toolset	A tool for managing ArcGIS Server maps and global caching. Contains tools for simplifying the process of extracting data through servers and simplifying web application printing.
	Quering in operating log	Figure 1-1: The classification and catalog structure of the visualized toolset

Table 1-1: The functions list of the visualized Toolset

The catalog structure of the extension toolset shown as in the following Figure 1-1

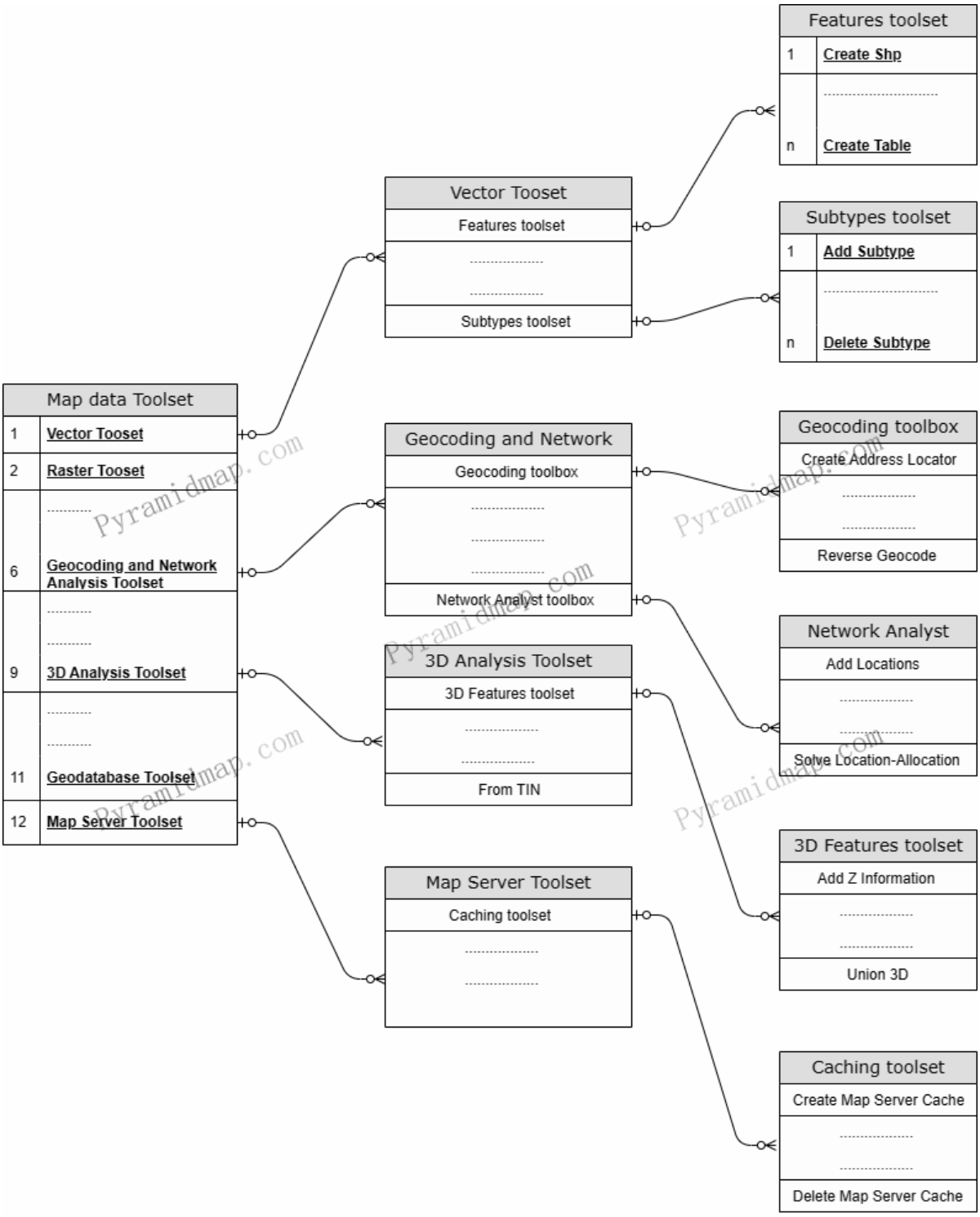


Figure 1-1: The classification and catalog structure of the visualized toolset

The three level navigation treeview of the toolset as belowe

- **1.Vector Dataset toolset**
- **2.Raster Dataset toolset**
- **3.Cartography toolset**
- **4.General Analyst and Statistics toolset**
- **5.Spatial Analyst and Statistical toolset**
- **6.Geometric Network toolset**
- **7.Geocoding and Network Analyst toolset**
- **8.Linear Reference and Tracking Analyst toolset**
- **9.3D Analyst toolset**
- **10.Geodatabase toolset**
- **11.Server toolbox**

The PyramidMap Python Visualized Toolset can help you quickly achieve map data processing. Using these tools requires separate authorization. Please contact the PyramidMap studio studio.

1.3: Contact us

PyramidMap focuses on GIS application services and is committed to building a complete ecosystem of efficient GIS data processing, enterprise level geographic data storage, distributed map services, web and mobile map applications. It can help you achieve map data processing more easily and quickly, until the terminal displays the entire process of the application as a streamlined service. Welcome visiting to: http://www.pyramidmap.com obtaining beta software and technical information. For more technical support, please contact PyramidMap studio.

Email: service@pyramidmap.com

Tele: (086)0535-82957588

Mobile: (086)18660789051

WeChat: A18660789051

QQ: 29862351

1.4: Software deployment

1: The PyramidMap Python Visualized Toolset only runs in a Windows environment currently.

2:The PyramidMap Python Visualized Toolset application is distributed in GZ or zip, and can be run after decompression,no setup or install is required.

3: Running mode: Locate the main.exe program under your deployment path, and double-click to run it. Or in the cmd window, switch to the deployment path through the cd command, enter the main program name, and press Enter to start, as shown in Figure 1-2.

Figure 1-2: The main program start running successful

After successful startup, enter the main screen of the program, as shown in Figure 1-3.

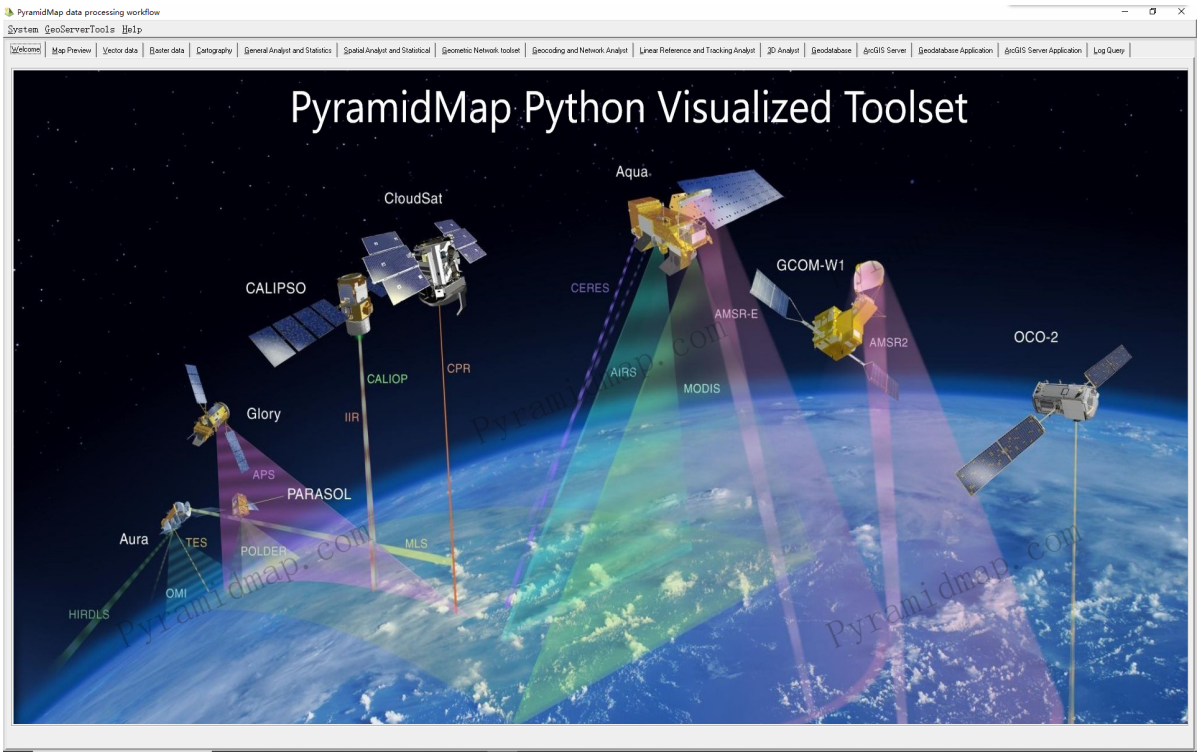


Figure 1-3: Program main screen

4: Environment dependency: The target map server for PyramidMap is an ArcGIS server. In addition to publishing map services, some map processing is also run based on the ArcGIS py environment. Therefore, the target computer is required to have an ArcGIS environment, which is a prerequisite for running PyramidMap.

5: When PyramidMap is running, make sure the ArcGIS License Server Administrator on the target computer is started. It also has corresponding function license, as shown in Figure 1-4:

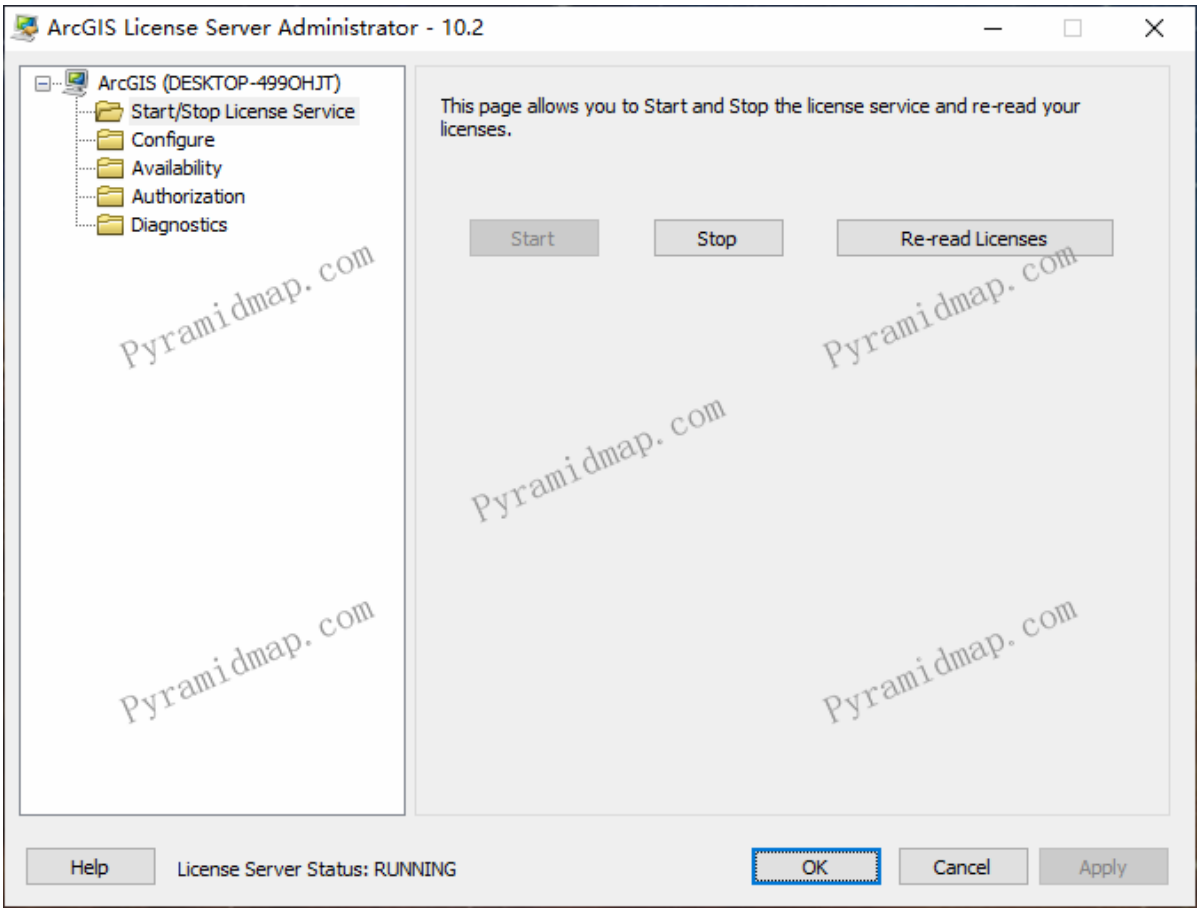


Figure 1-4: The ArcGIS license server administrator on the target computer is started

6:For the operation of geographic database (Oracle / SQL Server / PostgreSQL / MySQL), the target computer needs to have the client of the corresponding database and be correctly configured to establish a connection with the corresponding database. To publish the feature service, the ArcGIS Server also requires the client of the correspondding database to be configured correctly and be able to establish a connection with the corresponding database. It should be noted that the ArcGIS desktop is 32-bit and needs 32-bit database client, while the ArcGIS Server is 64 bit and needs 64 bit database client. About how to connection between ArcGIS anddatabase, you can refer to ArcGIS help document or online document.

2: Operation mode

Start PyramidMap python in Windows command line mode, and support GUI interface operation and command line output at the same time. The software adopts the client/server mode and integrates GDAL, ArcPy and OSR instruction sets internally, providing a complete process of vector and image data processing, as well as a visual interface-oriented system designed for this purpose. The platform supports the compatibility and conversion of multi-format heterogeneous map data sources such as Gdb/Mdb/SHp local file type, DBMS geographic database type, remote map server type, etc., including: feature type conversion, coordinate system conversion, map database import and export, map server publishing, map layer segmentation, merging, compression, pixel depth conversion, pyramid construction and other full-process processing.

3: Function list

The functions list of the PyramidMap Python Visualized Toolset shown as in the table 3-1

Functions	Sub item	instruction
System settings	Manage GIS server connection	Create and save GIS server connections for publishing services.
	Manage Geodatabase	Establish or enable an Enterprise Geodatabase and save the sde connections.
	Manage Geodatabase connection	Connecting Enterprise Geodatabase and save the sde connections.
Functions	Querying and displaying on map	Querying and displaying on vector or raster layers
	Vector data toolset	Performing basic vector class map feature class management.
	Raster data toolset	Performing raster data management and processing. Processing raster dataset attributes, creating and processing raster data.
	Cartography toolset	Create and optimize map data. This includes creating features, annotations, symbols, creation of grids and graticules, and the managing data-driven pages for layout.
	General analyst and statistics toolset	Provide display, analysis, and statistical tools based on spatial continuous surfaces or maps.
	Spatial analyst and statistical toolset	Provide spatial analysis, statistics and modeling tools for vector feature data and raster data.
	Geometric network toolset	Create, manage, and run tracking tasks on spatial geometric networks, and perform various network analyses. For example, water distribution, transmission lines, gas pipelines, communication pipelines, and hydrological rivers can all be modeled and analyzed using geometric networks for resource discovery.
	Geocoding and Network Analyst toolset	Geocoding determines an accurate spatial location for features through the address descriptor, and on the contrary, it can accurately describe the spatial location. Network analysis tools can maintain traffic network models and perform network analysis on paths, nearest facility points, service areas, start destination cost matrices, vehicle delivery (VRP), and location allocation.
	Linear reference and tracking analyst	Linear reference tools can collect data on linear features such as highways, urban streets, railways, rivers, pipelines, water supply networks, and sewer networks, and perform multimodal association through attribute data and spatial location for display, query, editing, and analysis. By associating with temporal data, analyze and calculate the continuously arranged elements in the trajectory.

Functions	Sub item	instruction
	3D Analyst toolset	Implement various analysis, data management, and data conversion operations on surface models and 3D vector data.
	Geodatabase toolset	Manage various geodatabases. Including data management tools for file type and relational enterprise geodatabases, as well as upgrading, managing, and maintaining on the theme.
	ArcGIS Server toolset	A tool for managing ArcGIS Server maps and global caching. Contains tools for simplifying the process of extracting data through servers and simplifying web application printing.
	Quering in operating log	Log recording and classification of process steps, as well as querying and tracing in different time periods.

Table 3-1: The functions list of the visualized toolset

4: System settings

4.1: Create GIS Server connection

PyramidMap can be used as an ArcGIS Server client, mainly to build automated processes in a workflow mode and publish map services to ArcGIS Server. PyramidMap creates an ArcGIS Server connection as shown in Figure 4-1.

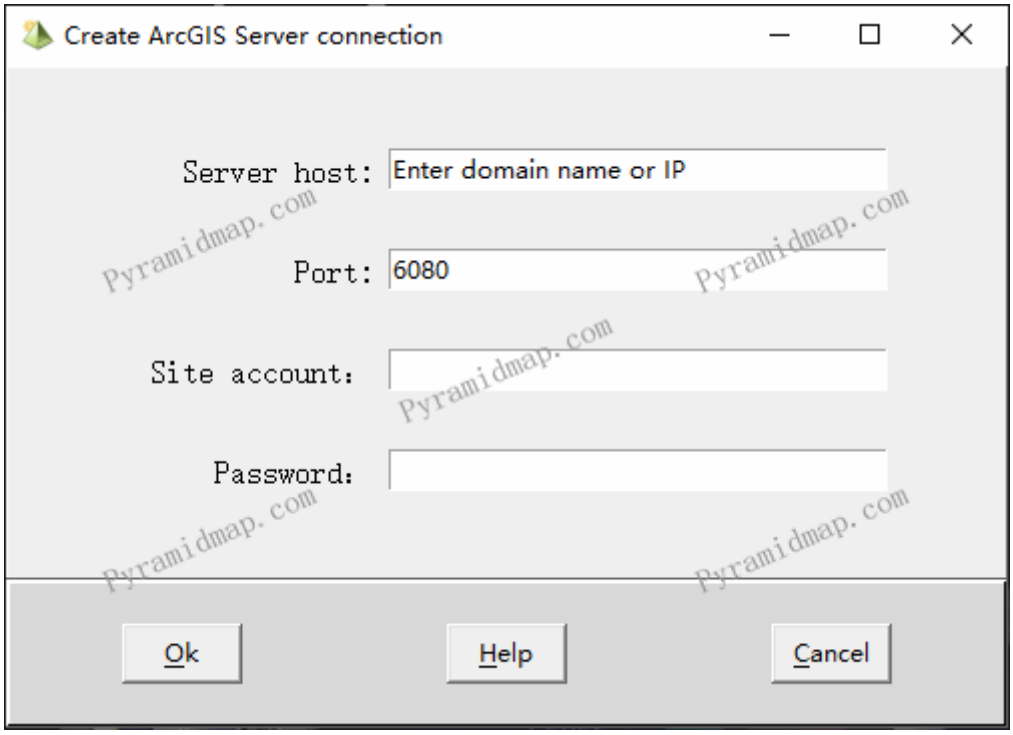


Figure 4-1: PyramidMap Create ArcGIS Server Connection

ArcGIS Server Address=ArcGIS Server IP address. The default port is 6080. If there is any change, the actual situation shall prevail. The administrator account is the administrator account of the ArcGIS Server site. Please contact the ArcGIS Server administrator to confirm. PyramidMap provides cmd output monitoring of the running process, as shown in Figure 4-2.

Figure 4-2: PyramidMap provides cmd output monitoring for creating ArcGIS Server connections

The creation results of ArcGIS Server connection are shown in Figure 4-3.

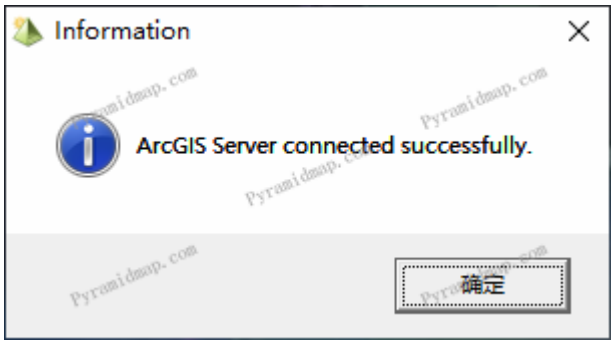


Figure 4-3: The result of creating ArcGIS Server Connection

After the ArcGIS Server connection created successfully, enter the PyramidMap's ArcGIS Server Management module to manage and preview the list of layers, as shown in Figure 4-4.

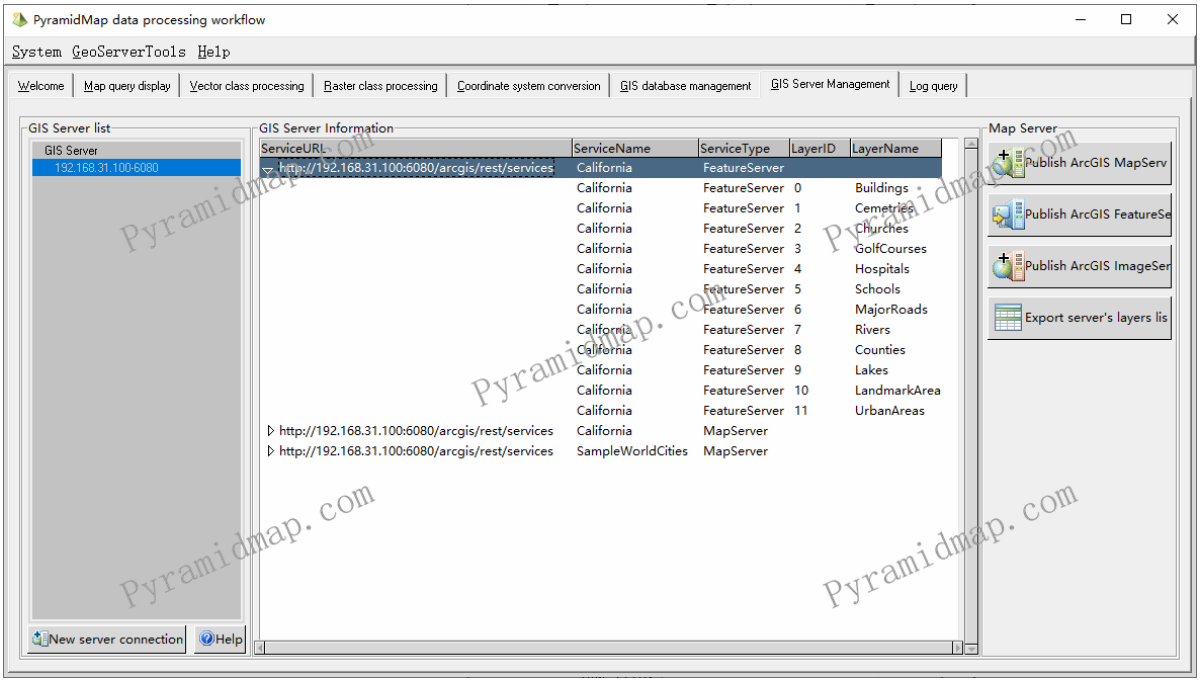


Figure 4-4: PyramidMap obtains the ArcGIS Server layer list

In the GIS server management module, we can preview and export the internal layer data, as shown in Figure 4-5.

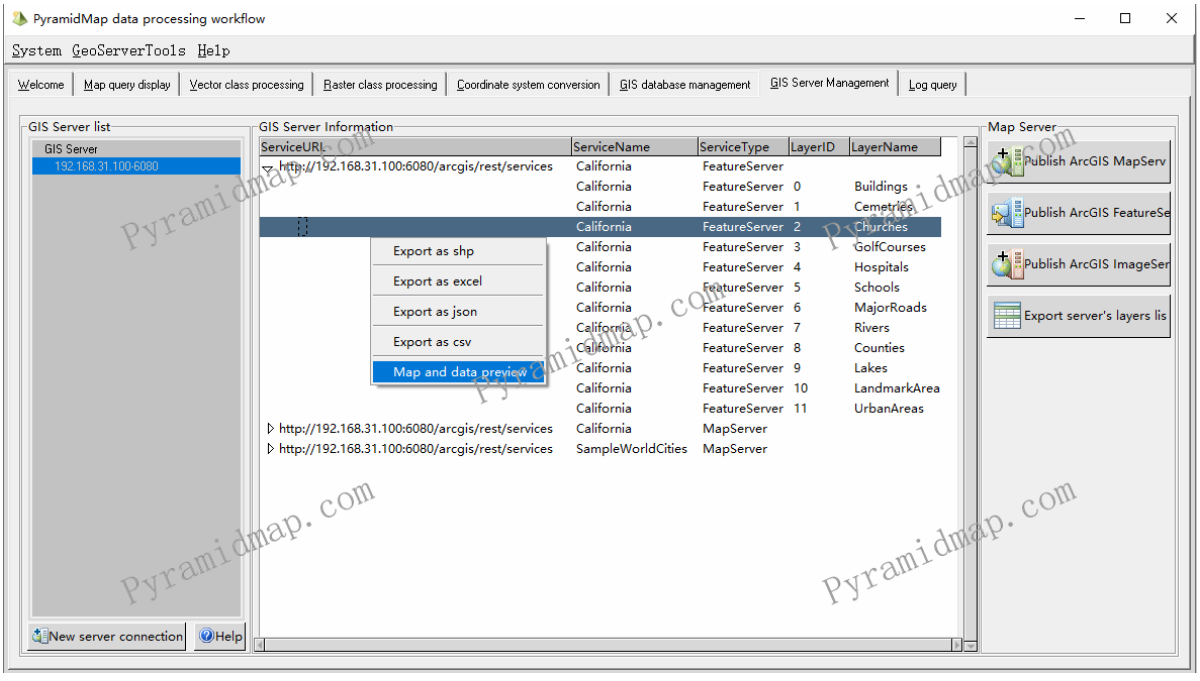


Figure 4-5: PyramidMap for ArcGIS Server Layer Preview and Export Conversion

In this example, we use the FeatureLayer type layer Churches in an ArcGIS Server to demonstrate the distribution of churches in Mississippi, as shown in Figures 4-6.

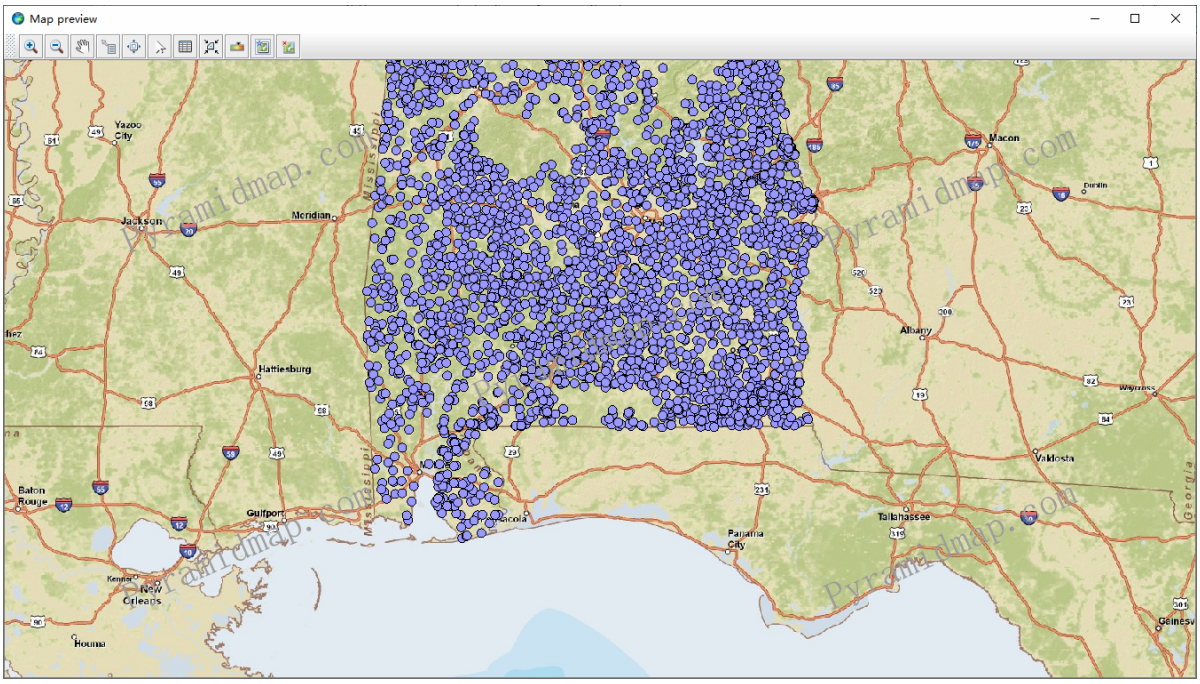


Figure 4-6: PyramidMap preview of FeatureLayer type layer Churches in ArcGIS Server

In summary, ArcGIS Server is responsible for storing, editing, querying maps, as well as outputting and managing map service interfaces. It provides service interfaces for map visualization applications on the web and mobile ends, and is widely used. We will continue to discuss and learn more about GIS servers in the following subjects [9: Publish map servers] and [11: GIS Server].

4.2: Create GIS Database and Connection

ArcGIS defines geographical database in two ways: file type geographical database and relational geographical database. The former includes Gdb/Mdb, and the latter covers Oracle, SQL Server, PostgreSQL, DB2, Infomix and other multi type DBMS relational database. In short, geographical database is based on ordinary database, which gives spatial geometry data storage and processing capabilities and permissions, so that it has the ability to store and process map data, and can render, display and edit on the client side. PyramidMap provides adaptation support on this basis. Before performing this operation, please confirm that the client program connected to the above database is already available on the computer. The database client can be downloaded from the database website, please note the corresponding version number. ArcGIS Desktop is a 32-bit program, so it only supports 32-bit database client connections. ArcGIS Server is a 64-bit program, and only supports 64-bit database client programs.

4.2.1: Create geodatabase in Oracle

PyramidMap creates a geo database in Oracle, as shown in Figure 4-7.

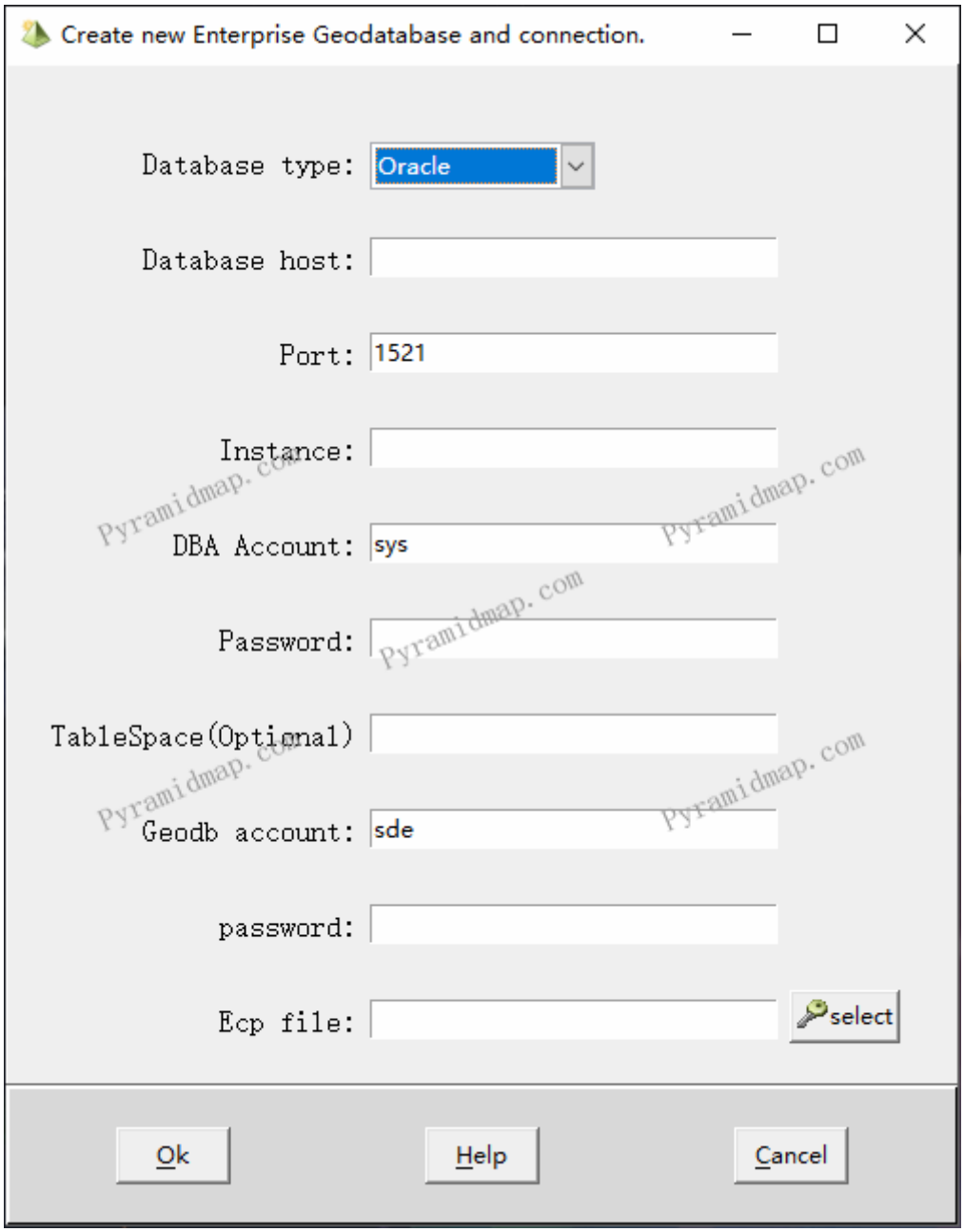


Figure 4-7: PyramidMap creates a geo database in Oracle

The service instance is the database name. It is the name of the database that needs to be endowed with spatial data processing capabilities. This process will be based on the existing Oracle Database and authorized to upgrade to geographical database. PyramidMap provides cmd output monitoring of the running process, as shown in Figure 4-8.

```
E:\ProjectDevelop\python\arcgis_en>python main.py
PyramidMap has been started and in running >>>>>>
++++++++Creating enterprise Geodatabase.++++++++
Installing St_Geometry ....
Successfully installed St_Geometry.

Executing: CreateEnterpriseGeodatabase Oracle 192.168.31.100:1521/geodata # DATABASE_AUTH sys ****** SDE_SCHEMA
sde ***** # "D:/ArcGIS Server 10.2/arcgis server 10.2.ecp"
Start Time: Tue Apr 04 11:00:53 2023
User has privileges required to create database objects.
Tablespace created.
Geodatabase admin user created.
Validated authorization file.
User has required privileges for geodatabase setup.
XML support is enabled for the database instance.
Created geodatabase tables and stored procedures.
Finished creating geodatabase schema.
WARNING: The ST_Geometry shape library path may be invalid, or the EXTPROC is not configured correctly in the O
racle instance. [ (Unable to determine current version of ST_SHAPELIB. Please check the ST_Geometry shape libra
ry path on the Oracle server, which is set to "c:\progra~2\arcgis\desktop10.2\DatabaseSupport\Oracle\Windows64\
st_shapelib.dll". Refer to the ArcGIS help topics for more details.
)]
Succeeded at Tue Apr 04 11:01:21 2023 (Elapsed Time: 28.07 seconds)
+++++++++

++++++++INFO: The establishment of enterprise geographic database is successful.++++++++
++++++++INFO: The enterprise geographic database connection file is created and saved successfully.++++++++
```

Figure 4-8: PyramidMap provides cmd output monitoring for the creating geo database

View the newly created geographical database in ArcMap, as shown in Figure 4-9.

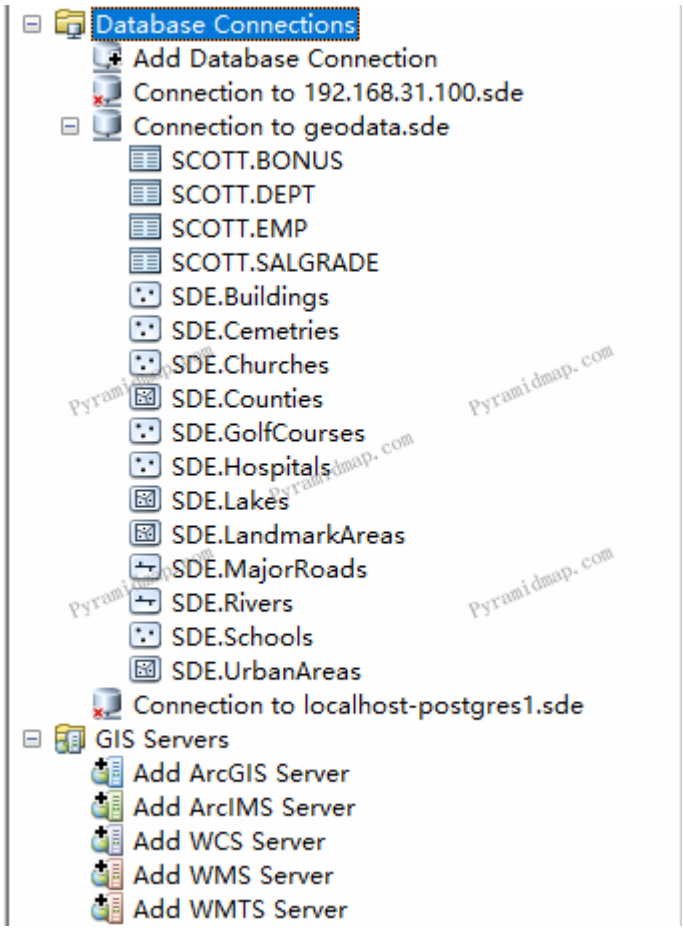


Figure 4-9: View the newly created geographical database in ArcMap

View the newly created geographical database in PyramidMap, as shown in Figure 4-10.

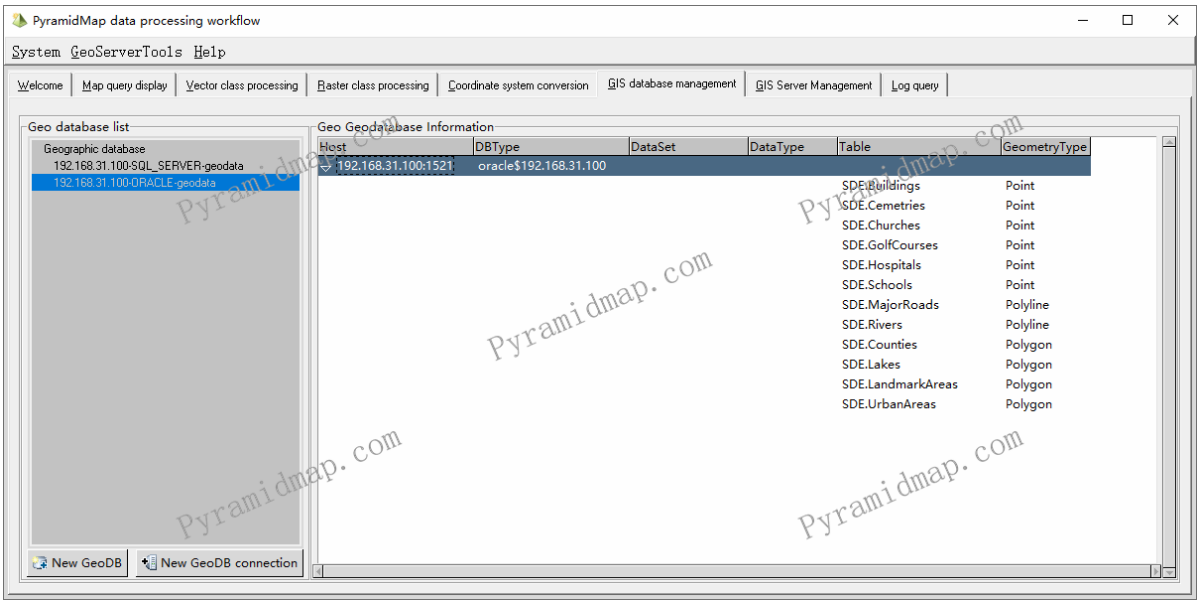


Figure 4-10: View the newly created geographical database in PyramidMap

The above process demonstrates the complete process of creating Oracle type enterprises and geographical database in PyramidMap. This process is also applicable to the creation of all types of geographical database, including SQL Server and PostgreSQL.

4.2.2: Create geodatabase in SQL server

PyramidMap creates a geo database in SQL Server, as shown in Figure 4-11.

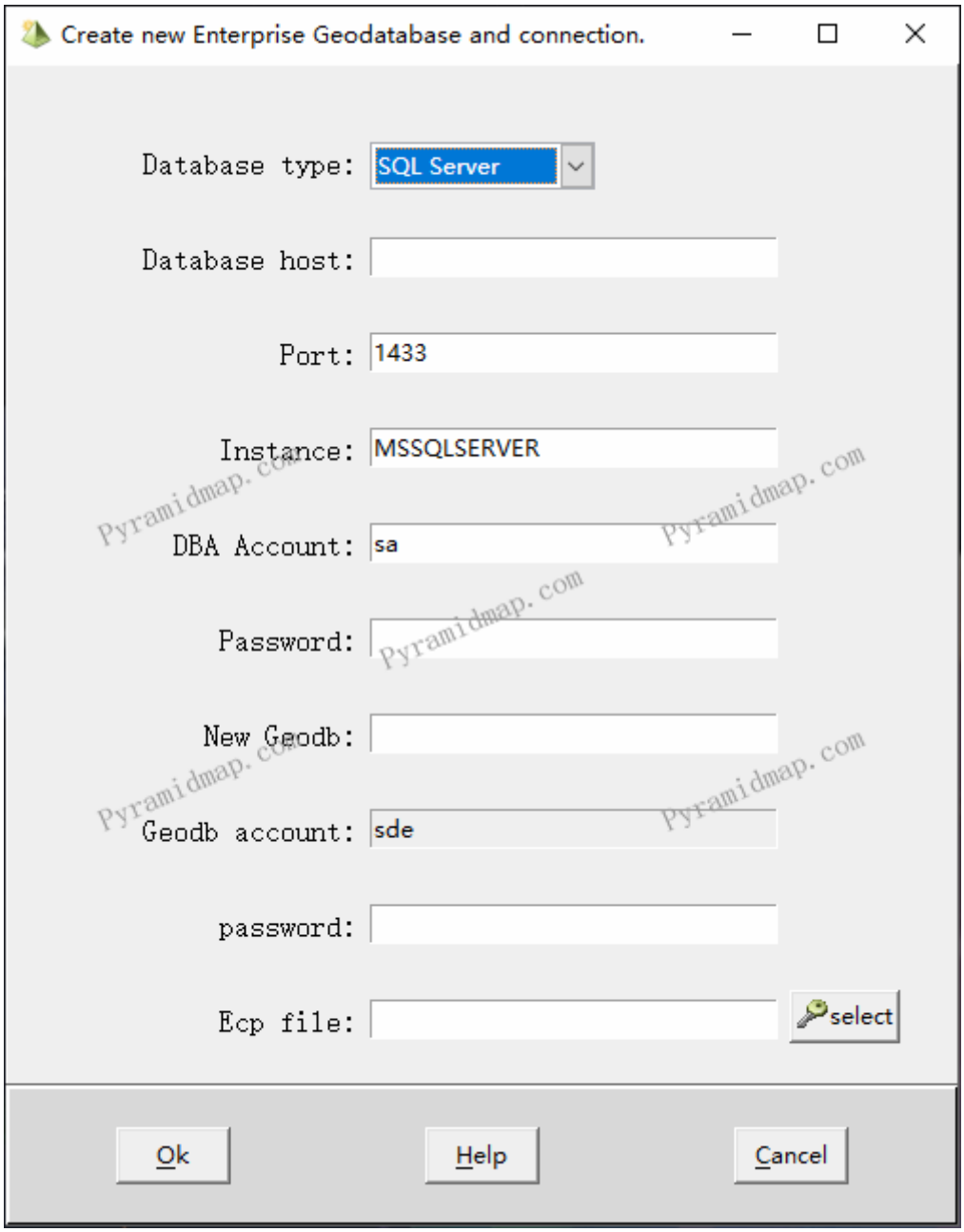


Figure 4-11: PyramidMap creates a geo database in SQL Server

Here the service instance of SQL Server is the data service name of SQL Server, which is MSSQLSERVER by default. Please contact the database administrator for confirmation. Please enter the name of the geographical database to be created in "Create geographical database", and the program will be created during execution. The default new database account is' sde 'and cannot be modified during the creation process. This process will authorize the upgrade of the existing SQL Server database to a geographical database. PyramidMap provides cmd output monitoring and message results for the running process, which is the same as the process of [4.2.1: Create geodatabase in Oracle].

