PPDM Association

PPDM Version 3.5

Spatially Enabling Reference Guide

Last updated for PPDM 3.5
Version 1.1

Prepared for PPDM Association by

Chuanyun Fei, University of Calgary

Chuck Smith, AnGIS Software Inc.

David Warren, The TPI Group Inc.

Eun Hwan Shin, University of Calgary

L.A. McCulloch-Smith, Geowrite Inc.

Leo Bynum, GIS Results

Steve Kent, Omnium Informatics

Trudy Curtis, TruBear Custom Design Inc.

Vance Lickiss, PC Connections

…the business driven standard

Table Of Contents

PPDM Copyright and Disclaimer 3
About This Document 4
Abstract 5
Introduction 6
Business Process Overview 7
Pilot Project Summary 9
Method 1 *9*
Method 2 *9*
Method 3 *10*
Requirements Evaluation *10*
Implementation Considerations 12
Spatially Enabling with Spatial Data Cartridge (SDC) *13*
Spatially Enabling with Spatial Data Engine (SDE) *19*
Frequently Asked Questions (FAQ) 27
SEIS_LINE_GEOMETRY: *27*
Appendix A: Internet Resources 28
World Wide Web *28*
Discussion Groups *28*
References 29
Index 30

PPDM Copyright and Disclaimer

About This Document

This reference guide has been prepared to help managers, analysts, DBAs, programmers, data managers, and users understand how the Data Model is intended to be used. Readers at many levels, from managerial to technical implementers will benefit from reading this document. General, high-level business information is contained at the beginning of the document, with each section becoming progressively more technical and detailed.

Sometimes the terms we use in this and other PPDM documents need to be defined. We provide definitions in a separate Glossary, which you can obtain from PPDM.

This reference guide contains the following sections:

- Introduction

  Provides background information for the process of spatially enabling the PPDM database.

- Business Process Overview

  Summarizes the impact, of the method of spatially enabling the PPDM model, on business requirements.

- Pilot Project Summary

  Discussion of the choice of methods of spatially enabling the PPDM model.

- Implementation Considerations

  Discusses the implementation of spatially enabling the PPDM model. Specific reference is made to Oracle SDC and ESRI SDE.

- Frequently Asked Questions

  Addresses technical and business questions that may be asked about the spatial enabling process.

- Appendix A - Internet Resources

  Discusses related information available on the World Wide Web.

- References

  Provides a listing of publications and references used to write this document.

Abstract

Industries worldwide are aware that much of the data we work with has a critical spatial component.

GIS (Geographic Information System) tools provide the means to integrate data within a spatial and geographically visual context. This is important because digital data is rapidly replacing paper maps, production records, and well logs. GIS technology provides easy and intuitive access to data both for technical professionals and managers, thereby upgrading the utility of current data stores and improving access to information by management. Consequently, GIS applications can be used throughout the life of a petroleum asset and can effectively shorten the cycle time in prospect generation.

However, tools for spatial and statistical analysis and visualization are only as useful as the quantity and quality of readily accessible data. Standards such as PPDM are thus the foundation upon which other enabling technologies, including GIS, depend. Linking multi-disciplinary relational data from the PPDM model with a GIS provides the ability to work with many types of proprietary and public data and to extract the most information from Earth Science data stores in real time in a low-cost PC-based environment.

The PPDM Association has designed a methodology for linking the power of the PPDM database with the dynamic spatial search and query capability of today's GIS Engines. This document describes methods for spatially enabling the PPDM model version 3.5 with two of the most commonly used spatial engines, SDE (ESRI Corporation) and SDO (Oracle Corporation). Supplemental materials are also provided; these include sample data and load procedures for SDO and SDE.

Be aware that it is not the intention of the PPDM Association to go into the software business with the release of these procedures. They come without any guarantees (see the Disclaimer at the beginning of this guide), and PPDM will only be able to provide limited support. These procedures provide examples of how the data can be loaded and are intended to get you started. You should have your systems analysts and GIS experts examine the procedures before running them to ensure that you know what they will (and will not) do.

Introduction

In addressing the issue of developing a standard methodology for spatially enabling PPDM, the work group considered work already done by members; technology available from major software vendors today, including Oracle and ESRI; and the work of standards organizations, such as the Open GIS Consortium (OGC).

Defining a standard format for the data structures within spatial engines is within the scope and mandate of the OGC (Open GIS Consortium). The PPDM work group agreed to adopt this standard as it is completed and adopted by industry. However, the major GIS vendors today are only beginning to conform to the OGIS standard, making it impractical to implement as part of PPDM at this time.

The objective of this module is to augment the PPDM to accommodate direct links to a GIS. This will greatly simplify the task of linking the value of the data stored in PPDM to the powerful spatial analysis capability of a GIS.

To accomplish this, we:

- investigate, evaluate, and recommend standardized PPDM structures to store point, line, and polygon data.
- define the method to link these structures to the existing relational entities in PPDM.
- define the method to link these structures with existing GIS engines, where feasible.

The defined structures must support the following business requirements:

- rapid query and retrieval of information based on spatial attributes
- rapid display of PPDM attribute data on a map
- access by current GIS tools

Commonly accepted standards are examined and adopted, where appropriate.

The scope of this project has been limited to defining the structures to store the spatial information and to defining scripts and methods for associating the selected data base structures with existing GIS engines. Defining GIS functionality, topology, and the indexing of topology are currently beyond the scope of the module.

