



An Oracle White Paper  
February 2011

# Oracle Spatial 11g GeoRaster

Introduction .....	1
Architecture .....	3
Data Model .....	5
GeoRaster Object.....	7
Features of GeoRaster .....	9
Database Creation.....	9
Database Administration.....	10
Data Manipulation.....	11
New Features in 11.1 .....	13
New Features in 11.2 .....	14
Benefits of Managing Raster Data in Oracle Database .....	14
Application of GeoRaster .....	15
Commercial Data Repository and Distribution .....	15
Defense and Security .....	16
Emergency Response .....	16
Enterprise Asset Management.....	17
State and Local Government .....	17
Agriculture Monitoring.....	18
Insurance Risk Assessment .....	18
Conclusion .....	19

## Introduction

GeoRaster is a feature of Oracle Spatial that lets you store, index, query, analyze, and deliver raster image and gridded data and its associated metadata. It provides data types and an object-relational schema to store multidimensional grid layers and digital images that can be referenced to positions on the Earth's surface or in a local coordinate system. If the data is georeferenced, you can find the location on Earth for a cell in an image; or given a location on Earth, you can find the cell in an image associated with that location.

GeoRaster is used with data from any technology that captures or generates raster data and images, such as remote sensing, photogrammetry, and thematic mapping. It is used in a wide variety of application areas, including location based services, geoimagery archiving, environmental monitoring and assessment, geological engineering and exploration, natural resource management, defense, emergency response, telecommunications, transportation, urban planning, and homeland security.

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

It meets the data management needs of broad application groups including:

- Remote sensing, photogrammetry, GIS and geosciences applications – users manage their geospatial raster and gridded data assets using a scaleable, secure, and robust RDBMS for defense, intelligence, agriculture, environment and natural resource management.
- Business applications – leverage raster-based data in conjunction with other basic location data (address, etc.) to evaluate site locations and track fixed and/or continuous assets.

These include Asset Management and Facilities Management particularly in energy and utilities.

- Image and Gridded Raster Data Repositories/Clearinghouses – support for clearinghouse servers that need to ingest, store, and disseminate very large volumes of geoinmagery and raster grids, through intranet or internet.

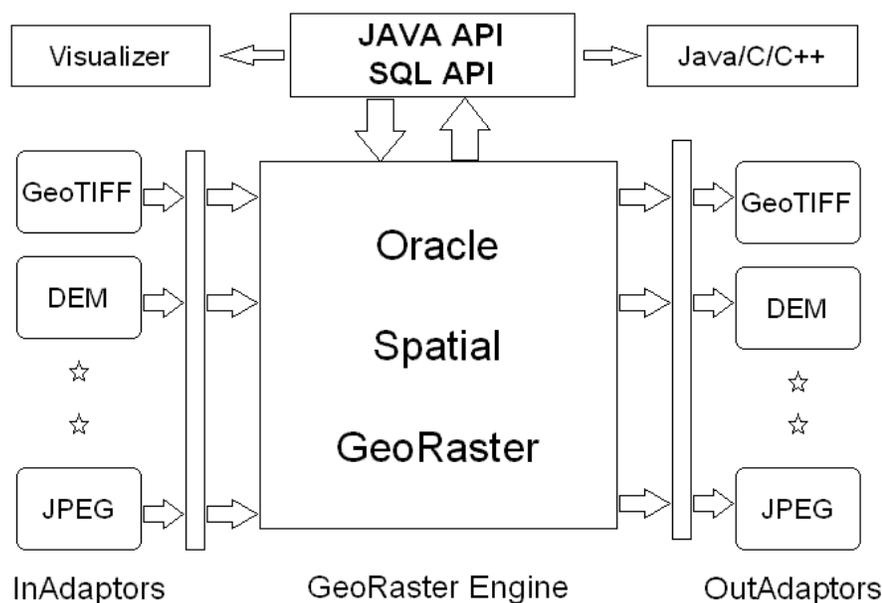
With GeoRaster, it is now possible to manage both georeferenced raster and spatial vector data in a single Oracle database. This means that the same storage, spatial referencing, indexing, and spatial operators can be used to store, query, and retrieve both raster and vector data. Moreover, all Oracle tools and utilities now support the management of raster data and related datasets in the data server. For the customer, this means that the cost of building and fielding applications that use this data will be reduced while the security, scalability and reliability of the application environment will improve dramatically.

This paper provides an introduction to the Oracle Spatial GeoRaster architecture, data model, object type, and other GeoRaster features. It explains the benefits of managing raster data in Oracle Database, and describes how applications in different vertical areas such as defense, emergency response, and insurance can use and benefit from GeoRaster.

