

Managing Large Collections of NGA Raster Data

Bernard Traphan and DJ Bauch

Abstract

NGA (formerly NIMA) raster data presents a unique set of challenges and opportunities. This paper will discuss how these challenges are being addressed at the Joint Information Operations Center to provide easy and efficient access to multiple TerraBytes of raster map data and imagery. NGA data is provided as discrete map images at a variety of scales, but with a common projection scheme. It is possible to take advantage of the characteristics of this scheme (the Equal Arc Second Digitized Raster Graphics, or ADRG scheme) to produce bands of raster data. This results in reduced round-trip traffic and thus increased performance between ArcGIS/ArcIMS & the ArcSDE server. The ADRG scheme divides the globe into eighteen latitudinal bands (nine in the northern hemisphere and nine in the southern hemisphere). Map products at any particular scale within each of these bands share common projection parameters and may be combined into a mosaic for that band and scale. ArcObjects programming is necessary to create these raster bands because ArcSDE doesn't support the ADRG product format. The purpose of this presentation will be to present the steps necessary to create these raster mosaics and to analyze the performance benefits that accrue.

Background

The National Geospatial Intelligence Agency (NGA) is the United States Department of Defense primary provider and custodian of map products. Much of their rich library of paper maps has been digitized and made available as ADRG.

Modeling Our World has this to say about ADRG:

ADRG (ARC Digitized Raster Graphics) data consists of raster images scanned and distributed on CD-ROM by the United States National Imagery and Mapping Agency (US NIMA). ADRG is geographically referenced using the equal arc-second raster chart/map (ARC) system in which the globe is divided into 18 latitudinal zones.¹

This description does not touch on the richness of the ADRG format, which is more fully described in the 85 pages of Military Specification MIL-A-89007 22 February 1990 (Arc Digitized Raster Graphics). The simple description does, however, touch on what may be the most important single feature of ADRG: the 18 latitudinal zones. This same scheme is also used in the later Raster Product Format standard (MIL-STD-2411, 6 October 1994), which forms the basis for Compressed ADRG (MIL-A-89038, 6 October 1994) and Controlled Image Base (MIL-C-89041, 15 May 1995). These other formats also benefit from the approach discussed herein, to

¹ Zeiler, Michael. Modeling Our World: The ESRI Guide to Geodatabase Design. ESRI Press. 1999. p. 158.

an even greater extent than the ADRG format. CADRG maps are broken down into 1536x1536 pixel rasters. There are about 70 CADRG raster frames per single TLM ADRG in Zone 1 and about 56 CADRG raster frames per single TLM ADRG in Zone 2. These formats were not independently investigated because we have not loaded CADRG or CIB data into our SDE instance.

The description from Modeling Our World also does not hint at the amount of data available in this format. There are currently about 10,750 ADRG CD-ROMs available, according to NGA's web site.² A recent edition of NGA's digital catalog includes records for a slightly larger number of products, the newest of which was produced at the end of 2002. About half of this total comprises 5,022 Topographic Line Maps at 1:50,000 scale.

NIMA published a "Notice of Inactivation for New Design" for ADRG 1 December 1997 (MIL-A-89007 NOTICE 1). The text of this notice reads:

MIL-A-89007 is inactive for new design, and is no longer used, except to support existing systems. The corresponding performance specification to be used in new system design is MIL-PRF-89038, Compressed Arc Digitized Raster Graphics (CADRG), dated 6 October 1994. CAUTION: The supersession information is valid as of the date of this notice and may be superseded by subsequent revisions of the superseding document.

An argument can be made that the cautioned supersession of the supersession is overdue, since CADRG turns out to be a poor substitute for ADRG for many purposes, but the inactivation for new design is of little concern. Of the 10,750 ADRG products currently available, 5,052 of them (or just less than half) have been issued since the notice of inactivation.

This table shows the number of ADRG products available at various scales.

Prefix	Map Type	Scale	# CDROMs
ARC4	GNC Global Navigation Chart	1:5,000,000	31
ARC3	JNC Jet Navigation Chart	1:2,000,000	124
ARC1	ONC Operational Navigation Chart	1:1,000,000	266
ARC2	TPC Tactical Pilotage Chart	1:500,000	642
ARC11	Russian General Staff Topographic Map	1:500,000	371
ARC5	JOG Joint Operations Graphic	1:250,000	1,937
ARC10	Russian General Staff Topographic Map	1:200,000	348
ARC6	TLM Topographic Line Map	1:100,000	792
ARC7	TLM Topographic Line Map	1:50,000	5,022
ARC12	Russian General Staff Topographic Map	1:50,000	100
ARC9	CG City Graphic	Varies (1:12,500 typical)	731

² http://www.nga.ic.gov/products/adrg/adrg_pro.html

ARCN	ACO Hydrographic	Varies	356
ARCM	MIM Military Installation Map	Varies (1:25,000 typical)	55
ARCS	Special	Varies	16

Of these ten thousand available CDROMs, approximately two thousand are currently loaded into a single ArcSDE instance in use at the Joint Information Operations Center (JIOC). These raster datasets are incorporated into base map documents and ArcMap layer files that also include elevation rasters (primarily DTED³), imagery (primarily POID⁴), and vector datasets (primarily VMAP⁵). The Information Operations Systems Division makes these map documents and layers available to the “Operations” and “Plans and Policy” Directorates for use in their analysis and presentation processes. Planners and analysts then add their own layers to these documents as they work, creating derived documents that reference personal geodatabases in addition to the foundation data found in ArcSDE.

