

SuperMap iServer Map Cache

SuperMap Software Co., Ltd.



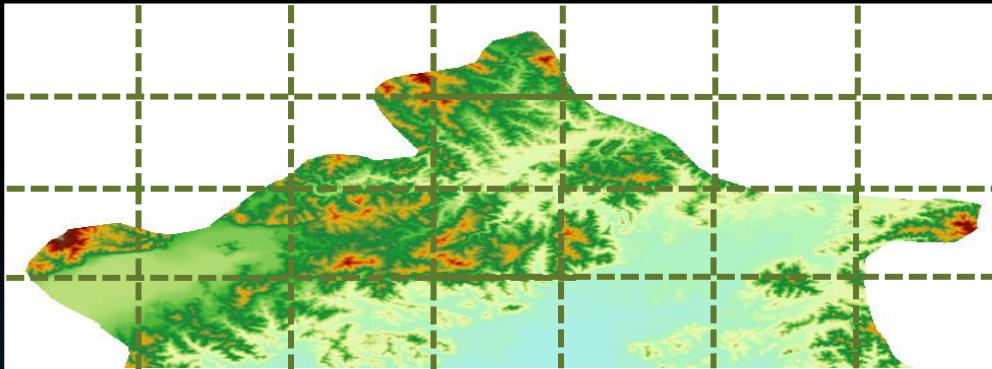
Main Contents

- **What is map cache?**
- **Why map cache?**
- **Mechanism of map cache.**
- **How to create map cache?**
- **How to use map cache?**
- **Suggestions for using map cache.**

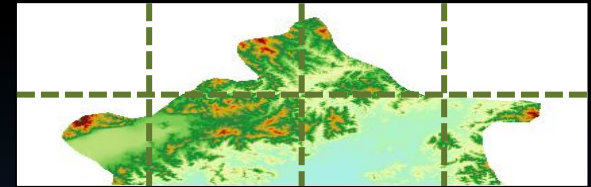
Decleration: all the 'iServer' mentioned in the ducument refers SuperMap iServer products if no other specific name mentioned

What is map cache?

- The pre-produced map tiles which can improve the map access efficiency



1:1000000

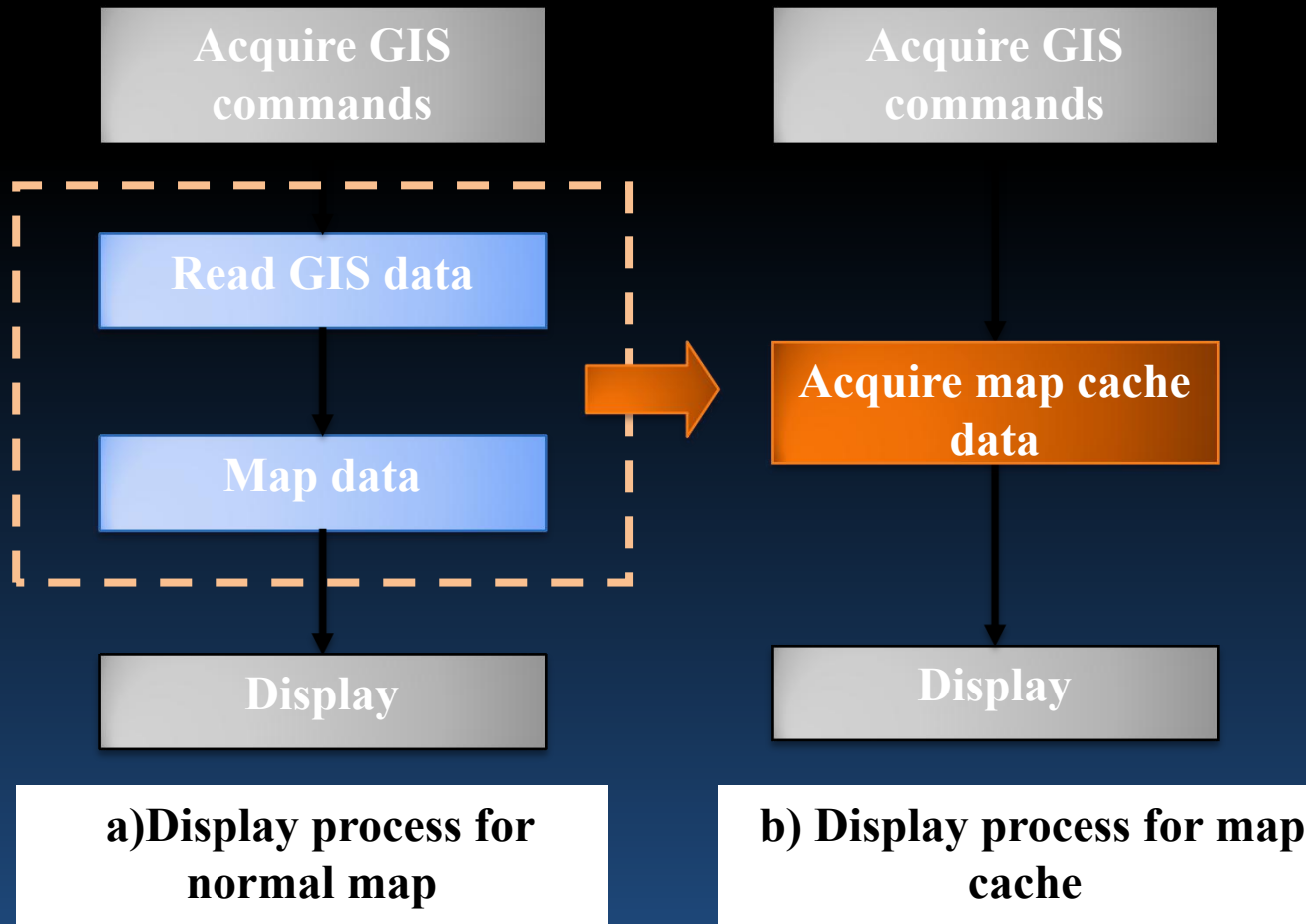


1:2000000

- Produce map cache according to pre-defined map scale.
- Produce according to tiling algorithm.

Why map cache?