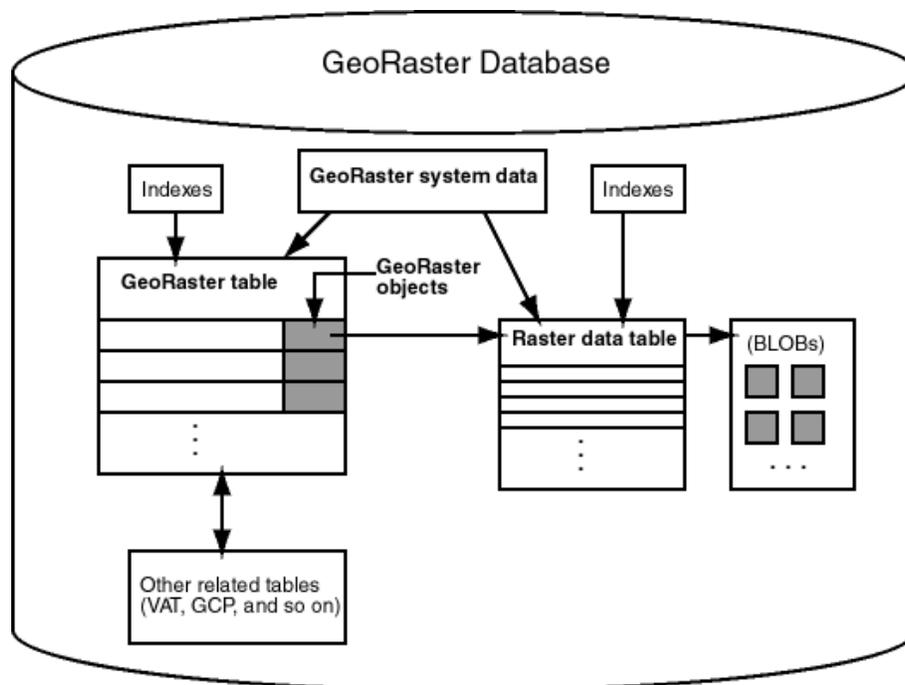
## Architecture

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

- GeoRaster Engine – This is the core, which provides the native GeoRaster object type and functionality including raster data and metadata indexing, update, query and manipulations.
- SQL API – Standard SQL access to the raster and grid-based data in GeoRaster databases.
- Java API – Pure Java access to the GeoRaster objects in the databases. Users can also use JDBC, OCI, and OCCI to access to the GeoRaster objects without calling the GeoRaster SQL or Java APIs.
- Viewing Tools: A variety of third party visualization and analysis tools now support GeoRaster. Oracle Fusion Middleware MapViewer supports GeoRaster. In addition, a standalone viewer comes with the Oracle Spatial GeoRaster installation and can be used as a development or DBA tool.
- Input and Output [data] adapters – Facilitate loading and unloading raster data between well-known image file formats and GeoRaster. A variety of third party ETL tools now support loading and unloading GeoRaster data. GeoRaster also provides limited importing and exporting capability on six standard image file formats through both the server-side SQL API and the client-side Java tool.



The core of Oracle Spatial GeoRaster is the physical schema designed to facilitate 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 table is any user-defined table, which has at least one data column of type `SDO_GEORASTER`. `SDO_GEORASTER` objects include metadata and information about how to retrieve GeoRaster cell data that is stored in another user-defined table called a Raster Data Table, which is 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). The relationship between a GeoRaster object and its raster data table is automatically managed by GeoRaster itself internally using a database dictionary. A GeoRaster database basically consists of a list of GeoRaster tables, in which each image or raster grid is stored as one GeoRaster object in one row. It can contain an unlimited number of GeoRaster objects and each object can be terabytes in size. The GeoRaster tables can be in different database schemas and the GeoRaster objects can be accessed across schemas.



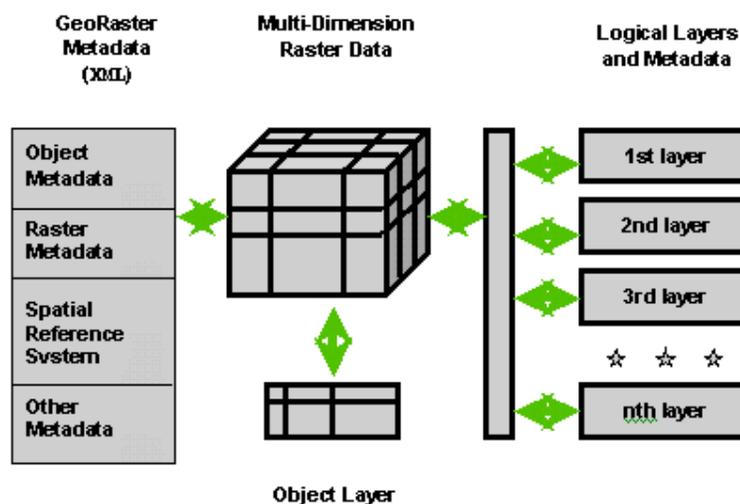
The specifics of the GeoRaster data model and how this architecture is implemented in Oracle Database are provided in the sections below.

