



# **Raster Data in ArcSDE<sup>®</sup> 9.1**

*An ESRI<sup>®</sup> White Paper • September 2005*

Copyright © 2005 ESRI  
All rights reserved.  
Printed in the United States of America.

The information contained in this document is the exclusive property of ESRI. This work is protected under United States copyright law and other international copyright treaties and conventions. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or by any information storage or retrieval system, except as expressly permitted in writing by ESRI. All requests should be sent to Attention: Contracts and Legal Services Manager, ESRI, 380 New York Street, Redlands, CA 92373-8100, USA.

The information contained in this document is subject to change without notice.

#### **U.S GOVERNMENT RESTRICTED/LIMITED RIGHTS**

Any software, documentation, and/or data delivered hereunder is subject to the terms of the License Agreement. In no event shall the U.S. Government acquire greater than RESTRICTED/LIMITED RIGHTS. At a minimum, use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in FAR §52.227-14 Alternates I, II, and III (JUN 1987); FAR §52.227-19 (JUN 1987) and/or FAR §12.211/12.212 (Commercial Technical Data/Computer Software); and DFARS §252.227-7015 (NOV 1995) (Technical Data) and/or DFARS §227.7202 (Computer Software), as applicable. Contractor/Manufacturer is ESRI, 380 New York Street, Redlands, CA 92373-8100, USA.

ESRI, the ESRI globe logo, 3D Analyst, ArcGlobe, ArcIMS, ArcSDE, ArcGIS, ArcObjects, ArcMap, Spatial Database Engine, SDE, ArcInfo, ArcView, ArcToolbox, ArcCatalog, [www.esri.com](http://www.esri.com), and [@esri.com](mailto:@esri.com) are trademarks, registered trademarks, or service marks of ESRI in the United States, the European Community, or certain other jurisdictions. Other companies and products mentioned herein are trademarks or registered trademarks of their respective trademark owners.

# Raster Data in ArcSDE 9.1

## An ESRI White Paper

<b>Contents</b>	<b>Page</b>
Introduction.....	1
Other Suggested Resources.....	1
Why Put Images in a Database?.....	1
Multiuser Access.....	1
Data Management.....	1
Data Security.....	2
Data Query.....	2
Uses of ArcSDE Raster Data.....	2
Basemaps—Enterprise GIS (utility or local government).....	2
Data Management/Distribution (data providers).....	2
Feature Attributes (utility, real estate, or local government)....	2
Basic Raster Concepts.....	2
How Raster Data Is Stored in the Database.....	3
Storage Parameters.....	5
Pyramids.....	5
Pyramid Reference Point.....	7
Partial Pyramid Update.....	7
Tile Size.....	7
Compression.....	7
Compression Ratios and Qualities.....	8
Configuration Keyword.....	10
Background Value.....	10
Nodata Value.....	10
Statistics.....	10
Colormap.....	11
Metadata.....	11
Data Loading.....	11
Preprocessing of Continuous Rasters.....	11
Loading Parameters.....	11
Data Loading in ArcGIS.....	13

<b>Contents</b>	<b>Page</b>
Create a New SDE Raster .....	13
Import a Raster Dataset.....	14
Mosaic Rasters to an Existing SDE Raster .....	16
Load Raster to a Table .....	18
Load Raster to a Feature Class.....	18
Multiple Raster Attributes for a Feature Class .....	18
Create a Raster Catalog.....	19
Insert to a Raster Catalog.....	20
Add Metadata Column to a Raster Catalog .....	20
Batch Loading.....	21
Load Rasters in a Workspace.....	21
Data Export.....	22
Export Raster Dataset .....	22
Export Rasters from a Raster Catalog.....	22
Export Footprints from a Raster Catalog.....	22
Extract Features and Rasters in a Certain Region.....	22
Transferring between Databases.....	22
Create a Raster Catalog View .....	23
Data Viewing .....	23
Viewing from ArcGIS.....	23
ArcCatalog Application .....	23
Raster Catalog View .....	23
Adding/Deleting Fields.....	24
Viewing Raster as Attribute.....	24
ArcMap Application .....	24
Raster Catalog Selection.....	24
Viewing Time Series Raster Catalog.....	24
Raster Catalog Rendering .....	25
ArcGlobe.....	26
ArcIMS .....	27
Personal Geodatabase .....	28
Upgrading from ArcSDE 8.3 to ArcSDE 9 .....	28
Custom Applications.....	28
ArcObjects Sample .....	29
ArcObjects for Loading .....	29
Loading Sample .....	29

<b>Contents</b>	<b>Page</b>
Scripting Sample.....	32
Geoprocessing Scripting Sample .....	32
Tips and Tricks .....	33
Creating Data .....	33
Loading Data.....	33
Loading Projected Data.....	33
Updating Raster Catalog Footprints.....	33
Storage .....	34
Loading Rasters in MrSID Format .....	34
Mosaicking Rasters with Colormap.....	34
Nodata.....	34
Viewing in ArcGIS .....	34
Mosaic versus Raster Catalog.....	35
Attribute Table.....	35
Raster Catalog Does Not Display .....	35
 <b>Appendixes</b>	
Appendix A—Bibliography.....	36
Appendix B—Raster Table Schema.....	38



---

# Raster Data in ArcSDE 9.1

**Introduction** With the release of ArcSDE® 9.1 and the ArcGIS® Desktop family of products, the use of rasters is designed to be easier and more powerful. New functionality in ArcGIS includes the following:

- Raster datasets and raster catalogs are now fully qualified members of the geodatabase.
- ArcGIS renders catalogs per image.

New raster functionality in ArcSDE includes

- Partial pyramid updates
- JPEG 2000 compression

This document is intended to support the use of raster data in ArcSDE by highlighting the basic concepts, user experience, and product overview for the support of raster data in ArcSDE. The document explains how ArcSDE provides efficient storage and retrieval of raster data in a client/server environment. Where possible, best practices for the loading, storage, and retrieval of raster layers are given. Raster data can be accessed by ArcGIS Desktop or applications customized with the ArcSDE Java™ and C APIs or the ArcObjects™ COM API.

## ***Other Suggested Resources***

*Modeling Our World—The ESRI Guide to Geodatabase Design*  
*Understanding ArcSDE*  
*Managing ArcSDE Services*  
*ArcSDE Configuration and Tuning Guide* (database specific)  
*ArcSDE Developer Help*  
*ArcObjects Developers Help*  
*Upgrading Geodatabase Rasters from ArcGIS 8.3 to ArcGIS 9* (technical paper)  
*Managing a Raster Database* (technical paper)

A list of specific resources is included in Appendix A—Bibliography.

## ***Why Put Images in a Database?***

### **Multiuser Access**

When many users are accessing the same raster files simultaneously, better performance is possible from a properly tuned, centralized database than from a file-based system.

### **Data Management**

A database allows common data management and retrieval for all geospatial data including raster, vector, metadata, and tabular data. A database also provides access to extremely large images (many gigabytes to many terabytes) of continuous spatial data (e.g., 30-meter digital elevation model composite of North America).

**Data Security** A database has tools for multiple security levels to be established and enforced. Users can be given access to the imagery that is relevant to the job they are being asked to work on.

**Data Query** A database allows for a common query environment. Queries can be made to show all data related to an area during a particular time period or for a particular subject.

***Uses of ArcSDE  
Raster Data***

**Basemaps—  
Enterprise GIS (utility  
or local government)** A water supply company had a lot of legacy data stored on paper and Mylar® maps. It needed to be able to supply these to its users as background images to be combined with vector data into a seamless hybrid map. Eventually, these images will be completely replaced by vector data. The images were scanned as one-bit TIFF images. The company has stored approximately 3,000 images (40 GB) centrally using ArcSDE and supplies them as background images to users running ArcMap™. Its users are inserting new vector data as well as updating existing vector data with new information.

**Data Management/  
Distribution (data  
providers)** An association of governments needed a central repository of imagery that could be easily accessed by its members as well as citizens who requested the imagery. It needed to provide the images in downloadable formats as well as set up and maintain a Web site that allowed interested parties to view the images online. The data was in eight-bit, three-band, one-meter digital orthophoto quarter quadrangles (DOQQ). It has stored 1 TB of imagery using ArcSDE, with more images expected as the areas are reflowed by satellites and aircraft.

**Feature Attributes  
(utility, real estate, or  
local government)** An organization may have pictures of locations that it needs to attach to a spatial feature. This could be a picture of a house linked to a parcel boundary or a picture of a pump or valve linked to a hydraulic network.

**Basic Raster  
Concepts** Vector data, such as coverages and shapefiles, represents geographic features with lines, points, and polygons. Rasters, such as images and grids, represent geographic features by dividing the world into discrete squares called cells. Cells are laid out in a grid, where each cell has a location relative to an origin and a value describing the feature being observed; for instance, the cell values in an aerial photograph represent the amount of light reflecting off the earth's surface.

Some rasters have a single band (a measure of some characteristic) of data while others have multiple bands (more than a single measure). When you create a layer from a raster, you can choose to display a single band of data or form a color composite from multiple bands. A grayscale aerial photo has a single band representing different levels of the land's surface reflectance. A satellite image commonly has multiple bands representing different wavelengths of energy from the ultraviolet through the visible and infrared portions of the electromagnetic spectrum.

Rasters also have a measure of the number of colors able to be stored in a cell. This is the bit or color depth. A bilevel or one-bit image will be able to display two colors—black and white. An eight-bit image will be able to display 256 colors (2<sup>8</sup>). The higher the bit depth, the more colors available for display but the larger the storage requirements.

To learn more about rasters, see *Understanding ArcSDE* (pp. 17–19) or *Modeling Our World* (chapter 9).

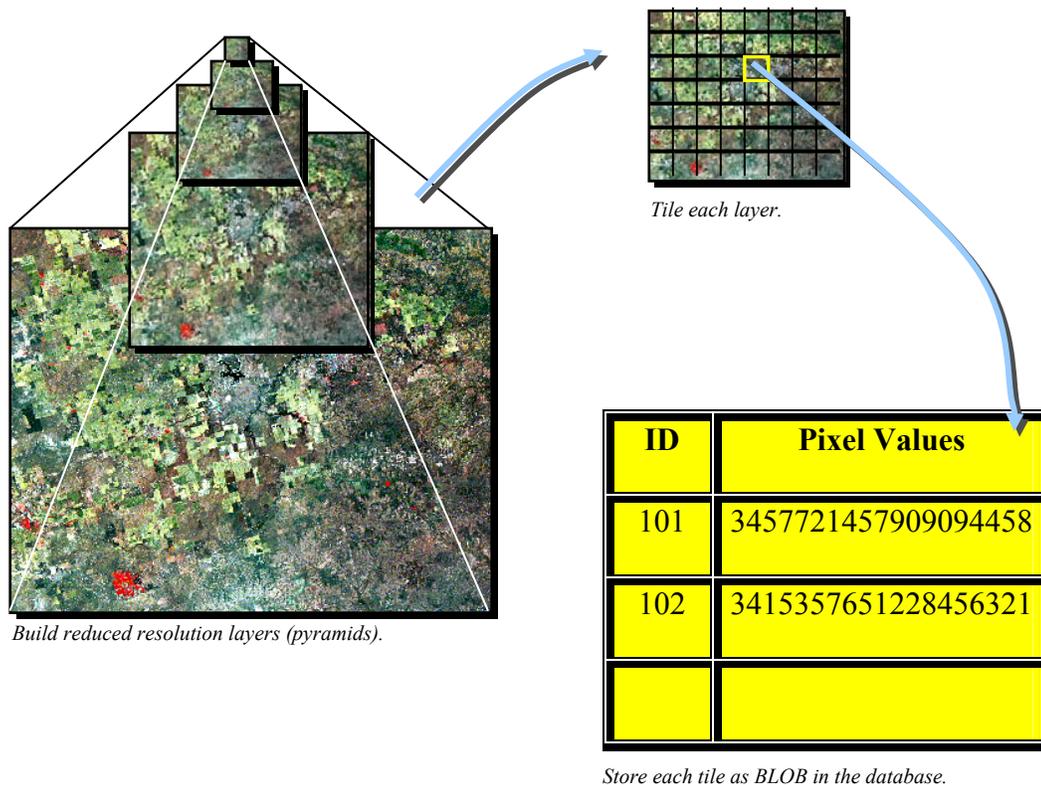
Raster layers are seen by client applications as one of two distinct types.

- Raster dataset—A single picture of an object or a seamless image covering a spatially continuous area. This may be a single original image or the result of many images appended together.
- Raster catalog—A collection of rasters displayed as a single layer. These rasters must all be in the same coordinate system, but they can have different properties.

***How Raster Data Is Stored in the Database***

When raster data is loaded into a database, it is converted into the Spatial Database Engine™ (SDE®) raster format. The raster is tiled using the user-specified tile size, and spatial indexes are built. The data is then resampled using the method specified to create pyramids. The tiles are stored as many small binary large objects (BLOBs) in a set of ArcSDE system and user tables. By doing this, when the raster is queried, only the necessary tiles are returned instead of the whole dataset, thus improving end user performance. Client display performance is optimized by reducing the amount of data transferred to the client application. This makes it possible to store large seamless raster datasets and serve them quickly to a client for display.