4.2.3: Create geodatabase in PostgreSQL

PyramidMap creates a geo database in PostgreSQL, as shown in Figure 4-12.

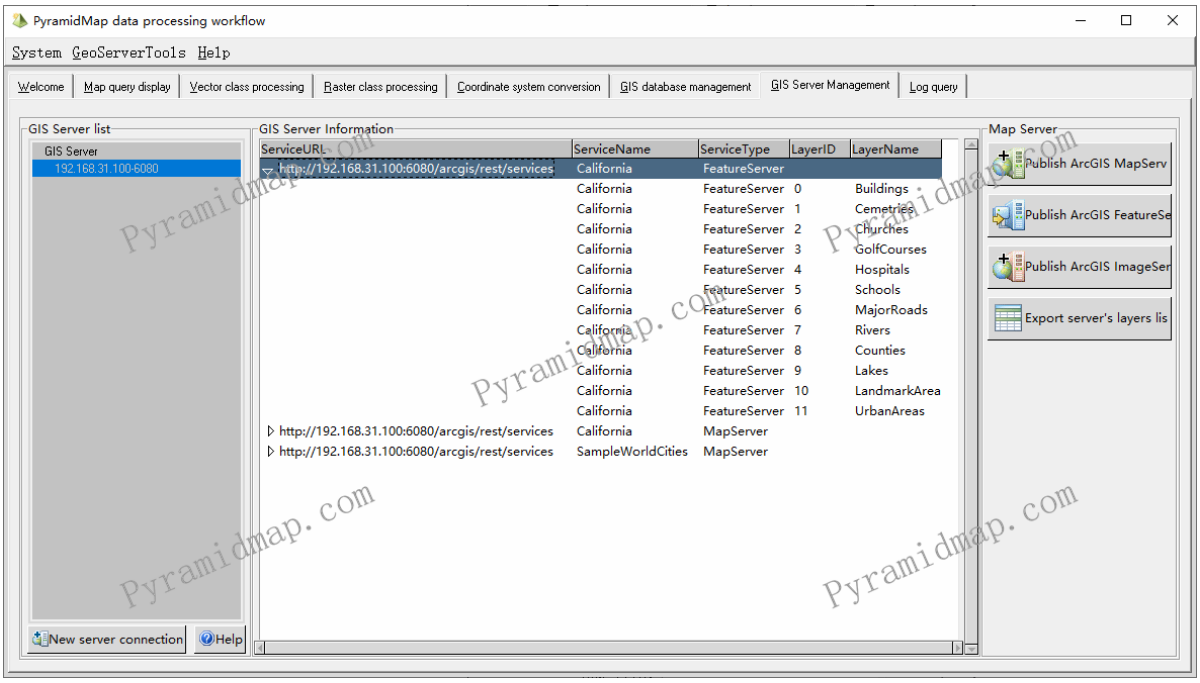


Figure 4-12: PyramidMap creates a geo database in PostgreSQL

Please enter the name of the geographical database to be created in "Create geographical database", and the program will be created during execution. The default new database account is' sde 'and cannot be modified during the creation process. This process will authorize the upgrade of the existing PostgreSQL database to a geographical database. PyramidMap provides cmd output monitoring and message results for the running process, which is the same as the process of [4.2.1: Create geodatabase in Oracle].

All the above creation processes require valid ArcGIS ecp authorization file support. The above is the process of creating a geographical database based on the existing database. If the geographical database already exists, you only need to create a connection. The following is the process of creating a geographical database connection.

4.3: Create GIS Database Connections

Unlike [4.2: Create GIS Database and Connections], the [Create GIS Database Connections] described in this section does not participate in the creation of GIS databases, but rather creates connection files based on existing GIS databases.

4.3.1: Create Oracle connection

If a geographic database already exists in Oracle, you can directly create a connection to this database in PyramidMap, as shown in Figure 4-13.

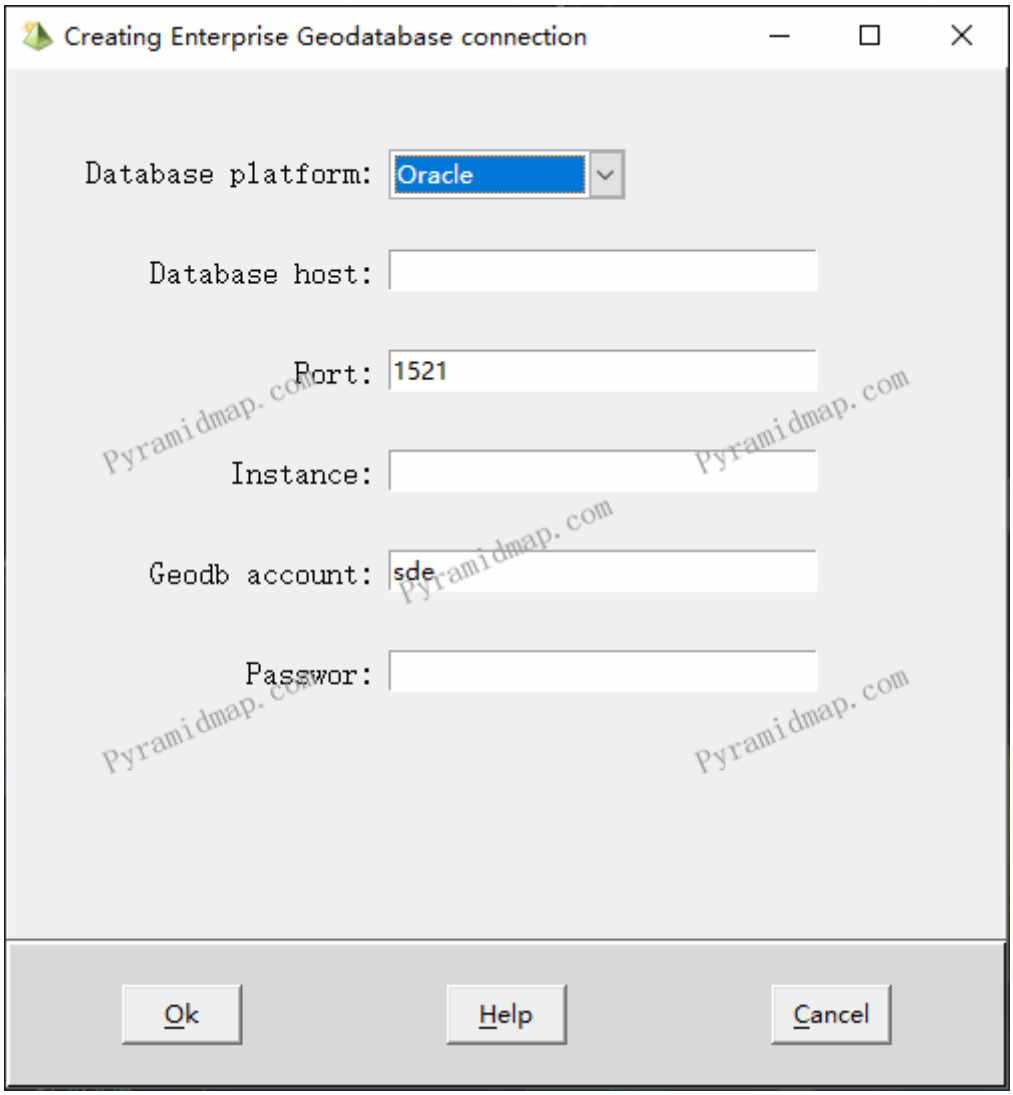


Figure 4-13: PyramidMap creates the Oracle connection

The instance is the name of the database that has created the geographical database function in Oracle.

4.3.2: Create SQL Server connection

If a geographic database already exists in SQL Server, you can directly create a connection to this database in PyramidMap, as shown in Figure 4-14.

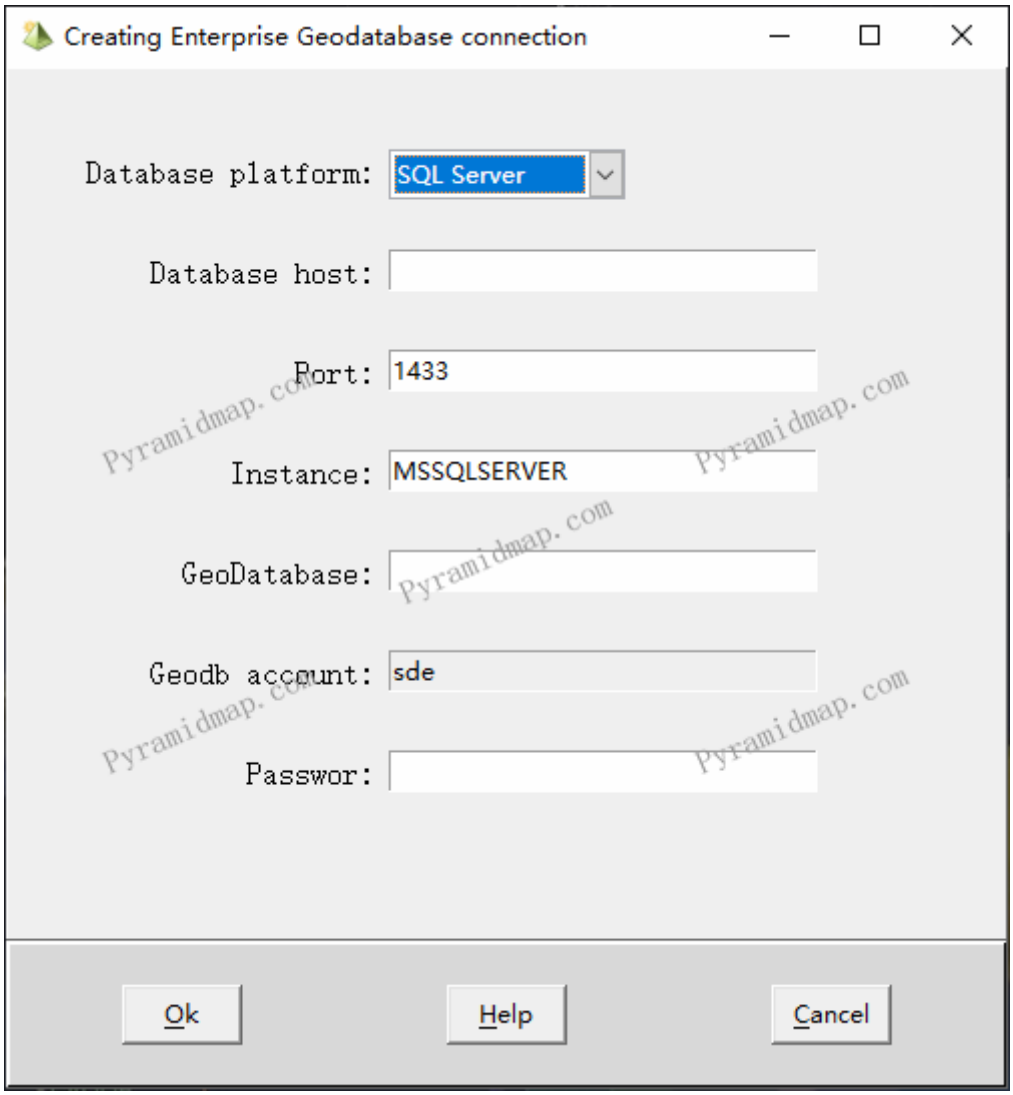


Figure 4-14: PyramidMap creates the SQL Server connection

The GeoDatabase input is name of the Geodatabase that has been created in SQL Server which you want to connect.

4.3.3: Create PostgreSQL connection

If a geographic database already exists in PostgreSQL, you can directly create a connection to this database in PyramidMap, as shown in Figure 4-15.

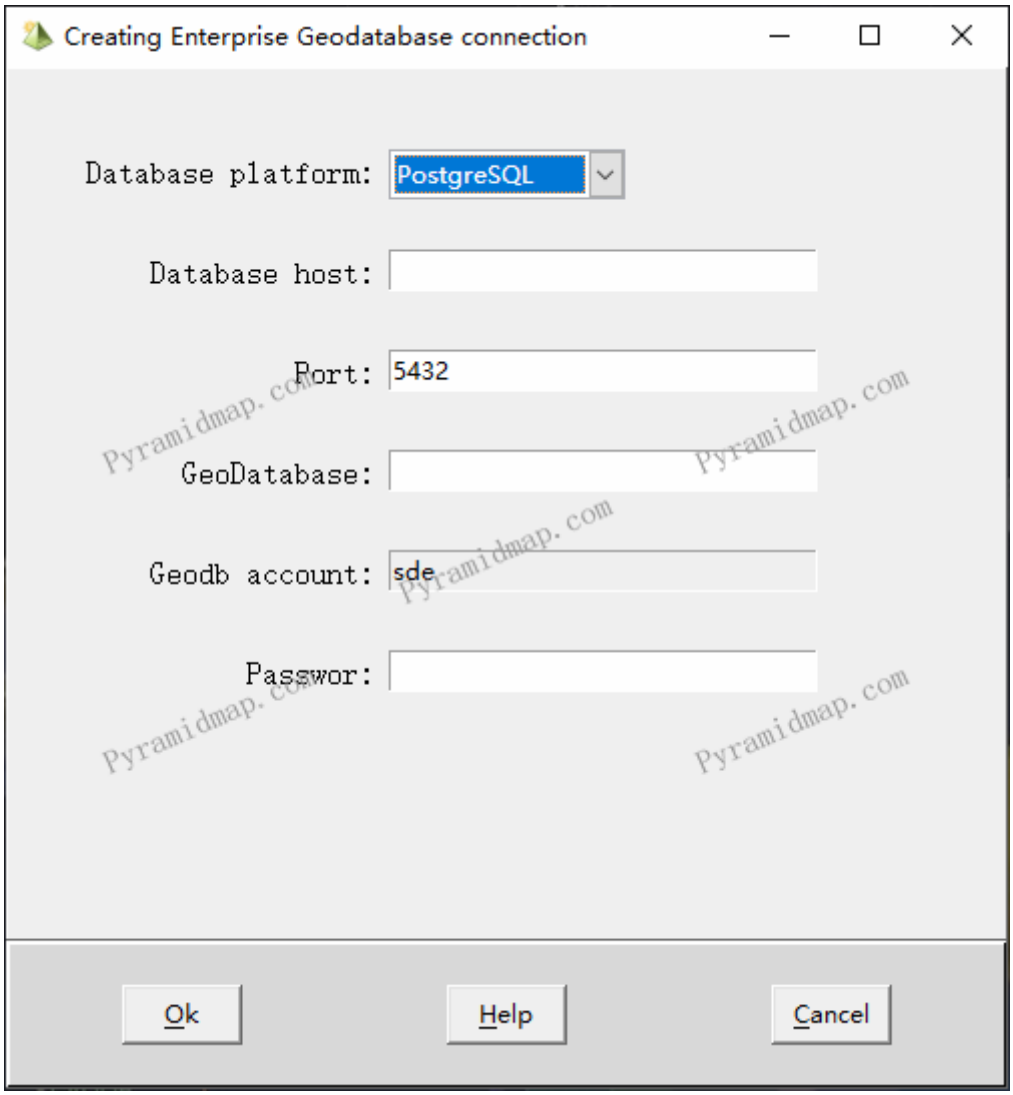


Figure 4-15: PyramidMap creates the PostgreSQL connection

The GeoDatabase input is name of the Geodatabase that has been created in PostgreSQL which you want to connect.

4.4: GIS database application

The GIS enterprise databases provide storage and access functions for map data, which is the foundation of GIS applications. In PyramidMap, you can directly access the GIS database, obtain its layer data, and process it, as shown in Figure 4-16.

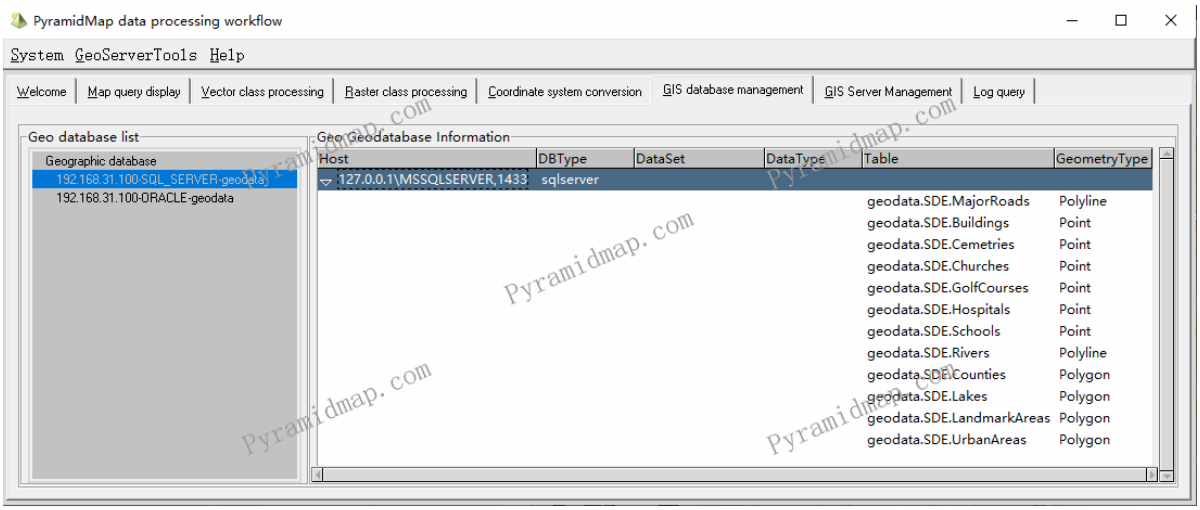


Figure 4-16: PyramidMap access Processing to GIS enterprise databases

PyramidMap will parse the map data in the GIS database, preview and process the database layers , as shown in Figure 4-17.

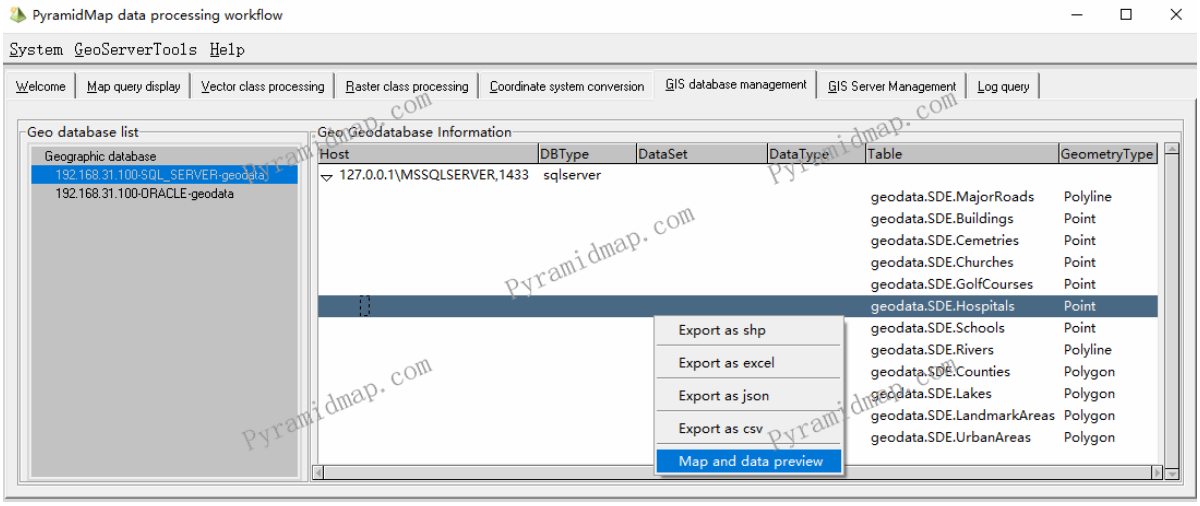


Figure 4-17: Preview and process the database layers in the PyramidMap

This instance takes Oracle geographical database as an example. The preview effect of PyramidMap on its Hospitals layer is shown in Figure 4-18.

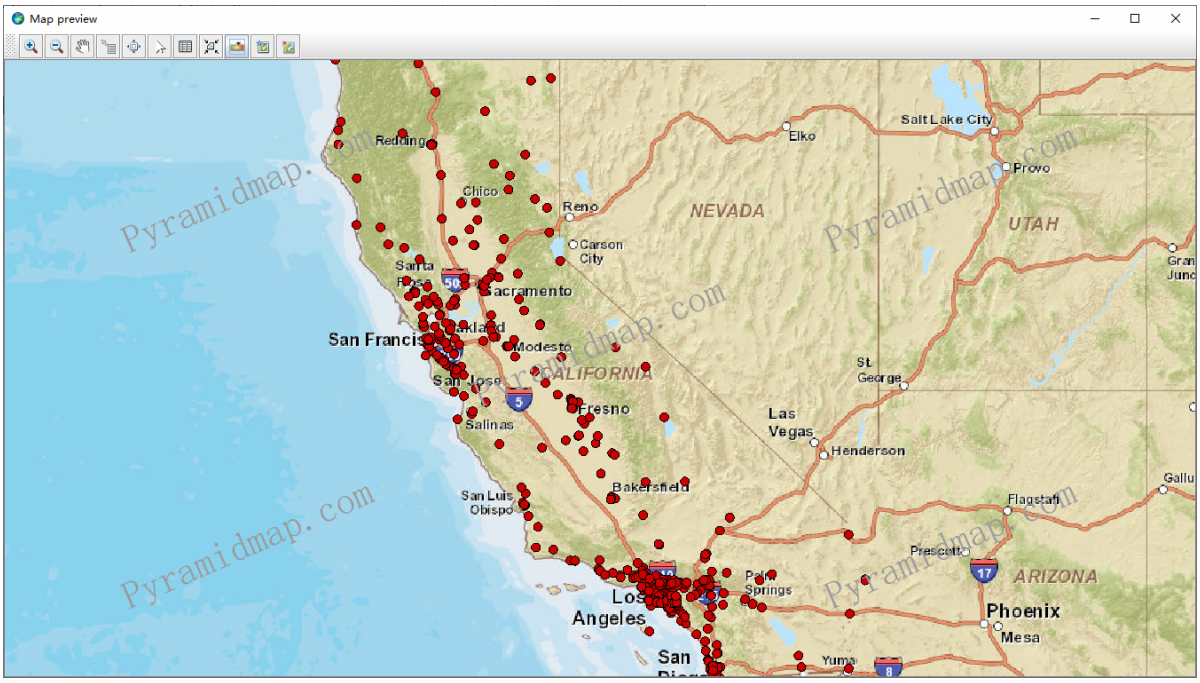


Figure 4-18: Preview and process the database layers in the PyramidMap

The above example demonstrates the complete process of importing layers into and echoing from geodatabase. We illustrate the role of geodatabase that is to store, edit, query just like the traditional database, and further support the versioning management of map data through the above example. This is the foundation for ArcGIS Server to implement FeatureServer online editing of layers, and its application is very extensive. We will continue to discuss and learn more about GIS databases in the following subjects: 9.2: FeatureServer Publishing Process, [9.2: Publish FeatureServer], [10: GIS Database], and [11: GIS Server].

5: Display map

PyramidMap uses resource management pool mode to uniformly manage map data in the map query display module, as shown in Figure 5-1.

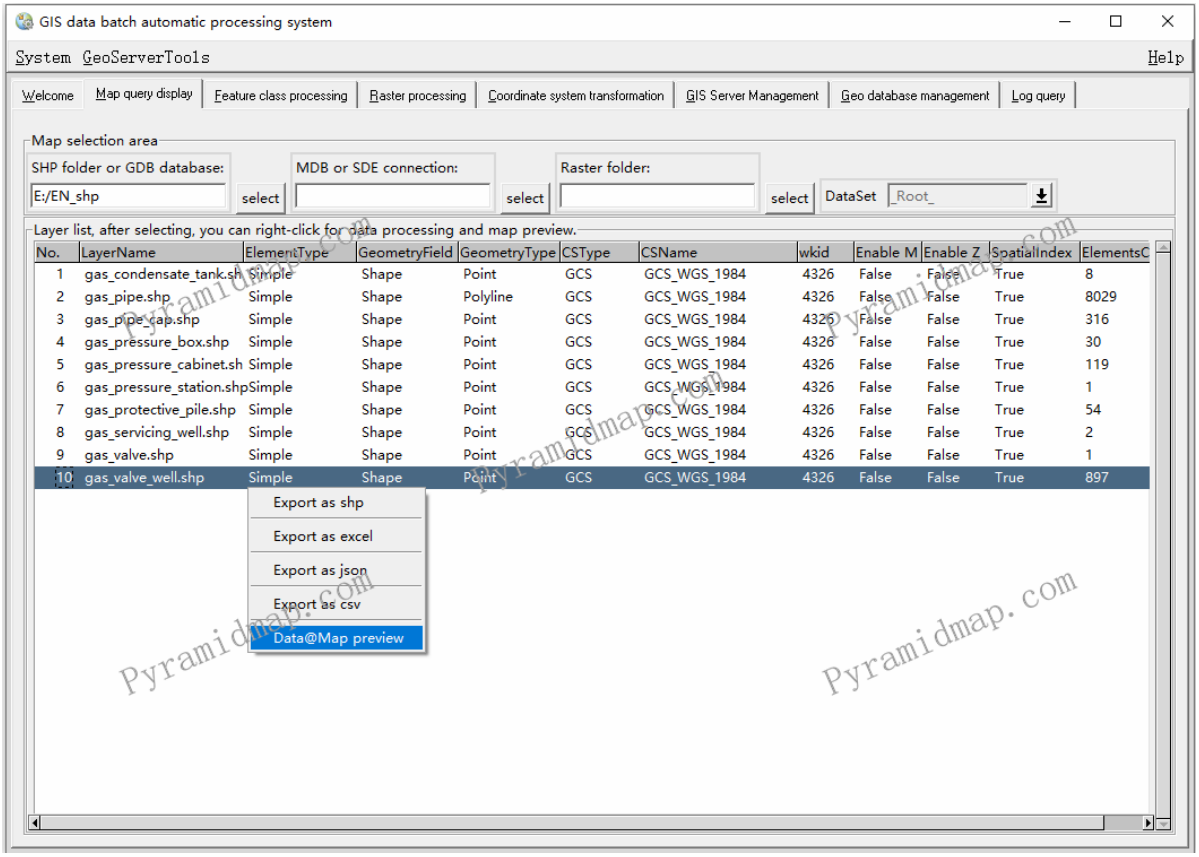


Figure 5-1: PyramidMap for resource management of map data

PyramidMap supports map data sources including Shp, Gdb file geographical database, Mdb personal geographical database, enterprise level geographic data based on DBMS, and raster image files, which are managed in a unified way by resource management pool. The map resource management pool displays layer names, feature types, geometric types, coordinate system types, names, and codes in a data list format, as well as whether there is a spatial index and the number of features in each layer. Through the right-click shortcut menu, layer preview and data conversion are achieved. The PyramidMap map window can display vector and image maps from different data sources, and query layer data lists. Taking Counties.shp in the resource pool as example displaying in PyramidMap as shown in Figure 5-2.

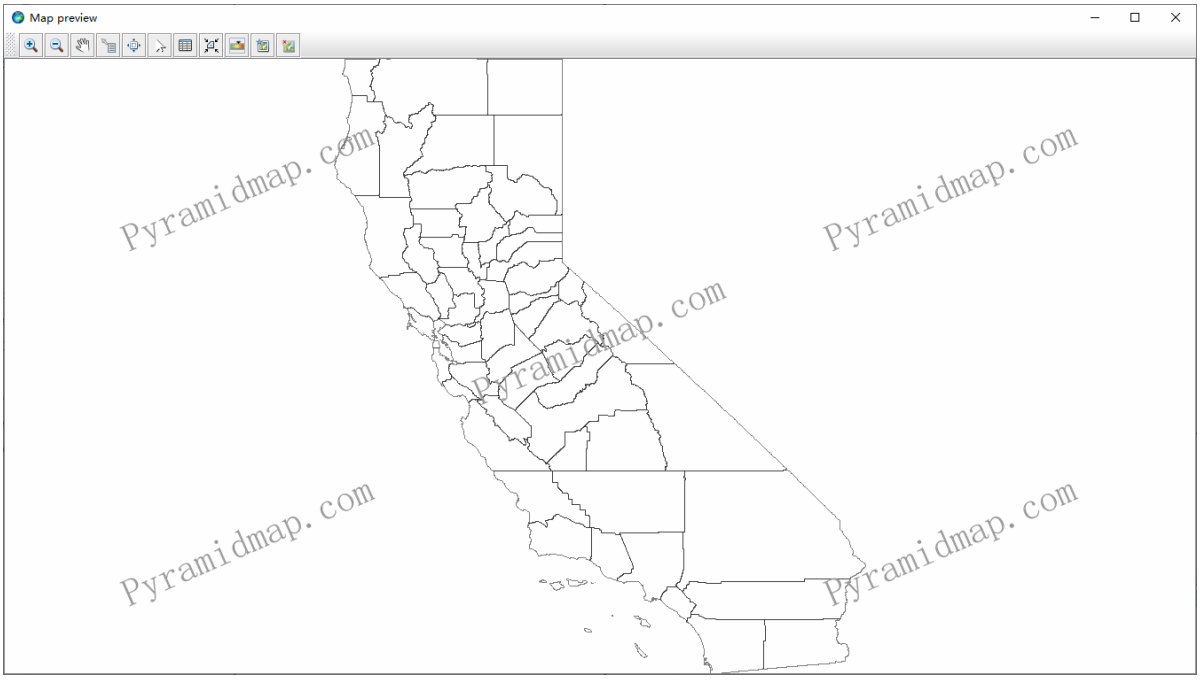


Figure 5-2: Taking Counties.shp in the resource pool as example displaying in PyramidMap

PyramidMap defines the rendering effect of map features through the visualization palette, as shown in Figure 5-3.

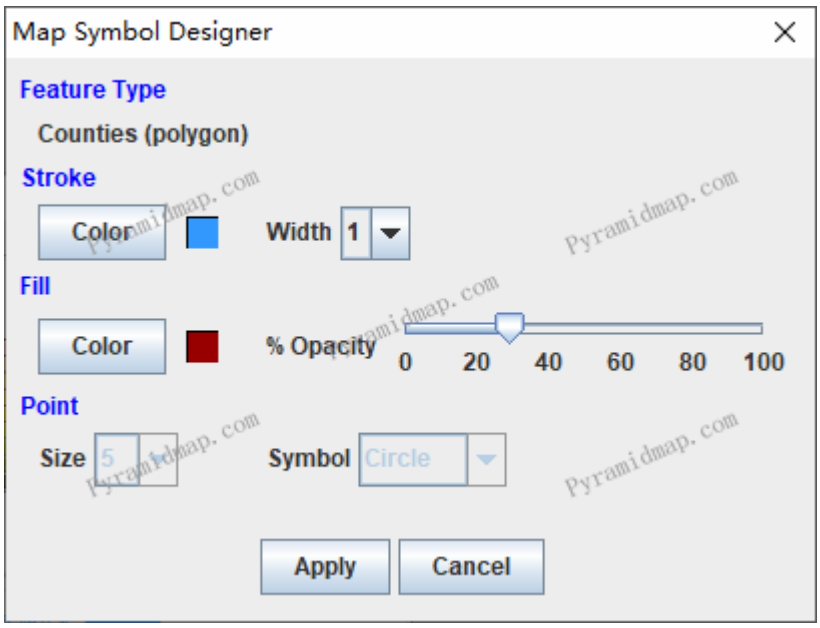


Figure 5-3: PyramidMap defines the rendering effect of map features through a visualization palette

PyramidMap defines symbols for different geometric types such as Point, Polyline, and Polygon in the visualization palette, describing stroke color, fill color, stroke width, the graphic symbol and the mark size (only applicable to Point types). The palette effect is shown in Figure 5-4.

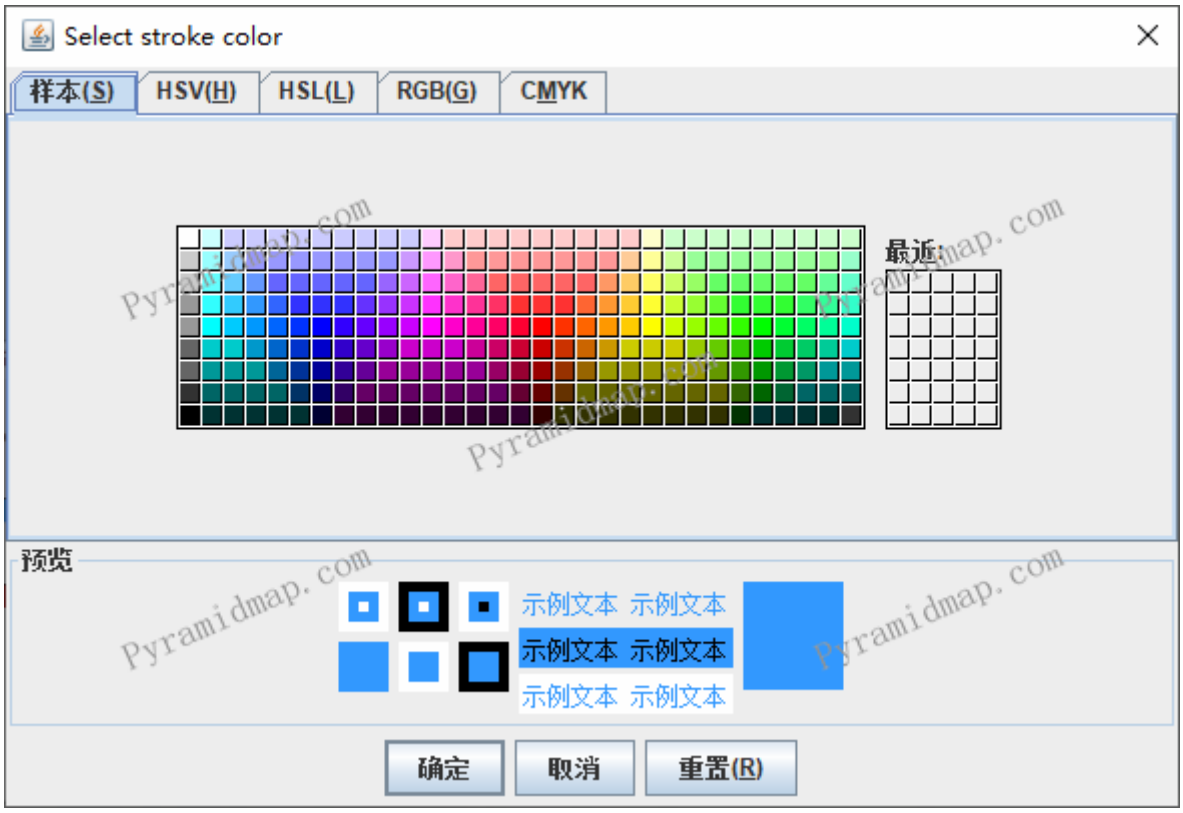


Figure 5-4: PyramidMap symbol definition visualization palette

Define the rendering symbols of the features through the palette, and the rendered effect is shown in Figure 5-5.

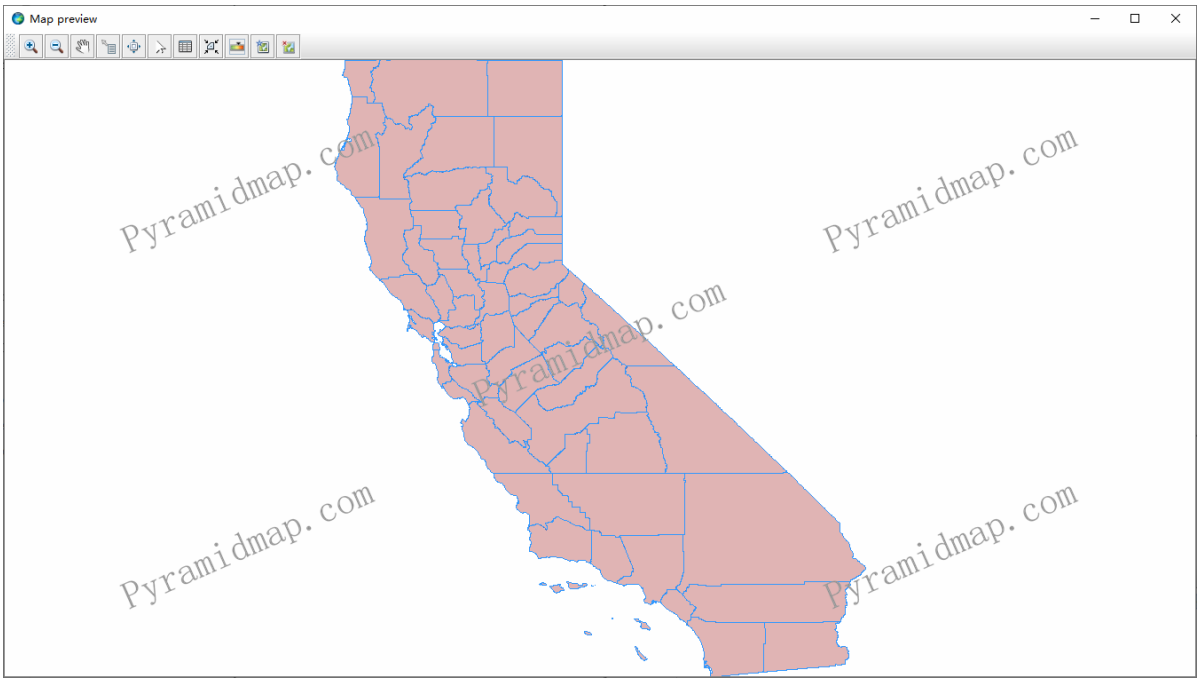


Figure 5-5: Layer rendering effect defined by symbols

PyramidMap provides multiple base image interfaces for accurate position reference and color rendering, as shown in Figures 5-6

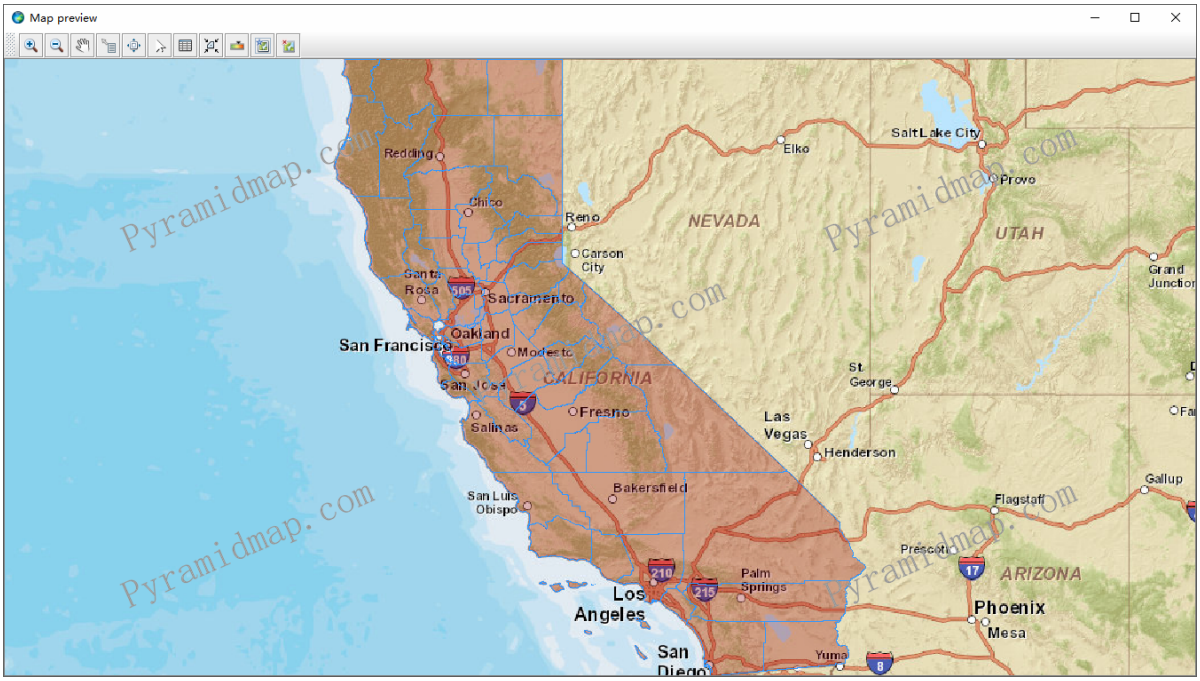


Figure 5-6: PyramidMap overlay base map interface for accurate position reference and color rendering

Click on the map to display the detailed information of the selected map features, as shown in Figure 5-7.

