

## Map caching tips and best practices with ArcGIS Server 10

(From the [ArcGIS Server Blog](#))

Today we'd like to share a list of techniques, best practices, and little-known facts that can improve your experience building map caches with ArcGIS Server 10.

### Configuring local cache directories on the server

ArcGIS Server 10 includes an option to speed performance when caching with multiple SOC machines. When you open the *Caching* tab of the *Service Properties* dialog box and check **Use local cache directory when generating tiles on the server**, each "bundle" of tiles is written into a local directory before being copied into the main, shared cache directory. This is faster than trying to write all the tiles into the main cache directory at once.

Although the local cache directory option is a performance boon and should always be used in multiple-machine deployments, there are some tips you should follow to get the most out of it:

- Uncheck this option if you know you'll be building your cache with just one SOC machine.
- Make sure you have enough space available in the local cache directories. It's recommended that you have at least 0.5 GB of available space in this location for each running map service instance (*ArcSOC.exe*) dedicated to caching. More space may be required for JPEG caches that use a high compression quality, or detailed PNG caches with high bit depth.
- If you use this option and your caching job fails, clean out the local directories before you begin caching again. The default location of the local cache directory is the temp folder for the SOC account unless you define a system variable on the computer named *ARCTMPDIR* whose value is some other folder.

### Using MSDs and MSD-based antialiasing

The faster your map draws, the faster you'll see tiles created. In ArcMap, use the **Analyze Map** button on the *Map Service Publishing* toolbar to identify performance bottlenecks. Fix all of the errors and address as many of the warnings and informational messages as you can. You should then use the toolbar to create a map service definition (MSD) and publish your map service using the MSD.

If you have a layer in the map that is causing an error in the analysis, consider isolating that layer in its own map service so that you can publish the remaining layers in an MSD-based map service.

MSD-based antialiasing is much faster than the antialiasing applied by the caching tools. If you intend on building your caches with antialiasing, enable the antialiasing in the MSD using the **Map Service Publishing Options** button of the *Map Service Publishing* toolbar. Then when you set up your tiling

scheme, leave the antialiasing checkbox (**Smooth line and label edges**) *unchecked*. You'll still see antialiasing in the cached images because it was applied in the MSD.

### Troubleshooting cache failures

Sometimes SOC processes can crash while tiles are being built. These crashes can have various possible causes which are often difficult to detect. Below are some tips for preventing and dealing with crashes.

- Before you start caching, make sure that adequate space is available in your local cache directories and your main cache directory.
- Check if any of the data locations used in your maps, such as ArcSDE connections, have become unavailable during the caching job. Sometimes data located on different drives may fail to be retrieved due to network problems. When this happens a crash can occur, or blank tiles may be generated.
- Attempt to re-run the cache at just the failed extents. These extents are reported in the geoprocessing results messages for the [Manage Map Server Cache Tiles](#) tool. There's a [clever utility on the Code Gallery](#) that can parse the failed extents and make a feature class from them. You can then feed that feature class right back into the Manage Map Server Cache Tiles tool to constrain tile creation to just the failed areas. If the caching job happens to work the second time, you will have a completed cache.
- If the failed extents do not get created successfully after a second try, open your source map document and examine the affected extents to make sure they draw properly in ArcMap (if a draw fails in ArcMap, it will certainly fail while caching).
- Using a copy of your original map document, make a second service that has no cache (draws dynamically). Then set your server log level to Info:Detailed and make several dynamic draw requests at the failed extents. You can then scrutinize the log file to find drawing and connection problems. You can read more about this technique in [Troubleshooting map service performance with log files](#).
- If other techniques fail to find the problem, try to create the failed extents using a second map service. If the tiles generate successfully, use the [Import Map Server Cache](#) tool to import them into your original cache.

### Converting the storage format of a specific data extent

Many of you have asked for better control in the [Convert Map Cache Storage Format](#) tool, so that you can convert from compact to exploded (or vice versa) for only a specific extent of cache data. You can actually do this already using the [Export Map Server Cache](#) tool. Export Map Server Cache lets you choose a specific area or scale from a given cache map service and convert its storage format, while at the same time exporting the tiles to a location you define.