- Cache technology

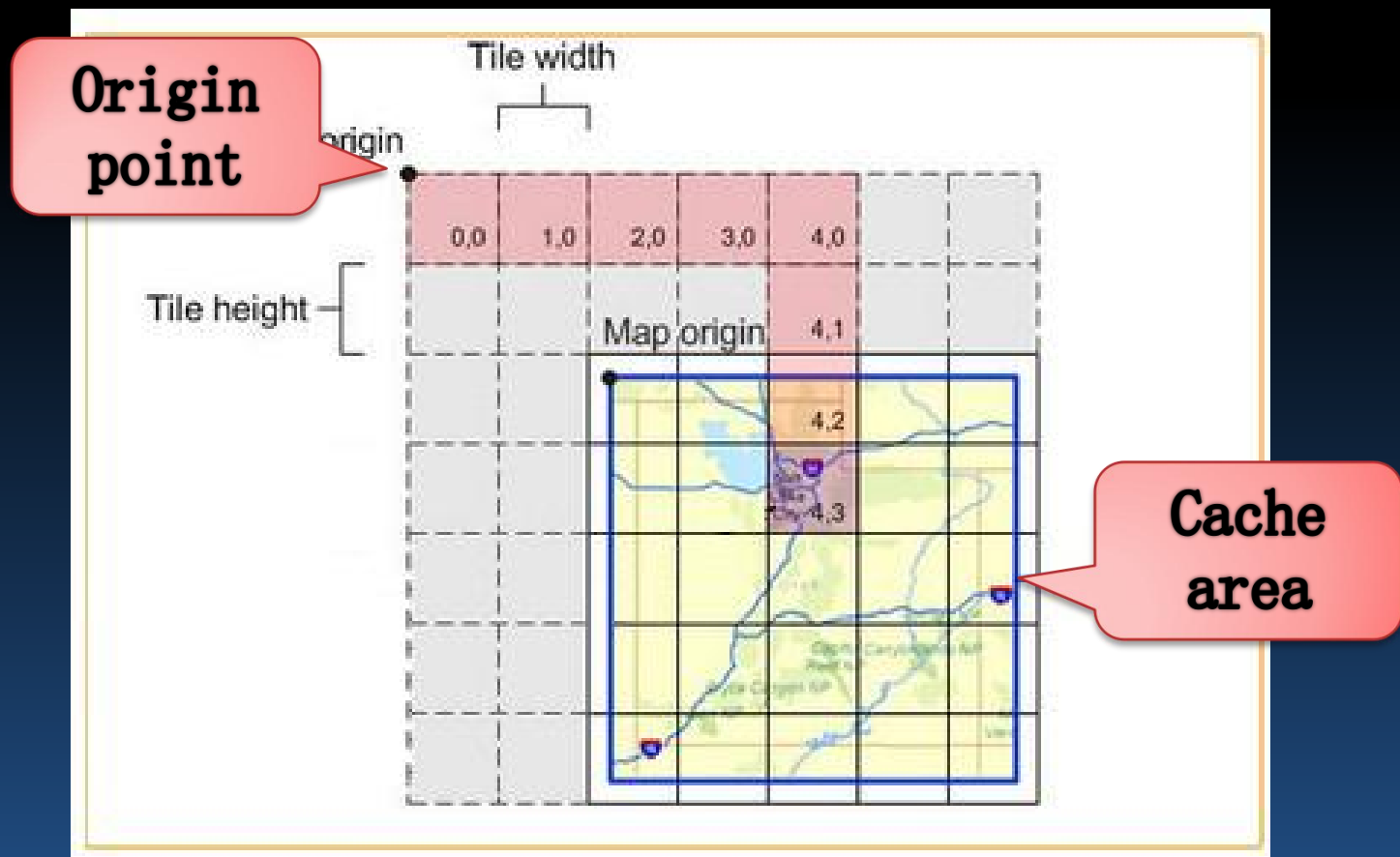


Cache Model

- **SuperMap products have unified cache model**
 - **From iDesktop to component products SuperMap iObjects Java/.Net, and server product iServer, they all have the same map cache model.**

Cache Construction Mechanism

- Construct cache through tile files generated by dividing layer and block, and consider the cache update problems from perspectives of coordinate system, map, layer style and data range



Map Cache Mechanism

Static cache mechanism

- Pre-cache
- SuperMap iServer directly use pre-produced cache data to respond to the mapping request from the client side, saving the time of server.
- For massive GIS data, they have lower update frequency, using cache mechanism is recommended.

Dynamic cache mechanism

- SuperMap iServer dynamically generate cache in map viewing, and provide quick response for the same requests in the future.