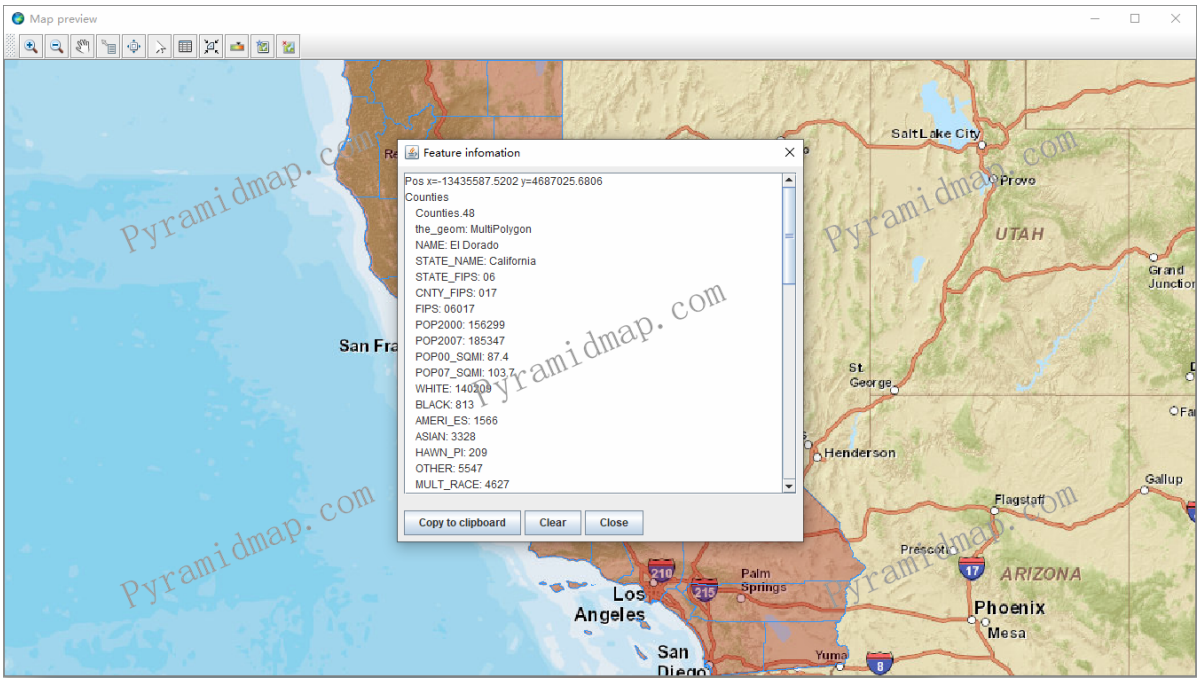


Figure 5-7: Display the detailed information of the selected map features

Open the layer data table and display all feature data of the layer, as shown in Figure 5-8.

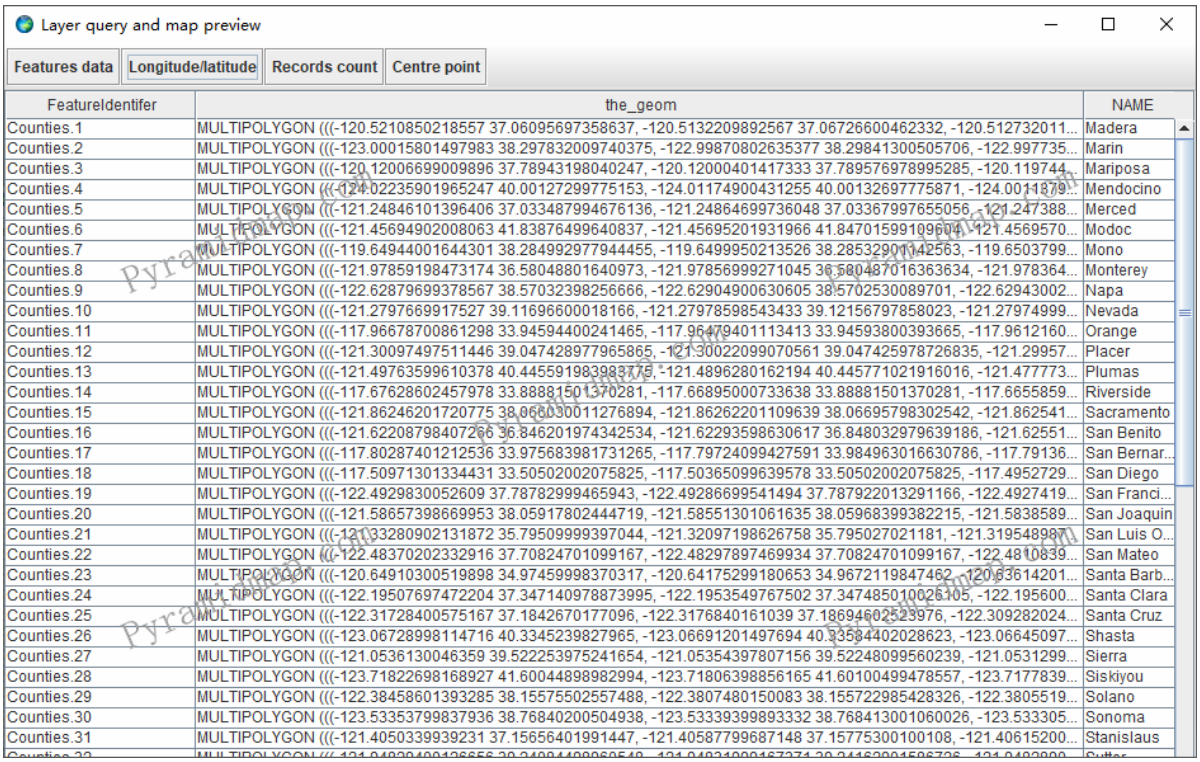
Layer query and map preview

Features data | Longitude/latitude | Records count | Centre point

FeatureIde...	the_geom	NAME	STATE_NA...	STATE_FIPS	CNTY_FIPS	FIPS	POP2000	POP2007	POP00_SQ...	POP07_SQ...	WHITE	I
Counties.1	MULTIPOL...	Madera	California	06	039	06039	123109	149180	57.2	69.3	76612	507
Counties.2	MULTIPOL...	Marin	California	06	041	06041	247289	251757	470.8	479.3	207800	714
Counties.3	MULTIPOL...	Mariposa	California	06	043	06043	17130	18680	11.7	12.8	15234	114
Counties.4	MULTIPOL...	Mendocino	California	06	045	06045	86265	90385	24.6	25.7	66671	536
Counties.5	MULTIPOL...	Merced	California	06	047	06047	210554	256700	106.9	130.3	118350	806
Counties.6	MULTIPOL...	Modoc	California	06	049	06049	9449	9631	2.2	2.3	8120	65
Counties.7	MULTIPOL...	Mono	California	06	051	06051	12853	13376	4.1	4.3	10818	61
Counties.8	MULTIPOL...	Monterey	California	06	053	06053	401762	425924	121.2	128.5	224682	150
Counties.9	MULTIPOL...	Napa	California	06	055	06055	124279	137087	157.6	173.8	99396	164
Counties.10	MULTIPOL...	Nevada	California	06	057	06057	92033	102193	94.4	104.9	85948	259
Counties.11	MULTIPOL...	Orange	California	06	059	06059	2846289	3081783	3559.8	3854.3	1844652	476
Counties.12	MULTIPOL...	Placer	California	06	061	06061	248399	339691	165.4	226.2	220053	203
Counties.13	MULTIPOL...	Plumas	California	06	063	06063	20824	22151	8.0	8.5	19113	130
Counties.14	MULTIPOL...	Riverside	California	06	065	06065	1545387	2100707	211.6	287.6	1013478	964
Counties.15	MULTIPOL...	Sacramento	California	06	067	06067	1223499	1421408	1229.2	1428.0	783240	121
Counties.16	MULTIPOL...	San Benito	California	06	069	06069	53234	57882	38.3	41.6	34695	573
Counties.17	MULTIPOL...	San Bernar...	California	06	071	06071	1709434	2051757	85.0	102.0	1006960	155
Counties.18	MULTIPOL...	San Diego	California	06	073	06073	2813833	3064142	663.8	722.9	1871839	161
Counties.19	MULTIPOL...	San Franci...	California	06	075	06075	776733	796417	16470.2	16887.6	385728	605
Counties.20	MULTIPOL...	San Joaquin	California	06	077	06077	563598	694530	395.6	487.5	327607	376
Counties.21	MULTIPOL...	San Luis O...	California	06	079	06079	246681	267623	74.3	80.6	208699	500
Counties.22	MULTIPOL...	San Mateo	California	06	081	06081	707161	720611	1553.0	1582.5	420683	248
Counties.23	MULTIPOL...	Santa Barb...	California	06	083	06083	399347	422299	145.2	153.5	290418	919
Counties.24	MULTIPOL...	Santa Clara	California	06	085	06085	1682585	1771177	1295.6	1363.8	905660	471
Counties.25	MULTIPOL...	Santa Cruz	California	06	087	06087	255602	264678	572.8	592.6	191931	247
Counties.26	MULTIPOL...	Shasta	California	06	089	06089	163256	183901	42.4	47.8	145826	122
Counties.27	MULTIPOL...	Sierra	California	06	091	06091	3555	3521	3.7	3.7	3348	7
Counties.28	MULTIPOL...	Siskiyou	California	06	093	06093	44301	46368	7.0	7.3	38573	580
Counties.29	MULTIPOL...	Solano	California	06	095	06095	394542	426952	443.5	480.0	222387	588
Counties.30	MULTIPOL...	Sonoma	California	06	097	06097	458614	483728	288.4	304.2	374209	652
Counties.31	MULTIPOL...	Stanislaus	California	06	099	06099	446997	529038	295.3	349.5	309901	115

Figure 5-8: Open layer data table

Open the latitude and longitude data table, as shown in Figures 5-9.



FeatureIdentifer	the_geom	NAME
Counties.1	MULTIPOLYGON (((-120.5210850218557 37.06095697358637, -120.5132209892567 37.06726600462332, -120.512732011...	Madera
Counties.2	MULTIPOLYGON (((-123.00015801497983 38.297832009740375, -122.99870802635377 38.29841300505706, -122.997735...	Marin
Counties.3	MULTIPOLYGON (((-120.12006699009896 37.78943198040247, -120.12000401417333 37.789576978995285, -120.119744...	Mariposa
Counties.4	MULTIPOLYGON (((-124.02235901965247 40.00127299775153, -124.01174900431255 40.00132697775871, -124.0011879...	Mendocino
Counties.5	MULTIPOLYGON (((-121.24846101396406 37.033487994676136, -121.24864699736048 37.03367997655056, -121.247388...	Merced
Counties.6	MULTIPOLYGON (((-121.45694902008063 41.83876499640837, -121.45695201931966 41.84701599109604, -121.4569570...	Modoc
Counties.7	MULTIPOLYGON (((-119.64944001644301 38.284992977944455, -119.6499950213526 38.28532901042563, -119.6503799...	Mono
Counties.8	MULTIPOLYGON (((-121.97859198473174 36.58048801640973, -121.97856999271045 36.580487016363634, -121.978364...	Monterey
Counties.9	MULTIPOLYGON (((-122.62879699378567 38.57032398256666, -122.62904900630605 38.5702530089701, -122.62943002...	Napa
Counties.10	MULTIPOLYGON (((-121.2797669917527 39.11696600018166, -121.27978598543433 39.12156797858023, -121.27974999...	Nevada
Counties.11	MULTIPOLYGON (((-117.96678700861298 33.94594400241465, -117.96479401113413 33.94593800393665, -117.9612160...	Orange
Counties.12	MULTIPOLYGON (((-121.30097497511446 39.047428977965865, -121.30022099070561 39.047425978726835, -121.29957...	Placer
Counties.13	MULTIPOLYGON (((-121.49763599610378 40.44559198398377[illegible], -121.4896280162194 40.445771021916016, -121.477773...	Plumas
Counties.14	MULTIPOLYGON (((-117.67628602457978 33.88881501370281, -117.66895000733638 33.88881501370281, -117.6655859...	Riverside
Counties.15	MULTIPOLYGON (((-121.86246201720775 38.[illegible]3011276894, -121.86262201109639 38.06695798302542, -121.862541...	Sacramento
Counties.16	MULTIPOLYGON (((-121.62208798407266 36.846201974342534, -121.62293598630617 36.848032979639186, -121.62551...	San Benito
Counties.17	MULTIPOLYGON (((-117.80287401212536 33.975683981731265, -117.79724099427591 33.984963016630786, -117.79136...	San Bernar...
Counties.18	MULTIPOLYGON (((-117.50971301334431 33.50502002075825, -117.50365099639578 33.50502002075825, -117.4952729...	San Diego
Counties.19	MULTIPOLYGON (((-122.4929830052609 37.78782999465943, -122.49286699541494 37.7879220132911 66, -122.4927419...	San Franci...
Counties.20	MULTIPOLYGON (((-121.58657398669953 38.0591780244471 9, -121.58551301061635 38.05968399382215, -121.5838589...	San Joaquin
Counties.21	MULTIPOLYGON (((-121.33280902131872 35.79509999397044, -121.32097198626758 35.795027021181, -121.319548987...	San Luis O...
Counties.22	MULTIPOLYGON (((-122.48370202332916 37.70824701099167, -122.48297897469934 37.70824701099167, -122.4810839...	San Mateo
Counties.23	MULTIPOLYGON (((-120.64910300519898 34.97459998370317, -120.64175299180653 34.967211984746 2, -120.6063614201...	Santa Barb...
Counties.24	MULTIPOLYGON (((-122.19507697472204 37.347140978873995, -122.1953549767502 37.347485010026105, -122.195600...	Santa Clara
Counties.25	MULTIPOLYGON (((-122.31728400575167 37.1842670177096, -122.3176840161039 37.1869460523976, -122.309282024...	Santa Cruz
Counties.26	MULTIPOLYGON (((-123.06728998114716 40.3345239827965, -123.06691201497694 40.33584402028623, -123.06645097...	Shasta
Counties.27	MULTIPOLYGON (((-121.0536130046359 39.522253975241654, -121.05354397807156 39.52248099560239, -121.0531299...	Sierra
Counties.28	MULTIPOLYGON (((-123.71822698168927 41.60044898982994, -123.71806398856165 41.60100499478557, -123.7177839...	Siskiyou
Counties.29	MULTIPOLYGON (((-122.38458601393285 38.15575502557488, -122.3807480150083 38.155722985428326, -122.3805519...	Solano
Counties.30	MULTIPOLYGON (((-123.53353799837936 38.76840200504938, -123.53339399893332 38.768413001060026, -123.533305...	Sonoma
Counties.31	MULTIPOLYGON (((-121.4050339939231 37.15656401991447, -121.40587799687148 37.15775300100108, -121.40615200...	Stanislaus

Figure 5-9: Open the layer longitude and latitude data table

The preview effect of PyramidMap on raster image layers is shown in Figures 5-10.

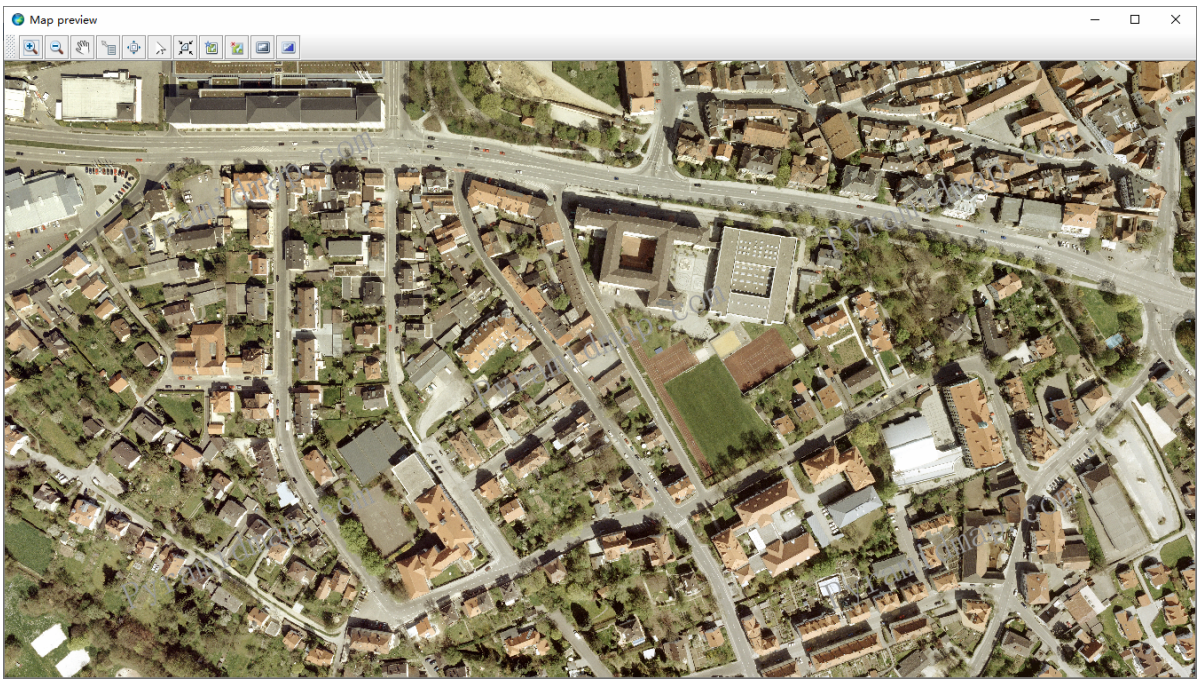


Figure 5-10: The preview effect of PyramidMap on raster image layers

In summary, PyramidMap can holographic display vector and grid layers through various channels such as map windows and data tables.

6: Feature Class process

PyramidMap provides the processing, transformation and service publishing of vector map elements in the "vector class processing" module, as shown in Figure 6-1.

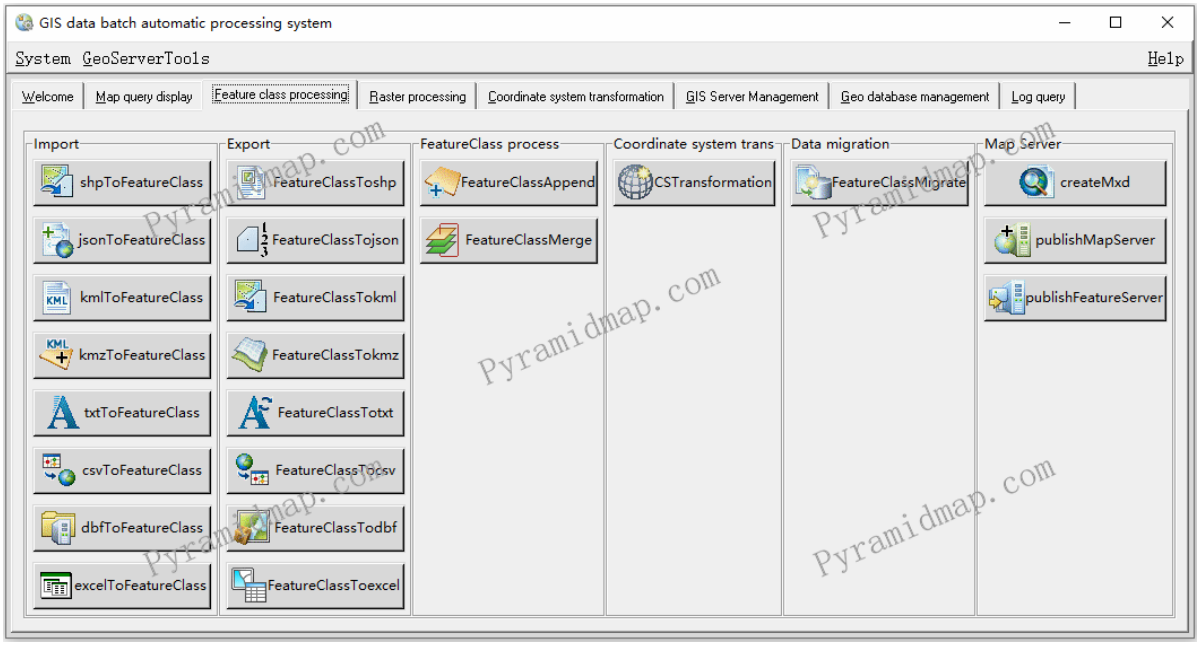


Figure 6-1: PyramidMap vector class processing, data conversion and service publishing module

6.1: Data to feature class

The function of converting data to feature class as shown in Figure 6-2.

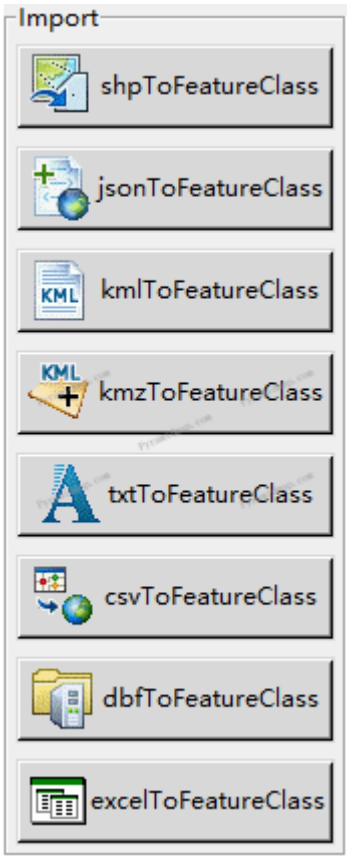


Figure 6-2: PyramidMap converts other data to feature class

The above functional areas support the conversion of SHP, JSON, KML, KMZ, TXT,CSV, DBF, Excel and other formats to map feature data. Click to enter the function interface, as shown in Figure 6-3.

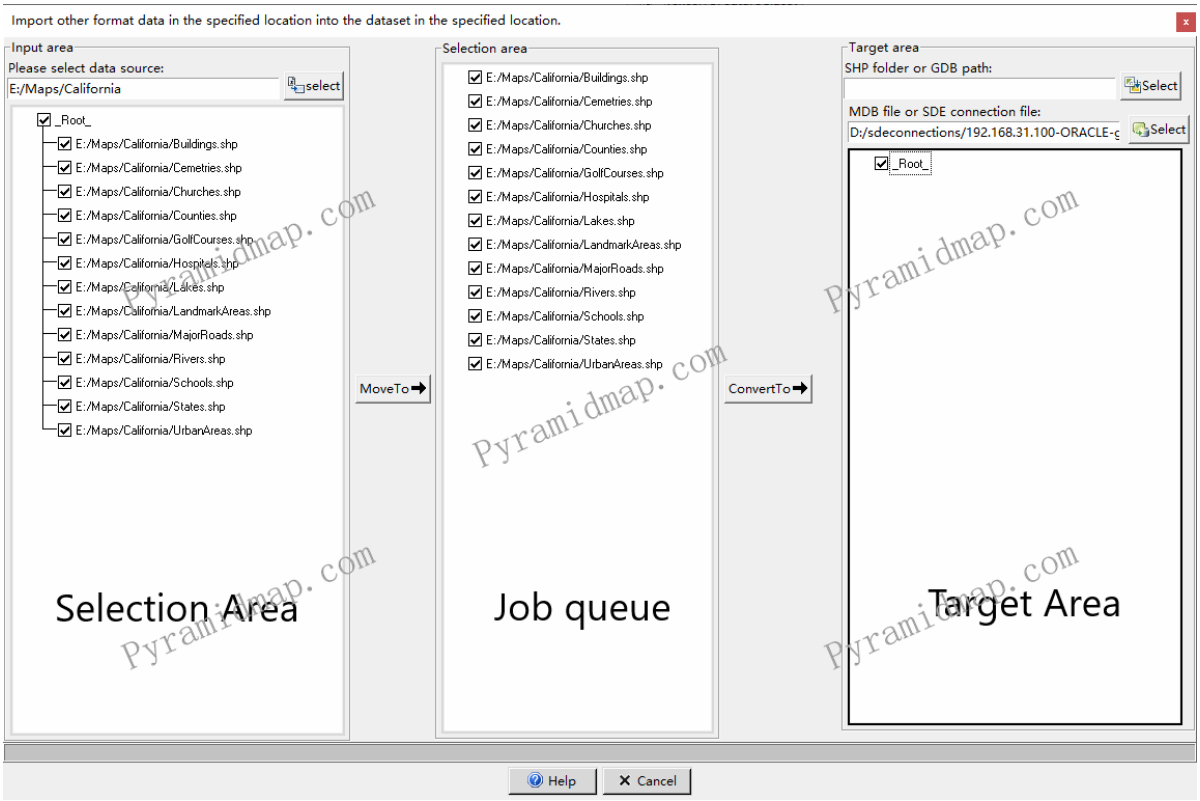


Figure 6-3: PyramidMap converts data in different formats into vector features interface

PyramidMap supports adding several selection areas to the job queue area. In the target area, the target path supports files, gdb files geodatabase, mdb personal geodatabase, and sde enterprise geodatabase, covering all types of GIS applications. From this, a data processing workflow is formed. Click "ConvertTo" to start batch data processing, and the execution model is shown in Figure 6-4:

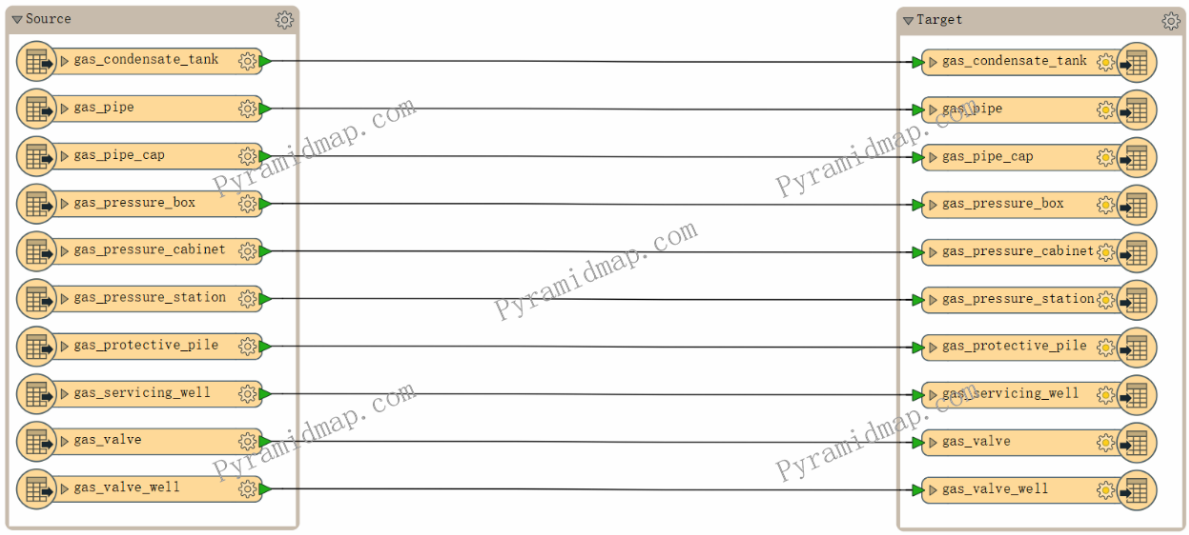


Figure 6-4: Import data to feature class processing model

In particular, exporting to a GIS enterprise geographical database is equivalent to importing feature classes into the database, which has the same effect as migrating to a geographical database in [6.4: Feature classs migration] and importing feature classes into a geographical database in [9.2: Publishing FeatureServer]. PyramidMap provides cmd output monitoring of the running process, as shown in Figures 6-5.

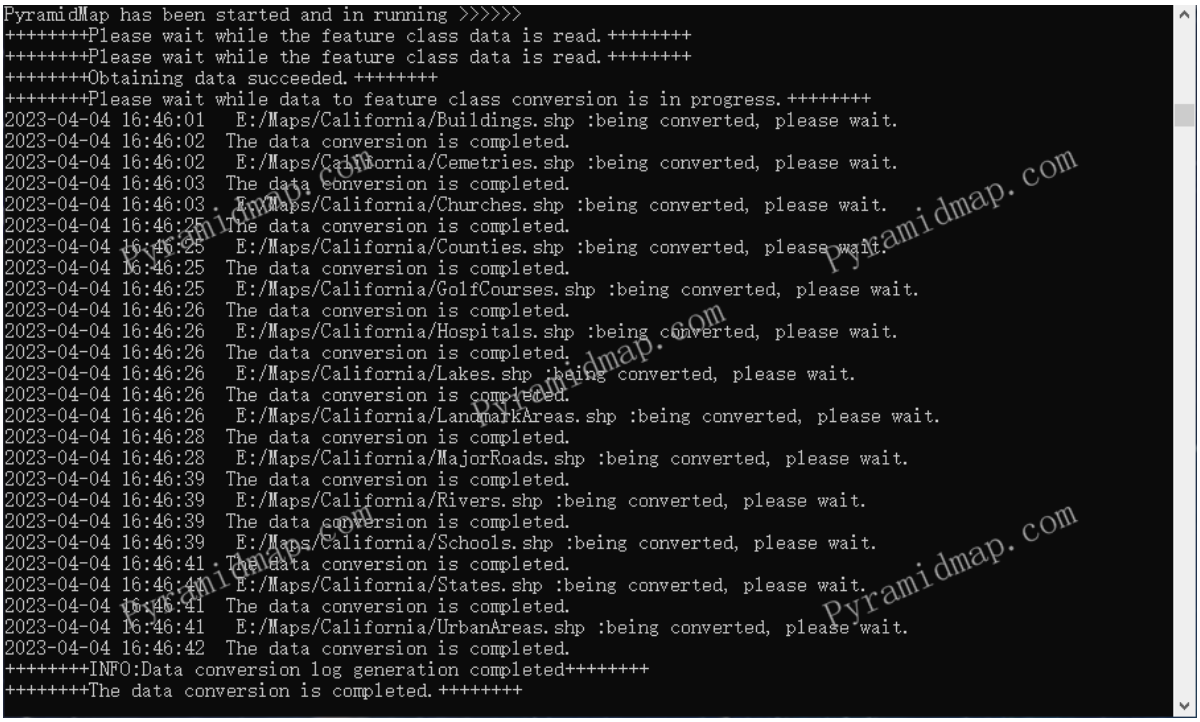


Figure 6-5: PyramidMap monitors the cmd output of the running process

The processing process automatically saves logs for later queries.

6.2: Feature class to data

As the reverse process of 6.1, PyramidMap converts map feature classes into multiple data formats, as shown in Figure 6-6.

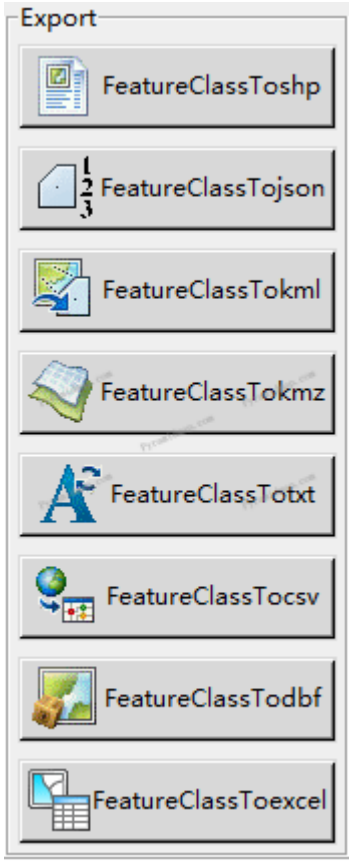


Figure 6-6: PyramidMap converts feature classes to other data

The above functional areas support featureclass to SHP, JSON, KML, KMZ, TXT,CSV, DBF, Excel and other types of data. Click to enter the function interface,as shown in Figure 6-7.

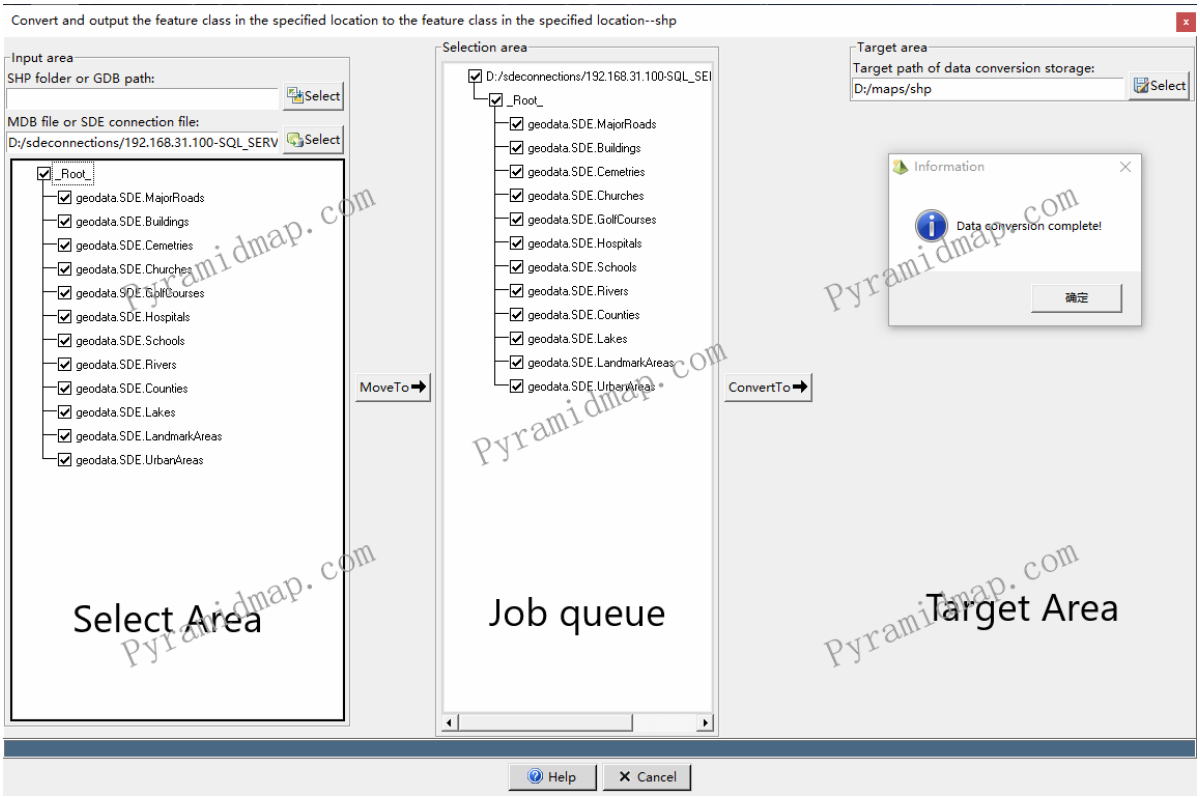


Figure 6-7: PyramidMap converts vector classes into multiple data formats interface

PyramidMap can summarize several selection areas to form a job queue area, and can reselect and clear the job list. The input data supports shp, gdb file type geographical database, mdb personal geographic database, sde enterprise type geographical database, covering all types of GIS applications. The output type supports shp, json, kml, kmz, txt, csv, dbf, and excel: this forms a data processing workflow. Click "Convert to" to start batch data processing. PyramidMap provides cmd output monitoring of the running process, as shown in Figures 6-8.

Figure 6-8: PyramidMap monitors the cmd output of feature classes transferred out

Taking the output MajorRoads.shp and Hospitals.shp layers as examples for sampling, the opening effect is shown in Figure 6-9.

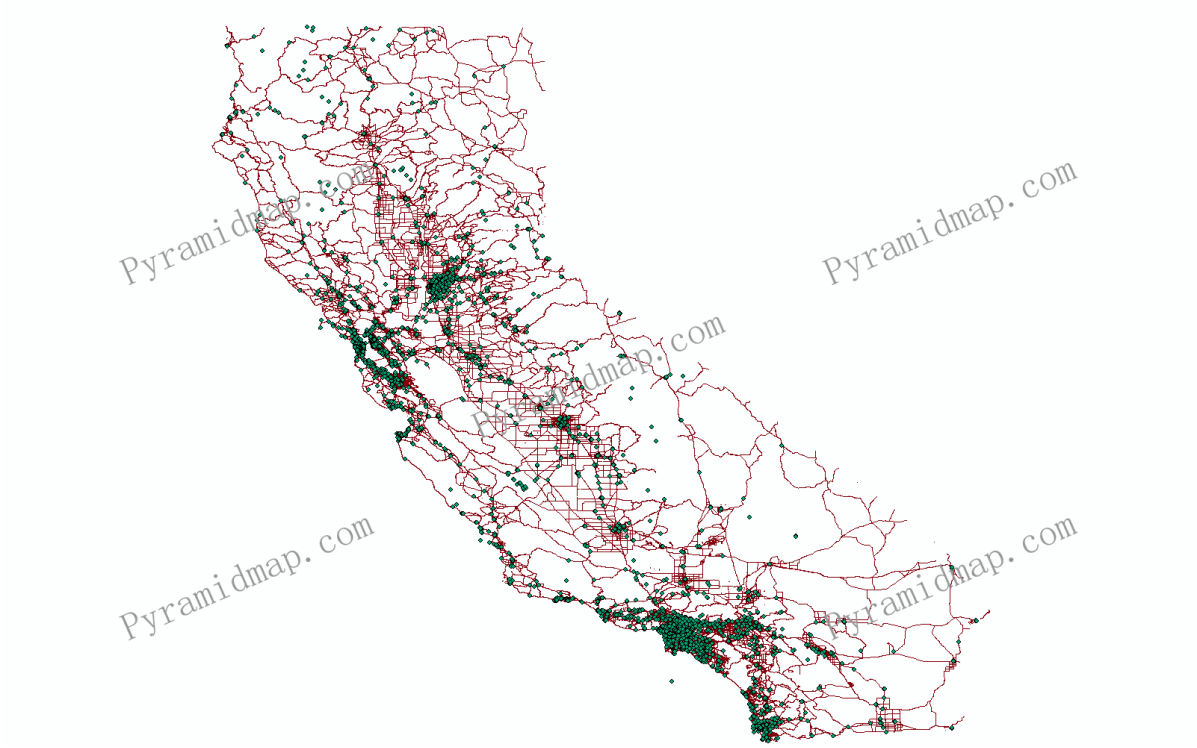


Figure 6-9: Open MajorRoads.shp and Hospitals.shp layer which converted output in PyramidMap

All actions will be saved as logs during processing.

6.3: Append and merge features

In the production process, there are many applications for appending or merging data from multiple layers onto one layer. PyramidMap supports user-defined implementation strategies and follows the generated specifications to achieve the operating of map features.

6.3.1: Features append

Map feature classes, including shp layers, gdb, mdb, and sde enterprise geographical database feature classes, are essentially data tables that store spatial geographic information. Feature class addition refers to adding multiple input datasets to the existing target dataset. The input dataset can be point, line, face feature class, table, grid, grid catalog, annotation feature class, or dimension feature class, with logical definitions shown in Figure 6-10.

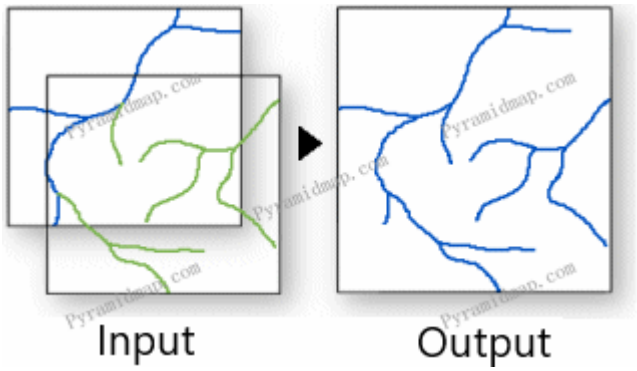


Figure 6-10: The logical definitions to PyramidMap features appending

The process of appending feature classes in PyramidMap is shown in Figures 6-11.

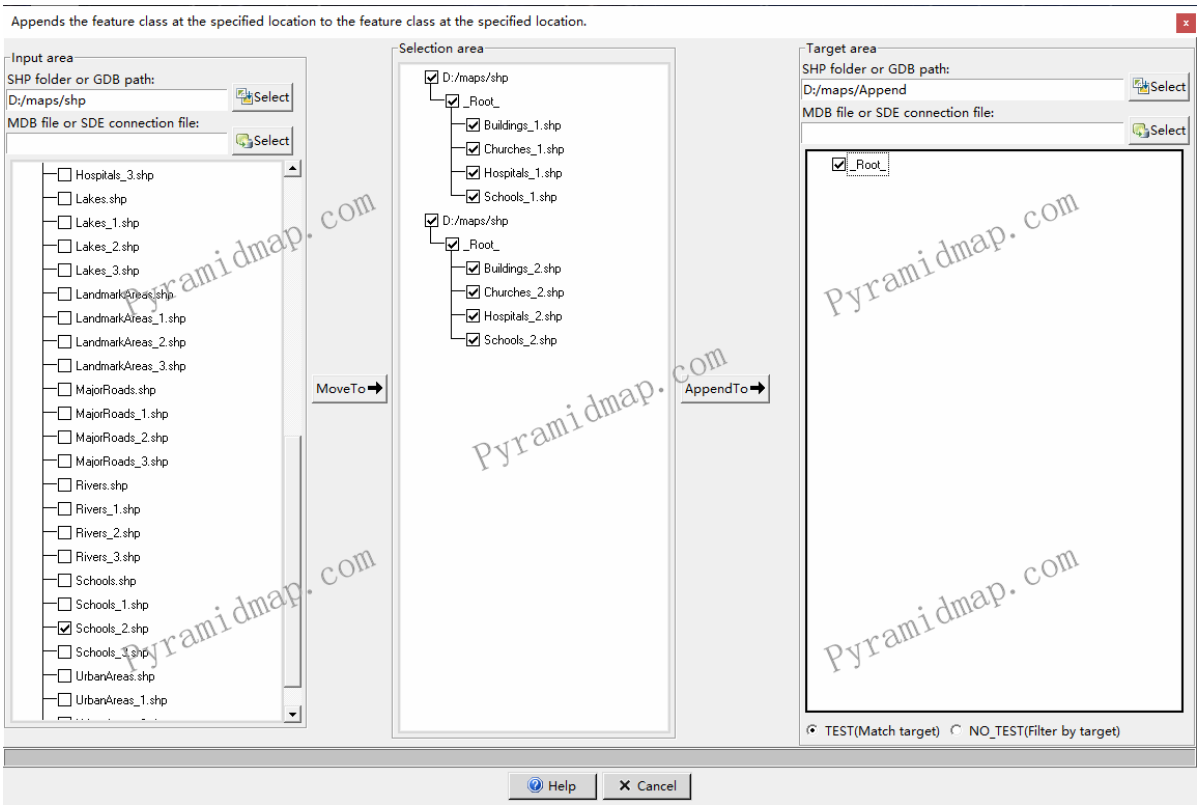


Figure 6-11: PyramidMap features appending interface

Additional data sources support shp, gdb, mdb, and enterprise geographical database. The first table is the parent table, and other table data is added to the parent table according to the same fields and data types. PyramidMap configures the addition strategy through the TEST or NO TEST parameters to determine whether to allow field structure differences or filter between the source and target data. The TEST mode is forcibly limited, and the source data and target data formats must be consistent, otherwise it cannot be run; The NO TEST mode will perform field mapping filtering, and only source data that meets the mapping rules will be appended. Specifically, for the NO TEST mode, PyramidMap provides the following field merging strategy, as shown in Figures 6-12.

