Working with the Geodetic *and* Spatial DataBlade Modules

The Geodetic and Spatial DataBlade modules are server extensions that manage spatial data using different GIS technologies. This technical note describes the differences between these DataBlade modules and how to migrate data between them.

Contents

Overview

The Geodetic DataBlade module and the Spatial DataBlade module are server extensions that manage GIS data in an Informix database. They use different core technologies that solve

different problems and complement each other:

- The Geodetic DataBlade module treats the earth as a globe. Geometric operations are precise regardless of location.

  The Geodetic DataBlade module is best for global datasets and applications, such as satellite imagery repositories.

- The Spatial DataBlade module treats the earth as a flat map. Projecting the curved earth onto a flat map creates distortion around the edges of the map, so geometric operations can be inaccurate around those edges.

  The Spatial DataBlade module is best used for regional datasets and applications.

This section explains the benefits of managing this data in an object-relational database (ORDBMS) and introduces the core technology for both DataBlade modules.

Why use an ORDBMS for spatial data?

First of all, what exactly is a DataBlade module? and why on Earth (*cough cough*) would you want to manage spatial data in the database?

A DataBlade module is a server extension. Typically it extends the server to manage a new data domain, such as geospatial data. Informix's object-relational database technology lets developers extend the server with new:

- User-defined types (UDTs).
- User-defined routines (UDRs), which are functions and procedures implemented in C, Java, or SPL (Informix Stored Procedure Language).
- Indexing methods.
- Other access methods; for example, to access to external data stores.

DataBlade modules take advantage of the following features of an object-relational database management system (ORDBMS).

Uses standard SQL

DataBlade functions are called in SQL statements, a language with which customers are already familiar. Customers can start taking advantage of new features immediately; they don't have to learn a new language.

Simplifies client applications

Since DataBlade functions are callable from SQL, it simplifies client application programming.

Furthermore, client applications don't have to be relinked with new versions of the source code. After a new version of the DataBlade is installed, SQL queries automatically start using the new version.

Finally, if the client runs on one machine architecture and the database on another, the Informix

server automatically performs any required conversion for transporting data between architectures. (OK, so DataBlade modules have to add some generic support code for the server to call, but even the DataBlade module code does not need to know the machine architectures.)

Manages complex data

The complexity of processing new data is pushed into the server, again simplifying application code.

Ensures Data Integrity

Since a DataBlade module runs as part of the server, it enjoys all the database services that ensure data integrity, such as:

- Concurrency control and transaction management.
- Backup and recovery.
- Any triggers that the customer may have created to enforce local rules.

Improves Performance

The database scales for multiple users far better than client applications do.

Side-by-side tests using the Spatial DataBlade module demonstrate that DataBlade module performance is measurably better than the exact same code running in a client application as you add more concurrent users.

Some of the features that help the database scale so well include:

- IDS buffers data in memory.

  Internally, the Informix server buffers database pages fetched from disk in memory. This means that if many users query overlapping data, the database can fetch results from the internal buffers and avoid expensive disk i/o. Database administrators can tune the size of those buffers (and can also tune many other related parameters, such as how many database pages should automatically be read ahead).

- Query optimizer reduces internal processing.

  Since DataBlade modules are tightly integrated into the server, the query optimizer can:

  - Determine if any indices are available to filter results.
  - Delay execution of expensive functions so that they execute on the fewest intermediate results possible.

- Just results are delivered.

  Just query results get delivered to the client, reducing network traffic.

- Random access to large data also reduces results delivered.

  An IDS 7.x blob ("dumb blob") must be completely read into memory before the data can be manipulated. Only a client application can manipulate a dumb blob; an SPL routine

cannot.

An IDS 9.x smart large object ("smart blob") provides random access to the data. The entire object does not have to be read; instead, the code can directly jump ("seek") to the interesting data, then clip out just the required data. A client application can manipulate smart blobs, but more importantly, a UDR written in C or Java can randomly access smart blobs. (An SPL routine can manipulate a UDT that contains a smart blob by invoking UDRs for that UDT.)

