WikiGraph

Mark Jordan, Jeremy Lenz, Robert McClure,
Austin Nakamura, Michael Rush, Khanh Tran,
Thomas Van Doren

Beta Release

Draft 1
22 February 2011

WikiGraph

Mark Jordan, Jeremy Lenz, Robert McClure, Austin Nakamura, Michael Rush, Khanh Tran, Thomas Van Doren

CSE 403 - CSRocks Inc.

Version	Primary Author(s)	Description of Version	Date Completed
1	Thomas Van Doren	Add initial information about build, source, and intro	2011-02-21
2	Thomas Van Doren	Add pattern information, intro, known issues	2011-20-22
3			

WikiGraph
Mark Jordan, Jeremy Lenz, Robert McClure, Austin Nakamura, Michael Rush, Khanh Tran, Thomas Van Doren

Introduction

The beta release is live, and the WikiGraph team couldn't be happier! Direct your browser to:

http://wikigraph.cs.washington.edu/

This release includes many features, even some that were not supposed to be ready until the final release. Graphs are displayed for general searches, autocomplete helps user pinpoint the phrase they are looking for, the ability to resize and support for a wide variety of screen size allows users with larger screens to have a better experience while still supporting smaller screen sizes, and it all links to the massive 800 million link database instance that the WikiGraph team has running on AWS!

This document contains information about getting started as a developer, building and testing the WikiGraph products, keeping the database up-to-date, and deploying builds to test and release endpoints. Known issues and an introspective on a pattern used to develop the flash client are also included.

With this release, the team has updated the Software Requirement Specification and its System Design Specification and Planning Document to accurately represent the current state of progress. The good news is that the team is ahead of schedule!

Source

Getting Started

Clone a copy of the repo from Google Code:

```
hg clone https://wiki-map-uw-cse.googlecode.com/hg/ wiki-map-uw-cse
```

Depending on what you want to do, you may have to install and/or configure your machine.

If you intend to build the flash client, the services, or database dumps, you will need:

- make
- PHP 5.3.5

The frontend flash client requires:

- Flex 4.1.0
- FlashPlayer Projector Content Debugger
- Ant

The services require:

- PHPUnit 3.4

The Developer Notes in wiki provide details on setting up the shared environment.

http://code.google.com/p/wiki-map-uw-cse/wiki/DeveloperNotes

Testing

An in depth description of how to run the unit tests work is availabe on the Build and Test Notes page in the wiki:

http://code.google.com/p/wiki-map-uw-cse/wiki/BuildAndTestNotes

The system testing that is conducted is explained in the WikiGraph SDS. Please see that document for further information.

Services

The services are considered the PHP scripts that accept requests from the flash client, query the database, and return results.These scripts are located in the `services/` directory of the repo. The unit tests for these scripts are in the `services/test/` directory of the repo.

PHPUnit is required to run the unit tests. A valid config file is also required. If a valid config does not exist, the build script can be used to generate a fake one.

The easiest way to run the unit tests is by executing the following command from the repo:

```
make checkapi
```

More detailed information about dependencies and this endpoint is at the Build and Test Notes page.

Client

The client is considered the source code and templates that creates the shockwave flash file and HTML/CSS/JS wrapper. These files are located in the `client/` directory of the repo. The unit tests for this code are in the `client/FlexClient/DrawGraph/test/` directory. To run the tests:

```
make checkclient
```

Please see the Build and Test Notes page for the required configuration to build and test locally (emulate Hudson). To simply build and play the SWFs, FlashDevelop is the easiest to build and immediately view the player. It is even possible to build and run the unit tests in FlashDevelop.

It is also possible to run the following commands from the `client/FlexClient/DrawGraph/` directory:

```
ant test    # compiles TestRunner.swf and runs tests, depends compile
ant compile # compiles WikiGraph.swf, depends on init
ant clean   # remove the output director from DrawGraph
```

That runs the unit tests and compiles the SWFs into `client/FlexClient/DrawGraph/output/bin/` and the test results into `client/FlexClient/DrawGraph/output/report/`. An HTML file is produced to easily view the test results.

Known Issues

Cubist

Cubist is not properly configured to run the flexunit tests that we have written for the frontend flash client. They require a display emulator (Xvnc for example) in order to actually run the tests. As of 6:00pm, Feb-22, 2011, Xvnc was not installed on cubist.

The team has written unit tests, and our build scripts support running them. We expect to have access to Xvnc on cubist sometime this week. CSE Support has been notified and is taking care of the issue.

This means that the automated unit tests for our frontend flash client will not be available until later this week. Our developers will be testing locally and our automated build system will not run unittests for the flash client.

We would like to discuss possible alternatives with you if that seems unreasonable.

Bugs in the client

While the client is much farther along than we originally committed, it does have a few bugs and nuances.

- Dynamic window resizing was causing issues, but this may have been resolved.
- “Page cannot be found” errors are getting thrown more frequently than they should, as a result of odd search ranking from the services.
- Some searches do not result in the search term as the center node (unexpected behavior)
- A loading spiral does not appear during load time so user can continue interacting; this causes the user to reach odd states rarely.
- Parameters sent to the services in queries are not getting escaped quite right which accounts for some of the unexpected page not found errors

Known bugs in the services

The services are working pretty well. They issues we are encountering on that front include performance and data structure.

- Search and autocomplete are case sensitive
- Search does not return a graph for exact matches
- Some searches are returning duplicate items
- The graph size is intentionally throttled to 24 to increase performance

All of these issues we intend to address by the the final release.

Pattern

The WikiGraph team does not have a strict coherence to any large scale pattern (like those in the Gang of Four). However, in the flash client, and in some of the OO PHP scripts, data encapsulation is used to protect the state of objects and encourage appropriate usage. One file that this is particularly noticeable is the Parse class (`client/FlexClient/DrawGraph/src/Parse.as`).