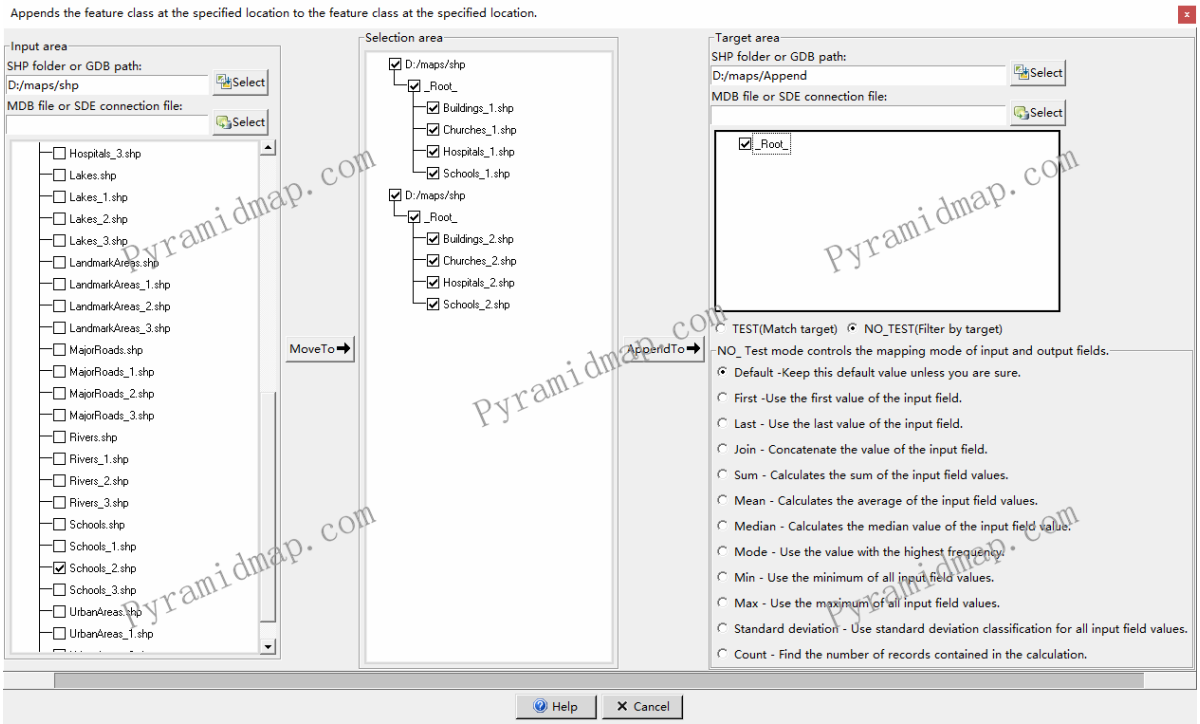


Figure 6-12: PyramidMap features append option in NO TEST mode

PyramidMap supports configuring the same addition strategy for batch feature classes, achieving one click completion. PyramidMap configures feature class addition strategies through parameters:

- TEST - The scheme (field definition) of the input dataset must match the scheme of the target dataset. If the scheme does not match, an error will be returned.
- NO_TEST - The scheme (field definition) of the input dataset does not need to match the scheme of the target dataset. However, if the fields of the input dataset do not match those of the target dataset, they will not be mapped to the target dataset unless explicitly set in the Field Mapping control.
- Field mapping strategy configuration in NO_TEST mode:
    - First - Use the first value of the input field.
    - Last - Use the last value of the input field.
    - Join - Concatenate the values of input fields.
    - Sum - Calculate the sum of input field values.
    - Mean - Calculate the average value of the input field values.
    - Median - Calculate the median of the input field values.
    - Mode - Use the value with the highest frequency.
    - Min - Use the minimum value among all input field values.
    - Max - Use the maximum value among all input field values.
    - Standard deviation - Use the standard deviation classification method for all input field values.
    - Count - Find the number of records included in the calculation.

Please follow the default mapping method unless you are determined clearly.

6.3.2: Features merge

Feature class merging multiple input datasets with the same data type into a new single output dataset. This tool can merge point, line, or surface feature classes or tables, with logical definitions shown in Figures 6-13.

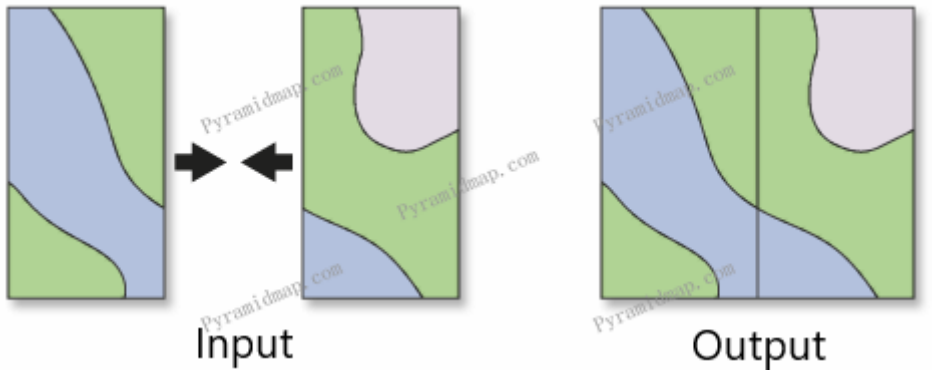


Figure 6-13: The logical definitions to PyramidMap features merging

The mapping method for feature class merging and appending fields is the same, but the difference is that the merging result will generate a new file and generate new values according to different merging strategies. The merging process is shown in Figure 6-14.

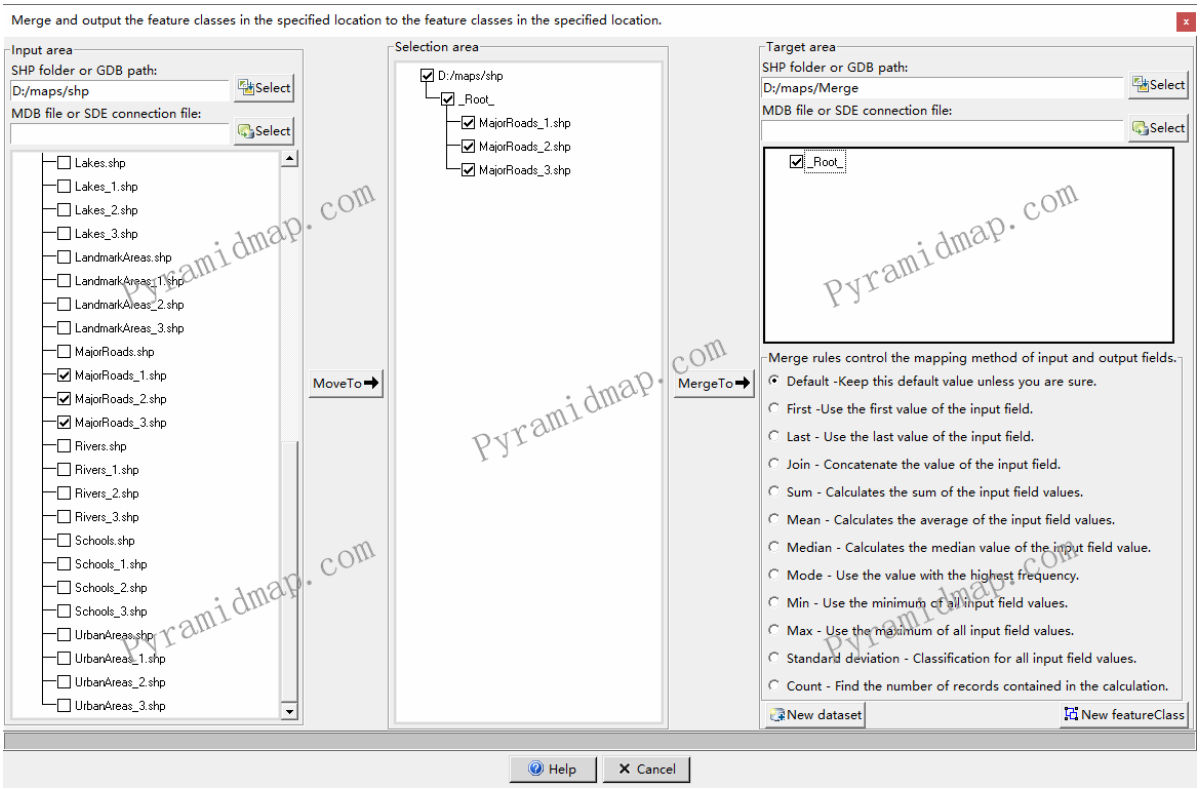


Figure 6-14: PyramidMap features merging interface

PyramidMap supports configuring the same merge strategy for batch feature classes, achieving one click completion. During the execution process, PyramidMap follows the following field merging strategy:

- First - Use the first value of the input field.
- Last - Use the last value of the input field.
- Join - Concatenate the values of input fields.
- Sum - Calculate the sum of input field values.
- Mean - Calculate the average value of the input field values.
- Median - Calculate the median of the input field values.
- Mode - Use the value with the highest frequency.
- Min - -Use the minimum value among all input field values.
- Max - Use the maximum value among all input field values.
- Standard deviation - Use the standard deviation classification method for all input field values.
- Count - Find the number of records included in the calculation.

6.4: Feature classes migration

Feature migration is the process of migrating feature classes from SHP, GDB, MDB, and SDE to a specified target path, and automatically completing data format conversion. This includes importing Shp/Gdb/Mdb map elements into the designated GIS enterprise geographical database, which is also the basic mode of GIS enterprise application. The data conversion models for feature class migration are shown in Figures 6-17 to 6-20, respectively.

6.4.1: Feature class migration model

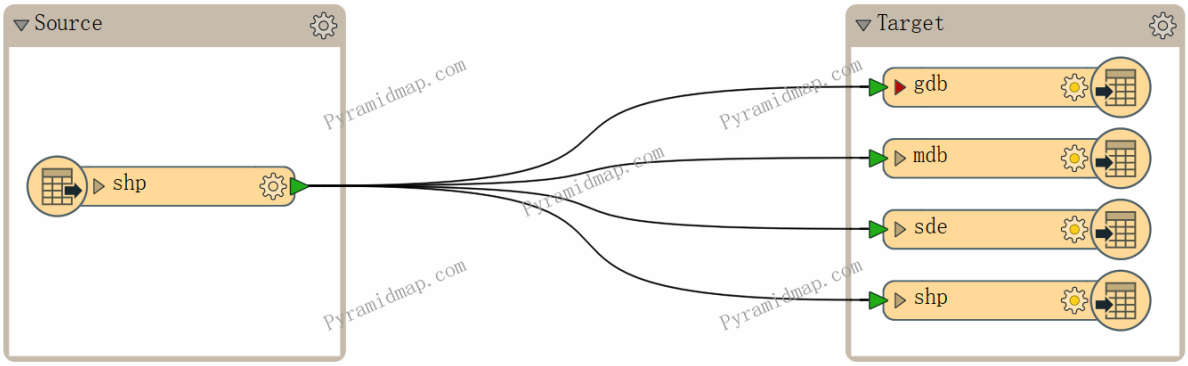


Figure 6-15: Shp feature class migration workflow model

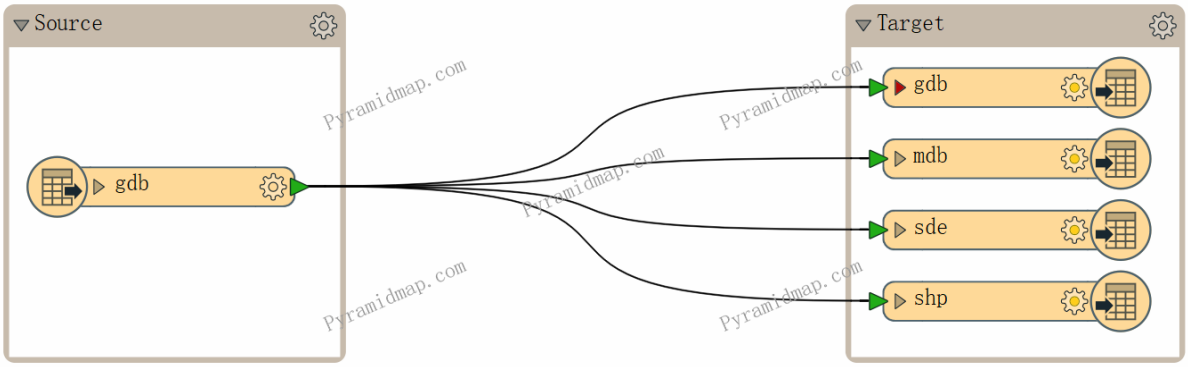


Figure 6-16: gdb feature class migration workflow model

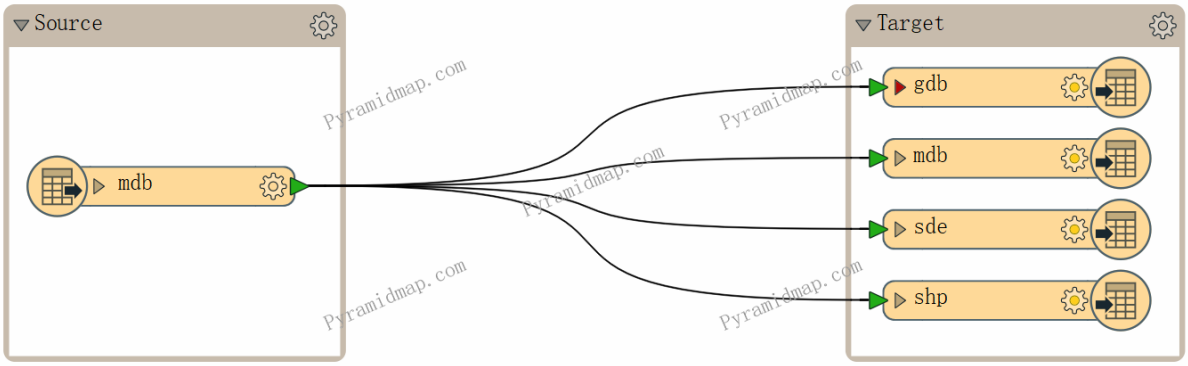


Figure 6-17: mdb feature class migration workflow model

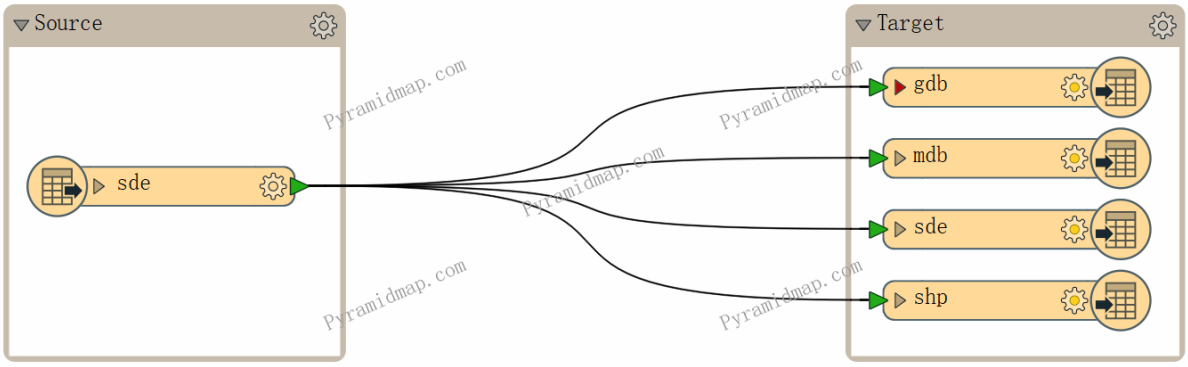


Figure 6-18: Geo database feature class migration workflow model

6.4.2: Feature class migration module

Feature class migration can migrate map features between Shp, Gdb, Mdb, enterprises, geographical database and other different physical storage locations, as shown in Figure 6-19.

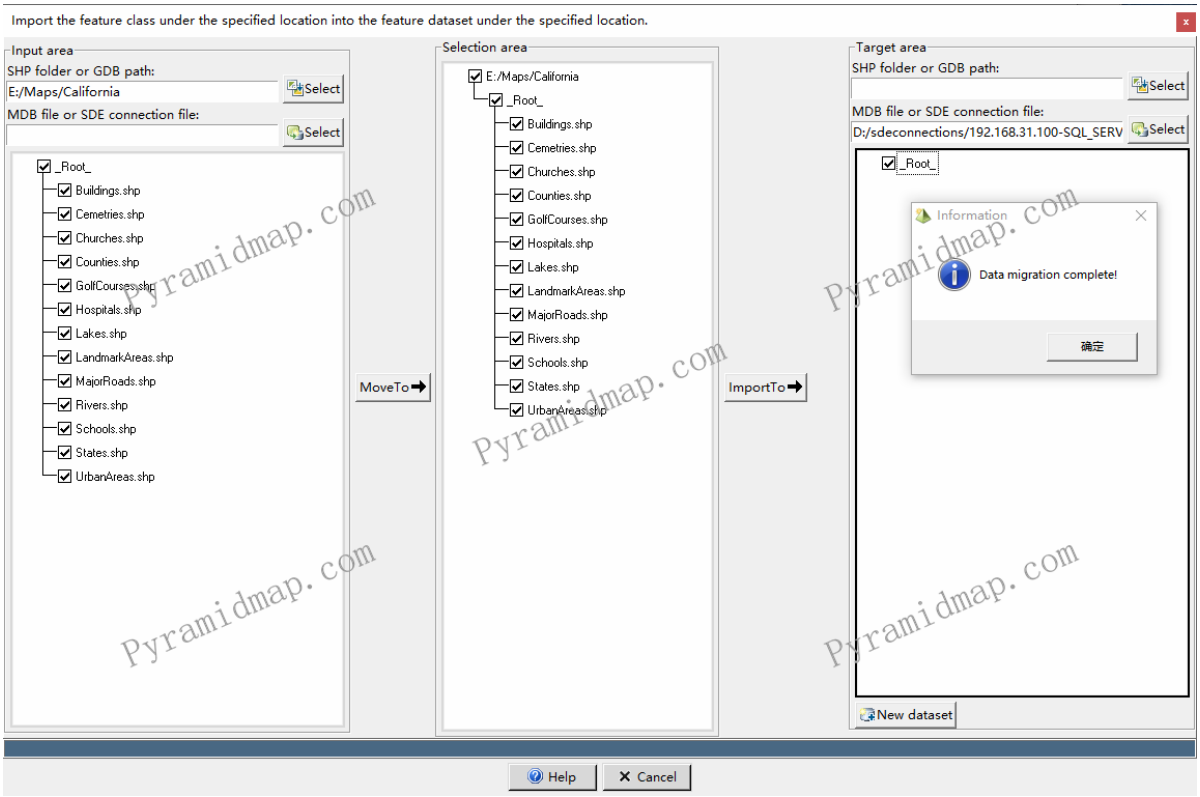


Figure 6-19: Geo database feature class migration interface

It can be understood that feature class migration, as a superset operation, has covered the operation of importing map features to GIS enterprises and geographical database. Migrating feature classes from Shp, Gdb, and Mdb to geographical database is equivalent to importing them to geographical database. Conversely, feature classes in geographical database can be transferred out to Shp, Gdb, and Mdb. PyramidMap provides cmd output monitoring of the operation process. Taking the migration of Shp layer to geographical database as an example, the output monitoring of the operation process is shown in Figure 6-20.

```
2023-04-04 23:38:47   E:/Maps/California/Buildings.shp :being imported into database, please wait.
E:/Maps/California/Buildings.shp Successfully converted:  D:/sdeconnections/192.168.31.100-SQL_SERVER-geodata.sde\Buildings_1
2023-04-04 23:38:56  Data imported successfully.
2023-04-04 23:38:56   E:/Maps/California/Cemetries.shp :being imported into database, please wait.
E:/Maps/California/Cemetries.shp Successfully converted:  D:/sdeconnections/192.168.31.100-SQL_SERVER-geodata.sde\Cemetries_1
2023-04-04 23:38:59  Data imported successfully.
2023-04-04 23:38:59   E:/Maps/California/Churches.shp :being imported into database, please wait.
E:/Maps/California/Churches.shp Successfully converted:  D:/sdeconnections/192.168.31.100-SQL_SERVER-geodata.sde\Churches_1
2023-04-04 23:40:52  Data imported successfully.
2023-04-04 23:40:52   E:/Maps/California/Counties.shp :being imported into database, please wait.
E:/Maps/California/Counties.shp Successfully converted:  D:/sdeconnections/192.168.31.100-SQL_SERVER-geodata.sde\Counties_1
2023-04-04 23:40:56  Data imported successfully.
2023-04-04 23:40:56   E:/Maps/California/GolfCourses.shp :being imported into database, please wait.
E:/Maps/California/GolfCourses.shp Successfully converted:  D:/sdeconnections/192.168.31.100-SQL_SERVER-geodata.sde\GolfCourses_1
2023-04-04 23:41:00  Data imported successfully.
2023-04-04 23:41:00   E:/Maps/California/Hospitals.shp :being imported into database, please wait.
E:/Maps/California/Hospitals.shp Successfully converted:  D:/sdeconnections/192.168.31.100-SQL_SERVER-geodata.sde\Hospitals_1
2023-04-04 23:41:03  Data imported successfully.
2023-04-04 23:41:03   E:/Maps/California/Lakes.shp :being imported into database, please wait.
E:/Maps/California/Lakes.shp Successfully converted:  D:/sdeconnections/192.168.31.100-SQL_SERVER-geodata.sde\Lakes_1
2023-04-04 23:41:06  Data imported successfully.
2023-04-04 23:41:06   E:/Maps/California/LandmarkAreas.shp :being imported into database, please wait.
E:/Maps/California/LandmarkAreas.shp Successfully converted:  D:/sdeconnections/192.168.31.100-SQL_SERVER-geodata.sde\LandmarkAreas_1
2023-04-04 23:41:17  Data imported successfully.
2023-04-04 23:41:17   E:/Maps/California/MajorRoads.shp :being imported into database, please wait.
E:/Maps/California/MajorRoads.shp Successfully converted:  D:/sdeconnections/192.168.31.100-SQL_SERVER-geodata.sde\MajorRoads_1
2023-04-04 23:41:58  Data imported successfully.
2023-04-04 23:41:59   E:/Maps/California/Rivers.shp :being imported into database, please wait.
E:/Maps/California/Rivers.shp Successfully converted:  D:/sdeconnections/192.168.31.100-SQL_SERVER-geodata.sde\Rivers_1
2023-04-04 23:42:02  Data imported successfully.
2023-04-04 23:42:02   E:/Maps/California/Schools.shp :being imported into database, please wait.
E:/Maps/California/Schools.shp Successfully converted:  D:/sdeconnections/192.168.31.100-SQL_SERVER-geodata.sde\Schools_1
2023-04-04 23:42:09  Data imported successfully.
2023-04-04 23:42:09   E:/Maps/California/States.shp :being imported into database, please wait.
E:/Maps/California/States.shp Successfully converted:  D:/sdeconnections/192.168.31.100-SQL_SERVER-geodata.sde\States
2023-04-04 23:42:17  Data imported successfully.
2023-04-04 23:42:17   E:/Maps/California/UrbanAreas.shp :being imported into database, please wait.
E:/Maps/California/UrbanAreas.shp Successfully converted:  D:/sdeconnections/192.168.31.100-SQL_SERVER-geodata.sde\UrbanAreas_1
2023-04-04 23:42:20  Data imported successfully.
++++++++INFO:Data warehousing log generation completed++++++++
```

Figure 6-20: The cmd monitoring output of feature class migrating in PyramidMap

At the end of the migration process, the feature data is successfully imported into the geographical database. Taking the Hospitals layer as an example, the effect of opening it in ArcMap is shown in Figure 6-21.

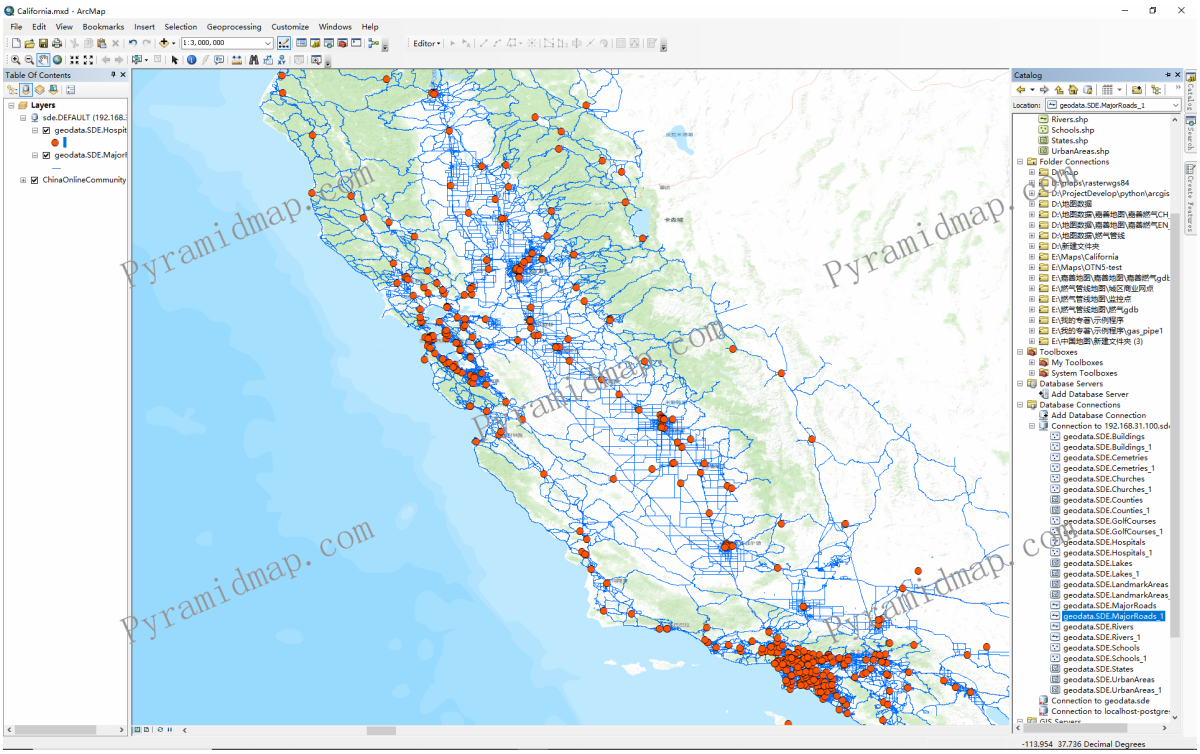
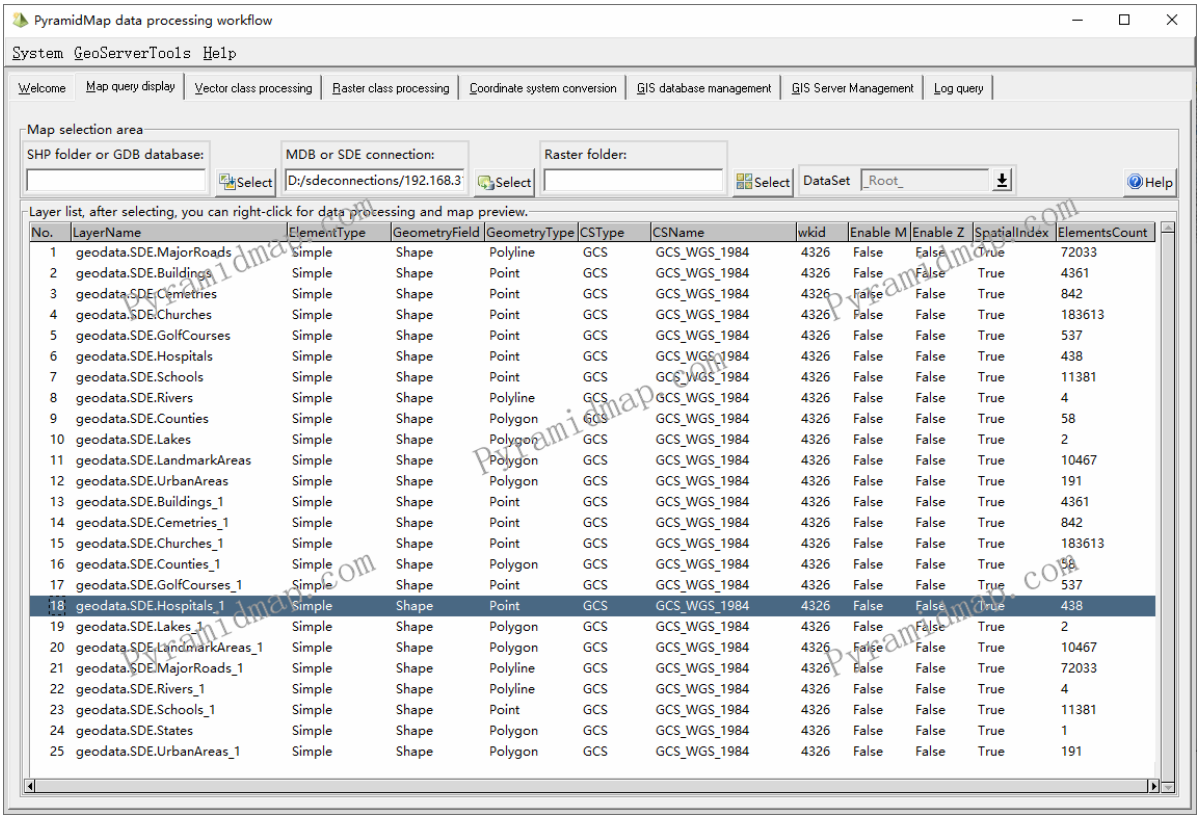


Figure 6-21: Open the feature class layer migrated to geographical database in ArcMap

Perform the same operation in PyramidMap to obtain the geographical database layer, as shown in Figure 6-22.



PyramidMap data processing workflow

System GeoServerTools Help

Welcome | Map query display | Vector class processing | Raster class processing | Coordinate system conversion | GIS database management | GIS Server Management | Log query

Map selection area

SHP folder or GDB database: Select | MDB or SDE connection: D:/sdeconnections/192.168.3 Select | Raster folder: Select | DataSet Root_ | Help

Layer list, after selecting, you can right-click for data processing and map preview.

No.	LayerName	ElementType	GeometryField	GeometryType	CSType	CSName	wkid	Enable M	Enable Z	SpatialIndex	ElementsCount
1	geodata.SDE.MajorRoads	Simple	Shape	Polyline	GCS	GCS_WGS_1984	4326	False	False	True	72033
2	geodata.SDE.Buildings	Simple	Shape	Point	GCS	GCS_WGS_1984	4326	False	False	True	4361
3	geodata.SDE.Cemetries	Simple	Shape	Point	GCS	GCS_WGS_1984	4326	False	False	True	842
4	geodata.SDE.Churches	Simple	Shape	Point	GCS	GCS_WGS_1984	4326	False	False	True	183613
5	geodata.SDE.GolfCourses	Simple	Shape	Point	GCS	GCS_WGS_1984	4326	False	False	True	537
6	geodata.SDE.Hospitals	Simple	Shape	Point	GCS	GCS_WGS_1984	4326	False	False	True	438
7	geodata.SDE.Schools	Simple	Shape	Point	GCS	GCS_WGS_1984	4326	False	False	True	11381
8	geodata.SDE.Rivers	Simple	Shape	Polyline	GCS	GCS_WGS_1984	4326	False	False	True	4
9	geodata.SDE.Counties	Simple	Shape	Polygon	GCS	GCS_WGS_1984	4326	False	False	True	58
10	geodata.SDE.Lakes	Simple	Shape	Polygon	GCS	GCS_WGS_1984	4326	False	False	True	2
11	geodata.SDE.LandmarkAreas	Simple	Shape	Polygon	GCS	GCS_WGS_1984	4326	False	False	True	10467
12	geodata.SDE.UrbanAreas	Simple	Shape	Polygon	GCS	GCS_WGS_1984	4326	False	False	True	191
13	geodata.SDE.Buildings_1	Simple	Shape	Point	GCS	GCS_WGS_1984	4326	False	False	True	4361
14	geodata.SDE.Cemetries_1	Simple	Shape	Point	GCS	GCS_WGS_1984	4326	False	False	True	842
15	geodata.SDE.Churches_1	Simple	Shape	Point	GCS	GCS_WGS_1984	4326	False	False	True	183613
16	geodata.SDE.Counties_1	Simple	Shape	Polygon	GCS	GCS_WGS_1984	4326	False	False	True	58
17	geodata.SDE.GolfCourses_1	Simple	Shape	Point	GCS	GCS_WGS_1984	4326	False	False	True	537
18	geodata.SDE.Hospitals_1	Simple	Shape	Point	GCS	GCS_WGS_1984	4326	False	False	True	438
19	geodata.SDE.Lakes_1	Simple	Shape	Polygon	GCS	GCS_WGS_1984	4326	False	False	True	2
20	geodata.SDE.LandmarkAreas_1	Simple	Shape	Polygon	GCS	GCS_WGS_1984	4326	False	False	True	10467
21	geodata.SDE.MajorRoads_1	Simple	Shape	Polyline	GCS	GCS_WGS_1984	4326	False	False	True	72033
22	geodata.SDE.Rivers_1	Simple	Shape	Polyline	GCS	GCS_WGS_1984	4326	False	False	True	4
23	geodata.SDE.Schools_1	Simple	Shape	Point	GCS	GCS_WGS_1984	4326	False	False	True	11381
24	geodata.SDE.States	Simple	Shape	Polygon	GCS	GCS_WGS_1984	4326	False	False	True	1
25	geodata.SDE.UrbanAreas_1	Simple	Shape	Polygon	GCS	GCS_WGS_1984	4326	False	False	True	191

Figure 6-22: Display the feature class layers migrated to geographical database in PyramidMap

Take the Hospitals_1 layer as an example opened in PyramidMap as shown in Figure 6-23.

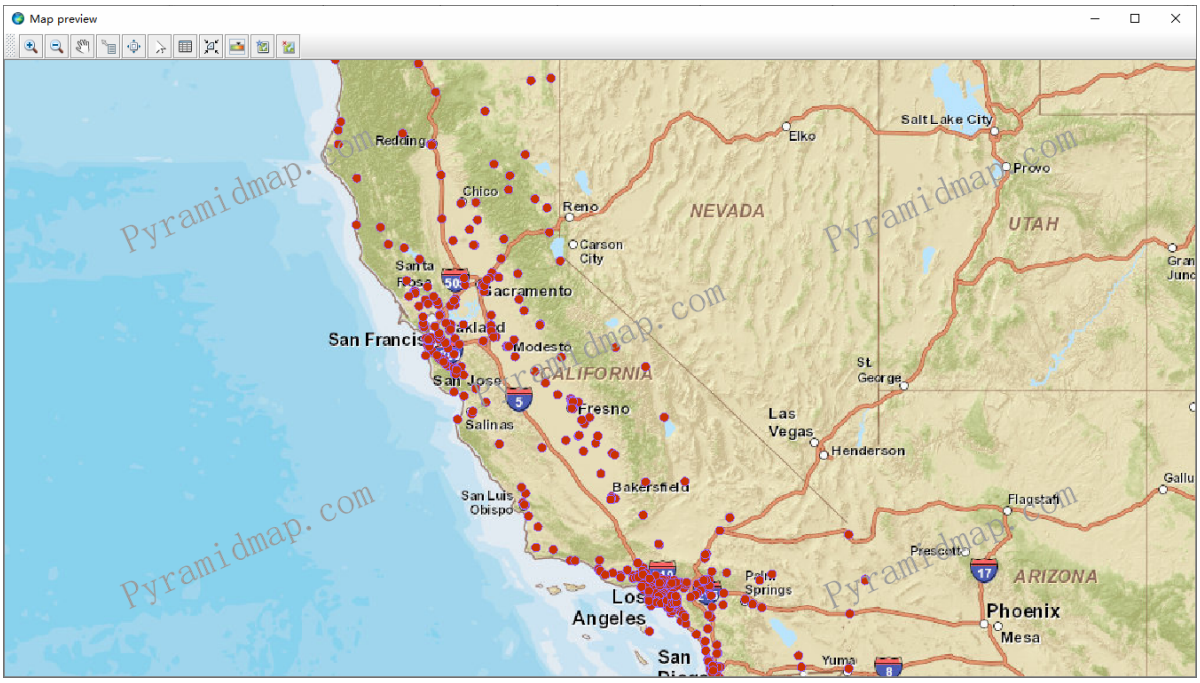


Figure 6-23: Display the feature class layers migrated to geographical database in PyramidMap

6.5: Standard coordinate system conversion

Supports the mutual conversion between all geographic and projection coordinate systems with EPSG standard definition. The source data in the conversion queue can have different coordinate systems, and can be converted to the specified geographic or projected coordinate system. The transformation model of coordinate system is as follows (only some representative geographical and projection coordinate systems are selected as examples), and other coordinate systems are treated in the same way as shown in Figure 6-24.

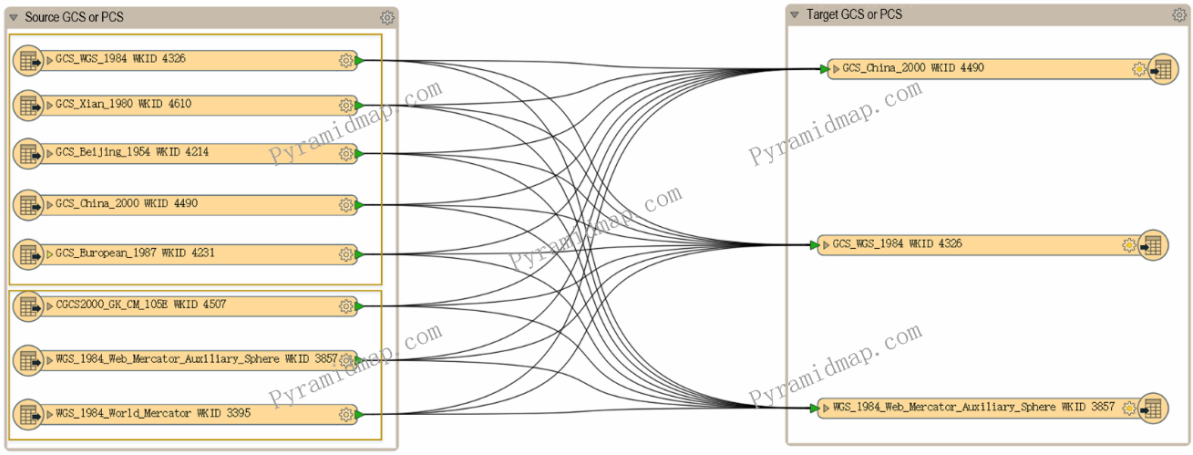


Figure 6-24: PyramidMap Standard coordinate system conversion model

Interface description of standard coordinate system transformation as shown in Figure 6-25.

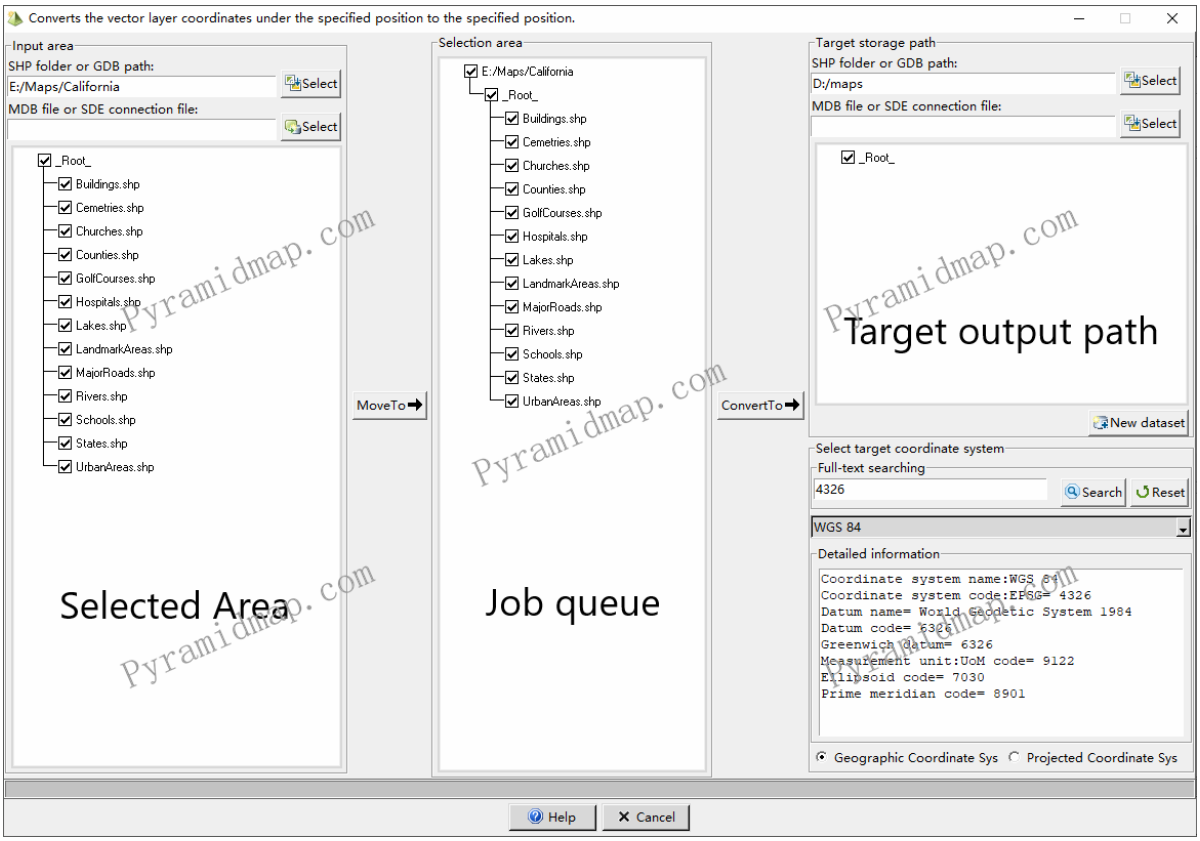


Figure 6-25: PyramidMap Standard coordinate system conversion interface

Select the target coordinate system and data converted output path in the target area. The target coordinate system supports GCS spatial geographic coordinate system and PCS plane projection coordinate system, which can be selected from the drop-down list, and supports fuzzy queries through full-text retrieval. The target output path supports Shp, GDB/MDB file types geographical database, Geodatabase enterprise geographical database, such as Oracle, SQL Server, Postgre, etc.

7: Raster processing

Raster processing in PyramidMap includes image compression, image segmentation, image merging, image conversion, noData value processing, and constructing image pyramids. The menu interface is shown in Figure 7-1.

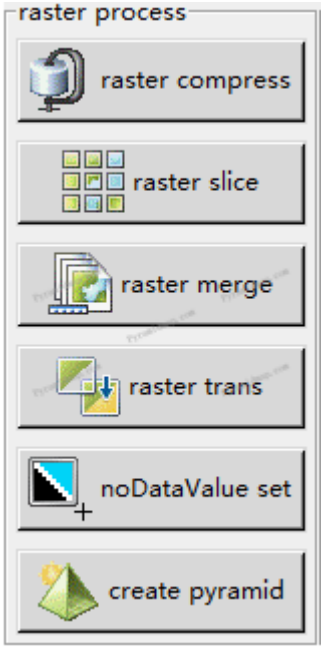


Figure 7-1: PyramidMap image data processing operation

7.1: Raster compression

The main advantage of compressing for image is that the compressed data requires less storage space, and because less information is transmitted, it accelerates the display of the data. The PyramidMap image compression interface is shown in Figure 7-2.