**Figure 1**  
**Raster Data Storage and Access in Database**



When a raster is created in an ArcSDE database, ArcSDE adds a raster column to the business table of the user's choice. This table may be an existing table or created for the user. Users may name the raster column whatever they like, as long as the name conforms to the underlying database column naming convention. ArcSDE has a restriction of one raster column per business table.

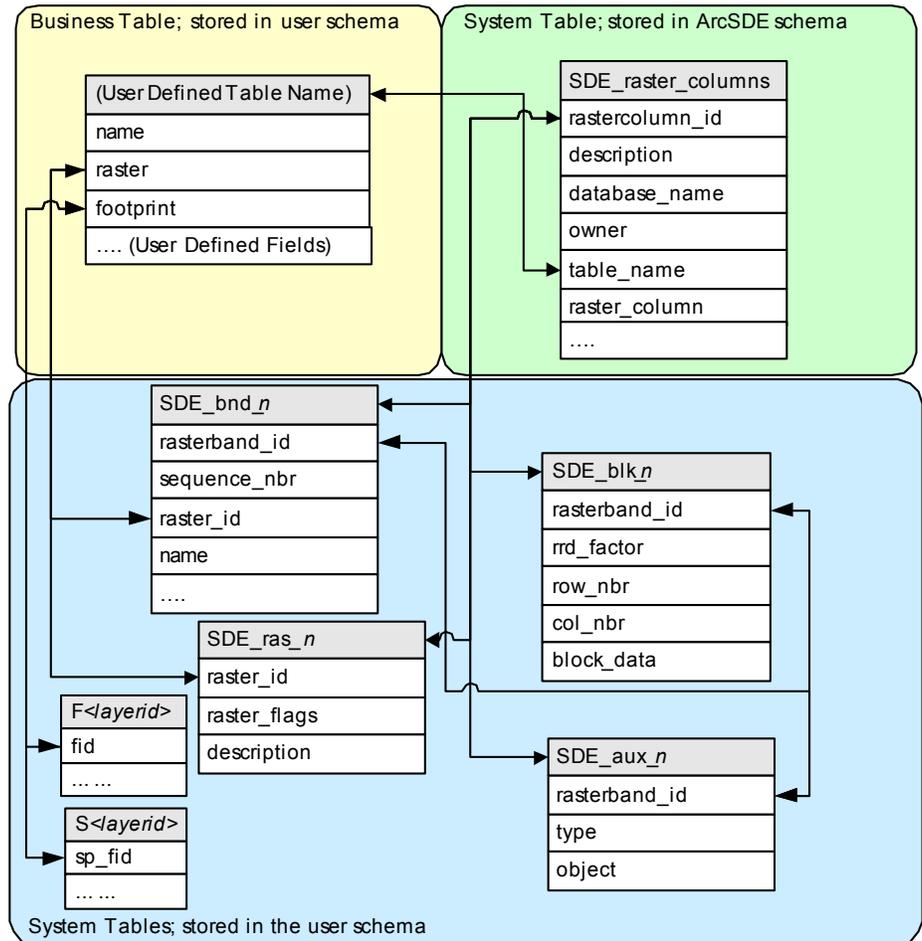
The raster column is a foreign key reference to the raster\_id column of the raster table (SDE\_ras\_n) created during the addition of the raster column. Also joined to the raster table's raster\_id primary key, the raster bands table (SDE\_bnd\_n) stores the bands for each image. The raster auxiliary table (SDE\_aux\_n) joins many-to-one to the raster bands table by rasterband\_id, which stores the metadata of each raster band. The rasterband\_id also joins the raster band's table to the raster block table (SDE\_blk\_n) in a many-to-one relationship. The raster block table's rows store the tiles determined by the dimensions of the block.

When ArcSDE adds a raster column to a table, it records that column in the SDE user's SDE\_raster\_columns table. The rastercolumn\_id is used in creating the names of the raster, raster bands, raster auxiliary, and raster block tables.

At ArcGIS 9, the footprint of each raster is also stored in the geodatabase, a polygon geometry column is added to the raster business table, and the geometry is stored in associated F\_ tables and S\_ tables just as feature classes do.

See appendix B for a detailed description.

**Figure 2**  
**Raster Data Storage Schema**



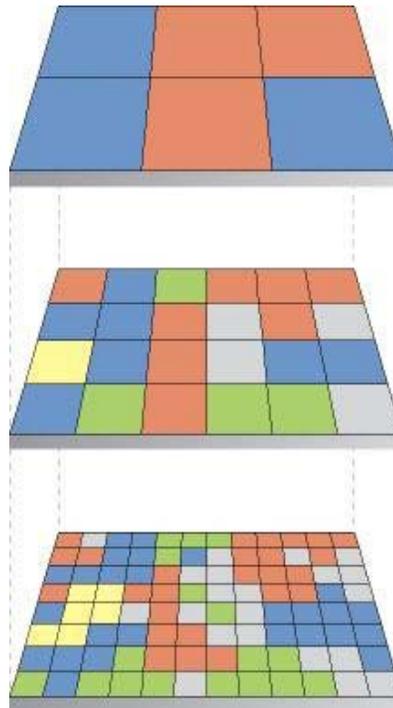
**Storage Parameters**

The user can use default storage parameters to store the raster data or set the storage parameters to suit particular data and server setup. Parameters are specified when creating or loading raster data to the database.

**Pyramids**

Pyramids are reduced resolution representations of the dataset that are used to improve display performance. The base layer of the pyramid is always the highest resolution. Pyramids can speed up display of raster data by fetching only the data at a specified resolution that is required for the display.

**Figure 3**  
**Raster Dataset Pyramid Layers**



*Pyramid creation. The spatial extent stays the same, but cell sizes change.*

Pyramids are created by resampling the original data. The resample methods instruct the server how to resample the data to build the pyramids. Three resampling methods are supported.

1. Nearest neighbor assignment should be used for nominal or ordinal data, where each value represents a class, member, or classification (categorical data such as land-use, soil, or forest type), or for color-mapped data such as scanned topographic sheets.
2. Bilinear interpolation should be used for continuous data. It interpolates four adjacent pixels.
3. Cubic convolution should also be used for continuous data. It interpolates 16 adjacent pixels.

Continuous data types include elevation, slope, intensity of noise from an airport, salinity of the groundwater near an estuary, and satellite or aerial imagery.

If continuous data is used, then benchmarking to compare image quality and performance with either of the last two interpolation methods should be used. Pyramid building is performed on the ArcSDE server side. If the original data is compressed, the server will

first decompress the data, then build the pyramids and compress the data again to insert into the block table.

### Pyramid Reference Point

The pyramid reference point is basically the virtual image origin; it is a new concept introduced at ArcSDE 9. This is an optional parameter that can be set. The concept is similar to the spatial domain of vector data. By choosing a proper value for the virtual origin, one can avoid doing full pyramid rebuild during mosaic operation—which often can be triggered if the actual image origin changes due to subsequent mosaics—and take full advantage of the partial pyramid update feature also available at ArcSDE 9 to reduce pyramid construction time.

Normally the virtual origin can be set to the upper left corner of a predefined extent, which contains all the rasters that you want to mosaic together. ArcSDE server will treat the virtual origin as the origin of the image internally and ignore the actual image origin during mosaic and pyramid construction. This, however, does not limit the user from loading rasters outside the predefined virtual origin (to its left and top), but doing so will render the virtual origin as nonexistent. It will end up with a regular mosaic as in releases prior to ArcSDE 9 where the full pyramids are rebuilt.

Setting a pyramid reference point does not necessarily modify the extent of the raster data; the point is only a property stored in ArcSDE for the server to reference when it builds pyramid layers.

### Partial Pyramid Update

This is a new feature at ArcSDE 9. In previous versions of ArcSDE, every time there were changes in the data, such as mosaicking of new data, the pyramid layers were rebuilt for the whole dataset. Rebuilding pyramids normally takes a long time for a large dataset even though the changes may be in a small area. Partial pyramid update does not rebuild the pyramid layers for the whole dataset; it only rebuilds the pyramid layers where changes have taken place. Building partial pyramids improves performance tremendously. However, partial pyramid update can be applied only when the changed area is to the right and lower sections of the pyramid reference point or virtual image origin.

### Tile Size

The tile size controls the number of pixels users want to store in each BLOB field. This is specified as a number of pixels in x and y. The default value is 128 x 128, which should be satisfactory for most applications. The best tile size setting depends on many factors such as data type (bit depth), database settings, and network settings. A smaller tile size (100 x 100) will result in more records in the raster block table, which may slow down the queries; a larger tile size (300 x 300) will require more memory to process, although it will create fewer records in the block table. If users do not want to use the default setting, they can experiment with data to choose a tile size.

### Compression

Data compression (optional, but recommended) compresses the tiles of data before storing them in the geodatabase. The compression methods available are none, LZ77, JPEG, or JPEG 2000.

The LZ77 algorithm is a "lossless" compression, meaning the unique values of cells in the raster dataset can be recovered. This is the same compression used by the PNG

image format and in ZIP compression. LZ77 compression can be applied to raster data of any type.

JPEG has a high compression ratio but is "lossy," meaning the values of cells in the raster dataset may be changed slightly. JPEG compression only applies to unsigned eight-bit data without a colormap. The user can specify quality for JPEG compression using values from 5 to 95, where 95 is the best quality and 75 is the default.

Using a wavelet compression technique, higher amounts of compression can be achieved with JPEG 2000 compression than any other compression type supported. Because it is not possible to control how JPEG 2000 builds pyramids, separate pyramids are still created. The compression quality can be expressed as a quality ranging in value between 0 and 255. Values greater than 0 compute a lossy compression, whereas a quality value set to 0 computes a lossless compression. The quality can also be specified as a fixed bit rate by supplying the bitrate=<bit\_rate> after the -q option using SDERaster command, for example, -q bitrate=1. The fixed bit rate method specifies the actual compression ratio that will be applied, whereas a quality value applies a relative compression ratio. In ArcGIS, no lossless JPEG 2000 compression is available, and the quality ranges from 1 to 255. The default quality for JPEG 2000 compression is 75. JPEG 2000 compression can be applied to 8-bit and 16-bit continuous raster data.

The primary benefit of compressing data is compressed data requires less storage space and is smaller to transfer to the client application. This results in better display performance at the client application. The amount of compression will depend on the data. The fewer unique cell values, the higher the compression ratio. The ArcSDE client performs compression and decompression. Where retention of pixel values is important, such as in categorical data or data used for analysis, use LZ77 compression. If individual pixel values are not important, such as in simple background images, use JPEG or JPEG 2000 compression.

### Compression Ratios and Qualities

Compression ratio varies on the raster data and types of compression and compression qualities. The following table shows the compression ratios for various compression types/qualities.

The original data is a DOQQ in TIFF format, the dimension is 6,614 x 7,663, three bands and eight bits, the uncompressed size is 145 MB. The data is loaded to ArcSDE without pyramids.

**Table 1  
Compression Quality and Performance**

Compression	LZ77	JPEG25	JPEG50	JPEG75	JP2 100	JP2 150	JP2 200
Storage (MB)	144	14	20	29	7	18	55
Ratio	1:1	10:1	7:1	5:1	20:1	8:1	2.6:1

The following graphics show the display qualities of the same area on the map with different compression type/quality.

J-9457

**Figure 4**  
**Compression Quality and Visualization**



**Configuration Keyword** The configuration keyword specifies options for storing the data. It is defined in the DBTUNE table. To improve performance, it is best to store the indexes and data in different locations on different physical devices. Note that the largest table will be the SDE\_blk\_n table as it holds the actual image data; all other tables are relatively small. See the *ArcSDE Configuration and Tuning Guide* for the DBMS for more information. If using Oracle, additional considerations include creating separate table spaces for the BLK and other tables. Use locally managed uniform extents with automatic segment space management. Consider varying the extent size on the BLK table based on the expected storage requirements. Choose a 16 KB or larger block size for the database.

```
AUX_STORAGE          "PCTFREE 0 INITRANS 4 TABLESPACE RASTER_OTHER"
AUX_INDEX_COMPOSITE "PCTFREE 0 INITRANS 4 TABLESPACE RASTER_OTHER
                     NOLOGGING"

BLK_STORAGE          "PCTFREE 0 INITRANS 4 TABLESPACE RASTER_BLK"
BLK_INDEX_COMPOSITE "PCTFREE 0 INITRANS 4 TABLESPACE RASTER_OTHER
                     NOLOGGING"

BND_STORAGE          "PCTFREE 0 INITRANS 4 TABLESPACE RASTER_OTHER"
BND_INDEX_COMPOSITE "PCTFREE 0 INITRANS 4 TABLESPACE RASTER_OTHER
                     NOLOGGING"
BND_INDEX_ID         "PCTFREE 0 INITRANS 4 TABLESPACE RASTER_OTHER
                     NOLOGGING"

RAS_STORAGE          "PCTFREE 0 INITRANS 4 TABLESPACE RASTER_OTHER"
RAS_INDEX_ID         "PCTFREE 0 INITRANS 4 TABLESPACE RASTER_OTHER
                     NOLOGGING"
```

**Background Value** A rotated raster or georeferenced raster may have a border of pixels that are considered as background. When the data is loaded to the database, the option is available to ignore those background pixels and turn them into nodata cells. If a background value is chosen to be ignored, all the pixels in the background with the specified value are changed to nodata pixels, but the pixels in the inside area with the specified value are intact.

