Oracle Database 10*g*: Managing Spatial Raster Data Using GeoRaster

An Oracle Technical White Paper
May 2005

ORACLE

Table of Contents

1 GEORASTER

1.1 Introduction

GeoRaster is a feature of Oracle Spatial in Oracle Database 10*g* that lets you store, index, query, analyze, and deliver GeoRaster data, that is, image and gridded raster data and its associated metadata. GeoRaster provides Oracle Spatial data types and an object-relational schema. You can use these data types and schema objects to store multidimensional gridded data and raster layers that can be referenced to positions on the Earth's surface or in a local coordinate system. Each image or a multi-layer grid is stored as a GeoRaster object in a GeoRaster column of any table. If the data is georeferenced, you can find the location on Earth for a cell in an raster; or given a location on Earth, you can find the cell in a raster layer associated with that location.

GeoRaster is designed to deliver enterprise-class data management capability to large image processing and GIS solutions. It is now possible for developers to integrate this powerful data management technology with the leading image processing and raster/grid analysis tools.

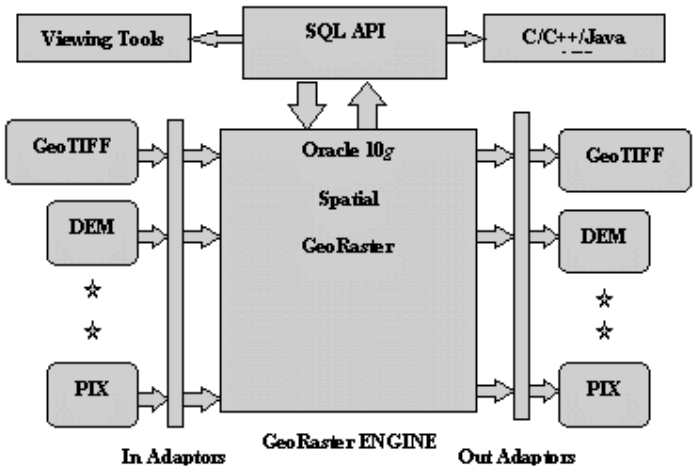
1.2 Architecture

The GeoRaster architecture provides the core functionality needed to support the use of image or grid-based raster data in Oracle Database 10*g*. At a very high level of abstraction, the GeoRaster architecture includes six basic components:

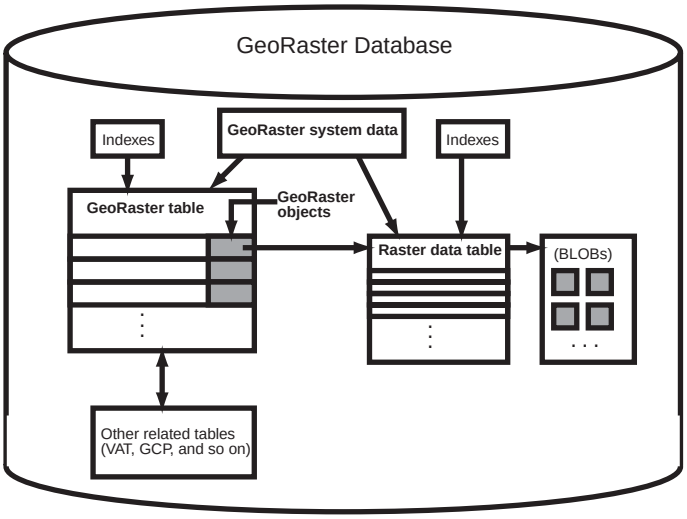
1. GeoRaster Engine – Provides the GeoRaster object type and core GeoRaster functionality including raster data and metadata operations and indexing.
2. SQL API – SQL access to the raster and grid-based data in GeoRaster.
3. C/C++/Java – OCI, OCCI, and Java access to the raster and grid-based data in GeoRaster with or without calling the GeoRaster API
4. Viewing Tools: A variety of third party viewing and analysis tools now support GeoRaster. Oracle AS MapViewer supports GeoRaster. In addition, a free downloadable viewer is available from Oracle.

