

Synology Download Station

Developer Guide to

File Hosting Module

Synology[®]

2011-10-28



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
File Hosting Module for Download Station	4
File Hosting Module Content	4
Built-in Plug-in Parsing Helper Variables	14
Pack the .Host Module	15
Apply Your File Hosting Module.....	15
Restricted Runtime Environment.....	17
Minimal Privilege	17
Restricted File Access	17
Document Revision History	18

Introduction

On Synology DiskStation with DiskStation Manager 3.2 (hereafter DSM 3.2) and onward, 3rd-party developers are allowed to develop file hosting modules for Download Station to download files with either a paid or free account.

This document is to provide some guidelines on how to develop file hosting modules for Download Station. Developers are advised to have prior programming experience with PHP before reading this document.

File Hosting Module for Download Station

File Hosting Module Content

Users are allowed to install file hosting modules on Download Station with .host file extension. A host module is actually a normal unix tgz or tar.gz archive. For example, if you have a host module named **megaupload.host**, after extracting the module, you will see the following files in the current directory:

```
DiskStation> tar zxf megaupload.host
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 host.php
```

INFO

This is a mandatory file for module configuration. It defines the name of the module and describes how to locate the actual host module to perform download procedures.

host.php

This is a php module which performs download procedures when host.php is being loaded by Download Station.

INFO Content

The content of INFO is UTF-8 encoded JSON format and usually looks like this:

```

{
  "name": "megaupload",
  "hostprefix": "megaupload.com",
  "displayname": "Megaupload",
  "version": "1.0",
  "authentication": "yes",
  "module": "megaupload.php",
  "class": "SynoFileHostingMegaupload",
  "description": "Premium or free account supported."
}

{
  "name": "youtube",
  "hostprefix": "youtube.com,youtu.be",
  "displayname": "YouTube",
  "version": "1.0",
  "authentication": "no",
  "module": "youtube.php",
  "class": "SynoFileHostingYoutube",
  "description": "Public video downloading supported."
}
    
```

Table 1. INFO file definition

Key	Description	Remark
name	Unique name of the file hosting module among all Download Station modules. If the name is duplicate, the host cannot be installed.	Mandatory
hostprefix	If the web address typed in here conforms to hostprefix format, the module will download the files whose url matches that of the address.	Mandatory
displayname	Descriptions of the module displayed in the file hosting settings of Download Station	Optional
Key	Description	Remark

<code>version</code>	Version number of the file hosting module.	Mandatory
<code>authentication</code>	Check if the file hosting module needs to be authenticated.	Mandatory
<code>module</code>	Filename of PHP module. Usually it is <code>host.php</code> , but you can use other filenames.	Mandatory
<code>class</code>	Unique PHP class name among all file hosting modules.	Mandatory
<code>description</code>	Descriptions shown in the file hosting settings of Download Station.	Optional

Host.php

`host.php` is the PHP script that defines a PHP class performing download procedures . This class should include a constructor and two mandatory methods which are used to communicate with the file hosting module for Download Station. Details of these two mandatory methods are explained in Table 2 [“Mandatory methods in `host.php` and arguments”](#).

The content of `host.php` should be like this:

```
<?php
/* Copyright (c) 2011 Synology Inc. All rights reserved. */

class SynoFileHostingMegaupload {

    private $Url;

    private $Username;

    private $Password;

    private $HostInfo;

    private $MEGAUPLOAD_COOKIE = '/tmp/megaupload.cookie';

    private $MEGAUPLOAD_LOGIN_URL =
'http://www.megaupload.com/?c=login&setlang=en';

    private $MEGAUPLOAD_ACCOUNT_URL =
'http://www.megaupload.com/?c=account&setlang=en';

    private $MEGAUPLOAD_PREMIUM_ACCOUNT_KEYWORD = 'premium';

    private $MEGAUPLOAD = 'megaupload';

    public function __construct($Url, $Username, $Password, $HostInfo) {

        $this->Url = $Url;

        $this->Username = $Username;

        $this->Password = $Password;

        $this->HostInfo = $HostInfo;

    }

    //This function returns download url.

    public function GetDownloadInfo() {

        $ret = FALSE;

        $VerifyRet = $this->Verify(FALSE);

    }

}
```

```

    if (LOGIN_FAIL == $VerifyRet) {

        $ret = $this->DownloadWaiting(FALSE);

    } else if (USER_IS_FREE == $VerifyRet) {

        $ret = $this->DownloadWaiting(TRUE);

    } else {

        $ret = $this->DownloadPremium($this->CookieValue);

    }

    if (file_exists($this->MEGAUPLOAD_COOKIE)) {

        unlink($this->MEGAUPLOAD_COOKIE);

    }

    return $ret;

}

//This function verifies and returns account type.
public function Verify($ClearCookie)

{

    $ret = LOGIN_FAIL;

    $this->CookieValue = FALSE;

    if (!empty($this->Username) && !empty($this->Password)) {

        $this->CookieValue = $this->MegauploadLogin($this->Username,
$this->Password);

    }

    if (FALSE == $this->CookieValue) {

        goto End;

    }

    if ($this->IsFreeAccount()) {

