

Synology Download Station Search Module Development Guide

Synology[®]

2011-3-14



Synology Inc.

© 2011 Synology Inc.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Synology Inc., with the following exceptions:

Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Synology's copyright notice.

The Synology logo is a trademark of Synology Inc.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Synology retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Synology-labeled computers.

Every effort has been made to ensure that the information in this document is accurate. Synology is not responsible for typographical errors.

Synology Inc.

3F-3, No. 106, Chang-An W. Rd. Taipei 103, Taiwan

Synology and the Synology logo are trademarks of Synology Inc., registered in the

United States and other countries.

Marvell is registered trademarks of Marvell Semiconductor, Inc. or its subsidiaries in the United States and other countries.

Freescale is registered trademarks of Freescale Semiconductor, Inc. or its subsidiaries in the United States and other countries.

Other products and company names mentioned herein are trademarks of their respective holders.

Even though Synology has reviewed this document, **SYNOLOGY MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY. IN NO EVENT WILL SYNOLOGY BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.**

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Synology dealer, agent, or employee is authorized to make any

modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Table of Contents

Introduction	4
Download Station Search Module	4
DLM Module Content	4
Content of INFO	5
Search.php	5
Stock Plug-in Parsing Helper Methods	6
Pack the DLM module	7
Apply Your Search Module	8
Restricted Runtime Environment.....	10
Minimal Privilege	10
Restricted File Access	10
Document Revision History	11

Introduction

On the Synology DiskStation with DiskStation Manager 3.1 (hereafter DSM 3.1) and onward, 3rd-party developers are allowed to develop the BitTorrent search modules for Download Station to search torrent seeds with external engines. By installing the BitTorrent search modules, users will be able to query and browse the search results on Download Station with customized external search engines.

This document is to provide a guideline to develop the search module of Download Station. Developers are encouraged to have PHP programming experience before reading this document.

Download Station Search Module

DLM Module Content

On DSM 3.1, users are allowed to install the BitTorrent search module onto Download Station with extension of ".dlm", which is the abbreviation of "Download Manager." A dlm module is actually a normal unix tgz or tar.gz archive. For example, if you have a dlm module named **mininova.dlm**, after extracting the module, you will see the following files in the current directory:

```
DiskStation> tar zxf mininova.dlm

DiskStation> ls -l

-rwxr-xr-x  1 root  root           230 Feb  1 11:46 INFO
-rwxr-xr-x  1 root  root           414 Feb  1 11:46 search.php
```

INFO

This is the mandatory module configuration file. It defines the name of the module and how to locate the actual php module to perform searching.

search.php

It's the actual php module which performs search action when being loaded by Download Station.

Content of INFO

The content of INFO is described by a JSON format using UTF-8 encoding and usually looks like below:

```
{
  "name": "mininova",
  "displayname": "mininova",
  "description": "The ultimate BitTorrent source!",
  "version": "1.0",
  "site": "http://www.mininova.org",
  "module": "search.php",
  "type": "search",
  "class": "SynoDLMSearchMininova"
}
```

Table 1. INFO file definition

name	The unique name of the search module among all Download Station modules. If the name was duplicate, the dlm wouldn't be installed.	Mandatory
displayname	Texts to display in Download Station torrent search setting. If displayname was missing, name would be used instead.	Recommended
description	A description to display in Download Station torrent search setting.	Recommended
version	The version number of the search module.	Mandatory
site	The URL of the original torrent search site.	Optional
module	The filename of PHP module that performs search action. Usually It is search.php, but you can use other file names.	Mandatory
type	The module type. Currently only "search" is supported.	Mandatory
class	The unique PHP class name within search.php among all torrent search modules.	Mandatory

Search.php

Search.php is the PHP script that defines a PHP class performing search action. The class should include two mandatory methods which are used to communicate with Download Station search engine. [Table 2. "Mandatory methods in search.php and arguments"](#) explains the details of these two mandatory methods.