Case Study

Hardware / Software Environment

Server1: Microsoft SQL Server 2000 with ArcSDE 8.3, Windows 2000 Server, 4 GB RAM, four 2.8 GHz Xeon processors

Server2: Microsoft IIS 5.0 with ArcIMS 4.0.1, Windows 2000 Server, ServletExec 4.1.1, 4 GB RAM, four 2.8 GHz Xeon processors

Client: ArcGIS 8.3, Windows 2000 Professional (5.0, Build 2195), 1 GB RAM, two 1.7 GHz Xeon processors

The machine hosting ArcSDE resides inside the organization firewall. All database-related files reside on a Storage Area Network (SAN).

Originally, the maps used within the JIOC accessed their data primarily from a Windows file share. These maps also typically included some layers constructed from data residing on local disk. The maps were sluggish, but usable. The machines of most users had to be upgraded with additional disk space to hold the ERDAS Imagine pyramid files that ArcGIS stores in C:\TEMP\rasterproxies automatically and with additional memory to avoid heavy paging when accessing map documents with numerous layers. The installation of ArcSDE at our location helped improve performance immensely, even before applying the techniques described in this paper.

³ DTED stands for Digital Terrain Elevation Data. It is available in multiple scales. DTED0 has 30 arcsecond pixel spacing. DTED1 has 3 arcsecond pixel spacing. DTED2 has 1 arcsecond pixel spacing.

⁴ POID stands for Precise Ortho Image Database. POID imagery is provided by NGA in small patches, typically covering a city or urban area.

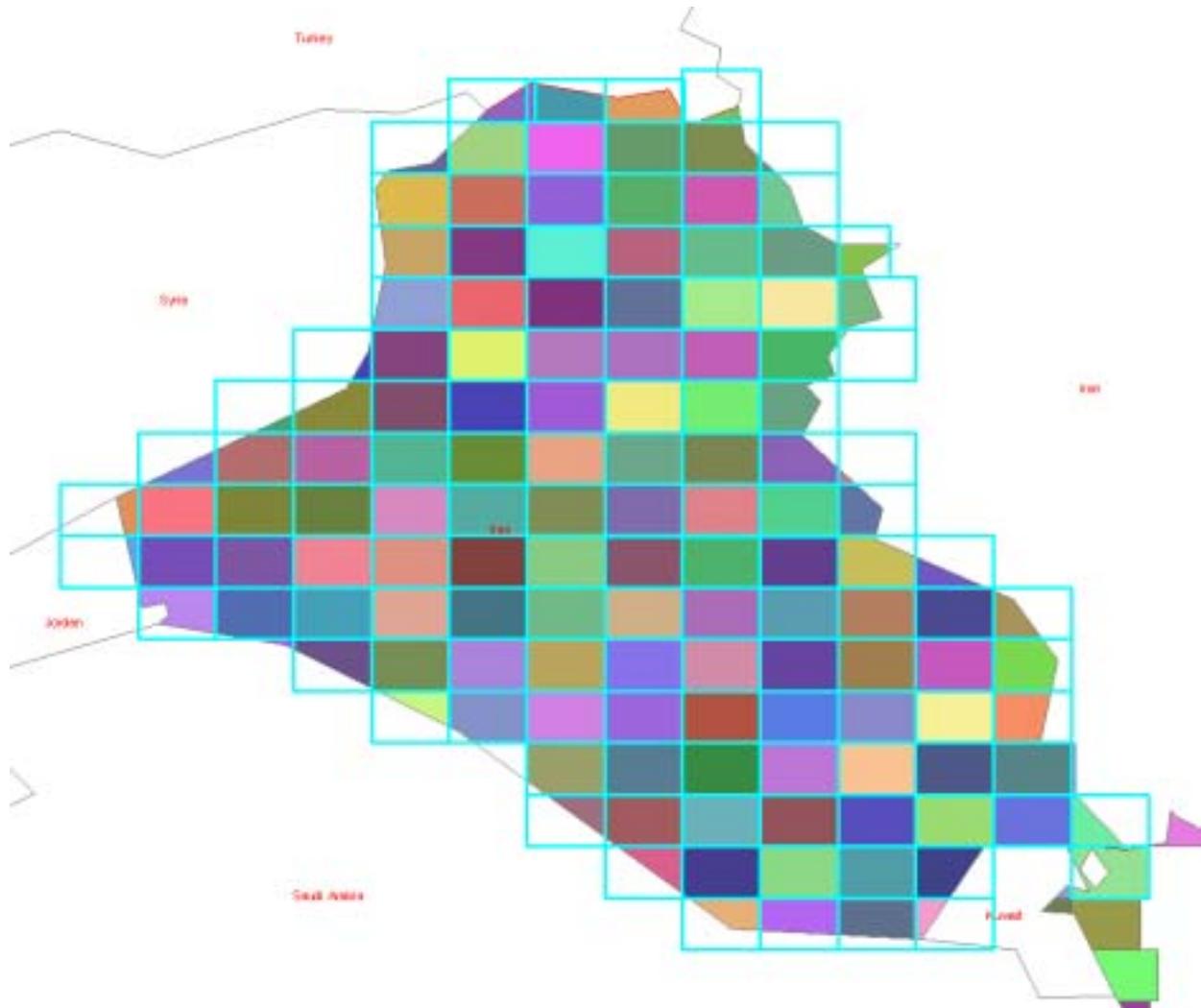
⁵ VMAP stands for Vector Map. The VMAP format is described in a series concerning the Vector Product Format (VPF) which is related to the Raster Product Format (RPF) format.