```



```

        $ret = USER_IS_FREE;

    } else {

        $ret = USER_IS_PREMIUM;

    }

End:

    if ($ClearCookie && file_exists($this->MEGAUPLOAD_COOKIE)) {

        unlink($this->MEGAUPLOAD_COOKIE);

    }

    return $ret;

}

//This function performs login action.

private function MegauploadLogin($Username, $Password) {

    $ret = FALSE;

    //Save cookie file

    $PostData = array('login'=>'1',

        'redir'=>'1',

        'username'=>$this->Username,

        'password'=>$this->Password);

    $queryUrl = $this->MEGAUPLOAD_LOGIN_URL;

    $PostData = http_build_query($PostData);

    $curl = curl_init();

    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, FALSE);

    curl_setopt($curl, CURLOPT_USERAGENT,

DOWNLOAD_STATION_USER_AGENT);

    curl_setopt($curl, CURLOPT_POST, TRUE);

    curl_setopt($curl, CURLOPT_POSTFIELDS, $PostData);

    curl_setopt($curl, CURLOPT_COOKIEJAR, $this->MEGAUPLOAD_COOKIE);

```

```

curl_setopt($curl, CURLOPT_HEADER, TRUE);

curl_setopt($curl, CURLOPT_RETURNTRANSFER, TRUE);

curl_setopt($curl, CURLOPT_URL, $queryUrl);

$LoginInfo = curl_exec($curl);

curl_close($curl);

if (FALSE != $LoginInfo && file_exists($this->MEGAUPLOAD_COOKIE)) {

    $ret = parse_cookiefile($this->MEGAUPLOAD_COOKIE);

    if (!empty($ret['user'])) {

        $ret = $ret['user'];

    } else {

        $ret = FALSE;

    }

}

return $ret;

}

//This function acquires the download url with a free account.
private function DownloadWaiting($IsLoadCookie)
{

    $DownloadInfo = FALSE;

    $page = $this->DownloadParsePage($IsLoadCookie);

    if (FALSE != $page) {

        preg_match('/http:\\\\\/(.*?) class="down_buttl"/', $page,
$urlmatch);

        preg_match('/Unfortunately, the link you have clicked is not
available./', $page, $nofile);

        if (!empty($nofile[0])) {

            $DownloadInfo[DOWNLOAD_ERROR] = ERR_FILE_NO_EXIST;

```

```

    }

    preg_match('/count=(.*/', $page, $countmatch);

    if (isset($urlmatch[1])) {

        $Href = 'http://'.$urlmatch[1];

        $DownloadInfo[DOWNLOAD_URL] = trim($Href);

    }

    if (isset($countmatch[1])) {

        $DownloadInfo[DOWNLOAD_COUNT] = trim($countmatch[1]);

    }

    $DownloadInfo[INFO_NAME] = trim($this->HostInfo[INFO_NAME]);

}

return $DownloadInfo;

}

private function DownloadParsePage($IsLoadCookie)

{

    $Option = array();

    $curl = curl_init();

    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, FALSE);

    curl_setopt($curl, CURLOPT_USERAGENT,
DOWNLOAD_STATION_USER_AGENT);

    if ($IsLoadCookie) {

        curl_setopt($curl, CURLOPT_COOKIEFILE,
$this->MEGAUPLOAD_COOKIE);

    }

    curl_setopt($curl, CURLOPT_RETURNTRANSFER, TRUE);

    curl_setopt($curl, CURLOPT_URL, $this->Url.'&setlang=en');

    $ret = curl_exec($curl);

    curl_close($curl);