**Nodata Value** Every cell location in a raster has a value assigned to it. If a cell is designated as nodata, then either no information or insufficient information about the particular characteristics of the location the cell represents is available to be able to assign the cell a value. When loading data to ArcSDE, a value can be picked to be the nodata value, in which case all the cells with that specified value will become nodata cells once loaded to the database.

**Statistics** The statistics of each band of the raster dataset in a database can be stored along with its pixel data; a histogram is also stored in the database. Statistics are normally required for displaying rasters with different stretch methods. Having current statistics built on a raster layer will always improve layer drawing performance if a stretch method is used.

Calculating statistics may take a long time for a large raster dataset. A new functionality is available in ArcSDE 9, allowing calculating statistics based on a certain pyramid layer. Since fewer pixels are scanned, the speed improves tremendously. Even though the statistics built on pyramid layers are not as accurate as the base layer, it is sufficient for displaying.

**Colormap** For a discrete raster dataset, for example, a land-use map, the value of each cell in the dataset does not have a real meaning but, rather, an index. It is common for the user to display a certain land-use type with a specified color; the relationship between the index and the color forms a colormap. The colormap of a raster dataset can be stored in the database as a BLOB if desired.

**Metadata** Metadata stores additional information on the raster data, such as date taken, sensor information, and spatial information. A raster dataset's metadata is stored in the system metadata table, and loading a raster dataset to the geodatabase will automatically import its metadata to the geodatabase. For a raster catalog, the metadata for each raster dataset in the catalog is optional; if you want to have the metadata stored, you need to add a field METADATA of type BLOB to the raster catalog table.

**Data Loading** The data loading process consists of preprocessing the raster images (if required), defining a set of appropriate parameters to hold the raster, and actually populating the Raster Columns table and a set of ArcSDE system tables.

There are several ways to load raster data into a database. This document gives an outline of

- ArcGIS/Geoprocessing—graphical user interface (GUI)
- SDERASTER—command line loader
- ArcObjects COM API—customized application
- Geoprocessing functions—scripting

### ***Preprocessing of Continuous Rasters***

The preprocessing of images can be done in software products such as ArcInfo<sup>®</sup>; ERDAS IMAGINE<sup>®</sup>; and to a limited extent, ArcView<sup>®</sup>. Preprocessing of data is most important for people wanting to create a seamless raster layer, edgematch, spectral match, or georeference before loading a collection of images. The ArcObjects Developer Help has several sample scripts that can help with preprocessing of images.

### ***Loading Parameters***

In ArcGIS, rasters can be loaded into the database as individual layers or as seamless layers by appending them to an existing raster. For the case of mosaicking rasters into a spatial continuous raster dataset in the database, all the rasters must have the same spatial definition and data type. If the input raster data has different properties, the data will be projected, resampled, and rescaled on the fly to the properties of the target raster.

Default storage parameters include

- Spatial reference: the spatial reference of the raster dataset
- Statistics: calculate with skip factor 1,1
- Compression type: LZ77
- Compression quality: 75
- Tile size: 128 x 128
- Pyramid option: builds pyramid
- Pyramid Level: builds full pyramids
- Pyramid resample method: nearest neighbor
- Configuration keyword: defaults
- Ignore background value: none

- Nodata value: none
- Mosaic method: last
- Colormap mode: first
- Mosaicking tolerance: 0
- Converting one-bit data to eight bit: false
- Colormap to RGB: false

Users can specify whether they want to generate reduced resolution layers (pyramid layers) and how the pyramid layers will be built. The reduced resolution layers cut down on access time in passing data up to the client by grabbing only the coarsest possible resolution of imagery to paint the screen. It increases database size but decreases access time.

Colormaps are automatically loaded to the database when raster data is loaded, if available. The colormap is used to map pixel values to RGB colors so the raster data is displayed the same way by default.

Statistics storage is calculated on request. When loading raster data to the database, the user can specify whether the statistics will be calculated and stored in the database. Since calculation of statistics on large datasets can be time consuming if the statistics are calculated based on the base data, it is suggested that they be calculated once the data is loaded if the data is to be permanent in the database. Another option for calculating statistics is to build the statistics on a pyramid layer instead of base data. Since statistics are primarily for display, they do not need to be very accurate. Calculating statistics on a pyramid layer can reduce the building time tremendously for a large dataset because it is working on a smaller number of pixels.

The pyramid layers on a one-bit file raster are built as eight bit so the bilinear resample method can be applied. As a result, it renders a more pleasant display. But ArcSDE raster requires that the base data and pyramid layers are stored with the same pixel type. To achieve similar display effect of a one-bit file raster for the same raster stored in ArcSDE, it is necessary to convert the one-bit raster to eight-bit so the pyramids are built with bilinear resample method. The one-bit to eight-bit data option allows the user to load a one-bit raster dataset to an eight-bit raster dataset by changing the value of 1 in the input raster dataset to 255 in the output raster dataset.

Raster with colormap is discrete in nature; the only available pyramid resampling method is nearest neighbor. Most of the time the pyramid layer of a raster with colormap displays as discontinuous patches. The colors in a colormap are a combination of RED, GREEN, and BLUE elements, and it is possible to convert a color into three components representing RED, GREEN, and BLUE. Storing a colormapped raster as a three-band raster enables the bilinear and cubic resampling method for pyramid construction, which in turn, renders a continuous display.

### ***Data Loading in ArcGIS***

The source raster can either be file based or in a database, and it can be imported to an existing or new SDE raster, stand-alone, or in a raster catalog. There are a few

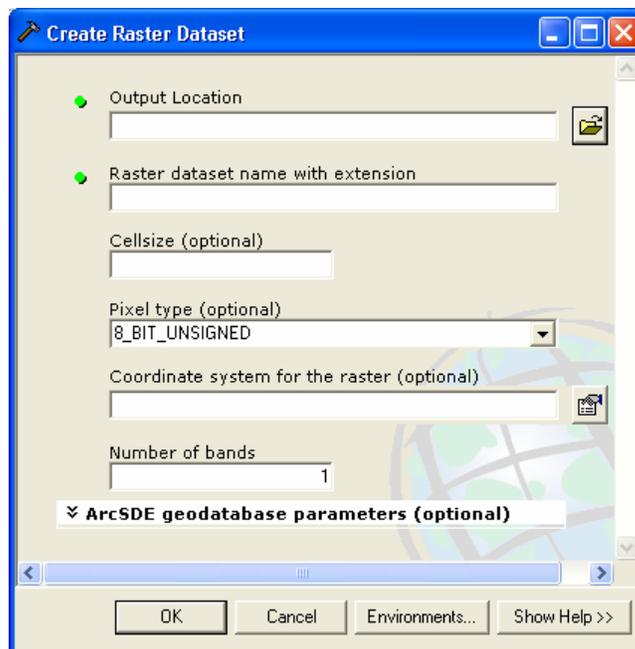
geoprocessing tools in ArcGIS that can be used to load raster data. These loading tools can be accessed from ArcToolbox™ and the context menu on the database connection in ArcCatalog™.

***Create a New SDE Raster***

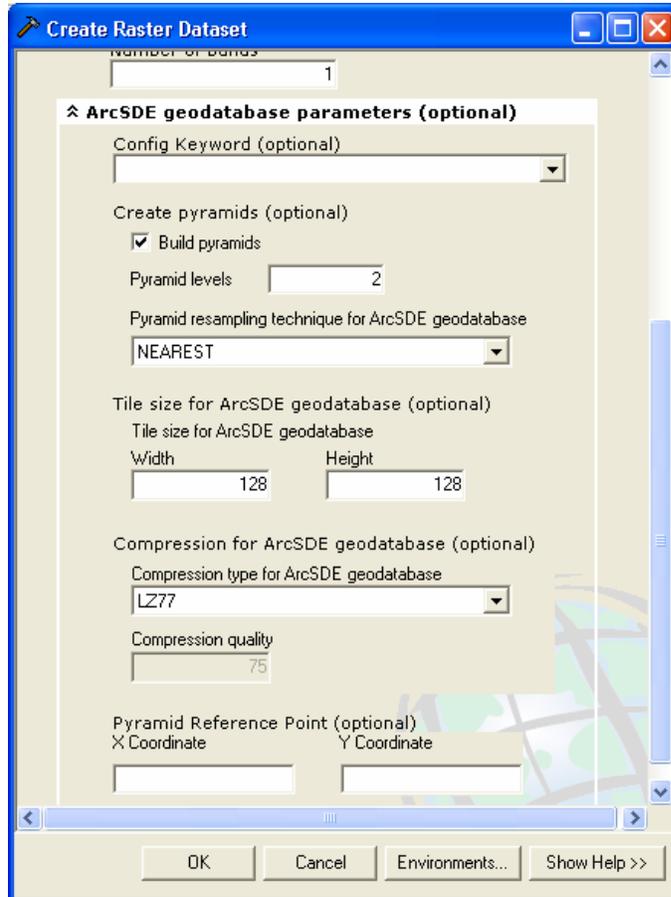
A new raster can be created in a geodatabase by selecting the New> Raster Dataset menu item from the database connection or launching the CreateRasterDataset tool from ArcToolbox. The new raster dataset in the database will have one cell; it is a holder for some raster properties, for example, pixel type, cell size (if specified), spatial reference, and number of band. Be sure to expand the ArcSDE geodatabase parameters section to define the storage parameters. Only the storage parameters on the dialog box are applied to the new raster dataset; any storage parameter settings in the ArcToolbox environment settings are ignored.

Normally a new raster dataset is created for mosaicking operations to take advantage of partial pyramid updates. The pyramid reference point must be set properly according to the top left corner of all the rasters that will be mosaicked to it. Once the reference point is set, it cannot be moved without rebuilding the whole pyramid. If the reference point is not set, the first raster that is mosaicked to it will set the point with the coordinates of its top left corner.

