DWGdirect™

Object-oriented C++ library for DWG data access and vectorization.

Features

- Full access to all data in a DWG file.
- Ability to create custom objects and modules.
- Support for Architectural Desktop (ADT) custom objects (currently in beta).
- Powerful data manipulation functionality.
- Full vectorization support.

Supported Platforms

- Windows (32-bit, 64-bit, CE)
- Macintosh
- SGI
- Solaris
- HP-UX
- IBM AIX
- Linux

DWG Version Support

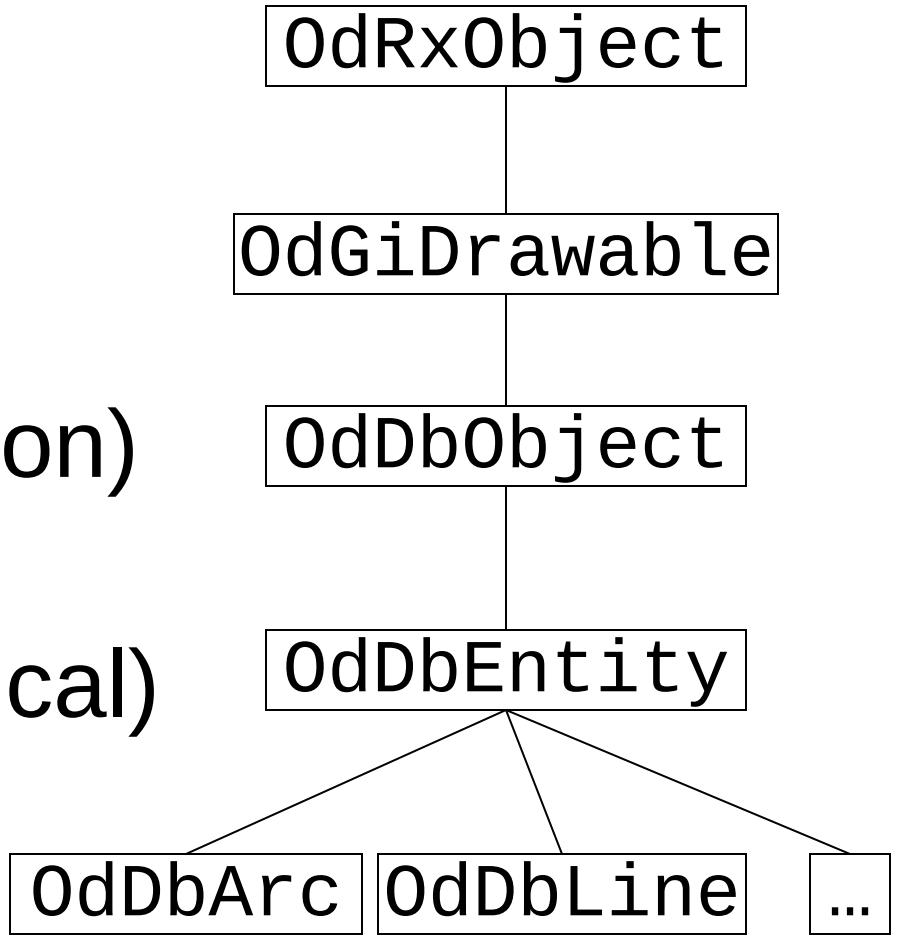
- Read support for R2.5 through R2005.
- Write support for R12 through R2005.
- Full read/write support for “round-trip” data.
- Complex entity conversion during save.

DWGdirect Packages

- Root – Core library functionality.
- Db – Database class (OdDb*)
- Ge – Geometry classes (OdGe*).
- Gi – Obect vectorization (OdGi*).
- Gs – High level vectorization (OdGs*).
- Br – B-Rep access (OdBr*).
- ModelerGeometry – ACIS support.

