# Pivots Module for Drupal
# README

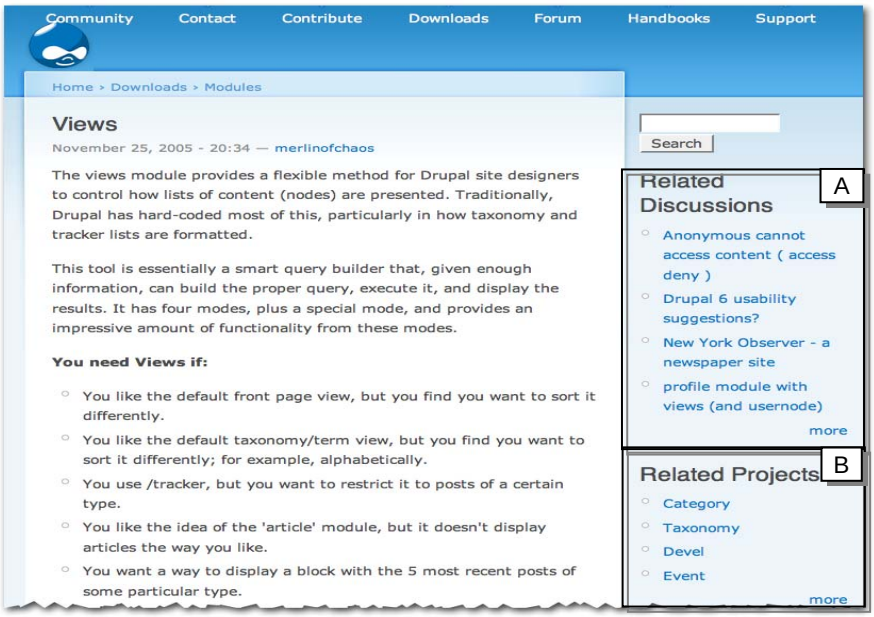


## I. Introduction

The basic idea of the *pivots module* is to help users navigate from one node to other related nodes based on shared characteristics in multiple dimensions. Specifically, conversation pivots allow readers to navigate from nodes to forum conversations about them; double pivots allow readers to navigate from nodes to other nodes mentioned in the same forum conversations. Future release of the module might make use of taxonomy, browse history, or other information in order to find relations between nodes.

The following screenshot illustrates how to use the pivots module in drupal.org to help users find related modules and discussions about the “Views” module.

The project was initiated by the research team at the School of Information, University of Michigan. To see how it fits into a broader research topic, or for more information, please visit http://pivots.cms.si.umich.edu, or contact Daniel Zhou at mrzhou@umich.edu.




**The Drupal.org mockup website with**
**(A) pivot to related conversations, and**
**(B) double-pivot to other modules.**

# II. For Administrators

## *Installation*

The module is developed and tested for Drupal 5.x, and only supports MySQL databases at the time being. To install, just extract the tar.gz file to the "sites/all/modules/" folder, and then enable the modules in the "Modules" section.

> ***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.

- *Pivots* module: This is the core framework module that needs to be enabled first. It doesn't include any pivot algorithm.
- *Magicword* module: This module implements the string matching algorithm. It uses node title with a nearby "magic word" as the probe when searching for related forum threads. Take the "Image" module for example. The algorithm will search text string "image module" in forum threads with "module" as the magic word, so that the general term "image" alone will not be in the results.
- *Alias* module: This module will add an "alias" field to selected content types so that content authors can specify aliases to a node. The algorithm will then use the aliases when searching for related conversation threads. For example, we can add an alias "cck" to the "Content Construction Kits" module so that all conversations that mention "cck" would be detected.

- *Double pivots* module: This module implements the algorithm that finds related nodes mentioned in the same conversations.
- *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.
- *Remote* module. 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).
- *Google* module. This module uses Google search engine and searches for related web links to items.