5. Input [data] adapters – Facilitate loading raster data from well-known file formats into GeoRaster.

6. Output [data] adapters – Facilitate unloading raster data into well-known file formats from GeoRaster.



The core of Oracle GeoRaster is the physical architecture for storing and managing raster or grid-based data inside the database. In the GeoRaster engine, the native data type called SDO_GEORASTER is defined and each image or raster grid is stored as a single object of this native type. A GeoRaster data table is any user-defined table, which has at least one data column of type SDO_GEORASTER. SDO_GEORASTER objects include information about how to retrieve GeoRaster cell data that is stored in a raster data table, otherwise known as an object table of type SDO_RASTER. The SDO_RASTER type includes a BLOB column called RASTERBLOCK, which stores the real raster blocks. Other information associated with the GeoRaster objects can be stored in separate columns or tables, such as a Value Attribute Table (VAT). GeoRaster itself, using a metadata table, automatically manages the relationship between a GeoRaster object and its raster data table. From a user perspective, a GeoRaster database is basically a list of tables, in which each image or raster grid is stored as one GeoRaster object in one row. It can contain unlimited number of GeoRaster objects and each object can be terabytes in size.



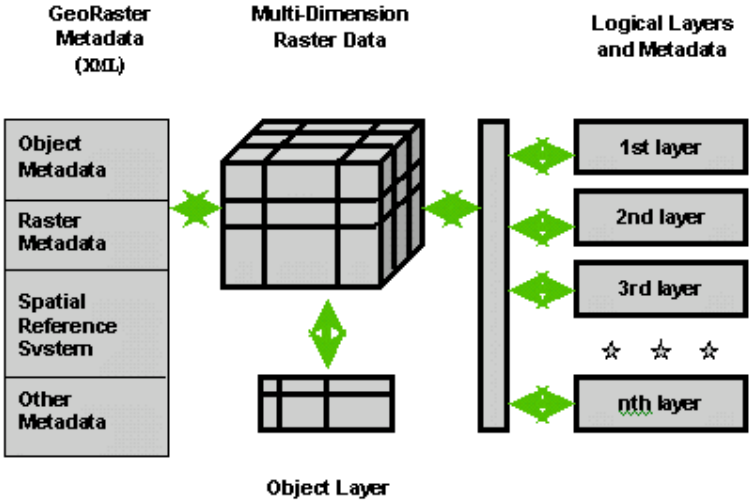
The specifics of the GeoRaster data model and how this architecture is implemented in Oracle Database 10*g* are provided in the sections below.

1.3 Data Model

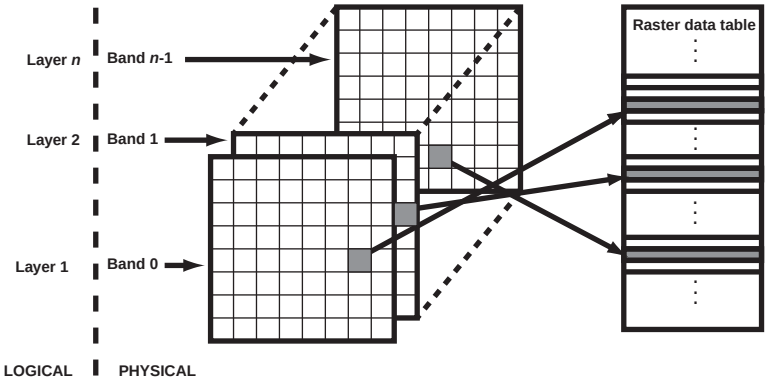
There are two basic kinds of raster data types supported in GeoRaster: grid-based data and image data.

- Grid-based data or gridded data is a general term used for raster data. It's a rectangular grid of cells that are aligned to the X and Y-axes overlying an area. Each cell in the grid has the same size; this size is the resolution of the grid. Grid data typically stores attribute values for each cell in the grid. Examples of attributes stored for each cell include digital terrain elevation, land use information, pollution concentration, land cover information, geological information, rainfall information, and many others.
- Digital imagery is a specialized type of raster data. It is a two dimensional array (a matrix or grid) of regularly spaced picture elements (pixels). An image is created from optical or other sensor data, and is collected using a variety of technologies including satellite remote sensing, airborne photogrammetry, and medical imaging devices. The size of the pixel is referred to as the resolution of the image. Digital images can be composed of many bands, referred to as multispectral or hyperspectral.

GeoRaster uses a generic raster data model for these data types. Conceptually, it is component-based, logically layered, and multidimensional. The core data in a raster is a multidimensional matrix of raster cells. Each cell is one element of the matrix, and its value is called the cell value. The matrix has a number of dimensions, a cell depth, and a size for each dimension. The cell depth is the data size of the value of each cell. It also defines the range of all cell values. The cell depth applies to each single cell, not an array of cells. This core raster data set can be blocked for optimal storage, retrieval and processing.



