

uDig – User-friendly Desktop Internet GIS Final Report



Submitted To: Program Manager
GeoConnections
Victoria, BC, Canada

Submitted By: Jody Garnett
Refractions Research Inc.
Suite 400 - 1207 Douglas Street
Victoria, BC V8W 2E7
E-mail: jgarnett@refractions.net
Phone: (250) 383-3022
Fax: (250) 383-2140



TABLE OF CONTENTS

1	EXECUTIVE SUMMARY	3
2	INTRODUCTION	4
3	PROJECT OUTCOMES.....	5
3.1	COMPARISON TO OBJECTIVES	6
3.1.1	<i>Use as Viewer and Basic Data Editor.....</i>	<i>6</i>
3.1.2	<i>Use as a GIS Platform.....</i>	<i>7</i>
3.1.3	<i>Broader GIS Community.....</i>	<i>7</i>
3.1.4	<i>Revenue</i>	<i>8</i>
3.2	TECHNICAL DISCUSSION	9
3.2.1	<i>Platform Architecture.....</i>	<i>9</i>
3.2.2	<i>Anatomy of a Plug-in</i>	<i>10</i>
3.2.3	<i>Catalog API.....</i>	<i>12</i>
3.2.4	<i>Of further technical interest.....</i>	<i>14</i>
3.3	BENEFITS REALIZED.....	15
3.3.1	<i>Benefits to Canadians.....</i>	<i>15</i>
3.3.2	<i>Benefits to the Project Team.....</i>	<i>16</i>
3.3.3	<i>Benefits to Open Source Projects.....</i>	<i>17</i>
4	RELATED COMMERCIAL ACTIVITIES	19

1 EXECUTIVE SUMMARY

The goal of the User-friendly Desktop Internet GIS (uDig) project was to create an open-source desktop GIS application, to make viewing, editing, and printing data from CGDI and local data sources simple for ordinary computer users.

The project partners included Refrations Research, the Open Planning Project (TOPP), the BC Ministry of Sustainable Resource Management (MSRM), and GeoInnovations. Refrations, TOPP, and MSRM provided programming and spatial domain expertise as contributions in kind, and GeoInnovations provided funding.

The results of the project were very positive: we have delivered a series of well-received releases. uDig is released as "open source" software, which means that other software developers can look at the internals, make changes if they wish, and redistribute the software as they wish.

Our company is now very experienced with Eclipse technologies, OGC specifications and the open source java development community. We are effectively using this expertise in our consulting business.

The products of our project provide technical benefits to many CGDI members:

- By enhancing the GeoTools and JUMP libraries and making them more widely usable by CGDI members.
- CGDI members, without any proprietary licensing mechanisms, can freely deploy the open-source uDig application.
- Supported versions of the *uDig* application and training / installation services will be available from Refrations, using materials developed in part by this project.
- Free access to OpenGIS compatible technologies like *uDig* will help promote the CGDI standards for Canada's spatial data infrastructure.

In general, the entire technical community within CGDI can benefit, because the results of the project have been released as open source software. We feel this project was very positive for ourselves, and for the CGDI community, and look forward to seeing our work deployed by CGDI network members.

2 INTRODUCTION

Open-source components are a critical part of the CGDI vision, because they allow organizations to deploy infrastructure widely, in a distributed fashion, without incurring multiple licensing fees. Open-source components are also the most tractable for fast support of new OpenGIS interoperability standards.

There were already many different pieces of open-source software that implement OpenGIS server standards: Mapserver implements WMS, GeoServer implements WMS and WFS-T, PostGIS implements SFSQL, DeeGree implements WMS and WFS, and so on. However, there was not a single piece of desktop software capable of binding information from all these servers together into a unified desktop view. *uDig* is the open-source application which will bring CGDI data sources to the desktop, and integrate them with local data sources for standard business processes – data viewing, data editing, and data printing.

