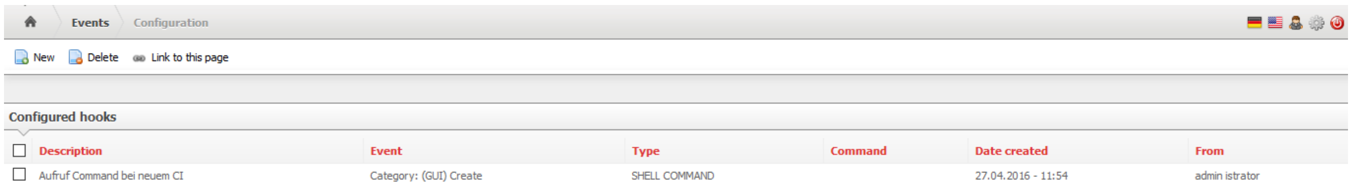
Event

The Event add-on allow for a high degree of automation when working with the IT documentation. If something is changed in the IT documentation, then third-party system will be informed about these changes. If, for example, a new VM is documented in i-doit, it can automatically be created and provisioned on a virtualization host.

Contents

Configuration

The configuration is accessed via **`Administration  CMDB settings  Events  Hooks`**.



Events Configuration

New Delete Link to this page

Configured hooks

	Description	Event	Type	Command	Date created	From
☐	Aufruf Command bei neuem CI	Category: (GUI) Create	SHELL COMMAND		27.04.2016 - 11:54	admin istrator

404 Page not Found

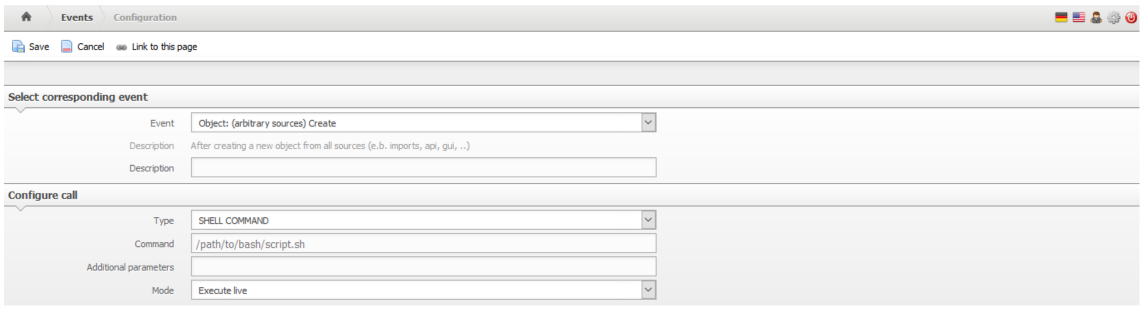
When you try to access the event configuration and receive an error message, stating that the page could not be found, this is in all likelihood due to a faulty configuration of the web server. The Apache module `rewrite` has to be activated and the processing of the `.htaccess` file in the installation folder of i-doit needs to be permitted (`AllowOverride All`).

Events are combined with commands. An event is triggered by a hook, an internal routine in i-doit. The following events are available:

- Category
    - Create (only via the web GUI)
    - Save
    - Archive/Delete/Restore/Purge
- Object
    - Create
    - Purge
- Object type
    - Create/Save
    - Purge

Thus there exist matching events for all states in the IT documentation. As many event-to-command combinations as desired are possible.

The command is carried out immediately once the configured event occurs. A shell script is executed for this purpose. This requires the rights to be executed by the user or group with whose rights the web server is running. On a Debian-based operating system this is the user `www-data` with the group of the same name. Because of this, the script under GNU/Linux requires the right bit for executing (x). The programming language is arbitrary but it should be supported by the operating system (Bash, PHP, Python, Perl etc.).



Events Configuration

Save Cancel Link to this page

Select corresponding event

Event	Object: (arbitrary sources) Create
Description	After creating a new object from all sources (e.b. imports, api, gui, ..)
Description	

Configure call

Type	SHELL COMMAND
Command	/path/to/bash/script.sh
Additional parameters	
Mode	Execute live