The raster data is logically layered. The core data is called the object layer or layer 0, and consists of one or more logical layers (or sublayers). For example, for multi-channel remote sensing imagery, the layers are used to model the channels or bands of the imagery. In GeoRaster, each layer is a two-dimensional matrix of cells that consists of the row dimension and the column dimension. The relationship between logical layers in the GeoRaster data model and the physical bands or channels of the source image data is depicted as follows.

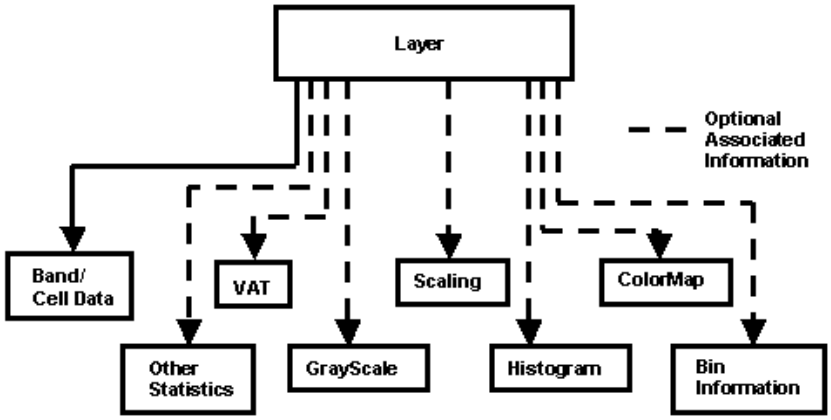


A GeoRaster object has specific metadata associated with it. Each layer of the GeoRaster data can have its own metadata and attributes. In the GeoRaster data model, metadata is everything but the core cell matrix. The GeoRaster metadata is further divided into different components that contain the following information:

- Object information
- Raster information
- Spatial reference system information
- Date and time (temporal reference system) information
- Spectral (band reference system) information

- Layer information for each layer

The Object Information includes metadata such as description and version information. The Raster Information includes metadata such as cell depth (1BIT, 32BIT_S, or 64BIT_REAL), dimensionality, blocking, interleaving, compression and information about pyramids. The Spatial Reference System metadata contains information required for georeferencing, that is the coefficients of affine transformation. The Layer Information contains metadata pertaining to each layer in a GeoRaster object. If the data is grid data, one or more Value Attribute Tables (VAT) can be used to maintain information about the values stored in each layer (e.g. elevation value, saturation level etc.). In addition, there is a comprehensive suite of metadata used to capture and track image/cell attribution, scaling factors, color related information (colormap, grayscale), histogram and other layer-based attributes essential for image management and used by client applications.



1.4 GeoRaster Object

Physically, the GeoRaster data model is embodied as two native data types and an object-relational schema inside Oracle ORDBMS.

At the top level, one raster data (an image or a grid) is stored in Oracle as an object of SDO_GEORASTER data type, which is defined as:

```
CREATE TYPE sdo_georaster AS OBJECT (
    rasterType NUMBER,
    spatialExtent SDO_GEOMETRY,
    rasterDataTable VARCHAR2(32),
    rasterID NUMBER,
    metadata XMLType);
```

Most metadata is stored as one attribute of the SDO_GEORASTER type. It is an XML document using the Oracle XMLType data type. The metadata is stored

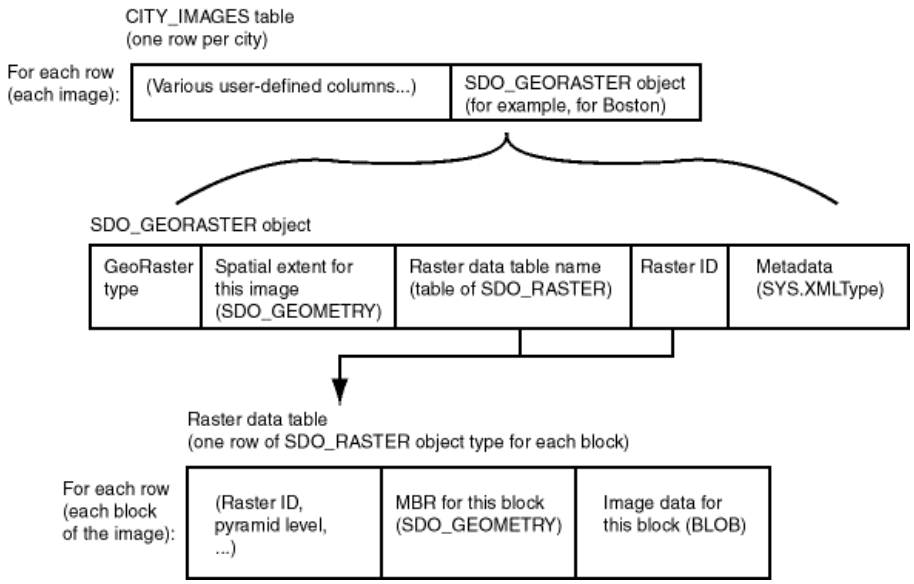
according to the GeoRaster metadata XML schema defined by GeoRaster. The spatial extent (footprint) of a GeoRaster object is part of the metadata, but it is stored separately as an attribute of the GeoRaster object. This approach allows GeoRaster to take advantage of the spatial geometry type and related capabilities, such as using spatial R-tree indexing on GeoRaster objects and building huge global imagery databases. Another attribute of the SDO_GEORASTER type is the rasterType, which contains dimensionality information and the data type that can be further defined.