Typically, the base maps constructed using the NGA raster data span a country of interest to JIOC analysts. One such country, used in this example, is Iraq. The foundation document, IraqBase.mxd includes:

- 4 GNC rasters
- 4 JNC rasters
- 4 ONC rasters
- 10 TPC rasters
- 15 JOG rasters
- 175 1:50K TLM rasters
- 21 CG rasters

for a total of 233 rasters, as well as a few 1:100K TLM rasters and a few ARCN rasters. These rasters are grouped into scale-dependent group layers that toggle on and off as the user zooms in and out on the map. The base maps are constructed this way to allow the community of users to apply the base map to a variety of applications over time. As they work, focus may shift back and forth between various regions or areas of interest and scale up and down between the country-wide level all the way down to just a neighborhood or a few city blocks. Although the techniques described herein apply to ADRG products at all scales, we'll focus just on the TLM map series, which exhibits the most dramatic change, and on the GNC map series to demonstrate that these techniques work at all scales.

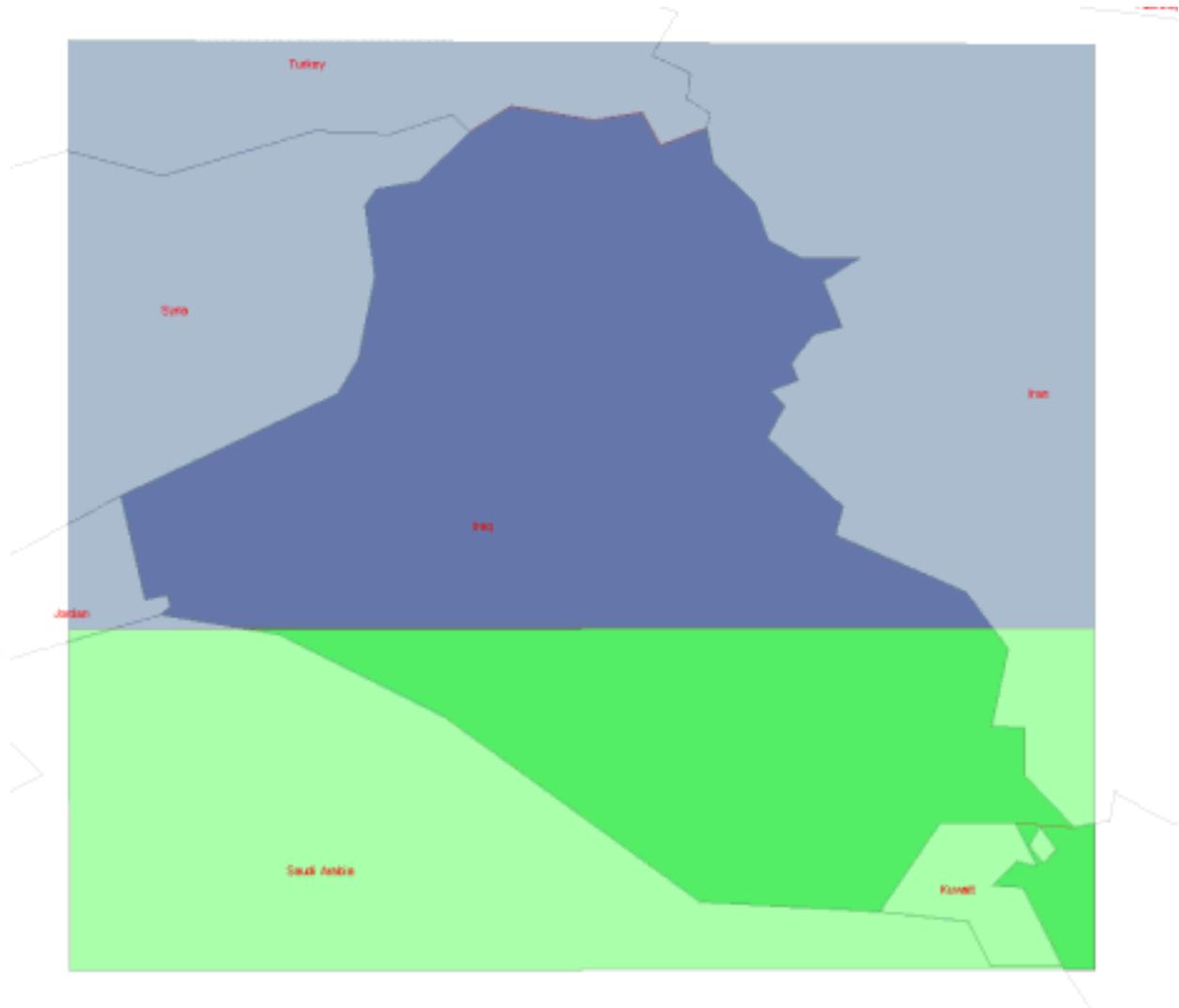
The TLM coverage for Iraq appears in the following illustration. Iraq may be covered by 175 TLMs. Note that if we were using CADRG, the corresponding number of rasters would be over ten thousand, just for the 1:50,000 scale maps.



