OpenSpirit Developer's Kit

OVERVIEW

The OpenSpirit Developer's Kit provides a rich set of distributed objects and services that make the development of your application quicker. An application that uses the OpenSpirit dev kit may access data from any OpenSpirit enabled datastore anywhere on your network.

GENERIC OBJECTS

- Session – provides the scope of collaboration for OpenSpirit enabled applications.
- Project Set – a named collection of vendor projects that may be used as a virtual project
- User Profile – manages user preferences

DATA OBJECTS

OpenSpirit has implemented a set of business objects that present a consistent object interface regardless of what datastore the underlying data is stored in. The OpenSpirit Subsurface Data Module provides implementations of these objects for several common project databases. The following list highlights some of these high-level objects. Additional data types are continuously being added.

- Project – a vendor project
- Wellbore List – a named collection of wellbores
- Well – well header information
- Wellbore – wellbore header and well bore path– may be accessed in time or depth
- Well Log – single and multidimensional logs– may be accessed in measured depth, TVD, or time
- Well Pick – may be accessed in measured depth, TVD, or time
- Well Velocity – well time-depth table (used to perform well time-depth conversion)
- Drilling Target – geometric shapes representing targets associated with a well plan
- Earth Model – a named collection of consistent horizons and faults
- Horizon and Horizon Properties – 2D grid representation of geologic horizons and associated properties
- Fault – fault "stick" polyline representation of geologic fault
- 2D Seismic Line and Dataset – post-stack 2D seismic data and line geometry
- 3D Seismic Survey and Dataset (Volume) post-stack 3D seismic data and survey geometry–may be accessed via random point, trace, planer, or sub-volume accessors

EVENTS

The following represent some of the key events that help coordinate the actions of applications connected to the same OpenSpirit session.

- Data Change – sent when a data object is updated, normally includes a bounding box to define spatial extent of change
- Object Selection – contains object references of data objects selected by a user
- Data Selection – contains data keys of data objects selected by a user
- Area of Interest (AOI) – contains polygon defining area of interest
- Point of Interest (POI) – contains the x,y,z coordinate for a selected point
- Cursor Location – contains the x,y,z coordinate of cursor position that is being "tracked"
- GIS Feature Selection – contains named point, polyline, or polygon features selected by a user

OBJECT SERVICES

These are some of the more important services that are used by OpenSpirit data servers included in the Subsurface Data Module and are also available directly to client applications.

- **Units** – over 800 predefined units and transformations (based on POSC reference sets)
- **Coordinate** – over 2300 predefined coordinate systems and transformations that supports
  - geodetic datum shifts
  - map projection
  - custom coordinate systems
  - local seismic bin-grid and ijk coordinate systems
- **Datastore Descriptor** – meta-data about vendor datastores and projects contained therein
- **Reference Value** – catalog of reference values for different data sources and mappings between them

(continued on back)

OpenSpirit
THE INTEGRATION COMPANY

OpenSpirit Developer's Kit

OBJECT SERVICES *(continued)*

- **Query** – federated query service to support data queries of two types:
  - An object query which returns an array of object references
  - An attribute query which returns an array of selected attributes

  The query syntax is based on a subset of SQL 92 that includes basic spatial queries.
- **User Alias** – service to map User IDs across operating systems to allow PC users to launch UNIX data servers under a different User ID
- **Notification** – a cross platform messaging service that is used for OpenSpirit events

```
int main(int argc, char **argv) {
    // Connect to the OpenSpirit services
  osp_Connection openspirit;
  openspirit.connect("Developer");
  // Obtain a session
  osp_Session session = osp_Session::getSession("MyFavoriteSession");
  // Create and execute a query object for querying osp_WellBore objects.
  osp_EntityQuery query(session, osp_WellBore::getEntityType());
  query.setWhereClause("well.field = 'Wrigley' AND well.currentOperator = 'BIG OIL'");
  query.execute();
  // Get the resulting well bores
  osp_WellBoreArray bores = osp_WellBore::narrow(query.getEntityResult());
  // Get the first well bore returned
  osp_WellBore bore = bores[0];
  // Print out the well bore's identifier
  cout << "Bore identifier=" << bore.getIdentifier().getValue() << endl;
  // Get all the picks associated with the well bore and print their names
  osp_WellPickArray picks = bore.getPicks();
  for (unsigned int i = 0; i < picks.size(); i++) {
    cout << "Pick name=" << picks[i].getPickName() << endl;
  }
  // Get all the logs associated with the well bore and print their kind
  osp_WellLogTraceArray logs = bore.getLogTraces();
  for (i = 0; i < picks.size(); i++) {
    cout << "Log kind=" << logs[i].getTraceKind() << endl;
  }
  // Get the well bore's geometry in WGS 84 coordinates
  osp_CoordPointArray points = bore.getPath(osp_CoordSystem::getWGS84()).points;
  // Print out the path
  for (i = 0; i < points.size(); i++) {
    cout << points[i].toString() << endl;
  }
  // Disconnect from OpenSpirit
  openspirit.disconnect();
}
```

	Windows	Solaris	Linux	Irix
Java	✓	✓	✓	✓
C++	✓	✓	✓	✓
C#, C++, VB.NET (.NET DLL)	✓			
VB, VBA, …(COM DLL)	✓			

SUPPORTED LANGUAGES AND OPERATING SYSTEMS

Client applications may be developed using any of the following languages on the indicated platforms. Note: C++ source code for the client libraries is provided along with make files to allow developers to build their own libraries in the event the pre-complied libraries are not compiled with your preferred options.

DOCUMENTATION

A developer's guide, reference documentation, and over 60 example client applications are included with the dev kit. Examples are provided in:

- Java
- C++ (Windows, Linux, Solaris, and Irix)
- C# (using .NET DLL on Windows)
- VBA (using COM DLL on Windows)

In addition, the source code is provided for the OpenSpirit well, section, and 3D Viewers that illustrates the use of the OpenSpirit libraries in conjunction with INT's JGeoToolkit graphics libraries.

EXAMPLE APPLICATION

This simple C++ client application illustrates the ease of using OpenSpirit. This example gets a pre-existing session for a user, and then issues a query to find all wellbores that satisfy some criteria and then prints out information about the selected wellbore, logs, and picks.

OpenSpirit
THE INTEGRATION COMPANY