The actual raster cell data is blocked into small subsets for large-scale GeoRaster object storage and optimal retrieval and processing. All blocks are stored in an object table of type SDO_RASTER, which is defined as follows:

```
CREATE TYPE sdo_raster AS OBJECT (
    rasterID NUMBER,
    pyramidLevel NUMBER,
    bandBlockNumber NUMBER,
    rowBlockNumber NUMBER,
    columnBlockNumber NUMBER,
    blockMBR SDO_GEOMETRY,
    rasterBlock BLOB);
```

This object table is called raster data table, or simply RDT table. Each block is stored in the RDT table as a binary large object (BLOB), and a geometry object (of type SDO_GEOMETRY) is used to define the precise extent of the block. Each row of the table stores only one block and the blocking information related to that block. This same blocking scheme is used to support pyramids as well and the pyramids are stored in the same RDT table.

The following figure shows the storage of GeoRaster objects, using as an example an image of Boston, Massachusetts in a table that contains rows with images of various cities.



As shown in the preceding figure, GeoRaster tables, created by a user, can contain attribute data (just like any other Oracle table). However, a GeoRaster table will also contain a column of object type SDO_GEORASTER. The SDO_GEORASTER object type has several attributes, two of which are needed to store and select cell data from the SDO_GEORASTER object: the rasterDataTable and the rasterID attributes provide the information required to store and retrieve the raster cell data.

Internally, GeoRaster uses a system data table (called the GeoRaster sysdata table) to maintain the relationship between GeoRaster objects and their related raster data tables. Even though stored in a separate RDT table, the raster cell data of a GeoRaster object is handled automatically by the GeoRaster functions, which are described in the next section.

2 FEATURES OF GEORASTER

GeoRaster provides a rich set of foundation functions in addition to providing both a logical model and a physical model that facilitate data management for raster information in the Oracle Database 10*g*. This section provides a general overview of the basic functional infrastructure available in GeoRaster.

GeoRaster functions can be grouped into the following categories:

- Database Administration (Create table, Load and Export, Index, etc.)
- Data Manipulation (Format Change, Subset, Compression, etc.)
- Cell Data and Metadata Update and Query

2.1 Database Administration

Database Administration includes functions performed to instantiate the GeoRaster environment in a database. After the raster data is loaded into GeoRaster objects,

multiple indices can be created on the GeoRaster tables, including spatial R-Tree index, B-Tree index, and function-based index on the GeoRaster metadata items. Key functions targeting database administration for GeoRaster include:

- SDO_GEOR.init - Initializes an empty GeoRaster object, which will be registered by GeoRaster in the xxx_SDO_GEOR_SYSDATA views.
- SDO_GEOR.createBlank - Creates a blank GeoRaster object, in which all cells have the same value.
- SDO_GEOR.copy - Makes a copy of an existing GeoRaster object.
- SDO_GEOR.importFrom - Imports an image or gridded raster data in a file or BLOB object into a GeoRaster object stored in the database.
- SDO_GEOR.exportTo - Exports a GeoRaster object or a subset of a GeoRaster object to a file or to a BLOB object.
- SDO_GEOR.validateGeoraster - Validates a GeoRaster object.
- SDO_GEOR.schemaValidate - Validates a GeoRaster object's metadata against the GeoRaster XML schema.