Rather than using individual TLMs, all of the TLMs that fall within the same zone may be combined into a single mosaic. The 18 zones mentioned in the ADRG description are defined as follows:

Northern Hemisphere Zone	Southern Hemisphere Zone	Equatorward Latitude	Midpoint Latitude	Poleward Latitude
1	A	0°	22.94791772°	32°
2	B	32°	41.12682127°	48°
3	C	48°	52.28859923°	56°
4	D	56°	60.32378942°	64°
5	E	64°	66.09421768°	68°
6	F	68°	70.10896259°	72°
7	G	72°	74.13230145°	76°
8	H	76°	78.17283750°	80°
9	J	80°	n/a	90°

The country of Iraq extends partially into zones 1 and 2. Since ADRG products within a map series all use the same projection within each zone, it is possible to combine the 175 TLMs into just two mosaics, illustrated below. Just two mosaics are sufficient and necessary regardless of the map series. The number of rasters incorporated in the map would decrease from 233 to 14. If we had started from CADRG rather than ADRG, we would be streamlining from over 10,000 rasters for the TLM alone to just two. It is our purpose to examine the performance implications of this change.



Each raster column that has been added into an ArcSDE business table, including those business tables created strictly to identify a raster has a corresponding entry in the RASTER_COLUMNS table.

With all of the available 1:50,000 topographic line maps loaded the SQL query

```
SELECT * FROM sde.sde.SDE_raster_columns WHERE database_name = 'IRAQ' AND table_name like 'ARC7%'
```

Yields 175 rows, ten of which are reproduced here:

rastercolumn_id	description	database_name	owner	table_name	raster_column	cdate	config_keyword	minimum_id	base_rastercolumn_id	rastercolumn_mask	srtd
1264	NULL	IRAQ	BAUCH	ARC7NH3708	IMAGE	1073097251	DEFAULTS	1	0	256	2
1224	NULL	IRAQ	BAUCH	ARC7NH3801	IMAGE	1072926508	DEFAULTS	1	0	256	2
1267	NULL	IRAQ	BAUCH	ARC7NH3802	IMAGE	1073098360	DEFAULTS	1	0	256	2
1251	NULL	IRAQ	BAUCH	ARC7NH3803	IMAGE	1073089787	DEFAULTS	1	0	256	2
1250	NULL	IRAQ	BAUCH	ARC7NH3806	IMAGE	1073089761	DEFAULTS	1	0	256	2
1217	NULL	IRAQ	BAUCH	ARC7NH3807	IMAGE	1072921236	DEFAULTS	1	0	256	2
1219	NULL	IRAQ	BAUCH	ARC7NH3808	IMAGE	1072922623	DEFAULTS	1	0	256	2
1227	NULL	IRAQ	BAUCH	ARC7NH3809	IMAGE	1072928588	DEFAULTS	1	0	256	2
1229	NULL	IRAQ	BAUCH	ARC7NH3810	IMAGE	1073079520	DEFAULTS	1	0	256	2
...

Each row in the raster columns table explicitly references a business table. These business tables are named IRAQ.BAUCH.ARC7NH3708, IRAQ.BAUCH.ARC7NH3801, IRAQ.BAUCH.ARC7NH3802, etc. The business tables are identified by the combination of the DATABASE_NAME, OWNER, and TABLE_NAME columns of the RASTER_COLUMNS table. The additional tables are also related to the business table by the ArcSDE software, which uses the RASTER_ID column to form suffixes that are concatenated to “SDE_aux_”, “SDE_blk_”, “SDE_bnd_”, or “SDE_ras_” to identify the names of these other tables. Each of the business tables created by loading an ADRG is identical in both intent and extent, and appears as reproduced here:

NAME	IMAGE
ESRI_SDERASTERDATASET	1

There are also 175 identical raster metadata tables (SDE_ras_#####), each of which appears as follows:

raster_id	raster_flag	description
1	NULL	NULL

There are 175 raster band tables (SDE_bnd_#####), each of which contains 3 records.

rasterband_id	sequence_nbr	raster_id	name	band_flags	band_width	band_height	band_type	block_width	block_height	block_origin_x	block_origin_y	eminx	eminy	emaxx	emaxy	cdate	mdate
1	1	1	NULL	425986	15488	11136	4195328	128	128	41.24792222	32.00063889	41.24792222	31.50001184	42.00204251	32.00063889	1073097251	1073097999
2	2	1	NULL	425986	15488	11136	4195328	128	128	41.24792222	32.00063889	41.24792222	31.50001184	42.00204251	32.00063889	1073097251	1073098037
3	3	1	NULL	425986	15488	11136	4195328	128	128	41.24792222	32.00063889	41.24792222	31.50001184	42.00204251	32.00063889	1073097251	1073098074

There are 175 auxiliary tables (SDE_aux_#####), each of which also contains 3 records.

rasterband	type	object
1	2	...
2	2	...
3	2	...