3 PROJECT OUTCOMES

This project has resulted in the release of uDig version 1.0, which includes the following capabilities:

- WFS client read/write support, to allow direct editing of data exposed via Transactional Web Feature Servers (WFS-T).
- WMS support, to allow viewing of data published via a web map server.
- Styled Layer Descriptor (SLD) support, to allow standardized persistence of layer displays.
- Web Catalog Server support, for quick location of available CGDI layers.
- Printing support, to allow users to create standard and large format cartography from their desktops using CGDI data sources.
- Standard GIS file format support, to allow users to directly open, overlay, and edit local Shape and GML files with CGDI online data.
- Coordinate projection support, to transparently integrate remote layers in the client application where necessary.
- Database access support, to allow users to directly open, overlay and edit data stored in PostGIS, OracleSpatial, ArcSDE, and MySQL.
- Cross-platform support, using Java as an implementation language, and providing one-click setup files for Windows, Linux and Solaris.
- Multi-lingual design, allowing easy internationalization of the interface, with French and English translations of the interface completed initially.
- Customizability and modularity, to allow third party developers to add new capabilities, or strip out existing capabilities as necessary when integrating the application with existing enterprise infrastructures.

3.1 Comparison to Objectives

The objective of this project was to build a user-friendly desktop internet GIS application as an open-source product, with source code freely licensed under the GNU General Public License (GPL).

3.1.1 Use as Viewer and Basic Data Editor

The application must meet the needs of ordinary GIS users, as a viewer and basic data editor:

- Use a simple and consistent user interface paradigm
- View and edit data from different internet sources (WMS, WFS)
- View and edit data from different local data sources (files, databases)
- Integrate all data sources into a single view and projection on the fly
- Provide an interactive experience, even when using slower internet data sources
- Have a printing capability, to put internet and local data onto a paper map
- Work on all major desktop platforms (Windows, OS/X, Linux)

We have succeeded in producing an impressive desktop application making use of the intended information in a friendly and consistent manner. We are pleased with the performance and quality of data access and projection.

The client is cross platform between Windows and Linux; we are waiting on the next release of OS/X before supporting that platform.

3.1.2 Use as a GIS Platform

In addition, the application framework should be extensible by third party developers. It must be easy to add or remove functionality from the application or to embed components of the application into another system. The application must:

- Have a plug-in interface to allow open-source or proprietary third-party functionality to be added at runtime.
- Have a component model that allows portions of the system to be detached from the main interface and used in external applications.
- Have complete source code documentation and a programmer's guide.

We have made use of the Eclipse Rich Client Platform, which provides a micro-kernel on top of which we have constructed a GIS Platform for general GIS development. This infrastructure has an excellent plug-in framework, for which many excellent books, and copious Internet documentation are available.

In addition to providing a framework, an entire "Plug-in Development Environment" is available with extensive tooling support, which we have been able to leverage for GIS development.

In practice, this means the uDig Programmers Guide is integrated into the development environment. Common activities, such as adding a custom tool to the application, are actively assisted with an "Extension Point" editor.

3.1.3 Broader GIS Community

The completed product should have a positive outcome in the geospatial community of practice:

- The tool should be widely adopted for use by GIS generalists, in organizations at all levels of government and industry.
- The open-source project should attract third party developers and collaborative funders to continue to support development.
- The open-source project should work with the OpenGIS consortium to advance open-source as a means of promoting interoperability goals.

We have already seen significant interest in uDig – we have found early adopters across a wide variety of industry, government and non-profit sectors. We have received word of several open-source projects starting up with the use of uDig in mind. The final uDig release already includes work contributed by community involvement.

Over the course of development we have followed with interest several OGC initiatives with mixed results. While our contributions to GeoAPI and the Go-1 working group have been positive, we spent a lot of time working against the Catalog Service 2.0 specification with little other than experience to show for it.

3.1.4 Revenue

The completed product will provide a basis for sales and marketing of:

- Supported versions of open-source spatial infrastructure, WFS servers, WMS servers and spatial databases. With a capable desktop client, integrating internet publishing and operational data management becomes possible for organizations.
- Supported versions of the *uDig* desktop client. For organizations with larger deployments, supported versions with a standard upgrade cycle and guaranteed testing methodology will be sold.
- Integration and customization services. For organizations wanting to add spatial data capability to their applications, or add special analytic or display functionality to *uDig*.

The above goals represent a progression; we are already working with customers on the integration and customization of *uDig*. Our intention is to maintain these relationships with continued support during deployment.

