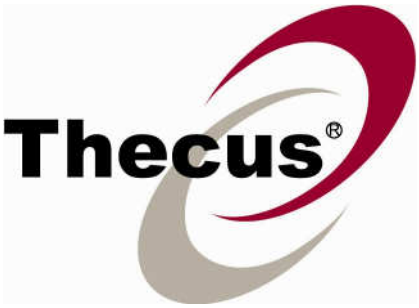



# Easy Module Create
# Step by Step (V 0.5)

**For Firmware version 5.00.0x and above**
**For N16000/N12000, FW 1.00.05 and above will be OK**

# Table of content



# Table of Modification:

| Date | Version | Dept. | Author | Note |
|---|---|---|---|---|
| 2011/01/28 | 0.3 | SW1 | Enian | |
| 2011/04/13 | 0.4 | SW1 | Enian | 1. Added Table of Modification<br>2. Added description for ModuleTargetNas and also a table for ModuleNasProtol |
| 2011/05/10 | 0.5 | SW1 | Enian | 1. Added description for ModuleLogin<br>2. Modofied the steps to make Module file<br>3. Added chapters 4 and 5 |

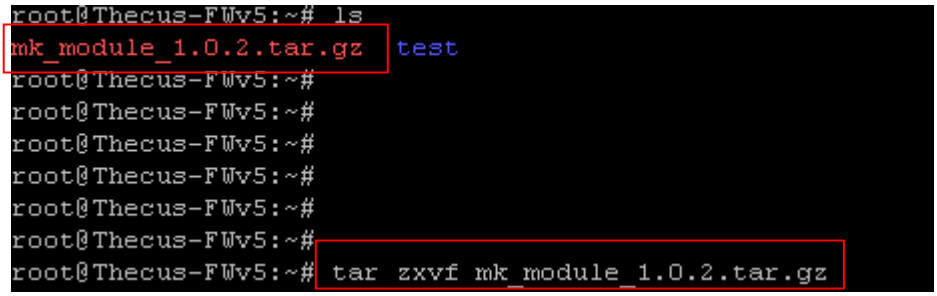
# Easy Module Create

This document is a quick guide for Thecus NAS users and also 3rd party module developers to develop a user module that can be installed in Thecus NAS. If you need certain function that standard NAS firmware doesn't have, developing a module by yourself is an alternative solution. This document will show how to make *install.rdf* from *install.conf* and *create_mod.sh*. And then make the *module.mod*

## 1. Decompression and Extraction

1.1 Download *mk_module_1.0.2.tar.gz* or newer version from Thecus FTP site, http://ftp.thecus.com/module/category-1/module/create_module_tool/ , or the other mirror sites. Transfer it from the computer to the development environment by *scp* (linux) or *winscp* (windows). The development environment refers to the guest OS provided by VMDK.

1.2 Type *tar zxvf mk_module_1.0.2.tar.gz* or newer version to extract the folder *mk_module*




## 2. Tree of *mk_module*

```
Module Name(Basic)
  |-- Binary
  |
  |-- Configure
  | |-- install.rdf
  | |-- license.txt
  |
  |-- Driver
  |
  |-- Shell
  | |-- enable.sh
  | |-- install.sh
  | |-- upgrade.sh
  | |-- module.rc
  | |-- uninstall.sh
  |
  |-- System
  | |-- conf
  | |-- etc
  | |-- include
  | |-- lib
  |      |-- syslib.sh
  |      | -- logevent
  |                | -- email
  |                | -- error
  |                | -- event
  |                | -- event_message.sh
  |                | -- information
  |                | -- sysinfo
  |                | -- warning
  |
  | |-- var
  | |-- lang
  |    |-- en
  |        |-- msg
  |    |-- de
  |        |-- msg
  |    :
  |    :
  |
  |-- Thecus
  |
  |-- WWW
  | |-- img
  | |-- index.htm
  |
  |-- create_module.sh
  |-- install.conf
```

| Folder/File | Description |
|---|---|
| syslib.sh | System libraries such as: etc_backup, sys_restore , logevent, etc. |
| logevent | Folder *logevent* keeps the system log and events |
| install.conf | The configuration of install.rdf |
| create_module.sh | To make *.mod* file and *install.rdf* |

