

“pivots-extra “ README

“pivots-extra “ README.....	1
I. Introduction	1
II. For Administrators	1
Installation.....	1
Upgrade.....	2
Uninstall.....	2
Configuration	2
The Java Program	2
Local Algorithm Configuration	3
Remote Algorithm Configuration	3
Google Algorithm Configuration.....	4
Scenario-based Configuration Walk-through	4
III. For Developers.....	5
IV. Appendix.....	6
Scheduled improvements	6
Credits.....	6

I. Introduction

This is a helper package for the “Pivots” module. It provides several additional pivot algorithms. Notably, included in this package there is a Java program tuned for large sites.

II. For Administrators

Installation

The module is developed and tested for Drupal 5.x, and only supports MySQL databases at the time being. It requires the “Pivots API” core module.

System requirements: Drupal 5.x and MySQL database.

Here is a brief description of the modules in the package. Please refer to the configuration section for details.

- *Local* module: This module provides the connector to external pivot programs running on the local server. It is especially useful for large scale Drupal site where external programs could provide more functionalities and better performance.
- *Google* Module (temporarily removed): This module uses Google search engine and searches for related web links to items.

- *Remote* module (temporarily removed). The module provides the connector to external pivot programs running on remote servers. It is useful for load balancing and flexibility (in such cases that related items or conversations are located on remote servers).

Upgrade

This release made several major code changes. If you have installed the previous version of the pivots module, please uninstall it first and then install this release.

Uninstall

To uninstall, first uncheck the box in the “Modules” section. Then go to the Uninstall page and select the modules to uninstall. It will automatically clean up the database.

To uninstall manually, first delete records that contains the term “pivots” in the {system} and {variable} database tables, and then drop the tables with names start with “pivots”.

Configuration

The Java Program

The Java program can be found under the folder “java”. Basically, it has two purposes:

- First, it re-implemented most of the pivot algorithms written in PHP code, such as the *Conversation Pivots* algorithm and the *Double Pivots* algorithm. The reason is to avoid PHP timeout and to improve performance.
- Second, it provides additional pivot algorithms based on full-text search engine, Lucene, which does not have a good PHP implementation.

Here is how to use it. First, go to the “java” subfolder, edit the “config.properties” file to use the correct database connection string. And also check your Java JRE version to JRE version 1.5 and above.

Then, open a command line console, and type:

```
> java -jar pivots_local.jar 3
```

where the number “3” is the pivot ID, which can be found in the “Pivots” list page. The Java program then reads the parameters of the pivot, and populates the pivot index table. If, for example, the pivot (ID=3) is a conversation pivot, the Java program will automatically use the corresponding code logic.

Three more comments:

- You can use the option “--reindex” for reindex. For example, “java -jar pivots_local.jar 3 --reindex”
- You can edit the “logging.properties” in the “JRE/lib” subdirectory, and change the setting “.Level=INFO” to “.Level=FINE” for more log details, which will display the progress.

- If your system is Linux/UNIX, you can use “nohup” to run it in background, e.g.:
> nohup java -jar pivots_local.jar 3 &