Business Process Overview

Industry analysts have estimated that at least 80% of the data used by geoscientists has a significant spatial component. The PPDM has been designed to capture geographical location data in order to spatially locate business objects, such as wells, land parcels, facilities, seismic lines, etc. Unfortunately, usage of this spatial data has been limited because it has not been tied to a spatial engine. Retrieving information based on spatial relationships can be slow, onerous, or even impossible using relational database tools. Yet this is the type of information that is required to perform solid business analysis of the data. The following are two examples of this type of query:

- Show the wells and seismic lines that intersect this land lease.
- Show all the wells within a one kilometer radius of this well that have a minimum penetration of Wabamun.

GISs have been built to derive exactly these types of spatial interrelationships. However, to make use of the power of a GIS, you must supply the data in a specified format. This format does not correspond to how PPDM currently stores its spatial data.

Linking relational data from the PPDM model with a GIS provides the ability to work with many types of proprietary and public data and to extract the most information from Earth Science data stores in real time in a low-cost PC-based environment. This marriage is an essential ingredient in the end-to-end solution for importing, editing, processing, analyzing, presenting, and archiving crucial project data.

Users require the ability to zoom into any area, such as a state, province, or sedimentary basin using a base map to display any combination of map features from a database that includes surface geology; shaded relief; subsurface geological, geophysical, engineering, cadastral, biological, socioeconomic, and many other information themes.

GIS tools provide the means to integrate data within a spatial and geographically visual context. The following examples show how GIS functionality can affect your business:

- GIS enables analysis, assessment, and evaluation of risk, incorporating such factors as distance from production for a given formation over a given period of time.
- Spatial and temporal analysis by type of hydrocarbon, well depth, formation, geological age, sedimentary basin, etc., lends itself, over a given period of time, to analysis using a GIS.
- Pie diagrams of oil and gas resources showing cumulative production, reserves, expected field growth, and mean undiscovered resources are best done in conjunction with GIS technology.

- Spatial display of information is central to decision making when planning 3D seismic surveys to meet quality, cost, and time targets.
- Spatial analysis is required for remediation decisions for sites where the soil and groundwater are contaminated by petroleum-derived product because of leakage of pipelines or underground storage tanks.
- Petroleum system studies require spatial data, such as zones of active source rocks and potential reservoir facies combined with attribute data, such as organic carbon content of a source rock unit and average porosity of a given reservoir unit.
- Spatial descriptions of reservoir properties (such as porosity and permeability) are important for the quantification of reservoir heterogeneity. Reservoir characterization requires an effective integration of reservoir data from various sources (e.g., seismic surveys, well logging, core analysis, pressure tests, and production history). The integration of this information is best done in a spatial context.
- Visualization and analytical techniques provided by GIS are very effective in assessing the accuracy and integrity of spatially distributed data. Thus, GIS plays an important role in data quality control and data management.

The union of standardized attribute data management and spatial data management will:

- promote the integration of surface and subsurface data for exploration and development
- foster partnerships among groups with common interests in resource management
- provide powerful and dynamic tools for environment assessment and resource management
- foster complementary technology such as time-lapse visualization of reservoir and aquifers

Combining the already proven technologies of the PPDM relational model with GIS analysis tools will enable petroleum companies and government agencies to provide their professionals with easily accessed, uniform, low-cost integrated data management, base mapping, and data analysis solutions.

Pilot Project Summary

The work group decided that creating a standard methodology for associating PPDM information generically with the spatial capability of a GIS would provide important and timely benefits for the membership today. Using relational SQL 92 technology, upon which the version 3.x series of PPDM is implemented, three alternatives were considered.

Method 1

A relational breakout table (Figure 1) is created between the PPDM table and the Geometry table. This solution does not require modifications to the proprietary spatial data store and provides flexibility to reuse geometry or create more than one geometry for each business object.

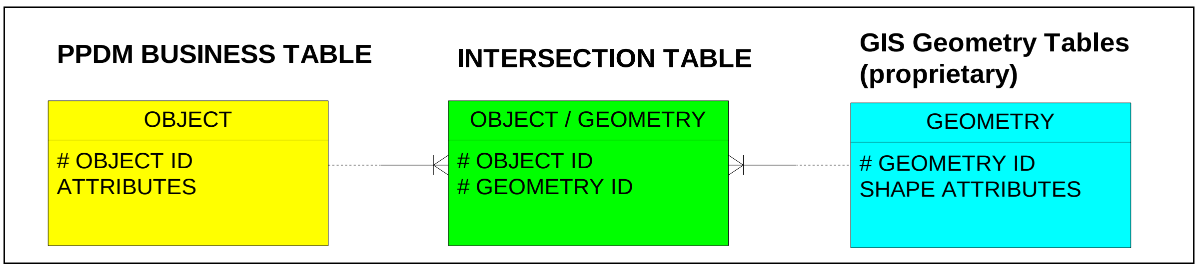


Figure 1: Method 1 requires a relational breakout table.

Method 2

In this option (Figure 2), a foreign key from the Geometry table is inserted into the appropriate PPDM table. This scenario allows each business object to have only one associated geometry. In many business cases, it is useful to define polygons or other spatial attributes using a variety of scales. Consequently, this method is not acceptable to the work group.

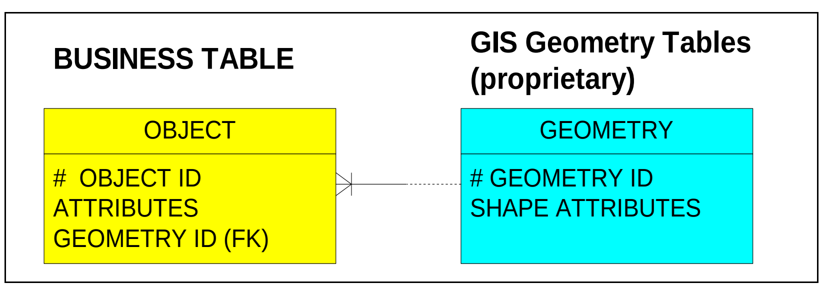


Figure 2: Method 2 was disallowed based on the work group's business requirements.

