



Workflow Designer

custom file attachment source

When you want to use an individual source of attachments to an email.

Create a files in **modules/Workflow2/extends/attachments/** with the filename **<individual>.inc.php**.

This file must include a Class, which extends from **\Workflow\Attachment**.

You must implement the following structure to be recognized by Workflow Designer:

```
namespace Workflow\Plugins\Mailattachments;

class URL extends \Workflow\Attachment {

    public function getConfigurations($moduleName) {
        // This give you all options to create a complex configuration.
        // Important is to use the <a Tag and make the configuration
        container toggable with classNames and data-type Attribute

        $configuration = array(
            'html' => '<a href="#" class="attachmentsConfigLink"
data-type="url">Attach File from URL</a>
                <div class="attachmentsConfig" data-type="url"
id="urlAttachmentContainer" style="display:none;">
                    Load file from this URL and attach it to the
E-Mail.<br/>
                    <div class="insertTextfield" style="display:inline;"
data-name="attachment_url" data-style="width:250px;"
data-id="attachment_url"></div>

Filename of this file:<br/><em>(leave empty to get filename from
downloaded file)</em><br/>
                    <div class="insertTextfield" style="display:inline;"
data-name="attachment_name" data-style="width:250px;"
data-id="attachment_name"></div>
                </div>',

            'script' => "
Attachments.registerCallback('url', function() {
    jQuery('#urlAttachmentContainer').hide();

    var url = jQuery('#attachment_url').val();
    var name = jQuery('#attachment_name').val();
```

```
// This return the configuration, which define your type "url" and
additional configurations. ID is handled unique per mail configuration.
return [
    {
        'id'      : 's#url#' + url + '#' + name,
        'label'   : name == ''?'filename from url':name,
        'filename' : '',
        'options' : {
            'name' : name,
            'val'  : url
        }
    }
];
});");

        // You can also return multiple configurations
$return = array($configuration);

return $return;
}

/**
 * @param string $key The key, you define within your ID during
configuration
 * @param array $value The values from configuration. Array with
options is Array index [2]
 * @param $context \Workflow\VTEntity
 * @return array|void
 */
public function generateAttachments($key, $value, $context) {
    global $current_user;
    $adb = \PearDatabase::getInstance();

    $url = $value[2]['val'];
    $url = \Workflow\VTTemplate::parse($url, $context);

    $filename = \Workflow\VTTemplate::parse($value[2]['name'],
$context);

    if(empty($filename)) {
        $filename = basename($url);
    }
}
```

```
$filecontent = \Workflow\VtUtils::getContentFromUrl($url);

if(empty($filecontent)) {
    return array();
}

$filename = preg_replace('/[^A-Za-z0-9-_.]/', '_', $filename);

if($this->_mode === self::MODE_NOT_ADD_NEW_ATTACHMENTS) {
    $tmpfile = tempnam(sys_get_temp_dir(), 'Url');
    @unlink($tmpfile);
    file_put_contents($tmpfile, $filecontent);

    $this->addAttachmentRecord('PATH', $tmpfile, $filename);
    return;
}

$upload_file_path = decideFilePath();

$next_id = $adb->getUniqueID("vtiger_crmentity");

file_put_contents($upload_file_path . $next_id . "_" . $filename,
$filecontent);

$filesize = filesize($upload_file_path . $next_id . "_" .
$filename);
$filetype = "application/octet-stream";

$sql1 = "insert into vtiger_crmentity
(crmid,smcreatorid,smownerid,setype,description,createdtime,modifiedtime)
values(?, ?, ?, ?, ?, ?, ?)";
$params1 = array($next_id, $current_user->id, $current_user->id,
"Documents Attachment",'Documents Attachment', date("Y-m-d H:i:s"),
date("Y-m-d H:i:s"));

$adb->pquery($sql1, $params1);

$sql2 = "insert into vtiger_attachments(attachmentsid, name,
description, type, path) values(?, ?, ?, ?, ?)";
$params2 = array($next_id, $filename, '', $filetype,
$upload_file_path);
$adb->pquery($sql2, $params2);
```

```
// This add a Record of Type Attachment, which is stored in CRM
as Attachment, by ID
$this->addAttachmentRecord('ID', $next_id);

// Can also be PATH, like this:
//$this->addAttachmentRecord('PATH', $filepath, $filename);

}

}

// This will register this handler as Attachment Plugin
\Workflow\Attachment::register('url',
'\Workflow\Plugins\Mailattachments\URL');
```