Cache Storage Mechanism

File storage

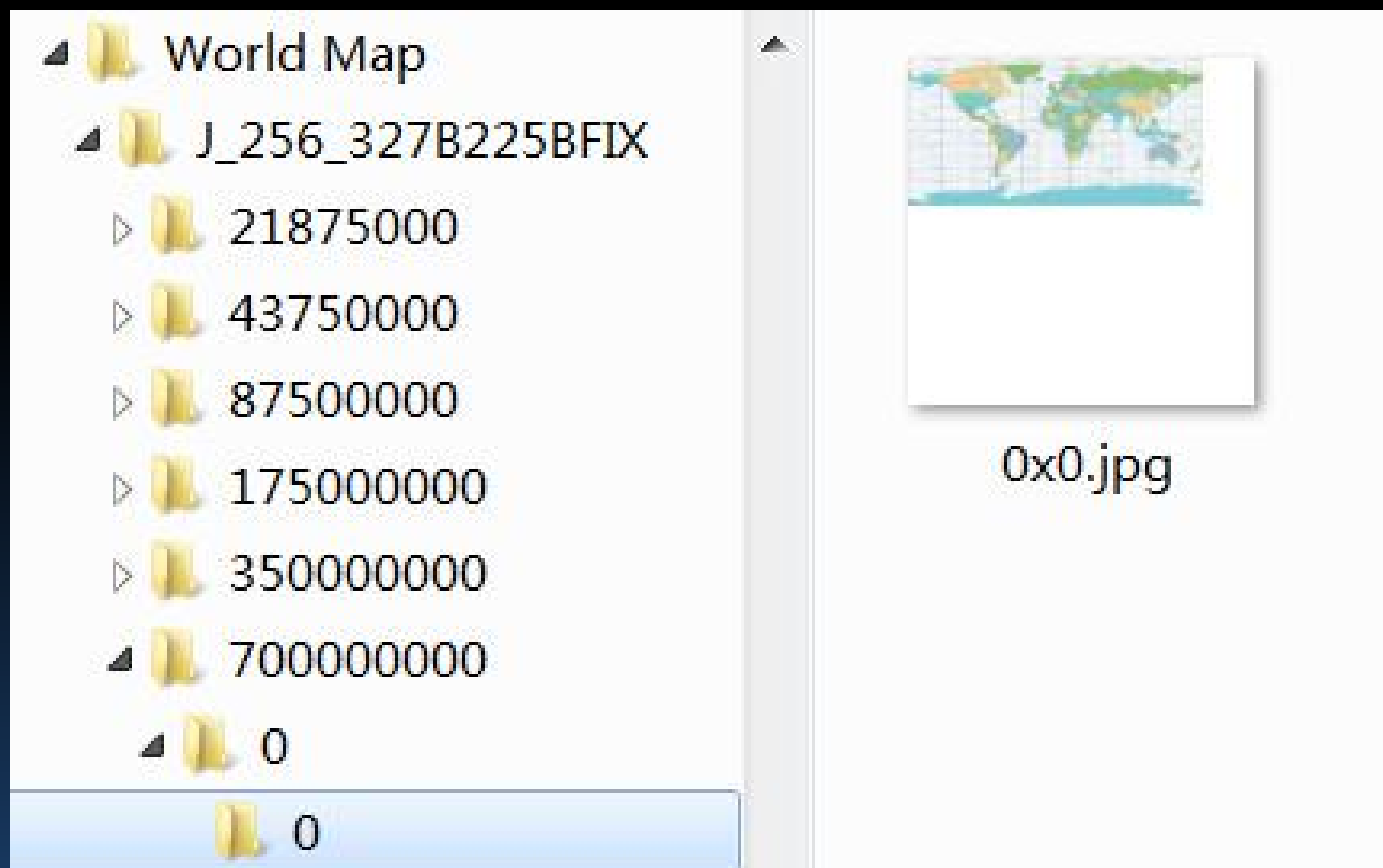
- Original picture
- Compact file

Database storage

- MBTiles
- FastDFS
- MongoDB

Original Picture Cache

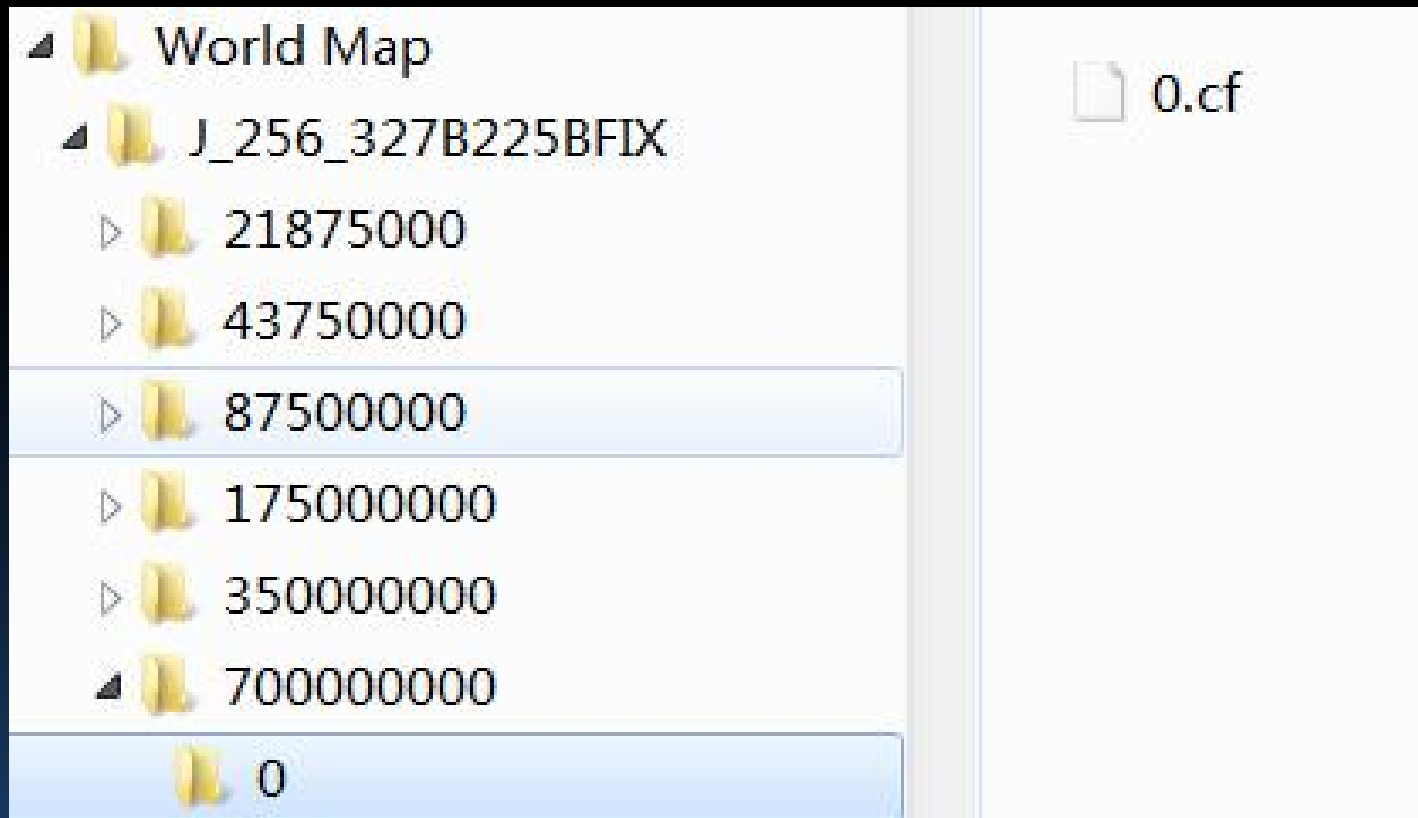
- Cache folder structure



V5.0 Cache structure

Compact File Cache

- Cache folder structure



V5.0 Cache structure

Comparison

- **Original picture cache**
 - **Readable**
 - **More utilization**
 - **Time consuming when copying and deploying**
- **Compact file cache**
 - **Less readable**
 - **Easy to copy and deploy**

MBTiles Cache

- **MBTiles** is a standard that putting tile map data stored into SQLite database and quickly use, manage and share.
- **Cache format:**



China_69470548_256X256_PNG.mbtiles

- **When there is MBTiles cache in the service component, the map will be acquired directly from cache other than service provider.**
- **MBTiles cache are usually used in offline cache in mobile terminal.**

FastDFS Cache

- **FastDFS is an open source lightweight distribution file system, which can manage the files, solving the problems of big volume storage and loading balance.**
- **FastDFS cache can do parallel tiling on multiple machines, distributed storage to improve the efficiency of cache.**
- **For service layer cache**

How to Create Cache

- **Generated pre-cache by SuperMap iDesktop**
- **Generate cache by iServer pre-cache server**
- **Generate by iServer distribution tiling service**