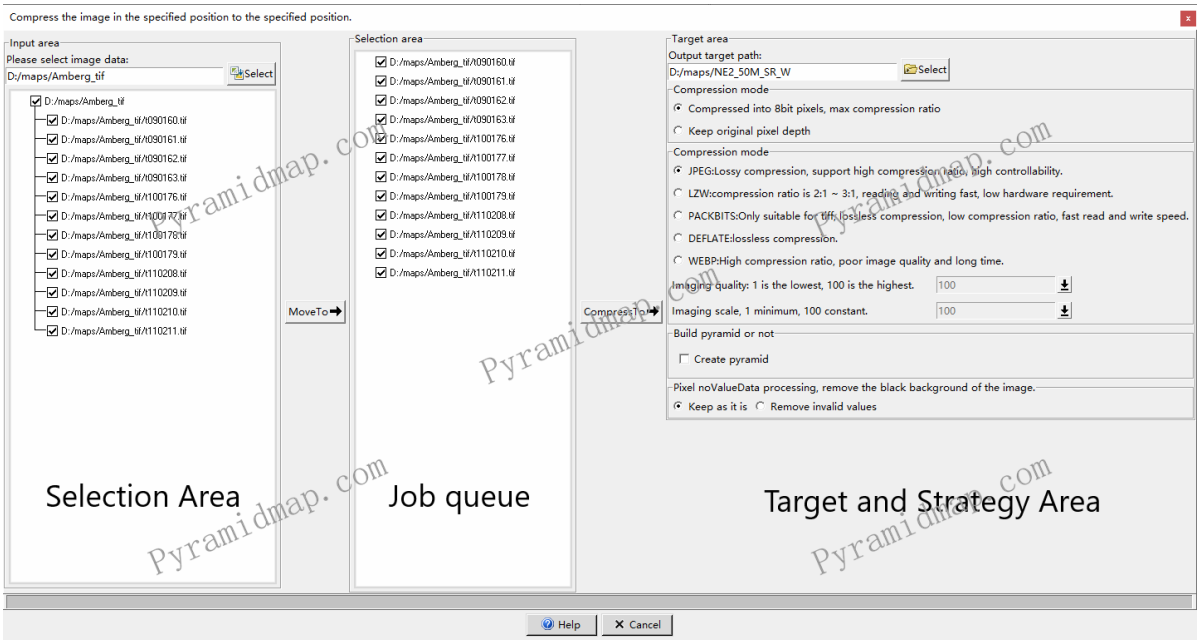


Figure 7-2: PyramidMap raster compressing operation

PyramidMap supports configuring the same compression strategy for batch data, achieving one click completion. PyramidMap configures image compression strategies through parameters:

- Compression depth: By compressing the pixel depth of the original image to 8 bits, the maximum compression ratio can be achieved, but at the cost of reducing the image color scale, grayscale, and image quality. The original pixel depth can be retained.
- Compression method: The data compression method can be lossy compression (JPEG/JPEG 2000, WEBP) or lossless compression (LZW, PackBits, and DEFLATE). Lossless compression means that the pixel values in the raster dataset will not change or be lost. If the pixel values of the raster dataset will be used for analyzing or deriving other data products, lossless compression or no compression should be chosen. The amount of compression depends on the data and compression quality. The more similar data, the higher the compression ratio. The lower the compression quality, the higher the compression ratio. Compared to lossless compression, lossy compression usually produces a higher compression ratio.
- Imaging quality: When selecting the **JPEG** compression method, the compression quality must be specified. The effective range of compression quality values is 1 to 100, with 75 being the default value.
- Imaging Scale: Specify the width/height ratio of the compressed image to the original image.
- Build a pyramid: Option whether to build a pyramid structure on the compressed image. Building a pyramid can perform hierarchical indexing on the image, which will accelerate the image loading speed. This is very effective in improving the loading and display speed of big data images.

- Pixel noData processing: Remove unnecessary values created around raster data. The specified value is different from other useful data in the raster dataset. For example, values with zero on the grid boundary are different from zero values within the grid dataset. The specified pixel value will be set to NoData in the output grid dataset. For file based grids, to **ignore the background value**, it must be set to the same value as **NoData**. Invalid values in the image can cause a black background. Choosing the invalid value processing option will automatically filter out invalid pixels during the compression process, thereby improving image quality and simplifying the processing process.

  The processing process will save the log, and a message prompt will be given after the processing is completed.

7.2: Raster tile

It is common to encounter scenarios where large amounts of satellite or aerial imagery are processed in map application scenarios, such as performing pixel operations. With the improvement of imaging quality and resolution, a remote sensing image can easily reach tens, hundreds G, or even T-levels, which poses a great challenge to computer performance. Even professional image workstations incur significant performance overhead. More importantly, the time cost consumed for this is something that all users are unwilling to accept. Therefore, strategic segmentation of images with large amounts of data, maintaining or converting the original values, data types, band numbers, and projections, is an important way to accelerate image processing. PyramidMap provides automated batch processing technology for this, as shown in Figure 7-3.

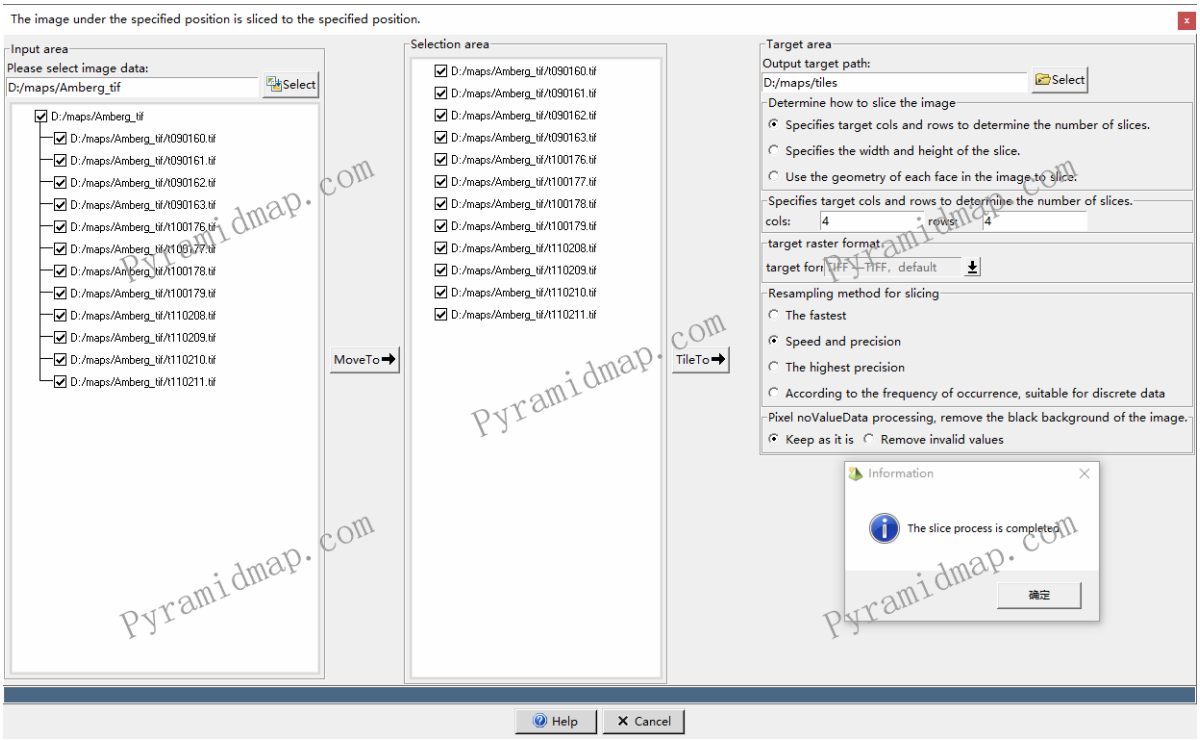


Figure 7-3: PyramidMap raster tiling interface

PyramidMap supports image slicing strategies with parameter configuration methods, as shown in Figure 7-4.

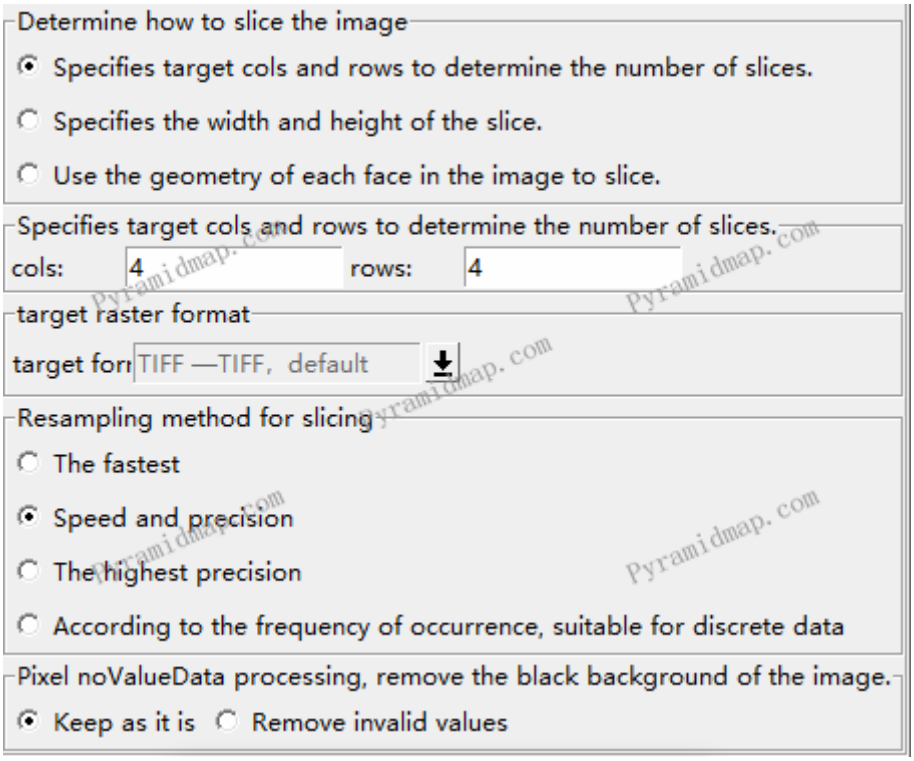


Figure 7-4: PyramidMap raster slicing strategy

The slicing method and quantity are determined by the number of sliced columns or sliced pixels, and the strategy area provides options for this. The output slice file supports multiple formats. The specific format of the filename extension is as follows:

- Esri BIL is .bil
- Esri BIP is .bip
- BMP is .bmp
- Esri BSQ is .bsq
- ENVI DAT is .dat
- GIF is .gif
- ERDAS IMAGINE is .img
- JPEG is .jpg
- JPEG 2000 is .jp2
- PNG is .png
- TIFF is .tif
- MRF is .mrf
- CRF is .crf
- Esri Grid has no extend name

The PyramidMap slicing strategy balances speed and accuracy, providing four resampling options to meet users' different needs for segmentation speed and accuracy. During the slicing process, it is possible to choose whether to perform invalid value processing on the segmented image to avoid the occurrence of black invalid pixels. If the segmented source image itself has already filtered out invalid values, it can be kept as is and not processed, depending on the predetermined state of invalid values in the source data. PyramidMap provides cmd output monitoring of the running process, as shown in Figure 7-5.

```
PyramidMap has been started and in running >>>>>>
++++++++Please wait while the feature class data is read.++++++++
++++++++Please wait while the feature class data is read.++++++++
2023-04-05 09:49:26   D:/maps/Amberg_tif/t090160.tif   being in sliced, please wait.
2023-04-05 09:49:58  being sliced successful.
2023-04-05 09:49:58   D:/maps/Amberg_tif/t090161.tif   being in sliced, please wait.
2023-04-05 09:50:30  being sliced successful.
2023-04-05 09:50:30   D:/maps/Amberg_tif/t090162.tif   being in sliced, please wait.
2023-04-05 09:50:56  being sliced successful.
2023-04-05 09:50:56   D:/maps/Amberg_tif/t090163.tif   being in sliced, please wait.
2023-04-05 09:51:18  being sliced successful.
2023-04-05 09:51:18   D:/maps/Amberg_tif/t100176.tif   being in sliced, please wait.
2023-04-05 09:51:43  being sliced successful.
2023-04-05 09:51:43   D:/maps/Amberg_tif/t100177.tif   being in sliced, please wait.
2023-04-05 09:52:08  being sliced successful.
2023-04-05 09:52:08   D:/maps/Amberg_tif/t100178.tif   being in sliced, please wait.
2023-04-05 09:52:33  being sliced successful.
2023-04-05 09:52:33   D:/maps/Amberg_tif/t100179.tif   being in sliced, please wait.
2023-04-05 09:53:00  being sliced successful.
2023-04-05 09:53:00   D:/maps/Amberg_tif/t110208.tif   being in sliced, please wait.
2023-04-05 09:53:25  being sliced successful.
2023-04-05 09:53:25   D:/maps/Amberg_tif/t110209.tif   being in sliced, please wait.
2023-04-05 09:53:52  being sliced successful.
2023-04-05 09:53:52   D:/maps/Amberg_tif/t110210.tif   being in sliced, please wait.
2023-04-05 09:54:20  being sliced successful.
2023-04-05 09:54:20   D:/maps/Amberg_tif/t110211.tif   being in sliced, please wait.
2023-04-05 09:54:47  being sliced successful.
++++++++INFO:raster slice log generation completed++++++++
```

Figure 7-5: PyramidMap output monitoring of tile process in cmd window

The output tiles saved in the target path, as shown in Figure 7-6.

Figure 7-6: Image tiles saved to the target path

The image tiling model and quality analysis are shown in Figures 7-7.

Figure 7-7: PyramidMap image slicing model and quality analysis

In summary, PyramidMap's slicing image processing technology can help you slice map images according to preset strategies, and make a trade-off between processing speed and imaging quality. Based on this, it supports the processing of invalid pixel values (noise) and pyramid hierarchy construction, thereby improving imaging quality and loading display speed, especially suitable for processing large data volume satellite or aerial remote sensing images.

7.3: Raster merge

Image merging and segmentation is an inverse process. PyramidMap can synthesize equal slices belonging to the same coordinate system into a complete image, as shown in Figure 7-8.

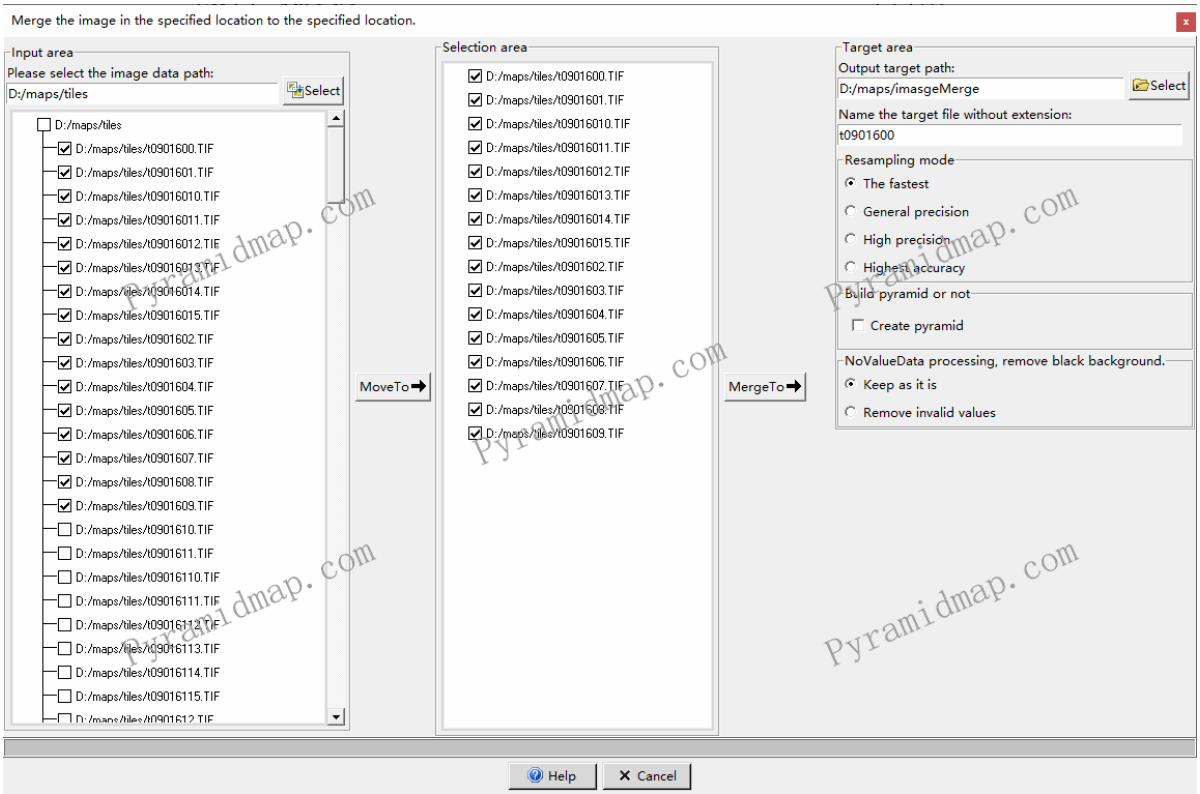


Figure 7-8: PyramidMap raster merging interface

The merge strategy is controlled by target area parameters, and users can construct merge modes as needed through speed and accuracy adaptation. PyramidMap supports invalid value processing and pyramid construction in the image merging process, thereby eliminating black edges and backgrounds in the image. If the source image has already filtered out invalid values, it can be kept as is and not processed, depending on the pre specified metadata status. Building a pyramid can help improve the loading speed of images and can be optionally executed. The image merging model and quality analysis are shown in Figures 7-9.

Figure 7-9: PyramidMap raster merging model and quality analysis

In summary, image segmentation and merging are a reciprocal process. In real production environments, remote sensing image segmentation and merging are common application scenarios, such as land ownership confirmation, natural disaster distribution, and natural resource regulation. As long as large-scale image data processing is involved, image segmentation and merging are almost always necessary steps.

7.4: Raster conversion

In many cases, due to the need for data analysis of raster images, data conversion is required, including data type, pixel depth, color mapping table and band, multi-dimensional data processing, etc. PyramidMap supports batch conversion of raster images according to preset strategies, and generates a data processing workflow with one click, as shown in Figure 7-10.

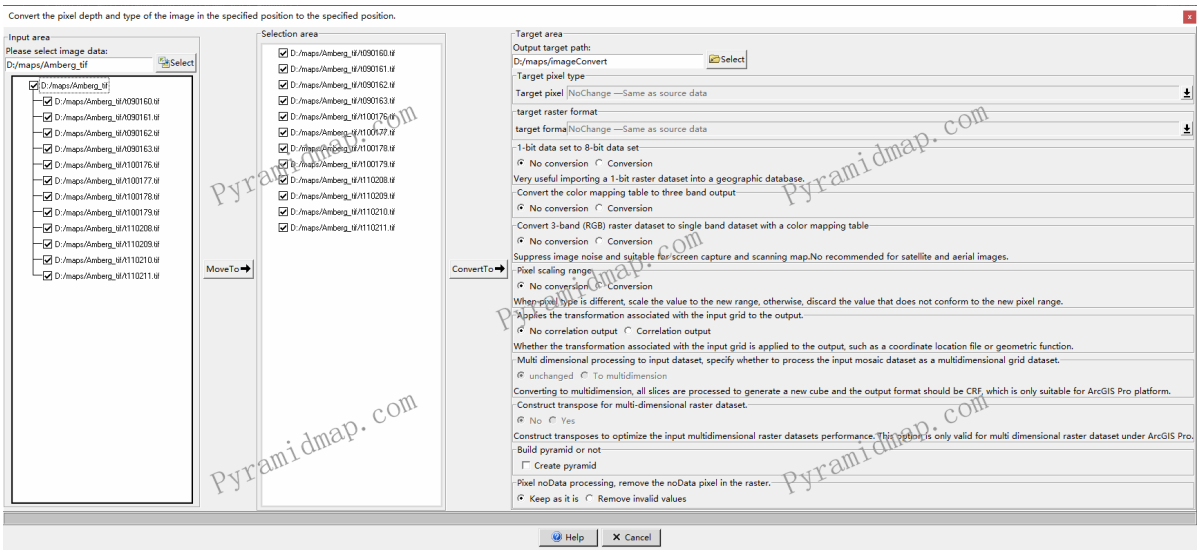


Figure 7-10: PyramidMap raster data conversion interface

The file types supported by this method include: Raster Dataset; Mosaic Dataset; Mosaic Layer; Raster Layer; File; Image Service, which can save the output results as BIL, BIP, BMP, BSQ, CRF, DAT, Esri Grid, GIF, IMG, JPEG, JPEG 2000, MRF, PNG, TIFF format or any geographical database grid dataset. PyramidMap supports a combined image data conversion strategy and sets the target pixel type, as shown in Figure 7-11.

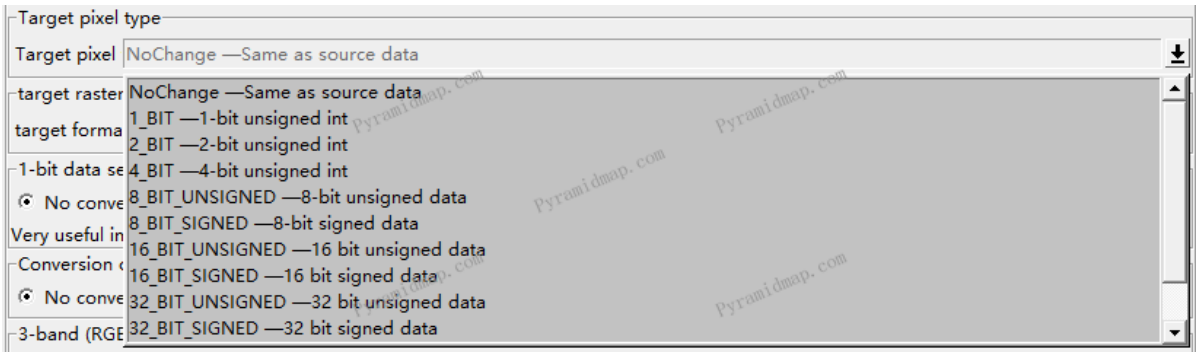


Figure 7-11: PyramidMap sets the target pixel type for image conversion

This tool can be used to scale pixel types from one bit depth to another. When scaling pixel depth, the grid displays the same bit depth, while the values are scaled to the specified new bit depth. Set the target data type, as shown in Figure 7-12.

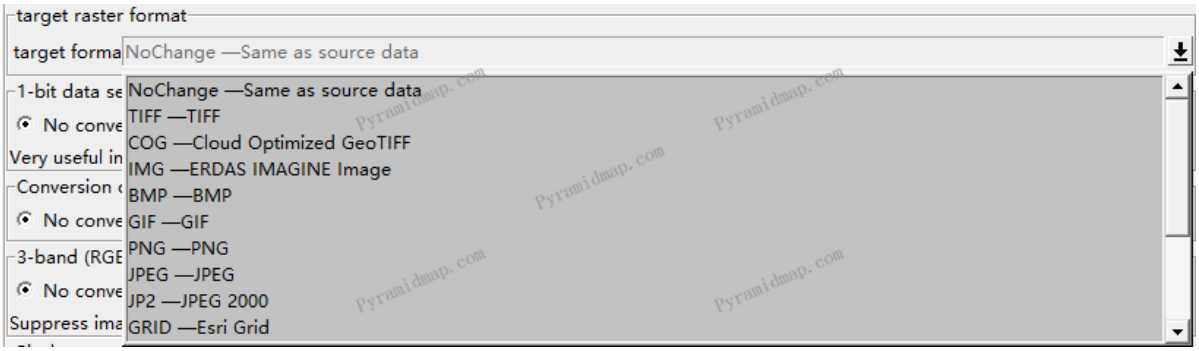


Figure 7-12: PyramidMap sets the target format type for image conversion

The specific format of the filename extension is as follows:

- Esri BIL is .bil
- Esri BIP is .bip
- BMP is .bmp
- Esri BSQ is .bsq
- ENVI DAT is .dat
- GIF is .gif
- ERDAS IMAGINE is .img
- JPEG is .jpg
- JPEG 2000 is .jp2
- PNG is .png
- TIFF is .tif
- MRF is .mrf
- CRF is .crf

- Esri Grid has no extend name

Set the target conversion mode, as shown in Figure 7-13.

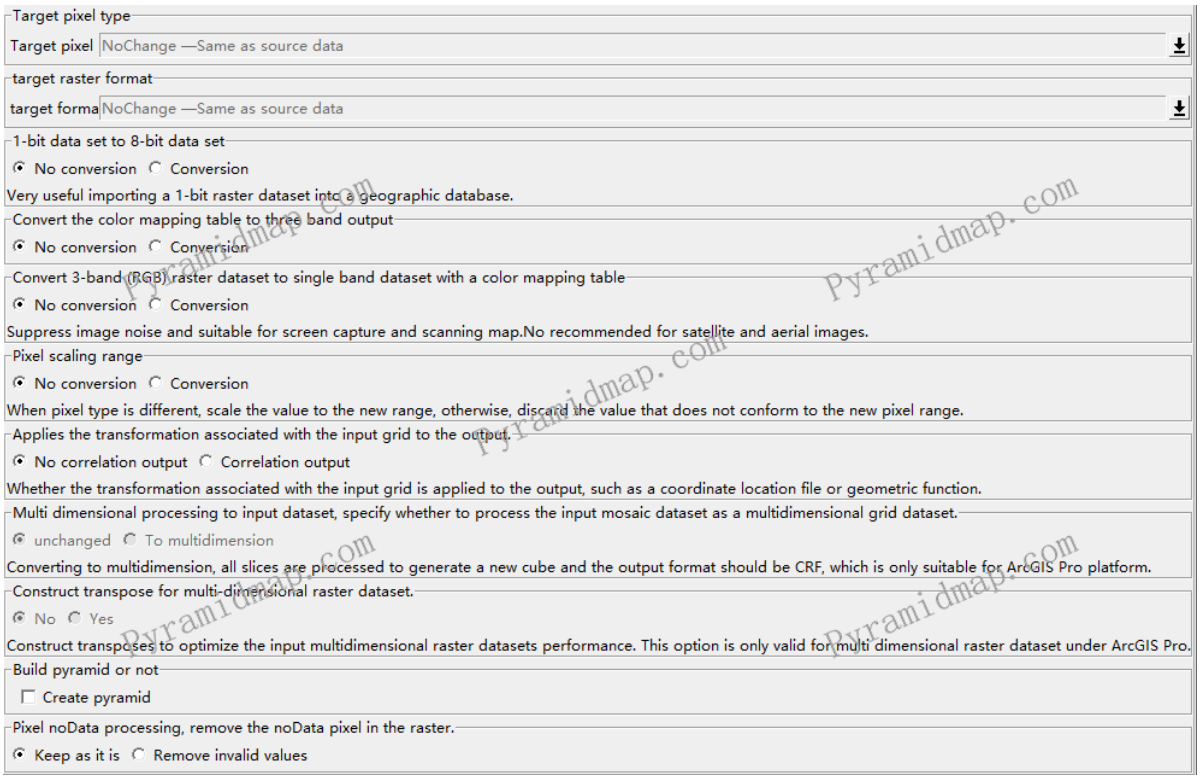


Figure 7-13: PyramidMap sets the target format type for image conversion

Grid images will be automatically converted and processed according to the above configuration strategy.

- 1-bit dataset to 8-bit dataset: This ensures that a suitable raster dataset is generated in the display, making the display appear unattractive.
- Convert the color mapping table to three band output: achieve embedding grids containing different color mapping tables, and enhance the data representation ability of image pixels.
- Convert 3-band (RGB) raster dataset to single band dataset with a color mapping table: Suppressing noise that often appears in scanned images, this is very suitable for screen capture, scanned maps, or scanned documents. However, it is not recommended to use it for satellite, aerial imagery, or thematic raster data.
- Pixel scaling range: When the output pixel type is different from the input pixel type (such as from 16 bits to 8 bits), you can choose to scale the value to fit the new range; Otherwise, values that do not match the new pixel range will be discarded.
- Apply the transformation associated with the input grid to the output: Specify whether to transform the file associated with the input grid, which is not included in the input grid, such as a grid coordinate system definition file or geometric function.
- Multi dimensional processing to input dataset: Specify whether to process the input mosaic dataset as a multidimensional grid dataset.

  Unchanged: The input will not be processed as a multidimensional grid dataset. If the input is multidimensional, only the currently displayed cutting piece will be processed. This is the default setting.

  To multidimensional:Converting to multidimension, all slices are processed to generate a new cube and the output format should be CRF, which is only suitable for ArcGIS Pro platform.

- Construct transpose for multi-dimensional raster dataset:Specify whether to construct transposes for the input multidimensional grid dataset, which segments the data along each dimension to optimize performance when accessing pixel values for all slices. 此选项只适用于ArcGIS Pro。

  No:No transpose will be constructed. This is the default setting.

  Yes:Construct transposes to optimize the input multidimensional raster datasets performance. This option is only valid for multi dimensional raster dataset under ArcGIS Pro.

- Build a pyramid or not: After data conversion, whether to construct a pyramid structure for the image.
- Pixel noData processing: Remove unnecessary values created around raster data. The specified value is different from other useful data in the raster dataset. For example, values with zero on the grid boundary are different from zero values within the grid dataset. The specified pixel value will be set to NoData in the output grid dataset. For file based grids, to ignore the background value, it must be set to the same value as NoData.

7.5: Raster coordinate system transformation

Similar to vector layer coordinate system conversion, image classes with standard coordinate system definitions can be converted between different coordinate systems. PyramidMap supports batch processing of raster image coordinate system conversion, and the operation process interface is shown in Figure 7-14.

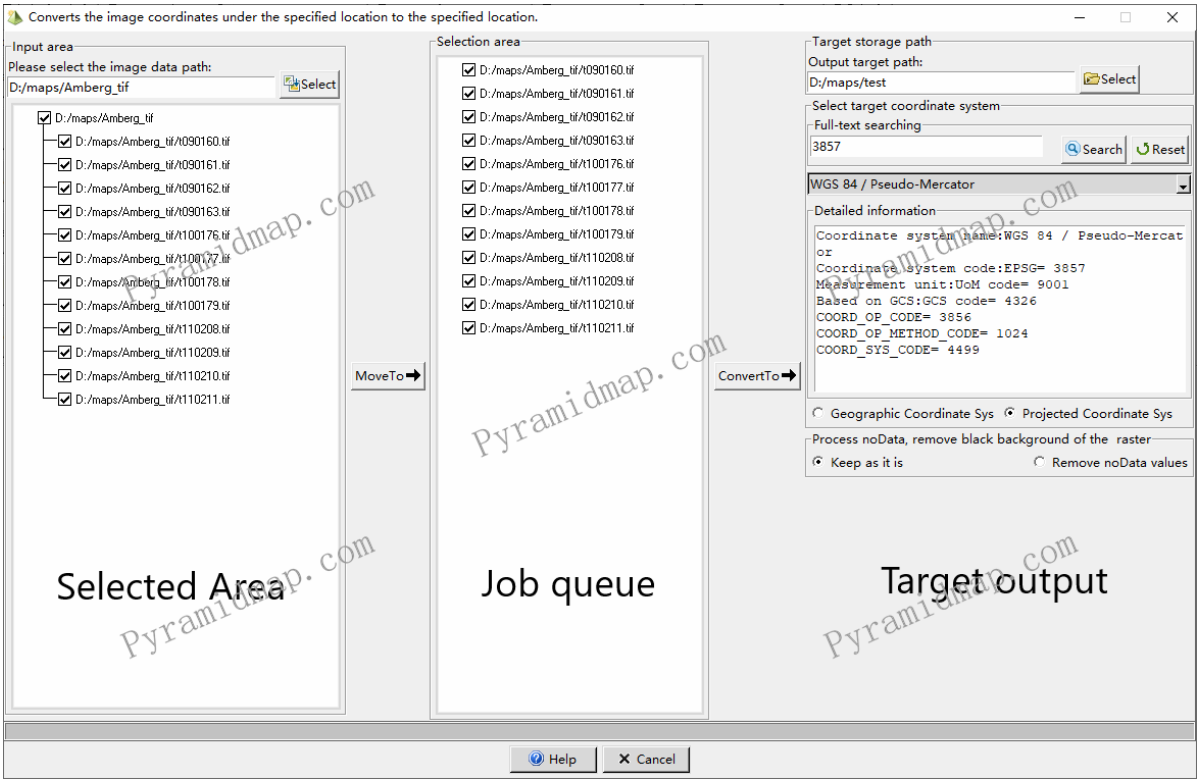


Figure 7-14: PyramidMap raster coordinate system conversion

Select the target coordinate system and data converted output path in the target area. The target coordinate system supports GCS spatial geographic coordinate system and PCS plane projection coordinate system, which can be selected from the drop-down list, and supports fuzzy query through full-text retrieval. The target storage supports folder paths.It can be converted between different coordinate systems. The program provides the conversion process processing to realize batch workflow.

7.6: Raster noData processing

When processing raster images, it is sometimes found that after classification and coloring of raster data, there are invalid pixels in certain areas, forming noise points. The presence of noise points can cause black edges or other invalid coloring of the image, as shown in Figure 7-15.

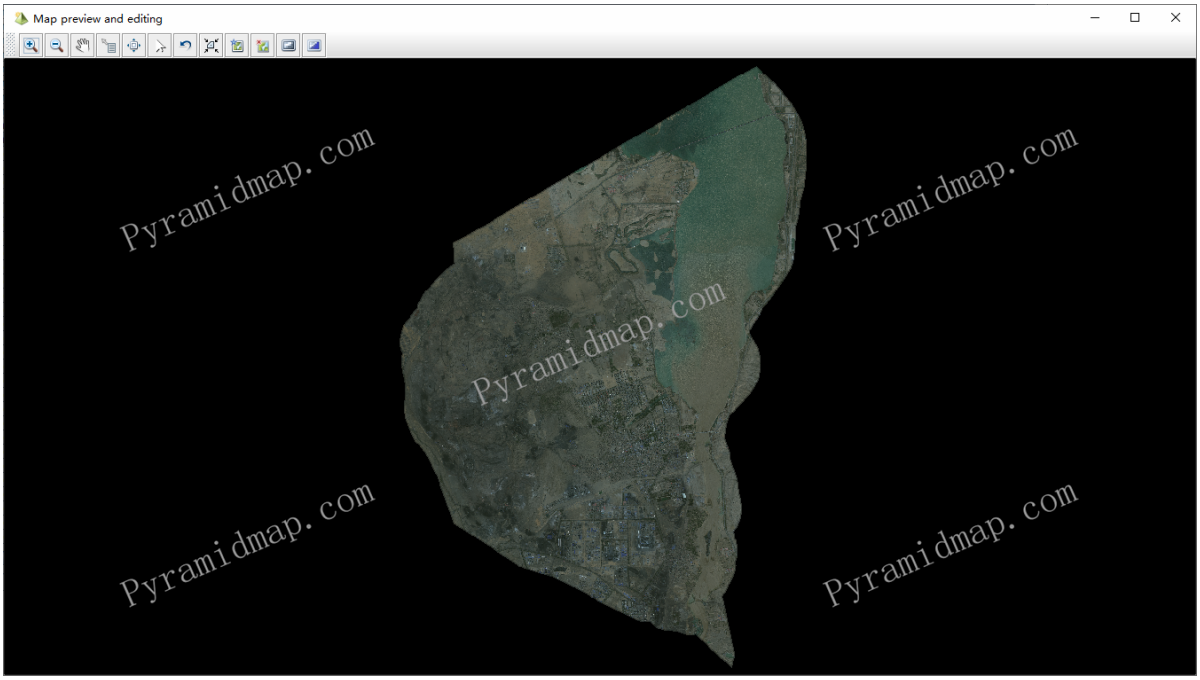


Figure 7-15: The noData values causes the black edge of the image

We process pixels in NoData area through code, and the operation interface flow is shown in Figure 7-16.

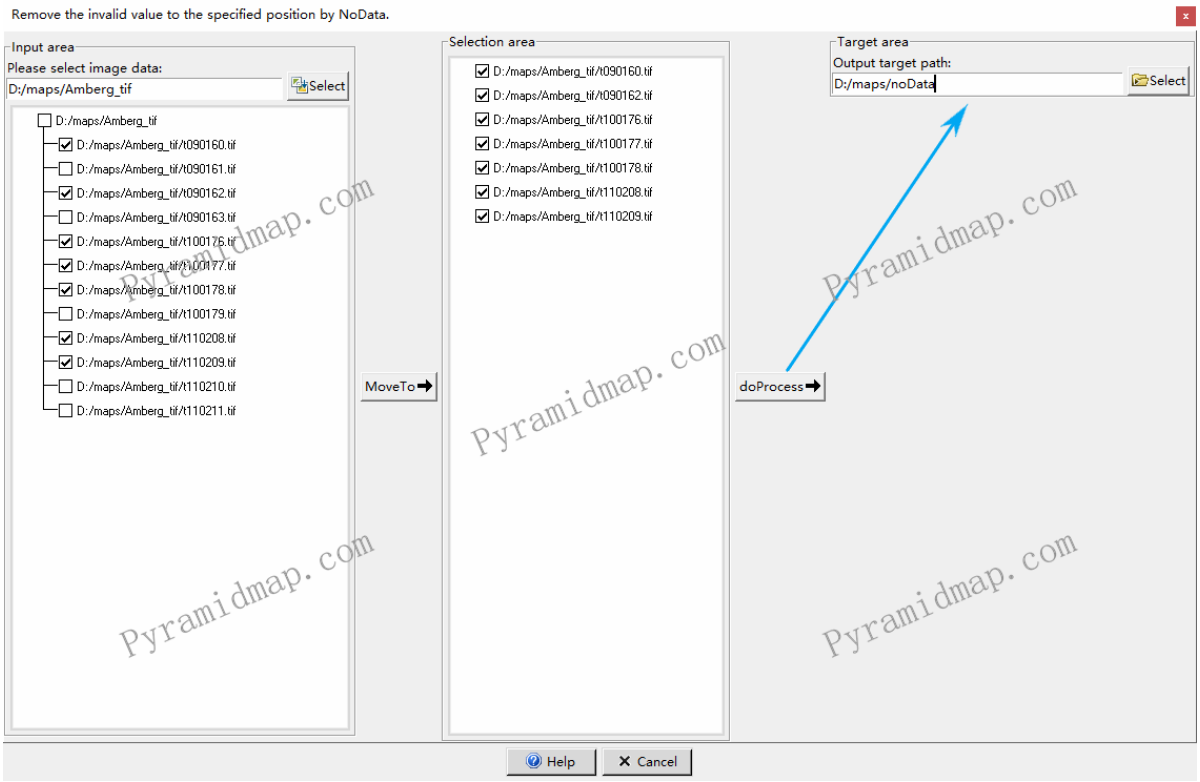


Figure 7-16: PyramidMap interface for noData value processing of raster

The processed results are shown in Figure 7-17.

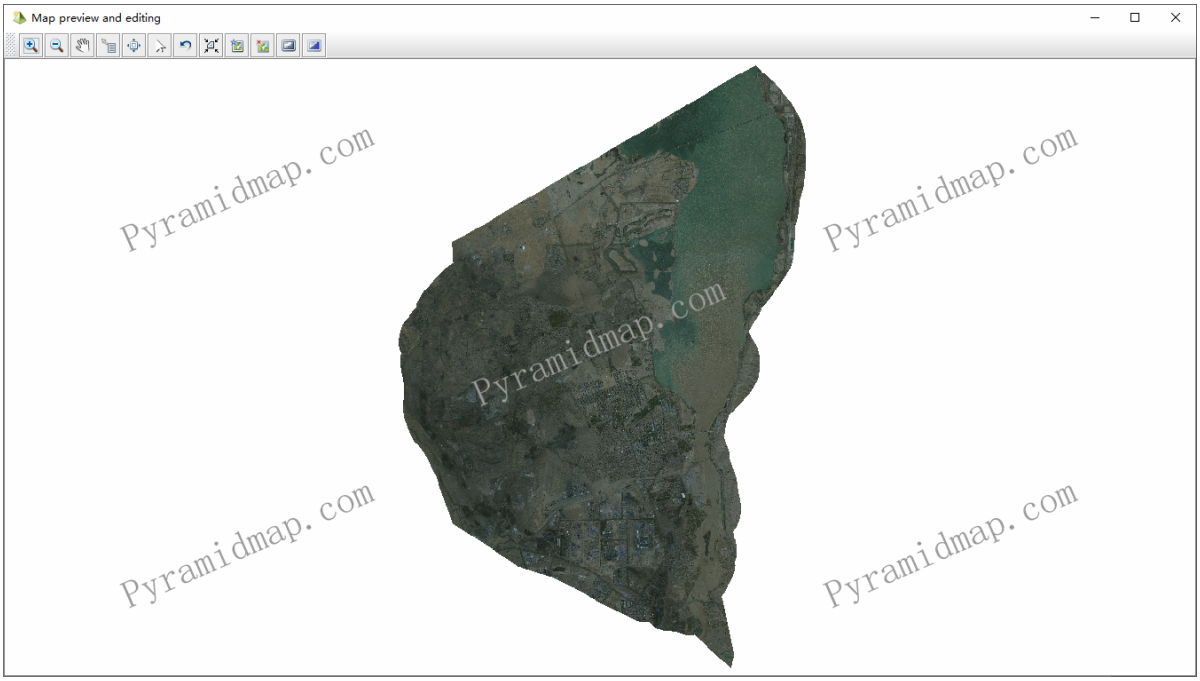


Figure 7-17: Raster being displayed after noData value processing

7.7: Raster pyramid building

Building a pyramid can improve the display performance of raster datasets and accelerate image display speed. The image pyramid is a set of images with different resolutions from fine to coarse generated by the original image according to certain rules. The pyramid structure of the image data is shown in Figure 7-18.

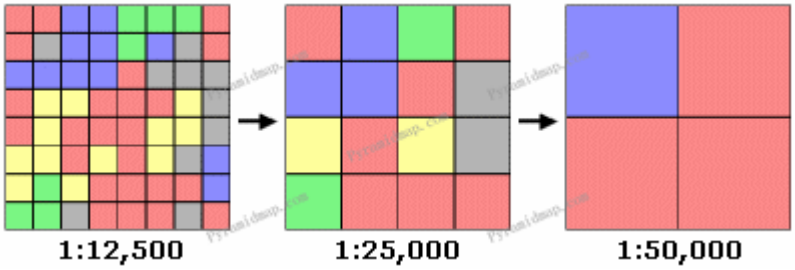


Figure 7-18: Schematic diagram of pyramid structure of a raster

Grid data with different resolutions have a vertical organization relationship from top to bottom: the closer it is to the top layer, the smaller the resolution of the data and the smaller the amount of data. Pyramids can accelerate the display speed of raster data by only retrieving data using the specified resolution (depending on display requirements). By using a pyramid, lower resolution data copies can be quickly displayed when drawing the entire dataset. As the magnification operation progresses, finer resolution levels will gradually be drawn, but performance will remain unchanged. You only need to build a pyramid once for each dataset, and the pyramid will be accessed every time the raster dataset is displayed. PyramidMap supports batch generation of image pyramid structures. The module to building raster pyramid interface in PyramidMap is shown in Figure 7-19.

