



Workflow Designer

custom Record Selection method

You are able to implement your own Record Selection method, which you can use within every task, which select records.

For example if you need some special filter or have an external system, you want to use to filter.

Create a file **<customname>.inc.php** in the folder **modules/Workflow2/extends/recordsource/**

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

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

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

class CustomClassName extends \Workflow\RecordSource {

    public function getSource($moduleName) {
        $return = array();

        $return = array(
            'id' => 'unique id of method',
            'title' => 'Headline of selection method',
            'sort' => 30, // Sort Index. Condition = 10, others = 20
        );

        return $return;
    }

    /**
     * @param $context \Workflow\VTEntity Context of current main
     Workflow Record
     * @param $sortField string|null [null] Query should sort by this
     field & direction, if != null
     * @param $limit integer|null [null] Add an limit to your query
     * @param $includeAllModTables bool [false] You should add all Tables
     from Target Record
     *
     * @return string
     */
    public function getQuery(\Workflow\VTEntity $context, $sortField =
    null, $limit = null, $includeAllModTables = false) {
        $sqlQuery = 'SELECT vtiger_crmentity.crmid /* Insert Fields */
        FROM ...';
    }
}
```

```
        return $sqlQuery;
    }

    public function getConfigHTML() {
        $html = '... HTML of configuration. Store anything within the
task[...] or better task[recordsource][...] array';

        return $html;
    }

    public function getConfigInlineJS() {
        // Return JavaScript, which will executed one time if Record
Selector is used
        return '';
    }

    public function getConfigInlineCSS() {
        // Return CSS, which will only be applied to your Record Selector
Configuration
        return '...';
    }
}

// Register your Selection method
\Workflow\RecordSource::register('unique id of method',
'\Workflow\Plugins\RecordSource\CustomClassName');
```

(You found this example source within example.php in extends/recordsource folder, too)

Important note

Your getQueries function should return the column crmid, which is the ID of the Record, which match.

AND: You must add

```
/* Insert Fields */
```

in your Query, on the place, where the module could optionally add more columns from target module. (That's why it is important to respect the **\$includeAllModTables** parameter or ignore and add all tables in every case)