2.2 Data Manipulation

Data manipulation highlights operations to optimally manage GeoRaster data in support of various application requirements. Important concepts include:

Georeferencing - GeoRaster currently supports six-parameter affine transformation that georeferences two-dimensional raster data. The affine transformation is a special type of the Functional Fitting polynomial model. If an affine transformation is provided and is valid in the metadata, the GeoRaster object is considered georeferenced.

Pyramiding - Pyramid levels represent reduced or increased resolution of a raster object that require less or more storage space, respectively. (GeoRaster currently supports only reduced resolution pyramids.) A pyramid level of 0 indicates the original raster data. In other words, there is no reduction in the resolution and no change in the storage space required. Values greater than 0 (zero) indicate increasingly reduced levels of resolution and reduced storage space requirements. The pyramids are stored persistently with the GeoRaster objects. Users can generate or remove pyramid data.

Change Format - Under the GeoRaster architecture, all raster cell data is stored in the Raster Data Table. How the cell data is physically organized in the Raster Data Table depends on cell depth, blocking sizes and interleaving types. These parameters can be flexibly changed for any existing GeoRaster object. The blocking size can be adjusted internally. GeoRaster supports cell depth from 1-bit to 32-bit integers and 32-bit or 64-bit real data. GeoRaster supports BSQ, BIL and BIP interleaving types. Application developers can use these parameter functions as a tuning device for physically adjusting the GeoRaster object to addresses various

application requirements. This eliminates the need for off-line re-adjustment and loading of the original raster data, or readjusting settings of existing application software.

Compression and decompression (in 10gR2) – Starting with 10g Release 2, GeoRaster supports compression to reduce the storage requirements for raster objects. In 10gR2 two standard types of compression, JPEG and DEFLATE, are natively supported and others are supported through third-party plugins. JPEG is lossy while DEFLATE is lossless. All GeoRaster functions that can be performed on uncompressed (decompressed) GeoRaster objects can be performed directly on compressed objects.

Subsetting – Performs either or both of the following operations: (1) spatial crop, cut, clip, or (2) layer or band subset or duplicate.

Scaling – Enlarges or reduces a GeoRaster object while having a choice of resampling methods (nearest neighbor, bilinear interpolation, cubic convolution, average) using four or six neighboring cells.

The following is a list of key functions provided by GeoRaster:

- SDO_GEOR.changeFormatCopy - Changes the storage format of an existing GeoRaster object, including changes of blocking, cell depth and/or interleaving and makes a copy. The compression and decompression process is also embedded into this function.
- SDO_GEOR.generateSpatialExtent - Generates a Spatial geometry that contains the spatial extent of the GeoRaster object.
- SDO_GEOR.georeference - Georeferences a GeoRaster object using specified cell-to-model transformation coefficients.
- SDO_GEOR.generatePyramid - Generates pyramid data for a GeoRaster object and store it with the object.
- SDO_GEOR.deletePyramid - Deletes pyramid data for a GeoRaster object.
- SDO_GEOR.subset - Performs either (or both) of the following operations: (a) spatial crop, cut, or clip, or (b) layer or band subset.
- SDO_GEOR.scaleCopy - Scales (enlarges or reduces) a GeoRaster object and inserts the result into a new object that reflects the scaling.
- SDO_GEOR.mosaic - Mosaics seamless GeoRaster objects into one GeoRaster object.

2.3 Cell Data and Metadata Update and Query

GeoRaster cell data and metadata update and query are crucial to successful use of GeoRaster in the Oracle Database 10*g* environment. Many functions are provided for these purposes, and some key functions include:

- SDO_GEOR.getID - Returns the user-defined identifier value associated with a GeoRaster object.
- SDO_GEOR.setID - Sets a user-defined identifier to be associated with a GeoRaster object, or deletes the existing value if you specify a null id parameter.
- SDO_GEOR.getVersion - Returns the user-specified version of a GeoRaster object.
- SDO_GEOR.setVersion - Sets the user-specified version of a GeoRaster object.
- SDO_GEOR.getInterleavingType - Returns the interleaving type for a GeoRaster object.
- SDO_GEOR.getSpatialDimNumber - Returns the number of spatial dimensions of a GeoRaster object.
- SDO_GEOR.getSpatialDimSizes - Returns the number of cells in each spatial dimension of a GeoRaster object.
- SDO_GEOR.getTotalLayerNumber - Returns the total number of layers in a GeoRaster object.
- SDO_GEOR.getBlockSize - Returns the number of cells for each dimension in each block of a GeoRaster object as an array showing the number of cells for each row, column, and (if relevant) band.
- SDO_GEOR.isSpatialReferenced - Returns TRUE if the GeoRaster object is spatially referenced, or FALSE if the GeoRaster object is not spatially referenced.
- SDO_GEOR.setSpatialReferenced - Specifies whether or not a GeoRaster object is spatially referenced, or deletes the existing value if you specify a null isReferenced parameter.
- SDO_GEOR.getSRS - Returns information related to the spatial referencing of a GeoRaster object.
- SDO_GEOR.setSRS - Sets the spatial reference information of a GeoRaster object, or deletes the existing information if you specify a null srs parameter.
- SDO_GEOR.getModelSRID - Returns the coordinate system (SDO_SRID value) associated with the model (ground) space for a GeoRaster object.
- SDO_GEOR.setModelSRID - Sets the coordinate system (SDO_SRID value) for the model (ground) space for a GeoRaster object, or deletes the existing value if you specify a null srid parameter.
- SDO_GEOR.getBeginDateTime - Returns the beginning date and time for raster data collection in the metadata for a GeoRaster object.