Distributes Data

Informix has two data distribution mechanisms:

- Enterprise Replication (ER), which asynchronously replicates data between different servers.
- ISTAR, which supports distributed queries that cross database servers.

ER and ISTAR will support UDTs in the upcoming 9.3 server release.

Geodetic DataBlade

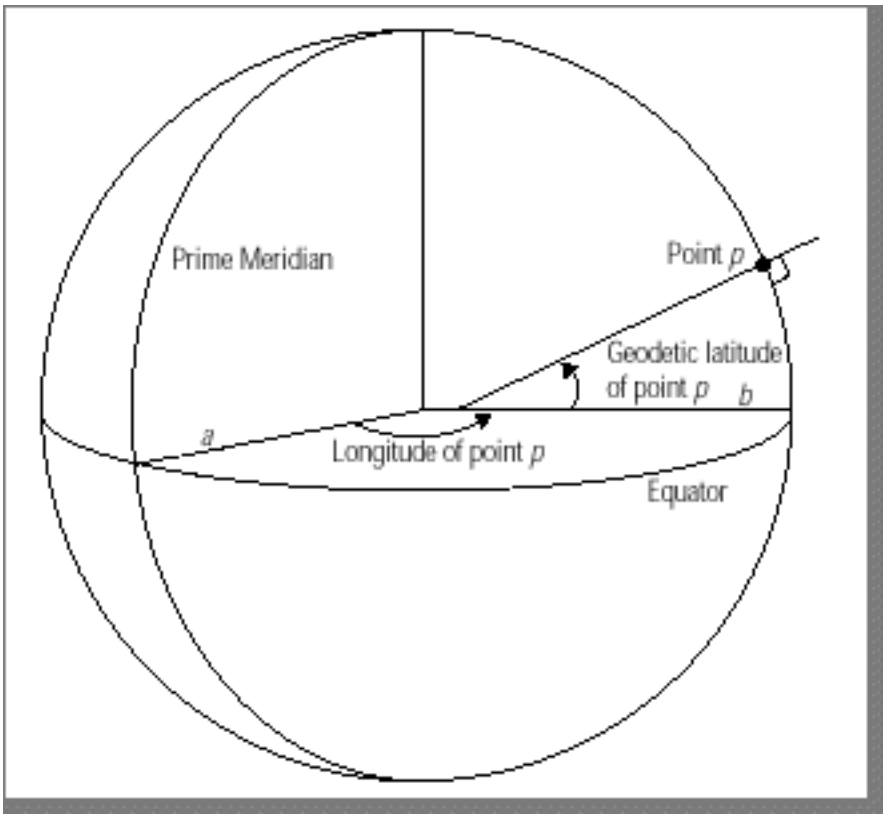
The following description is adapted from the Informix Geodetic DataBlade Module User's Guide, chapter 1, "Concepts and Principles".

The Informix Geodetic DataBlade module is developed in partnership with Geodyssey, Ltd., of Calgary, Alberta, Canada.

The Geodetic DataBlade module takes its name from the discipline known as **geodesy**. Geodesy is the study of the size and shape of the Earth (or any body modeled by an ellipsoid, such as the Sun or the celestial sphere). The Geodetic DataBlade module is designed to handle objects defined on the Earth's surface with a high degree of precision.

To do this, it uses a latitude and longitude coordinate system on an ellipsoidal Earth model, or **geodetic datum**, rather than a planar, **x-** and **y**-coordinate system. This avoids distortions, inaccuracies, and imprecision that can be introduced using flat-Earth projection.

A geodetic datum is a reference system that describes the surface of the Earth, and it is based on an approximation of the general shape of the Earth by an ellipsoid of rotation (also called a **spheroid**). This is the three-dimensional shape described by an ellipse when it is rotated around one of its axes. The Geodetic DataBlade module defaults to the World Geodetic System 1984 (WGS-84) datum, but it supports over 175 geodetic datums and over 40 ellipsoid specifications, and also allows defining new datums.