# 3. Procedure

Step 1: Decompress *mk_module_1.0.2.tar.gz* to get the folder *mk_module*

```
root@Thecus-FWv5:~# tar zxvf mk_module_1.0.2.tar.gz
mk_module/
```

```
root@Thecus-FWv5:~# ls
mk_module  mk_module_1.0.2.tar.gz  test
```

Step 2: Change the folder name of *mk_module*; such as *mv mk_module Basic*

Step 3: Add/put the necessary shell scripts or binary files to folder *Binary*

Step 4: If necessary, add the additional library, mentioned in chapter 6, to the shell scripts (install.sh, upgrade.sh, install.sh, and enable.sh) in folder *Shell*

Step 5: Put necessary system files to the sub folder under *System*. If there is a configure file, put it in etc.

Step 6: Modify or add Module UI pages to folder *WWW*

Step 7: Make sure the variable name *module_name* in every shell scrip matches above step 2 in *Shell* and *WWW*

Step 8: Modify install.conf to fit your needs. For more details, refer to chapter 4.

Step 9: Run ./create_module.sh, and then put the generated files to folder *target*. For more details, refer to chapter 5.

For example:

```
root@Thecus-FWv5:~# mv mk_module Basic                                    2
root@Thecus-FWv5:~# cd Basic/Binary/
root@Thecus-FWv5:~/Basic/Binary# vi test.sh                               3
root@Thecus-FWv5:~/Basic/Binary# chmod 755 test.sh
root@Thecus-FWv5:~/Basic/Binary# cd ../Shell/
root@Thecus-FWv5:~/Basic/Shell# vi install.sh                             4
root@Thecus-FWv5:~/Basic/Shell# cd ../System/
root@Thecus-FWv5:~/Basic/System# cd etc                                   5
root@Thecus-FWv5:~/Basic/System/etc# cd ..
root@Thecus-FWv5:~/Basic/System# cd conf
root@Thecus-FWv5:~/Basic/System/conf# vi module.conf
root@Thecus-FWv5:~/Basic/System/conf# cd ../../WWW/                       6
root@Thecus-FWv5:~/Basic/WWW# vi index.htm
root@Thecus-FWv5:~/Basic/WWW# cd ..
root@Thecus-FWv5:~/Basic# vi install.conf                                 8
root@Thecus-FWv5:~/Basic# ./create_module.sh                              9
root@Thecus-FWv5:~/Basic# ls
Binary  Configure  create_module.sh  Driver  install.conf  Shell  Source  System  target  Thecus  WWW
root@Thecus-FWv5:~/Basic# cd target/
root@Thecus-FWv5:~/Basic/target# ls
Basic_1.0.0.mod  Basic_1.0.0.mod.sum
root@Thecus-FWv5:~/Basic/target#
```