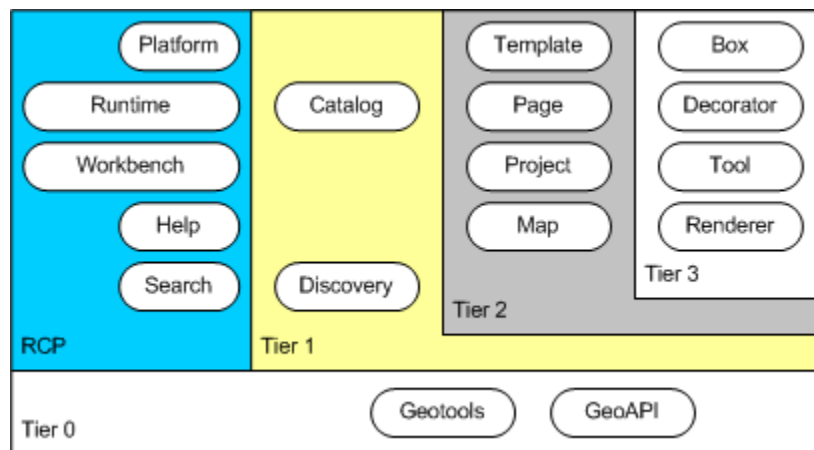
3.2 Technical Discussion

The Java GIS community is already host to several open-source toolkits of great capability. These toolkits provide a great number of services, and allow software developers to communicate with a wide variety of spatial information.

The uDig framework goes the extra mile, providing a well-structured application for one to extend. A software developer can focus on tasks needed for a specific problem. The framework can handle the finding, connecting and display of spatial information. This leaves developers free to concentrate on the actual problem at hand.

3.2.1 Platform Architecture

The uDig GIS platform is built around the concept of plug-ins to the base Eclipse Rich Client Platform. Several well-known Eclipse plug-ins are used, such as the Eclipse Modeling Framework and the Graphic Modeling Framework.



It is useful to consider the uDig application as a series of tiers, each with different responsibilities and capabilities.

Tier 0: The Toolkits used, such as GeoTools and GeoAPI, provide useful abstractions (such as Features) and data connectivity. These libraries are all gathered into a single plug-in so they may share a classloader.

Tier 1: Catalog Services provide a common API for data access and may serve as a foundation for different GIS applications.

Tier 2: Project services operate as a uDig application providing Maps and Printing under user control.

Tier 3: Rendering and Tool services allow control of display and screen interaction.

3.2.2 Anatomy of a Plug-in

The basic building block of uDig is the concept of a plug-in. Although our mandate was to build an extensible system, our choice of the Eclipse Rich Client Platform has provided us an excellent industrial strength system (the OSGI standard originally targeted for cell-phones).

The plugin.xml file is the bare bones definition of a plug-in. This xml file is read by the eclipse runtime and is used as a manifest. It contains all the dependency information, and connections needed for the plug-in.

```
<plugin
  id="net.refractions.udig.tool.info"
  name="%plugin.name"
  version="0.8.0"
  provider-name="Jesse Eichar, Refractions Research"
  class="net.refractions.udig.tool.info.InfoPlugin">
  <runtime>
    <library name="info.jar">
      <export name="*" />
    </library>
  </runtime>
  <requires>
    <import plugin="org.eclipse.ui" />
    <import plugin="org.eclipse.core.runtime" />
    <import plugin="net.refractions.udig.project" />
    <import plugin="net.refractions.udig.project.ui" />
  </requires>
  ...
</plugin>
```

The plugin.xml file provides a series of xml fragments describing its services (known as extensions). These extensions are against well-known schemas provided by the RCP Platform or the GIS Platform.

```
<extension
  point="net.refractions.udig.project.ui.tool">
  <modalTool
    cursor="default"
    class="net.refractions.udig.tool.info.InfoTool"
    tooltip="%info.tooltip"
    icon="icons/etooll16/info_mode.gif"
    onToolbar="true"
    categoryId="information"
    name="%info.name"
    id="net.refractions.udig.tool.info.infoMode">
    <cursor
      hotspotY="5"
      hotspotX="5"
      image="icons/pointer/info_source.png" />
    </modalTool>
  </extension>
```

As you can see, the xml fragment contains enough information to construct a user interface, all without loading any java classes until absolutely needed.

Careful inspection of the previous example will reveal two mysteries: “%info.name” and “%info.tooltip”.