```

```

        return $ret;
    }

    private function IsFreeAccount()
    {
        $ret = TRUE;

        $curl = curl_init();

        curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, FALSE);

        curl_setopt($curl, CURLOPT_USERAGENT,
DOWNLOAD_STATION_USER_AGENT);

        curl_setopt($curl, CURLOPT_COOKIEFILE, $this->MEGAUPLOAD_COOKIE);
        curl_setopt($curl, CURLOPT_RETURNTRANSFER, TRUE);

        curl_setopt($curl, CURLOPT_URL, $this->MEGAUPLOAD_ACCOUNT_URL);

        $AccountRet = curl_exec($curl);

        curl_close($curl);

        $replace = array("\r\n", "\n", "\r");

        $AccountRet = str_replace($replace, '', $AccountRet);

        preg_match('/Account type:(.*)<\b>/', $AccountRet, $match);

        if (isset($match[0])) {

            $compare = strtolower($match[0]);

            if (strstr($compare,
$this->MEGAUPLOAD_PREMIUM_ACCOUNT_KEYWORD)) {

                $ret = FALSE;

            }

        }

        return $ret;
    }

```

```

}

//This function get premium download url.

private function DownloadPremium($CookieValue) {

    $DownloadID = $this->GetMegauploadDownloadID($this->Url);

    if (FALSE == $DownloadID) {

        return FALSE;

    }

    $page = $this->DownloadParsePage($IsLoadCookie);

    if (FALSE != $page) {

        preg_match('/Unfortunately, the link you have clicked is not
available./', $page, $nofile);

        if (!empty($nofile[0])) {

            $DownloadInfo[DOWNLOAD_ERROR] = ERR_FILE_NO_EXIST;

        }

    }

    if (empty($DownloadInfo[DOWNLOAD_ERROR])) {

        $returl =
'http://www.megaupload.com/mgr_dl.php?d='.$DownloadID.'&u='.$CookieValu
e;

        $DownloadInfo = array();

        $DownloadInfo[DOWNLOAD_URL] = trim($returl);

    }

    return $DownloadInfo;

}

//This function get megaupload download ID.

private function GetMegauploadDownloadID($Url) {

    $ret = FALSE;

    preg_match('/\?d=([a-zA-Z0-9]+)/', $Url, $DownloadID);

    if (!empty($DownloadID[1])) {

```

```

        $ret = $DownloadID[1];

    }

    return $ret;

}

}

?>

```

Table 2. Mandatory methods in host.php and arguments

Method	Argument	Description
constructor	\$Url, \$Username, \$Password, \$HostInfo	Allow the url typed in by users as well as the info, account, and password of the file hosting services to be acquired by the module.
GetDownloadInfo	N/A Return the actual download link.	Return the actual download link. wget will download files from the url which GetDownloadInfo returns.
Verify	\$ClearCookie: Check if the cookie file should be cleared after authentication.	Implement the method to verify users' account and password.

Built-in Plug-in Parsing Helper Variables

Some of the built-in helper variables are defined in /usr/syno/synoman/webman/modules/DownloadStation/common.php. They can be used by every host.php and are introduced as follows:

```

define('LOGIN_FAIL', 4);

define('USER_IS_FREE', 5);

define('USER_IS_PREMIUM', 6);

define('ERR_FILE_NO_EXIST', 114);

define('ERR_REQUIRED_PREMIUM', 115);

define('ERR_NOT_SUPPORT_TYPE', 116);

define('DOWNLOAD_STATION_USER_AGENT', "Mozilla/4.0 (compatible; MSIE 6.1; Windows XP)");

define('DOWNLOAD_URL', 'downloadurl'); // Real download url

```

```

define('DOWNLOAD_FILENAME', 'filename'); // Saved file name

define('DOWNLOAD_COUNT', 'count'); // Number of seconds to wait

define('DOWNLOAD_ISQUERYAGAIN', 'isqueryagain'); // 1: Use the original url
query from the user again. 2: Use php output url query again.

define('DOWNLOAD_ISPARALLELDOWNLOAD', 'isparalleldownload');//Task can
download parallel flag.

define('DOWNLOAD_COOKIE', 'cookiepath');
    
```