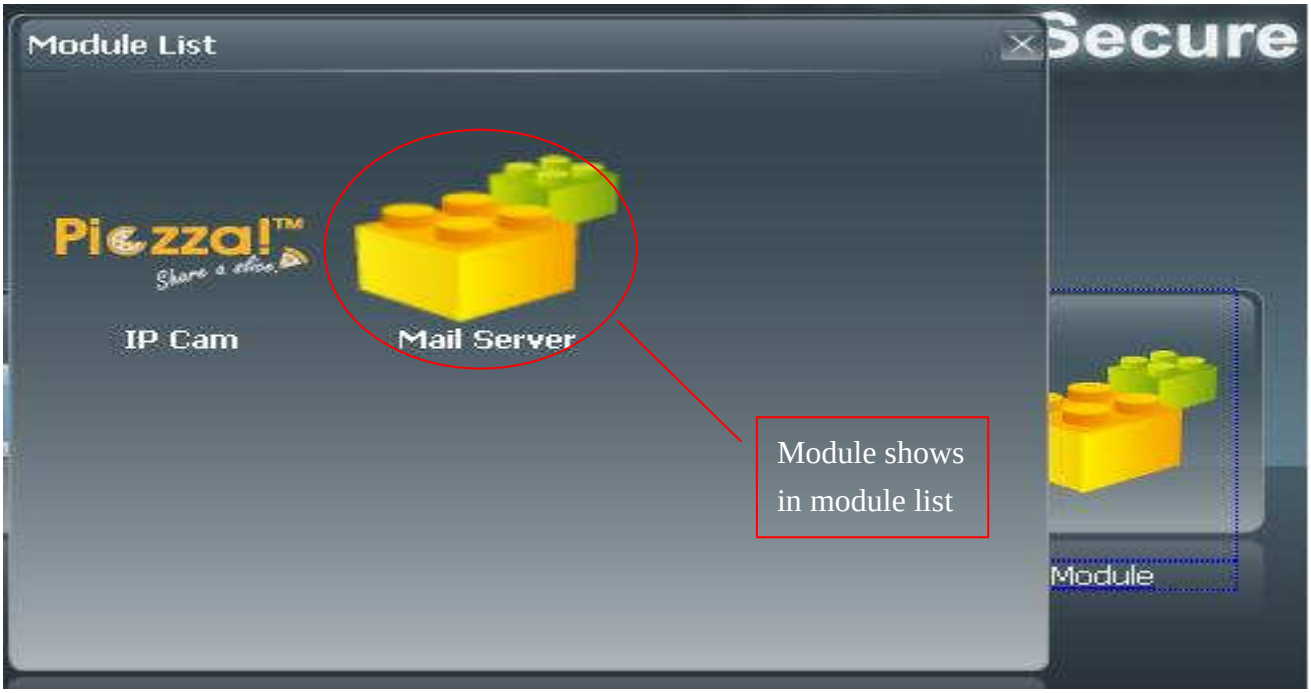
# 4. install.conf

## 4.1 Rule:

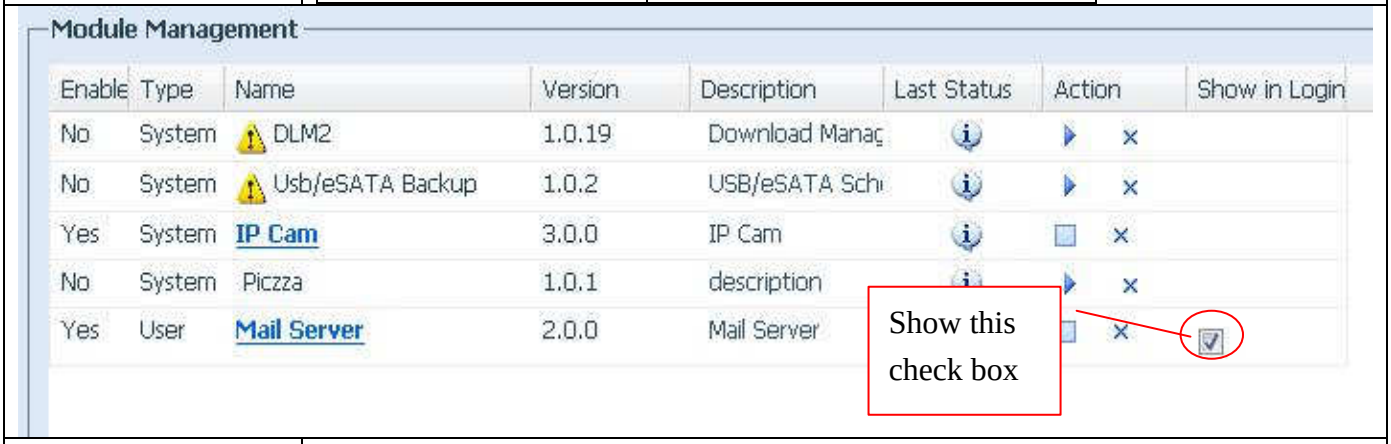
- Use ' to set all the variables
- While Value string contains ' , replace it by '/''