The content of search.php looks like below:

```

class SynoDLMSearchMininova {

    private $curl = 'http://www.mininova.org/rss/';

    public function __construct() {

    }

    public function prepare($curl, $query) {

        $url = $this->curl . urlencode($query);

        curl_setopt($curl, CURLOPT_URL, $url);

    }

    public function parse($plugin, $response) {

        return $plugin->addRSSResults($response);

    }

}

```

Table 2. Mandatory methods in search.php and arguments

prepare	<p>\$curl: object An object for cURL connection handle. Download Station uses PHP cURL to perform HTTP request.</p> <p>\$query: string Keyword the user input in Download Station</p>	<p>The method will be called before Download Station search engine performs search. The module should construct a proper HTTP request in this method and use <code>curl_setopt</code> to set an URL for search request. If the module needs to set up additional HTTP headers or use POST for the request, please consult cURL manual for more instructions.</p>
parse	<p>\$plugin: object The plug-in class instance.</p> <p>\$response: string A string variable containing the actual response from server excluding HTTP headers</p> <p>Return value: integer Return the number of results successfully parsed and added of the plug-in.</p>	<p>The method is called after Download Station search engine successfully gets response from the torrent search servers. The module will parse the response either by itself or using stock plug-in helpers.</p>

Stock Plug-in Parsing Helper Methods

Download Station search engine provides some stock helper methods to help module developers to handle several typical response formats, such as RSS and JSON. If the response format can't be parsed using helper methods, module writers can still parse the response manually and add the result one-by-one. Please see [Table 3, "Helper methods to add results into final result set"](#) for detailed helper method explanation.

Table 3. Helper methods to add results into final result set

<p>\$plugin ->addRSSResults</p>	<p>\$response: string The RSS response passed by parse method. Return Value: integer Return the number of results successfully parsed and added.</p>	<p>Description Parse the RSS response and add the parsed results to final result set. Usage \$plugin->addRSSResults(\$response)</p>
<p>\$plugin ->addJsonResults</p>	<p>\$response: string The RSS response passed by parse method \$resultKey: string The JSON response will contain a key that is a list of results. This argument should be the key name of that. \$fieldmapping: array A PHP hash array to map fields from keys within a search result. Available keys are title, download, hash, size, page, date, seeds, leechs and category Return Value: integer Return number of results successfully parsed and added</p>	<p>Description Parse the JSON response and add the parsed results to final result set. Usage \$plugin->addJsonResults(\$response, 'list', array('title' => 'title', 'download' => 'torrentLink', 'hash' => 'hash', 'size' => 'size', 'page' => 'link', 'date' => 'pubDate', 'seeds' => 'seeds', 'leechs' => 'leechs', 'category' => 'category'));</p>
<p>\$plugin ->addResult</p>	<p>\$title: string The torrent title. \$download: string URL of the torrent file. \$size: integer or float The file size of the torrent. \$datetime: string The added time of torrent file in search server with format such as "2010-12-30 13:20:10" \$page: string URL to the page referring this torrent. This page usually contain torrent detailed information \$hash: string The hash value of the torrent. The value could be empty string. \$seeds: integer The number of seeders of this torrent \$leechs: integer The number of leechers of this torrent \$category: string The category of this torrent returned by server</p>	<p>Description Add single parsed result to final result set. Module writer can manually parse the response and add the results one-by-one. Usage \$plugin->addResult(\$title, \$download, \$size, \$datetime, \$page, \$hash, \$seeds, \$leechs, \$category);</p>