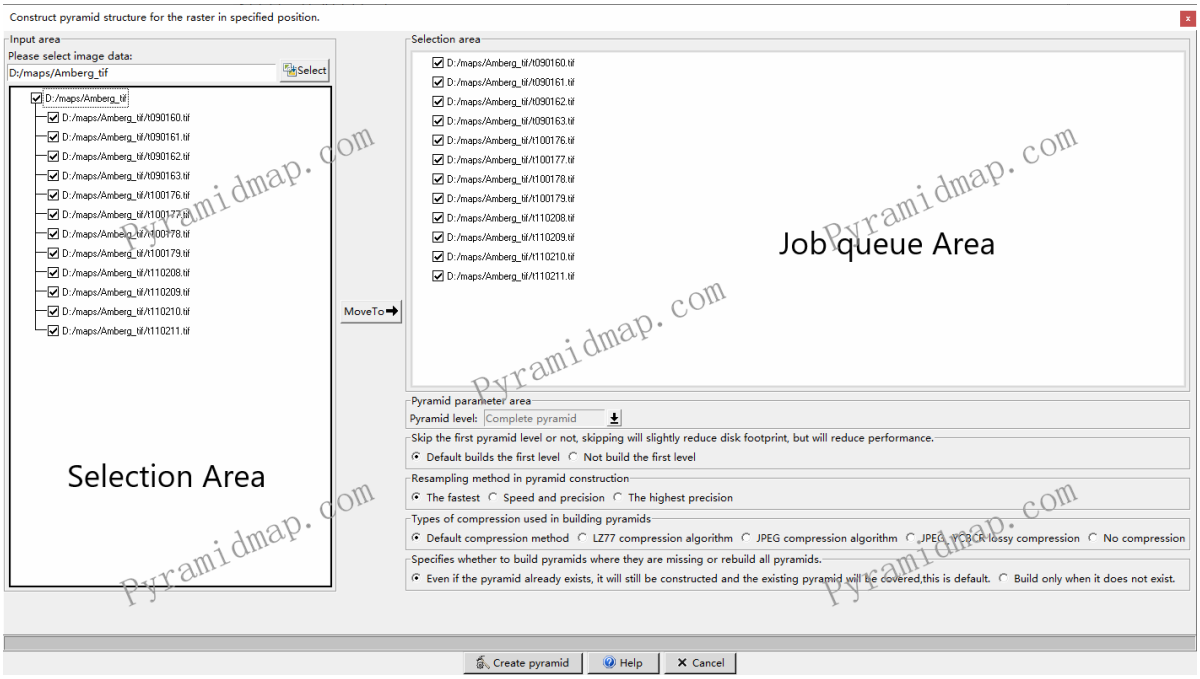


Figure 7-19: Schematic diagram of pyramid structure of a raster

The pyramid parameter area provides a construction strategy. The pyramid level specifies the pyramid level to be constructed, with a default value of **-1**. A complete pyramid will be constructed, and the maximum number of pyramid levels that can be specified is **29**. When the number of levels is set to **0**, the pyramid structure will be deleted; Skip the first pyramid level. Skipping the first level will slightly reduce the amount of disk space occupied, but will lower the performance of these proportions; The resampling method for constructing a pyramid will balance construction speed and accuracy; The compression type used when building a grid pyramid. The compression type used when constructing a grid pyramid will balance size and quality. In the default compression method, if the wavelet compression method is used to compress the source data, the JPEG compression type will be used to construct a pyramid; Otherwise, **LZ77** will be used, which can be used for any data type. JPEG - The **JPEG** compression algorithm used to construct a pyramid. Only data that meets the JPEG compression specifications can use this compression type. If you choose **JPEG**, you can set the compression quality, as shown in Figure 7-20.

Figure 7-20: Set compression quality for JPEG mode

The compression quality used when constructing a pyramid using the **JPEG** compression method. The value must be between 0 and 100. The closer the value is to 100, the higher the image quality, but the lower the compression ratio.
Specify whether to build the pyramid in the missing location or rebuild all the pyramids (even if they already exist), providing the following two options.

- Even if the pyramid already exists, it will still be built; Therefore, the existing pyramid will be covered. This is the default setting.
- Build a pyramid only when it does not exist.

7.8: Creates raster mosaic dataset

Mosaic dataset is a newly released feature of ArcGIS after version 10, which is aimed at managing image data. Embedded datasets can be indexed to perform queries on image sets, and can be used to manage and publish massive multi resolution, multi sensor images. They also provide dynamic embedding and real-time processing capabilities for image data. Its biggest advantage is its advanced image query function and real-time processing function, and it can also serve as a data source for providing ImageServer image services. PyramidMap creates an image mosaic dataset entry as shown in Figures 7-21.

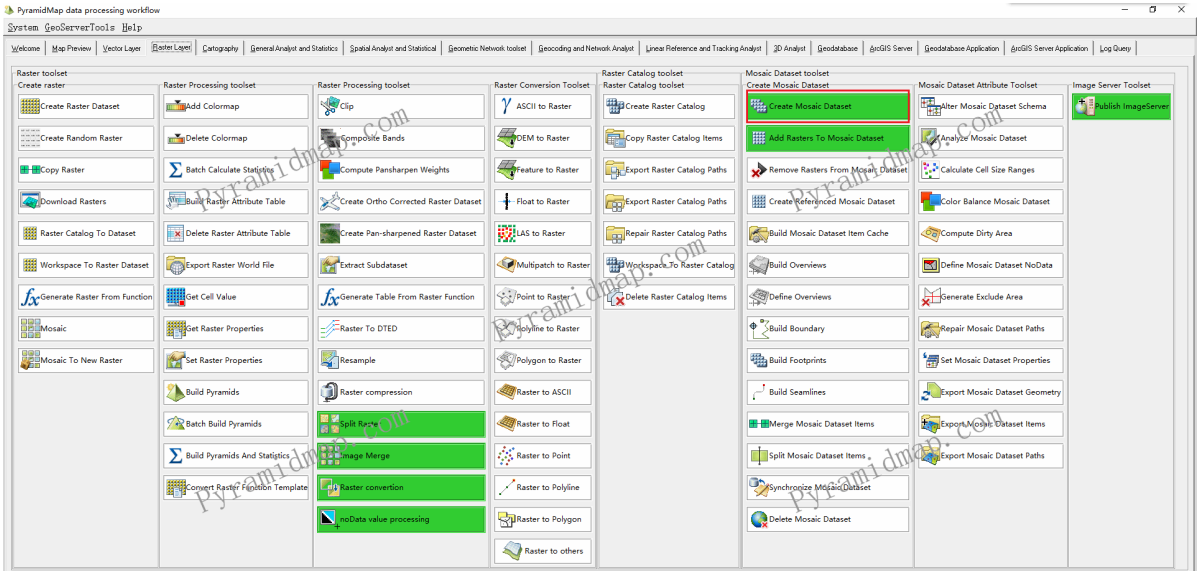


Figure 7-21: PyramidMap Create Raster Mosaic Dataset Entry

Creating a raster mosaic dataset is to create an empty mosaic dataset in the geodatabase. The mosaic dataset must be created in the geodatabase which includes the gdb file type database and the DBMS based enterprise geodatabase including but not limited to Oracle, SQL Server, Postgre, DB2, etc. After creating a mosaic dataset, you can add raster to fill it. The interface for constructing the mosaic dataset is shown in Figures 7-22.

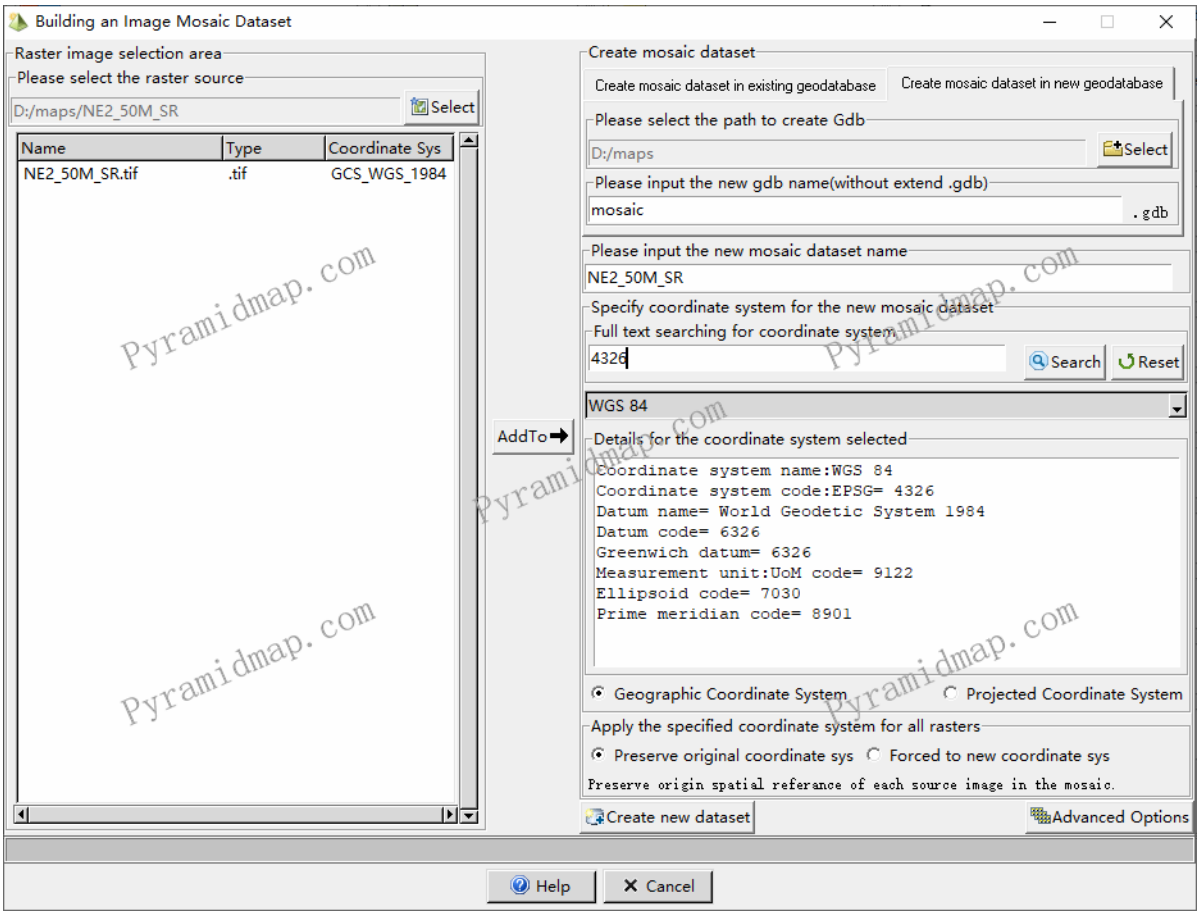


Figure 7-22: The module of creating raster mosaic dataset in PyramidMap

- The name of the mosaic dataset must not violate the restrictions of the geodatabase or the underlying database. For example, names cannot start with numbers, and it is not recommended to use Chinese or special characters for naming.
- The program provides the unified spatial reference for all items embedded in the dataset. If the source raster does not have a spatial reference, you can assignment it in the configuration during the mosaic processing, otherwise, it will be applied to the spatial reference of the mosaic dataset. This parameter is used in conjunction with the 'Specify coordinate system for all grids' option in' Advanced Options'. When loading data into a mosaic dataset, the spatial reference specified will be used for all grids item, those different coordinate systems will be covered by the target coordinate system.

This module will guide you completing mosaic dataset construction and appending rasters to it. Specify the execution strategy for creating mosaic datasets and importing image data through the "Advanced Options", as shown in Figure 7-23 for detailed configuration.

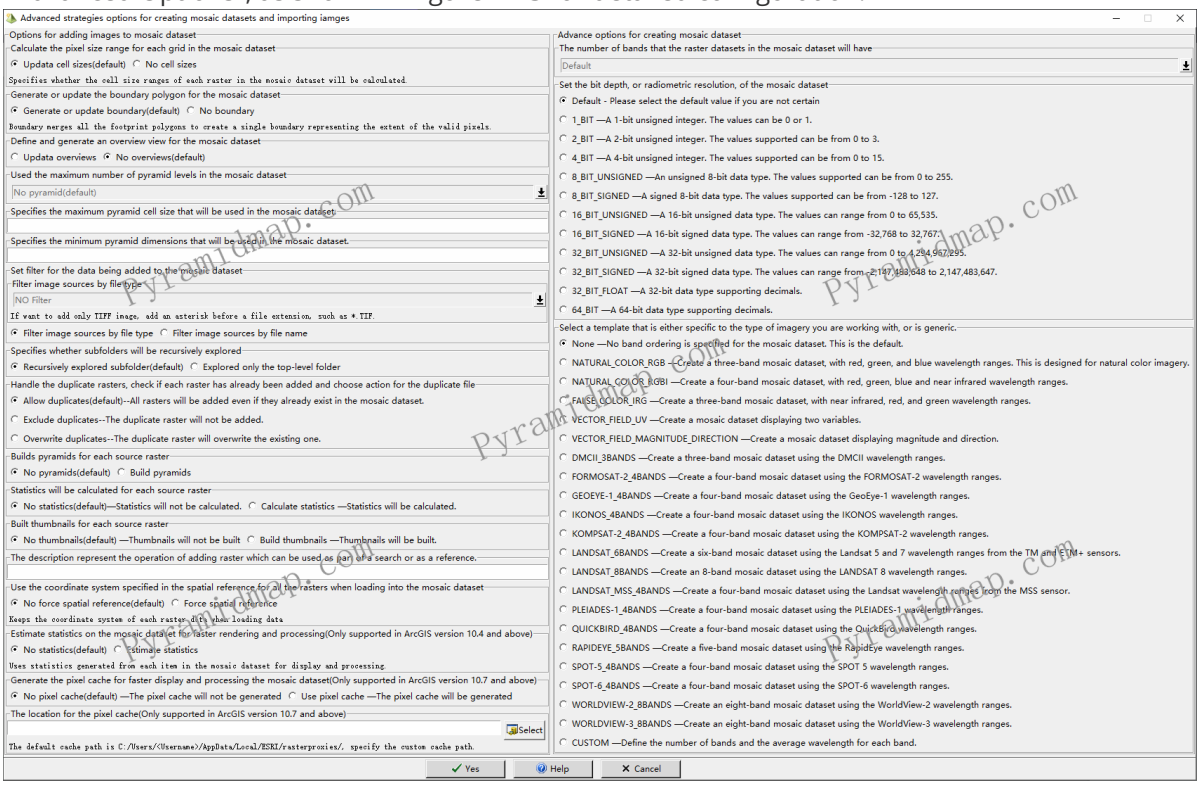


Figure 7-23: Strategy options for creating mosaic datasets and adding rasters

The strategy options includes two parts: creating mosaic dataset and adding rasters to it. There are detailed description for every option item. The program will execute according to the default option values even without your special setting anything, completing the creation of the dataset and the import of image data. After completion, the program creates a mosaic dataset according to the preset strategy, as shown in Figure 7-24.

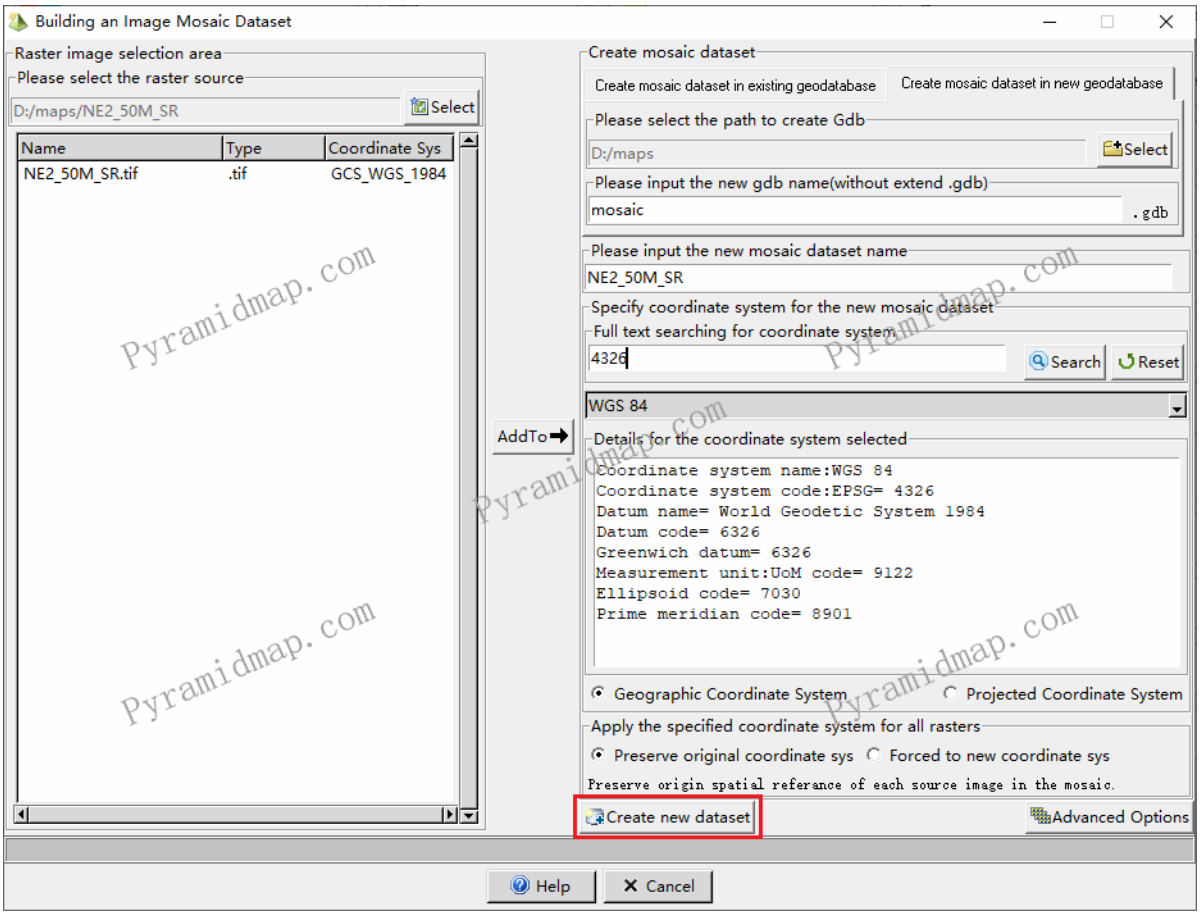


Figure 7-24: Creating a mosaic dataset according to a preset strategy

The prompt for successful mosaic dataset creation is shown in Figure 7-25.

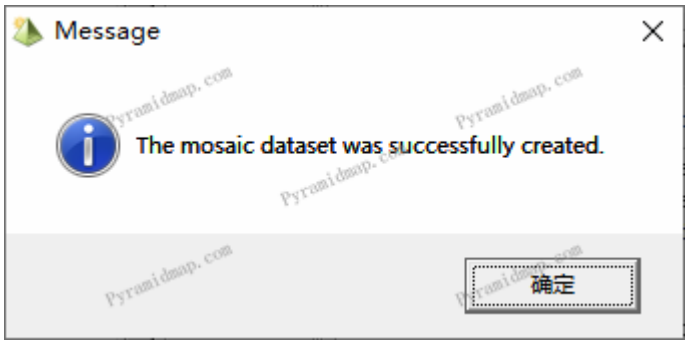


Figure 7-25: Prompt for successful creation of mosaic dataset

The next, we will add rasters into the newly created mosaic dataset, as shown in Figures 7-26.

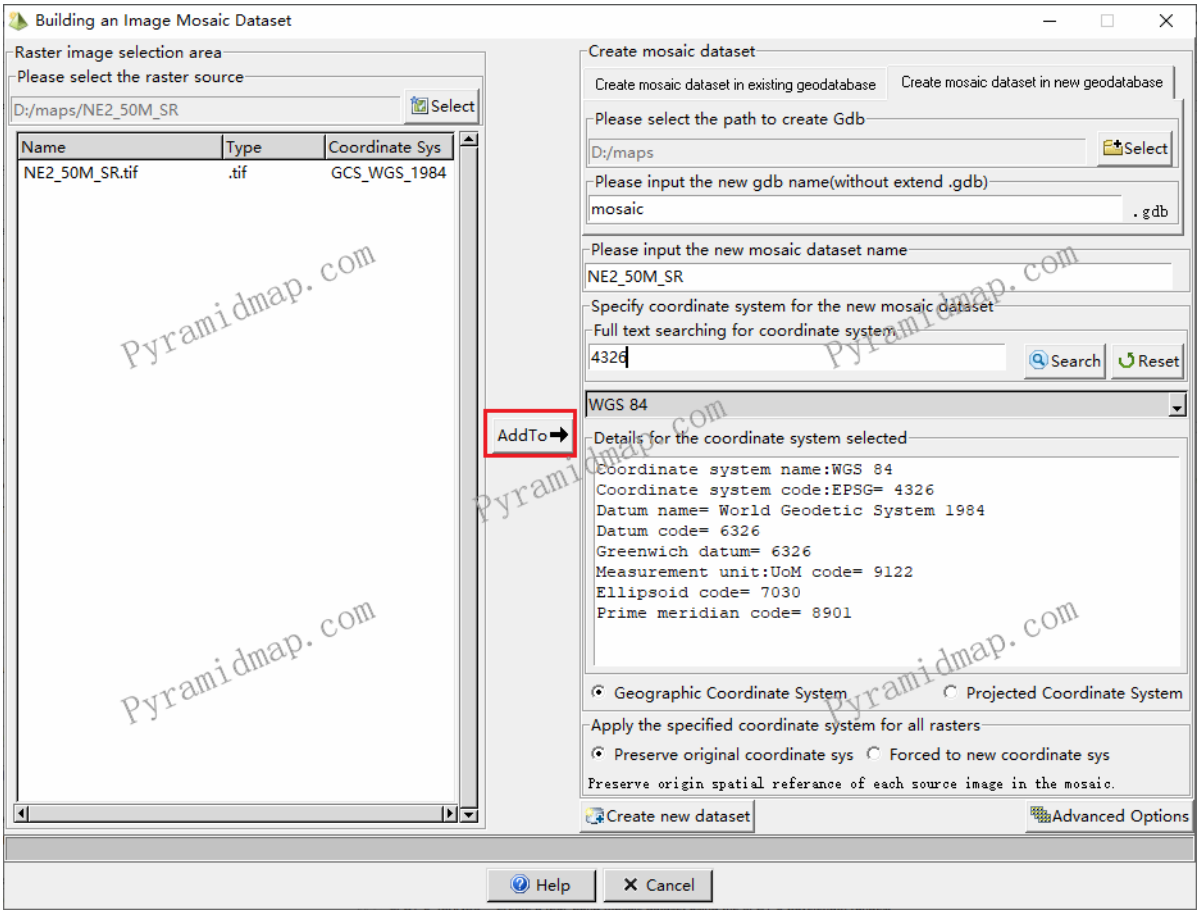


Figure 7-26: Add rasters into a Mosaic Dataset Created Newly

The successful prompt is shown in Figure 7-27.

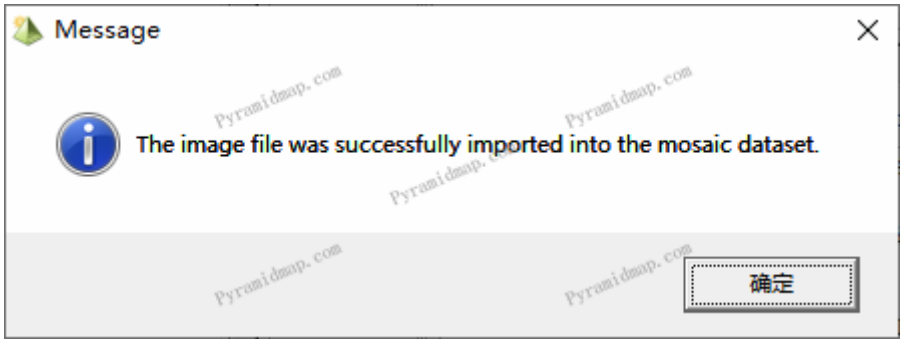


Figure 7-27: Prompt for rasters added to the mosaic dataset successfully

Open and preview the mosaic dataset in ArcMap as shown in Figure 7-28.

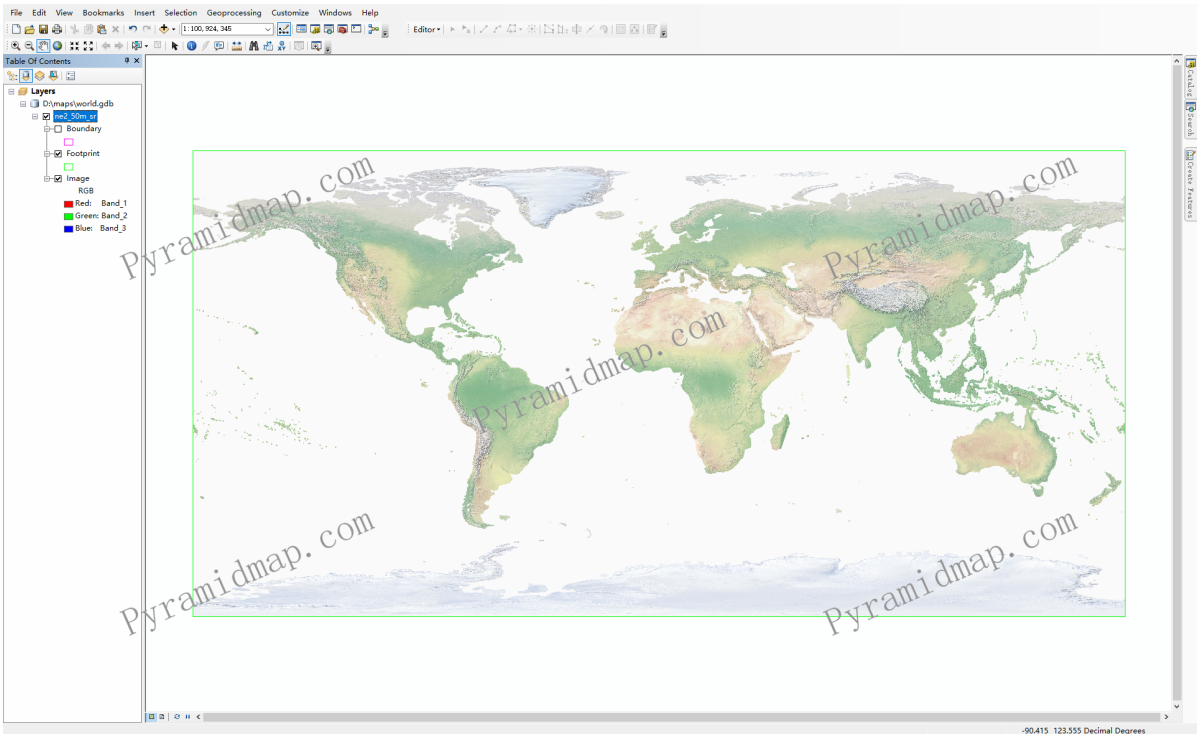


Figure 7-28： Open and preview the mosaic dataset in ArcMap

At this point, You has completed the process of creating a complete raster mosaic dataset and adding rasters to it with the PyramidMap's help. In summary, it is a very simple process which transforming complex raster processing into a simple visualization process within the PyramidMap.

7.9: Publish image server

ArcGIS Image Server provides analysis and processing services for large-scale images, elevation data, grids, and other remote sensing data, and provides image service interfaces to web and mobile devices. ArcGIS Image Server supports effective and efficient utilization of large image and grid collections, enabling dynamic image services through dynamic mosaic and analysis processing. Unlike the dynamic layer service MapServer and FeatureServer, Image service can be directly published using the mosaic dataset source without mxd documents after the ArcGIS10 version. We have implemented a simple and easy-to-use map image publishing process based on [7.8: Creates Raster Mosaic Dataset]. The image service publishing portal is shown in Figures 7-29.

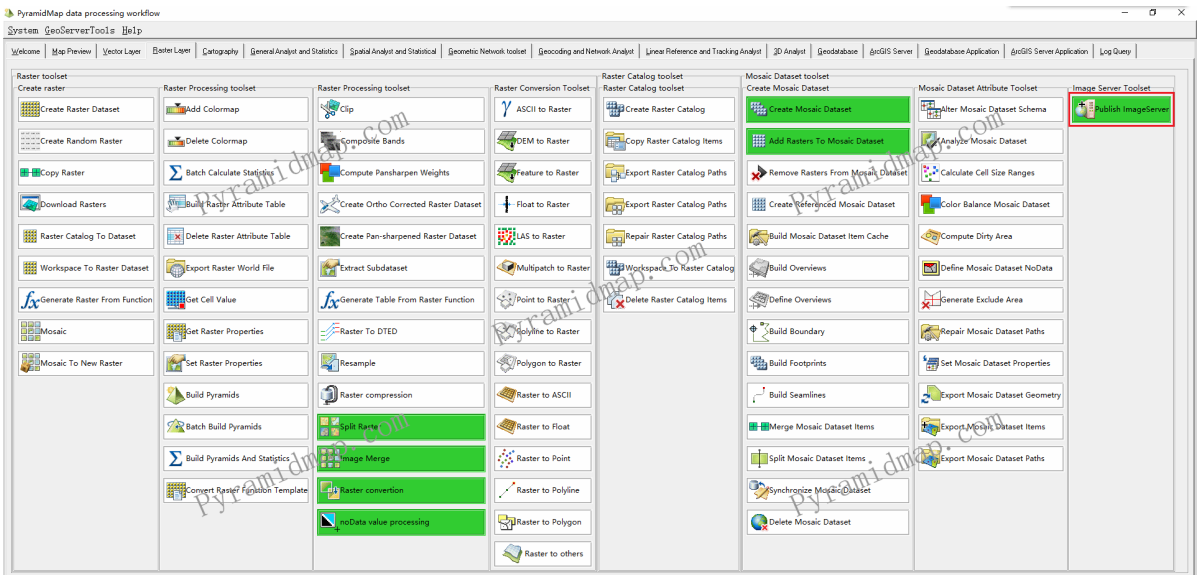


Figure 7-29: Publishing image service Portal

PyramidMap provides image data sources in two ways: selecting mosaic datasets from existing geographic databases or creating new ones.

① Select the existing mosaic dataset as the data source and publish it as an image service, as shown in Figure 7-30.

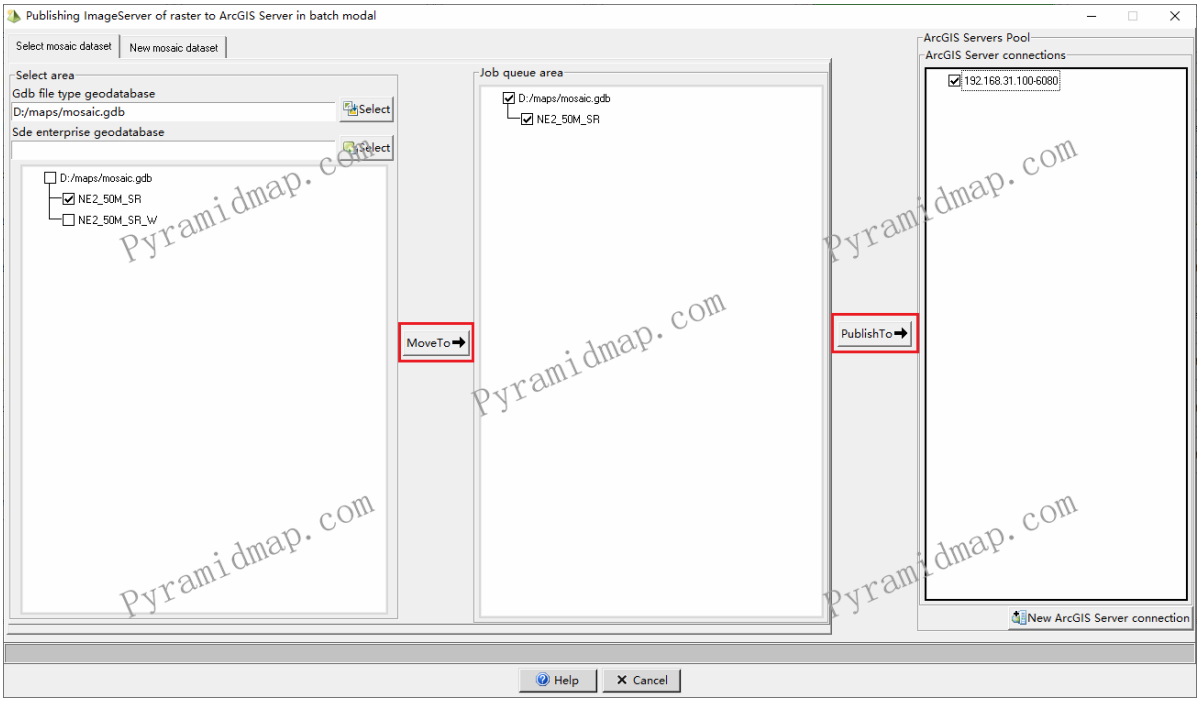


Figure 7-30: Select the existing mosaic dataset as the data source and publish it as an image service

② Create a new embedded dataset as a data source and publish it as an imaging service, as shown in Figure 7-31.

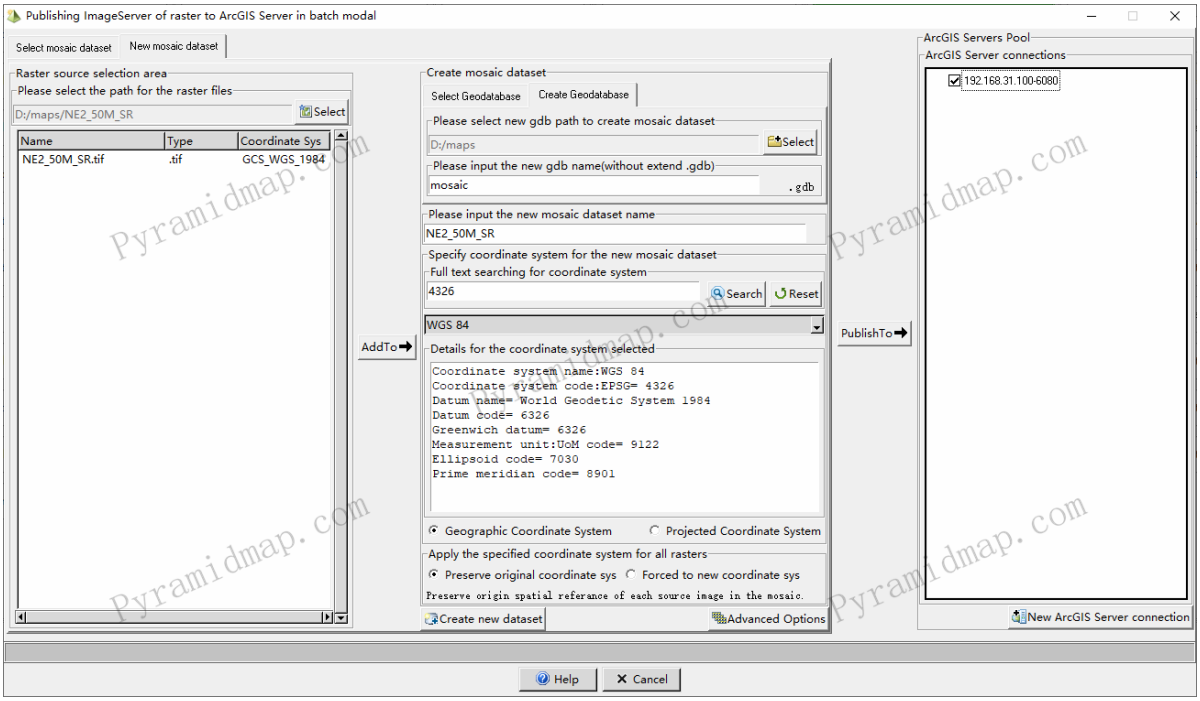


Figure 7-31: Create a new mosaic dataset as a data source and publish it as an imaging service

PyramidMap provides cmd output monitoring of the running process, as shown in Figure 7-32.

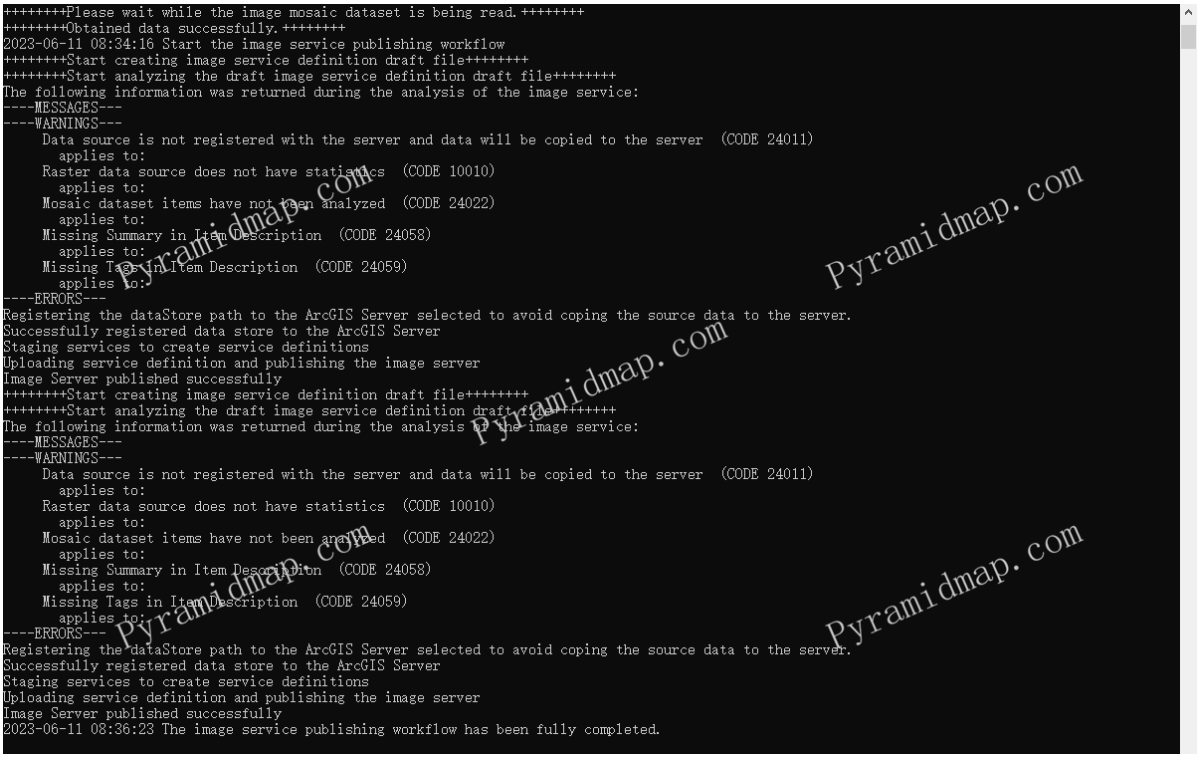


Figure 7-32: The cmd monitoring of the imageserer publishing in PyramidMap output

The prompt for publishing image services successfully shown as in Figure 7-33.

Figure 7-33: Prompt for publishing image Server Successfully in PyramidMap

Next, let's preview the image services just published in ArcGIS Server, the service list is shown in Figure 7-34.

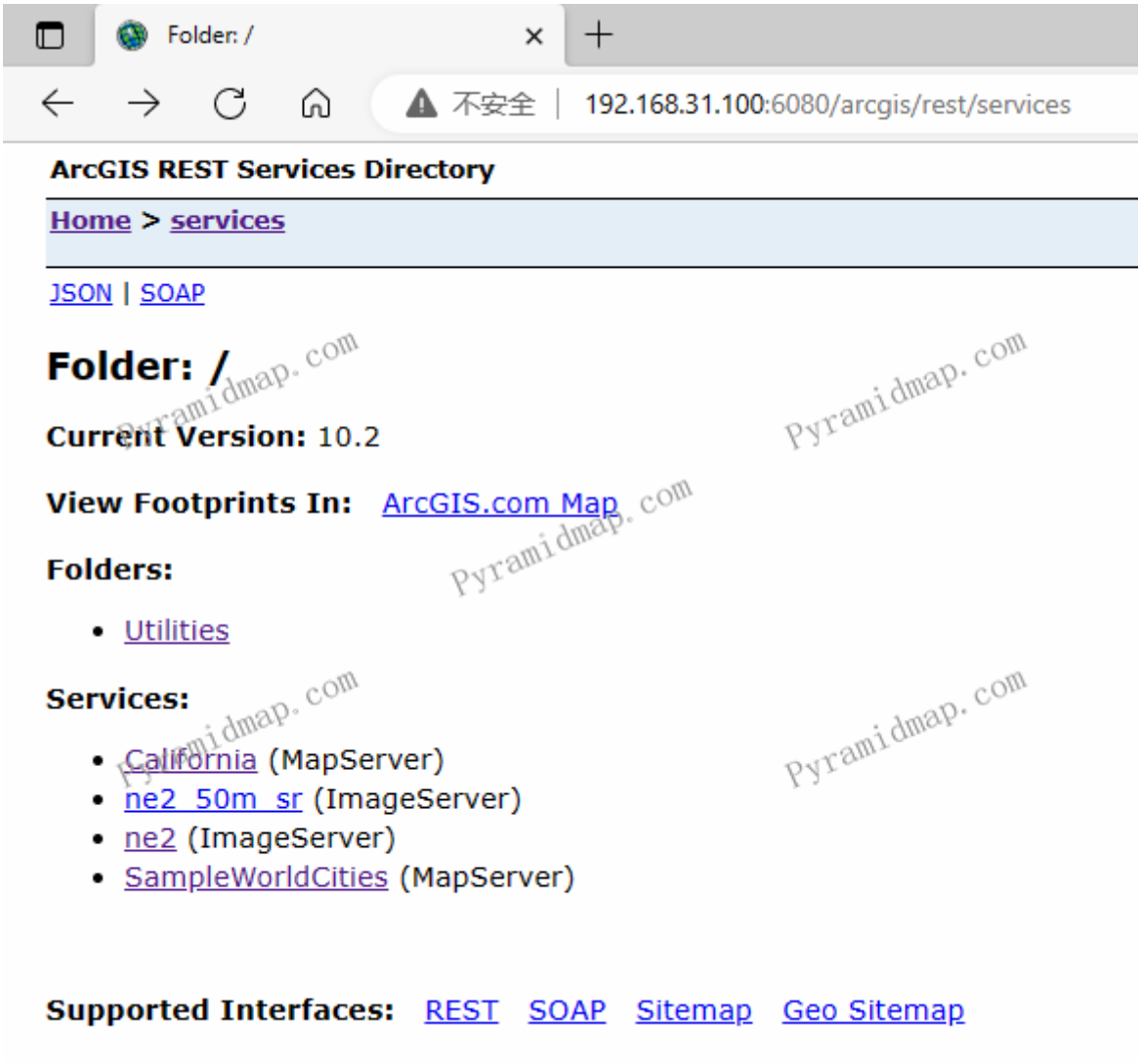


Figure 7-34: List of map and image services in ArcGIS Server

The preview effect on the web side is shown in Figure 7-35.