## 4.2 Global variable:

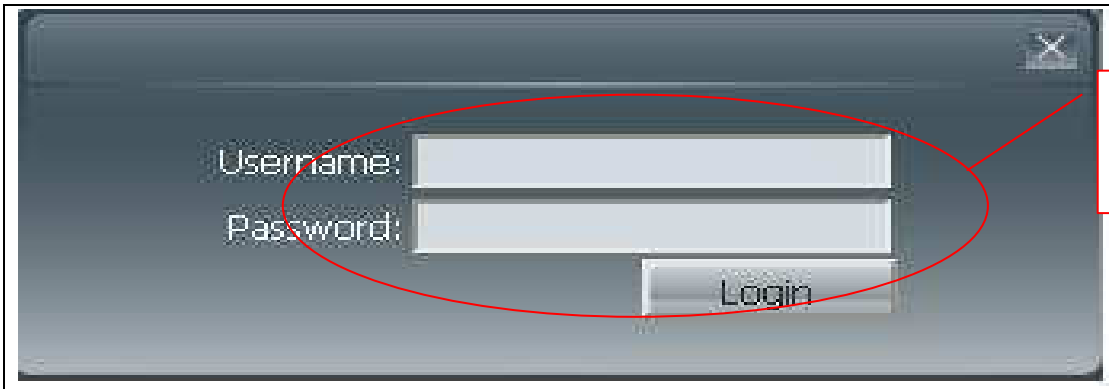
| Variable Name | Description |
|---|---|
| ModuleName | Module name (Refer to <md:Name> in install.rdf) |
| ModuleVersion | Module version (Refer to <md:Version> in install.rdf) |
| ModuleDesp | Module description (Refer to <md:Description> in install.rdf) |
| ModuleAuthor | Module author (Refer to <md:Authors> in install.rdf) |
| ModuleRef | Module reference (Refer to <md:Thanks> in install.rdf) |
| ModuleReboot | Will NAS reboot after module enable/disable? (Refer to <md:Reboot> in install.rdf) |
| ModuleHomePage | Location of Module homepage (Refer to <md:HomePage> in install.rdf) For example: www/index.html |
| ModuleWebUrl | Module related URL (Refer to <md:HomePage> in install.rdf) |
| ModuleIcon | Module icon (Refer to <md:Icon> in install.rdf)<br>Icon should be under www/ |
| ModuleMacStart | Define the lower limit of MAC address of the NAS<br>(Refer to <md:NasMacStart> in install.rdf) |
| ModuleMacEnd | Define the upper limit of MAC address of the NAS<br>(Refer to <md:NasMacEnd> in install.rdf) |
| ModuleShow | Defines if the user module shows an icon in the Module List of admin UI login page. (Refer to <md:Show> in install.rdf)<br><br>32 bit Firmware: 5.02.01 and above<br>64 bit Firmware: 2.02.01 and above |




| | |
|---|---|
| ModulePublish | Defines if the 'Show in Login' option column will have a checkbox or not. (Refer to <md:Publish> in install.rdf)<br><br>32 bit Firmware: 5.02.01 and above<br>64 bit Firmware: 2.02.01 and above |




| | |
|---|---|
| ModuleLogin | Define if the module authentication will follow the NAS login mechanism while entering the module via the Module List in NAS homepage. (Refer to <md: Login > in install.rdf)<br><br>32 bit Firmware: 5.02.01 and above<br>64 bit Firmware: 2.02.01 and above |




| | |
|---|---|
| ModuleKey | The folder name where module will be installed to (Refer to <md:Key> in install.rdf). Value allows a-z, A-Z, 0-9, and _ |

4.3 Depend Module variable:

| Variable Name | Description |
|---|---|
| ModuleDependMod[0]<br>ModuleDependVer[0]<br>ModuleDependUrl[0] | The information of the first Module that should be installed prior to the current one.<br><br>ModuleDependMod[0] indicate the Folder name of the first module (Refer to <md:DependName> in install.rdf)<br><br>ModuleDependVer[0] indicate the version number of the first module (Format: x.x.x) (Refer to <md:DependVer> in install.rdf)<br><br>ModuleDependUrl[0] indicate the URL to download the first module Refer to <md: DependUrl> in install.rdf) |
| .<br>.<br>. | |
| ModuleDependMod[n]<br>ModuleDependVer[n]<br>ModuleDependUrl[n] | The information of the n+1 Module that should be installed prior to the current one<br><br>ModuleDependMod[0] indicate the Folder name of the n+1 module (Refer to <md:DependName> in install.rdf)<br><br>ModuleDependVer[0] indicate the version number of the n+1 module (Format: x.x.x) (Refer to <md:DependVer> in install.rdf)<br><br>ModuleDependUrl[0] indicate the URL to download the n+1 |

| | |
|---|---|
| | module Refer to <md: DependUrl> in install.rdf) |

PS: 1. For example, while making a mail server module, you may need to define mysql module should be installed first. In this case, you can specify the mysql module by n=0 here.

2. The [n] of Depend Module variable should start from 0, and then 1, 2, 3, …n, n+1…

3. While define the module dependency, ModuleDependMod, ModuleDependVer, and ModuleDependUrl should be defined all together.