Pack the .Host Module

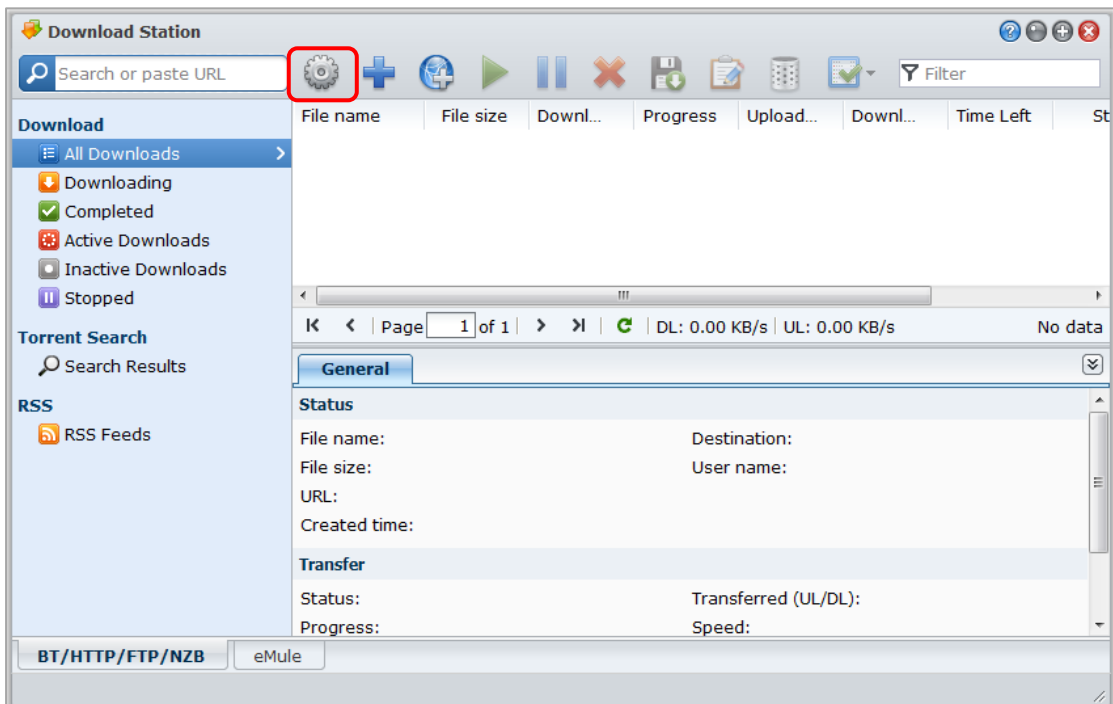
Module developers can use the following command to pack the .host module:

```
linux> tar zcf myhost.host INFO host.php
```