Method 3

This option (Figure 3) assumes that the user will be able to insert a foreign key from the PPDM table into the Geometry table. While this option allows a business object to have one associated geometry, it does not allow each geometry to be shared by more than one business object. In addition, it assumes that PPDM will be able to define the storage method in the target spatial data store. Consequently, this option was also eliminated.

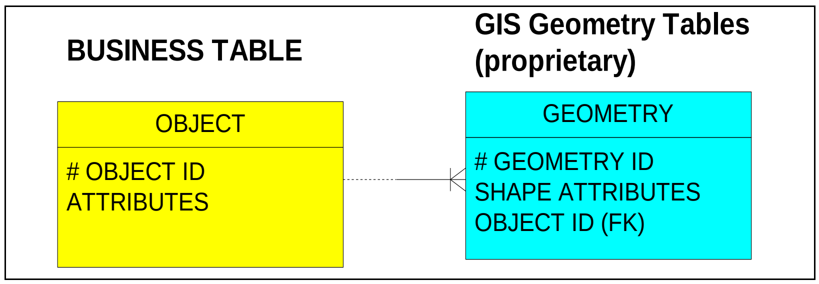


Figure 3: Method 3 was disallowed based on the work group's business requirements.

Requirements Evaluation

The following table shows the key business requirements against which each method was evaluated. Failure to meet one or more of the criteria eliminated it.

Requirement	**Method 1**	**Method 2**	**Method 3**
More than one geometry for a single business object (i.e., more than one scale of polygon for an area).	Yes	*No*	Yes
Support reuse of geometry, if desired.	Yes	Yes	*No*
Good separation between Geometry (GIS) tables (proprietary) and PPDM tables.	Yes	Yes	*Would have to add columns to SDE/SDC tables*.
Extensible solution (user can define additional columns, if desired)	Yes	Yes	Cannot ensure this.
Ease of writing queries	More difficult; add one join to queries	Easy; fewer joins	Queries depend on platform used.
Ease of implementation and maintenance	More flexible, so more complex.	Less difficult	Cannot be standardized; depends on platform.

Figure 4: This table illustrates compliance of each method with critical business requirements. Items highlighted in red and italicized indicate critical requirements that eliminated methods 2 and 3.

Implementation Considerations

For testing purposes, the work group decided to use SDE (ESRI) and SDC (Oracle relational option), as these are widely used in the industry. Next, the work group generated intersection tables as outlined in the Pilot Project Summary (Method 1) for sample points (wells), lines (seismic lines) and polygons (areal blocks). The PPDM tables were populated with business information and relevant coordinate information loaded into the GIS. The intersection table was used to link the two sets of data.

Queries based on common usage were run against each engine to ensure that they would work correctly, retrieve the correct information, and perform well. Detailed performance measurements or comparisons were not conducted during this phase of testing, and are deemed to be beyond the scope of this project.

Documentation of the procedures, scripts, and methods used to populate and query SDE and SDC were created and are available from the PPDM Web site.

The implementation process includes the construction of "intersection tables" as shown in Figure 5. This table outlines the intersection tables to be created, the parent tables containing the required spatial data, and the geometry type for that feature. The geometry type reflects the best topological representation, including point, polyline, or complex polygon, for the spatial data being considered. Some issues regarding the selection of a geometry type are addressed in the FAQ section of this document.

Intersection Table	**Parent Table**	**Populated with Lat/Long values from (options)**	**Geometry Type**
AREA_GEOMETRY (pilot)	AREA	AREA_BOUNDARY AREA (bounding box)	COMPLEX POLYGON
RM_INFO_ITEM_GEOMETRY	RM_INFORMATION_ITEM	RM_INFORMATION_ITEM (bounding box)	COMPLEX POLYGON
LLD_GEOMETRY	LAND LEGAL DESCRIPTION	LAND LEGAL DESCRIPTION (bounding box) LAND_POLYGON_BOUNDARY	COMPLEX POLYGON
MONUMENT_GEOMETRY	MONUMENT	MONUMENT	POINT
SEIS_BIN_GEOMETRY	SEIS_BIN_GRID	SEIS_BIN_POINT (points or perimeter) SEIS_BIN_GRID (bounding box)	POINT
SEIS_LINE_GEOMETRY (pilot. See notes in FAQ section)	SEIS_LINE	SEIS_LINE (first and last points) SEIS_POINT (end and bend points, all points)	POLYLINE
STRAT_FIELD_GEOMETRY	STRAT_FIELD_STATION	STRAT_FIELD_STATION	POINT?
WELL_GEOMETRY	WELL	WELL, WELL NODE	POINT
WELL_DIR_SURVEY_GEOMETRY	WELL_DIR_SURVEY	WELL_DIR_SRVY_STATION	POLYLINE
WELL_NODE_GEOMETRY	WELL_NODE	WELL_NODE	POINT

Figure 5: Intersection tables to be implemented.

INTERSECTION TABLE

OBJECT/ GEOMETRY
OBJECT_ID # GEOMETRY_ID # GEOMETRY_VERSION_OBS_NO --- GEOMETRY_TYPE (ref) SOURCE (ref) GEOMETRY_SOURCE (ref) OBJECT_SOURCE (ref) COORDINATE_SYSTEM_ID (fk) COORDINATE_ACQTN_ID (fk) GEOMETRY_DATE PREFERRED_IND EFFECTIVE_DATE EXPIRED_DATE

Figure 6: Structure of intersection table.

Spatially Enabling with Spatial Data Cartridge (SDC)

Important: Please read the disclaimer at the beginning of this reference guide before continuing.

Build Knowledge

1. Read this reference guide.
2. Use these procedures as a template for your own plans.

   These procedures are provided as examples only.