Create Map Cache

- Use SuperMap iDesktop to create cache
 - Select map. Click ‘create map cache’ in right-click menu

No.	Scale	Label
1	1:700000000	700000000
2	1:350000000	350000000
3	1:175000000	175000000
4	1:87500000	87500000
5	1:43750000	43750000
6	1:21875000	21875000
*		

Set scale

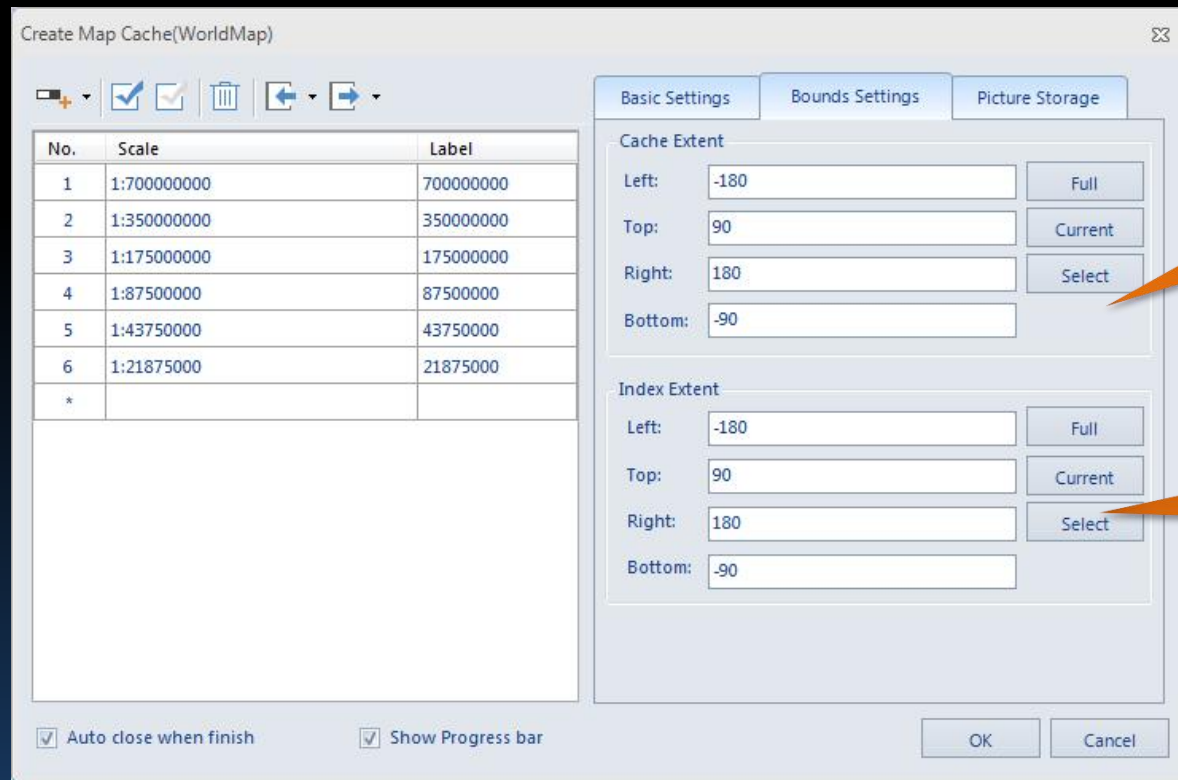
Select cache version

5.0 supports cache appending

Set cache path

Create Map Cache

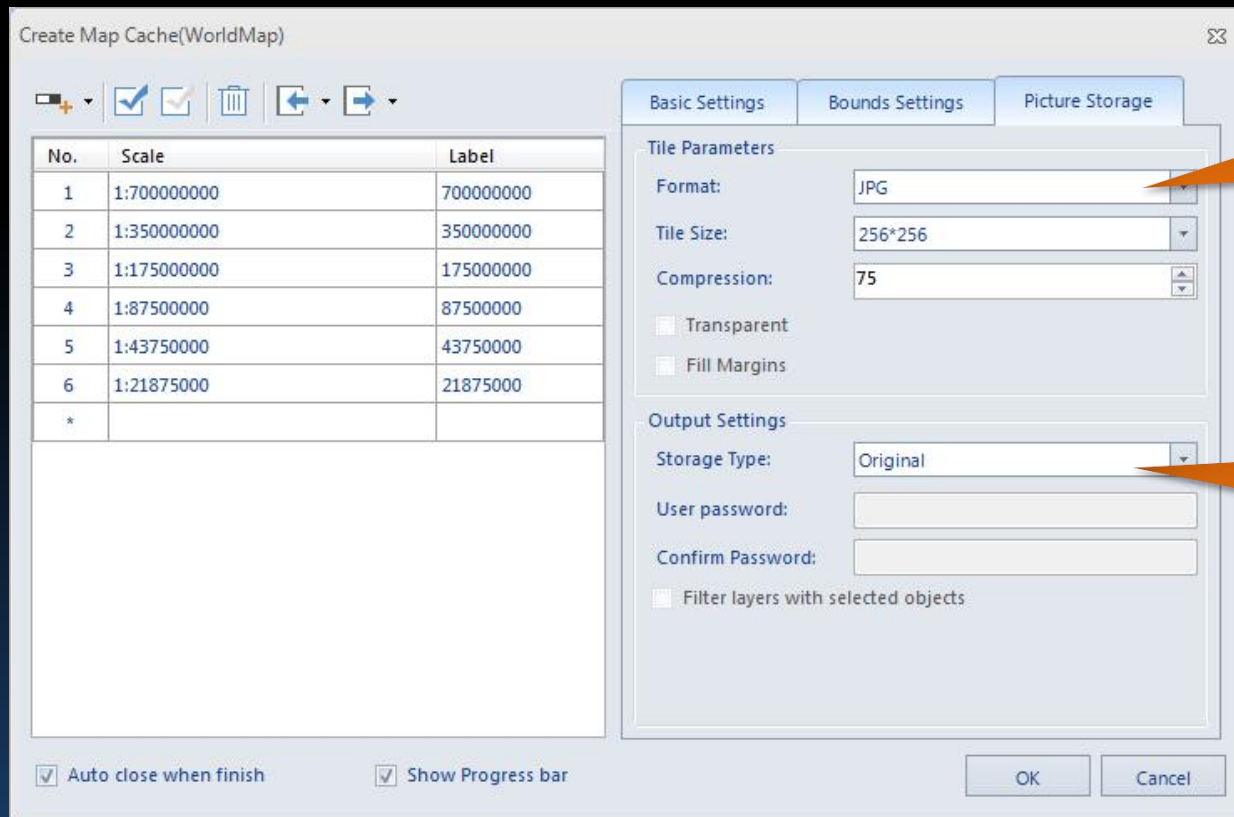
- Use SuperMap iDesktop to create map cache



Index range directly influence the location of tiles, when appending cache, making sure the ranges are the same.