The Geodetic DataBlade module's coordinate reference system uses **geodetic latitude** and **longitude** to describe locations relative to the Earth:

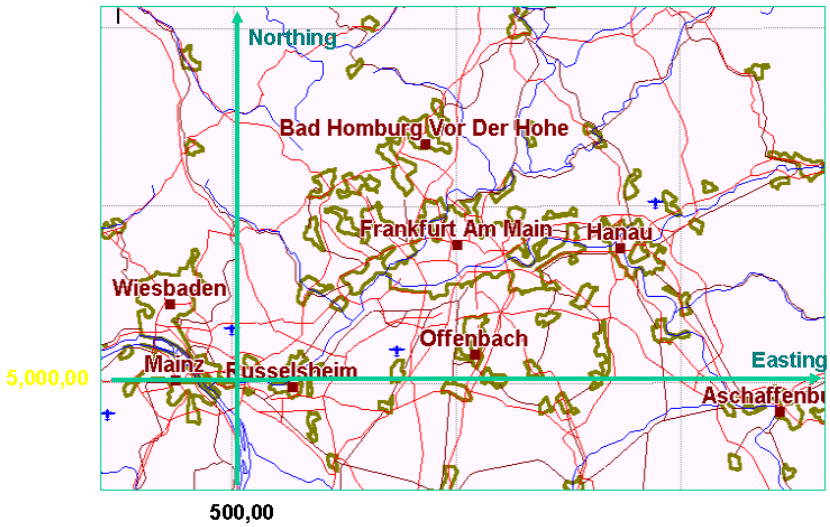
- The geodetic latitude of point ***P*** in the diagram above is the angle between the datum surface normal at the point and the equatorial plane.
- Longitude is the angle in the equatorial plane between the line ***a*** that connects the Earth's center with the Prime Meridian and the line ***b*** that connects the center with the meridian on which the point lies. A **meridian** is a direct path on the surface of the datum that is the shortest distance between the poles.

Geodetic latitude and longitude are always based on a specific datum.

Spatial 8.1 DataBlade

The Informix Spatial 8.1 DataBlade module is developed in partnership with ESRI. It is the successor to the ESRI SDE 3.0.2.2 DataBlade module, which is currently generally available. As of this writing, the Spatial DataBlade module is in beta test.

The Spatial DataBlade module uses planimetric (flat-plane) geometry, which means that it approximates the round surface of the earth by projecting the surface onto flat planes using various transformations.



To apply simple, flat-plane coordinates to the earth, first you apply a transformation to map the round earth's surface to a flat piece of paper. This is called a **projection**. You can't wrap paper around a ball without wrinkling or tearing it, so it always introduces distortions of distances, angles, and areas. To keep the distortion within bounds, the area covered by the map must be limited. World maps cannot be used for reliable angle, direction, distance, or area measurements.

The rectilinear coordinates (x and y) in a projected map are usually called **Easting** and **Northing**. Often, the theoretical origin (0,0) is arbitrarily put well out of the way of the map in question so that all the coordinates in the map are positive.

Contents

DataBlade Features

User-Defined Types

Each DataBlade creates new user-defined types (UDT) that are organized into a type hierarchy, with many subtypes under a single supertype. The types themselves are slightly different for each DataBlade module, although there is some overlap (point, line, polygon):

Geodetic Type Hierarchy

GeoObject

GeoPoint	GeoBox	GeoCircle	GeoEllipse	GeoLineseg	GeoRing	GeoString	GeoPolygon

Spatial Type Hierarchy

ST_Geometry

Curve	*Surface*		*Geometry Collection*		
ST_LineString	ST_Polygon	ST_Point	ST_MultiPolygon	ST_MultiLineString	ST_MultiPoint

Each subtype inherits the functionality of the supertype.