Pack the DLM module

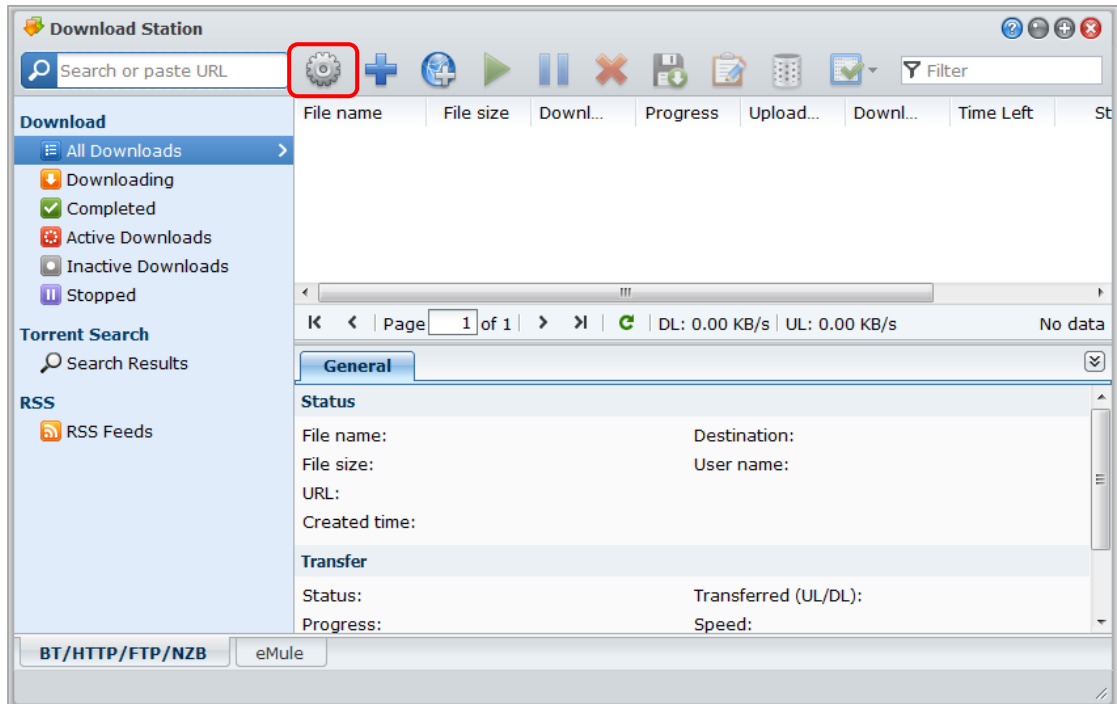
Module developers can use the following command to pack the dlm module:

```
linux> tar zcf mymodule.dlm INFO search.php
```

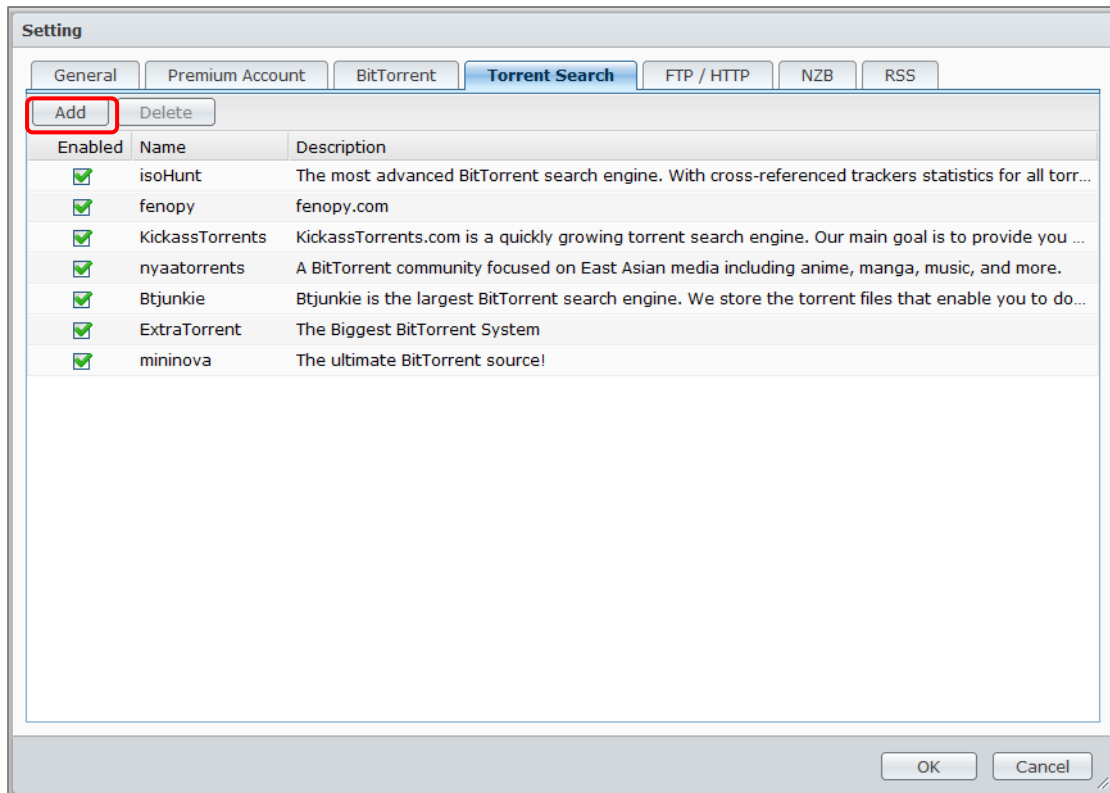
Apply Your Search Module

When the .dlm module is completed, follow the steps below to add it into your Download Station.