**Figure 5**  
**Create Raster Dataset Geoprocessing Tool**



**Figure 6**  
**Storage Parameters for ArcSDE Raster Dataset**

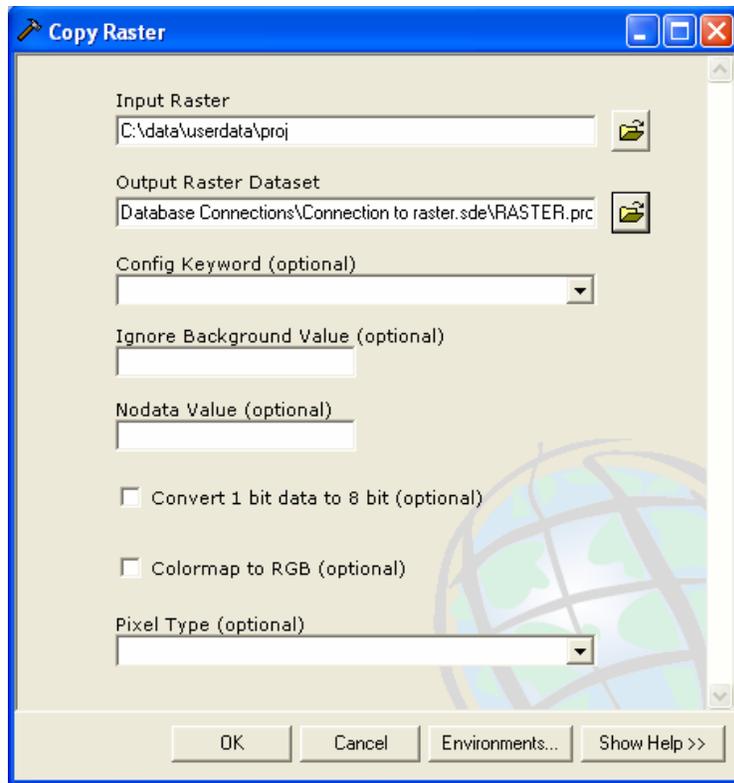


***Import a Raster Dataset***

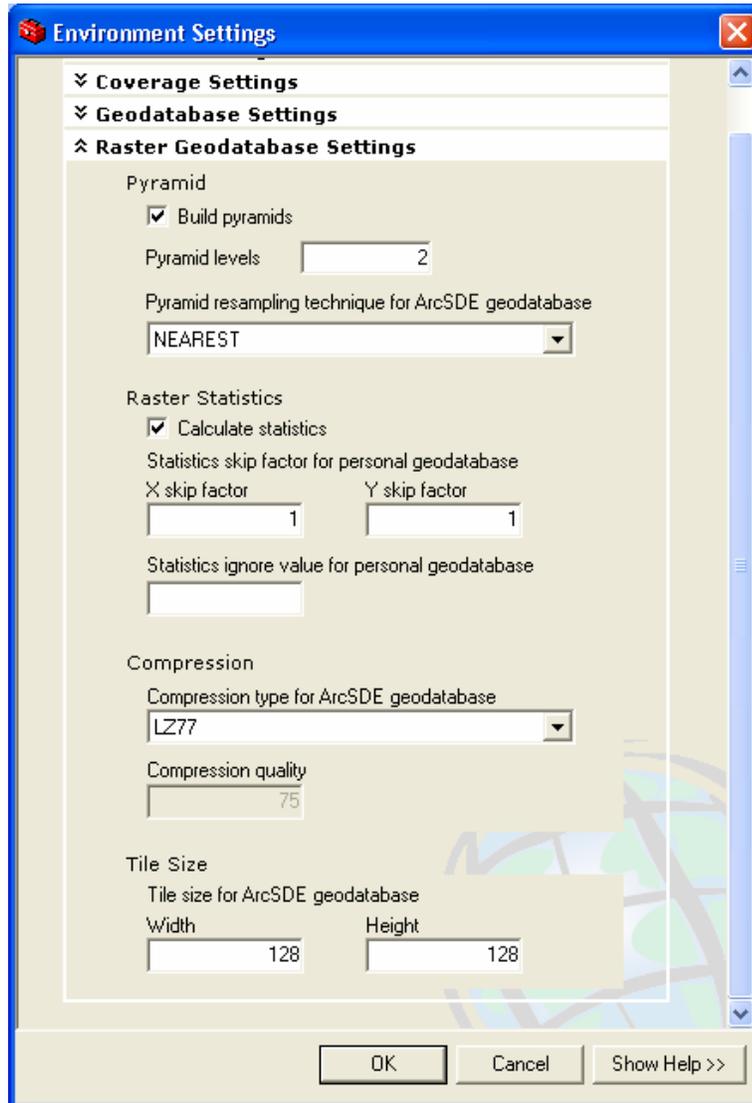
The CopyRaster geoprocessing tool loads a raster dataset to another raster dataset. This tool can be accessed from ArcToolbox or the raster dataset context menu Export> Raster To Different Format. The raster properties, such as cell size, spatial reference, and pixel type, are not changed in the output raster dataset; however, you can change other properties. If the data has a border that needs to be excluded, the background value can be specified so the cells at the border with that value will be changed to nodata cells. If a certain value in the whole raster dataset needs to be excluded, specify that value as nodata. If the input raster is a one-bit raster, it can be loaded as an eight-bit raster by checking the check box next to Convert one-bit data to eight bit; doing so will make the raster more easily viewed when zooming out. A raster with colormap can be converted to a three-band raster dataset, so the display is more pleasant since bilinear interpolation can be used to build pyramids and resample. If a file raster is stored in a higher bit depth (e.g., 16 bit), most of the pixels are in the low bit (e.g., 8 bit); if the pixels with values in the high bit are nodata cells, you can change the bit depth when loading to the geodatabase without losing any pixel information and at the same time save storage space. This is useful when loading rotated or projected rasters with irregularly shaped background areas.

The storage parameters are taken from Raster Geodatabase Settings in the Environment Settings; you can specify the parameters to be used to store the raster data in database.

**Figure 7**  
**Copy Raster Geoprocessing Tool**



**Figure 8**  
**Environment Settings for Geodatabase Raster Dataset**



***Mosaic Rasters to an Existing SDE Raster***

Multiple input rasters can be mosaicked to an existing raster dataset in the database using the Mosaic tool. The tool can be accessed from the raster dataset context menu and ArcToolbox.

To mosaic to the existing raster, the input rasters must have the same number of bands as the existing raster dataset. If the input rasters have different number of bands from the existing raster dataset, the operation will fail. If the input rasters have different spatial references, the input rasters will be projected to the spatial reference of the existing raster on the fly. If the input rasters have different cell sizes, resampling will take place to

adjust the cell size to be the same as the target raster dataset. Rasters with different pixel types can be mosaicked by scaling the input rasters on the fly.

When mosaicking rasters with overlapping areas, one of the mosaicking modes can be applied. By default, the LAST option is used upon which the new data values will replace the original data values in the overlapping area. The FIRST option will retain the values of the original dataset and ignore the incoming rasters in the overlapping area. The MEAN option averages the overlapped cell values from the input rasters and the original raster, the MAX option picks the maximum cell value, and MIN takes the minimum cell value. The BLEND option calculates a new cell value based on the weight of the input raster dataset and the original data at that cell location.

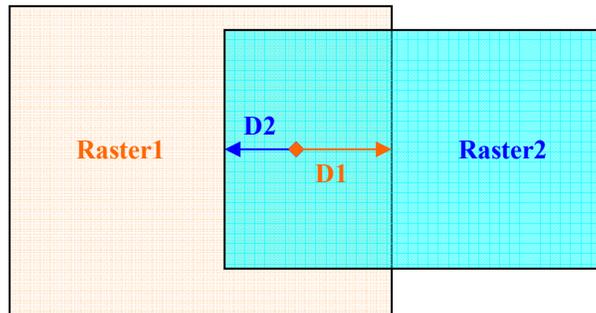
It is worth noting that if a mosaic mode other than LAST is used, it will take longer to mosaic the data since the original data has to be pulled out and operated on.

The algorithm for BLEND option is

$$V = ( V1 * D1 + V2 * D2 ) / ( D1 + D2 )$$

Where V is the calculated value of the overlapping cell, V1 is the value of the cell in Raster1, V2 is the value of the same cell in Raster2, D1 is the distance from the cell to the right edge of Raster1, and D2 is the distance from the cell to the left edge of Raster2.

**Figure 9**  
**Algorithm for BLEND Mosaic Mode**



When mosaicking rasters with colormap, one of the colormap modes can be chosen to instruct the mosaicking operation on how to coordinate all the colormaps from the input and the target rasters. FIRST is the default colormap mode, where the colormap of the target raster keeps intact. REJECT mode will not allow the mosaicking operation if the input raster or the target raster has a colormap. MATCH mode examines the colormaps of all the rasters and combines all the colors into one set of colors to form a colormap. LAST mode applies the colormap of the last raster in the list to the target raster.

Similar to CopyRaster tool, you can also specify a background value to be ignored, a nodata value to be excluded, and one-bit to eight-bit conversion.

***Load Raster to a Table***

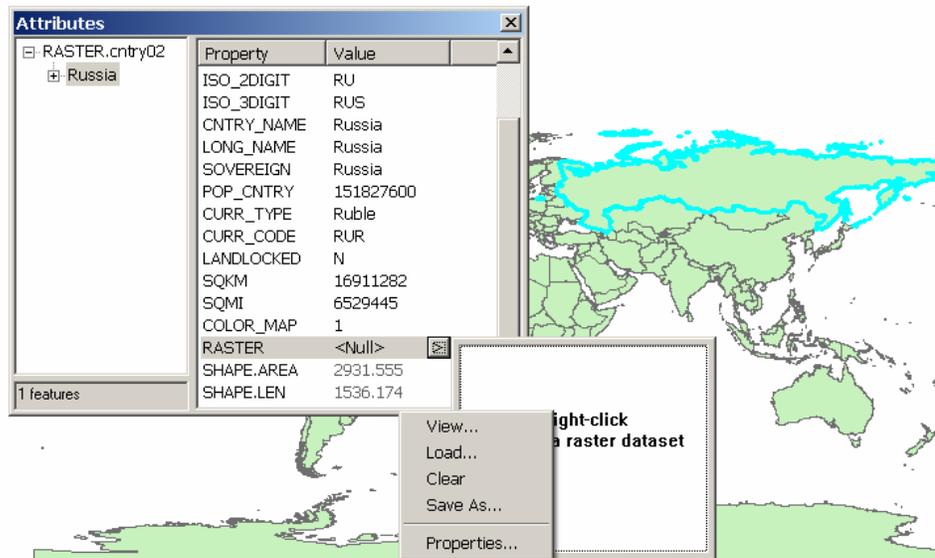
Raster field is now a standard field type in the geodatabase. It can be added to any table or feature class. Adding a raster field to a table is no different from adding any other type of fields. Editing a raster field in a table is the same as editing any other type of field in a table. The table needs to be registered as versioned before the fields can be modified. The workflow is similar to that of editing a raster field in a feature class, which is discussed next.

Only one raster field is allowed in a table.

***Load Raster to a Feature Class***

The raster field in a feature class can be edited in ArcMap after the feature class is registered. The editing process is similar to that of editing other types of fields. While in the editing session, you can select a feature and open the attribute inspector, then locate the raster field and modify it.

**Figure 10**  
**Editing Raster Attribute of a Feature Class**



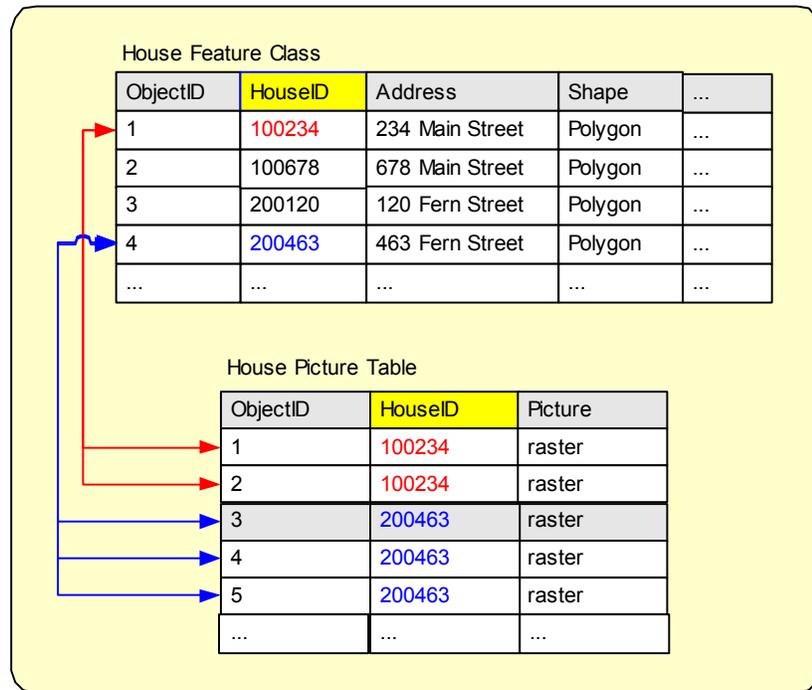
***Multiple Raster Attributes for a Feature Class***

In some cases, a feature class may have more than one raster field to associate with, for example, a real estate feature class with all the houses on the market needs multiple pictures of the same house taken from different angles. The restriction of one raster column per table limits the option for multiple rasters in the same feature class. However, this can be accomplished by a relationship class based on the house feature class and a table with a raster column storing all the house pictures using a one-to-many relationship.

Here is an example in which a feature class of houses has more than one raster related to each feature in the feature class. The feature class has a unique field HOUSEID for house identification. The table has a field HOUSEID and a field PICTURE, which is a type of raster. Each HOUSEID in the table has one or many entries, and each entry

represents one picture for that HOUSEID. A relationship class is established on the house feature class and house picture table on the common field of HOUSEID; therefore, the pictures for each feature can be accessed by identifying the feature.

**Figure 11**  
**Relationship of House Feature Class and House Picture Table**



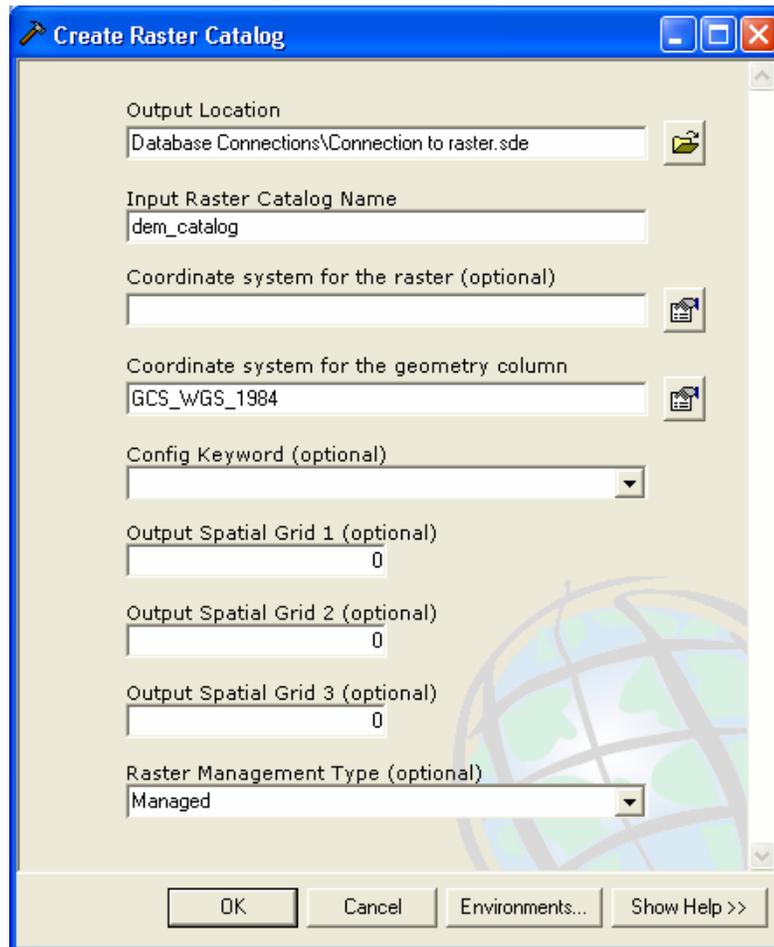
**Create a Raster Catalog**

A raster catalog is a way of organizing individual rasters as a seamless layer without having to mosaic them. They are useful for users who expect to get frequent data updates, need to maintain the data from areas where images overlap, or need to manage their holdings. Each raster in a catalog must use the same spatial reference and should have the same spectral properties.