## 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.

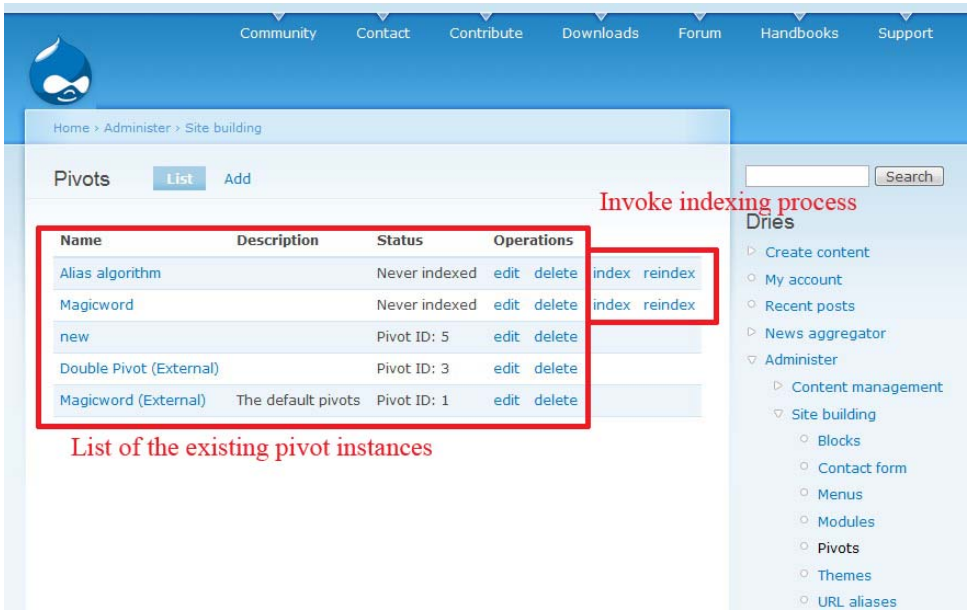
## Uninstall

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*

After enabling the module, you can access the configuration interface at *Administer -> Site Building -> Pivots*. The default page lists all the existing pivots, where you can edit, delete, and invoke indexing process. You can also add a new pivot.

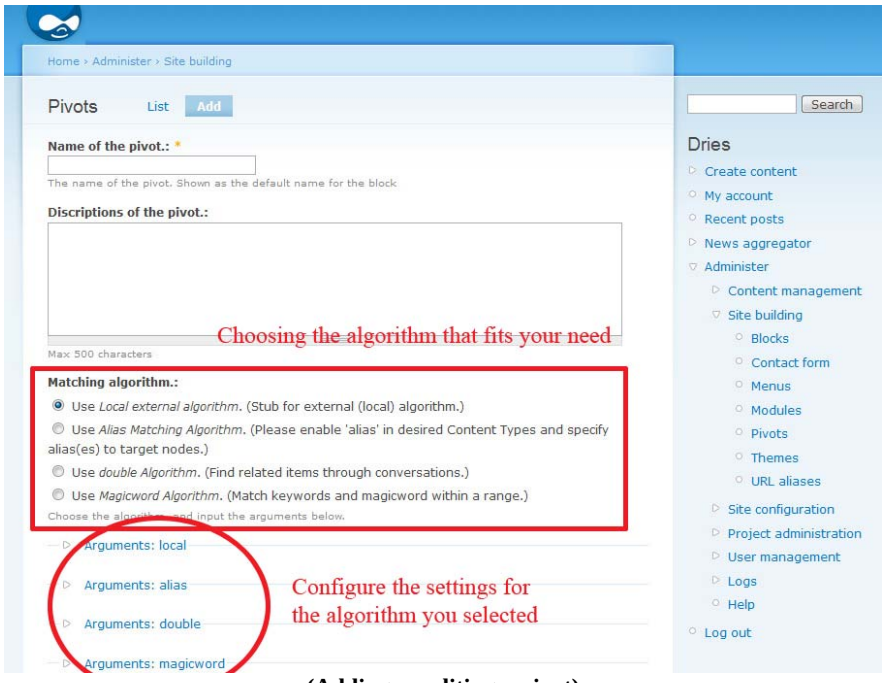


**(The list of existing pivots)**

***Q: Why multiple pivots?***
A: There are multiple pivot algorithms with different parameters for different purposes. This multiple pivots structure provides more flexibility than creating one pivot for the entire site.