Create Map Cache

- Use SuperMap iDesktop to create cache



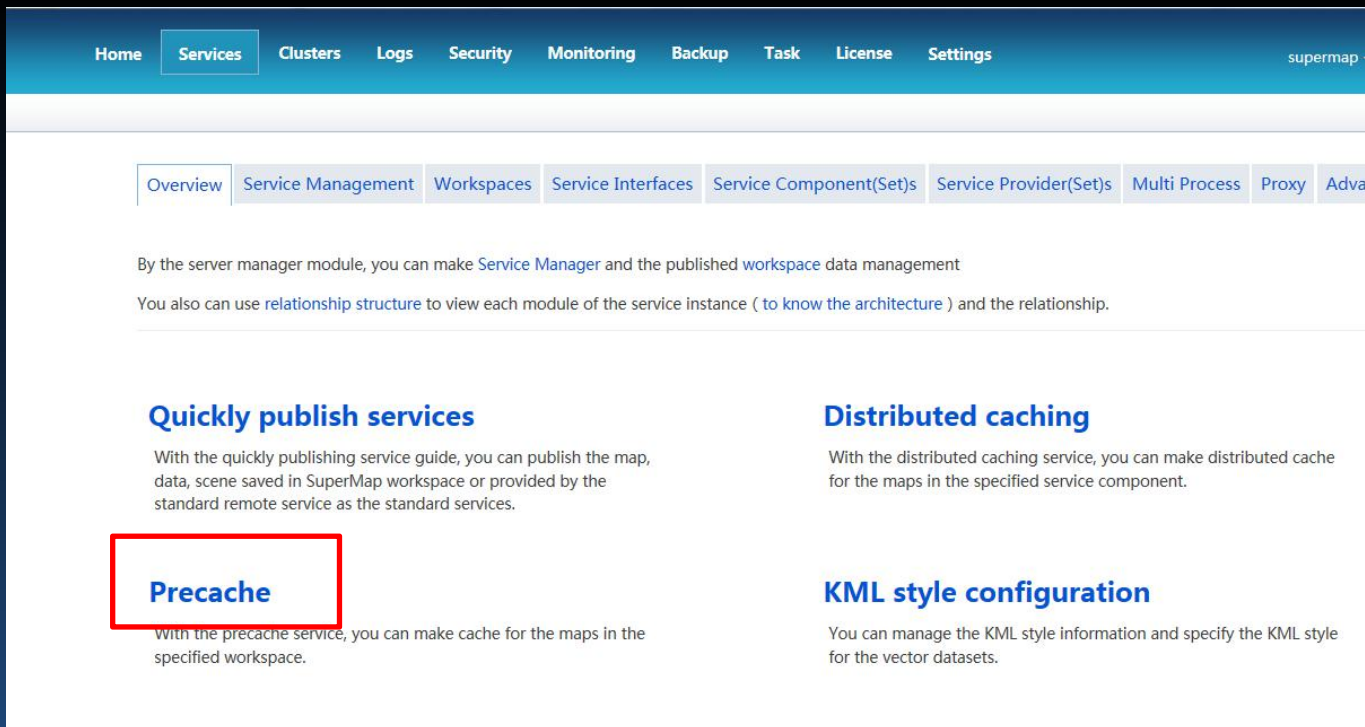
Set pic parameter

Choose storage type

No need to encrypt cache when generating compact map cache.

Create Pre-cache

- Create cache in iServer pre-cache server.
- Also can create MBTiles cache
- Method: service management tool->service->overview->pre-cache



The screenshot shows the SuperMap iServer web interface. The top navigation bar includes links for Home, Services, Clusters, Logs, Security, Monitoring, Backup, Task, License, and Settings. The 'Services' menu is expanded, showing sub-options: Overview, Service Management, Workspaces, Service Interfaces, Service Component(Set)s, Service Provider(Set)s, Multi Process, Proxy, and Advanced. The 'Overview' sub-menu is selected, displaying a description of the server manager module and its capabilities. Below the description, there are four main sections: 'Quickly publish services', 'Distributed caching', 'Precache', and 'KML style configuration'. The 'Precache' section is highlighted with a red rectangular box. The 'Precache' section text reads: 'With the precache service, you can make cache for the maps in the specified workspace.'

Home Services Clusters Logs Security Monitoring Backup Task License Settings supermap

Overview Service Management Workspaces Service Interfaces Service Component(Set)s Service Provider(Set)s Multi Process Proxy Advanced

By the server manager module, you can make Service Manager and the published workspace data management

You also can use [relationship structure](#) to view each module of the service instance ([to know the architecture](#)) and the relationship.

Quickly publish services

With the quickly publishing service guide, you can publish the map, data, scene saved in SuperMap workspace or provided by the standard remote service as the standard services.

Distributed caching

With the distributed caching service, you can make distributed cache for the maps in the specified service component.

Precache

With the precache service, you can make cache for the maps in the specified workspace.

KML style configuration

You can manage the KML style information and specify the KML style for the vector datasets.

Create Pre-cache

- Create cache in iServer pre-cache server.
 - SuperMap iServer takes a process of map pre-cache generation as a task.
 - Multiple pre-cache tasks can run simultaneously in a pre-cache task list, however, only one task can take place in the tasks under the same map.

Add/Edit precache task

Workspace: *

Map:

Image size:

Image format:

Cache version: (Refer to [Cache Scheme](#))

Default: For all versions of iServer6R.

Cache scales:

-
-
-
-
-
-

Transparent:

Storage type:

Specify cache bounds:

Configured to generate other tiles packages

Create MBTiles:

createUTFGridCache:

createVectorTileCache:

Precache Tasks

Map name	Task info	Progress	Operation
WorldMap_Day	Size: 256x256 Format: Original Cache type:4.0 Type: PNG Workspace: ../samples/data/World/World.sxwu	There are 92829469 images. Finished: 338 images. <input type="text" value="84 hours, 36 minutes, 18 seconds"/>	<input type="button" value="Stop"/>

Pre-cache service supports adding scale, but please pay attention that all the pictures will be re-produced after editing

Create Map Cache

- **Create cache in iServer pre-cache server**
 - **For tile cache**
 - Cache will automatically be stored in [installation folder]\webapps\iserver\output\cache
 - **For MBTiles cache**
 - Cache will automatically be stored in [installation folder]\webapps\iserver\output\sqlite
 - Cache generated by iServer pre-cache server does not need to deploy the cache path.