Finally, there are 175 block tables (SDE_blk_#####), each of which contains between about 35,000 and 43,000 records. This illustration shows the first few records in the table and the final few, including all the records where rrd_factor = 6.

1	0	0	2	...
1	0	0	3	...
1	0	0	4	...
1	0	0	5	...
...	thousands of records...			...
1	6	0	0	...
1	6	0	1	...
1	6	1	0	...
1	6	1	1	...
2	6	0	0	...
2	6	0	1	...
2	6	1	0	...
2	6	1	1	...
3	6	0	0	...
3	6	0	1	...
3	6	1	0	...
3	6	1	1	...

The raster columns table explicitly references entries in the SDE_SPATIAL_REFERENCES table, which at our organization contains over a hundred entries, but all of the rasters loaded from ADRG wind up referencing the same entry (SRID value 2 in our database), which is the worldwide unscaled WGS84 datum. The only reason they don't all reference SRID 1 is that we had loaded some data from Shapefiles prior to loading any rasters, creating the "NAD 27 Assumed Geographic" ESRI default spatial reference. The first couple of entries in the sde_spatial_references table appear below.

srld	description	auth_name	auth_srld	falsex	falsey	xyunits	falsez	zunits	falsem	munits	srtext
1	NULL	NULL	NULL	-107.2656605	20.959106	127999999.9	0	1	0	1	GEOGCS["GCS_Assumed_Geographic_1",DATUM["D_North_American_1927",SPHEROID["Clarke_1866",6378206.4,294.9786982]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]]
2	NULL	NULL	NULL	0	0	1	0	1	0	1	GEOGCS["GCS_WGS_1984",DATUM["D_WGS_1984",SPHEROID["WGS_1984",6378137.0,298.257223563]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]]
...

The cell size and pixel alignment problem

ADRG rasters may be loaded into ArcSDE using ArcCatalog, ArcToolbox, or ArcObjects. They may not be loaded directly using the sderaster command, which understands only a small number of raster file formats. When they're loaded with ArcCatalog or ArcToolbox, however, they'll have black pixels around the edges of the geographic coverage that each CDROM references that we'll want to eliminate prior to doing the mosaic. If we use ArcObjects, our code can eliminate the black borders while performing the load or mosaic strips, as discussed later.

The following illustration shows how an ADRG loaded from CDROM via ArcToolbox or ArcCatalog appears.



It seems that we should be able to painlessly mosaic the ADRGs at a given scale within a band, once black borders are removed. That turns out to be optimistic. The command

```
c:\ARCSDE\sqlexe\bin\sderaster -o import -l GNCBand1,image ^  
-v 1 -f "-l GNC02Band1,image -v 1" -c lz77 -a 0 -M 1 ^  
-G 4326 -k DEFAULT -S "Band 1 for GNC Data" -L 0 ^  
-I bilinear -t 128,128
```

successfully creates a new raster layer with transparent pixels replacing the black borders that surround an ADRG loaded via ArcCatalog. The modified raster appears below.



We can now attempt to add another ADRG to the mosaic. The command

```
c:\ARCSDE\sqlexe\bin\sderaster -o mosaic -l GNCBand1,image ^  
-v 1 -f "-l GNC07Band1,image -v 1" -L 0 -N -a 0
```

Yields:

```
Connecting to server arcsde, port 5151, as user JIOC\BAUCH
```

```
Connecting to server arcsde, port 5151, as user JIOC\BAUCH
```

```
Image Dimension.....: 12544, 8192, 3  
Pixel Type.....: uchar  
Image Extent.....:  
  minx      :          143.37930838060646  
  miny      :          0.00330011667779
```

```
maxx      :                204.20323134239686
maxy      :                36.68789713903837
Error: SE_stream_execute (-303)
Error: Image mosaicking is not allowed
Incomplete...
```

Obviously, something has gone wrong. This error message typically results from a cell size problem or cell alignment problem. It could also result from the absence of a common spatial reference. In this case, it appears to be a cell size problem. Examining the original ADRG file properties in ArcCatalog shows that the GNC02 Band 1 image (a file that happens to be named “GNUS0101.IMG”) shows that it has an X cellsize of 0.004849 and a Y cellsize of 0.004479 and the GNC07 Band 1 image (a file that happens to be named “GNNQ0101.IMG”) has an X cellsize of 0.004849 and a Y cellsize of 0.004479. After they’ve been loaded into ArcSDE, however, GNC02Band1 reports a Y cellsize of 0.004480 while GNC07Band1’s cell sizes remain unchanged. We have not yet determined precisely why the reported cell sizes sometimes change when the rasters are loaded into SDE.