   This document covers Point, Line, and Polygon types of spatially enabling.
3. Understand all uses of the database by other applications, reports, scripts, etc., that may be affected by spatial enabling of the data. Develop an action plan to review and handle any impact.
4. Review the spatial enabling plan.
   a. Ensure that you understand all the changes that will be made to the database.

   b. Verify that you understand how the changes will affect any software or procedures that currently run against your 35B database.

5. Make sure that you have PPDM3.5b installed in Oracle (the script is written in PL*SQL, so it will only work on an Oracle database).

   If your installation uses another RDBMS, you will need to rewrite the spatial enabling script. If this is the case, we hope that these scripts will give you a good starting point.

6. Refer to Table 1 to see all tables used or altered by the spatial enabling template scripts.

   TABLE 1

   Below are the tables read or written to with the sample scripts:

TABLE NAME	**Oracle User**
AREA	PPDM35B
AREA_BOUNDRY	PPDM35B
AREA_GEOMETRY	PPDM35B
AREA_POLYGON	PPDM35B
SEIS_LINE	PPDM35B
SEIS_LINE_GEOMETRY	PPDM35B
SEIS_POINT	PPDM35B
WELL	PPDM35B
WELL_GEOMETRY	PPDM35B
WELL_NODE	PPDM35B
AREA_GEOMETRY_SDODIM	SDC
AREA_GEOMETRY_SDOGEOM	SDC
AREA_GEOMETRY_SDOINDEX	SDC
AREA_GEOMETRY_SDOLAYER	SDC
SEIS_LINE_GEOMETRY_SDODIM	SDC
SEIS_LINE_GEOMETRY_SDOGEOM	SDC
SEIS_LINE_GEOMETRY_SDOINDEX	SDC
SEIS_LINE_GEOMETRY_SDOLAYER	SDC
WELL_GEOMETRY_SDODIM	SDC
WELL_GEOMETRY_SDOGEOM	SDC
WELL_GEOMETRY_SDOINDEX	SDC

WELL_GEOMETRY_SDOLAYER	SDC

Set Up Spatial Enabling Environment

Ideally, you should not test these procedures against your (only) production copy of the database.

1. Good data management practices should be followed at all times.
   a. Create a test environment to experiment in, to make sure you can control the impact of the results.
   b. Do a complete back-up of your database, so you can recover in event of a problem.
2. Ensure that the database is set up for:
   - Sufficient table space to create and populate tables, as listed in the Technical Details section. The size for the tables depends on how much data is loaded into the tables for processing.
   - Sufficient system space (rollback, initial, and next extents). Again the size needed depends on how much data is loaded into the tables being processed.
3. Capture information about your current database.

   Other special tuning or support installations (such as stored procedures, etc.) could be affected by the spatial enabling.
4. Prepare Oracle.

   An account should have already been created in Oracle for the SDC data. If you have not already done so, create one now using the directions provided by your vendor. This document uses the characters *SDC* to refer to this account name.

   This SDC account will need select, insert, and reference privileges into the PPDM3.5 data account.
5. Decide what type of geometry you want information for (point, line, polygon); then, decide which of the spatially enabled tables in PPDM you want to use.

Modify the Spatial Enabling Template Scripts

The spatial enabling scripts for SDO populate each of the three geometry types in PPDM (points, complex lines, and complex polygons). One set of scripts is provided for each geometry type (there are two scripts for each set).

The Points scripts use the WELL table as an example.

The Complex Lines scripts use the SEIS_LINE table as an example

The Complex Polygon scripts use the AREA Module as an example.

Important: It is your responsibility to make sure the script will do what you need it to do.

Make the changes you require in order to have the script work correctly for each module you spatially enable.

Points (Well)

There are two scripts to spatially enable the point geometries. These scripts are called CR_POINT.SQL and LD_POINT.SQL.

The first script, CR_POINT.SQL, was designed to create the tables (see list below) used to spatially enable the Well section of the data.

Tables Created:

- WELL_GEOMETRY_SDOLAYER
- WELL_GEOMETRY_SDOGEOM
- WELL_GEOMETRY_SDODIM
- WELL_GEOMETRY_SDOINDEX

The tables are prefixed with the name of the spatial table being enabled, in this case WELL_GEOMETRY.

To enable another point candidate, change all references of WELL_GEOMETRY to the name of the table to be spatially enabled. For example, if you were to spatially enable the Monument section of the data, change WELL_GEOMETRY to MONUMENT_GEOMETRY.

Also in this script is the text "*CHANGE ME".* This text gets loaded into the WELL_GEOMETRY_SDOLAYER table and should be changed to describe the type of data held within the table. An example would be to change it to 'Well Geometry for Northern Alberta' (up to 50 characters in length). It is to your advantage to be as descriptive as possible. Do not remove the single quotation marks, only the text.

The second script, LD_POINT.SQL, script was designed to populate the following tables:

OWNER	TABLE NAME
• SDC	WELL_GEOMETRY_SDOGEOM
• PPDM35B	WELL_GEOMETRY
• SDC	WELL_GEOMETRY_SDOINDEX

This script assumes PPDM35B is the account name that houses the PPDM data. There are three occurrences of PPDM35B in the script. Make sure you change all three to the account name that houses your data.

On or about line 48, you will see the following line of code:

```
v_point.source_document, v_point.well_numeric_id,'POINT', 'Y', 'PPDM',
```

Change ‘PPDM’ (this is the source column) to the source of the data (probably your own company name).

Lines (Seis_line)

There are two scripts to spatially enable complex line geometries. These scripts are called CR_LINE.SQL and LD_LINE.SQL.

The first script, CR_LINE.SQL, was designed to create the tables (see list below) used to spatially enable Seismic Lines.