## Data Model

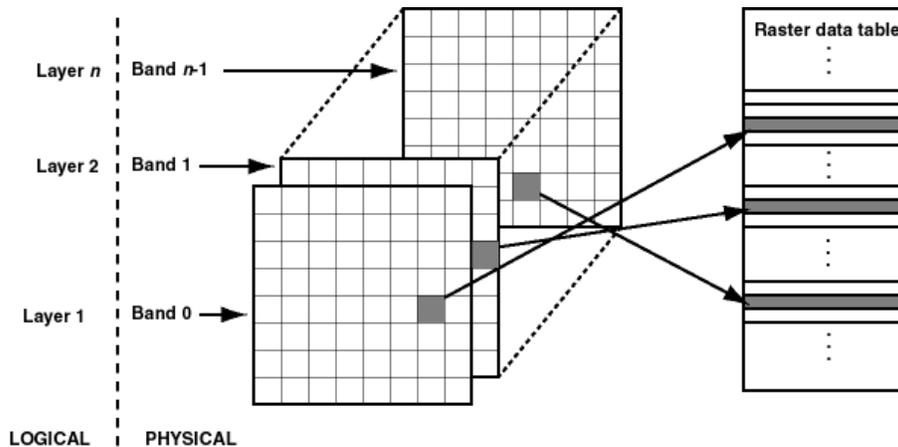
There are two basic raster data types supported in GeoRaster: grid-based data and image data.

- Grid-based, or gridded, data is a general term used for raster data such as digital terrain elevation, land use and land cover information, pollution concentration, geological information, and rainfall information. It is 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 stores the attribute values or attribute index values for each cell in the grid. If the index values are stored, different attribute values can be associated with the index values and are typically stored in a relational table, called Value Attribute Table or Raster Attribute Table.
- 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 spectral sensors, and is collected using a variety of technologies including satellite remote sensing and airborne photogrammetry. 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. Raster Attribute Tables can be associated with the image or its bands.

GeoRaster uses an integrated 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 and it applies to all cells. This core raster data set can be blocked for optimal storage, retrieval and processing. Pyramids (generalized, lower-resolution versions of the image – useful for fast retrieval in web applications) of the core raster data can be generated, stored and processed the same way.



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 sublayers are used to model the channels or bands of the imagery. In GeoRaster, each sublayer 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.



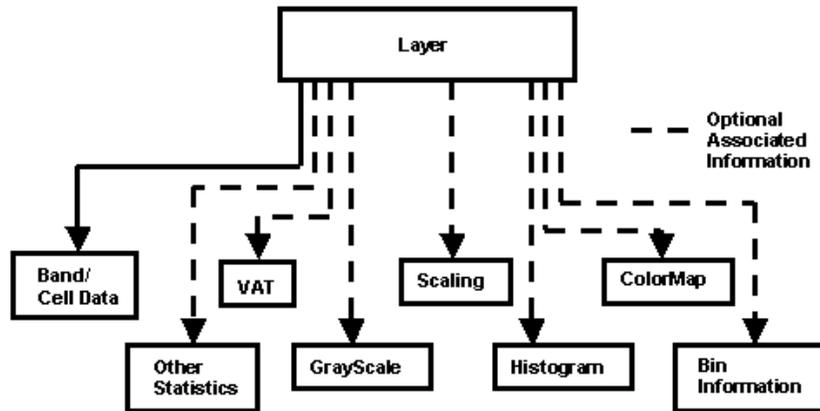
Besides the core matrix of cells, a raster data object has specific metadata associated with it. In GeoRaster, the metadata are organized into components that contain the following information:

- 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 user-defined ID, description and version information. The Raster Information includes metadata such as cell depth (e.g., 1BIT, 32BIT\_S, or 64BIT\_REAL), dimensionality, blocking sizes, interleaving types, compression and information about pyramids. The Spatial Reference System metadata contains information required for georeferencing, in which a generic polynomial model is defined and ground control points (GCPs) can be stored.

The Layer Information contains metadata pertaining to each logical layer in a GeoRaster object and it consists of many subcomponents. The major subcomponents of the Layer Information metadata include user-defined layer ID, scaling function, bin function, RGB colormap, grayscale lookup table, statistics and histogram, NODATA values and value ranges, and bitmap mask. A Value Attribute Table can be used to maintain information about the values stored in each layer and the table name can be registered in the Layer Information metadata. Each layer, including object layer, can have one bitmap mask associated with it, which is registered in the Layer Information metadata. As special

metadata, the bitmap masks are special raster grids with a cell depth of 1. The physical bitmap masks are stored together with the core raster data.