Distributed Tiling Service

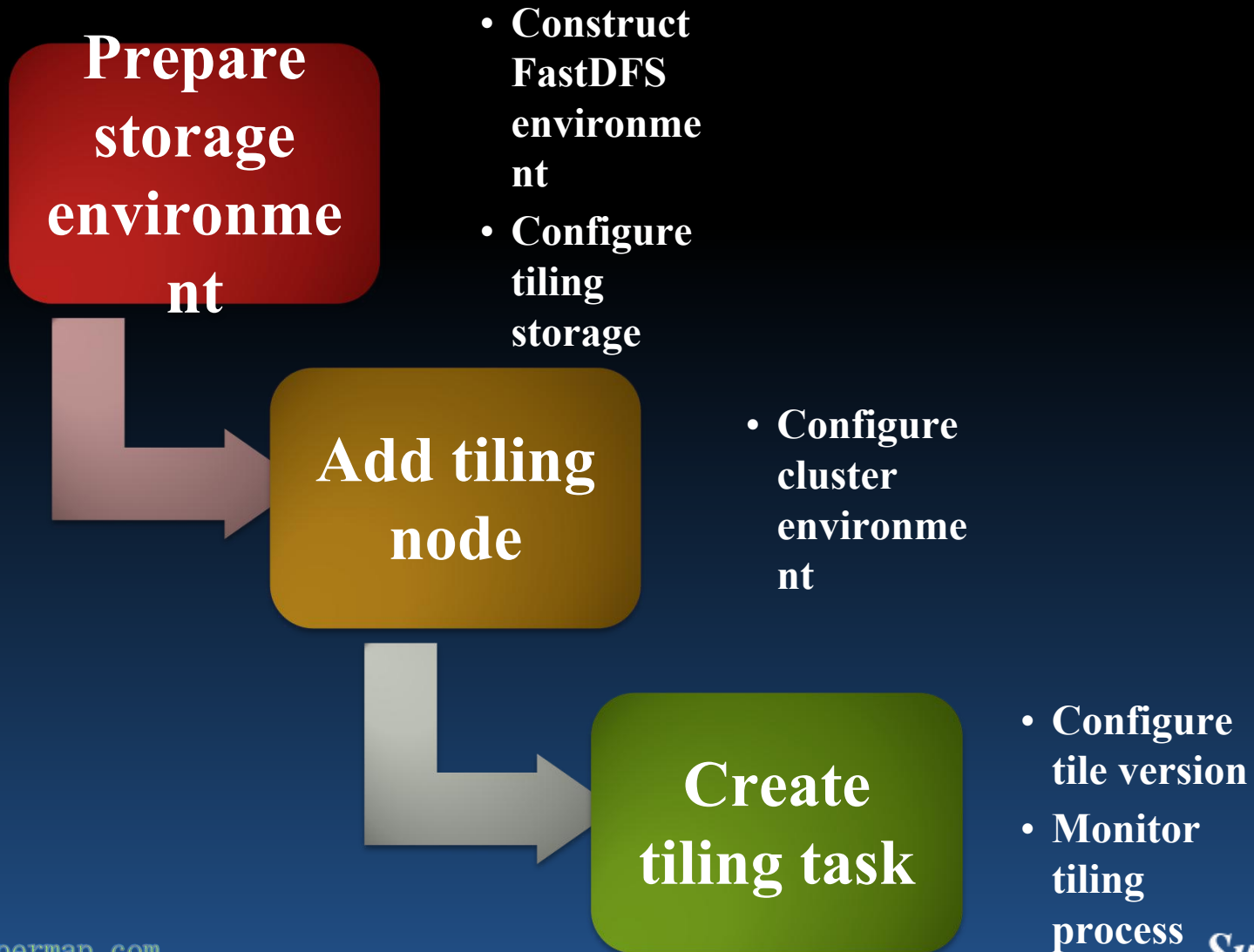
- **SuperMap iServer raised distributed map tile creation and dispatching technology.**
 - Through one **TileMaster** and multiple **TileWorker** to construct distributed map tiling system.
- **Distributed tiling technology supports distributed tiling and distributed storage management of map tiles.**
 - Supports multiple distributed file system, NAS and big data storage system (like FastDFS, MongoDB)

Create Distributed Tiling FastDFS

- Use iServer distributed tiling service to create FastDFS cache.
- iServer supports parallel tiling on multiple machines, can add multiple tiling nodes on different machines.
- Can effectively avoid the disadvantages of traditional cache tiling technology, like long response time, no malfunction disorder, etc.

Create Distributed Tiling FastDFS

- Steps



Prepare Storage Environment

- **Construct FastDFS**
 - **Version requirements**
 - **FastDFS 4.00 or above**
 - **FastDHT 1.21 or above**
 - **Libevent 1.4.x or above, latest version stable is recommended.**
 - **Berkeley DB 5.3 or above**
 - **FastDFS supports UNIX systems like Linux, FreeBSD**
 - **FastDFS needs to collaborate with FastDHT, which is an effective Hash system based on key value pair, it can be used to store massive key value pair, like file name mapping sheet, session data, user related data, etc.**
 - **The installation of FastDHT depends on libevent and oracle Berkeley Db**