Information about the event is referred to the shell script. The information is encoded as JSON with BASE64. The following is an example of a JSON string when saving a category entry (the BASE64 decoding already took place):

```
{
   "success": 0,
   "objectID": "2912",
    "categoryID": 1,
    "categoryConst": "C__CATG__GLOBAL",
    "categoryDataID": "2949",
    "multivalue": false,
    "changes": [],
    "postData": {
        "data_id": "2948",
        "properties": {
            "title": {
                "tag": "title",
                "value": "Headquarter Network",
                "title": "LC__UNIVERSAL__TITLE"
            },
            "created": {
                "tag": "created",
                "value": "2016-04-26 14:41:01",
                "title": "LC__TASK__DETAIL__WORKORDER__CREATION_DATE"
            },
            "created_by": {
                "tag": "created_by",
                "value": "admin",
                "title": "LC__UNIVERSAL__CREATED_BY"
            },
            "changed_by": {
                "tag": "changed_by",
                "value": "admin",
                "title": "LC__CMDB__LAST_CHANGE_BY"
            },
            "purpose": {
                "tag": "purpose",
                "value": "1",
                "id": "1",
                "title_lang": "LC__CMDB__CATG__PURPOSE_PRODUCTION",
                "title": "LC__CMDB__CATG__GLOBAL_PURPOSE"
            },
            "category": {
                "tag": "category",
                "value": "2",
                "id": "2",
                "title_lang": "Demo",
                "title": "LC__CMDB__CATG__GLOBAL_CATEGORY"
            },
            "sysid": {
                "tag": "sysid",
                "value": "SYSID_1461677372",
                "title":"SYSID"
            },
            "cmdb_status": {
                "tag": "cmdb_status",
                "value": "6",
                "id": "6",
                "const": "C__CMDB_STATUS__IN_OPERATION",
                "title_lang": "LC__CMDB_STATUS__IN_OPERATION",
                "title": "LC__UNIVERSAL__CMDB_STATUS"
            },
            "type": {
                "tag": "type",
                "value": "77",
                "id": "77",
                "const": "C__OBJTYPE__SUPERNET",
                "title_lang": "LC__OBJTYPE__SUPERNET",
                "title": "LC__REPORT__FORM__OBJECT_TYPE"
            },
            "tag": {
                "tag": "tag",
                "title": "LC__CMDB__CATG__GLOBAL_TAG",
                "value": []
            },
            "description": {
                "tag": "description",
                "title": "LC__CMDB__LOGBOOK__DESCRIPTION"
            }
        }
    },
    "data": {
        "2949":{
            "title": "Headquarter Network",
            "status": "Normal",
            "created": "2016-04-26 14:47:57",
            "created_by": "admin",
            "changed": "2016-04-26 14:47:57",
            "changed_by": "admin",
            "purpose": "Produktion",
            "category": "Demo",
            "sysid": "SYSID_1461674878",
            "cmdb_status": "In Betrieb",
            "type": "Supernet",
            "tag": null,
            "description": "",
            "_id": 2912,
            "_title": "<span class=\"hide\">Headquarter Network<\/span><a name=\"Headquarter Network\" href=\"?objID=2912\" id=\"lb_58_2912\">Headquarter Network<\/a><script type='text\/javascript'>if ($('lb_58_2912')) new Tip('lb_58_2912', '', {ajax: {url: '\/i-doit\/?ajax=1&call=quick_info&objID=2912'}, delay: '0', stem: 'topLeft', style: 'default', className: 'objectinfo'});<\/script>",
            "_created": "<span data-date=\"2016-04-26 14:47:57\" class=\"hide\"><\/span>26.04.2016 (admin)",
            "_changed": "<span data-date=\"2016-04-26 14:47:57\" class=\"hide\"><\/span>26.04.2016 (admin)",
            "_cmdb_status": "<div class=\"cmdb-marker\" style=\"background-color:#33C20A;\"><\/div> In Betrieb"
        }
    }
}
```

You can see that the **`General`** category was saved successfully for the "Headquarter Network" object.

Additional parameters that are given to the shell script can also be set. These are static, in other words without a placeholder.

i-doit Controller

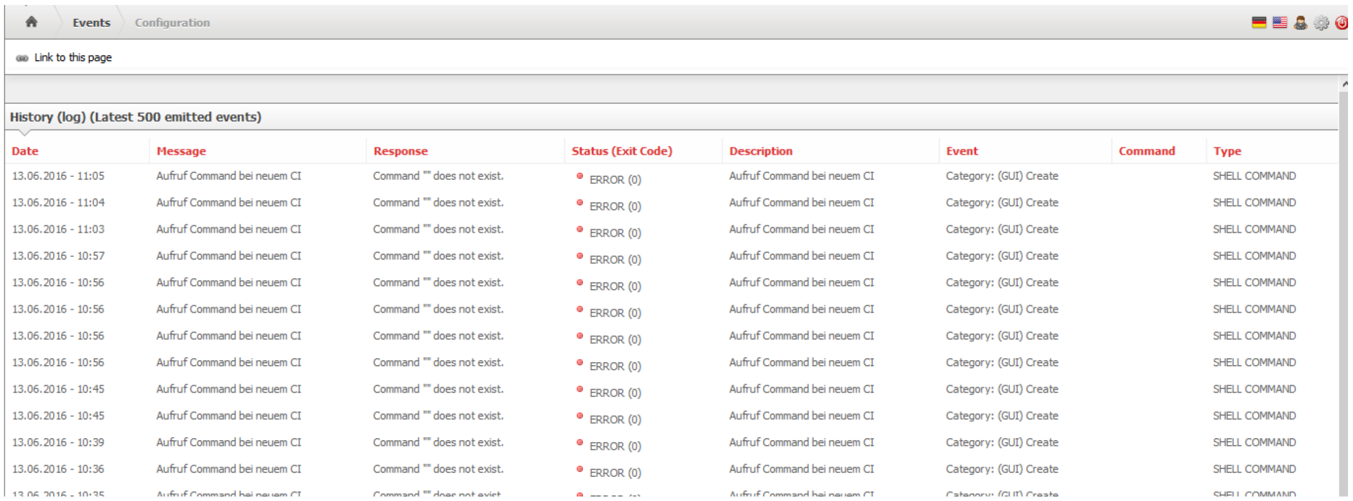
The described shell scripts are not only suited to control third-party systems but also i-doit itself. There is no reason not to use such a shell script in order to access the command line tool of i-doit, the controller or the API. This way, automated tasks can be handled within the IT documentation.