When creating a raster catalog, both raster column and footprint column properties need to be specified. Just as when you create a feature class, it is especially important to set up an appropriate geometry column spatial domain so that all the raster datasets will fit in the spatial domain to prevent error when loading the raster data.

The Create Raster Catalog tool can be launched from ArcToolbox or a geodatabase connection to create a raster catalog. The output geodatabase location and raster catalog name need to be specified. In most cases, it is necessary to specify the coordinate system including spatial domain for the geometry column instead of using the default.

**Figure 12**  
**Create Raster Catalog Geoprocessing Tool**



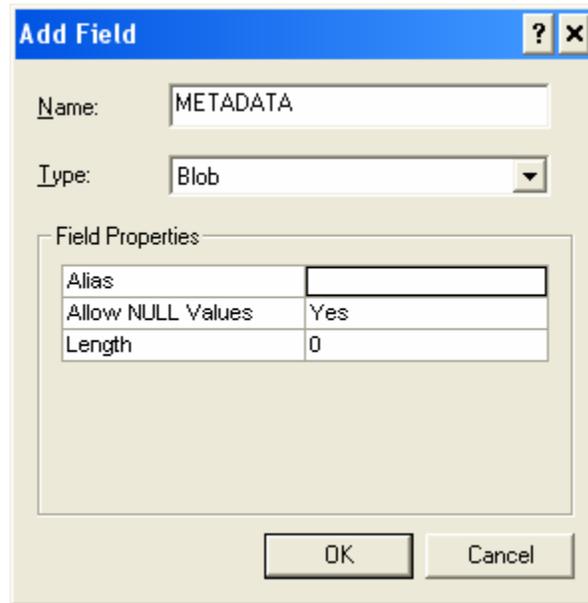
***Insert to a Raster Catalog***

When raster datasets are inserted to a raster catalog using ArcToolbox tools, the extent of each raster dataset is extracted and projected to the coordinate system of the footprint column. The projected extent is then stored in the footprint column. If the raster dataset has a different spatial reference from that of the raster column, the raster will be projected on the fly to the spatial reference of the raster column and stored in the database. The storage parameters set in the Environment/Geodatabase Raster Settings are used to store the data.

***Add Metadata Column to a Raster Catalog***

By default, the metadata for each raster dataset in the raster catalog is not stored. To store the metadata, you need to add a column called METADATA of type BLOB to the raster catalog table once it is created.

**Figure 13**  
**Add Metadata Field to a Raster Catalog**



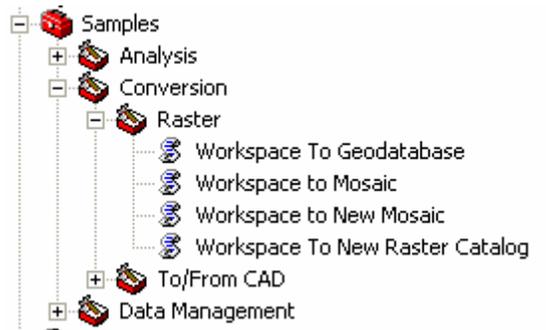
***Batch Loading***

The Raster to Geodatabase (multiple) tool in ArcToolbox supports batch loading of multiple rasters into a geodatabase. The raster datasets can be loaded as individual raster datasets in the geodatabase or as raster catalog items in a raster catalog. The storage parameters to store the rasters are obtained from Raster Geodatabase Settings in ArcToolbox Environment Settings. The input raster datasets can be file-based rasters, geodatabase stand-alone raster datasets, or raster datasets in a raster catalog.

***Load Rasters in a Workspace***

There are four tools in the Sample toolbox for loading rasters in a workspace to a geodatabase, mosaicking to an existing raster dataset in geodatabase, mosaicking to a new raster dataset in the geodatabase, and loading to a new raster catalog in the geodatabase. The WorkspaceToNewRasterCatalog tool accumulates the spatial extents of all the rasters in the input workspace and calculates an appropriate spatial domain for the footprint column for the new raster catalog. The geometry column has the same spatial reference as the first raster in the input workspace.

**Figure 14**  
**Samples Toolbox**



**Data Export**

Once in a while, you may want to extract certain raster datasets, raster datasets in a raster catalog, or a portion of them to another location, either another geodatabase or file. With the rich ArcToolbox tools and functions in the ArcGIS Desktop user interface, it is easy to accomplish data export.

***Export Raster Dataset***

Raster datasets can be exported to file-based raster datasets, geodatabase raster datasets, or raster datasets in a raster catalog. The CopyRaster tool provides an easy way to perform this task. A dataset can be exported as a whole or extracted by a specified envelope. The Clip tool can be used to extract a raster dataset by providing the bounding extent. Or if you are using ArcMap, you can use the Data Export menu to export a portion of the data; you can also specify the output raster cell size.

***Export Rasters from a Raster Catalog***

Raster datasets in a raster catalog are treated the same as stand-alone raster datasets in the geodatabase. If you know what datasets you will be exporting from a raster catalog, you can use CopyRaster tool and populate the input rasters parameter with the desired raster dataset in the raster catalog by multiple selecting and dragging.

***Export Footprints from a Raster Catalog***

The footprints stored in a raster catalog identify the spatial distributions of the raster datasets in that raster catalog and can be used for other analysis. If the analysis only uses the footprints column, it is wise to export the footprints to another feature class without the raster column. The Copy Feature tool accepts a raster catalog as input and exports its geometry column to an output feature class.

***Extract Features and Rasters in a Certain Region***

If it is desired to extract both raster data and feature data stored in the same geodatabase for a certain area for disconnected editing or data distribution, this can be done using the Disconnected Editing toolbar in ArcMap. The output collection of raster and feature data can be stored in another geodatabase, either personal or enterprise.

**Transferring between Databases**

In ArcGIS 9.1, it is possible to move raster data within or between databases by copying/pasting; however, the operation will not maintain the raster storage properties but will reload the data using the default storage parameters. If it is a raster catalog, the geometry column will be populated with the extent of each raster dataset in the raster catalog whether the geometry column is populated or not in the original raster catalog.

Details on data transferring options are discussed in the *Managing a Raster Database* technical paper.

### **Create a Raster Catalog View**

For security and management reasons, you may want to grant privileges on certain rasters in a raster catalog to a certain user, so that user can only access the rasters that he/she has been granted and cannot access the rest in the raster catalog. For instance, if you have a raster catalog covering the whole state, you can add an additional attribute field with county names, then grant privileges based on the county name to the analysts who are working on a particular county.

To create a database view for a raster catalog, you will need to run the following ArcSDE command, specifying the criteria in the `where_clause`.

```
sdetable -o create_view -T <view_name>...
```

Once the spatial view is created, it will be shown as a feature class in ArcGIS since it is not registered with the geodatabase properly. Right-click the feature class and register with geodatabase from ArcCatalog, and it will be registered as a raster catalog view.

Now that the view is just like a raster catalog, you can grant privilege on that raster catalog view to other users. They will only be able to access the raster datasets that are available in the raster catalog view.

### **Data Viewing**

Rasters in a geodatabase behave the same as rasters in a file system, just as features in a database are treated similar to any other feature source. The difference is that you have to make a connection to the database to access the raster data in that database. In ArcGIS 9, many improvements have been made to access and display raster catalogs.

### ***Viewing from ArcGIS***

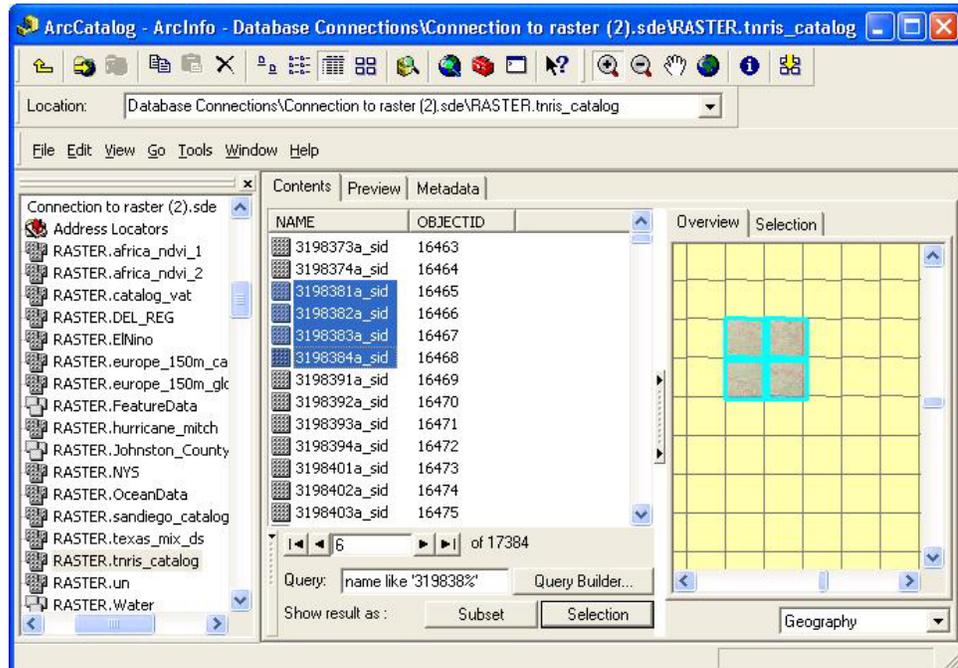
#### **ArcCatalog Application**

ArcCatalog provides the environment for managing data including creating and loading raster data. The data can also be viewed as image or table. For raster catalogs, an improved user interface offers more flexibility for the user to examine the raster data and metadata in the raster catalog.

#### **Raster Catalog View**

The raster catalog view can display not only the raster catalog as a continuous raster layer but also its table of contents. The overview panel in the contents view makes it possible to display only selected raster datasets in the context of the raster catalog, so the geographic distribution of the selected raster datasets can be revealed. The selection panel displays only the selected raster datasets without the raster catalog layer as background. Another benefit of the overview panel is the ability to access the properties and metadata of each individual raster dataset. A query filter can also be used to select multiple raster datasets.

**Figure 15**  
**Raster Catalog View in ArcCatalog**



**Adding/Deleting Fields**

Similar to a feature class, a raster catalog's schema can be modified by adding and deleting fields from its table representation. Neither raster type nor geometry type field can be added to or removed from a raster catalog.

**Viewing Raster as Attribute**

Raster can be added to a feature class or a table as an attribute. The raster attribute can then be accessed by identifying the features, the raster attribute is listed on the identify dialog box, and the associated raster can be displayed.

**ArcMap Application**

**Raster Catalog Selection**

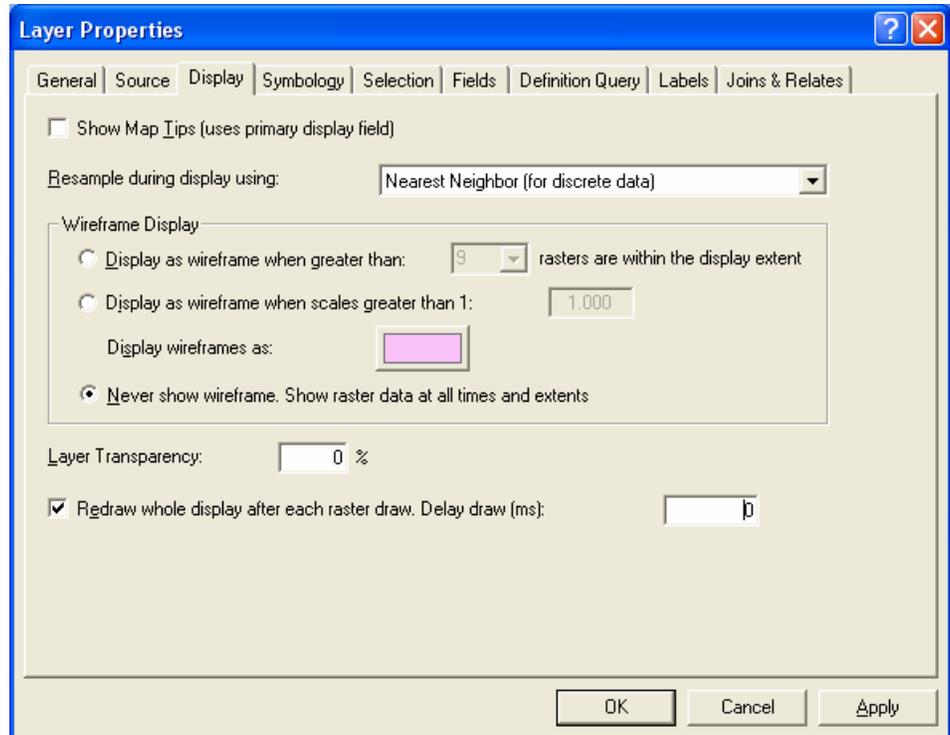
Raster catalogs sometimes can be huge, and display of the whole layer does not give much information to the user, especially for analysis or editing. The selection functionality in ArcMap provides the ability for the user to select rasters in the raster catalog. Similar to a feature class, a raster catalog layer supports attribute, spatial, and graphic selection. The user can select rasters in a geographic area or rasters related to other features spatially. Attribute selection can be applied so only the rasters that satisfy the condition are selected.