Prepare Storage Environment

- Steps of installing FastDFS



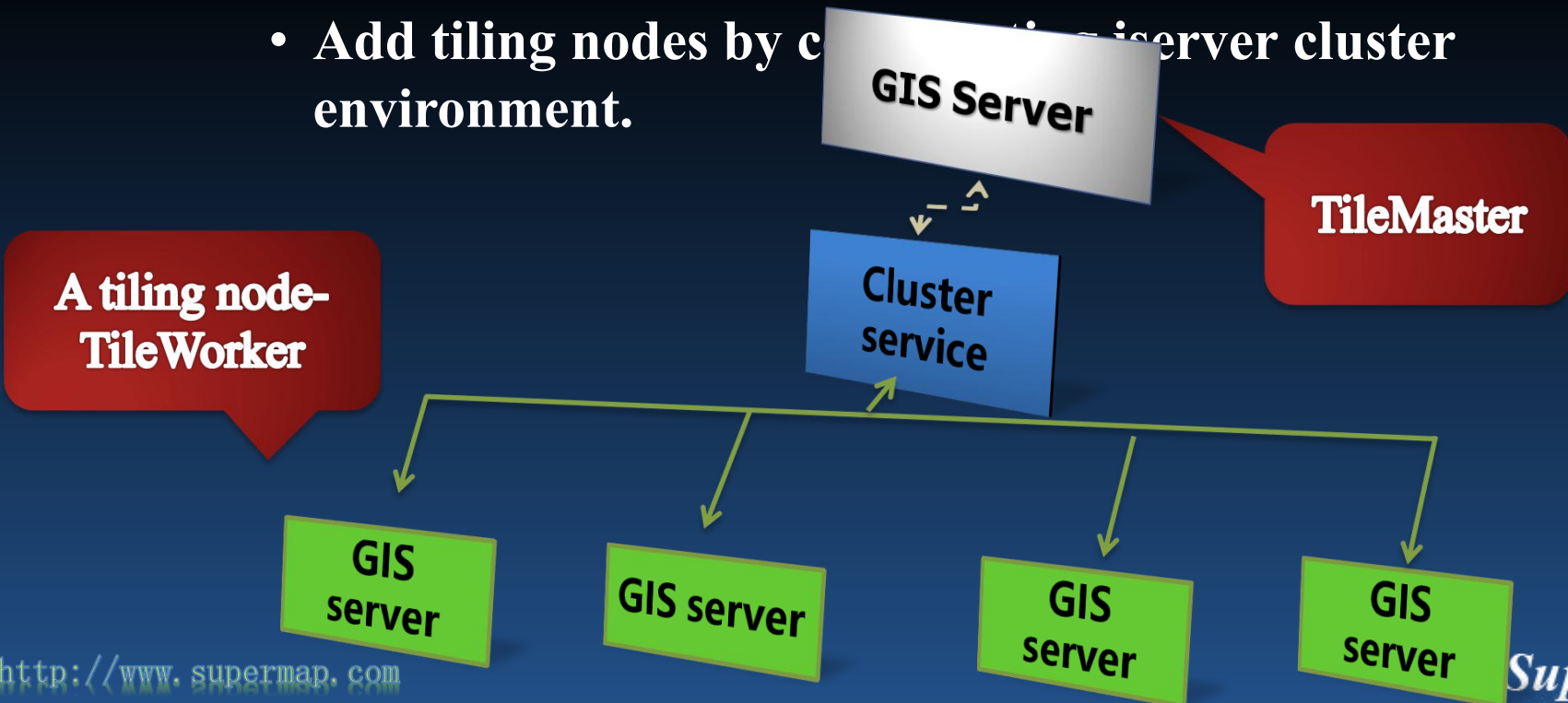
Refer the help document for more information

Add Tiling Nodes

- **Terms**
 - **TileMaster: Create nodes of tiling tasks**
 - **Main function is to divide tiling tasks into multiple unit task according to scale and geographic range, and dispatch, manage tiling units, assign to TileWorker**
 - **TileWorker: Add nodes which execute tiling task**
 - **TileWorker executes the tasks dispatched by TileMaster**
- **The relation between TileMaster and TileWorker is one to more.**

Add Tiling Nodes

- **Tiling mode**
 - Tiling on the single machine (**TileMaster = TileWorker**)
 - Default mode, no operation needed.
 - Parallel tiling on multiple machine
 - Add tiling nodes by creating a tiling server cluster environment.



Add Tiling Tasks

- View the added tiling nodes information in TileMaster
 - TileMaster management tool->Home>>Distributed tiles

The screenshot shows the TileMaster web interface. The top navigation bar includes 'Home', 'Services', 'Clusters' (selected), 'Logs', 'Security', 'Monitoring', 'Backup', 'Task', 'License', and 'Settings'. On the right, there are links for 'supermap', 'Help', and 'Engl'. Below the navigation bar, there are tabs for 'Cluster Overview', 'Use Cluster', 'Configure Cluster', 'Join Cluster', 'Distributed Tiles' (selected), and 'Distributed Tiles Repository'. The main content area contains two paragraphs of text describing the distributed tiling service. To the right of the text are two buttons: '+ Create tiling task' and '+ Create GDP tiling task'. Below the text, there is a section titled 'Task being implemented:' with a progress bar for 'usmap' at 0%. The progress bar is accompanied by the text '数据预处理中, 已完成: 0张 (共: 2500张); 运行时间: 0秒; 速度: 0 张/秒;'. To the right of the progress bar are 'Start' and 'Delete' buttons. Below this section is a section titled 'Finished task:' followed by a table with the following columns: 'Map name', 'Component name', 'Start time ^', 'Time cost for task', 'Tile count', 'Tile type', and 'Storage type'.

Home Services **Clusters** Logs Security Monitoring Backup Task License Settings supermap ▾ Help Engl

Cluster Overview Use Cluster Configure Cluster Join Cluster **Distributed Tiles** Distributed Tiles Repository

Distributed tiling service supports parallel map tiling on multiple machines. Tiling nodes from different machines can be added for a common tiling task.

Distributed tiling service is parallel map tiling function based on cluster. Tiling nodes can be added through adding [cluster members](#).

+ Create tiling task
+ Create GDP tiling task

Task being implemented:

usmap 0% 数据预处理中, 已完成: 0张 (共: 2500张); 运行时间: 0秒; 速度: 0 张/秒; Start Delete

Finished task:

Map name	Component name	Start time ^	Time cost for task	Tile count	Tile type	Storage type
----------	----------------	--------------	--------------------	------------	-----------	--------------

Add Tiling Tasks

- Add tiling tasks in distributed tiling page of TileMaster

The screenshot shows the 'Add tile task' form in the TileMaster interface. The navigation bar includes Home, Services, Clusters, Logs, Security, Monitoring, Backup, Task, License, and Settings. The sub-navigation bar includes Cluster Overview, Use Cluster, Configure Cluster, Join Cluster, Distributed Tiles, and Distributed Tiles Repository. The form fields are: Service component (UGCMapCom-testus), Map (usmap), Tile type (Image), Storage type (SMTiles), Storage location (E:/Deskpro Versions/supermap_1server_8), Scale scheme (Recommended scale), and Cache scale (empty). A scale range selector is set to 2: 1/64000000. A list of scale options is shown: 1/64000000, 1/32000000, 1/16000000, 1/8000000, 1/4000000, 1/2000000, 1/1000000. A 'Tiles count: 4012' is displayed. The 'Advanced settings' section includes: The main node joins in tiling task: Yes; Cache bounds: -2356258.8299719472, -1333372.562106357, 2258927.6567376414, 1565161.9463572262; Original point: X: -2356258.8299719472, Y: 1565161.9463572262; Tile size: 256*256; Picture format: PNG; Transparent: No; Enable auto avoiding: No; Enable data pre-processing: Yes; Grid row: 50, column: 50; Convert picture into PNG8: Yes. Buttons for 'Create tile task' and 'Cancel' are at the bottom.