4.4 NAS FW setting variable：

| Variable Name | Description |
|---|---|
| ModuleTargetNas [0]<br>ModuleNasProtol [0]<br>ModuleNasVersion [0] | Define the first information (manufacture, model type, and F/W version) that allowed to install this module<br><br>ModuleTargetNas [0] indicate the first manufacture that allowed to install this module (Refer to < md:TargetNas > in install.rdf)<br><br>ModuleNasProtol [0] indicate the first model type that allowed to install this module (Refer to <md: NasProtol > in install.rdf)<br><br>ModuleNasVersion [0] indicate the first F/W version that allowed to install this module (Refer to <md: NasVersion > in install.rdf) |
| .<br>.<br>. | |
| ModuleTargetNas [n]<br>ModuleNasProtol [n]<br>ModuleNasVersion [n] | Define the n+1 information (manufacture, model type, and F/W version) that allowed to install this module<br><br>ModuleTargetNas [0] indicate the n+1 manufacture that allowed to install this module (Refer to < md:TargetNas > in install.rdf)<br><br>ModuleNasProtol [0] indicate the n+1 model type that allowed to install this module (Refer to <md: NasProtol > in install.rdf)<br><br>ModuleNasVersion [0] indicate the n+1 F/W version that allowed to install this module (Refer to <md: NasVersion > in install.rdf) |

PS: 1. For example, while making a module for Thecus NAS N7700, you may need to define:

ModuleTargetNas[0]='Thecus'

ModuleNasProtol[0]='N7700'

ModuleNasVersion[0]='5.00.00.18'

2. The [n] of NAS FW setting variable should start from 0, and then 1, 2, 3, …n, n+1…
3. While define the NAS FW setting variable, ModuleTargetNas, ModuleNasProtol, and ModuleNasVersion should be defined all together.
4. The value of ModuleNasProtol is listed in below table.

| Model Name of Thecus NAS Products | ModuleNasProtol |
|---|---|
| N5500/1U4600/N7700PRO/N8800PRO/N7700PLUS/N8800PLUS<br>N7700/N8800/N7700SAS/N8800SAS<br>N4200/N4200ECO/N4200PRO<br>1U4200XXX/N5200XXX/N8200XXX/N2200XXX/N3200XXX<br>N0503 | N7700 |
| N4100RPO | N4100RPO |
| N16000 | N16000 |
| N12000 | N12000 |

4.5 Example: (# for remark)

```
#!/bin/sh
ModuleName='Hello Word'
ModuleVersion='1.0.0'
ModuleDesp='My First module'
ModuleKey='Basic'
ModuleAuthors=''
ModuleRef=''
ModuleReboot='No'
ModuleHomePage='www/index.htm'
ModuleWebUrl=''
## Support Nas MAC start value ##
#ModuleMacStart='00:14:FD:11:C8:41'
ModuleMacStart=''
## Support Nas MAC end value
#ModuleMacEnd='00:14:FD:11:C8:4B'
ModuleMacEnd=''
ModuleShow='1'
ModulePublish='1'
ModuleIcon=''

## Module Depend ##
#    ModuleDependName[0]='test test'
#    ModuleDependVer[0]='1.0.0'
#ModuleDependUrl[0]='http://172.16.65.247'

## Depend Module 1 name ##
#ModuleDependName[1]='test1 test1'
#ModuleDependVer[1]='1.0.0'
#ModuleDependUrl[1]='http://172.16.66.224'

## NAS FW setting ##
## Module Support NAS 0 producer    ##
ModuleTargetNas[0]='Thecus'
ModuleNasProtol[0]='N7700'
ModuleNasVersion[0]='5.00.00.18'

## Module Support NAS 1 producer    ##
ModuleTargetNas[1]='Thecus'
ModuleNasProtol[1]='N4100PRO'
ModuleNasVersion[1]='5.00.00.18'
```