Furthermore, the user can create a column that is of the supertype, such as GeoObject or ST_Geometry, then mix subtypes in that one column. Or the user can create a column of the subtype, thereby restricting the column to values of that one type.

External Formats

Both DataBlade modules support external, public formats.

Spatial

- ESRI Shape

  The ESRI shape format is public. The Spatial DataBlade module comes with utilities that load and unload shapefiles. Additionally, source code is freely available for accessing and manipulating shape files:

  - #

- OpenGIS

  The OpenGIS Consortium specified a data transfer standard for geospatial data. The Spatial DataBlade module provides functions that convert between a UDT value and the OpenGIS well-known text (WKT) and well-known binary (WKB) formats. The specific functions are listed below in the section on "Converting Spatial to/from OpenGIS formats".

Geodetic

- OpenGIS

  The section below on "Converting Geodetic to/from OpenGIS formats" describes how Geodetic support the OpenGIS formats.

- See the online release notes for additional supported formats.

More Features

Feature	Description	Geodetic	Spatial
Storage	No size restriction is imposed on a UDT value. Small values get stored in-row, large values get stored in a smart large object. The location of a particular value is transparent to the end user.	yes	yes
Index support	Can create R-tree index on UDT column.	yes	yes
Client application support	ESQL/C access to UDT objects	yes	yes
	ODBC access to UDT objects	yes	yes
	Java API	yes	*TBD*

Enterprise Replication (ER)	The Informix server replicates data between database servers and a variety of replication models is available. UDT replication is supported starting in 9.3.	yes	*TBD*
Parallel Query (PDQ)	PDQ means that the Informix server executes one query on many virtual processes in parallel.	yes	*TBD*

Contents

DataBlade Functions

This section lists Geodetic and Spatial functions side-by-side.

Functions that describe properties of an object

Function Description	Spatial	Geodetic
The area of a polygon, etc.	ST_Area	Area
The orientation of a GeoEllipse or bearing of a GeoLineseg.		Azimuth
The boundary of an object, which is the interface between its interior and exterior.	ST_Boundary	
Returns: 2 if an object has neither Z or M values 3 if an object has either Z or M values 4 if an object has both Z and M values	ST_CoordDim	
Returns: 0 if object has neither length or area 1 if object has a non-zero length 2 if object has a non-zero area	ST_Dimension	Dimension
The last point of a LineString.	ST_EndPoint	Coords(obj, NPoints(obj))
The minimum bounding rectangle of an object.	ST_Envelope	
The outer ring of a multi-ring polygon.	ST_ExteriorRing	
The Nth component of a geometry collection.	ST_GeometryN	
The geometry subtype.	ST_GeometryType	IsGeoBox, IsGeoPoint, *etc.*
The Nth interior ring of a multi-ring polygon.	ST_InteriorRingN	Ring

Determine if the first and last vertices of a linestring are the same.	ST_IsClosed	
Determine if an object has no vertices. (Empty objects are created, for example, when you compute the intersection of two non-intersecting polygons.)	ST_IsEmpty	
Determine if a linestring is both closed and simple.	ST_IsRing	
Determine if a linestring crosses itself, *etc.*	ST_IsSimple	
Determine if a polygon is topologically correct.	ST_IsValid	IsValidGeometry
Determine if a polygon complies with the SDTS definition of a polygon. (SDTS is a U.S. government standard.)		IsValidSDTS
The length of a linestring.	ST_Length	Length
The perimeter of a polygon.	ST_Perimeter	Length
The number of components in a geometry collection.	ST_NumGeometries	
The number of interior rings in a multi-ring polygon.	ST_NumInteriorRing	NRings
The number of vertices in a polygon, linestring, etc.	ST_NumPoints	NPoints
Nth vertex of a linestring.	ST_PointN	Coords
The spatial reference identifier of an object.	ST_SRID	SRID
The first point of a linestring.	ST_StartPoint	Coords(obj, 1)
The X coordinate of a point.	ST_X	Longitude
The maximum X coordinate of an object.	SE_Xmax	
The minimum X coordinate of an object.	SE_Xmin	
The Y coordinate of a point.	ST_Y	Latitude
The maximum Y coordinate of an object.	SE_Ymax	
The minimum Y coordinate of an object.	SE_Ymin	