In tests, even when the cell size problem does not exist, the mosaic operation of the sderaster command still fails. Obviously, ArcSDE detects a cell alignment problem – even though there should not be an alignment problem according to the ADRG specification.

Even in the face of this problem, it is still possible to create the mosaic. It is necessary to revert from using the sderaster command-line tool to using ArcCatalog or ArcToolbox, both of which will overcome the cell size and cell alignment problems. Unfortunately, neither approach will allow a database-to-database operation, as the sderaster command does. To work around the problem, the rasters with transparent backgrounds must temporarily be moved from the database out to a file system. ArcCatalog allows export to three formats: ESRI GRID, TIFF, and ERDAS Imagine. Exporting from SDE using the export operation of the sderaster command to produce TIFF files is not worthwhile. Exporting from SDE to TIFF format, either from the sderaster command or from ArcCatalog reintroduces the black bands surrounding the raster. Exporting from SDE to MrSID typically fails. ERDAS Imagine seems like the best bet.

Once the ERDAS Imagine files have been produced, they may then be re-exported into the database with ArcCatalog or ArcToolbox. Unlike a normal export, it is wise to choose “Change Settings” and to turn off the calculation of statistics and the construction of pyramids.

Perhaps it is not obvious, but multiple occurrences of this step may be performed in parallel, even with a single ArcSDE raster as a destination. Although there is a small amount of overlap between adjacent latitudinal bands, with transparency enabled there will be no overlap between adjacent longitudinal ADRG quadrangles. This statement is not necessarily true of CADRG. Presumably, even if there were some overlap, however, the overlapped pixels would be duplicates of one another. We did observe that this frequently results in ArcCatalog locking up after the data has been uploaded to ArcSDE. ArcCatalog never receives the final response from ArcSDE that it is expecting. When this happens, the Windows Task Manager can be used to end one of the ArcCatalog tasks that report a status of “Not Responding”. When the non-responsive ArcCatalog is terminated, the others usually immediately unlock.

It is also possible to set up and perform the load from ArcToolbox as a batch process, but not from ArcCatalog, which will not allow changing the name of the input file from row to row of the batch grid. If the batch is saved and restored from a “BAT” file, then it will be necessary to edit the batch window to remove the ‘#’ character that the tool inserts for the catalog name. It will also be necessary to edit the batch grid to turn off the building of pyramids and calculation of statistics, since those options are not saved to the batch file. Unfortunately, the ArcToolbox frequently just reports that image mosaics are not allowed after uploading the first raster. To correct this problem, begin as though loading just a single raster and choose “Change Settings”. Set the Coordinate System to GCS_WGS_1984 and turn off the generation of statistics and pyramids. You may then add the remaining rasters to the batch grid, either manually or by loading a batch file and make sure that these parameters are unchanged for each entry. Loading a batch file and creating a mosaic does not seem to be possible as a single step, since you will be unable to set the coordinate system for all the files in the batch.

At this point, the tables will all have been created, but the auxiliary table will be empty. Also, all the records in the block table will have `rrd_factor` equal to 0. In the band table, the `band_flags` will be set to zero.

Follow the mosaicking operation with commands to create raster pyramids and to calculate statistics.

```
c:\ARCSDE\sqlexe\bin\sderaster -o pyramid -l GNCBand1,image ^  
-v 1 -L -1 -I bilinear
```

This will create the pyramids for the raster and store them in the block table. It will also update the `band_flags` in the band table to reflect that the bands have pyramids (value 524289, for example). It is important to note that removing the raster pyramids (by running the pyramid command with the option “-L 0”) does not actually remove the pyramids from the database. It simply updates the `band_flags` in the band table to indicate the absence of pyramids.