In the parse class, only the four method which actually interact with other entities in the system are public. The helper method is private such that it cannot be used unintentionally by naive (or malicious) programmers. The fields of the Parse class are not accessible outside of the class. By encapsulating these values, Parse has greater control over their lifetime. The internal intermediate state of Parse is inconsequential to outside entities. Therefore, there is no need to allow access to these parameters.

Building WikiGraph

WikiGraph is built on an instance of the continuous integration service, Hudson. When new commits are made, the relevant projects are automatically built. The build include unittesting, compiling the necessary files (i.e. SWFs for the frontend), packaging the needed files into a tar ball and placing them in the build output directory on cubist.

The status of present and past build can be seen at the below url. The Hudson console allows developers to initiate build, see the output of current and past builds, and can report useful information about each build (build time, test results, number of failures, etc).

http://wikigraph.cs.washington.edu/build

Each of the clients can be built individually. They can also be build on a local machine, provided the machine has been setup appropriately.

Please visit the Build and Test Notes page for a detailed description of setting up and running the tests and build. The proceeding sections merely glaze the surface and assume a lot.

http://code.google.com/p/wiki-map-uw-cse/wiki/BuildAndTestNotes

Universal build targets include:

```
make clean
make all  # depends on both services and client
make build  # depends on both services and client
```

Services

To build the services the system must have PHPUnit 3.4, PHP 5.3.5+, a set of valid mysql connection parameters, and make. The following targets are supported:

```
make checkapi
make api
make hudsonapi
make testapi
make hudsontestapi
```

All of the above targets take the optional parameters:

```
[OUTPUT=.] [BUILDTAG=unknown-WikiGraph-unknown]
[CONFIG=<config.php>]
[DBHOST=localhost DBPORT= DBUSER=root DBPASS=
DBNAME=wikigraph LINKURL=http://en.wikipedia.org/wiki/]
```

If you do want to build the services locally, it is beneficial to setup a local instance of the database (see below, or better yet the Build and Test Notes wiki page).

Client

To build the client the system must have make, ant, Flex 4.1.0+, Adobe Flash Player Project Debugger, and the appropriate environment variable and configuration (easier said than done :-)). The following targets are supported:

```
make checkclient
make graph
make hudsongraph
make test
make hudsontest
```

All of the above targets take the optional parameters:

```
[OUTPUT=.] [BUILDTAG=unknown-WikiGraph-unknown]
[CONFIG=<config.php>]
[DBHOST=localhost DBPORT= DBUSER=root DBPASS=
DBNAME=wikigraph LINKURL=http://en.wikipedia.org/wiki/]
```

The CONFIG and DB* options are not used for the client build, but they can be specified nevertheless.

If you have a strong desire to emulate the Hudson build locally, visit the Build and Test Notes page to find out about the appropriate configuration. It is quite easy to build and run WikiGraph.swf and TestRunner.swf in FlashDevelop. The files are equivalent to those produced by the Makefile.

It is also possible to run the following commands from the `client/FlexClient/DrawGraph/` directory:

```
ant test    # compiles TestRunner.swf and runs tests, depends compile
ant compile # compiles WikiGraph.swf, depends on init
ant clean   # remove the output director from DrawGraph
```

That runs the unit tests and compiles the SWFs into `client/FlexClient/DrawGraph/output/bin/` and the test results into `client/FlexClient/DrawGraph/output/report/`. The WikiGraph.swf file is equivalent to the one create in the root Makefile.

Database

Located in the database/ directory of the repo is a Makefile which can download the latest database dump from wikipedia (or elsewhere), parse the abstract files, add the dump to the WikiGraph database, remove unneeded table/rows/columns, and create the appropriate schema for WikiGraph. It works as a first time install of the database or an update to a more recent dump.

See the Build and Test Notes page for detailed setup information. This is an easy way to setup a local instance of the wikigraph database.

Hudson

The WikiGraph team has setup the Hudson Continuous Integration service on cubist (url below). This system automatically polls the repository every three minutes. For each project, it recognizes certain changes as need to rebuild. For example, the WikiGraph-FlashClient project recognizes any change in the client module of the repo and executes a build.

Below is a list of all the Hudson builds and a description.

Project	Description
WikiGraph-FlashClient	Builds the Flash client for production (graph/)
WikiGraph-Services	Builds the Services for production (api/)
WikiGraph-Test-FlashClient	Build the Flash client for testing (test/)
WikiGraph-Test-Services	Builds the Services for testing (test-api/)

Hudson uses the Makefile at the root of the repo to execute builds. All of our project produce jUnit style result document reports, in addition to standard console output. These reports are used to chart successes and failures on Hudson. There are four projects on Hudson

Note: Unit testing for WikiGraph-FlashClient and WikiGraph-Test-FlashClient have been disabled until cubist is upgraded to support the framework.

For a list of commands executed at build time for each project, please visit the Build and Test Notes wiki page.

Deploying WikiGraph

There is a deployment script located in the `bld/` directory name `deploy`. It requests that a user login via ssh to cubist, and then it takes action accordingly.

```
Usage: deploy -b <build-number> <Project-Name>
       deploy suggest <Project-Name>
```

In the first form it deploys the given project to the appropriate location. If a build number is provide, that build is deployed instead of the latest.

In the second form it suggests the most recent builds for the project

This script is still under development and may be subject to bugs. Noteably it has not been tested on the flash client build since cubist cannot support building (well, testing) the flash client.

This information, in an evolving form, can be found on the Deploy Notes wiki page:

http://code.google.com/p/wiki-map-uw-cse/wiki/DeployNotes