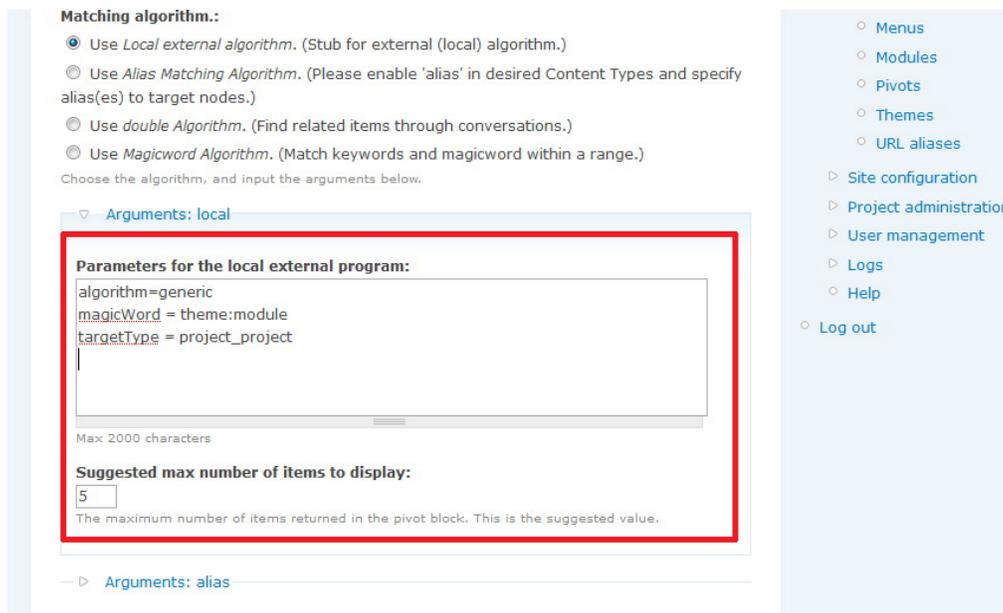
To use the Lucene full-text search engine algorithm, you will need the “Local Algorithm” module below.

Local Algorithm Configuration

This “algorithm” is rather a “program stub” for external pivot programs running on the local server. It provides a GUI interface for external pivot programs such as the Java program discussed above. Below is a brief description of how to configure the Java program. For more information, please contact the developer.

The configuration is simple. Just write “key=value” pairs in the parameters textbox, as shown in the next screenshot. Note that it is case-sensitive. For example:

```
algorithm = generic    # this line tells the Java program which algorithm to use
magicWord = module:theme  # the magicwords
targetType = project_project  # the target content type
```



(Configuration for the external Java program: pivots_local.jar)

Remote Algorithm Configuration

(NOTE: temporarily removed)

This “algorithm” is rather a “program stub” for external pivot programs running on a remote server. The purpose of this module is for load balancing and flexibility where the

content of the pivot block is generated by a remote server. The configuration is beyond the scope of this document. For more details, please contact the developer.

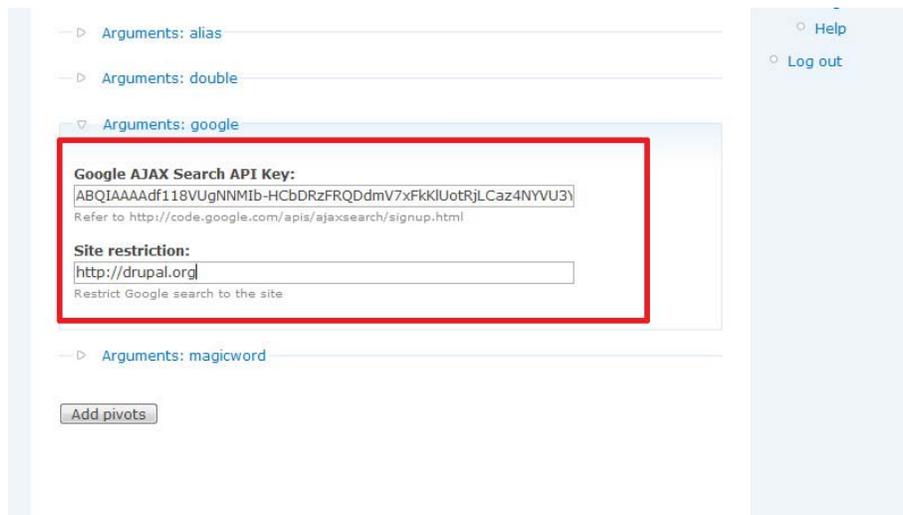
Google Algorithm Configuration

(NOTE: temporarily removed)

This algorithm uses Google search engine and searches for related web links to a node by using the title of the node as probe.