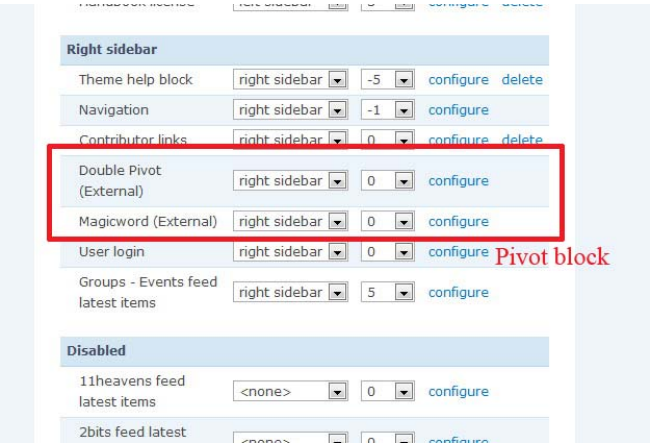
The next screenshot is the interface for adding a new pivot or editing an existing pivot. After specifying the name and description of the pivot, you need to select the algorithm to be used for this pivot. Then, you can set the parameters for the algorithm you selected. The configuration of each algorithm will be covered in the following sections.

***NOTE***: Due to implementation difficulties, the current GUI is not intuitive in that you have to select the algorithm and then set the parameters in separate steps. In future release, we will group the algorithm and its settings together in the same place.




**(Adding or editing a pivot)**

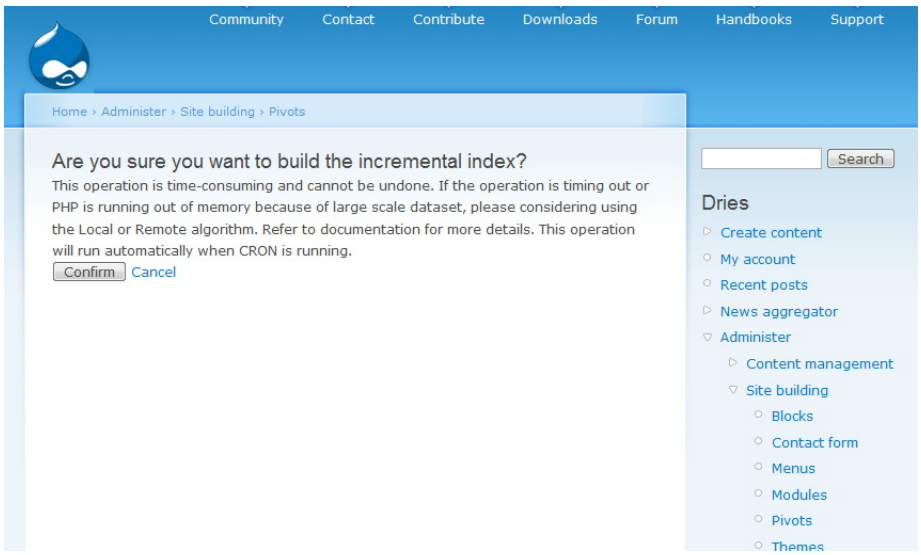
After creating a pivot, a new Drupal block will be listed in the “Blocks” admin page, as shown in the next screenshot. This block is where the pivot displays related nodes to the current node. Therefore, please remember to go to the “Blocks” admin page and activate the block after creating a pivot. If a pivot is deleted, its corresponding block will be deleted automatically.




**(Configure the pivot block in the "Block" section)**

For most pivot algorithms such as the Magicword algorithm, you need to do an additional *indexing* operation. Its job is to pre-compute relations between nodes and save them to a database table that can be retrieved later for faster runtime access. The indexing process is incremental – unchanged nodes will not be indexed again between indexing cycles. To build index, you can press "index" in the "Pivots" page, or, it will run automatically in the CRON job. You can also use "reindex" to fully rebuild the index table.