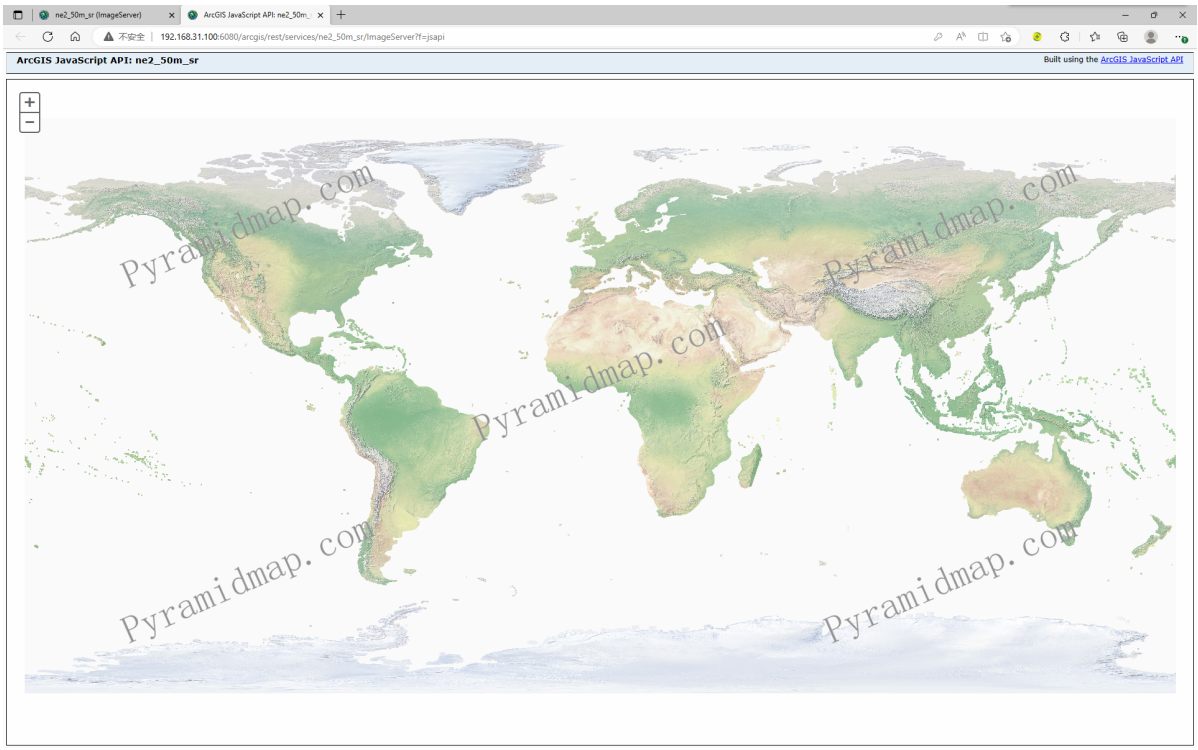


Figure 7-35: The image service in ArcGIS Server previewed on the Web

In summary, it is a very simple process to publish complex image services through the visualization interface guidance in PyramidMap. PyramidMap is dedicated to providing you with a visual map image service publishing process. The purpose of PyramidMap is to use a simple visualization process to guide and assist users in completing complex base map data processing. Lack of visual support will be a daunting task.

8: Create mxd workflow

8.1: Vector layers mxd

The program supports batch combination of SHP layer, GDB, MDB and SDE to form MXD map document.The process model for creating vector mxd is shown in Figure 8-1.

8.1.1: Work model

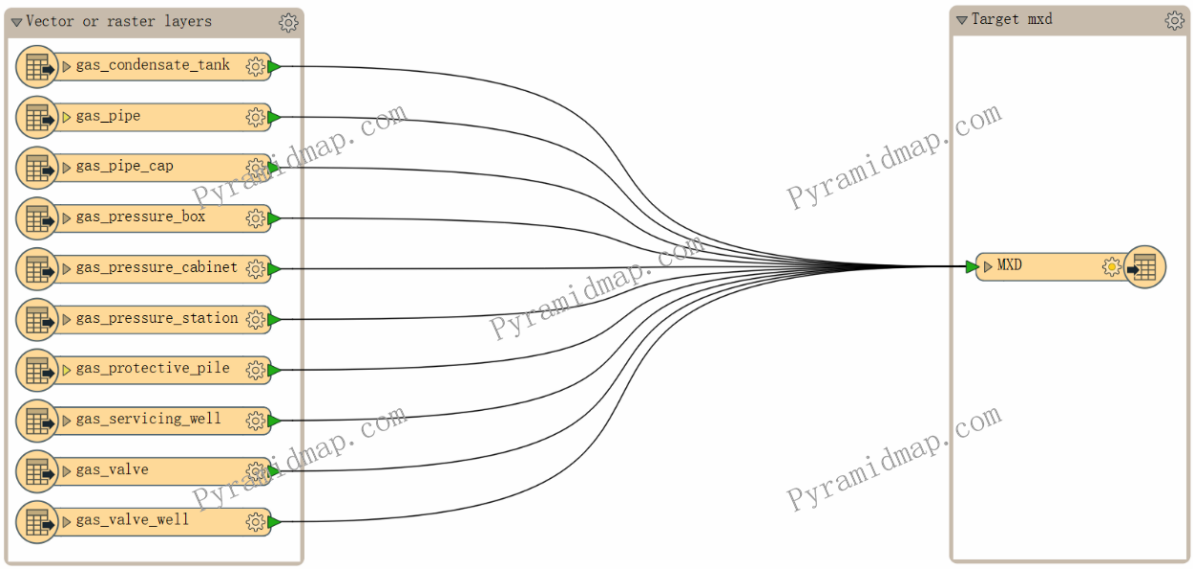


Figure 8-1: PyramidMap creates vector mxd processing model

Specification: layers in the same document must have the same coordinate system.

8.1.2: Operation interface

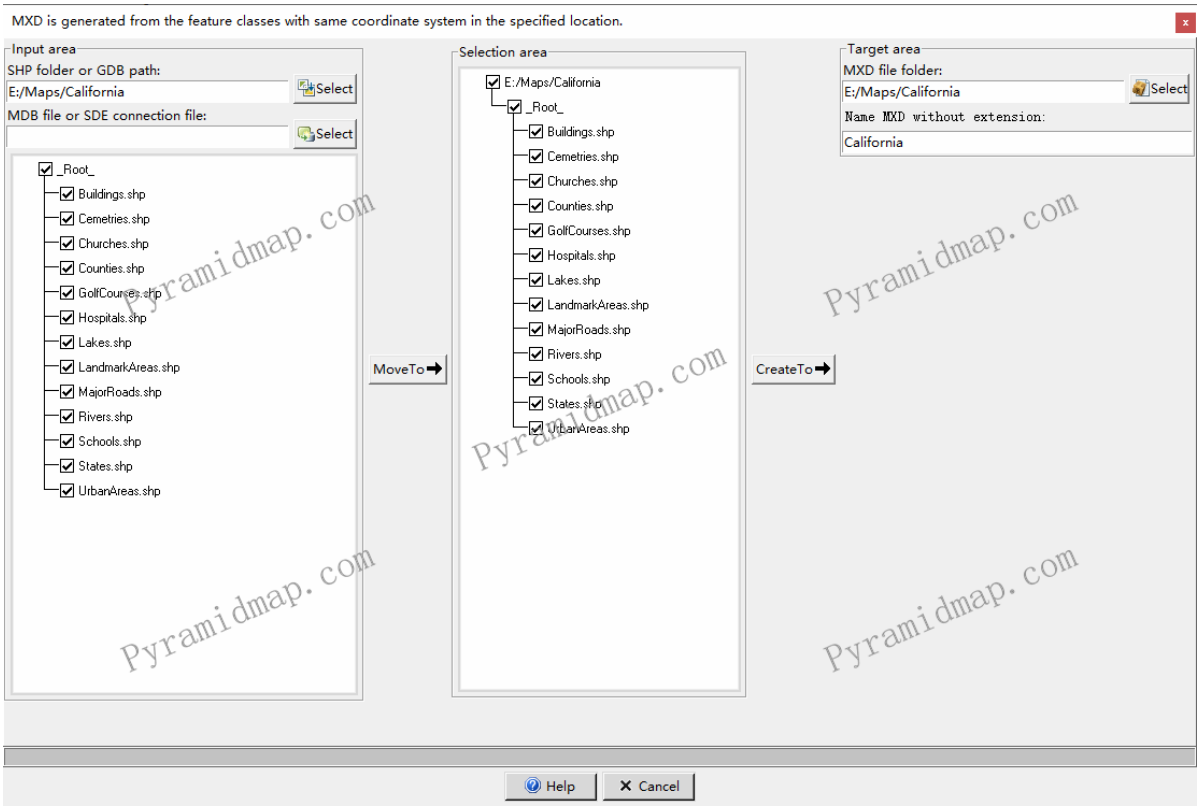


Figure 8-2: PyramidMap creates vector mxd processing interface

MXD map documents can be used to publish map services of MapServer and featureserver types. The default service name is MXD name. When you want to publish, you can rename MXD in the publishing process. However, for convenience and error avoidance, it is recommended to name MXD in English.

8.2: Raster layers mxd

The program supports batch combination of img, TIF, GIF and other image layers to form MXD map documents. The process model for creating image mxd is shown in Figure 8-3.

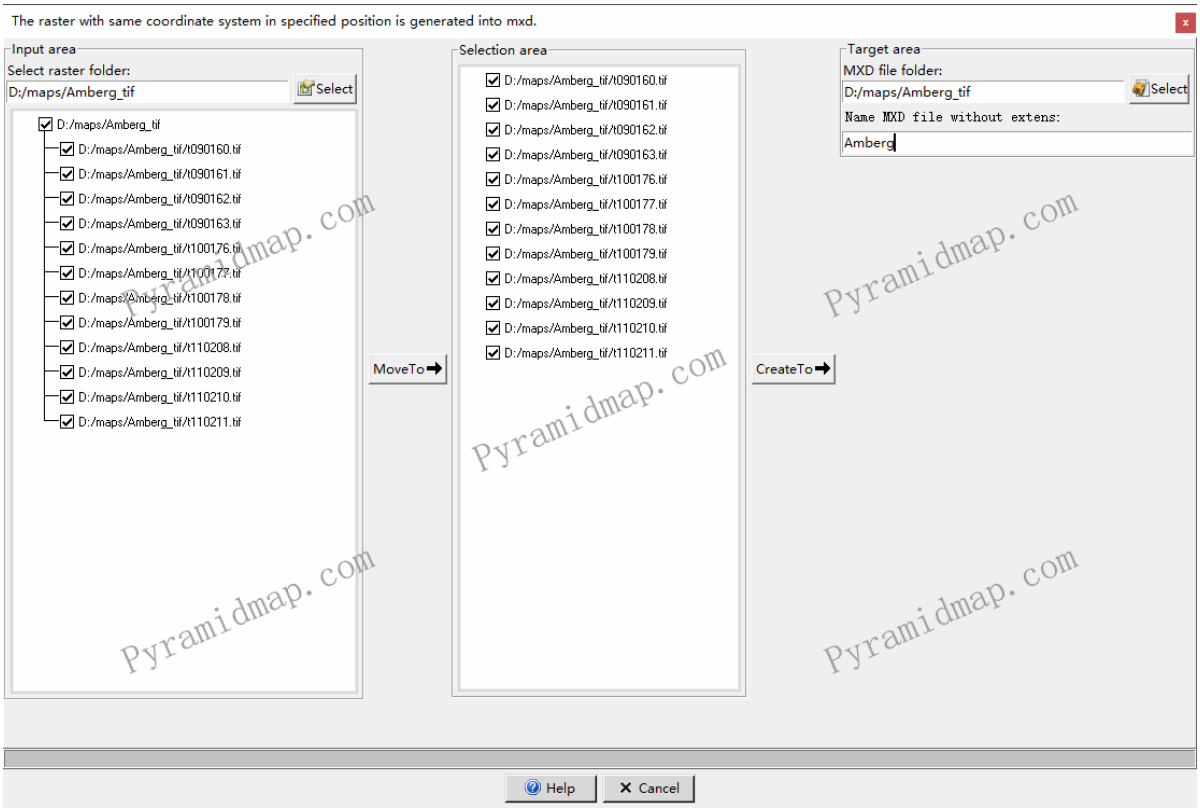


Figure 8-3: PyramidMap creates image mxd processing interface

9: Publish map servers

The purpose of map applications is to provide rich, flexible, convenient, accurate, and exquisite map services for web and mobile terminals, which is currently the most widely used mode of GIS application. It is necessary to provide access to online map data for web and mobile terminals through professional map servers, such as common ArcGIS Server, SuperMap iServer, GeoServer, etc. PyramidMap mainly provides automated processes for map services such as MapServer, FeatureServer, and ImageServer for ArcGIS Server. The purpose is to guide users through visual interfaces and assist them in achieving automated map publishing processes. PyramidMap uses a visual interface to develop a map publishing process model, simplifying the map service publishing process, especially for complex feature layer service processing, which can reduce the professional technical requirements and labor intensity of staff, and improve work efficiency.

9.1: Publish MapServer

9.1.1: Workflow model

PyramidMap publishes MapServer type map services to ArcGIS Server in the mode of mxd map documents. The process model is shown in Figure 9-1.

Figure 9-1: PyramidMap publishes MapServer process model

9.1.2: Operation interface

Taking a city map as an example, the layer structure and display effect in ArcMap are shown in Figure 9-2.

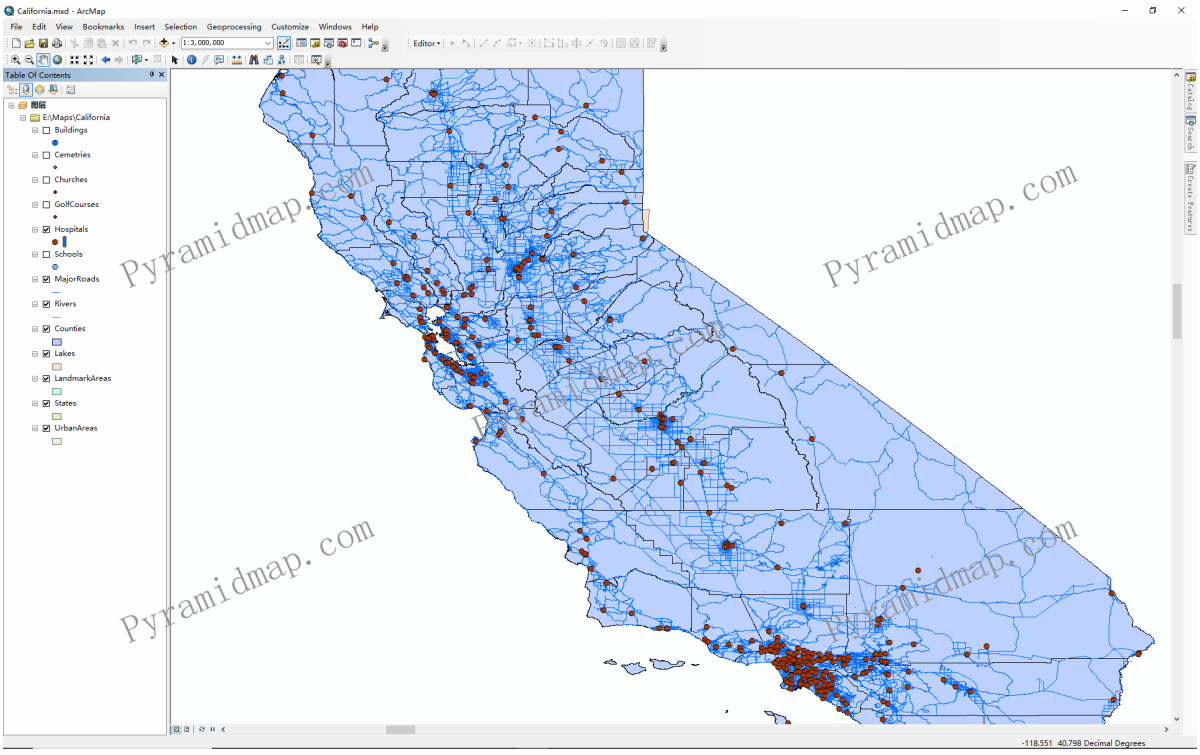


Figure 9-2: A city map layers structure and display in ArcMap

Next, we will package the layers collection into an mxd document and publish it as a MapServer type map service for ArcGIS using the PyramidMap built-in engine. The process is shown in Figure 9-3.

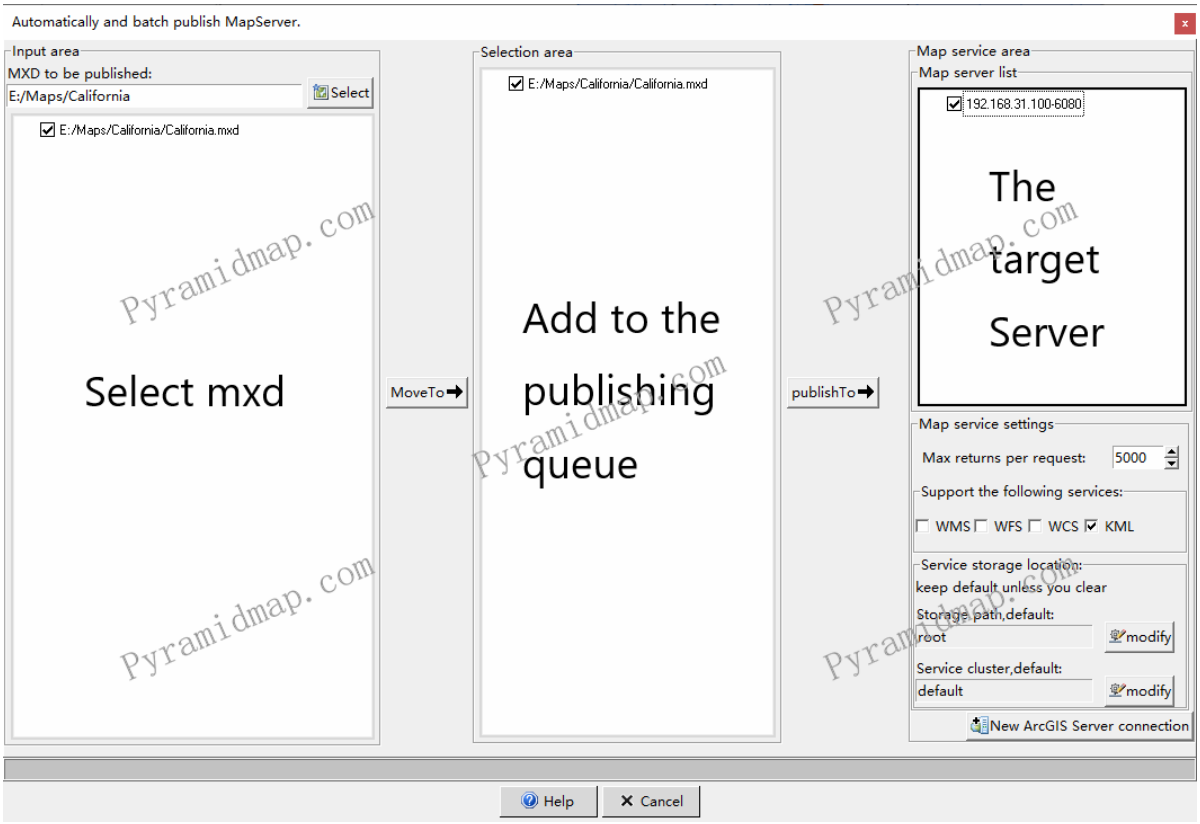


Figure 9-3: PyramidMap packages layers into mxd documents and publishes them as MapServer to server

PyramidMap provides cmd output monitoring of the running process, as shown in Figure 9-4.

Figure 9-4: PyramidMap monitors the status of the publishing process

After successful publishing, the system message prompt is shown in Figure 9-5.

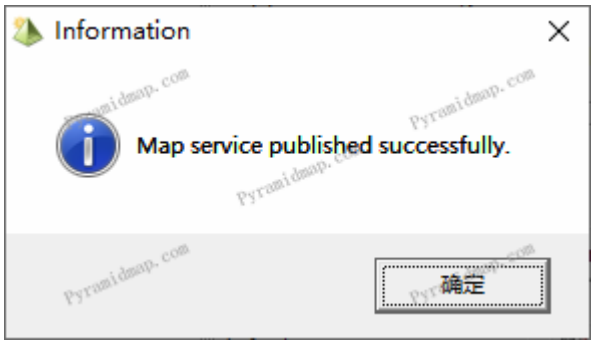


Figure 9-5: The system message prompt for MapServer successful published

The MapServer layers published successful in ArcGIS Server are shown in Figure 9-6.

Figure 9-6: The MapServer layers in ArcGIS Server

The web map view implemented through the ArcGIS JavaScript API interface is shown in Figure 9-7.

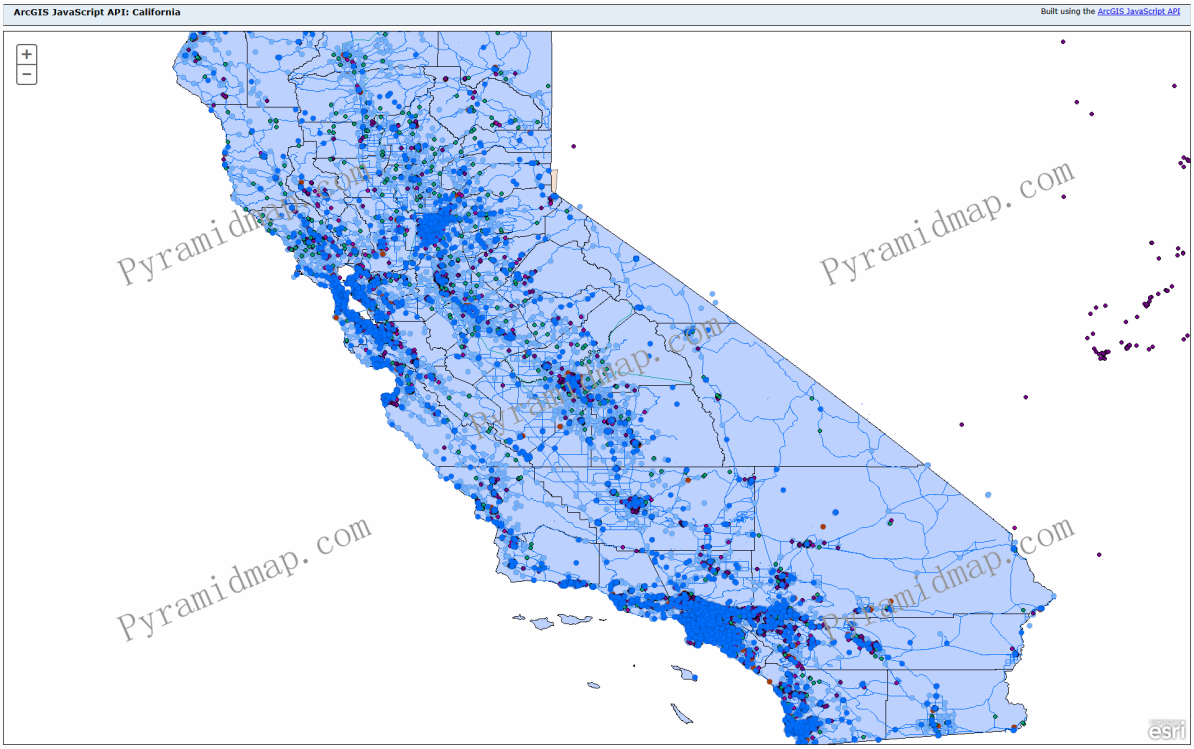


Figure 9-7: The web map viewer for layers of MapServer which published in ArcGIS Server

9.2: Publish FeatureServer

The FeatureServer in ArcGIS differs from MapServer is in that editing functionality on top of the latter, supporting data and graphical editing of map features on the web and mobile terminals, and directly saving map data to the geo database. Both FeatureServer and Web Feature Server (WFS) which based on OGC standardization support feature editing functions on various terminals. The difference between FeatureServer and WFS is that the former only belongs to the private technology system of ArcGIS, it can exert more efficient map processing capabilities and provide more flexible and rich graphical processing capabilities for web and mobile terminals only within the ArcGIS technology system .

Publishing FeatureServer in ArcGIS is quite complicated and tedious. From initializing geographical database to creating geographical database connection, from registration and hosting of geographical database to ArcGIS Server to importing database of feature layer, and finally publishing as FeatureServer type service in ArcGIS Server, almost every link requires professional GIS technical support, even for professional GIS technicians, It's not an easy job either.

PyramidMap integrates the above steps into a visual process through a built-in engine, achieving one click completion of map selection, map import, and service publishing, achieving true map automation process.

9.2.1: Workflow model

PyramidMap publishes FeatureServer to ArcGIS Server in the mode of mxd documents. The process model is shown in Figure 9-8.

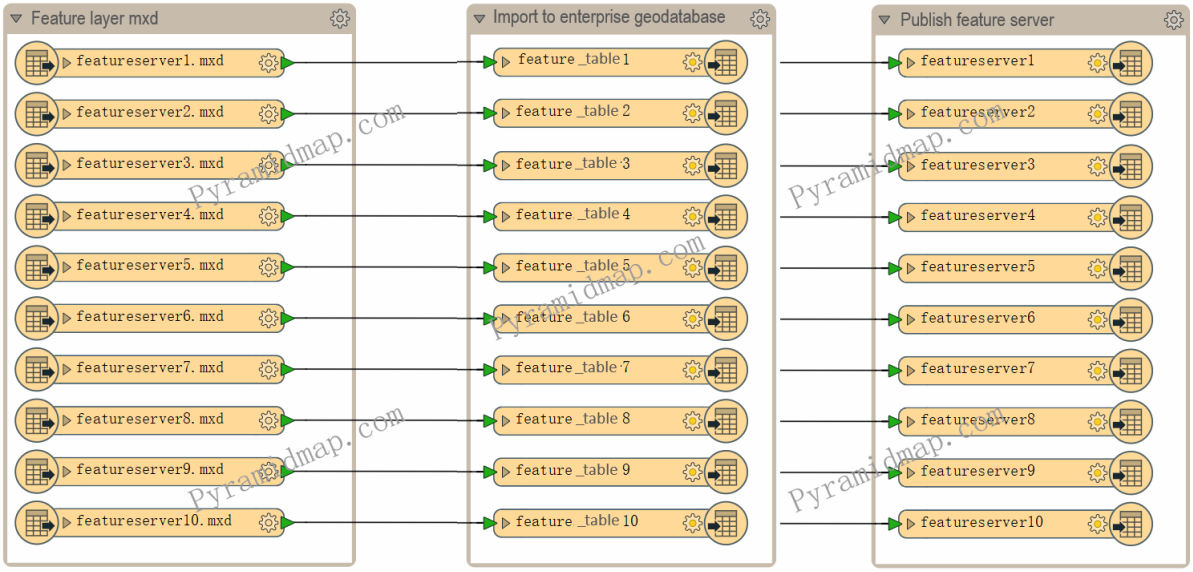


Figure 9-8: PyramidMap Publishing featureServer process Model

9.2.2: Operation interface

Taking a city map as an example, the layer structure and display in ArcMap are shown in Figure 9-9.

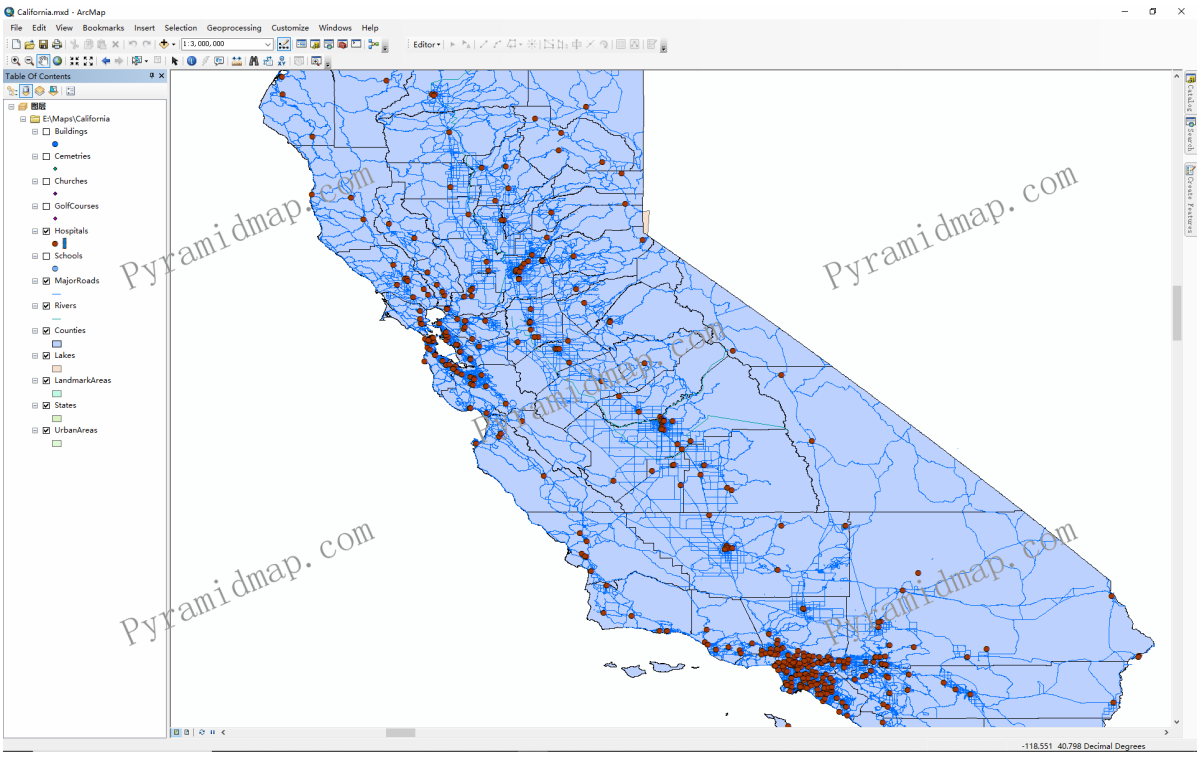


Figure 9-9: A city map layers structure and display in ArcMap

Next, we will publish the feature layers represented by the above California. mxd as the FeatureServer to ArcGIS Server through the PyramidMap engine built-in. After completion, the map data is hosted to the SQL Server geographic database. The operation interface is shown in Figure 9-10.

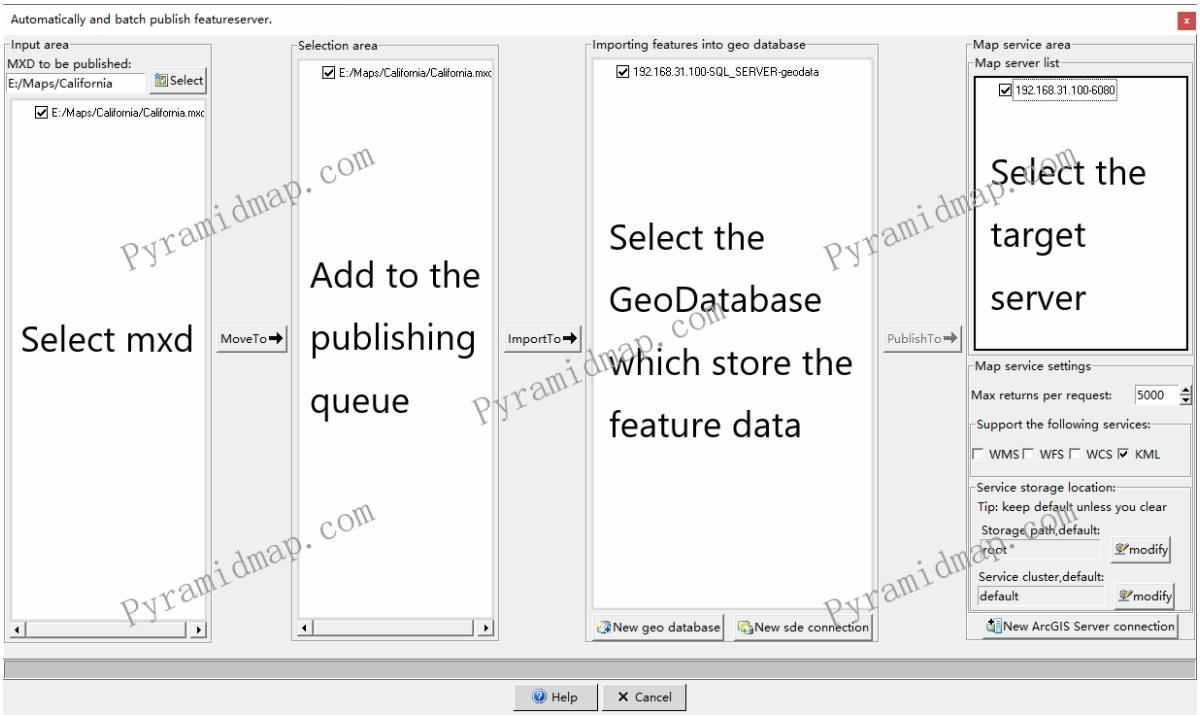


Figure 9-10: Publishing FeatureServer interface in PyramidMap

①:The max returns per request: it's the maxium return features number per user request, this depends on your needs and the capacity of the server, not the larger the better.

②:The following service wms/wfs/wcs are standardized map service that follows OGC specification and can be shared among different map platforms.

PyramidMap provides cmd output monitoring of the running process, as shown in Figure 9-11.

Figure 9-11: PyramidMap monitors the status of the publishing process

After successful publishing, the system message prompt is shown in Figure 9-12.

Figure 9-12: The system message prompt for FeatureServer successful published

The FeatureServer layers are shown in Figures 9-13.

California (FeatureServer)

View In: ArcGIS.com Map

View Footprint In: ArcGIS.com Map

Service Description:

Has Versioned Data: false

MaxRecordCount: 5000

Supported Query Formats: JSON, AMF

Layers:

- Buildings (0)
- Cemetries (1)
- Churches (2)
- GolfCourses (3)
- Hospitals (4)
- Schools (5)
- MajorRoads (6)
- Rivers (7)
- Counties (8)
- Lakes (9)
- LandmarkAreas (10)
- UrbanAreas (11)

Description:

Copyright Text:

Spatial Reference: 4326 (4326)

Initial Extent:

XMin: -125.41816977312318
YMin: 33.324436642734334
XMax: -113.93251149798037
YMax: 40.85543072618719
Spatial Reference: 4326 (4326)

Full Extent:

XMin: -170.732222200394
YMin: -14.3355556001101
XMax: 163.028333300143
YMax: 64.9540207526566
Spatial Reference: 4326 (4326)

Units: esriDecimalDegrees

Document Info:

The web view window implemented through the ArcGIS JavaScript API interface is shown in Figures 9-14.

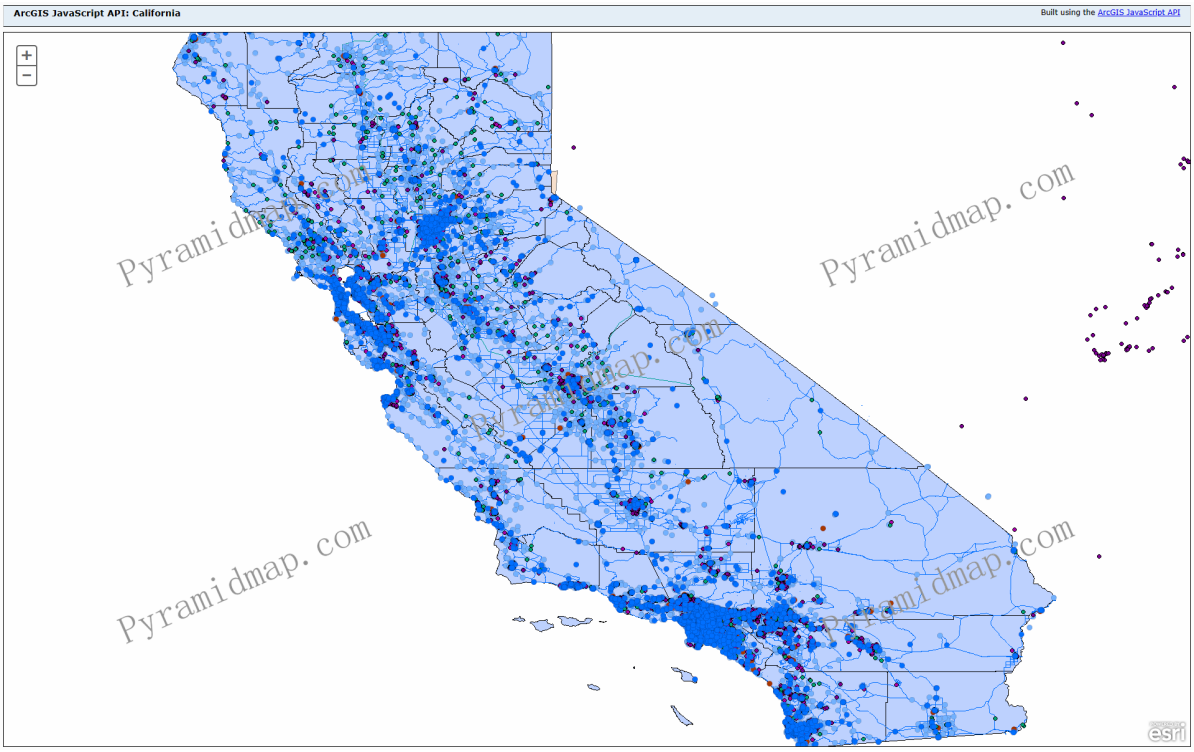


Figure 9-14: The display effect of FeatureServer in the web map window after published successfully

The biggest difference between FeatureServer and MapServer is its support for direct modification and editing of map features on the web or mobile terminal. Taking the Hospitails layer in the FeatureServer collection as an example, its data list is displayed on the web and editing and modification functions are provided, as shown in Figure 9-15.

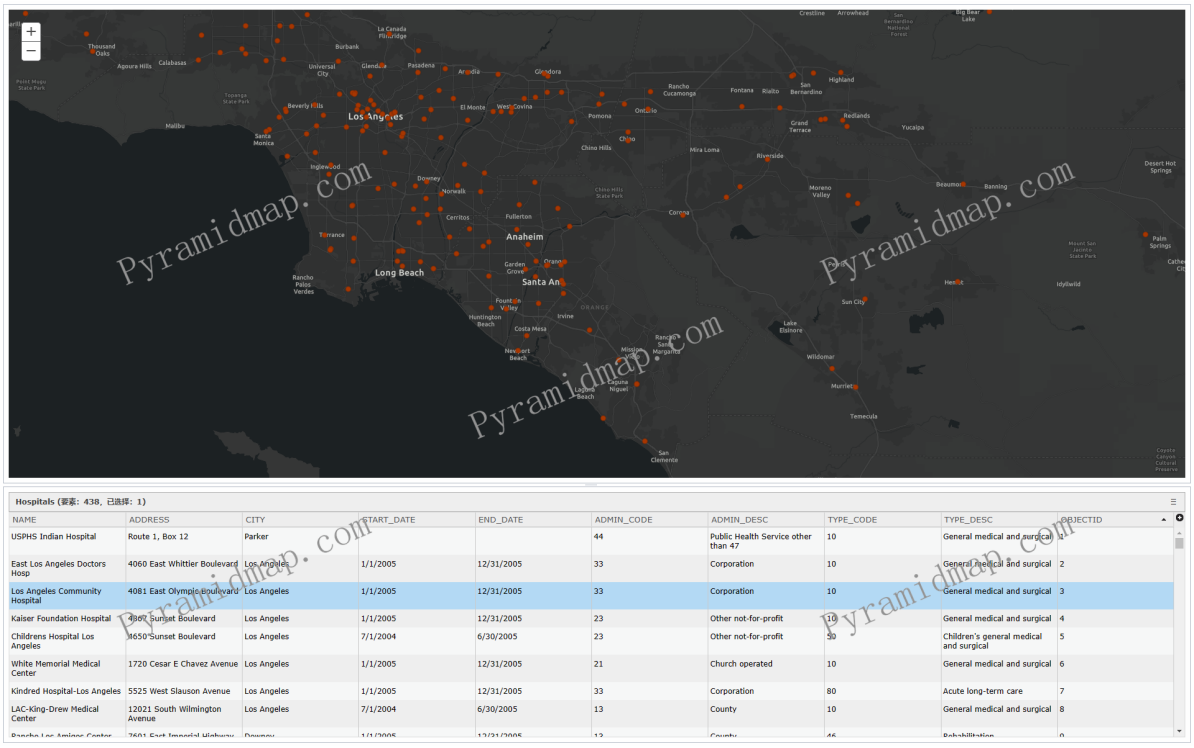


Figure 9-15: FeatureServer displays its layer data on the web and implements editing functions

As the visualization client of ArcGIS Server, PyramidMap manages the layers in the server and supports map data conversion and preview. For detailed functions, see the 10: GIS Server chapter.

10: GIS database

GIS database provides storage function for map data. In ArcGIS system, GIS databases are divided into two categories: Gdb and Mdb file type databases and DBMS relational enterprise geographical database represented by Oracle, SQL Server, Postgre and DB2. Among them, the Gdb file geographical database can be used by multiple users at the same time, but one data can only be edited by one user. Therefore, a file geographical database can be accessed by multiple editors, but different data must be edited. Mdb is based on Microsoft Access data files, with a maximum capacity of 2 GB, and only one user can edit data in a personal geographical database at a time. The advantages of Gdb and Mdb file type databases are lightweight and flexible, which can meet the general lightweight applications. However, the more complex spatial geometry data processing and map data versioning management must be carried out in the enterprise level geographical database. PyramidMap mainly aims at ArcGIS, and realizes the visual map data processing process based on enterprise geographical database.

10.1: Layers List

PyramidMap can preview and export the layers in the enterprise geoDatabase as shp, excel, json, csv and other format data. Taking SQL Server as an example, PyramidMap obtains its layer list, as shown in Figure 10-1.