# 5. create_module.sh

Run *./create_module.sh* and then you will get folders *target* and *Source*

- *Source* folder: the files and sub-folders in *Source* will be the same with what mentioned in chapter 3. In addition, *install.rdf* will be generated in Source/Configure/
- *target* folder: the module file and module sum file will be here.
- To run *./create_module.sh*, make sure *install.conf* is existing

```
root@THECUS-FWv5:~/Basic#
root@THECUS-FWv5:~/Basic# ls
Binary  Configure  create_module.sh  Driver  install.conf  Shell  System  Thecus  WWW
root@THECUS-FWv5:~/Basic# ./create_module.sh
root@THECUS-FWv5:~/Basic# ls
Binary  Configure  create_module.sh  Driver  install.conf  Shell  Source  System  target  Thecus  WWW
root@THECUS-FWv5:~/Basic# cd target
root@THECUS-FWv5:~/Basic/target# ls
Basic_1.0.0.mod  Basic_1.0.0.mod.sum
root@THECUS-FWv5:~/Basic/target# cd ../Source/
root@THECUS-FWv5:~/Basic/Source# ls
Binary  Configure  Driver  Shell  System  Thecus  WWW
root@THECUS-FWv5:~/Basic/Source# cd Configure/
root@THECUS-FWv5:~/Basic/Source/Configure# ls
install.rdf  license.txt
root@THECUS-FWv5:~/Basic/Source/Configure#
root@THECUS-FWv5:~/Basic/Source/Configure#
root@THECUS-FWv5:~/Basic/Source/Configure#
```

PS. *mk_module_1.0.2.tar.gz* includes a Basic Source already. So, to make a Basic Module from it, just do steps 1 and 2 in chapter 3, and then run ./create_module.sh directly.

```
root@THECUS-FWv5:~#
root@THECUS-FWv5:~# ls
mk_module_1.0.1.tar.gz
root@THECUS-FWv5:~# tar zxvf mk_module_1.0.1.tar.gz
mk_module/
mk_module/Driver/
mk_module/System/
mk_module/System/etc/
mk_module/System/var/
mk_module/System/conf/
root@THECUS-FWv5:~#
root@THECUS-FWv5:~# ls
mk_module  mk_module_1.0.1.tar.gz
root@THECUS-FWv5:~# mv mk_module Basic
root@THECUS-FWv5:~# cd Basic/
root@THECUS-FWv5:~/Basic# ls
Binary  Configure  create_module.sh  Driver  install.conf  Shell  System  Thecus  WWW
root@THECUS-FWv5:~/Basic# ./create_module.sh
root@THECUS-FWv5:~/Basic# ls
Binary  Configure  create_module.sh  Driver  install.conf  Shell  Source  System  target  Thecus  WWW
root@THECUS-FWv5:~/Basic# cd target/
root@THECUS-FWv5:~/Basic/target# ls
Basic_1.0.0.mod  Basic_1.0.0.mod.sum
```

# 6. Additional libraries

*mk_module_1.0.2.tar.gz* also provides additional libraries thus module developers can use in *install.sh, upgrade.sh, uninstall.sh,* and *enable.sh*. They are under *System/lib* and include *libsys* and *logevent*.

## 6.1 System/lib/libsys

### 6.1.1 Usage

1. You will have to include *libsys* in the beginning codes of *install.sh, upgrade.sh, uninstall.sh,* and *enable.sh*

| Shell Script Name | include libsys |
|---|---|
| install.sh、upgrade.sh | . /raid/data/tmp/module/System/lib/libsys |
| uninstall.sh、enable.sh | . /raid/data/module/$module_name/sys/lib/libsys |

2. Table 6.1.2 indicates the necessary function calls in *install.sh, upgrade.sh, uninstall.sh,* and *enable.sh*

PS: To call any function within *unistall.sh*, you have to put it before the line: rm -rf "/raid/data/module/$module_name", 否則會無法執行

### 6.1.2 Table of Functions

| Function Name | Syntax | Return Value | Description |
|---|---|---|---|
| etc_backup | etc_backup “module_name” | 0/1 (success/fail) | ■ Backup sys/etc within the codes of install.sh. The backup selection can be set in System/conf/backup.list. It means, backup.list must exist in System/conf<br>■ The backup source is limited to the files and (sub)folders in sys *backup.list*<br>■ The folder path in *backup.list* should be the relative path of sys/etc<br>■ The format of backup.list is (say, to backup sys/etc/test.db)<br>source file,target file<br>Ex.: test.db,test.db |
| set_msg_log | set_msg_log “module_name” “msg variable or message” | None | ■ Show message and last status log when install.sh and upgrade.sh runs.<br>Show message when uninstall.sh runs.<br>Show last status log when enable.sh runs |