Please be aware that there are two problems with indexing: 1) latency problem, where new nodes will not get into the pivot block immediately; 2. PHP timeout problem, which is discussed in the sidebar below.




**(Confirm page about building index, which might cause PHP timeout.)**

***IMPORTANT: About PHP timeout during indexing.***

For Drupal site with large dataset, the indexing process might not finish before PHP timeout. Basically, there are three approaches to solve the problem:

First, by following the example of the "Search" module, a pivot can index a limited number of nodes in a short batch that will not timeout, and then repeat it until all nodes get indexed. This approach is ***not*** implemented in the current release for the following reasons:

1). It will impose extra admin overhead (e.g. configuring cron) which is non-trivial for small site admins.
2). It doesn't have good scalability/performance.
3). It's not applicable for algorithms that don't support incremental index, e.g., Lucene/Solr-based algorithm, or some double pivots algorithms.

However, this approach might be implemented in future release if there is a need to do so.

Second, the module provides an identical Java implementation to the PHP implementation of the pivot algorithm. It can read the pivot settings from the database, do the same calculation, and generate the index table for the corresponding pivot, and it doesn't have the timeout problem. (***Note***: the Java code was not thoroughly tested, and thus not included in the current package. For details, please contact the developer.)

Finally, you can use external pivot programs running either locally or on a remote server. Such programs might be written in Java or other programming languages. They need the "Local" helper module or the "Remote" helper module to import results into Drupal. This package includes such an external program. Please refer to the "Local Algorithm Configuration" section for details.

Next, we will discuss how to set the parameters for each algorithm.

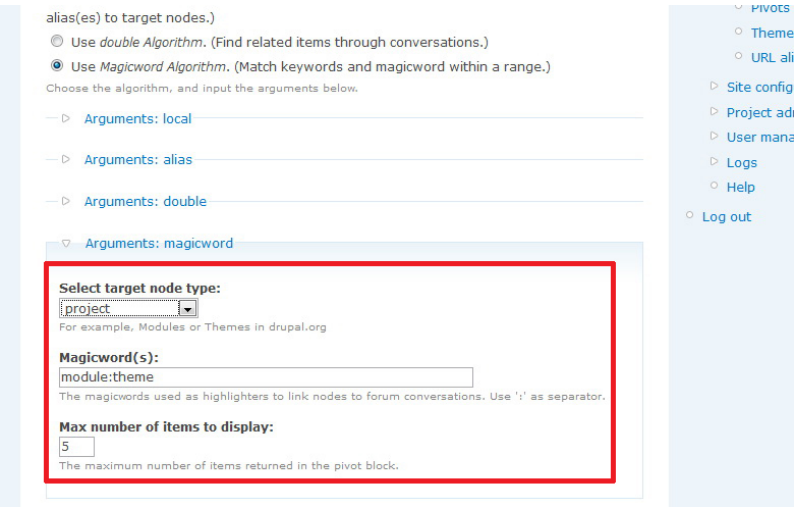
## Magicword Algorithm Configuration

This is a string matching algorithm that uses node title with a nearby "magic word" as probe when searching for related forum threads. Take the "Image" module for example. The algorithm will search text string "image module" in forum threads with "module" as the magic word, so that the general term "image" alone will not be included in the results.

You need to set three parameters:

- Target node type: For which content type you want the pivot to find related conversations. For example, in drupal.org, this could be the "project".
- Magicword(s): The *magicword*, when appeared near the node title string in a conversation thread, is used as an indicator for a match. Use the colon ":" as separator for multiple magicwords.

- Max number of items to display: To limit the number of items to display in the pivot block. This parameter is commonly used in other algorithms.




