При разработке модулей и компонентов Joomla! часто требуется добавить кнопки в административной панели (бэкэнде), ниже мы рассмотрим как это сделать.

## Использование JToolBarHelper

Когда Вы добавляете метод JToolBarHelper в административной части Joomla!, он выводит соответствующий значок и формирует функцию-обработчик. Вы можете изменить функцию по умолчанию так:

```
$this->registerTask('add', 'edit');
```

Первый параметр JController::registerTask определяет задачу, а второй - метод выполнения. В примере мы изменили функцию по умолчанию для кнопки "Добавить".

## Кнопки по умолчанию / функции, встроенные в Joomla! 1.5

```
JToolBarHelper::title();
JToolBarHelper::spacer();
JToolBarHelper::divider();
JToolBarHelper::custom();
JToolBarHelper::customX();
JToolBarHelper::preview();
JToolBarHelper::help();
JToolBarHelper::back();
JToolBarHelper::media manager();
JToolBarHelper::addNew();
JToolBarHelper::addNewX();
JToolBarHelper::publish();
JToolBarHelper::publishList();
JToolBarHelper::makeDefault();
JToolBarHelper::assign();
JToolBarHelper::unpublish();
JToolBarHelper::unpublishList();
JToolBarHelper::archiveList();
JToolBarHelper::unarchiveList();
JToolBarHelper::editList();
JToolBarHelper::editListX();
JToolBarHelper::editHtml();
```

```
JToolBarHelper::editHtmlX();
JToolBarHelper::editCss();
JToolBarHelper::editCssX();
JToolBarHelper::deleteList();
JToolBarHelper::deleteListX();
JToolBarHelper::trash();
JToolBarHelper::apply();
JToolBarHelper::save();
JToolBarHelper::cancel();
JToolBarHelper::preferences();
JToolBarHelper::addEntry();
Ниже представлен синтаксис пользовательской кнопки Joomla!:
JToolBarHelper::custom('task','icon','icon over','alt', boolean, boolean);
Значки, связанные с этими функциями должны хранится в директории:
www/administrator/templates/khepri/images
CSS-файл, используемый для вывода кнопок Joomla!, хранится в:
www/administrator/templates/khepri/css/icon.css
Этот CSS-файл подключен через административный шаблон Joomla! Если Вы желаете
включить свой собственный значек в панель упраления Joomla!, то Вам
необходимо будет домавить соответствующие строки в файл CSS, например:
icon-32-iconname {
background-image: url(../../images/menu/icon-32-iconfile.png);
}
```

Как Вы видите, название значка и название файла значка могут не совпадать. Вы можете поместить файл значкаи и собственный CSS-файл в любую папку вашего

компонента. Чтобы подключить собственный CSS-файл к шаблону административной части Joomla, используйте
JHTML::_('stylesheet','THISCOMPONENTNAME.css','components/THISCOMPONENT/assets/css/');
При нажатии на кнопку Joomla! используется JavaScript, расположенный в includes/js/joomla.javascript.js
function submitbutton(pressbutton) {
submitform(pressbutton);
}
Вы можете переопределить эту функцию под собственные нужды.Чтобы сохранить кнопки на панели инструментов в рабочем состоянии,Вы должны заменить document.adminForm.task.value значением нажимной кнопки:
function submitbutton(pressbutton) { // Some conditions document.adminForm.task.value=pressbutton; // More conditions
submitform(pressbutton); }

## Кнопка загрузки (Upload button) Joomla!

```
Meтод 1

// Add an upload button and view a popup screen width 550 and height 400

$alt ="Upload";
$bar=&
JToolBar::getInstance('toolbar');
$bar->
appendButton('Popup','upload',$alt,'index.php', 550, 400);

Метод 2

// You view button for popup media manager tools

JToolBarHelper::media_manager('/');
```