Functions that create new objects

Function Description	Spatial	Geodetic
Create a polygon which consists of all points a specified distance away from an object.	ST_Buffer	

Create a point which is near the center of an object.	ST_Centroid	Center
Create the convex hull (the minimum bounding convex polygon) of an object.	ST_ConvexHull	
Create an object which is the set difference of two objects. (logical AND NOT of space)	ST_Difference	
Create a reduced-resolution copy (i.e. with fewer vertices) of a linestring, polygon, etc.		Generalize
Create an object which is the set intersection of two objects. (logical AND of space)	ST_Intersection	
Create a point which is a specified distance and bearing from another point.		Move
Create a point which is guaranteed to be within a polygon.	ST_PointOnSurface	
Create an object which is the set symmetric difference of two objects. (logical XOR of space)	ST_SymmetricDiff	
Convert the coordinates of an object from one spatial reference system to another.	ST_Transform	
Create an object which is the set union of two objects. (logical OR of space)	ST_Union	

Functions that determine spatial relationships between two objects

Function Description	Spatial	Geodetic
Calculate the bearing from one point to another point.		Azimuth
Proximity test: Determine if one object is a specified distance away from another object.		Beyond
Determine if one object completely contains another object.	ST_Contains	Contains
Determine if a linestring crosses a polygon, etc.	ST_Crosses	
Determine if two objects do not intersect.	ST_Disjoint	Outside
Calculate the minimum distance between two objects.	ST_Distance	Distance

Determine if the bounding boxes of two objects intersect.	SE_EnvelopesIntersect	
Determine if two objects have identical X,Y coordinate values.	ST_Equals	Equal
Determine if two objects intersect.	ST_Intersects	Intersect
Used in nearest-neighbor queries.	SE_Nearest	Nearest
For nearest-neighbor queries which only measure the distance between objects' bounding boxes.	SE_NearestBbox	NearestBbox
Determine if two objects are equal and coordinates are in the same order.	ST_OrderingEquals	
Determine if two objects of the same dimension intersect.	ST_Overlaps	Intersect
Determine if a specific DE-9IM spatial relationship between two objects exists.	ST_Relate	
Determine if the boundaries, but not the interiors, of two objects intersect.	ST_Touches	
Determine if one object is completely within another object. (commutator of Contains)	ST_Within	Inside
Proximity test: Determine if one object is within a specified distance of another object.		Within

Functions that deal with Z values

Both the Spatial and Geodetic DataBlade modules allow you to store a double-precision value with each vertex of an object. This is intended for storing the altitude or depth of a vertex, but it can be used for many other purposes. Z values are not considered when testing the spatial relationship between two objects.

Function Description	Spatial	Geodetic
Determine if an object has Z values.	SE_Is3D	HasZValue
Set an object's bottom and top components of its AltRange equal to the minimum and maximum Z values.		SetAltRangeZ
The Z value of a point.	SE_Z	ZValue
The maximum Z value of an object.	SE_Zmax	
The minimum Z value of an object.	SE_Zmin	

Functions that deal with Measures (M values)

The Spatial DataBlade module allows you to store an additional double-precision value with each vertex of an object. This is intended for storing the distance of a vertex from the beginning of a linestring, but it can be used for many other purposes. M values are not considered when testing the spatial relationship between two objects.