**(Magicword algorithm configuration)**

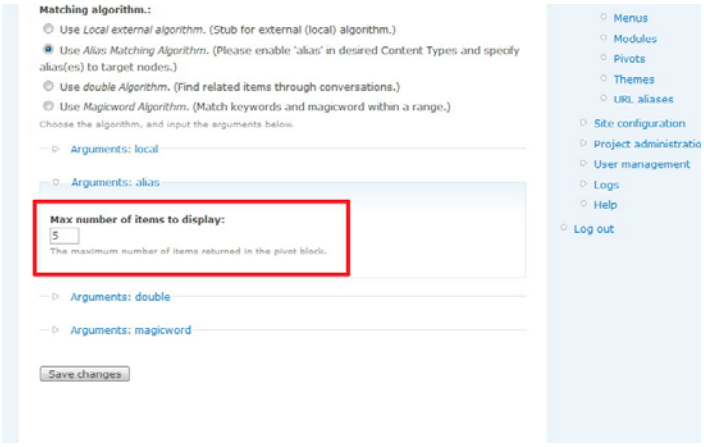
***NOTE:***

1. For large Drupal site, this test string search algorithm is not efficient. Future release will include algorithms based on full-text search engine like Lucene/Solr.
2. In future release, this algorithm might be merged with the Alias algorithm for better results.

## Alias Algorithm Configuration

This algorithm takes into account the alias(es) of a node. For example, the “Content Construction Kits” module is usually referred to as “cck” in conversations rather than the full module title. By specifying the alias, all conversations that mention “cck” would be detected. This module will add an “alias” field to selected content types so that content authors can specify aliases to a node. The algorithm will then use the aliases when searching for related conversation threads.

The only parameter for the Alias-algorithm pivot is the “Max number of items to display”, shown in the next screenshot.

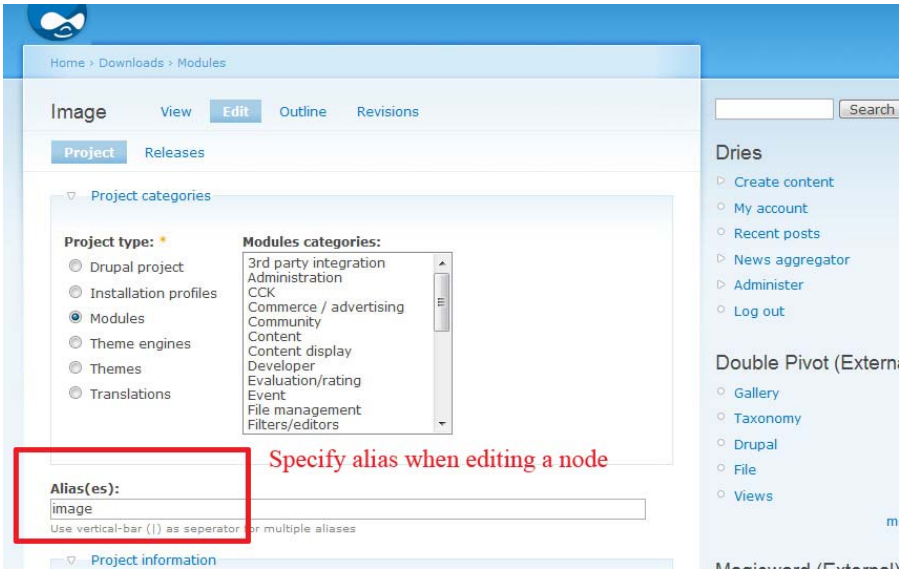



**(Alias Algorithm Configuration)**

However, you need to go to the Content Type Management page, and activate the "alias" support for desired content types, as show in the next screenshot.


**(Enable alias support for a content type)**

Then, you, the Drupal administrator, or the node author(s), need to specify alias(es) for the node, as shown in the next screenshot.