Tables Created:

- SEIS_LINE_GEOMETRY_SDOLAYER
- SEIS_LINE_GEOMETRY_SDOGEOM
- SEIS_LINE_GEOMETRY_SDODIM
- SEIS_LINE_GEOMETRY_SDOINDEX

The tables are prefixed with name of the table being spatially enabled, in this case SEIS_LINE_GEOMETRY.

To enable another point candidate, change all references of SEIS_LINE_GEOMETRY to the name of the table being spatially enabled.

Also in this script is the text “*CHANGE ME”*. This text gets loaded into the SEIS_LINE_GEOMETRY_SDOLAYER table and should be changed to describe the type of data held within the table. An example would be to change it to ‘SEISMIC LINE Geometry for Northern Alberta’ (up to 50 characters in length). It is to your advantage to be as descriptive as possible. Do not remove the single quotation marks, only the text.

The LD_LINE.SQL script was designed to populate the following tables:

OWNER	TABLE NAME
• SDC	SEIS_LINE_GEOMETRY_SDOGEOM
• PPDM35B	SEIS_LINE_GEOMETRY
• SDC	SEIS_LINE_GEOMETRY_SDOINDEX

This script assumes PPDM35B is the account name that houses the PPDM data. There are nine occurrences of PPDM35B in the script. Make sure you change all nine to the account name that houses your data.

On or about line 141 line, you will see the following line of code:

```
'PPDM', &sysuser, SYSDATE, &sysuser, SYSDATE);
```

Change ‘PPDM’ (this is the source of the data) to the name of the company providing the data.

Remember not to remove the single quotes.

Polygon (Named Area)

There are two scripts to spatially enable complex polygon geometries. These scripts are called CR_AREA.SQL and LD_AREA.SQL.

The first script, CR_AREA.SQL, was designed to create the tables (see list below) used to spatially enable the AREA section of the data.

Tables Created:

- AREA_GEOMETRY_SDOLAYER
- AREA_GEOMETRY_SDOGEOM
- AREA_GEOMETRY_SDODIM
- AREA_GEOMETRY_SDOINDEX

The tables are prefixed with name of the table being spatially enabled, in this case AREA_GEOMETRY.

To enable another point candidate, change all references of AREA_GEOMETRY to the name of the table being spatially enabled.

Also in this script is the text “*CHANGE ME”*. This text gets loaded into the AREA_GEOMETRY_SDOLAYER table and should be changed to describe the type of data held within the table. An example would be to change it to ‘Area Geometry for Northern Alberta’ (up to 50 characters in length). It is to your advantage to be as descriptive as possible. Do not remove the single quotation marks, only the text.

The LD_AREA.SQL script was designed to populate the following tables:

ACCOUNT	TABLE NAME
• SDC	AREA_GEOMETRY_SDOGEOM
• PPDM35B	AREA_GEOMETRY
• SDC	AREA_GEOMETRY_SDOINDEX

This script assumes PPDM35B is the account name that houses the PPDM data. There are 13 occurrences of PPDM35B in the script. Make sure you change all 13 to the account name that houses your data.

On or about line 167, you will see the following line of code:

```
v_preferred, 'PPDM', &sysuser, SYSDATE, &sysuser, SYSDATE);
```

Change ‘PPDM’ (this is the data source) to the name of the company that provided the data.

Remember not to remove the single quotes.

Run the Spatial Enable Scripts

1. Make sure you have read the disclaimer and the rest of this guide.
2. Log into the SDC account.
3. Execute the scripts in their paired order:
   - For POINT:

     CR_POINT.SQL

     LD_POINT.SQL
   - For LINE:

     CR_LINE.SQL

     LD_LINE.SQL
   - For POLYGON:

     CR_AREA.SQL

     LD_AREA.SQL
4. Check the tables that were altered and ensure you are satisfied that the spatial enabling worked correctly and that your data was not compromised. If necessary, modify the scripts and start the process over.

Spatially Enabling with Spatial Data Engine (SDE)

In this section, we discuss the six programs needed to spatially enable with SDE and show you how to obtain data from SDE. Note that the PPDM Association will not customize the programs. To learn how to use SDE, visit ESRI’s Web site.

The programs outlined here were written in Pure C. You can modify them in the C or C++ developing environment.

Important: Please read the disclaimer at the beginning of this reference guide before continuing.

Spatially enabling with SDE is accomplished by running the programs provided. PPDM has provided a sample set of scripts for each of the three types of geometries in PPDM. These geometries are points (e.g., WELL_NODE), complex lines (e.g., SEIS_LINE), and complex polygons (NAMED_AREA). The sample scripts include:

Point Geometries:

- loadWellNode.exe
- insertWellNodeGeometry.exe

Complex Line Geometries:

- loadSeisLine.exe
- insertSeisLineGeometry.exe

Complex Polygon Geometries:

- loadAreaBoundary.exe
- insertAreaBoundaryGeometry.exe

The load programs obtain the object ID, latitude, and longitude from the appropriate tables in the PPDM database; create three tables in SDE; and add the data to the SDE tables. These tables are GEO_% (the name is based on the name of the table being spatially enabled), a feature table (named F_%), spatial index table (named S_%). Additional information is stored in the SDE Layers table.

The insert programs populate the PPDM %_GEOMETRY table with data derived from the PPDM business table being spatially enabled (the unique ID, numeric ID, latitude, and longitude) and the unique identifiers from the SDE tables. This allows the %_GEOMETRY table to function as the link between PPDM and SDE. While these programs only populate critical data, it is possible to modify the scripts so that additional information is loaded into the remainder of the table structure.