**Viewing Time Series Raster Catalog**

A time series raster catalog normally stores a series of rasters for a certain region over time. The footprints are identical for each of the raster datasets in the raster catalog. If displayed as a raster catalog layer, it will only show the last one because they overlap completely. A new mechanism is developed in ArcMap to view time series raster

catalogs one raster at a time with the effect of animation, and the delay time between each raster display can be set.

**Figure 16**  
**Settings for Time Series Raster Catalog**



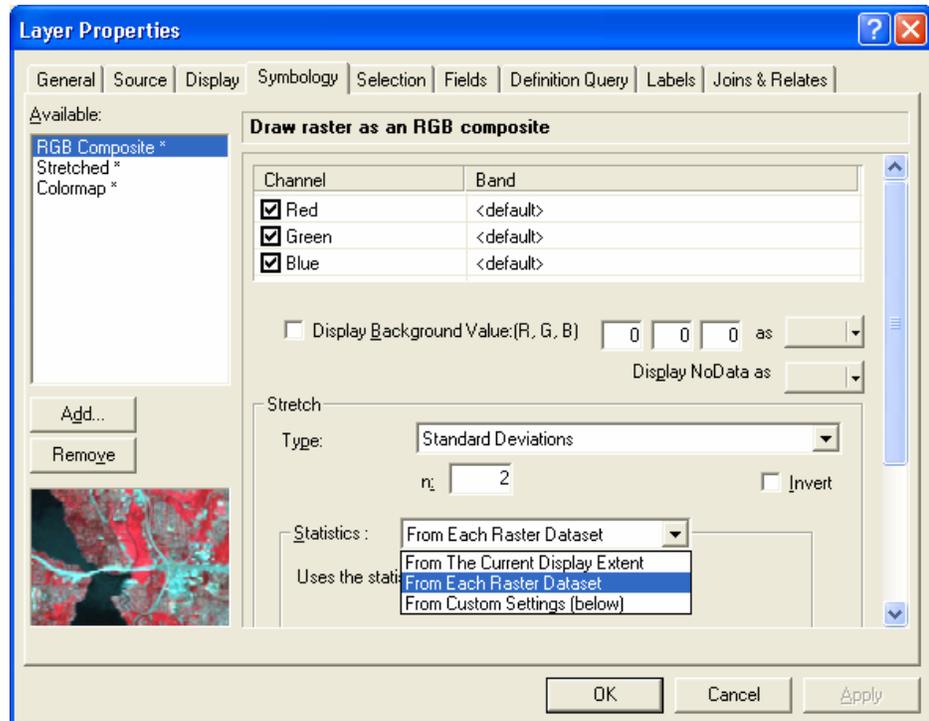
### Raster Catalog Rendering

In previous releases, the raster catalog layer can only be rendered with one raster render type. If the raster catalog is heterogeneous, some of the rasters may not be displayed simply because the render is not suitable for those types of raster. For example, a single band raster cannot be rendered with an RGB render. In ArcGIS 9, rendering of raster catalogs has been improved. Raster catalogs have the same new rendering capabilities as raster datasets. A raster catalog can be displayed with different render types depending on the properties of each raster dataset in the raster catalog. By default, a list of renders is available in the order of their priorities. The order is RGB composite render, stretched render, and colormap render. When displaying a raster catalog, the renders in the list are tried on each raster in order and the most appropriate renderer is automatically applied to the suitable raster in the raster catalog. The available render list can also be modified, so you can add or remove various render types to suit your raster catalog layer.

Another improvement in raster catalog display is the option to use one set of statistics on all the raster datasets in the raster catalog. This is particularly useful when the raster datasets in the raster catalog have different statistics. For instance, you have a raster catalog full of raster datasets of elevation and each dataset has a different range of values and statistics. By default, the statistics of each raster are used on each raster dataset, and

the display will look discontinuous. Once unified statistics are applied to all the rasters, the display becomes continuous. The statistics can either be from an existing raster dataset or imported from an XML file.

**Figure 17**  
**Render with Custom Statistics**



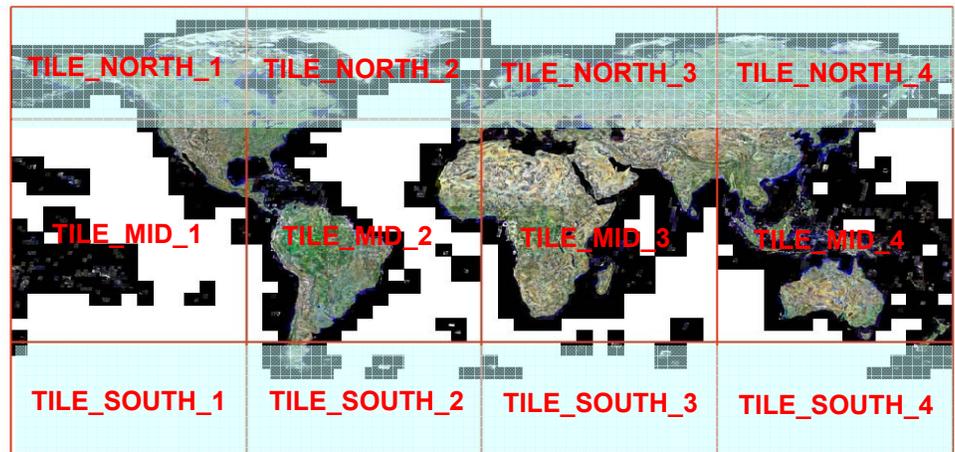
**ArcGlobe** The ArcGlobe™ application included with ArcGIS 3D Analyst™ enables users to manage and visualize, from a local or global perspective, extremely large sets of three-dimensional geographic data. ArcGIS 3D Analyst provides the capability to seamlessly interact with any geographic information as data layers on a three-dimensional globe.

Any layer added to an ArcGlobe application is projected to the Cube projection. If the data layer is not in Cube projection originally, it will be projected on the fly. Considering the specific requirements of ArcGlobe, if your raster data stored in ArcSDE is primarily used for ArcGlobe applications, Cube projection is recommended.

- **Pyramids**—Creating pyramids on a large raster dataset is always necessary to achieve the best performance. It also helps in building the data cache.
- **Projection**—Since ArcGlobe uses Cube projection natively, it is beneficial to store your raster data in Cube projection in ArcSDE, so no on-the-fly reprojection is performed when the data is visualized in ArcGlobe. It will also reduce the time consumed for building layer data cache.

- **Layer Data Cache**—Generating a Layer Data Cache does not have to happen on the ArcSDE server machine where all the data is stored. As ArcGlobe is a multithreaded application, building full data cache on multiple CPU machines will cut the time required for building the data cache. Also, machines with good disk access rates help to reduce the time required to commit the data cache to disk. Using compression for the data cache reduces the cache size. In addition, this will lower memory usage when the data cache is used in ArcGlobe. It is recommended that you use the JPEG compression with its default setting when building data cache.
  
- **Data organization**—There are two options to organize the data, a raster catalog of multiple raster datasets or multiple stand-alone raster datasets. A proper organization of data is dependent on mode of data visualization to be used in ArcGlobe. For on-demand mode of data visualization, a raster catalog or multiple datasets do not have significant impact on performance. For full layer data caching, raster catalogs may pose a problem because the resulting cache data is a combination of all the raster datasets in the catalog; as a result, if building full data caching is intended, it is optimal to store multiple stand-alone raster datasets. When working with continental or worldwide data, dividing the datasets conforming to the Globe cube faces (see figure) results in an optimal performance especially for an on-demand mode of visualization in ArcGlobe. There are a total of 12 cube faces, of which the north and south four faces can be combined into one single tile. This will result in six tiles that are coincident with six cube faces.

**Figure 18**  
**Globe Domains**



**ArcIMS**

ArcSDE raster data can be served over the Internet using ArcIMS®. Use a text editor to create the AXL file referencing your raster layer. The following is an example of <MAP> tag in an AXL file.

```
<MAP>
<PROPERTIES>
  <ENVELOPE minx="-125" miny="20" maxx="-80" maxy="60"
    name="Initial_Extent" />
```

```

    <MAPUNITS units="decimal_degrees" />
</PROPERTIES>
<WORKSPACES>
    <SDEWORKSPACE name="sde_ws-0" server="spserver"
        instance="port:5151" database="raster" user="raster"
        password="go" geoindexdir="C:\Temp\" />
</WORKSPACES>
<LAYER type="image" name="US NED" visible="true" id="0">
    <DATASET name="RASTER.RASTER.US_NED.RASTER"
        workspace="sde_ws-0" />
</LAYER>
</MAP>

```

## Personal Geodatabase

As an alternative to ArcSDE, raster data now can be stored in a personal geodatabase in two management types—managed and unmanaged. Raster data stored in a managed manner is first converted to an IMG file in a specific location relative to the Access database, and the information is recorded in the database. The raster object in the database will have full control over the IMG file and can modify or delete the file. In an unmanaged situation, the raster data stays untouched in its original form, and a pointer to the file location is stored in the database. The raster object in the database can read the data but cannot modify the data like a managed raster object can. The raster dataset in a personal geodatabase can only be in managed type, while the raster catalog can be either managed or unmanaged. The management type is specified when creating a raster catalog.

## Upgrading from ArcSDE 8.3 to ArcSDE 9

Raster data already stored in ArcSDE 8.3 needs to be upgraded once the environment changes to ArcSDE 9. It is easy to upgrade from ArcSDE 8.3 to ArcSDE 9. Raster datasets stored in the database can be used without any process after ArcSDE is upgraded from 8.3 to 9. For embedded raster catalog, one simple step in the ArcCatalog user interface will complete the upgrade process. The Register with Geodatabase context menu item is designed for this use and basically adds a geometry column to the raster catalog table and populates the column with the extents of each raster. Reference raster catalogs stored in the database are no longer supported in ArcSDE 9. To find more details on upgrading raster data, refer to the *Upgrading Geodatabase Raster Data from ArcGIS 8.3 to ArcGIS 9* technical paper.

## Custom Applications

ArcObjects is a wrapper of the ArcSDE C API. The ArcGIS loading application is a Windows® application written using ArcObjects and runs on both ArcCatalog and ArcMap. The C API has the core functionalities for managing raster data in a database.

When writing customized loading applications, the ArcSDE C API, Java API, or ArcGIS ArcObjects can be used to suit different situations. ArcObjects is not only available on the Windows platform but also on UNIX®/Linux® by using the Java edition. The ArcSDE C and Java APIs are available on all supported ArcSDE platforms.

The C and Java APIs for raster data implement Raster Streams, Raster Column, Raster Value, Raster Band, Raster Cell, Raster Block, Raster Pyramids, and Raster Band Statistics. Logically, the SDE Raster API is similar to the SDE Geometry API with

Streams, Layers, Geometry Columns, and Shapes. Applications access the pixel data of a raster or Raster Band through a Raster Stream.

See the *ArcSDE Developers Guide* for more information.

## ArcObjects Sample

### *ArcObjects for Loading*

ArcObjects functions are implemented based on ArcSDE C APIs. These functions are primarily for loading rasters into a database. Since the new schema of raster catalog is similar to a feature class, many interfaces that work with only feature classes can be applied to raster catalog as well. ArcObjects is also available in the .NET environment.

### Loading Sample

The following two samples demonstrate the workflow for loading raster data to a raster dataset or raster catalog. These and other raster samples can also be found in the *ArcObjects Developer Help* (<http://arcgisdeveloperonline.esri.com/ArcGISDeveloper>). It is worth mentioning that there may be many different ways to accomplish one loading task; the samples shown here only represent one of them.