## 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\_GEOASTER data type, which is defined as:

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

The GeoRaster metadata is stored as the metadata attribute of the SDO\_GEOASTER 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\_GEOASTER type is the rasterType, which contains dimensionality information and the data type that can be extended.

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:

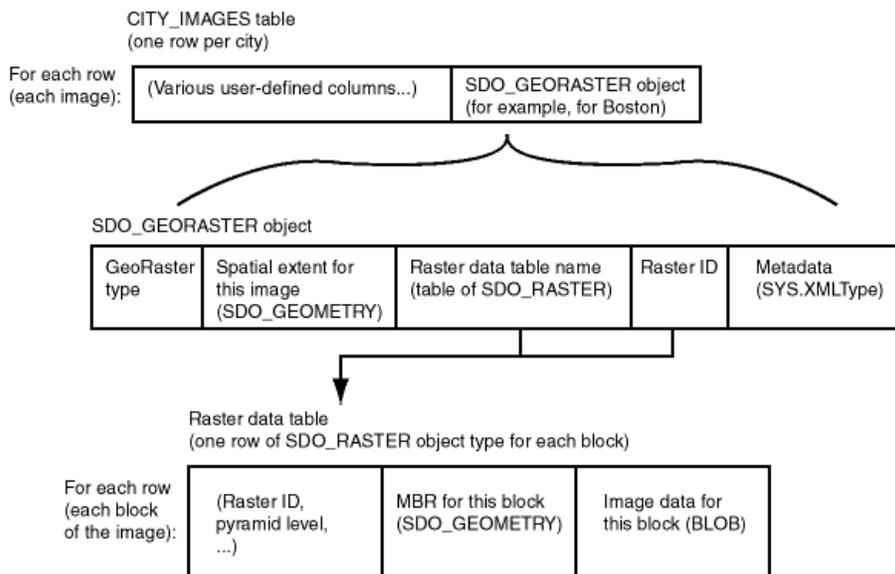
```
CREATE TYPE sdo_raster AS OBJECT (  
    rasterID NUMBER,  
    pyramidLevel NUMBER,  
    bandBlockNumber NUMBER,  
    rowBlockNumber NUMBER,  
    columnBlockNumber NUMBER,  
    blockMBR SDO_GEOMETRY,  
    rasterBlock BLOB);
```

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.

Both pyramid and bitmap masks of the GeoRaster object are stored in the same raster data table of the GeoRaster object, using the same blocking scheme. Pyramids can also be generated on the bitmap masks and stored in the same way as the pyramid of the GeoRaster object.

A blank GeoRaster object is a special type of GeoRaster object in which all cells have the same value. There is no need to store its cells in any SDO\_RASTER block; instead, the cell value is registered in the metadata in the `blankCellValue` element. GeoRaster 11g also supports empty raster blocks to save storage space with large mosaic objects and to improve raster processing speed. It is used when there is no raster data available for a specific raster block of a large GeoRaster object. Such GeoRaster data can be considered as a special sparse data type. There is still an entry in the raster data table for each empty raster block, but the length of the BLOB is zero, i.e., empty. Pyramids and bitmap masks can have empty blocks too.

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, the CITY\_IMAGES table, created by a user, is called a GeoRaster table, which is a relational table containing a column of object type SDO\_GEORASTER. The GeoRaster tables can contain any number of other columns (just like any other Oracle table).

The rasterDataTable and the rasterID attributes of the SDO\_GEORASTER object type provide the information required to store and retrieve the raster cell data in its raster data table. Internally, GeoRaster uses a system dictionary (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.

## 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. This section provides a general overview of the basic functional infrastructure available in GeoRaster.

GeoRaster operations can be grouped into the following categories:

- Database Creation
- Database Administration
- Data Manipulation

### Database Creation

From a user perspective, a GeoRaster database is basically an Oracle database containing a list of GeoRaster tables, in which each image or raster grid is stored as a GeoRaster object in a row of a GeoRaster column. It can contain a virtually unlimited number of GeoRaster objects in one or more database schemas and a single GeoRaster object can be terabytes in size.