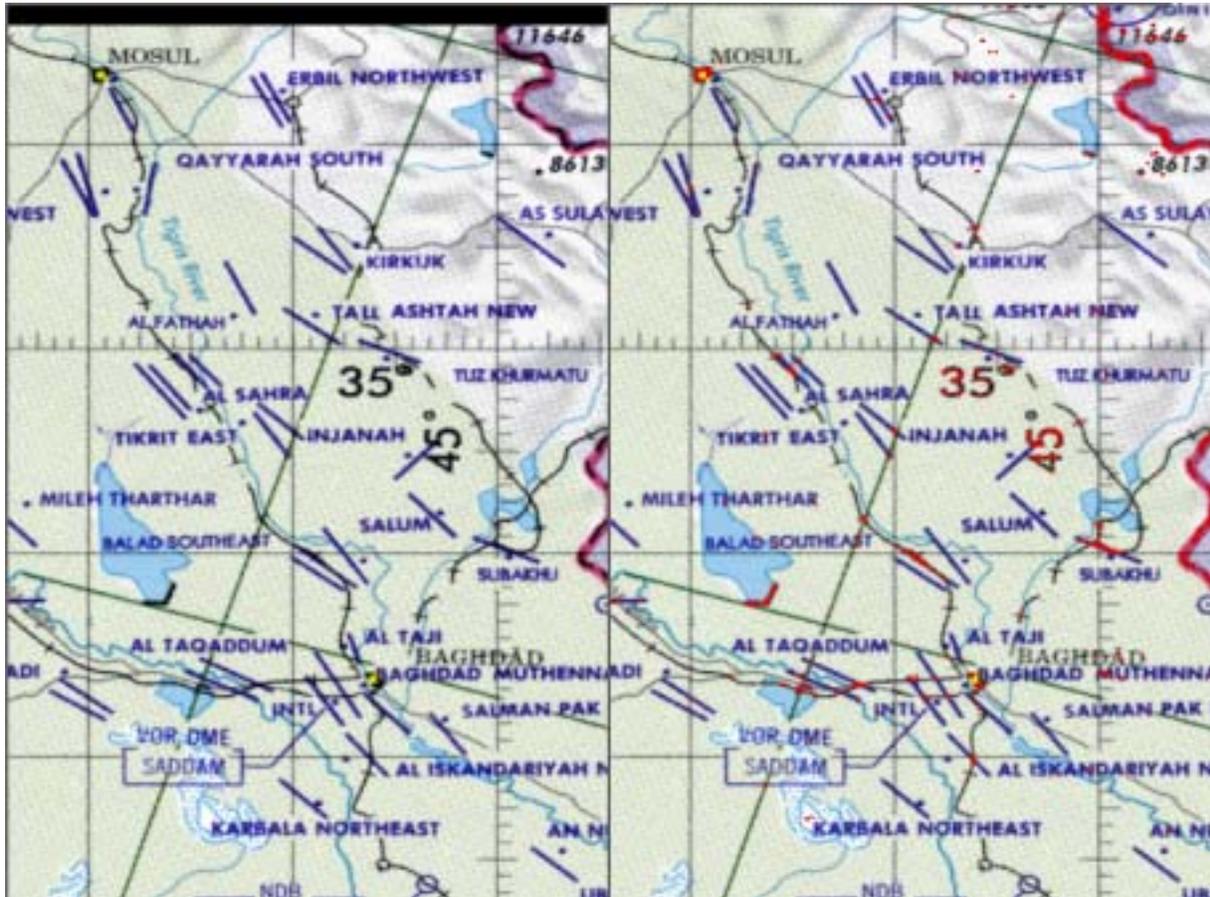
Follow the creation of pyramids with statistics calculation.

```
c:\ARCSDE\sqlexe\bin\sderaster -o stats -l GNCBand1,image -v 1
```

This will create records in the auxiliary table, band by band. It will also update the band table to reflect that the bands have statistics (setting the 524289 used in the example above to 557057).

There is a moderately annoying problem that occurs with some ADRG images as a result of the elimination of the black borders. The problem is that the absolute black (0, 0, 0) that appears in the borders of the ADRG is also occasionally used as a black in the ADRG itself. The problem remains manageable, however, because most of the blacks that appear in the ADRG are not absolute black. The problem does not manifest in all of the ADRGs. This problem could be avoided if the “-R #:#” flag for the `sderaster` import operation worked with these images. Attempting to use that flag just produces an error message. The problem could also be avoided by using `ArcObjects IRasterSdeConnection2` rather than `IRasterSdeConnection` in custom code, and using the `Raster` property rather than the `InputRasterName` property. The custom code may then trim away the black borders without altering any of the pixels in the interior of the ADRG.

This problem can be observed in a fragment of a GNC ADRG. The original GNC appears as illustrated on the left below. On the right is the same GNC after the black borders have been removed. The raster has been drawn into a frame with a red background to highlight the appearance of transparent pixels within the image. Red now shows through several areas that formerly were black, particularly in some of the text and in the border between Iraq and Iran.



Performing a Mosaic with ArcObjects

Code like the following may be used to create a mosaic from all the rasters in a directory. This code is similar to an example that appears in the May 2002 ESRI White Paper, **Raster Data in ArcSDE 8.2**.

```
Sub MosaicDirToSDE(sDir As String, sServer As String, sInstance As String, _
sDB As String, sUser As String, sPasswd As String, sSDERaster As String)
    Dim pSDEConnection As IRasterSdeConnection
    Dim pSDEStorage As IRasterSdeStorage
    Dim pSDEServerOp As IRasterSdeServerOperation
    Dim pWsFact As IWorkspaceFactory
    Dim pWs As IWorkspace
    ' --- all the datasetnames in the directory
    Dim pEnumDNS As IEnumDatasetName
    Dim pDsName As IDatasetName
```

```

Dim iCount As Long
Dim pName As IName
Dim pGeoDs As IGeoDataset

' --- Open the workspace of the specified dir
Set pWsFact = New RasterWorkspaceFactory
Set pWs = pWsFact.OpenFromFile(sDir, 0)

' --- Get all the datasetnames
Set pEnumDNS = pWs.DatasetNames(esriDTRasterDataset)

' --- Set SDErasterLoader
Set pSDEConnection = New RasterSdeLoader
pSDEConnection.ServerName = sServer
pSDEConnection.Instance = sInstance
pSDEConnection.Database = sDB
pSDEConnection.UserName = sUser
pSDEConnection.Password = sPasswd
pSDEConnection.SdeRasterName = sSDEraster

' --- Loop through all the datasetnames and mosaic them
iCount = 0
Set pDsName = pEnumDNS.Next
Do While Not pDsName Is Nothing
    pSDEConnection.InputRasterName = sDir + "\" + pDsName.Name
    ' -- full path name --
    Set pSDEStorage = pSDEConnection
    pSDEStorage.CompressionType = esriRasterSdeCompressionTypeRunLength
    pSDEStorage.PyramidOption = esriRasterSdePyramidDonotBuild
    Set pSDEServerOp = pSDEConnection
    ' --- Create the first one and mosaic the rest
    If iCount = 0 Then
        Set pName = pDsName
        Set pGeoDs = pName.Open
        Set pSDEStorage.SpatialReference = pGeoDs.SpatialReference
        pSDEServerOp.Create
    Else
        pSDEServerOp.Mosaic
    End If

    Set pDsName = pEnumDNS.Next
    iCount = iCount + 1
Loop

pSDEServerOp.ComputeStatistics
pSDEStorage.PyramidOption = esriRasterSdePyramidBuildWithFirstLevel
pSDEStorage.PyramidResampleType = RSP_BilinearInterpolation
pSDEServerOp.BuildPyramids
End Sub