#### Example 1

**Load raster datasets in a directory to a seamless raster dataset in the geodatabase.**

```
Public Sub DirToMosaic(sDir As String, pDataset As _
    IRasterDataset)
    'load all raster datasets in the input directory to the
    'raster dataset
    'the geodatabase raster dataset has to exist, if not
    'create it first
    'with proper storage parameters.

    Dim pEnumDatasets As IEnumDataset
    Dim pRasterDs As IRasterDataset2
    Dim pWsFact As IWorkspaceFactory
    Dim pFileWs As IWorkspace
    Dim pSDEds As IRasterDatasetEdit

    On Error GoTo er

    'get the list of datasets in the dir
    Set pWsFact = New RasterWorkspaceFactory
    Set pFileWs = pWsFact.OpenFromFile(sDir, 0)
    Set pEnumDatasets = pFileWs.Datasets(esriDTRasterDataset)
    pEnumDatasets.Reset

    Set pSDEds = pDataset

    'load raster datasets from the dir
    Set pRasterDs = pEnumDatasets.Next
```

```

Do While Not pRasterDs Is Nothing
  Debug.Print pRasterDs.CompleteName
  pSDEs.Mosaic pRasterDs.CreateFullRaster, 0.5
  Set pRasterDs = pEunmDatasets.Next
Loop

'cleanup
Set pEunmDatasets = Nothing
Set pRasterDs = Nothing
Set pWsFact = Nothing
Set pSDEs = Nothing
Set pFileWs = Nothing
Exit Sub
er:
  MsgBox err.Description, , "Directory to mosaic"
End Sub

```

### Example 2

#### Batch Inserting Rasters from a File Directory to a Raster Catalog in the Database

```

Public Sub DirToGDBCatalog(pWs As IWorkspace, pCatalog As _
IFeatureClass)
  'insert all the rasters in the input workspace to the
  ' geodatabase raster catalog
  'the geodatabase raster catalog has to exist, if not
  ' create it first with proper
  'definition for both raster and geometry column.

  Dim pSDERasterWs As IRasterWorkspaceEx
  Dim pEunmDatasets As IEnumDataset
  Dim pCatalogFeature As IRasterCatalogItem
  Dim pRasterDs As IRasterDataset
  Dim pCursor As IFeatureCursor
  Dim pRow As IFeatureBuffer

  'get the list of datasets in the input workspace
  Set pEunmDatasets = pWs.Datasets(esriDTRasterDataset)
  pEunmDatasets.Reset

  'load raster datasets from the input workspace
  Set pRasterDs = pEunmDatasets.Next
  Set pCursor = pCatalog.Insert(False)

  Dim pCat As IRasterCatalog
  Set pCat = pCatalog
  Dim pDs As IDataset

```

J-9457

```

'loop through all the datasets and load
Do While Not pRasterDs Is Nothing
    Set pDs = pRasterDs
    Set pRow = pCatalog.CreateFeatureBuffer
    pRow.Value(pCat.RasterFieldIndex) = _
    createRasterValue(pRasterDs)
    pCursor.InsertFeature pRow
    Set pRasterDs = pEunmDatasets.Next
Loop

'cleanup
Set pCatalog = Nothing
Set pSDERasterWs = Nothing
Set pEunmDatasets = Nothing
Set pCatalogFeature = Nothing
Set pRasterDs = Nothing

End Sub

Public Function createRasterValue(pRasterDs As _
IRasterDataset) As IRasterValue
    Dim pRasterVal As IRasterValue
    Dim pRasStoreDef As IRasterStorageDef
    On Error GoTo er

    '++ set storage parameter
    Set pRasStoreDef = New RasterStorageDef

    '++ use LZ77 compression, bilinear interpolation for full
    'pyramids building
    pRasStoreDef.CompressionType = _
        esriRasterSdeCompressionTypeRunLength
    pRasStoreDef.PyramidResampleType = _
        RSP_BilinearInterpolation
    pRasStoreDef.PyramidLevel = -1

    '++ set raster value to raster field
    Set pRasterVal = New RasterValue
    Set pRasterVal.RasterDataset = pRasterDs
    Set pRasterVal.RasterStorageDef = pRasStoreDef

    Set createRasterValue = pRasterVal

    '++ cleanup
    Set pRasterVal = Nothing
    Set pRasStoreDef = Nothing
    Exit Function
er:
    MsgBox Err.Description, , "createRasterValue"

End Function

```

## Scripting Sample

### *Geoprocessing Scripting Sample*

The geoprocessing framework provides a handful of tools for loading raster data to the geodatabase. For batch loading practices, scripting with geoprocessing tools is more desirable. There are some samples published online. You can also access some sample tools in the Samples toolbox in ArcGIS that is available out of the box.

#### **Example 1**

**Load raster datasets in a directory to a seamless raster dataset in the geodatabase.**

```
import win32com.client, os
gp = win32com.client.Dispatch\
    ("esriGeoprocessing.GpDispatch.1")
workspace = "c:/rasterdata"

#The output raster dataset
out_raster = "database connections/connection to \
    raster.sde/raster.redlands"

gp.workspace = workspace

#The raster datasets in the input workspace
in_raster_datasets = gp.ListRasters()

in_raster_dataset = in_raster_datasets.next()
in_rasters = in_raster_dataset

while in_raster_dataset <> "":
    in_raster_dataset = in_raster_datasets.next()
    in_rasters = in_rasters + ";" + in_raster_dataset

gp.Mosaic_management(in_rasters, out_raster)
```

#### **Example 2**

**Batch Inserting Rasters from a File Directory to a Raster Catalog in the Database  
Using Default Storage Parameters**

```
import win32com.client, os
gp = win32com.client.Dispatch\
    ("esriGeoprocessing.GpDispatch.1")
gp.workspace = "c:/rasterdata"
out_catalog = "database connections/connect to \
    raster.sde/raster.redlands_catalog"

in_raster_datasets = gp.ListRasters()

in_raster_dataset = in_raster_datasets.next()
```

```
while in_raster_dataset <> "":  
    gp.CopyRaster_management(in_raster_dataset,  
        out_catalog)  
    in_raster_dataset = in_raster_datasets.next()
```

## Tips and Tricks

### *Creating Data*

When creating a geodatabase raster dataset in ArcGIS, the storage parameters set in the ArcToolbox Environment Settings are ignored. You need to set them on the Create Raster Dataset dialog box instead.

It is necessary to examine all the data before creating a raster catalog and loading the data to it, especially the spatial extent of all the rasters. A proper spatial domain on the footprint column of a raster catalog is critical for ensuring all the data will be loaded properly. If any data to be loaded is outside the spatial domain, it will not be loaded. Like a feature class, once the spatial domain is set on the footprint column, it cannot be modified without reloading the data. In most cases, if the input rasters have a spatial reference, it is much easier to set the spatial reference of the footprint column as Geographic Coordinate System (GCS) and set the proper spatial domain because the spatial domain of GCS is always within -180 to 180 in x direction and -90 to 90 in y direction.

### *Loading Data*

To improve performance for loading rasters into a geodatabase, it is recommended that pyramids be built while loading. But you have to be sure that the pyramid reference point is set properly for the case of mosaicking. Because building pyramids is computationally intensive, always use the machine(s) with the fastest processor to build them; in some cases, this may mean using direct connect from a client machine.

Tests conducted suggest that using separate loading processes to load rasters into different rows (images) in the same raster catalog can speed up the loading of rasters tremendously. As performance will vary depending on the processor, it is recommended that the load process be tested on a small amount of data. However, it is not possible to have multiple processes loading into the same raster dataset simultaneously yet.

Refer to the *Managing a Raster Database* technical paper for details and case studies on loading large amounts of raster data to ArcSDE.

### *Loading Projected Data*

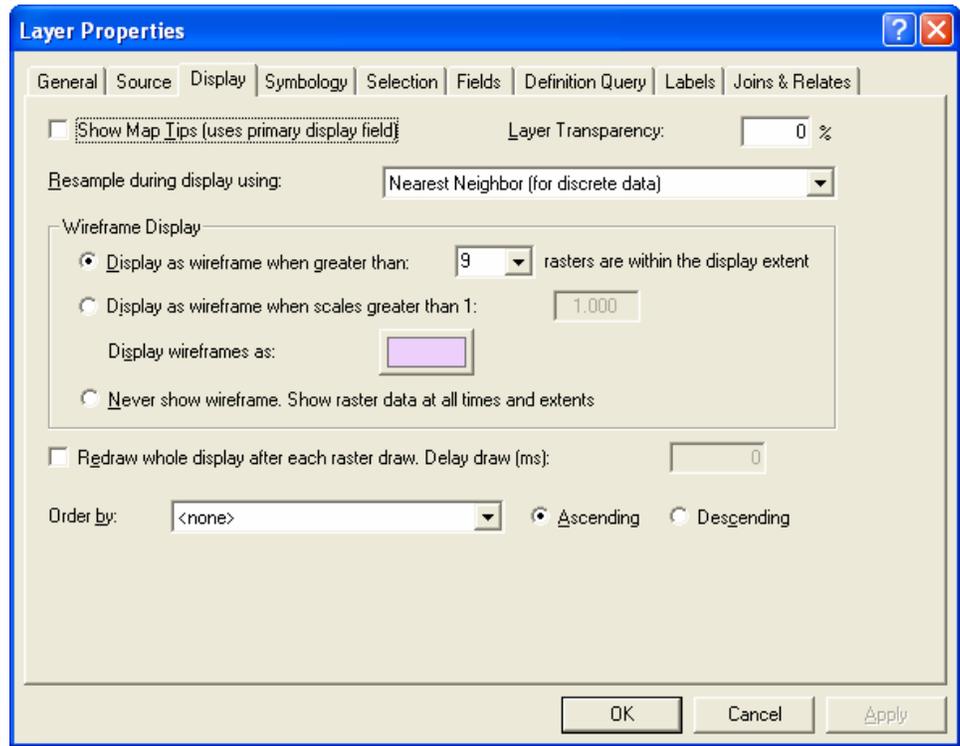
When a file raster dataset is projected, in some cases the pixel type is changed to accommodate the nodata area generated by the projection. The pixel depth is promoted, for example, from 8 bit to 16 bit. The option is available for changing the pixel type back to its original when loading the projected and pixel depth-promoted dataset to ArcSDE without losing any pixel information. In ArcGIS 9.1, the CopyRaster tool adds an option to specify pixel type for the output raster dataset.

### *Updating Raster Catalog Footprints*

If for any reason the footprint column of a raster catalog is empty or out of date, you can run Update Footprints in ArcCatalog to update the footprints of all or selected raster datasets in the raster catalog with their extents.

Storage	Each raster tile is stored as a BLOB in the raster block table to minimize storage requirements when choosing a tile size. Consider the current DBMS page size to make sure that a good proportion (75 percent) of the tiles will be stored in line with the rest of the row data.
Loading Rasters in MrSID Format	Because of the nature of wavelet compression and the way data blocks are transferred to ArcSDE, the compressed data tiles have to be decoded multiple times before loading, and this inevitably affects the loading performance. It is recommended that you convert MrSID to IMG first without building pyramids, since pyramids cannot be carried over and must be rebuilt, then load the IMG file to ArcSDE. Tests suggest that it is faster than loading MrSID directly. This issue will be resolved in future releases.
Mosaicking Rasters with Colormap	With the new geoprocessing framework, it is much easier to mosaic rasters with colormaps. You can choose one of the colormap modes on the Mosaic tool dialog box so the colormap of the mosaic raster will be constructed in a certain way if the colormaps of the input raster datasets are different.
Nodata	If the input raster dataset has nodata, a bit mask will be generated on the fly at the time of loading to the database. The bit mask is stored along with pixel data on each band. You can also specify a value to be converted to nodata when loading/mosaicking rasters.
Viewing in ArcGIS	To improve the viewing performance of raster catalogs, set the layer properties in ArcMap to show a wire frame instead of the data when zoomed out or when many rasters will be visible within the display extent.

**Figure 19**  
**Layer Property Setting**



The default resampling method for viewing raster layers in ArcGIS is nearest neighbor, which does not affect the method specified when loading the data. This is the fastest method of resampling when viewing the data.

**Mosaic versus Raster Catalog**

Mosaicked rasters perform better than a raster catalog with the same set of rasters. Because a mosaicked raster is one entity, it eliminates the possibility of color mismatching that could happen in a raster catalog. However, it is easier for a single raster in a raster catalog to be accessed or updated, and raster catalogs can have extra fields appended to their business table to store more information about each raster dataset.

**Attribute Table**

The current server implementation of raster in ArcSDE does not support the concept of raster attribute tables, and once the raster data is loaded to ArcSDE, its attribute table is not carried over. This is being worked on and will be available in future releases.

**Raster Catalog Does Not Display**

If the raster catalog is loaded without any error and the footprint column is populated, but somehow the raster catalog will not display, the spatial grid size for the footprint column may be too small. The best practices for setting a proper grid size for a feature class should be followed by considering the footprint is the same as the shape column in a feature class. Normally, it is appropriate to get the average footprint size and multiply it by 3. If the footprints are not uniform and vary a lot, you may need to consider using more than one spatial grid size.

## Appendix A—Bibliography

Related topics can also be found in the documentation that is installed with ArcSDE and hard-copy documentation with ArcSDE and ArcGIS.

### Raster Concepts

- *Modeling Our World*, Michael Zeiler, ESRI, 1999. "Cell-Based Modeling with Rasters," Chapter 9.
- *Understanding ArcSDE*, Understanding\_ArcSDE.pdf, ESRI, 2004. "Raster Data in a Geodatabase," page 25.