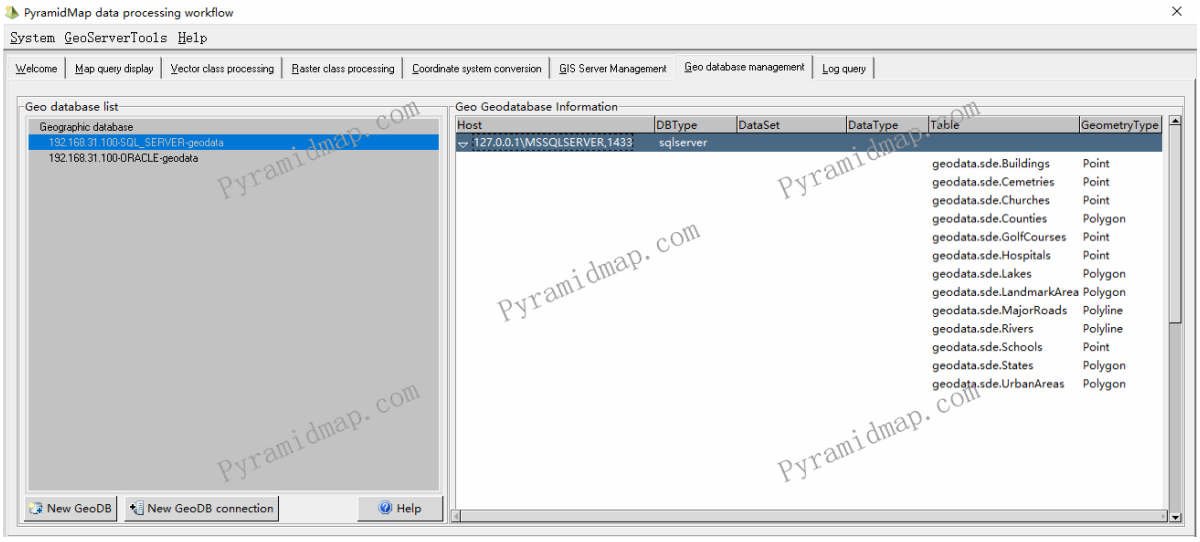


Figure 10-1: Get feature tables in the geodatabase

PyramidMap supports right-click shortcut operations for data export, conversion, and preview of layers in the list. For detailed procedures, see [10.2: Layer Preview] and [10.3: Layer Export].

10.2: Layer Preview

Select the data list and right-click to select map and data preview to preview the selected data table， as show in figure 10-2.

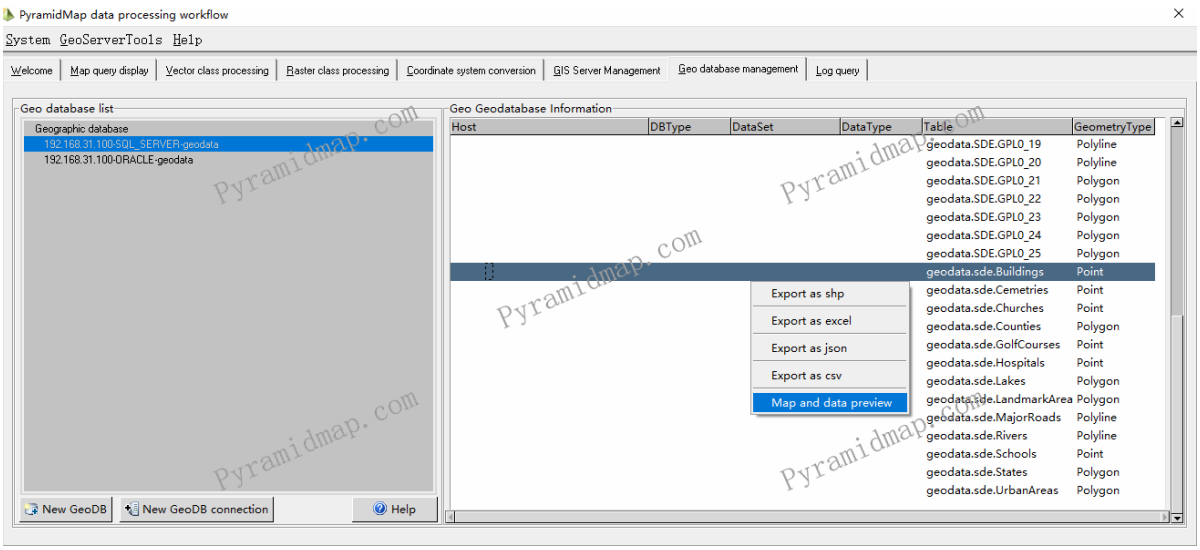


Figure 10-2: Preview feature tables in the geodatabase

Sample and display by feature geometry type. The preview effect of the selected Point type layer Building is shown in Figure 10-3.

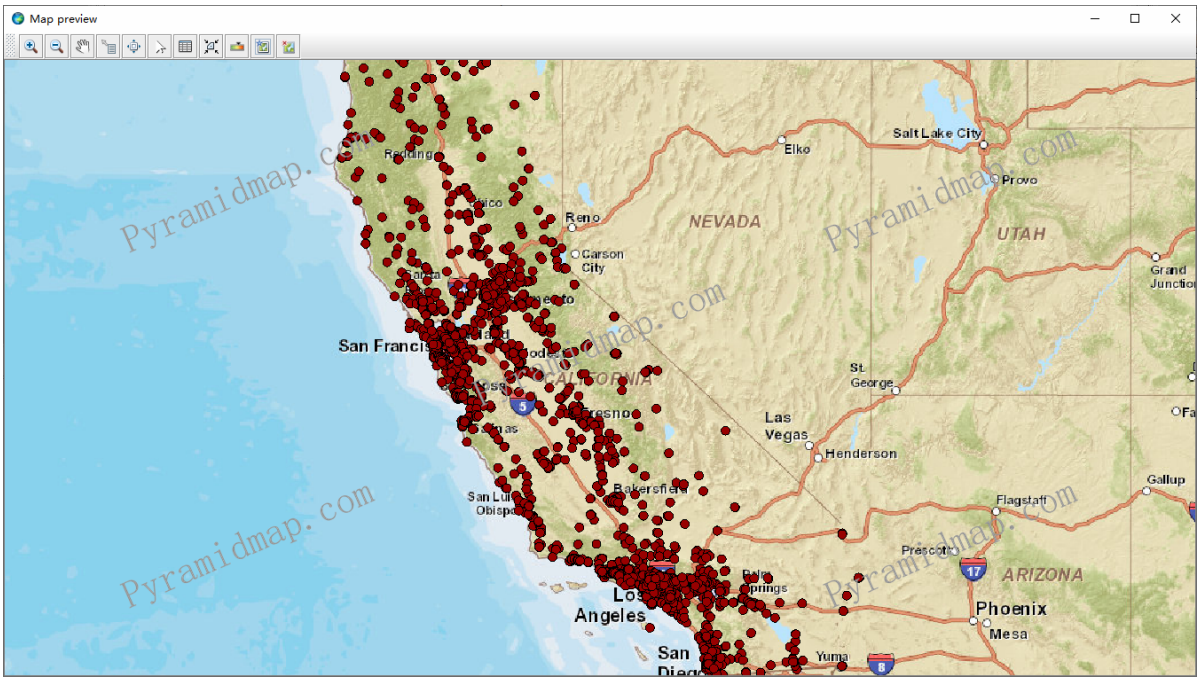


Figure 10-3: PyramidMap displays the Building layer in Geodatabase

The preview of the selected Polyline type layer MajorRoads is shown in Figure 10-4.

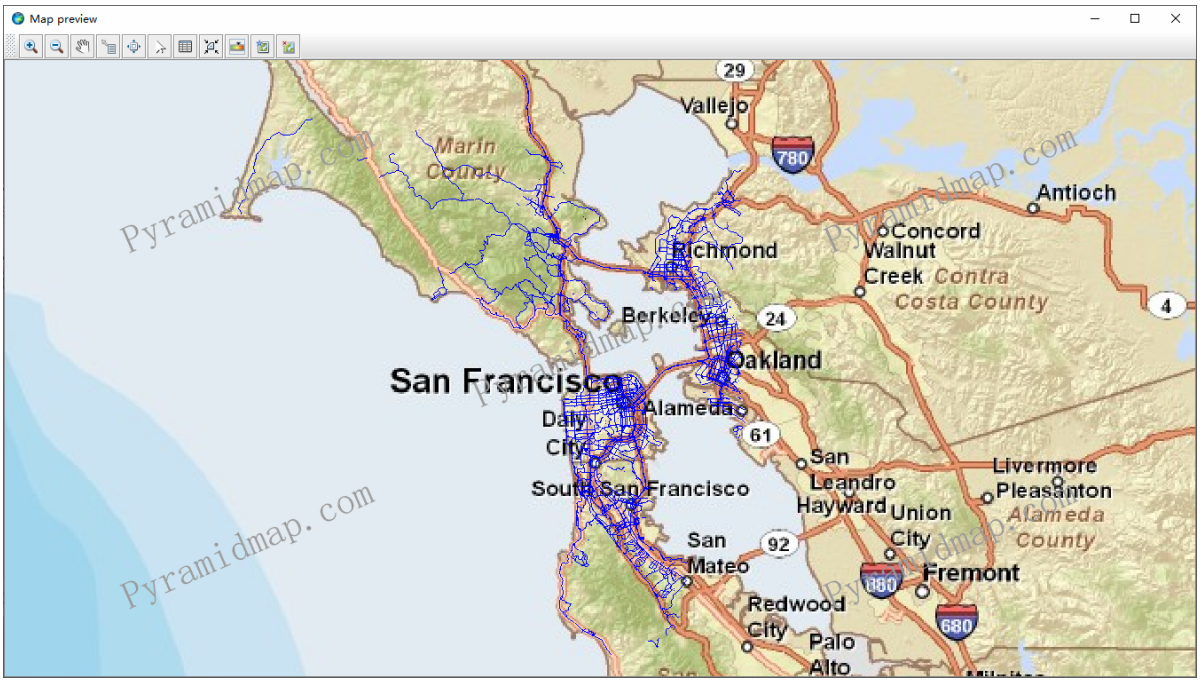


Figure 10-4: PyramidMap displays the MajorRoads layer in Geodatabase

The preview of the selected Polygon type layer UrbanAreas is shown in Figure 10-5.

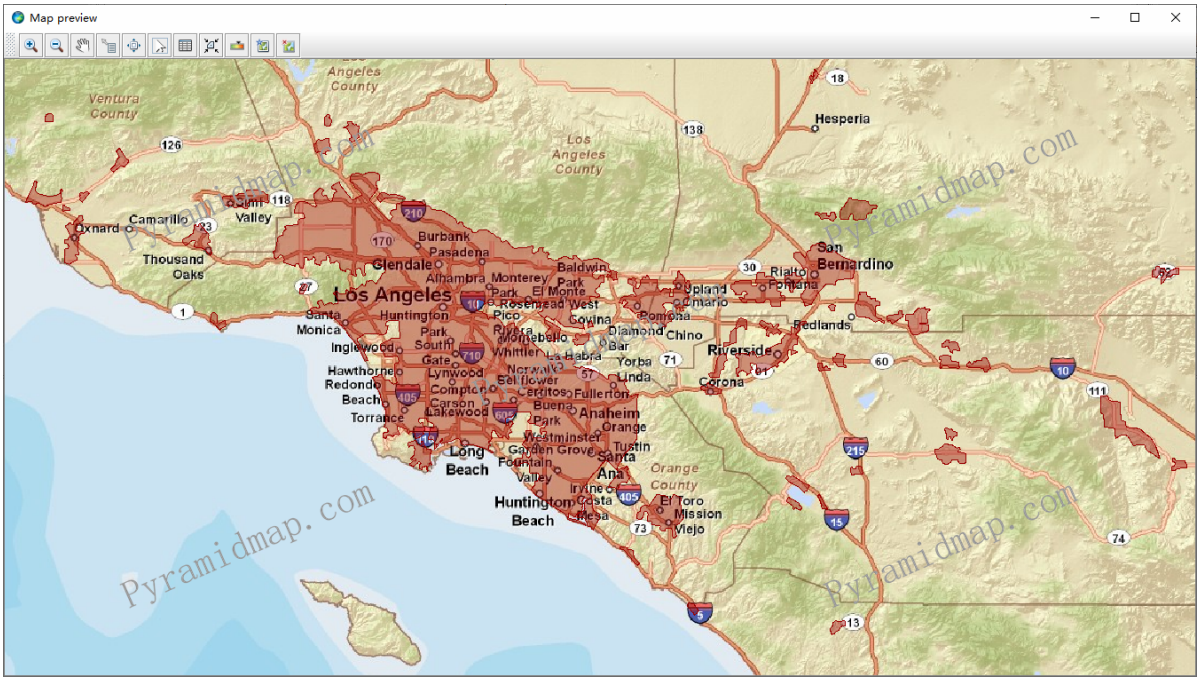


Figure 10-5: PyramidMap displays the UrbanAreas layer in Geodatabase

10.3: Layer Export

Taking the selected layer exported as JSON as an example for sampling, as shown in Figure 10-6.

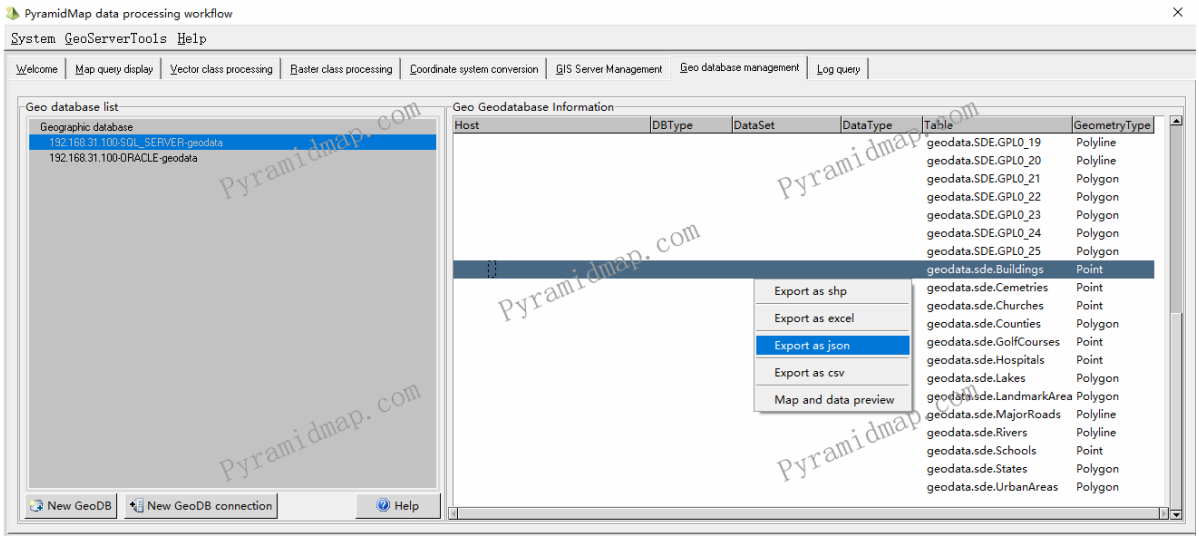


Figure 10-6: Export feature tables from the geodatabase to json

Select the data list, right-click to select the export type and path, you can create a data table to export to the specified path,export data support SHP,Excel, JSON, CSV format. PyramidMap provides cmd output monitoring of the running process, as shown in Figure 10-7.

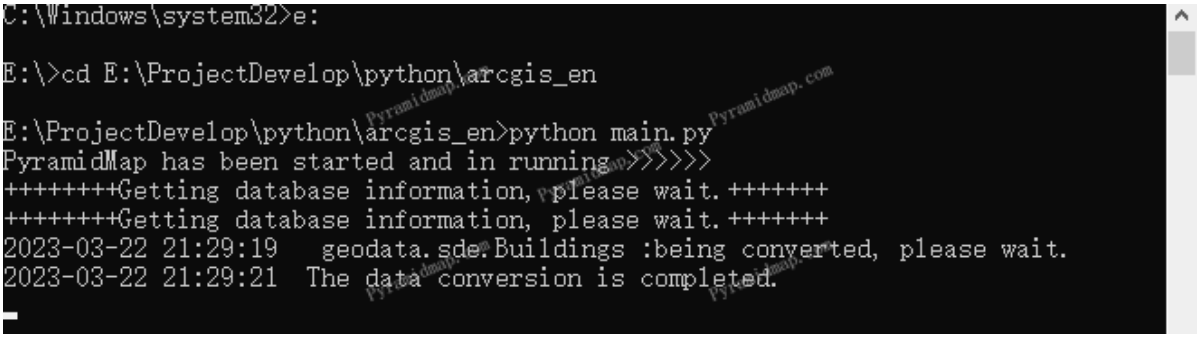


Figure 10-7: PyramidMap provides cmd output monitoring of the running process

PyramidMap map data export tips as shown in Figure 10-8.

Figure 10-8: The system message prompt for layer successful exported

The exported json data fragment is shown in Figure 10-9.

Figure 10-9: The exported json data fragment

11: GIS server

11.1: Layers List

The important role of GIS Server is to provide online map data for the web, various mobile terminals, and embedded devices, achieving map display and online editing functions. As a continuation of [9: Publish MapServer], PyramidMap can serve as a visualization client for ArcGIS Server to visualize the hosted layers in the server, including exporting Shp, Excel, GeoJson, Csv, and previewing maps. PyramidMap accesses the ArcGIS Server layer and performs the operation interface as shown in Figure 11-1.

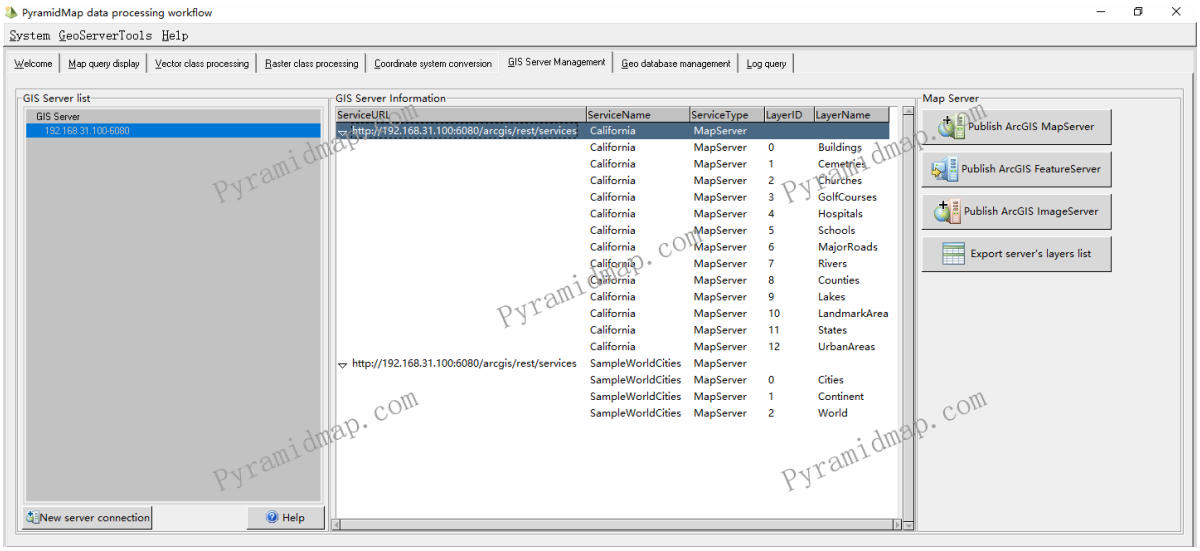


Figure 11-1: PyramidMap get the layers in ArcGIS Server

PyramidMap supports right-click shortcut operations for data export, conversion, and map preview of layers in the list as shown in Figure 11-2.

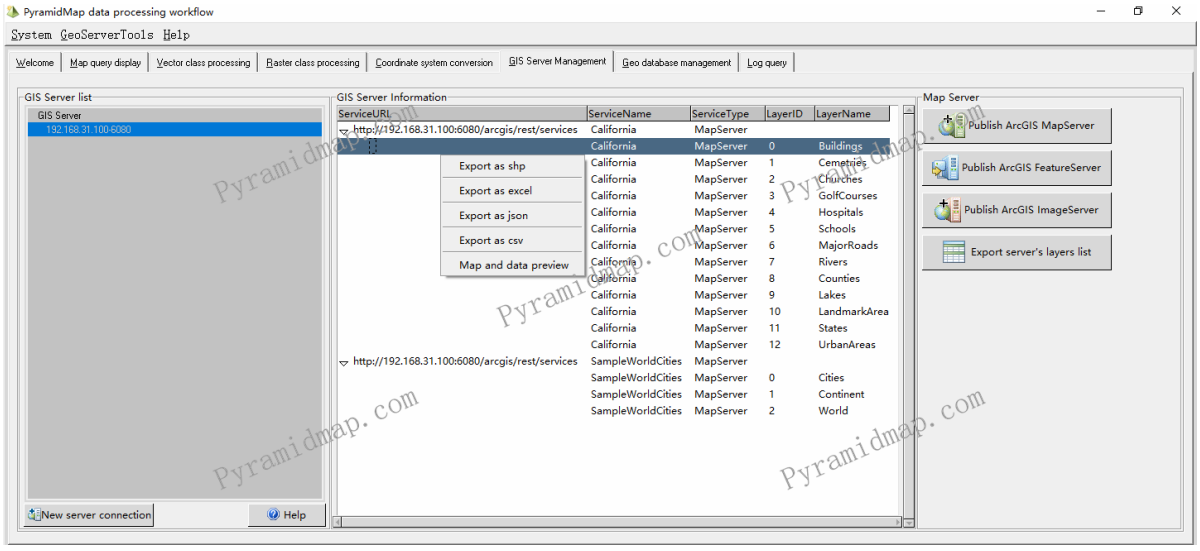


Figure 11-2: PyramidMap Operations on ArcGIS Server Layers

The database layer is exported to multi-format data such as shp, excel, json, and csv through the shortcut menu, and supporting layer's previewing. For the detailed process, see the chapter [11.2: Layer Preview] and [11.3: Layer Export].

11.2: Layer Preview

Right click on the layer in the GIS server list and select "Map and Data Preview", as shown in Figure 11-3.

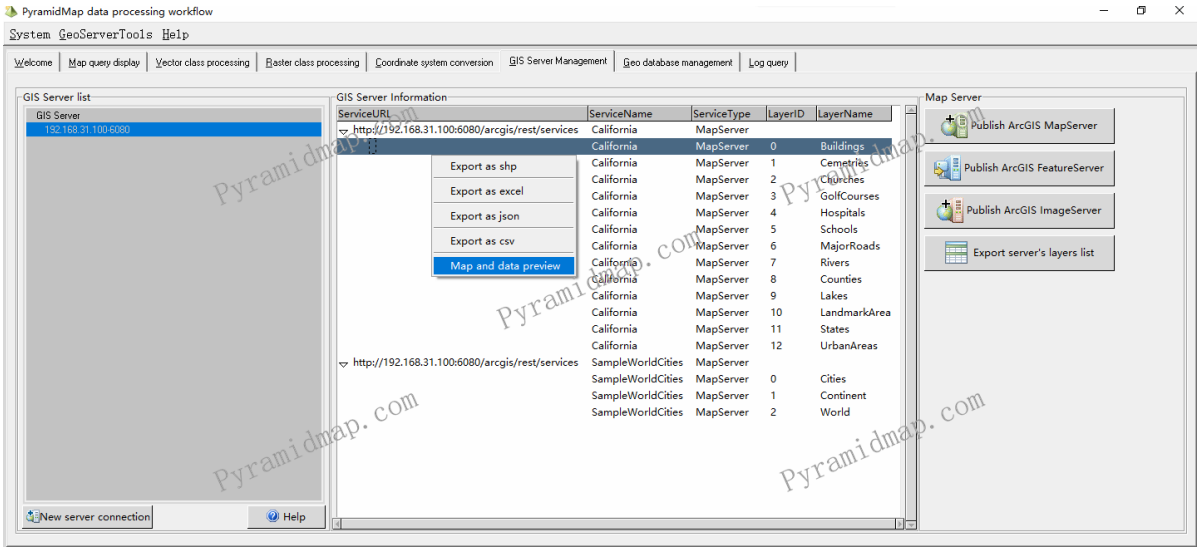


Figure 11-3: PyramidMap previews ArcGIS Server layers

Sample and display by feature geometry type. The preview effect of the selected Point type layer Building is shown in Figure 11-4.

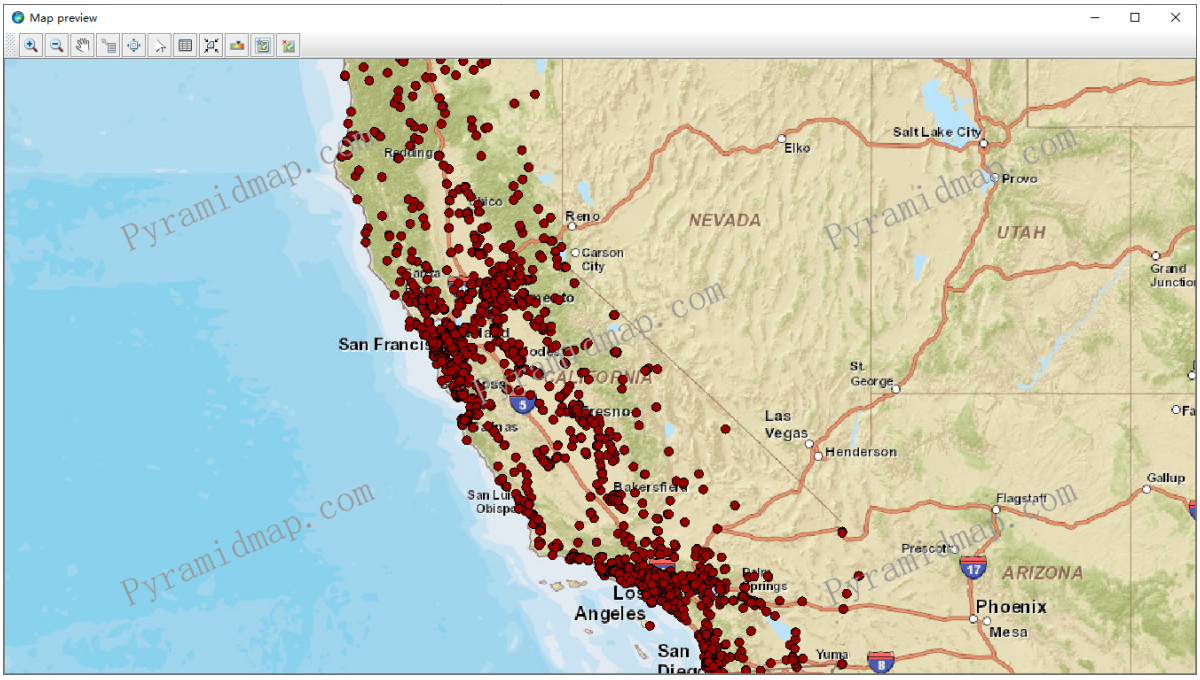


Figure 11-4: PyramidMap displays the Building layer in ArcGIS Server

The preview of the selected Polyline type layer MajorRoads is shown in Figure 11-5.

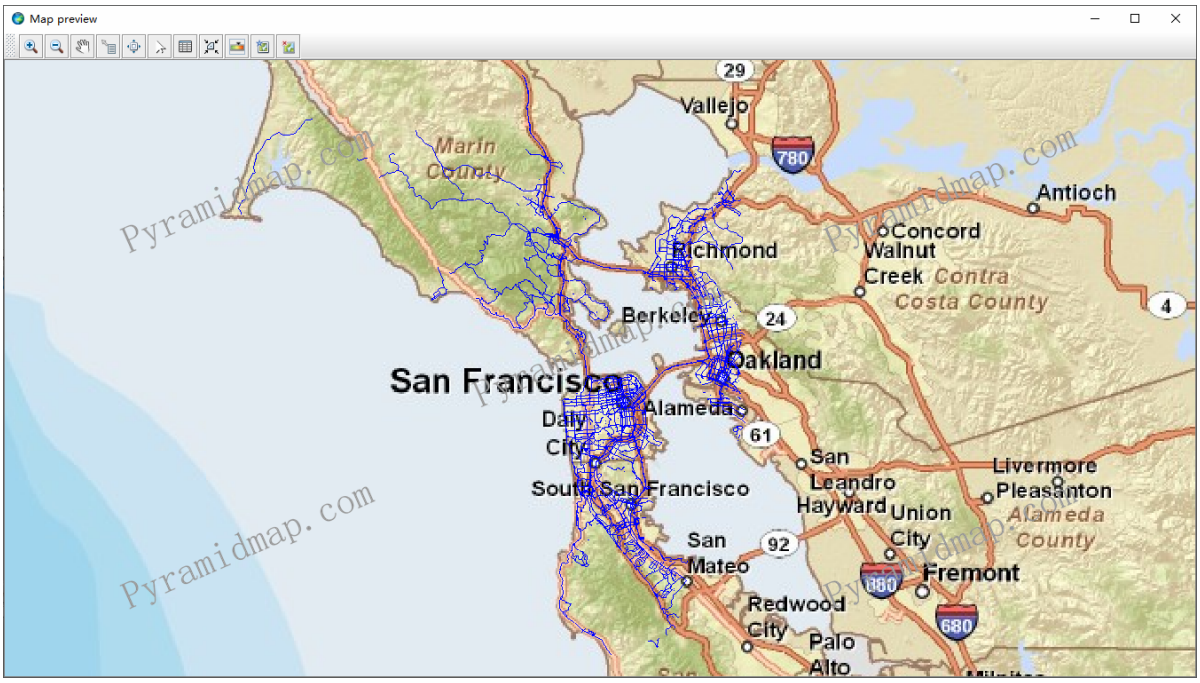


Figure 11-5: PyramidMap displays the MajorRoads layer in ArcGIS Server

The preview of the selected Polygon type layer UrbanAreas is shown in Figure 11-6.

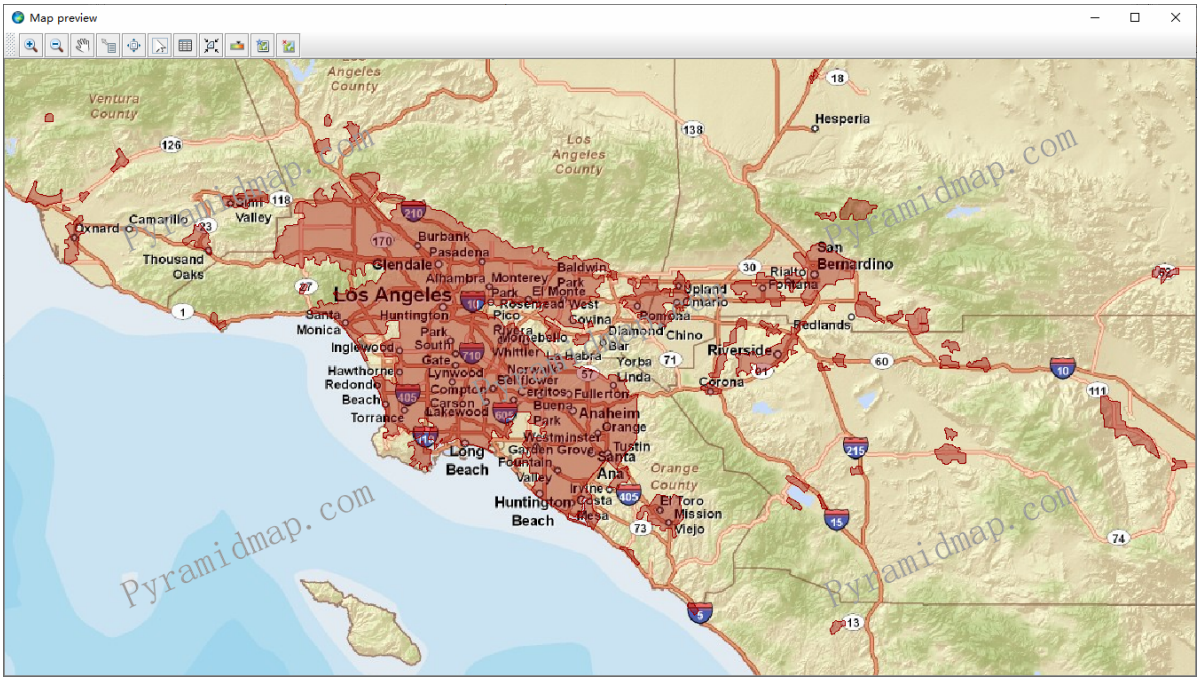


Figure 11-6: PyramidMap displays the UrbanAreas layer in ArcGIS Server

11.3: Layer Export

PyramidMap exports ArcGIS Server layers to various data types such as Shp, Excel, Csv, Json, etc. Sampling takes Excel as an example, as shown in Figure 11-7.

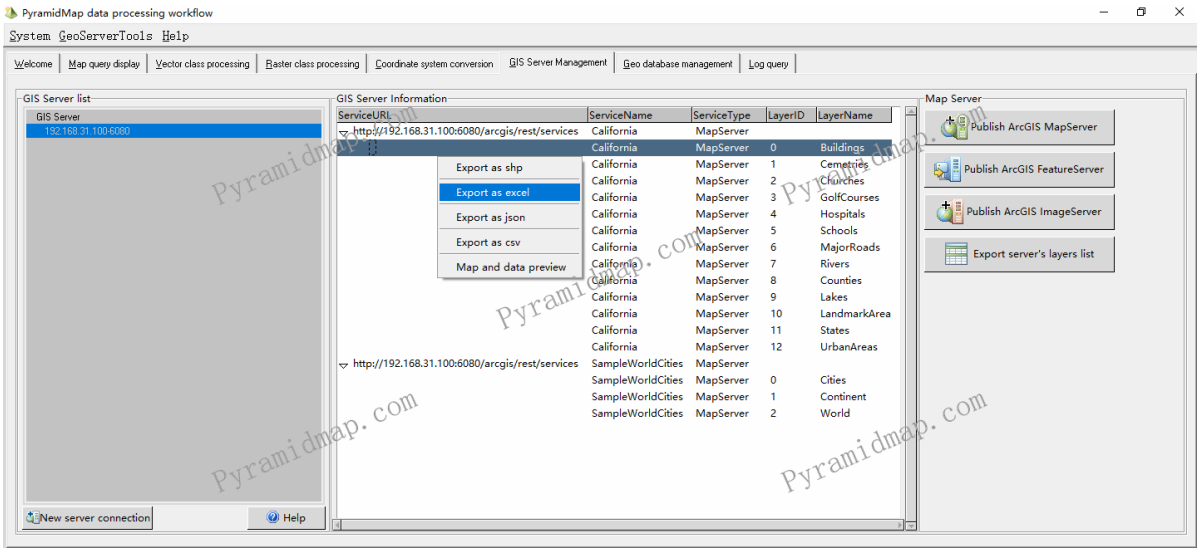


Figure 11-7: PyramidMap Exports ArcGIS Server Layer to Excel

Select the data list, right-click to select the export type and path, and export the data table to the specified path. PyramidMap provides cmd output monitoring for running processes, as shown in Figure 11-8.

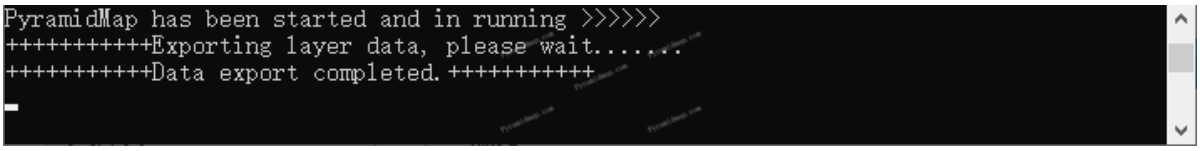


Figure 11-8: PyramidMap provides cmd output monitoring for exporting ArcGIS Server Layer to Excel

The Excel data exported from the selected layer is shown in Figure 11-9.

	A	B	C	D	E	F
1	FID	NAME	STCTYFIPS	ELEV_METER	LABEL_FLAG	
2	0	Yuma Territorial Prison	4027	47	0	
3	1	Colorado River Water Pollution Control Center	4012	115	0	
4	2	La Paz County Courthouse	4012	128	0	
5	3	Parker City Hall	4012	127	0	
6	4	Yuma City Hall	4027	49	0	
7	5	Yuma Courthouse	4027	52	0	
8	6	Colorado River Indian Agency Headquarters	4012	126	0	
9	7	McDonald Mountain House	6045	380	0	
10	8	Breckenridge Lodge	6029	1824	0	
11	9	Eleven P Office	6031	383	0	
12	10	Maryvale Orphanage	6037	106	0	
13	11	Soledad State Prison	6053	63	0	
14	12	Los Angeles City Fire Station	6037	386	0	
15	13	Anglers Lodge	6075	37	0	
16	14	Anne Bremer Memorial Library	6075	25	0	
17	15	Bohemian Club	6075	34	0	
18	16	Bourn Mansion	6075	80	0	
19	17	Brooks Exhibit Hall	6075	19	0	
20	18	Cable Car Barn and Museum	6075	61	0	
21	19	California Historical Society	6075	88	0	
22	20	Casa Ciele	6075	98	0	
23	21	Century Club of California	6075	59	0	
24	22	China Basin Building	6075	1	0	
25	23	Chinese Cultural and Trade Center	6075	12	0	
26	24	Civic Center Auditorium	6075	18	0	
27	25	Columbus Tower	6075	8	0	
28	26	Davies Symphony Hall	6075	20	0	
29	27	DeYoung Building	6075	13	0	
30	28	Embarcadero Center	6075	3	0	
31	29	Far West Library for Educational Research and Development	6075	7	0	
32	30	Federal Reserve Bank of San Francisco	6075	3	0	
33	31	Fire Station Number Two	6075	18	0	
34	32	Firehouse Number 44 (historical)	6075	92	0	
35	33	Flood Building	6075	13	0	
36	34	Fox Plaza	6075	18	0	
37	35	Fugazi Hall	6075	30	0	
38	36	George R Moscone Convention Center	6075	8	0	
39	37	Golden Gate Valley Branch Library	6075	34	0	
40	38	Hallidie Building	6075	14	0	
41	39	Hartford Building	6075	15	0	
42	40	Hills Plaza	6075	6	0	
43	41	Hobart Building	6075	9	0	
44	42	Japanese Cultural and Trade Center	6075	44	0	
45	43	Jessie Street Substation	6075	12	0	
46	44	Jewish Community Center of San Francisco	6075	83	0	
47	45	Koshland Mansion	6075	87	0	
48	46	Lick Baths (historical)	6075	14	0	
49	47	Marina Branch San Francisco Public Library	6075	11	0	

Buildings

Figure 11-9: The excel data table exported by PyramidMap from ArcGIS Server Layer

12: Log query

12.1: Log category directory

PyramidMap logs and queries all map operations by function classification, as shown in Figure 12-1.

Figure 12-1: PyramidMap logs and queries all map operations by function classification

12.2: Log query

PyramidMap performs log query on all map operations by function classification, as shown in Figure 12-2.

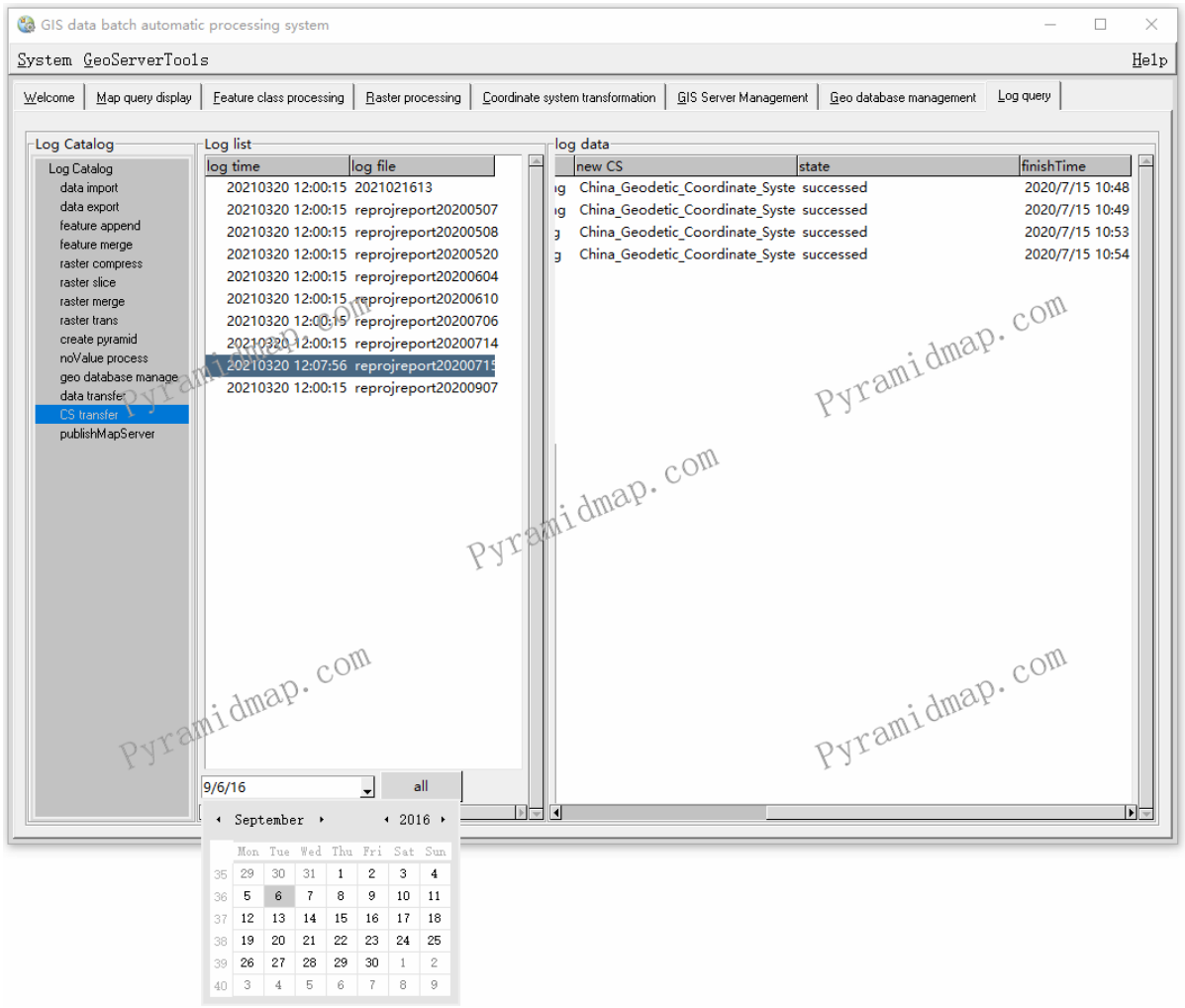


Figure 12-2: PyramidMap logs and queries interface