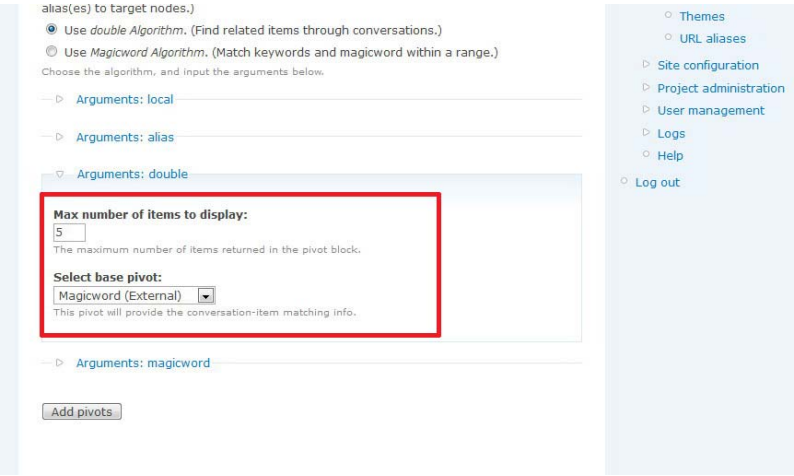

**(Specify alias(es) when editing a node)**

***NOTE:*** The Alias Algorithm is quite important in some cases. For example, in drupal.org, popular modules like "Content Construction Kits" or "TinyMCE WYSIWYG Editor" will seldom get matched in any conversations without using the alias. Therefore, in future release this algorithm might be merged with the Magicword algorithm.

## Double Pivots Algorithm Configuration

This algorithm finds related nodes based on the information that they are mentioned in the same conversations.

This algorithm needs information on item-conversation matching, which can be provided by another pivot. You can set the parameter "base pivot" to tell the algorithm which "item-conversation matching" results to use, as show in the next screenshot.




**(Double pivots algorithm configuration)**

## Local Algorithm Configuration

This "algorithm" is rather a "program stub" for external pivot programs running on the local server. This release includes such an external program written in Java, which can be found in the "java" subfold of this package. Below is a brief description of how to configure the Java program. For more information, please contact the developer.