Function Description	Spatial	Geodetic
Determine if an object has M values.	SE_IsMeasured	
Create an object which represents the points or line segments that are at a specified measure along an object.	SE_LocateAlong	
Create an object which represents the points or line segments that are between two specified measures along an object.	SE_LocateBetween	
The M value of a point.	SE_M	
The maximum M value of an object.	SE_Mmax	
The minimum M value of an object.	SE_Mmin	

Functions that deal with AltRanges and TimeRanges

The Geodetic DataBlade module allows you to associate an altitude range (an ordered pair of double precision values) and a time range (an ordered pair of datetimes) with an object.

Function Description	Spatial	Geodetic
Obtain the altitude range of an object.		AltRange
Obtain the start time of a time range.		Begin
Obtain the lower bound of an altitude range.		Bottom
Obtain the stop time of a time range.		End
Set an object's bottom and top components of its AltRange equal to the minimum and maximum Z values.		SetAltRangeZ
Modify the altitude range of an object.		SetAltRange
Modify the time range of an object.		SetTimeRange
Obtain the time range of an object.		TimeRange
Obtain the upper bound of a time range.		Top

Contents

Migrating Data between Geodetic and Spatial

The OpenGIS WKB and WKT formats let you migrate data between Geodetic and Spatial. Both DataBlade modules provide functionality for converting UDT values between the internal representation and WKB / WKT.

Converting Spatial to/from OpenGIS Formats

The Spatial DataBlade module provides the functions summarized below for converting UDT values between OpenGIS formats and the internal DataBlade format:

	to ST_Geometry	from ST_Geometry
WKB	ST_GeomFromWKB	ST_AsBinary
WKT	ST_GeomFromWKT	ST_AsText

Converting Geodetic to/from OpenGIS Formats

Geodetic does not use functions to convert data to a specific format.

Instead, the GeoParam metadata table manages the data format for transmitting data between client and server. If the "data format" parameter is set to "OGC", then binary i/o is in WKB format and text i/o is in WKT format. (For specific details, see Chapter 7 in the Informix Geodetic DataBlade Module User's Guide).

Caveats

There isn't always a direct mapping between Geodetic and OpenGIS data types.

The table below (from the Geodetic on-line release notes) summarizes the mappings. The mapping arrow indicates the supported conversions between OpenGIS and Geodetic types. Bidirectional arrows signify a direct mapping. Unidirectional arrows signify a conversion which is only possible in one direction. For example, GeoCircles are converted to Polygons on output, but there is no conversion from an OpenGIS polygon to a GeoCircle.

Geodetic type	Mapping	OpenGIS type
GeoPoint	⟷	Point
GeoLineseg	⟶	LineString
GeoString	⟷	LineString
GeoRing	⟷	LineString

GeoBox	→	Polygon
GeoCircle	→	Polygon
GeoEllipse	→	Polygon
GeoPolygon	↔	Polygon or GeometryCollection of Polygons
not supported		MultiPoint
not supported		MultiLineString
not supported		MultiPolygon
not supported		GeometryCollection of Points
not supported		GeometryCollection of MultiPoints
not supported		GeometryCollection of LineStrings
not supported		GeometryCollection of MultiLineStrings
not supported		GeometryCollection of GeometryCollections

Contents

For more information

Geodetic DataBlade Module

The Informix Geodetic DataBlade Module User's Guide is available for download from the following Informix web site:

- #

Spatial DataBlade Module

The Managing Spatial Data: the ESRI Spatial Database Engine for Informix (White Paper) is available for download from the following Informix web site:

- #

Once the Spatial DataBlade is generally available, the user guide will also be downloadable from the Informix web site.

OpenGIS

Information about the OpenGIS consortium is available at #.

Contents

Glossary

Terms and acronyms used by this tech note include:

DataBlade Module	Server extension
IDS	Informix Dynamic Server
ORDBMS	Object-relational database management system
UDR	User-Defined Routine
UDT	User-Defined Type
WKB	Well-Known Binary, OpenGIS data transfer format
WKT	Well-Known Text, OpenGIS data transfer format

Contents