You need to set two parameters, as shown in the next screenshot:

- The API Key: This is required by the Google AJAX Search API. Refer to <http://code.google.com/apis/ajaxsearch/signup.html> for details.
- Site restriction: This can tell Google to search within the site specified here. Otherwise it will search the entire Internet.



(Google Algorithm Configuration)

Scenario-based Configuration Walk-through

This scenario is conjured up for Drupal.org, which has about 70,000 conversation threads. It has a content type “Project” for modules and themes. The infrastructure team wants to enable the pivots module to help users find modules/ themes based on related discussions and other related modules/themes. Since it is a fairly large site, we will use the external Java program rather than the PHP code.

1. Download the “pivots” module and the “pivots-extra” module from Drupal.org, and extract it to the Drupal “site/all/modules” folder.
2. Go to “Administer->Site building->Modules”, find the “Pivots” subsection, and enable the “Pivots API”, “Conversation Pivots”, “Double Pivots” modules.
3. Go to “Administer->Site building->Pivots”, click “Add”

4. In the following page, put “Conversation Pivot” in the “Name” textbox, select “Use conversation pivots algorithm”, click “Arguments: Conversation pivots algorithm” tab, select “project” as the target node type, input “module|theme|project” in the “magicwords” textbox, check “Enable Alias” and “Enable Logging”, and finally click “Add pivots”.
5. (Optional: Go to the projects page, and add alias to modules or themes. For example, you might want to add “cck” as the alias for “Content Construction Kit”, or “tinymce” for “TinyMCE WYSIWYG Editor”.)
6. In the “Pivots” list, verify the pivot ID is 1.
7. Then, click the “Add” tab again to create another pivot.
8. In the following page, put “Double Pivot” in the “Name” textbox, select “Use double pivots algorithm”, click “Arguments: Double pivots algorithm” tab, select “Conversation Pivot” as the “base pivot”, check “Enable logging”, and finally, click “Add pivots”.
9. In the “Pivots” list, verify the “Status” column that the pivot ID is 2.
10. Go to “Administer->Site building->Blocks”, find the two pivots we just created, activate them.

The steps above create two pivots in the Drupal system. And then we will use the external Java program to populate the index table. The pivots will then automatically display the index table data to end users in the pivots block.

1. Open a terminal connection, execute:
`> java -version`
 and make sure the version is higher than 1.5
2. Change to the “pivots-extra/java” subdirectory, and edit the “config.properties file” to use the correct database connection string.
3. (Recommended) Go to the “\$JRE/lib” directory, and edit the “logging.properties” file, change “.level=INFO” to “.level=FINE”.
4. In the “pivots-extra/java” subdirectory, execute:
`> nohup java -jar pivots_local.jar 1 &`
 or, to increase Java VM heap memory size, execute:
`> nohup java -Xms512M -Xmx1024M -jar pivots_local.jar 1 &`
5. If you set the logging.properties, then you can check the “nohup.out” file for progress information. (Note: it may take hours for the program to finish)
6. When it finishes, then we can proceed to the double pivots, by executing:
`> nohup java -jar pivots_local.jar 2 &`
 or,
`> nohup java -Xms512M -Xmx1024M -jar pivots_local.jar 2 &`

For subsequent incremental indexing, we can add the Java commands to the Linux system cron job list. The incremental indexing will not take too much time.

III. For Developers

N/A

IV. Appendix

Scheduled improvements

- Develop similarity search based on Lucene/Solr.
- Add “Google search” module and the “remote” module.

Credits

Daniel Zhou (mrzhou@umich.edu, danithaca@drupal)

Chief developer; PhD student at the School of Information, University of Michigan

Michael Hess (mlhess@umich.edu)

System administrator; technical support; consultant

Paul Resnick (presnick@umich.edu)

Chief architect of the pivots research; advisor; professor at the School of Information, University of Michigan.