



# Workflow Designer

custom fileactions

If you want to create a new action, what to do with files, generated by blocks, you could use this interface.

Existing examples of this interface is the directly download, store in Filestore, ...

Add a file in `/modules/Workflow2/extends/fileactions/` with the filename  
“`<individual>.inc.php`”

This file must contain a class which extends from `\Workflow\FileAction`.

This class must have the following structure:

```
<?php
namespace Workflow\Plugins\FileActions;

class [IndividualNameA] extends \Workflow\FileAction {

    /**
     * @param String $moduleName - The module from the workflow, this
     action will be used
     * @return array - Returns an array with the Option, this file could
     provide:
     *
     *         array(
     *             'title' => '<title of this FileAction (could
     be translated in lang Files)>',
     *             'options' => $options
     *         )
     *
     *         $options is also an array with configuration
     options,
     *
     *         the User should input, if he choose this
     action
     *
     *         $options = array(
     *             '<configKeyA>' => array(
     *                 'type' =>
     '<templatefield|templatearea|picklist|checkbox>',
     *                 'label' => '<label show before this
     configuration option (could be translated)>',
     *                 'placeholder' => '<placeholder of input
     field>',
     *
     *                 // if type = checkbox
     *                 // 'value' => 1
     *                 // if type = picklist
     *                 // 'options' => array('ID1' =>
```

```
'value1', 'ID2' => 'value2', ...)  
        ),  
        ...  
    )  
  
    */  
    public function getActions($moduleName) {  
        $return = array();  
  
        return $return;  
    }  
  
    /**  
     * @param array $configuration - Array with all configuration  
options, the user configure  
     * @param string $filepath - The temporarily filepath of the file,  
which should be transformed  
     * @param string $filename - The filename of this file  
     * @param \Workflow\VTEntity $context - The Context of the Workflow  
     * @param array $targetRecordIds  
     * @return void  
     */  
    public function doAction($configuration, $filepath, $filename,  
$context, $targetRecordIds = array()) {  
  
    }  
  
}  
  
\Workflow\FileAction::register('[SanitizedIndividualNameA]',  
'\Workflow\Plugins\FileActions\[IndividualNameA]');
```