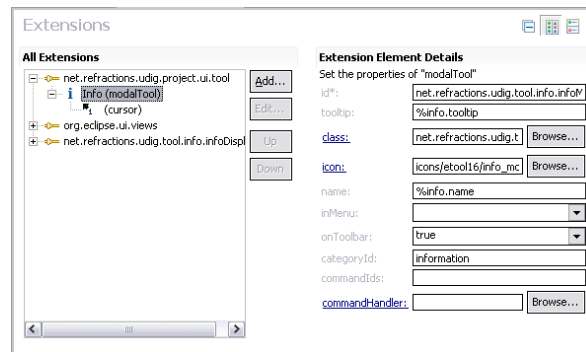
As you may have guessed, this is where internationalization starts to enter the picture. The file plugin.properties contains the required definitions, and additional language packs can provide values for any text, or icons mentioned.

When it comes time to actually write the above tool, a lot of work is done for you. In addition to creating an outline, the GIS Platform makes common use of “context” classes, and super classes, to provide you with exactly the information useful for the task at hand.

```
public class InfoTool extends AbstractModalTool implements ModalTool {
    public InfoTool2( int targets ) {
        super(MOUSE);
    }
    public void mouseReleased(MapMouseEvent e) throws Exception {
        List list = context.getInfo(e.getPoint() );
        int x = e.getPoint().x;
        int y = e.getPoint().y;
        InfoView infoView = (InfoView)
            context.showView( "net.refractions.udig.tool.info.internal.infoView" );
        infoView.setInformation(new InfoList(list, x, y, context.getMap()));
    }
}
```

The code example acquires information for a specific point, uses the runtime Platform to locate the “infoView”, and updates the information for display.

For most GIS development the overhead associated with the framework is low; the handling of xml files is made accessible through the use of a user interface.



The Plug-in framework easily scales to hundreds of contributions (provided everybody follows the rules). With these kinds of numbers in mind there are a few more players to introduce:

- Feature – a grouping of plug-ins for a shared responsibility
- Fragment – an overlay for a plug-in providing platform, or language specific functionality
- Activity – a user level grouping of Features into something understandable (like printing support)
- Site – a packaging of feature content for use with automatic updating

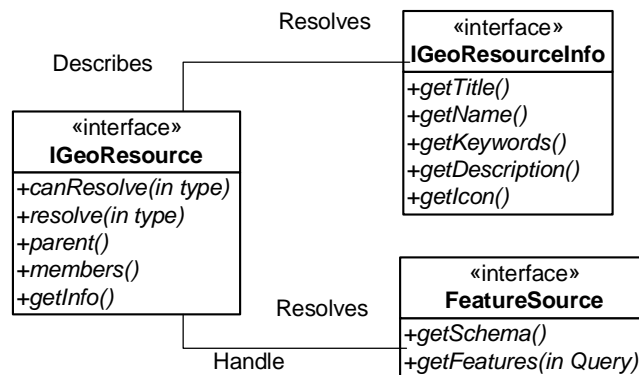
3.2.3 Catalog API

We will briefly mention the central point to having a GIS client – access to information.

The Catalog API is split between three abstractions:

- ICatalog – catalog of spatial resources; both local and remote catalogs are available.
- IService – represents an external service such as a Web Feature Service or individual shapefile.
- IGeoResource – actual spatial information such as a Web Map Server layer, or the contents of a shapefile.

As an example we will use the most specific abstraction: IGeoResource represents real information, the kind you can display on screen or work with.



IGeoResource is implemented as a “handle”, or a promissory note, which you can turn into the information you want at a later time. More to the motivation, the handle can be used in a thread where the entire application won’t be blocked.

The other interesting thing that an IGeoResource can do is “bridge” to an IGeoResourceInfo object containing metadata (such as title and description).

There are two other things to pay careful attention to:

- The IGeoResource is resolved into the requested information. This is a blocking operation as we are a client application and resources are often located remotely.
- The IGeoResource resolved into the requested information. This is an example of an “extensible interface”. Client code can provide additional resolve targets. This allows functionality to be added (such as statistics) without disturbing the core API shared with the entire application.

The Catalog API represents a succinct and powerful abstraction for remote data access.

3.2.4 Of further technical interest

This project has produced a great body of interesting work, all of which is available through the project website at <http://udig.refractions.net>.

Of special note is:

- A whitepaper on the GML schema parser. This represents a significant step forward for the handling of GML in the open source community. Expect to hear more things about this parser as it is picked up by other projects.
- Use of Eclipse Modeling Framework for rapid evolution of the uDig model to meet requirements. This is an example of model driven design, in which the compile-edit-debug cycle has been expanded to include modeling of the problem at hand.
- Use of the Graphical Editing Framework for page layout and template definition.
- Use of Java 5 language features at the API level, for a smaller and more explicit definition.