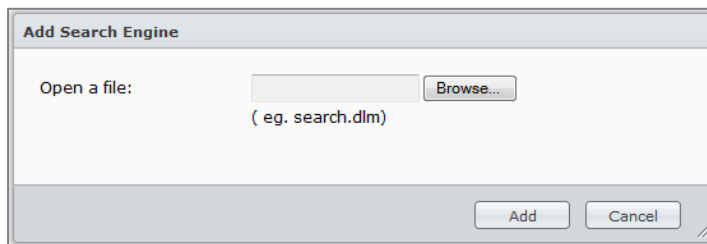
- 1 In Download Station, click the icon for **Setting**.



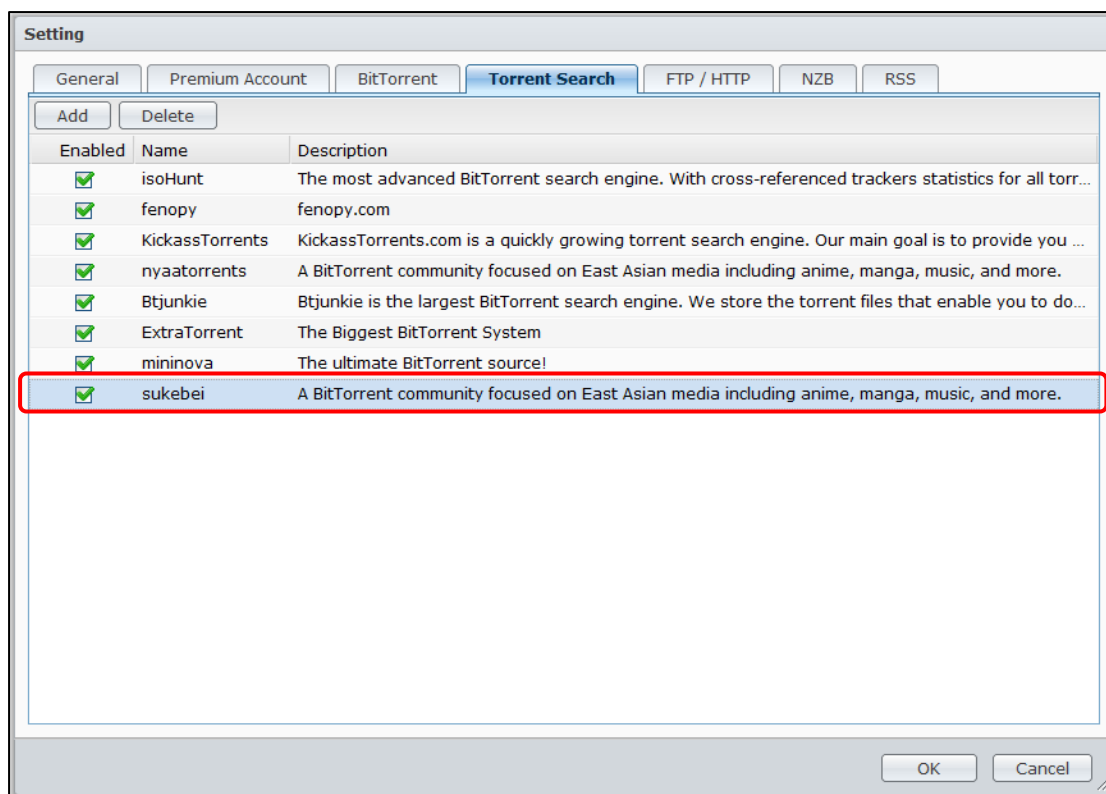
- 2 In **Setting** page, click **Torrent Search** tab > **Add**.



3 Specify the path of your .dlm file and click **Add** to add the search engine into the list.



4 The search module is added into the list. Click **OK** to confirm. Now you can search torrents with the newly added search engine.



Restricted Runtime Environment

PHP scripts in Download Station search module are run within a very restricted environment to protect DiskStation from malicious scripts. The following restrictions apply to any PHP scripts run by Download Station:

Minimal Privilege

Download Station search module is run using "nobody" privilege.

Restricted File Access

Download Station search module cannot access files in shared folders and system configuration directories.

Document Revision History

This table describes the changes to the Synology Download Station Search Module Development Guide.

Date	Note
2011-2-22	First creation of this document
2011-3-14	Official Release