Hello whoever is has opened this archive and taken to his time to read this readme.txt.

This file encloses my researches about the url rewriting for Xoops. I started the project under Xoops 2.5.3, one year ago, and did this one with the 2.5.5.

This is just a sample of the system I suggest.

With this methode, you'll convert urls like:

```
modules/mymodule/index.php?UrwIsActive=1&UrwExt=ext&UrwSeparator=0&UrwMaxKeyWSize=10&UrwMaxKeyW=12&op_1=1&op_
2=2&op_3=edit&op_4=data
```

into

```
new_name/new_pages/ext,1,12,10,0,1,2,edit,data/this-is-not-a-description.html
```

This methode can be customised on both core or module level. With settings like:

- Activate or deactivate the url rewriting ;
- Number of keywords used ;
- Minimum size of keywords used ;
- Type of operators used (above exemple is the coma ,) ;
- Suppression of the Xoops typical "module directory" in the url;
- Replacement of the module name by any custom name ;
- Replacement of any module file name by any custom name ;
- Use of any file extension, if any ;
- Shorter and SEO oriented urls.

Installation :

1/ Take a brand new Xoops 2.5.5 version and copy the content of this archive into it. Install your xoops, normally.
2/ Go in admin, and install the "mymodule" module. This module is a sample of how it can work.
3/ Go in the main page of this module and see how the url rewriting works.

Of course, be sure your server accept the .htaccess.
If you face some troubles (your demo site is maybe installe in a sub-domain), click on the ".htaccess generator" and copy/past the content on your .htaccess file at your xoops root.

Code use :
In the module (header.php for instance), include the urw class and settings:

```
#### URW Class inclusion
include_once( XOOPS_ROOT_PATH . '/class/xoopsurw.php' );

$urw  = &XoopsUrlRewrite::getInstance();

// urw is active or not on the module
   $urw->UrwIsActive(  $xoopsModuleConfig['urw_isactive'] );

// Available operators for each and every rewrited page.php (see settings below)
   $xoopsModuleConfig['urw_ops'] = array(
                                                 'index|id:UrwIsActive,
                                                        tx:UrwExt,
                                                        id:UrwSeparator,
                                                        id:UrwMaxKeyWSize,
                                                        id:UrwMaxKeyW,
                                                        id:op_1,
                                                        id:op_2,
                                                        tx:op_3,
                                                        tx:op_4',
                                                  'item| id:id,
                                                         tx:op'
                                                  );

   $urw->UrwOperators( $xoopsModuleConfig['urw_ops'] );

// current module extension * Optional: can be bypassed by website setting
   $urw->UrwExt( 'html' );

// operators separators     * Optional: can be bypassed by website setting
   $urw->UrwSeparator( '|' );

// Replacement name for the module ex : mymodule => my_url_rewrited_module
   $urw->UrwModuleName( 'cinema' );

// Replacement name for the module files ex : index => my_index_page and item => my_item_page.
   $urw->UrwPagesNames( array( 'index' => 'theater',
                               'item'  => 'movie'  ) );
#### URW Module settings ###
```

In the module pages there are 3 methodes to send datas to templates:

# **Methode 1**
# Send rewrited link at php level

```
  $link =  '<a href="' . $urw->XoUrlRewrite( $description, $url) . '" title="',$description.'">' . $caption . 
'</a>';
  $xoopsTpl->assign('link',         $link);
```

## In smarty template:

```
  <{$link}>
```

# **Methode 2**
# Send clas to smarty for conversion in the smarty

```
  $xoopsTpl->assign('urw',          $urw ); ## Send class to smarty
  $xoopsTpl->assign('caption',      $caption);
  $xoopsTpl->assign('description',  $description);
```

## In smarty template:

```
   <a href="<{$urw->XoUrlRewrite('My name is Bond, James Bond, 'index.php?id=007&op=mission')}>">Secret 
Agent</a>
```

or

```
   <a href="<{$urw->XoUrlRewrite('My name is `$name`, `$surname` `$name`", 
"index.php?id=`$id`&op=`$op`")}>"><{$profession}></a>
```

or

```
   <a href="<{$urw->XoUrlRewrite($description, 'index.php?id=`$id`')}>"><{$profession}></a>
```

# **Methode 3**
# Mixed methode

```
   $link = $urw->XoUrlRewrite( $description, $url );
   $xoopsTpl->assign('urw_link',     $link);
```

## In smarty template:

```
   <a href="<{$urw_link}>"><{$profession}></a>
```

Note that all the operators are not mandatory, neither do they need to be sorted out.