Performance

The commands are executed synchronously. For example, they are executed immediately as soon as the **`Save`** button is used and the system will wait until the shell script has ended. With many and/or extensive shell scripts this slow will down the web GUI of i-doit and thus negatively affect its usability. Because of this, it may be worthwhile to build a queue: i-doit runs a shell script which loads the parameters and stores them in a queue (for example in a temporary file). Another script is executed which processes this queue, using Cronjob or something similar. This way, an asynchronous workflow is created so that the usability of i-doit is not appreciably affected.

Logging

When an event is triggered and it is linked to a command, then this execution is logged. The last 500 entries are listed at **`Administration  CMDB settings  Events  History (Log)`**.



History (log) (Latest 500 emitted events)

Date	Message	Response	Status (Exit Code)	Description	Event	Command	Type
13.06.2016 - 11:05	Aufruf Command bei neuem CI	Command "" does not exist.	● ERROR (0)	Aufruf Command bei neuem CI	Category: (GUI) Create		SHELL COMMAND
13.06.2016 - 11:04	Aufruf Command bei neuem CI	Command "" does not exist.	● ERROR (0)	Aufruf Command bei neuem CI	Category: (GUI) Create		SHELL COMMAND
13.06.2016 - 11:03	Aufruf Command bei neuem CI	Command "" does not exist.	● ERROR (0)	Aufruf Command bei neuem CI	Category: (GUI) Create		SHELL COMMAND
13.06.2016 - 10:57	Aufruf Command bei neuem CI	Command "" does not exist.	● ERROR (0)	Aufruf Command bei neuem CI	Category: (GUI) Create		SHELL COMMAND
13.06.2016 - 10:56	Aufruf Command bei neuem CI	Command "" does not exist.	● ERROR (0)	Aufruf Command bei neuem CI	Category: (GUI) Create		SHELL COMMAND
13.06.2016 - 10:56	Aufruf Command bei neuem CI	Command "" does not exist.	● ERROR (0)	Aufruf Command bei neuem CI	Category: (GUI) Create		SHELL COMMAND
13.06.2016 - 10:56	Aufruf Command bei neuem CI	Command "" does not exist.	● ERROR (0)	Aufruf Command bei neuem CI	Category: (GUI) Create		SHELL COMMAND
13.06.2016 - 10:56	Aufruf Command bei neuem CI	Command "" does not exist.	● ERROR (0)	Aufruf Command bei neuem CI	Category: (GUI) Create		SHELL COMMAND
13.06.2016 - 10:45	Aufruf Command bei neuem CI	Command "" does not exist.	● ERROR (0)	Aufruf Command bei neuem CI	Category: (GUI) Create		SHELL COMMAND
13.06.2016 - 10:45	Aufruf Command bei neuem CI	Command "" does not exist.	● ERROR (0)	Aufruf Command bei neuem CI	Category: (GUI) Create		SHELL COMMAND
13.06.2016 - 10:39	Aufruf Command bei neuem CI	Command "" does not exist.	● ERROR (0)	Aufruf Command bei neuem CI	Category: (GUI) Create		SHELL COMMAND
13.06.2016 - 10:36	Aufruf Command bei neuem CI	Command "" does not exist.	● ERROR (0)	Aufruf Command bei neuem CI	Category: (GUI) Create		SHELL COMMAND
13.06.2016 - 10:35	Aufruf Command bei neuem CI	Command "" does not exist.	[illegible]	Aufruf Command bei neuem CI	Category: (GUI) Create		SHELL COMMAND

Releases

Version	Date	Changelog
1.1.1	2019-07-31	`[Bug] Drop-down empty in hooks for object types`
1.1	2019-01-30	`[Bug] Constant of user defined categories is missing`
1.0.1		`[Bug] i-doit 1.12 compatibility`
1.0	2018-07-03	Initial release