The load and insert programs are run as pairs. Two scripts are available for each type of geometry to be spatially enabled (points, complex lines, and complex polygons). For example, if you want to spatially enable seismic lines, you would run *loadSeisLine* first. This procedure creates the necessary table structures in SDE. Next, you would run *insertSeisLineGeometry* to populate those tables and the PPDM SEIS_LINE_GEOMETRY table. Similarly, if you are spatially enabling wells, you must run *loadWellNode* first, then *insertWellNodeGeometry*.

Preparations

Before you can run the programs, you must make sure that the geometry source in the PPDM has been populated first. Other preparations are necessary as well.

1. Make sure that SDE Development Client is installed and that you have access to it.
2. Obtain the six programs from PPDM’s Web site (www.ppdm.org).

   To do this, click the link to the Download page, enter your user name and password, and download the zipped file that contains the six programs. Note that the original uncompiled code is also available. You can use this as a starting point for your own customized code.

3. Populate R_GEOMETRY_TYPE with the values POINT, LINE, POLYGON, COMPLEX POLY, and POLYLINE.

   The last two—COMPLEX POLY and POLYLINE—are optional. However, you must populate POINT, LINE, and POLYGON; otherwise, the programs will fail.

4. For each table that can be spatially enabled, make sure that the NUMERIC_ID column is populated with unique numbers.

   The SDE programs will copy the value in the NODE_NUMERIC_ID column into the WELL_NODE_GEOMETRY intersection table. This numeric value can be used to improve query performance in SDE.

Running the Programs

Important: When typing program names at a DOS prompt, be sure to type these on the command line as we have here. These program names are case sensitive.

1. Ensure that the SDE client has been loaded on the machine you will run the scripts on.

   Check the SDE manuals for details about installing and configuring SDE.

2. Make sure that the PPDM database can write to the SDE database account, which, for example, could be named ESRI_SDE.
3. Complete the procedures in one or more of the following sections:
   - Spatially Enabling Seismic Lines
   - Spatially Enabling Wells
   - Spatially Enabling Area Boundaries

   You can complete the sections in any order. For example, you may want to spatially enable wells first. If so, complete that procedure first.

   Remember that you do not have to complete all three sections. If you only want to spatially enable wells, you only have to complete that procedure.

Spatially Enabling Seismic Lines

The loadSeisLine script was developed to load the first and last shotpoints for each seismic line using the point ID, latitude, and longitude values denormalized into the SEIS_LINE table. If you prefer to load all points, or to use point and coordinate information from the SEIS_POINT table, modify the loadSeisLine script accordingly.

1. To run the loadSeisLine.exe program, type **loadSeisLine** at a DOS prompt.

2. Enter the following *six* variables at the DOS prompt in the order shown, making sure that each variable is separated by a space; then, press the Enter key only after typing the last variable:

   In other words, you enter them all on one command line, not on six. The program will show you how to input these variables if you have trouble.

   <server_name> (which is the name of the server where SDE is loaded; e.g., ORACLE1)

   <instance_name> (which is the name the SDE instance is called on that server; e.g., esri_sde)

   <database> (which is the name of the Oracle database; e.g., ORCL)

   <user_name> (which is the user account name within the Oracle database, where the PPDM 3.5 tables are located; e.g., ppdm35)

   <user_password> (which is the password for opening this account; e.g., ppdm35)

   <new_sde_table> (which is the name of the SDE table that will be created. Name this table GEO_XXX, where XXX is the name of the PPDM table to be spatially enabled, such as GEO_SEIS_LINE.)

   If these variables are not accepted, contact your system administrator to resolve the problem, then start over.

   When the variables are accepted, the program begins running. You can watch its progress. The program displays messages indicating that it is loading a line. When the program has finished running, it shows you how many lines it has loaded.

3. After you have run the loadSeisLine program, run the insertSeisLineGeometry.exe program next by typing **insertSeisLineGeometry** at the DOS prompt.

4. Enter the following *six* variables at the DOS prompt in the order shown, making sure that each variable is separated by a space; then, press the Enter key only after typing the last variable:

   <server_name> (which is the name of the server where SDE is loaded; e.g., ORACLE1)

   <instance_name> (which is the name the SDE instance is called on that server; e.g., esri_sde)

   <database> (which is the name of the Oracle database; e.g., ORCL)

   <user_name> (which is the user account name within the Oracle database, where the PPDM 3.5 tables are located; e.g., ppdm35)

   <user_password> (which is the password for opening this account; e.g., ppdm35)

<new_sde_table> (which is the name of the SDE table that will be created. Name this table GEO_XXX, where XXX is the name of the PPDM table to be spatially enabled, such as GEO_SEIS_LINE.)

Spatially Enabling Wells

1. To run the loadWellNode.exe program, type **loadWellNode** at a DOS prompt.
2. Enter the following *six* variables at the DOS prompt in the order shown, making sure that each variable is separated by a space; then, press the Enter key only after typing the last variable:

   In other words, you enter them all on one command line, not on six. The program will show you how to input these variables if you have trouble.

   <server_name> (which is the name of the server where SDE is loaded; e.g., ORACLE1)

   <instance_name> (which is the name the SDE instance is called on that server; e.g., esri_sde)

   <database> (which is the name of the Oracle database; e.g., ORCL)

   <user_name> (which is the user account name within the Oracle database, where the PPDM 3.5 tables are located; e.g., ppdm35)

   <user_password> (which is the password for opening this account; e.g., ppdm35)

   <new_sde_table> (which is the name of the SDE table that will be created. Name this table GEO_XXX, where XXX is the name of the PPDM table to be spatially enabled, such as GEO_WELL_NODE.)

   If these variables are not accepted, contact your system administrator to resolve the problem, then start over.

   When the variables are accepted, the program begins running. You can watch its progress. The program displays messages indicating that it is loading well nodes. When the program has finished running, it shows you how many nodes it has loaded.