To build a GeoRaster database, users use the standard SQL statements or PL/SQL language and leverage the GeoRaster API and/or third party tools and solutions. The key steps include:

- Create GeoRaster tables and RDT tables using the standard DDL.
- Load raster data from files in various raster formats into the GeoRaster tables using third-party ETL tools, the `SDO_GEOR.importFrom` procedure, or the client-side Java loader. GeoRaster provides some loading capability on TIFF, GeoTIFF, JPEG, BMP, GIF, PNG, and JP2. Georeferencing information can be loaded from ESRI world files, GeoTIFF files and Digital Globe RPC text files. The data can be loaded individually or in batch.
- Generate and update the spatial extent on each GeoRaster object if necessary. The spatial extent is of `SDO_GEOMETRY` type.
- To validate the GeoRaster objects, call `SDO_GEOR.validateGeoRaster`. To validate the metadata, call `SDO_GEOR.schemaValidate`.
- Build appropriate indexes on various columns of the GeoRaster tables e.g., a spatial R-tree index on the spatial extent attribute of the GeoRaster column and B-tree indexes on other columns. Users can also build function-based indexes on the metadata attribute of the GeoRaster objects.
- The GeoRaster objects can be updated and deleted. They can be copied from one place to another using the `SDO_GEOR.copy` procedure.
- To export the GeoRaster objects into files in various raster formats, use third-party ETL tools, the `SDO_GEOR.exportTo` procedure, or the client-side Java exporter. GeoRaster provides some support on exporting TIFF, GeoTIFF, JPEG, BMP, GIF, PNG, and JP2. Georeferencing information can be exported to ESRI world files, GeoTIFF files and Digital Globe RPC text files.

Once the database is created and the data are loaded into GeoRaster, users can manage it, tune it, perform spatial queries and various advanced processing as described in the next sections.

## Database Administration

The GeoRaster object type is a native Oracle data type. This approach allows users to manage the GeoRaster database using most standard RDBMS features, such as backup, partitioning and table security. In Oracle Spatial 11g, GeoRaster supports Oracle Workspace Manager for raster versioning and Oracle Label Security for row-level (raster block) security. In addition, GeoRaster provides the `SDO_GEOR_ADMIN` package, which includes more than 10 functions, to help manage and maintain GeoRaster databases. For example,

- The `listGeoRasterObjects`, `listGeoRasterColumns`, `listGeoRasterTables`, `listRDT` subprograms in the `SDO_GEOR_ADMIN` package help check the status of existing GeoRaster objects and related objects in the current schema or the database.
- The `SDO_GEOR_UTL.renameRDT` renames the RDT in the database to solve conflicts, which might happen during data migration.
- The `SDO_GEOR_ADMIN.maintainSysdataEntries` automatically maintains the `SYSDATA` entries in the current schema or the database.
- The `SDO_GEOR_ADMIN.upgradeGeoRaster` checks for and corrects errors after database upgrading.
- The standalone GeoRaster Viewer is a DBA and development tool. It can connect to multiple databases or database schemas and display the GeoRaster objects. It allows zoom in and zoom out, scroll through the raster at any pyramid level and can query cell values and coordinates. It provides basic image processing operators and can view bitmap masks.

## Data Manipulation

Besides leveraging the standard enterprise database features, GeoRaster provides over 100 raster data and metadata operations through the SQL API to optimally manage and manipulate GeoRaster objects in support of various application requirements. Users can achieve several goals using the following list of key operations provided by GeoRaster:

- Adjust the internal raster blocking size of any GeoRaster objects to improve scalability, optimize space usage and speed up raster processing and query. While each block must have the same size, the raster blocking sizes along different dimensions can be random numbers, not necessarily a power of 2.
- Change the band interleaving types among BSQ, BIL and BIP to best fit into different applications.
- Change the cell depth. Cell depths of 1-bit to 32-bit integers and 32-bit and 64-bit real numbers are supported.
- Generate pyramids using different resampling approaches and remove pyramids. Reduced resolution pyramids are supported. GeoRaster supports resampling methods of nearest neighbor, bilinear interpolation, cubic convolution, and average using four or six neighboring cells.
- Compress or decompress GeoRaster objects in lossless DEFLATE and lossy JPEG compression types. Some wavelet compressions are supported through third-party plugins. All GeoRaster functions that can be performed on uncompressed (decompressed) GeoRaster objects can be performed directly on compressed objects.
- Georeference GeoRaster objects using PL/SQL functions or load georeferencing information from files. A generic polynomial georeferencing model is supported for georeferencing rectified and unrectified airborne photos and satellite images. It supports up to a power of 5 and 3-D model coordinates. The affine transformation, DLT, RPC, and other models are special types of this

generic model. GCPs can be stored and used to georeference GeoRaster objects. The following geometric models are supported with GCP georeferencing: Affine, Quadratic Polynomial, Cubic Polynomial, DLT, Quadratic Rational, and RPC.