3.3 Benefits Realized

3.3.1 Benefits to Canadians

The *uDig* project provides benefits to Canadians in several important ways.

- By enhancing the GeoTools, GeoAPI and JTS libraries and making them more widely usable by CGDI members.
- By testing compliance of many OWS providers, and providing test cases and patches where applicable (to projects such as GeoServer).
- The open-source *uDig* application can be freely deployed by anyone, without any proprietary licensing mechanisms.
- Supported versions of the *uDig* application and training / installation services will be available from Refrations, using materials developed in part by this project.
- Free access to OpenGIS compatible technologies like *uDig* will help promote the CGDI standards for Canada's spatial data infrastructure.

3.3.2 Benefits to the Project Team

3.3.2.1 Refractions Research

This project has been an excellent opportunity to undertake a pure research and development exercise, of a large scope, and to study problems that are within our realm of professional expertise.

The *uDig* project has resulted in important marketable products and services:

- Refractions will offer for sale a supported version of *uDig*, with one-click installation on Windows, Linux, and OS/X. The supported version will include user documentation, annual versioned updates, and quality assurance guarantees.
- Refractions will offer consulting services in customizing *uDig*, either integrating it into business applications, or adding spatial processing capabilities for tasks such as data conflation and automatic data processing.
- Refractions will offer consulting services in creating vertically integrated spatial data infrastructures, using the *uDig* and related open-source components, such as GeoServer, Mapserver and PostGIS/PostgreSQL.

3.3.2.2 The Open Planning Project

The *uDig* project adds more features to GeoServer, like a WFS client connector (allowing cascading WFS features), coordinate re-projection support and quality control of the code base. These new features improve the tool and further the goals of The Open Planning Project.

The Open Planning Project wanted to see new features added to the GeoServer core in order to make GeoServer more attractive and usable by other organizations. Like CGDI, they want interoperable systems in place to allow easy collaboration between organizations, and integration of different data from different sources.

3.3.2.3 British Columbia Ministry of Sustainable Resource Management

MSRM plans to make increasing use of online data updates from resource companies and various levels of government in order to maintain corporate databases. Where appropriate, access to these data sets via an interactive tool set will allow clients to directly apply changes to the database using the WFS-T OpenGIS standard. *uDig*, or components of *uDig*, may be used to provide access to WFS-T interfaces by outside parties.

3.3.3 Benefits to Open Source Projects

The Project Team has synchronized its development efforts with several dynamic open source communities: GeoAPI, GeoServer and GeoTools.

3.3.3.1 *GeoAPI*

The benefits realized by the GeoAPI project have been substantial both in terms of review and code contributions.

- Collaboration on a shared feature model for toolkit interoperability
- Research into Metadata representation and Catalog support
- Production of a Grid Coverage API based on the Grid Coverage Exchange specification

3.3.3.2 *GeoServer*

The benefits realized by the GeoServer project have been substantial both in terms of documentation and code contributions:

- WMS client code for use in test cases
- Improved rendering and SLD handling for use by both projects
- Opportunity for the development of a standalone configuration application
- Substantive development support

We enjoy an ongoing relationship with the GeoServer project and will be playing off each other in several marketing initiatives.

3.3.3.3 *GeoTools*

GeoTools has received many benefits:

- A Grid Coverage Exchange implementation
- Improvements to Parameter API
- An SAX based GML parser based on executing XML schema information into real objects using a stack computer.
- Coordinate Reference System definitions and parsing support
- The WMS client code is being hosted as a GeoTools module.
- The WFS client code is being hosted as a GeoTools module.
- Improvements to the Feature Type model

By way of explanation, most code that does heavy lifting or is experimental in nature has been pioneered in the GeoTools code base in order to have access to quality feedback as early as possible in the development cycle.

This project has also resulted in the addition of a few more Refrations staff to the GeoTools Project as Module Maintainers.

4 RELATED COMMERCIAL ACTIVITIES

Refractions Research is an open source spatial services company, and requires expert knowledge and qualifications to backstop our value as a consulting company to agencies implementing open source solutions. In addition to producing a useful product and programming tools, this project has added to our corporate store of knowledge.

The product of this project, the uDig software, is a key part of our marketing efforts, to bring open source spatial solutions to governments and organizations across North America.