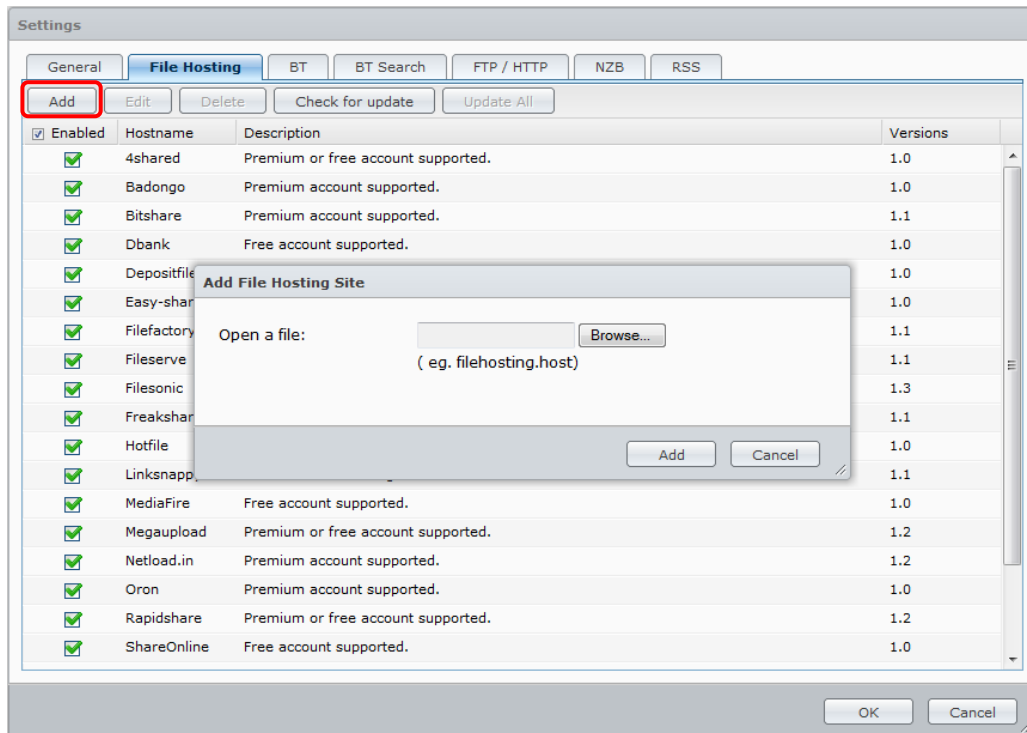
Apply Your File Hosting Module

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

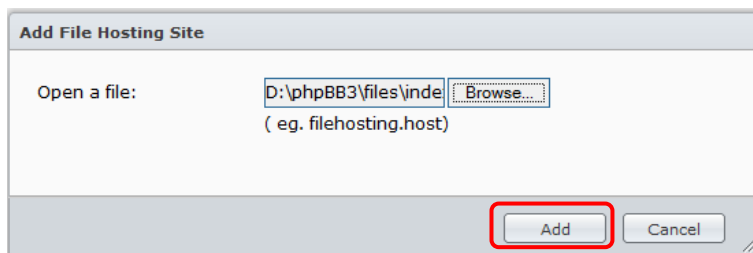
- 1 In Download Station, click the icon for **Setting**.



2 In **Setting** page, click **Add** under the **File Hosting** tab.

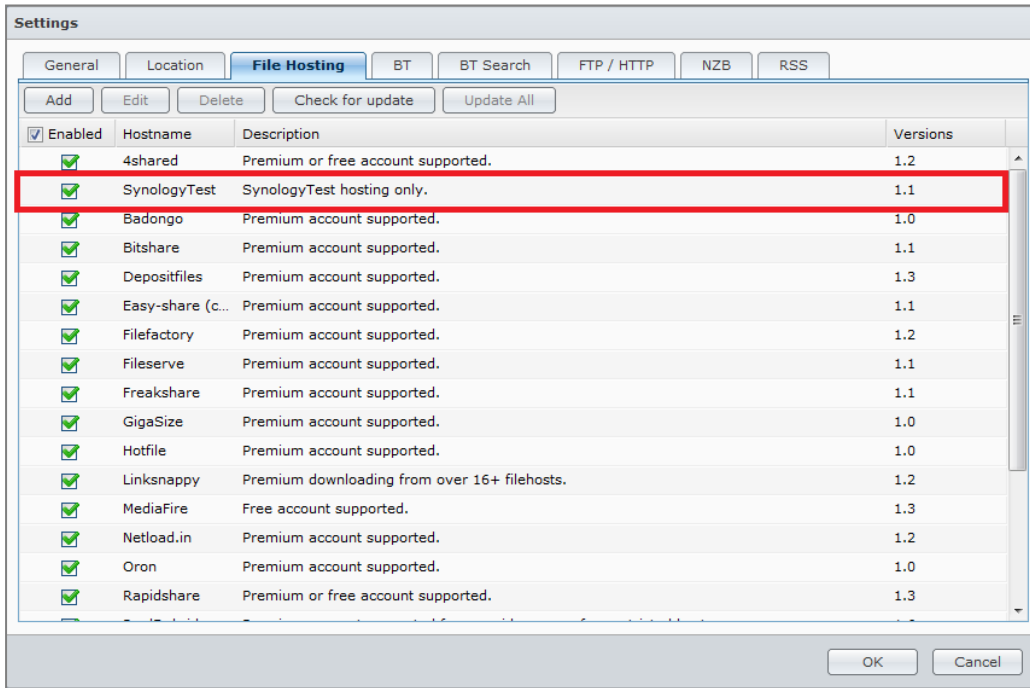


3 Specify the path of your .host file and click **Add** to add the file hosting to the list.



4 The file hosting module is added to the list.

Click **OK** to confirm. Now you can download the links from the newly added file hosting module.



Restricted Runtime Environment

PHP scripts in the file hosting module for Download Station are run within a very restricted environment in order to protect DiskStation from malicious scripts. The following restrictions apply to any PHP scripts run by Download Station.

Minimal Privilege

Download Station file hosting module is run using "nobody" privilege.

Restricted File Access

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

Document Revision History

This table describes the changes to the Synology Download Station Developer Guide to File Hosting Module.

Date	Note
2011-10-12	First creation of this document
2011-10-28	Official release