



# Workflow Designer

custom workflow trigger

Your module is able to create your own trigger type of Workflows. You can select this type during creation of Workflow or within Start block.

This example show how to register your trigger. You need to call this function only once. Registration is stored within database.

```
if(vtlib_isModuleActive('Workflow2')) {
    /**
     * @var $workflowObj \Workflow2
     */
    $workflowObj = CRMEntity::getInstance('Workflow2');
    $workflowObj->addTrigger('unique_trigger_key', 'title',
        'YourModuleName', 'Description');
}
```

unique\_trigger\_key A unique key to identify your trigger

title Shown title of your trigger in select box

YourModuleName Used to store relationship to your module for garbage collection

Description A short summary, when this trigger is executes

To execute all configured workflows, use this code:

```
$moduleModel = Vtiger_Module_Model::getInstance("Workflow2");

$moduleModel->runTrigger("unique_trigger_key", $crmid, [$envValues]);
```

\$envValues is optional. \$crmid is the record, which should used for execution. In this case you don't have any chance to influence execution of configured workflows. All active Workflows will be executed one after another.

If you want to have more opportunities to change execution, you can use the following code:

```
$triggerKey = "unique_trigger_key";
$crmid = 123; // Target record id
$envValues = array(); // Additional environment variables

global $root_directory;
require_once($root_directory."/modules/Workflow2/autoload_wf.php");

$wfManager = new \Workflow\Manager();
```

```
if(!empty($crmid)) {
    $context = \Workflow\VTEntity::getForId($crmid);
    $workflows = $wfManager->GetWorkflows($context->getModuleName(),
    $triggerKey);
} else {
    $context = \Workflow\VTEntity::getDummy();
    $workflows = $wfManager->GetWorkflows(false, $triggerKey);
}

$user = Users::getActiveAdminUser();
\Workflow\VTEntity::setUser($user);

if(!empty($envValues)) {
    $context->loadEnvironment($envValues);
}

/**
 * @var $workflows \Workflow\Main[]
 */
foreach($workflows as $wf) {
    $wf->setContext($context);

    if(!$context->isDummy()) {
        if(!$wf->checkCondition($context)) {
            continue;
        }

        // This function check if condition of Workflow allow
execution
        if(!$wf->checkExecuteCondition($context->getId())) {
            continue;
        }
    }

    $wf->start();
}
```