First, you need to write the parameters in the text box in the form of "key=value" pairs, 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
```

Then, go to the "java" subfolder, edit the "config.properties" file to use the correct database connection string. Also, check your Java JRE version to be JRE 1.5+.

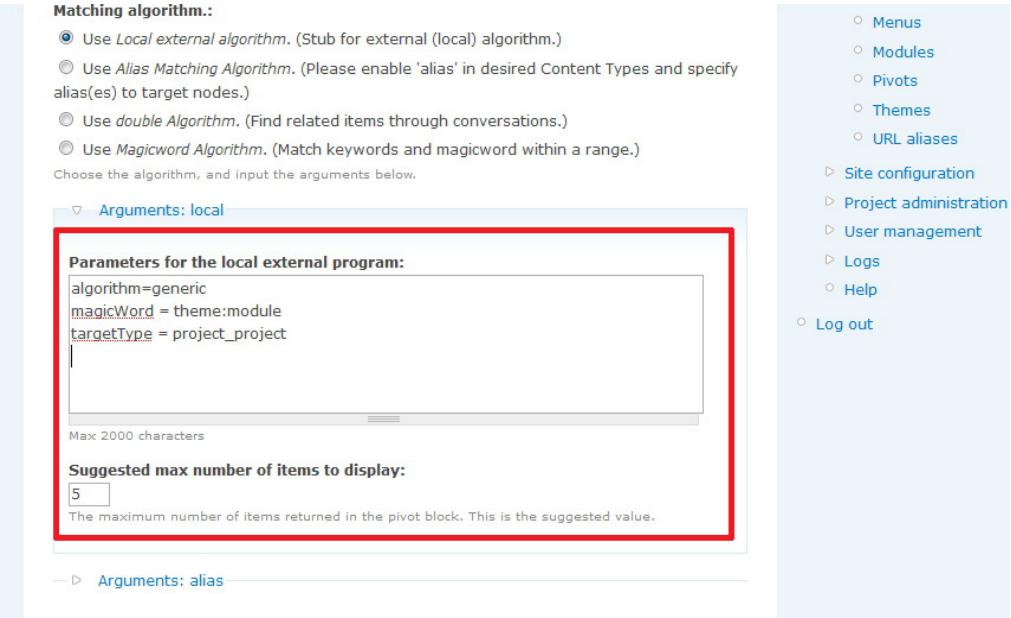
Then, open a command line console, and type:

> java -jar pivots_local.jar 3

where the number “3” is the ID of the pivot you just edited, which can be found in the “Pivots” list page. And the Java program is activated to generate the pivot index table.

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 &




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

## Remote Algorithm Configuration

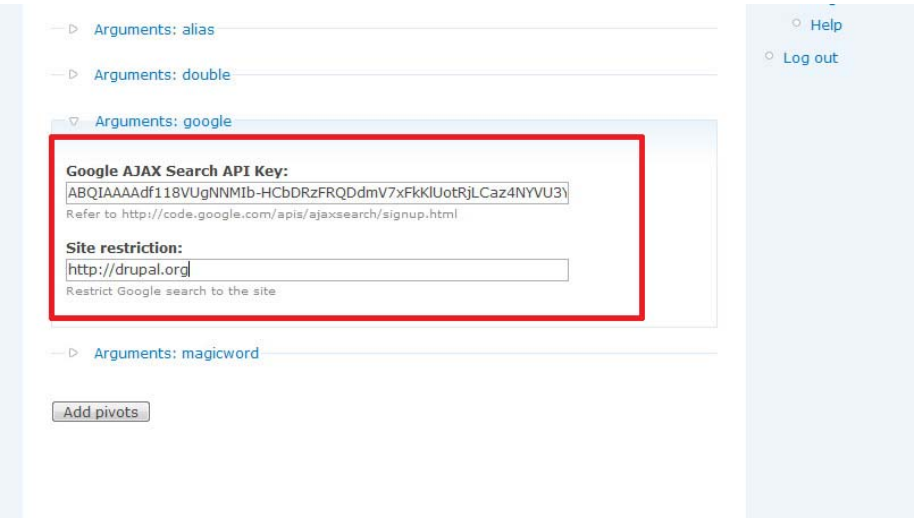
This “algorithm” is rather a “program stub” for external pivot programs running on a remote server. The purpose of this module is for load balancing and flexibility where the content of the pivot block is generated by a remote server. The configuration is beyond the scope of this document. For more details, please contact the developer.

## Google Algorithm Configuration

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: The 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*

### Scenario 1: small scale

This scenario is for a small Reading Club site built on Drupal. Suppose it has a content type "Book", and it enables the "Forum" module for discussions about the books. The webmaster wants to enable the pivots module to display related discussions and related books when a user is on the page of a specific book.

1. Download the pivots 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 "Pivots", "Magicword" and "Double pivot"
3. Go to "Administer->Site building->Pivots", click "Add"
4. In the following page, put "Conversation Pivot" in the "Name" textbox, select "Use Magicword Algorithm", click "Arguments: magicword", select "Book" as the target node type, leave the "Magicwords" textbox blank (because usually the book title is unique, and doesn't need a magicword as indicator), and then, click "Add pivots".
5. In the "Pivots" list page, find the "Conversation Pivot" we just created, and click "index".
6. Then, click the "Add" tab again to create the double pivot.

7. In the following page, put “Related Books” in the “Name” textbox, select “Use double Algorithm”, click “Arguments: double”, select “Conversation Pivot” as the base pivots, and then, click “Add pivots”.
8. In the “Pivots” list page, find the “Related Books” we just created, and click “index”.
9. Go to “Administer->Site building->Blocks”, find the two pivots we just created, select “right sidebar” in the “Region” select box, and click “Save blocks”.

Then, when a user browses a “Book” page, the pivots blocks will be in the right sidebar showing related books and conversations. The index will be updated periodically if CRON is setup correctly. Otherwise, the webmaster needs to click “index” in the “Pivots” list page once in a while.

## Scenario 2: large scale

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.

1. Download the pivots 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” and “External - Local” modules.
3. Go to “Administer->Site building->Pivots”, click “Add”
4. In the following page, put “Conversation Pivot” in the “Name” textbox, select “Use Local external Algorithm”, click “Arguments: local”, in the parameters textbox input:
   algorithm = generic
   magicWord = module:theme
   targetType = project_project
   and then, click “Add pivots”.
5. In the “Pivots” list, verify the “Status” column that the pivot ID is 1.
6. Then, click the “Add” tab again to create another pivot.
7. In the following page, put “Double Pivot” in the “Name” textbox, select “Use Local external Algorithm”, click “Arguments: local”, in the parameters textbox input:
   algorithm = double
   basePivot = 1
   and then, click “Add pivots”.
8. In the “Pivots” list, verify the “Status” column that the pivot ID is 2.