```

Performance Impact

The May 2002 ESRI White Paper (using ArcGIS 8.1.2) reported on a scenario involving 2000 640x480 grayscale rasters that were placed into a raster catalog and a single 32,168 x 19,680 pixel raster mosaic. In this experiment, the reported display times varied from one second for the

seamless mosaic to two minutes for the raster catalog when displaying a 1,000 x 630 section of the image. This translates to a performance improvement factor of 120.

In our tests, we constructed two maps. In one map we placed 100 TLM-50 (1:50,000) ADRGs. In the other map we placed two Mosaics constructed according to the ADRG latitude bands. We could not use all 175 available TLMs for Iraq because we don't have all of them loaded in our SDE instance.

The first test was to see how long it would take just to open a map, with the raster data disabled. Even in this situation, there are queries sent to the ArcSDE server for each raster layer in the map. For each layer, ArcSDE queries the business table, then the band table to count the number of bands and to extract the other columns in the band table, the raster table, and finally extracts the metadata (object column) from the auxiliary table for each of the bands within the raster layer.

The results of testing how long it takes to open the map from ArcView 8.3:

Open Map

2 layers	5 seconds
100 layers	78 seconds

The second test, which resembles something that a user might be expected to do, involves viewing an area of the map at 1:500,000 scale. This view includes all or part of six separate ADRG images.

Draw Map

2 layers	3 seconds
100 layers	20 seconds

The final test is a stress test. It doesn't resemble anything that a user ought to do. We want to see how long it would take to draw all of the raster data in a world view. In typical use, the TLM raster layers would be disabled once the scale reaches some cutoff value and the active view would switch to one of the other map series.

Draw All

2 layers	3 seconds
100 layers	285 seconds

Note that the mosaic-based map exhibits the behavior that ArcView is supposed to exhibit, namely that the time to draw the map remains approximately constant, regardless of scale. The map constructed from individual TLMs takes almost 100 times as long to display.

The impact of using mosaics is particularly important with ArcIMS. While ArcView is forgiving of poorly performing collections of rasters, such as this example, ArcIMS is not. Typically, attempting to create an ArcIMS service from a map containing more than a few dozen rasters will simply not work. Instead, ArcIMS will time out while attempting to start the service.

Conclusion

The benefits of creating mosaics from ADRG data have been conclusively demonstrated. The performance advantages are such that it really doesn't make sense to work with individual ADRG images in ArcSDE.

The downside is that it remains difficult to construct these mosaics using the standard ESRI tools, and the results are imperfect. Better results are achievable through custom programming with ArcObjects.

Wish List

Support for the ADRG format in sderaster would be useful. We don't expect this to happen unless or until there is a version of sderaster built on ArcObjects.

A capability like the `-R #:#` flag for the sderaster import operation that works for three-band non-rotated rasters would be useful, so that interior black pixels don't get eliminated from the raster.

The ability to specify the transparent colors and to eliminate borders from within the ArcCatalog and ArcToolbox tools would be useful. These abilities would be provided by the evolved ArcObjects interface `IRasterSdeServerOperation2` that adds a Raster property to the features of the `IRasterSdeServerOperation` interface that appears to be used in the ArcGIS 8.3 version of the Raster to Geodatabase tool. As a side effect, this would also allow the tool to be used for geodatabase-to-geodatabase loads, as supported by sderaster via the `-f "-l <sde-raster>"` syntax.

The problems with the batch mode for the Raster to Geodatabase tool, particularly the failure to save and restore the full state of the batch grid, should be fixed.

Author Information

Bernard Traphan
Joint Information Operations Center/J62
PO Box 769178
San Antonio, TX 78245
Phone: 210 414 5226
Fax: 888 678 7315
Email: Bernard.Traphan@jioc.osis.gov

D J Bauch
Contractor, SAIC Information Operations Division
Joint Information Operations Center/J62
2 Hall Blvd, Suite 217
San Antonio, TX 78243-7008
Phone: 210 977 4771
Fax: 210 977 4021
Email: Danny.Bauch@jioc.osis.gov