- Add or remove bitmap masks for GeoRaster objects and individual bands or layers. These bitmap masks are stored inside the GeoRaster objects. Pyramids of masks can be created and stored inside the GeoRaster objects as well. Masks are compressed when the GeoRaster object is compressed.
- Add or delete NODATA. Multiple NODATA values and multiple NODATA value ranges are supported for GeoRaster objects and their individual bands or layers.
- Add or delete scaling function, bin function, colormap and grayscale information.
- Search GeoRaster objects using Oracle Spatial index and operators, such as area-of-interest queries, spatial join queries, and other topology-based spatial operations.
- Crop images and perform subsetting to create new GeoRaster objects for persistent storage in the database or as an answer to a specific query for web distribution and display. Clipping the query or subset result along the polygon (irregular) boundary is supported.
- Query cell coordinates and do coordinate transformation between GeoRaster cell space and model space. GeoRaster supports sub-cell or sub-pixel addressing (floating row and column numbers) in the GeoRaster cell spaces.
- Query cell values or evaluate a direct location based on neighboring cell values by using a specified interpolation method, and returns the raster values for the specified bands or layers for that location.
- Mosaick large raster datasets inside the database. The GeoRaster mosaic function allows gaps, overlaps, and missing source GeoRaster objects.
- Merge or union GeoRaster objects or layers. Extract a single layer or a subset of layers from a GeoRaster object.
- Partially edit and update a window of raster data and its pyramids using another image or gridded data. Change the value of a single cell.
- Enlarge or shrink raster data using nearest neighbor, bilinear interpolation, cubic convolution, or average resampling.
- Reproject a GeoRaster object to a different Oracle Spatial coordinate system, either on-the-fly or persistently.
- Analyze statistics and generate histograms on the whole object or individual layers.
- Query, delete and update most items of the metadata through tens of functions or procedures. GeoRaster cell data and metadata update and query are crucial to successful use of GeoRaster in the Oracle Database. Some key subprograms include updating version information, querying the user-defined ID, the dimension sizes and blocking sizes, checking the SRS information, querying and updating time information, to name just a few.

The SQL API is provided for database creation, database administration and data manipulations. It also aids in the development of and integration with applications on top of GeoRaster. Users can use Java, C or C++ to leverage this API and the Java API or directly access the binary data of the open GeoRaster data model. 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.

## New Features in 11.1

This section outlines some of the new features in GeoRaster in Oracle Spatial 11g Release 1 (11.1).

### New Functions and Procedures

The current release provides over 30 new subprograms and other enhancements, including the following:

- The union or merging of GeoRaster objects and layers.
- Mosaic support allows for gaps, overlaps, and missing source GeoRaster objects.
- New GeoRaster template functions ease third-party software integration, so that the developers do not need to deal directly with the Oracle BLOB and XMLType types.
- Statistical analysis and histogram generation.
- GeoTiff, JPEG 2000, and Digital Globe RPC file formats are supported for loading and exporting GeoRaster objects. JPEG files can be loaded without decompression.
- GeoRaster supports the use of Oracle SecureFiles.

### New Metadata and Raster Support

The current release has additional metadata and raster type support including:

- A generic and complex polynomial georeferencing model is supported for georeferencing rectified and unrectified airborne photos and satellite images. The affine transformation, DLT, RPC, and other models are supported as special cases of this generic model.
- Multiple NODATA values and multiple NODATA value ranges are supported for GeoRaster objects and their individual bands or layers.
- GeoRaster objects can have empty raster blocks to save storage space and improve processing speed.

### Enhanced Ease of Use, Reliability, and Manageability

Certain configuration and administration tasks have been automated in this release. For example,

- When you create a GeoRaster table, you no longer need to create the GeoRaster DML triggers for the table. These DML triggers are created automatically, and their automatic creation and operation provides greater reliability.
- Internal changes that monitor DDL events on raster tables and activities on GeoRaster system data enhance the manageability, reliability, robustness, and usability of GeoRaster.
- Raster data versioning with Oracle Workspace Manager is supported.
- Raster data row-level security with Oracle Label Security is supported.

## New Features in 11.2

This section outlines some of the new features in GeoRaster in Oracle Spatial 11g Release 2 (11.2).

- The Java API, which consists of interfaces and classes that support features available with the GeoRaster feature of Oracle Spatial. This API provides a complete mapping of the SDO\_GEOASTER object type and its metadata to Java objects, and it offers Java methods to manipulate GeoRaster objects.
- Support ground control point (GCP) storage and georeferencing. The following geometric models are supported with GCP georeferencing: Affine Transformation, Quadratic Polynomial, Cubic Polynomial, DLT, Quadratic Rational, and RPC.
- Support raster reprojections both on-the-fly and persistently.
- Support grid interpolations, which evaluate a direct location using a specified interpolation method, and return the raster values for the specified bands or layers for that location.
- Support clipping along the polygon (irregular) boundary in raster query and subsetting.
- Provide the capability to automatically adjust the user-specified blocking sizes for storage parameters to an optimal value to reduce padding space, thus optimize GeoRaster storage.

Further details can be found in the Oracle Spatial GeoRaster Developer's Guide.