3. After you have run the loadWellNode program, run the insertWellNodeGeometry.exe program next by typing **insertWellNodeGeometry** at the DOS prompt.
4. Enter the following *six* variables at the DOS prompt in the order shown, making sure that each variable is separated by a space; then, press the Enter key after typing the last variable:

   <server_name> (which is the name of the server where SDE is loaded; e.g., ORACLE1)

   <instance_name> (which is the name the SDE instance is called on that server; e.g., esri_sde)

<database> (which is the name of the Oracle database; e.g., ORCL)

<user_name> (which is the user account name within the Oracle database, where the PPDM 3.5 tables are located; e.g., ppdm35)

<user_password> (which is the password for opening this account; e.g., ppdm35)

<new_sde_table> (which is the name of the SDE table that will be created. Name this table GEO_XXX, where XXX is the name of the PPDM table to be spatially enabled, such as GEO_WELL_NODE.)

Spatially Enabling Area Polygons

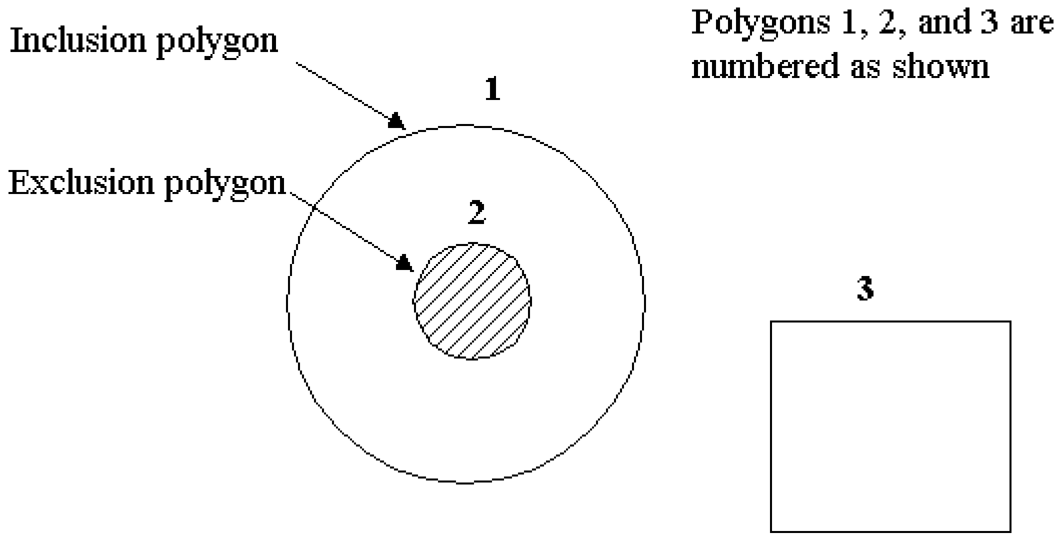
Spatially enabling complex polygons is more complicated than spatially enabling points or lines. Polygons may be separated or may contain "holes" or areas that are not included, as shown in the figure below.

If you are loading a donut-shaped area comprised of an outer circle that represents an inclusion polygon and an inner circle that represents an exclusion polygon, you must perform a few extra steps in PPDM to make sure the data will load properly. The figure below shows a set of three polygons that comprise a single complex polygon. Polygon 1 is an area of inclusion, polygon 2 bounds the area of exclusion, and polygon 3 is separated from the other two.

The complete set of polygons represents a single AREA (perhaps it is an organizational area). Each of the three polygons will be entered into AREA_POLYGON as a row of data (for a total of three rows). It is critical that the rows (in AREA_POLYGON) for the area of inclusion and its area of exclusion be loaded with unique IDs that increment by 1. For example, if Polygon 1 is loaded with POLYGON_ID = 123, then Polygon 2 *must* be loaded with POLYGON_ID = 124.

In each AREA_POLYGON row, the BOUNDARY_DIRECTION column must be specified as COUNTERCLOCKWISE or CLOCKWISE. Polygons that are areas of inclusion should be counterclockwise; areas of exclusion, clockwise. The POLYGON_TYPE column should also be set to INCLUSION or EXCLUSION, as appropriate.

The boundary points for each polygon are stored in the AREA_BOUNDARY table. For the polygon whose boundary represents the outer perimeter of an area of inclusion (Polygons 1 and 3 in the figure), you must enter the points in the AREA_BOUNDARY table in a counterclockwise direction; for the exclusion polygon, enter them clockwise. Also, be sure to close the polygons properly by repeating the first point location as the last point. If you do not close the polygons this way, SDE may have difficulty processing the data correctly.



Note: The loadAreaBoundary program will check the BOUNDARY_DESCRIPTION column and the order of the points for the donut-shaped area represented by polygons 1 and 2. If a BOUNDARY_DIRECTION is specified as counterclockwise, but the points are entered clockwise, the program will correct the order of the points and make them counterclockwise.

1. To run the loadAreaBoundary.exe program, type **loadAreaBoundary** at a DOS prompt.
2. Enter the following *six* variables at the DOS prompt in the order shown, making sure that each variable is separated by a space; then, press the Enter key after typing the last variable.

   In other words, you enter them all on one command line, not on six. The program will show you how to input these variables if you have trouble.

   <server_name> (which is the name of the server where SDE is loaded; e.g., ORACLE1)

   <instance_name> (which is the name the SDE instance is called on that server; e.g., esri_sde)

   <database> (which is the name of the Oracle database; e.g., ORCL)

   <user_name> (which is the user account name within the Oracle database, where the PPDM 3.5 tables are located; e.g., ppdm35)

   <user_password> (which is the password for opening this account; e.g., ppdm35)

<new_sde_table> (which is the name of the SDE table that will be created. Name this table GEO_XXX, where XXX is the name of the PPDM table to be spatially enabled, such as GEO_AREA_BOUNDARY.)