- SDO_GEOR.setBeginDateTime - Sets the beginning date and time for raster data collection in the metadata for a GeoRaster object, or deletes the existing value if you specify a null beginTime parameter.
- SDO_GEOR.hasPseudoColor - Checks if a layer contains pseudocolor information (colormap).
- SDO_GEOR.getColorMap - Returns the colormap representing pseudocolor display of a layer.
- SDO_GEOR.setColorMap - Sets the colormap for a layer in a GeoRaster object, or deletes the existing value if you specify a null colorMap parameter.
- SDO_GEOR.getVAT - Returns the name of the value attribute table (VAT) associated with a layer.
- SDO_GEOR.setVAT - Sets the name of the value attribute table (VAT) associated with a layer of a GeoRaster object, or deletes the existing value if you specify a null vatName parameter.
- SDO_GEOR.getPyramidMaxLevel - Returns the level number of the top pyramid of a GeoRaster object.
- SDO_GEOR.getModelCoordinate - Returns the coordinates in the model (ground) coordinate system associated with the point at the specified cell (raster) coordinates.
- SDO_GEOR.getCellCoordinate - Returns the coordinates in the cell (raster) coordinate system associated with the point at the specified model (ground) coordinates.
- SDO_GEOR.getCellValue - Returns the value of a single cell located anywhere in the GeoRaster object by specifying its row, column, and band number in its cell coordinate system, or by specifying a point geometry in the model coordinate system and its logical layer number.
- SDO_GEOR.changeCellValue - Updates the value of raster data cells in a specified window and specified bands of a GeoRaster object.
- SDO_GEOR.getRasterSubset – A generic query resulting in a single BLOB object containing all cells of a specified pyramid level that are inside or on the boundary of a specified window and containing only the specified layers.
- SDO_GEOR.getRasterData - Creates a single BLOB object that contains all raster data of the input GeoRaster object at the specified pyramid level.
- SDO_GEOR.getCompressionType -Returns the compression type for a compressed GeoRaster object.
- SDO_GEOR.calcCompressionRatio - Returns the compression ratio of a specified GeoRaster object.

Many of the functions discussed above are extended, augmented or leveraged by partner technologies delivered as load/transform/export tools, comprehensive remote sensing and image processing client tools, or in the form of visualization engines built on top of the GeoRaster model.

3 SUMMARY

The introduction of GeoRaster in Oracle Database 10*g* creates significant new capacity for managing large volumes of raster data. Oracle is the only provider of commercial database management software that can store raster data and grid-based geospatial information in the database as a named type. It facilitates full integration of raster datasets with other enterprise datasets and supports better business applications. The data model allows flexible blocking of raster data so that large-scale GeoRaster objects can be stored and easily managed. By using the embedded GeoRaster API users can freely change the internal storage of any existing GeoRaster objects into different block sizes, interleaving, cell depth and compression. This can reduce the cost of building and fielding of applications. Spatial index and other indices provide fast retrieval of raster metadata and cell data. Selective query tools and other operations can be easily built upon it. It also provides good extensibility by using XML technology for describing the metadata. Further development will provide numerous useful features and advantages. It is now possible for users of file-based image processing and raster data applications to benefit from the scalability, security and performance of Oracle Database 10*g* and take advantages of GRID computing to support the mission critical applications.

Managing Spatial Raster Data Using GeoRaster: An Oracle Technical White Paper
May 2005
Authors: Jeffrey Xie and Jayant Sharma
Contributing Author:

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com