9. 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 generate the pivot index table. The Java program does not have

the PHP timeout problem, and it can take advantage of more sophisticated algorithms in the future release such as similarity search based on Lucene/Solr.

1. Open a terminal connection, execute:
   > java -version
   and make sure the version is higher than 1.5
2. Change to the “pivots/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. Change to the “pivots/java” subdirectory, and 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 &

That’s it. And when a user browses a module or theme, the related discussions or projects will be displayed in the pivots block, helping users find more information. For subsequent incremental indexing, we can add the Java commands to the system cron job list.

# III. For Developers

This section is for those who want to develop their own pivot algorithm based on the pivots module framework, or for those who want to make improvements to existing code. The development process follows the XP (eXetreme Programming) practice, and we are now in a preliminary stage. You might expect to see frequent code refinement in the near future.

## *Files*

- pivots.*: These are the core files for the Pivots module. "pivots.inc" has all the shared functions.
- pivots_magicword.*: These are the files for the Magicword module. It is a good example to see how to implement a pivot algorithm.
- pivots_alias.*: These are the files for the Alias module.
- pivots_local.*: These are the files for the local module.
- pivots_remote.*: These are the files for the remote module.
- pivots_google.*: These are the files for the google module.

- java (folder): this is the folder containing the external java programs, including the source code.

## *Hooks*

To write your own pivots algorithm, you can simply implement the hooks as see fit. Some of the hooks have default implementation if you don't write your own logic. Please refer to the comments in "pivots.inc" for details, and see "pivots_magicword.*" for an example.

The hooks are:

- pivots_hook_args_list(): the list of arguments to be serialized automatically to database..
- pivots_hook_args_serialize(): you can write your own version of arguments serialization functions.
- pivots_hook_args_unserialize():you can write your own version of arguments serialization functions.
- pivots_hook_args_form(): to generate the arguments input form that will be used in the configuration GUI..
- pivots_hook_args_form_handler(): to handle the arguments input from the form.
- pivots_hook_status(): to return the status of the pivot which will be displayed in the "Pivots" list.
- pivots_hook_block_content(): to generate the pivots content in the block.
- pivots_hook_block_title(): to generate the pivots block title.
- pivots_hook_page(): to generate the pivots content in the page; usually invoked when users click "show more" in the pivots block.
- pivots_hook_index(): to generate the pivot index table, incrementally.
- pivots_hook_reindex(): to generate the pivot index table from scratch.

## *Shared functions*

There are some shared functions at the end of the "pivots.inc" file. The *pivots_algorithm_** functions are a set of helper functions for algorithm registration, polymorphism, and so forth. Please refer to the comments in the source file for details.

# IV. Appendix

## *Scheduled improvements*

- Modify the "Select algorithm and set parameters" user interface, group algorithm and its arguments settings together.
- Test the identical Java implementation of the algorithms and include it in the package.
- Develop similarity search based on Lucene/Solr.
- (Optional) Merge the Magicword algorithm and the Alias algorithm.
- (Optional) Develop the PHP incremental index mechanism like the "Search" 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.

## *Change log*

- October 21, 2007: Release a new version with major code revision.
- June 14, 2007: Upload the first version to cvs.drupal.org.