| | | | |
|---|---|---|---|
| | | | ■ When using msg variables, the syntax is<br>folder: en,es,de,tw,zh,ro,it,fr,pl,ja,ko,pt<br>File:msg<br>You will need related files and (sub)folders in System/message<br>■ File:msg allows: a~z, A~z, 0~9, and _<br>■ Example:<br>msg1=”hello”<br>msg2=”test” |
| set_event | set_event<br>“module_name”<br>“event_id”<br>“info/error/ warning”<br>“yes/no(for email)”<br>“para1” “para2” … | None | ■ Log system events (make sure the folder logevent and the files in it are exiting in System/lib/)<br>■ event_idshould match sys/lib/logevent/event_message.sh |
| create_module_folder | create_module_folder<br>“folder_path” | 0/1 (success/fail) | ■ create the folder where the module will be installed to (it will be located at /raid/data/_Module_Folder_/)<br>■ folder_path should be a relative path of /raid/data/_Module_Folder_/ |
| sys_restore | sys_restore ”module_name” | None | ■ Restore all files and (sub)folders in module sys |

### 6.1.3 Example (install.sh):

```
#!/bin/sh

res='fail'

module_name='Basic'
. /raid/data/tmp/module/System/lib/libsys
ret=`etc_backup "$module_name"`

mkdir "/raid/data/module/cfg/"              > /dev/null 2>&1
mkdir "/raid/data/module/cfg/module.rc/"    > /dev/null 2>&1
mkdir "/raid/data/module/$module_name/"     > /dev/null 2>&1
mkdir "/raid/data/module/$module_name/bin/"     > /dev/null 2>&1
mkdir "/raid/data/module/$module_name/shell/"   > /dev/null 2>&1
mkdir "/raid/data/module/$module_name/sys/"     > /dev/null 2>&1
mkdir "/raid/data/module/$module_name/www/"     > /dev/null 2>&1
mkdir "/raid/data/module/$module_name/drv/"     > /dev/null 2>&1

cp -f   /raid/data/tmp/module/Shell/module.rc   "/raid/data/module/cfg/module.rc/$module_name.rc"     > /dev/null 2>&1
cp -rf  /raid/data/tmp/module/Binary/*    "/raid/data/module/$module_name/bin"                        > /dev/null 2>&1
cp -rf  /raid/data/tmp/module/Shell/*     "/raid/data/module/$module_name/shell"                      > /dev/null 2>&1
cp -rf  /raid/data/tmp/module/System/*    "/raid/data/module/$module_name/sys"                        > /dev/null 2>&1
cp -rf  /raid/data/tmp/module/WWW/*       "/raid/data/module/$module_name/www"                        > /dev/null 2>&1
cp -rf  /raid/data/tmp/module/Driver/*            "/raid/data/module/$module_name/drv"
cp -f   /raid/data/tmp/module/Configure/license.txt    "/raid/data/module/$module_name/COPY"          > /dev/null 2>&1

set_msg_log "$module_name" "msg1"
set_msg_log "$module_name" "hello for message"
set_event "$module_name" "1001" "info" "no"
ret=`create_module_folder "basic"`

res='pass'

echo $res
```

### 6.1.4 Example (uninstall.sh):

```
#!/bin/sh

res='fail'
module_name=$1
. /raid/data/module/${module_name}/sys/lib/libsys
/raid/data/module/cfg/module.rc/"$module_name.rc" stop

set_msg_log "$module_name" "unistall for message"
set_event "$module_name" "1002" "info" "no"
rm -rf "/raid/data/module/cfg/module.rc/$module_name.rc"
rm -rf "/raid/data/module/$module_name"
rm -f "/img/htdocs/module/$module_name"

res='pass'

echo $res
```

## 6.2 System/lib/logevnet

- System/lib/logevent provides the related lib to log system events. Also, the set_event in System/lib/libsys can do the same thing.
- To log and send a system event by lib/logevent/event

  1. install.sh , upgrade.sh

     /raid/data/tmp/module/System/lib/logevent/event “module_name” “$event_id” “event_level (info/error/warning)” “email” “para1” "para2" …

  2. uninstall.sh,enable.sh

     /raid/data/module/$module_name/sys/lib/logevent/event “module_name” “$event_id” “event_level(info/error/warning)” “email” “para1” "para2" …

- event_id should match the variable number defined in lib/logevent/event_message.sh