### Raster Data Storage

- *ArcSDE Configuration and Tuning Guide for DB2*, ArcSDE\_Config\_Gd\_DB2.pdf, ESRI, 2004. "Appendix A, Storing Raster Data," pages 80–87.
- *ArcSDE Configuration and Tuning Guide for Informix*, ArcSDE\_Config\_Gd\_Informix.pdf, ESRI, 2004. "Appendix B, Storing Raster Data," pages 102–112.
- *ArcSDE Configuration and Tuning Guide for Oracle*, ArcSDE\_config\_Gd\_Oracle.pdf, ESRI, 2004. "Appendix B, Storing Raster Data," pages 125–135.
- *ArcSDE Configuration and Tuning Guide for SQL Server*, ArcSDE\_Config\_Gd\_SQLServer.pdf, ESRI, 2004. "Appendix C, Storing Raster Data," pages 107–116.
- *Managing ArcSDE Application Servers*, Mnging\_ArcSDE\_Applictn\_Svrs.pdf, ESRI, 2004. "Appendix B, ArcSDE Table Definitions, Raster Tables," pages 109–110.
- *Understanding ArcSDE*, ESRI, 2004. "Raster Data Storage," page 17, Understanding\_ArcSDE.pdf.

### Performance and Tuning

- *ArcSDE Configuration and Tuning Guide for Oracle*, ESRI, 2004. "Appendix A, Estimating the Size of Your Tables and Indexes: The Raster Data Tables," pages 121–123, ArcSDE\_config\_Gd\_Oracle.pdf.

### ArcSDE Management

- *ArcSDE 9 Developer Help*, ESRI. "Administrator Command References," Concepts chapter.
- *Managing ArcSDE Services*, "Raster Parameters," page 35 (.pdf page 41), Managing\_ArcSDE\_Services.pdf.

**Loading Data**

- *Building a Raster Database*, ESRI, 2004. "Building a Raster Geodatabase," pages 327–344.
- *Managing a Raster Database*, ESRI White Paper, ESRI, 2005. <http://support.esri.com/index.cfm?fa=knowledgebase.whitepapers.viewPaper&PID=17&MetaID=1018>

**Upgrading Data**

- *Upgrading Geodatabase Raster Data from ArcGIS 8.3 to ArcGIS 9*, ESRI Technical Paper, 2004. <http://support.esri.com/index.cfm?fa=knowledgebase.whitepapers.viewPaper&PID=19&MetaID=884>

## Appendix B—Raster Table Schema

The text that follows describes the schema of the tables associated with the storage of raster data. It can also be found online: *ArcSDE 9.1 Developer Help, Development Environments-> C API >Programming Concepts->Raster Data Management.*

### RASTER\_COLUMNS Table

When the user adds a raster column to a business table, ArcSDE adds a record to the RASTER\_COLUMNS system table maintained in the SDE user's schema. ArcSDE also creates four tables to store the raster images and metadata associated with each one.

**Table B-1**  
**RASTER\_COLUMNS Table**

<i>NAME</i>	<i>DATA TYPE</i>	<i>NULL?</i>
rastercolumn_id	SE_INTEGER_TYPE	NOT NULL
description	SE_STRING_TYPE(65)	NULL
database_name	SE_STRING_TYPE(32)	NULL
owner	SE_STRING_TYPE(32)	NOT NULL
table_name	SE_STRING_TYPE(160)	NOT NULL
raster_column	SE_STRING_TYPE(32)	NOT NULL
cdate	SE_DATE_TYPE	NOT NULL
config_keyword	SE_STRING_TYPE(32)	NULL
minimum_id	SE_INTEGER_TYPE	NULL
base_rastercolumn_id	SE_INTEGER_TYPE	NOT NULL
rastercolumn_mask	SE_INTEGER_TYPE	NOT NULL
srid	SE_INTEGER_TYPE	NULL

- **rastercolumn\_id**—The table's primary key. The value is also used in naming the tables of the raster data store. For example, if the rastercolumn\_id is 10 the raster table is created with the name sde\_ras\_10, the band table is created with the name sde\_bnd\_10, and so on.
- **description**—The optional general description of the rasters stored in the raster column.
- **database\_name**—The database that stores the business table and the tables of the raster data store. For those DBMSs that do not support multiple databases within a single server instance, this value will always be NULL.
- **owner**—The owner of the raster column's business table.
- **table\_name**—The business table name the raster column has been added to.

- **raster\_column**—The raster column name.
- **cdate**—The date the raster column was added to the business table.
- **config\_keyword**—The DBTUNE configuration keyword whose storage parameters determine how the tables and indexes of the raster are stored in the database. For more information on DBTUNE configuration keywords and their storage parameters, review Chapter 3, Configuring DBTUNE storage parameters, in the *ArcSDE Configuration and Tuning Guide* for your DBMS. The ArcSDE default value is *defaults*.
- **minimum\_id**—Defined during the creation of the raster, it establishes the initial value of the raster table's raster\_id column. Defaults to 1.
- **base\_rastercolumn\_id**—If a view of the business table is created that includes a raster column, an entry is added to the RASTER\_COLUMNS table that has its own rastercolumn\_id. The rastercolumn\_id of the view's base business table is inserted into the base\_rastercolumn\_id. This base\_rastercolumn\_id maintains referential integrity to the base business table. It ensures that actions performed on the business table are reflected in the view. For example, if the business table's raster column is dropped, it will also be dropped from the view (i.e., the column is dropped from both the base business table and the view), and the RASTER\_COLUMN entries are removed from both the business table and view.
- **rastercolumn\_mask**—Stores the internal properties of the raster column. For example, the geodatabase distinguishes a raster column as a raster dataset or a raster catalog by setting the rastercolumn\_mask.
- **srid**—The coordinate reference system the raster was created with. The client application uses it to project the raster.

**Business Table**

In the example that follows, the fictitious REDLANDS business table contains the raster column image. This is a foreign key reference to the raster table created in the user's schema. In this case, the raster table contains a record for a Thematic Mapper image of Redlands.

**Table B-2  
Business Table**

<i>NAME</i>	<i>DATA TYPE</i>	<i>NULL?</i>
Name	SE_INTEGER_TYPE	NOT NULL
raster	SE_INTEGER_TYPE	NOT NULL

- **Name**—The table's primary key
- **Image**—A raster column and foreign key reference to a raster table containing the raster

**Raster Table  
(SDE\_RAS\_  
<rastercolumn\_id>)**

The raster table, created as SDE\_RAS\_<rastercolumn\_id> in the database, stores a record for each image stored in a raster column. The rastercolumn\_id is assigned by ArcSDE whenever a raster column is created in the database. A record for each raster column in the database is stored in the ArcSDE RASTER\_COLUMNS system table maintained in the SDE user's schema.

**Table B-3  
Raster Metadata Table**

<i>NAME</i>	<i>DATA TYPE</i>	<i>NULL?</i>
raster_id	SE_INTEGER_TYPE	NOT NULL
raster_flags	SE_INTEGER_TYPE	NULL
description	SE_STRING_TYPE(65)	NULL

- **raster\_id**—The primary key of the raster table and unique sequential identifier of each image stored in the raster table
- **raster\_flags**—A bitmap set according to the characteristics of stored image
- **description**—A text description of the image

**Raster Band Table  
(SDE\_BND\_  
<rastercolumn\_id>)**

Each image referenced in a raster may be subdivided into one or more raster bands. The raster band table, created as SDE\_BND\_<rastercolumn\_id>, stores the raster bands of each image stored in the raster table. The raster\_id column of the raster band table is a foreign key reference to the raster table's raster\_id primary key. The rasterband\_id column is the raster band table's primary key. Each raster band in the table is uniquely identified by the sequential rasterband\_id.

**Table B-4  
Raster Band Metadata Table**

<i>NAME</i>	<i>DATA TYPE</i>	<i>NULL?</i>
rasterband_id	SE_INTEGER_TYPE	NOT NULL
sequence_nbr	SE_INTEGER_TYPE	NOT NULL
raster_id	SE_INTEGER_TYPE	NOT NULL
name	SE_STRING_TYPE(65)	NULL
band_flags	SE_INTEGER_TYPE	NOT NULL
band_width	SE_INTEGER_TYPE	NOT NULL
band_height	SE_INTEGER_TYPE	NOT NULL
band_types	SE_INTEGER_TYPE	NOT NULL
block_width	SE_INTEGER_TYPE	NOT NULL
block_height	SE_INTEGER_TYPE	NOT NULL
block_origin_x	SE_FLOAT_TYPE	NOT NULL
block_origin_y	SE_FLOAT_TYPE	NOT NULL
eminx	SE_FLOAT_TYPE	NOT NULL
eminy	SE_FLOAT_TYPE	NOT NULL
emaxx	SE_FLOAT_TYPE	NOT NULL
emaxy	SE_FLOAT_TYPE	NOT NULL
cdate	SE_DATE_TYPE	NOT NULL
mdate	SE_DATE_TYPE	NOT NULL

- **rasterband\_id**—The primary key of the raster band table that uniquely identifies each raster band.
- **sequence\_nbr**—A sequential number combined with the raster\_id to form a composite key as the second way to uniquely identify the raster band.
- **raster\_id**—The foreign key reference to the raster table's primary key. Uniquely identifies the raster band when combined with the sequence\_nbr as a composite key.
- **name**—The name of the raster band.
- **band\_flags**—A bit mask set according to the characteristics of the raster band. The band properties stored by the band\_flags bit mask include
  - Maximum pyramid level
  - The pyramid interpolation type
  - Whether or not the band has a colormap
  - Whether or not image statistics are present
  - Whether or not a coordinate transformation exists
- **band\_width**—The width of the band specified as a number of pixels.
- **band\_height**—The height of the band specified as a number of pixels.
- **band\_types**—A bit mask set according to the characteristics of the raster band. The band properties stored by the band\_types bit mask include
  - Compression type
  - Pixel depth of the raster data
- **block\_width**—The pixel width of the band specified as a number of tiles.
- **block\_height**—The pixel height of the band specified as a number of tiles.
- **cdate**—The creation date.
- **mdate**—The last modification date.

If the raster has a map extent, the optional **eminx**, **eminy**, **emaxx**, and **emaxy** will hold the coordinates of the extent. If the image does not have an extent, the **eminx**, **eminy**, **emaxx**, and **emaxy** are inverted (the **eminx** and **eminy** contain largest possible value, and the **emaxx** and **emaxy** contain the smallest possible value).

- **block\_origin\_x**—The geographic coordinate of the left-most pixel
- **block\_origin\_y**—The geographic coordinate of the upper-most pixel
- **eminx**—The band's minimum X ordinate
- **eminy**—The band's minimum Y ordinate
- **emaxx**—The band's maximum X ordinate

- **emaxy**—The band's maximum Y ordinate

**Raster Blocks Table**  
(**SDE\_BLK\_< rastercolumn\_id >**)

Created as SDE\_BLK\_< rastercolumn\_id >, the raster blocks table stores the actual pixel data of the raster images. ArcSDE evenly tiles the bands into blocks of pixels. Tiling the raster band data enables efficient storage and retrieval of the raster data. The raster blocks can be configured so that the records of the raster block table fit with an Oracle data block, avoiding the adverse effects of data block chaining.

The rasterband\_id column of the raster block table is a foreign key reference to the raster band table's primary key. A composite unique key is formed by combining the rasterband\_id, rrd\_factor, row\_nbr, and col\_nbr columns.

**Table B-5**  
**Raster Blocks Table**

<i>NAME</i>	<i>DATA TYPE</i>	<i>NULL?</i>
rasterband_id	SE_INTEGER_TYPE	NOT NULL
Rrd_factor	SE_INTEGER_TYPE	NOT NULL
Row_nbr	SE_INTEGER_TYPE	NOT NULL
col_nbr	SE_INTEGER_TYPE	NOT NULL
block_data	SE_BLOB_TYPE	NOT NULL

- **rasterband\_id**—The foreign key reference to the raster band table's primary key.
- **rrd\_factor**—The reduced resolution dataset factor determines the position of the raster band block within the resolution pyramid. The resolution pyramid begins at 0 for the highest resolution and increases until the raster band's lowest resolution level has been reached.
- **row\_nbr**—The block's row number.
- **col\_nbr**—The block's column number.
- **block\_data**—The block's tile of pixel data.

**Raster Band Auxiliary Table**  
(**SDE\_AUX\_< rastercolumn\_id >**)

The raster band auxiliary table, created as SDE\_AUX\_< rastercolumn\_id >, stores optional raster metadata such as the image colormap, image statistics, and transformation. The rasterband\_id column is a foreign key reference to the primary key of the raster band table.

**Table B-6**  
**Raster Auxiliary Table**

<i>NAME</i>	<i>DATA TYPE</i>	<i>NULL?</i>
rasterband_id	SE_INTEGER_TYPE	NOT NULL
type	SE_INTEGER_TYPE	NOT NULL
object	SE_BLOB_TYPE	NOT NULL

- **rasterband\_id**—The foreign key reference to the raster band table's primary key.
- **type**—An enumerator set according to the characteristics of the data stored in the object column. The column can contain the following enumerations:
  - SE\_RASTERBAND\_OBJECT\_TYPE\_STATS 2—The object column contains optional image statistics.
  - SE\_RASTERBAND\_OBJECT\_TYPE\_COLORMAP 3—The object column contains the optional image colormap.
  - SE\_RASTERBAND\_OBJECT\_TYPE\_XFORM 4—The object column contains the optional coordinate transformation list.