## Benefits of Managing Raster Data in Oracle Database

By effectively managing raster, vector and attribute data in a single data management environment – with common storage, indexing, spatial referencing, query optimization, security, and user management – Oracle reduces the processing overhead and eliminates the complexity of coordinating and synchronizing disparate sets of spatial data. Other benefits include:

- Raster, gridded, vector, XML and various types of attribute data can be stored on a single server.
- Better management of spatial data – through SQL access.
- Support by leading 3rd party image processing, GIS and visualization tools.

- Consolidation of disparate data management environments, such as GIS, remote sensing, business data.
- Greatly reduced complexity of systems management and better use of existing resources.
- Support for standard spatial types. The GeoRaster data model is an open and integrated raster data model.
- Scalability, data security, replication, partitioning, bulk load utilities.
- Breaking size barriers – support a virtually unlimited number of GeoRaster objects. A single GeoRaster object can be terabytes in size. Support large map sizes.
- Internet deployment – Enables a large number of concurrent users to access an application at virtually no additional cost with 24 x 365 uptime.
- Support for both short and long transactions.
- Oracle Spatial 11g vector data type compliance with the OpenGIS® Simple Features Specification for SQL, Revision 1.1, Types and Functions Alternative.<sup>1</sup>
- Reduced training, software, support and application integration costs resulting from consolidating raster, vector and attribute storage.
- Risk reduction – GeoRaster information is integrated into Oracle Database resulting in scalability, reliability, and fast performance.
- Market validation– In four separate surveys since 1999, IDC has found that Oracle holds 80-90% of the overall geospatial data management market.

## Application of GeoRaster

There is a wide range of use cases for GeoRaster. The section below highlights some use cases for deploying GeoRaster with 3rd party analysis and visualization tools.

### Commercial Data Repository and Distribution

- Problem – The volume of satellite imagery and aerial photos is growing exponentially. Digital Elevation Models (DEMs), thematic rasters, digital vector data and maps are created on a daily basis. They need to be securely and efficiently managed and quickly distributed to enterprises and consumers.

---

<sup>1</sup> For more details regarding standards compliance, please visit [www.oracle.com/technology/products/spatial](http://www.oracle.com/technology/products/spatial).

- Context – Numerous government agencies and commercial companies are collecting and producing all kinds of spatial data. Some data are sold to enterprises or delivered to the mass market through the internet.
- Solution – GeoRaster can be used to manage an unlimited number of imagery and raster datasets, including DEM and thematic raster maps. Each image or raster dataset can be terabytes in size. Satellite imagery and aerial photos can be mosaicked for large areas of interest. Oracle Spatial provides a robust and secure environment for managing vector spatial data types. Relational tables can be used to manage metadata and spatial attributes. Spatial indexing and other indexing technologies facilitate fast query of spatial data and metadata. Oracle Database provides a secure and scalable environment for multi-user concurrent access to the spatial datasets. The Oracle enterprise grid computing technologies enhance scalability and performance while providing the benefits of lower cost, higher quality and greater flexibility to meet the challenge of spatial data growth. ETL tools are used to load and export the spatial data for archiving and distribution. Internet-based 3D visualization clients and API can deliver the data and location based services to consumers in real time.
- Critical Fact – A large-scale enterprise database management system with raster data management capabilities provides an effective and efficient way to solve the problems associated with raster data archiving, management, processing and distribution.

## Defense and Security

- Problem – To detect threats to security and to ascertain threat levels while managing an appropriate response matrix, both friendly and unfriendly assets need to be tracked.
- Context – Detecting changes over time on the ground is a prerequisite to effective planning and response.
- Solution – Aerial and satellite remote sensing platforms are tasked to generate images from an area of interest on a regular basis. Images and DEM data are collected, georeferenced, and loaded into a database as raster layers. The current raster is mosaicked with surrounding rasters to create seamless coverage of areas of interest. Client tools are used to examine current images in association with historic images for the same surface coverage archived in the database. Visualization tools are used to display the imagery in 3D by combining the DEM raster layers.
- Critical Fact – Image data is the only geographic information that can be acquired in a scheduled/timely manner (tasked) for a specific local providing rapid access to current, accurate geodata. Such imagery and DEM data have huge volume and are costly and sensitive. Managing them in a secure and scalable environment is critical.

## Emergency Response

- Problem – Assess on-the-ground damage and develop suitable response scenarios given damage to critical infrastructure.

- Context – Making a rapid assessment of damage to infrastructure in the aftermath of a disaster event is critical to a timely response and mitigation.
- Solution – Raster data from aerial and satellite remote sensing platforms are used to compare “before and after” conditions on the ground. GeoRaster data is used in conjunction with associated vector data for continuous asset infrastructure (e.g., road, rail, power grid, gas, telco) to determine a) damage b) response scenario c) viable corridors for first responders and d) routing to appropriate facilities.
- Critical Fact – Only raster data can provide the near real-time data acquisition needed to accomplish damage assessment and meet the time critical requirements of first responders. A raster data layer provides an ideal backdrop to display infrastructure data (e.g., pipelines, transmission lines) in a readily comprehensible form. Spatial GeoRaster databases provide historical spatial datasets, fast data archiving and rapid data retrieving.