Database Object Hierarchy

- RTTI
- world/vp draw
- database (common)
- database (graphical)
- entity specific



DWGdirect Initialization

```
// Custom OdDbSystemServices object
OdStaticRxObject<MyServices> svcs;

odInitialize(&svcs);
// invoke DWGdirect client code here
odUninitialize();
```

File Loading/Saving

```
OdDbDatabasePtr pDb =
   svcs.readFile(“a.dwg”);

OdRxObjectImpl<OdWrFileBuf> fb;
fb.open(“out.dwg”);

pDb->writeFile(&fb, OdDb::kDwg,
                    OdDb::vAC18, false);
```

Damaged File Recovery

```
OdDbAuditInfo aiInfo;

OdDbDatabasePtr pDb =
  svcs.recoverFile(svcs.createFile(“a.dwg),
                                  &aiInfo);

// Recover info (warnings, errors, etc.) is
// sent to aiInfo object.
```

Data Integrity, Audit/Repair

- Data access functions validate property data during object “set” methods.
- Comprehensive audit/repair functionality validates object level data, as well as inter-object relationships.
- Corrupt file generation due to user error is significantly reduced.

Opening and Closing Objects

```
//Objects must be explicitly opened before they
//can be accessed.

OdDbBlockTableRecordPtr pMs =
  pDb->getModelSpaceId().safeOpenObject();

// pMs will be open for read-only access, until
// smart pointer pMs goes out of scope, or until
// it is explicitly released
```

Notifications

- Support for transient and persistent reactors attached to database objects (modified, copied, closed, etc. events).
- OdDbDatabaseReactor class provides reactor support for OdDbDatabase operations (object appended, erased, modified, etc.).

Database Manipulation

- **OdDbEntity::explode()** converts complex entities (MTEXT, MLINE, 3DSOLID, etc.), into simpler entities.
- **OdDbDimension::recomputeDimBlock()** creates dimension blocks from settings.
- **OdDbDatabase::purge()** determines if references exist to a passed in set of objects.

Object Cloning

- OdDbDatabase::deepCloneObjects() provides deep cloning support for database objects.
- OdDbDatabase::wblock() provides cloning of entire blocks, or whole databases.

Undo and Transaction Support

- OdDbDatabase functions startTransaction, endTransaction, and abortTransaction allow blocks of editing commands to be applied/aborted as a whole.
- Undo recording allows undo/redo of database modifications.

Paging and Unloading

- Unloading support allows unmodified objects to be automatically unloaded from memory when they are closed (partially loaded database).
- Paging support allow an object to be paged out to a custom storage module, when the object is closed.
- Memory footprint is reduced.

Vectorization Support

- Objects draw themselves via worldDraw and/or viewportDraw virtual functions.
- Flexible API allowing client to handle either high level or low level geometry.
- Thumbnail image generation.

Protocol Extension

```
OdDbEntityPtr pEnt = id.safeOpenObject();
//Retrieve the registered protocol
// extension object for this object type.
OdSmartPtr<OdDbEntity_Dumper>
  pEntDumper = pEnt;

pEntDumper->doSomething();
```

ACIS Support

- Optional ACIS module provides read/write support for relevant SAT/SAB versions.
- Full conversion support between SAT & SAB.
- Mesh generation.
- B-Rep traversal via OdBr* classes.
- Creation of simple solids.

Custom Applications

- Clients can create custom “DRX” applications that support custom database objects derived from OdDbObject and OdDbEntity.
- Custom objects implement their own overrides for draw and other virtual methods.
- Custom modules are loaded on-demand by DWGdirect.

Custom Commands

- Custom DRX applications can create and register custom commands, that are available globally within a DWGdirect client application.
- Commands can query the client application for data.

ADTdirect

- ADTdirect is a set of DRX modules providing support for Architectural Desktop custom objects.
- Full read support for all ADT objects.
- Full rendering support for AecDbWall, AecDbDoor, AecDbWindow, AecDbSlab, AecDbRoof, AecDbOpening, and others.
- Creation support for AecDbWall, AecDbSlab, and some others.

Import/Export Modules

- DWF import.
- DWF export.
- SVG export.

DST Support

- Full read/write support for 2005 Sheet Set (.DST) files.
- Data extraction and editing capabilities via OdSm* classes.