The screenshot shows the 'Add storage location' form in the TileMaster interface. The navigation bar and sub-navigation bar are the same as in the previous screenshot. The form fields are: Storage ID: * (empty), Tiles storage type: * (FastDFS), and FDHT Groups: |> * (empty). A tooltip for 'FastDFS' says 'How to prepare FastDFS environment'. There are 'Add' and 'Remove' buttons for the FDHT Groups. Buttons for 'Add storage location' and 'Cancel' are at the bottom.

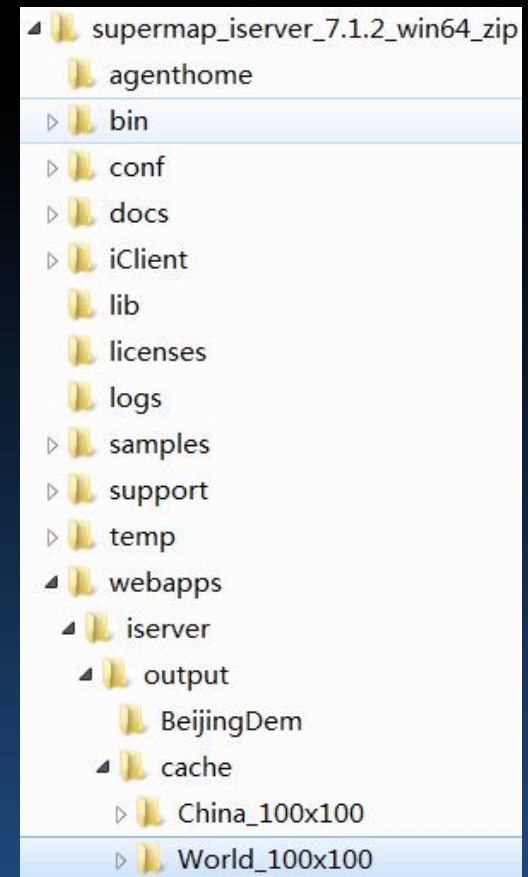
Create Distributed Tiling MongoDB

- **MongDB can be used for map tile storage**
- **Same steps as FastDFS**
- **Store different databases**

Please refer to iServer help document

How to use cache

- Usage of static cache (pre-cache)
 - Copy pre-cache generated by SuperMap iDesktop (Original/compact) to [iServer installation folder] \webapps\iserver\output\cache



How to use cache

- Service component cache configuration

Add service component

Service component name: *

Service component alias:

Service component type: *
Map Component

Used service provider/set:

Name of service provider/set	Selected
ugcMapProvider-China400	<input type="checkbox"/>
ugcMapProvider-World	<input checked="" type="checkbox"/>
ugcMapProvider-Changchun	<input type="checkbox"/>
ugcMapProvider-Jingjin	<input type="checkbox"/>
ugcMapProvider-temperature	<input type="checkbox"/>
map-ChinaProvinces	<input type="checkbox"/>
ugcMapProvider-testus	<input type="checkbox"/>
ugcMapProviderSet	<input type="checkbox"/>

Interface bound to component: *

Name of bound interface	Selected
wms111	<input type="checkbox"/>
wms130	<input type="checkbox"/>
rest	<input checked="" type="checkbox"/>
wmts100	<input type="checkbox"/>
wmts-china	<input type="checkbox"/>

GeneralSetting

Enable map tile caching:

Storage type: SMTiles

Storage location: SMTiles, FastDFS, MongoDB

Enable attribute tile caching:

Enable vector tile caching:

Whether cache is read-only:

Cache survival time: 0

How to use cache

- The client side controls if use cache

```
//创建图层对象
```

```
layerWorld = new SuperMap.Layer.TiledDynamicRESTLayer("World", url, {  
transparent: true, cacheEnabled: true }, { maxResolution: "auto", scales: [1/  
5000000, 1/2500000, 1/500000, 1/250000] });
```

Notice

- **When to use cache?**
 - **Massive data, especially with image data**
 - **Data with low update frequency.**
- **Choose suitable scale**
 - **Set more scale levels to have a smoother displaying when viewing**
 - **Consider if the elements need to be displayed in different scales**

Notice

- **Cache path setting**
 - **%SuperMap iServer**
_HOME%\webapps\iserver\output\cache
 - **When using SuperMap iDesktop to generate cache, can directly use the path mentioned above.**
- **Map creation**
 - **Consider Anti-Aliasing on line or text type**
 - **Filter small objects**
 - **For complicated map, avoid too many objects, consider resampling.**

Notice

- **Keep the same workspace**
 - The parameters may influence include map style, data connection status, layer order, default scale, map range, etc.
 - Cannot have the connection layer without datasource or dataset.
 - When editing objects in dataset, like adding, deleting, updating, etc., doesn't influence the cache picture outside the editing area.
- **Set map cache range and index range**
 - It is recommended setting index range as the same as extent map range of published map, being consistent with search index of SuperMap iServer.

Notice

- **Cached pictures and output devices**
 - **Different devices have different resolutions, so the cache on different devices may not work together, like map joining.**
 - **Configure map cache and device resolution as irrelevant**
 - **Change CustomDPIEnable attribute in SuperMap.xml file under bin folder, SuperMap iDesktop installation folder, as true.**
 - **Change the corresponding CustomDPIEnable attribute, in SuperMap.xml file under bin folder in SuperMap iServer installation folder\support\objectsjava as true.**
 - **Keep CustomDPIX and CustomDPIY in SuperMap.xml as the same value.**

Notice

- **Compact cache password setting**
 - **SuperMap iServer does not support setting password for compact cache, therefore, setting password when using SuperMap iDesktop to generate compact cache is not recommended.**

Notice

- **The scale in iDesktop and in iClient have to be the same.**
- **The main versions of iDesktop and iServer have to be the same**
- **If there are codes that set background transparency attribute on iClient (like the parameter transparent= true in TiledDynamicRESTLayer), then the cache images need to be stored in iServer installation folder\webapps\iserver\output\cache_t**
- **When tiling, first tile 3-4 level of scale, test its effect and then execute the tasks.**
- **In the optimization function, tiling cache after the styles of data and map are fixed is recommended.**

Thank You!

Website: www.supermap.com

Email: globalsupport@supermap.com

Skype: [supermapsupport](https://www.skype.com/people/supermapsupport)

MSN: globalsupport@supermap.com