If these variables are not accepted, contact your system administrator to resolve the problem, then start over.

When the variables are accepted, the program begins running. You can watch its progress. The program displays messages indicating that it is loading area boundaries. When the program has finished running, it shows you how many area boundaries it has loaded.

3. After you have run the loadAreaBoundary program, run the insertAreaBoundaryGeometry.exe program next by typing **insertAreaBoundaryGeometry** at the DOS prompt.
4. Enter the following *six* variables at the DOS prompt in the order shown, making sure that each variable is separated by a space; then, press the Enter key after typing the last variable:

   <server_name> (which is the name of the server where SDE is loaded; e.g., ORACLE1)

   <instance_name> (which is the name the SDE instance is called on that server; e.g., esri_sde)

   <database> (which is the name of the Oracle database; e.g., ORCL)

   <user_name> (which is the user account name within the Oracle database, where the PPDM 3.5 tables are located; e.g., ppdm35)

   <user_password> (which is the password for opening this account; e.g., ppdm35)

   <new_sde_table> (which is the name of the SDE table that will be created. Name this table GEO_XXX, where XXX is the name of the PPDM table to be spatially enabled, such as GEO_AREA_BOUNDARY.)

Frequently Asked Questions (FAQ)

SEIS_LINE_GEOMETRY:

We can choose to implement seismic lines to support use as COMPLEX LINES or as POINTS. Following detailed discussions with Leo Bynum (testing for SDE) and David Warren (testing for SDC), we propose that we implement as POLYLINE because it best supports the following business cases:

Business Need	POINT Method	LINE Method
Does this line cross a polygon or area of interest, even though none of the individual points actually fall in the area of interest?	Not supported	Yes
What is the closest shotpoint to the well? In SDE, you must first find all the lines within a predefined radius (SDE does not determine what this radius should be, so you have to decide), then query using $x^2 + y^2 = z^2$ point by point.	Yes, but need coded support	Yes, if you have created a line with many nodes using SEIS POINT. Need coded support
What is the total length of this line?	Yes, only if the SPATIAL_SEQ_NO has been populated and made available to the query.	Yes

Figure 7: Comparison of point and line methods in support of business needs for seismic lines.

Appendix A: Internet Resources

World Wide Web

PPDM Association

http://www.ppdm.org

Oracle Corporation

http://www.oracle.com

Oracle Technology Network

http://technet.oracle.com

Oracle Canada Corporation

http://www.oraclecanada.com

ESRI Corporation

http://www.esri.com

Open GIS Consortium

http://www.opengis.org

SQL Standards Home Page

http://www.jcc.com/sql_stnd.html

Discussion Groups

comp.databases.oracle

comp.soft-sys.gis.esri

comp.infosystems.gis

References

Curtis, Trudy, "*Spatially Enabling the Public Petroleum Data Model*", February 2000.

Curtis, Trudy et al, "*PPDM Version 3.5 Spatial Enabling: Business Requirements*", May 1999.

Curtis, Trudy et al, "*PPDM Version 3.x Spatially Enabling PPDM :Work Group Charter*", July 1999.

Bynum, Leo and Raad, Mansour, "*SDE From Design to Deployment*", 1998. Presented at UC98.

Warren, David, "*Oracle8 and Spatial Data*", Presentation.

Spatial Workgroup, "*Developing PPDM Standards for Spatial Petroleum Data*", January 1999.

Oracle Spatial User's Guide and Reference, Release 8.1.6, Oracle Corporation.

Oracle 7 Spatial Data Option User's Guide and Reference, Oracle Corporation.

Oracle 7 Spatial Data Option Application Developer's Guide, Oracle Corporation.

Oracle 8 Spatial Cartridge User's Guide and Reference, Release 8.0.5, Oracle Corporation.

Oracle 8i Spatial User's Guide and Reference, Release 8.1.5, Oracle Corporation.

SDE C Developer's Guide, ESRI Corporation

Index

CR_AREA.SQL
- using in spatially enabling with SDC, 18

CR_LINE.SQL
- using in spatially enabling with SDC, 17

CR_POINT.SQL
- using in spatially enabling with SDC, 16

GIS
- discussed, 5

GIS tools
- use of, 7

intersection tables
- implementing, 12

LD_AREA.SQL
- using in spatially enabling with SDC, 18
- using to spatially enable with SDC, 19

LD_LINE.SQL
- using in spatially enabling with SDC, 17, 18

LD_POINT.SQL
- using in spatially enabling with SDC, 16, 17

OGC
- scope and mandate, 6

PPDM and GIS
- combining, 8
- methods of combining, 9

scope of project
- defined, 6

SDC
- defined, 13
- modifying the template scripts, 16
- setting up the environment, 15
- spatially enabling lines, 17
- spatially enabling points (wells), 16
- spatially enabling polygons, 18
- used in testing, 12

SDC scripts
- running, 19
- tables created by, 16, 17, 18, 19
- tables read or written to, 14

SDE
- considerations for loading polygons, 25
- defined, 20
- prerequisites for spatially enabling with, 21
- programs needed for spatially enabling, 20
- running programs for, 21, 22
- spatially enabling polygons, 25
- spatially enabling seismic lines, 22
- spatially enabling wells, 23
- used in testing, 12
- using insert programs, 21
- using load programs, 20

SEIS_LINE_GEOMETRY
- questions about, 28

spatial and attribute data
- managing, 8

spatial data
- use of, 7
- used in PPDM, 7

spatial engine
- tying spatial data to, 7

spatially enabling with SDC
- procedure for, 13

testing

methods used, 12
using PPDM
in spatial analysis, 5