## Enterprise Asset Management

- Problem – Optimize preventative maintenance, field service and operations across a network of continuous and fixed assets (e.g., stations, substations and pipeline).
- Context – Accomplish ongoing monitoring and management across a variety of assets to support normal preventative maintenance and operations.
- Solution – Incorporate raster data from aerial photography into the existing geospatial data used to map assets under one single enterprise database management system. Raster data used in conjunction with vector information representing property boundaries, lease zones and easements enable field service personnel to save time and zero in on problem areas on the ground.
- Critical Fact – Raster data increases the efficiency of resources in the field and drives down costs making operations more efficient. Oracle Database provides an integrated environment for raster data, vector data and other enterprise information.

## State and Local Government

- Problem – State and local government have limited resources to manage zoning, tax assessment, etc.
- Context – State and local government budgets are stretched like no time in recent memory. With tax base often shrinking and public service expenses increasing, local officials are compelled to improve efficiency and to maximize coordination and communication between local departments in government.
- Solution – a set of base maps of digital raster data (digital orthophoto quads, DEM, and thematic raster maps) when stored in the enterprise database in state government provides a common frame of reference that can be used across multiple departments and cities (Transportation, Tax Assessment, Zoning, School Administration, etc.) to support decision making and the formulation and implementation of policy.

- Critical Fact – Raster data when stored in a database provides a common frame of reference that can be used across departments and state agencies. Benefits include data sharing, ease of access by many users, integration, and reduction in overall costs.

### Agriculture Monitoring

- Problem – National agricultural agencies need to document and verify agricultural utilization of the land. Government and farmers need to better planning their agricultural activities.
- Context – Farmers often are required to report the land use practices they employ and the crops they seed, allowing officials to project earnings based on yield. In addition, it is common under some circumstances to compensate a farmer for land that is left fallow or unseeded.
- Solution – Use of aerial image data acquired on an annual or monthly basis in conjunction with vector information from the land management agency enables officials and local agriculturalists to create an accurate record of acres in crop and acres fallow. Other raster datasets include LULC information, DEM, soil type information, geological information, underground water and rainfall information. Synthetic analysis of those raster layers help agricultural assessments and planning.
- Critical Fact – Image-based information acquired on an annual or monthly basis can provide the information needed to make the assessments to support equitable taxation and remuneration. Raster data analysis and historical comparison using a spatial database help governments and farmers to better plan their activities.

### Insurance Risk Assessment

- Problem – Minimize exposure to risk. The risk assessment process includes complex analysis of thousands of, and sometimes tens of thousands of, layers of gridded data containing loss estimates by catastrophic event type.
- Context – Efficient management and timely processing of these large datasets are crucial. In some instances, the generation of loss estimates using catastrophe models and subsequent overlay analysis for producing risk or hazard maps can take days, depending on how the data is stored and indexed. Reducing this to a day or a matter of hours means obvious boost in productivity and revenue.
- Solution – Incorporate raster (gridded) data from catastrophe models into one enterprise geospatial database management system. Each event type and its loss estimate are stored as a separate raster object covering the region of interest. The data is stored in blocks for very efficient piecewise access by individual cell, or a cell and its immediate neighborhood, or subset of cells within a rectangular region of interest (or block). Efficient spatial and non-spatial index based access methods facilitate extremely fast retrieval of relevant cell values from the many thousands of layers that must be processed. As a result a computational process that, in some cases took many hours and occasionally days, was completed in a matter of minutes using the appropriate raster analysis and modeling client tools, and a geospatially enabled DBMS.
- Critical Fact – The vastly reduced computational processing time increases the efficiency of the risk assessors and drives down costs, making operations more efficient.

## Conclusion

The availability of GeoRaster in Oracle Spatial 11g 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 native data type. It enables 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 and other indexes provide fast retrieval of raster metadata and cell data. Selective raster query tools and other operations can be easily built upon GeoRaster. It also provides good extensibility by using XML technology for describing the metadata. 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 and take advantage of enterprise grid computing technologies to support mission critical applications and integration.



Oracle Spatial 11g GeoRaster  
February 2011  
Authors: Qingyun (Jeffrey) Xie and Jayant  
Sharma  
Contributing Author: Jean Ihm

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



Oracle is committed to developing practices and products that help protect the environment

Copyright © 2011, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. UNIX is a registered trademark licensed through X/Open Company, Ltd. 1010

**Hardware and Software, Engineered to Work Together**