

TACTIC Tutorial

| REVISION HISTORY | | | |
|------------------|------|-------------|------|
| NUMBER | DATE | DESCRIPTION | NAME |
| | | | |

Contents

| | | |
|----------|---|----------|
| 1 | TACTIC Tutorial 1 | 1 |
| 1.1 | Javascript and Custom Layout Editor Related Questions | 1 |
| 1.2 | Expression Related Questions: | 5 |
| 1.3 | Tactic UI Related Questions | 6 |
| 1.4 | Questions related to videos and file checkins | 6 |
| 1.5 | Tactic Table Related Questions | 7 |
| 1.6 | Source Code Related Questions | 7 |
| 1.7 | New features TACTIC 4.2: | 8 |

1 TACTIC Tutorial 1

1.1 Javascript and Custom Layout Editor Related Questions

Question: Is there any way to run python code instead of javascript code in any widget in the table column definition?

Answer: Define a JS script in TACTIC's script editor, the purpose of which is to run a python script also in the script editor. The JS script would be something like:

```
var server = TacticServerStub.get();
var rtn = server.execute_python_script('folder/python_script', kwargs)
// You can pass in kwargs, a dictionary of variables you want to pass in the python script
```

Question: Is it possible to run an external python script not in Tactic but on the server?

Answer: One way to do this is to have a UI element activate a JS script in TACTIC, which calls a python script in TACTIC, which opens a python subprocess to run your external script.

1) Define a JS script in TACTIC's script editor, the purpose of which is to run a python script also in the script editor. The JS script would be something like:

```
var server = TacticServerStub.get();
var rtn = server.execute_python_script('folder/python_script')
```

2) Define a python script, called *folder/python_script* in this example, which calls your external python script via the command line. Ex:

```
import subprocess
subprocess.call(["python", "/tmp/test.py"])
```

If your external script is supposed to return something, you can read the output with syntax similar to this:

```
import subprocess
popen = subprocess.Popen('python', ['/tmp/test.py'], shell=False, stdout=subprocess.PIPE)
popen.wait()
output = ''
value = popen.communicate()
if value:
    output = value[0].strip()
    if not output:
        err = value[1]
        print err
return output
```

In doing so, the JS script will also have the return value stored in the variable "rtn".

Question: How to load an external js file in custom layout editor. Where can you save the js file?

Answer: You can include some external js and make use of some of its key functions immediately in the same call, you are required to put function definition within that function() { } block.

In the HTML:

```
<div>
<div class="my_class"/>
</div>
```

In the behavior:

```
<behavior class="my_class" event="load">
var js_files = [
"/assets/file_name_1.js", "/assets/file_name_2.js"
];
```

```
spt.dom.load_js(js_files , function() {
    // you can call functions defined in those js files path
    require; define; // require, define are functions defined in those js_files path
})
</behavior>
```

You can save the js file in tactic, there are three ways to do that: a) To get it working locally: you can put it in spt_js (not recommended because it is in TACTIC code), and load using:

```
spt.dom.load_js(['/context/spt_js/file_name.js'], function() {
```

b) You could also put it temporarily in the assets folder(This is mapped in apache to /assets/...) and load through:

```
spt.dom.load_js(['/assets/file_name.js'], function() {
```

c) Create a plugin and put it in the plugin folder. The plugin folder is mapped /plugins/...

```
spt.dom.load_js(['/mapped /plugins/file_name.js'], function() {
```

Question: How do you load the item into a new tab as opposed to the main body?

Answer: To load the item into a new tab as opposed to the main body, you need to call:

```
spt.tab.set_main_body_tab();
spt.tab.add_new(element_name, tab_title, class_name, kwargs);
```

The tab code and function can be found in "src/tactic/ui/container/tab_wdg.py"

Question: How do you add a new tab when there are multiple sets of tabs that can be added to?

Answer: If there are multiple sets of tabs (ex: a dashboard of tabs under a tab), you can decide which set to add a new tab to with code similar to the following:

```
var tabs_el = bvr.src_el.getElement(".spt_tab_top");
if (tabs_el) {
    spt.tab.set_tab_top(tabs_el)
    spt.tab.add_new(element_name, tab_title, class_name, kwargs);
}
```

Question: How to load a specific view into the current tab?

Answer:

```
var title = "Ticket List";
var view = "ticket_list";
var search_type = "scrum/ticket";

var kwargs = {
    'view': 'ticket_list',
    'search_type': 'scrum/ticket'
}

var class_name = "tactic.ui.panel.ViewPanelWdg";

spt.tab.set_main_body_tab();
spt.tab.load_selected(view, title, class_name, kwargs);
```

Question: Can you load a json file in Tactic through custom layout editor? Where can we save the file?

Answer: Yes you can, using :

```
spt.json("file_path", function( ) {
```

in a behavior (as this is a javascript function)

In TACTIC, the json file can be anywhere that is mapped by Apache. TACTIC does have /assets mapped and /plugins (if you want to create a plugin). The easiest way is just to create a folder in /assets and then use that path.

Question: Say you have a custom script in the script editor to "centralize" some code. How can you call it from a number of different custom layout views.

Answer: We don't have a great mechanism for this because it has to go to the server to get the script every time. It doesn't cache the code on the server. We have generally used .js files for central Javascript functions.

We do have some code that is outside the api that may help in src/context/spt_s/custom_project.js but it is not part of our API. We might map that into our API at some point, however, it will help you in the short run:

1)

```
var script = spt.CustomProject.get_script_by_path("test/my_script")
var ret_val = eval(script);
```

This will return the last evaluated variable in "script".

2) We do have another function:

```
spt.CustomProject.run_script_by_path("")
```

but it does a straight eval, which doesn't have an assumed return value, so the first example is the best bet.

Question: How can you connect subjects automatically based on dependencies. Say when you create new asset all those process(Task) should be connected automatically based on pipeline.

Answer: The "connection" table in sthbw database is mainly used for arbitrary connections between subjects. It is a generalized mechanism and not specific to tasks: for example, it allows media to be attached to jobs or reference images attached to shots. It can be used to "connect" tasks together to show a dependency, but we have never used it as such.

It can be accomplished using a trigger and the script, but this is not the best way store task dependency as a default. Data should be stored somewhere (ie: the depend_id column may be too simple given the complexities of pipelines) At this point, TACTIC doesn't have a built-in generalized solution for this. This is definitely something that can be discussed and we can incorporate into the development schedule.

So for now, you can do this by having triggers. So say you can have an event for inserting a new shot/asset which will run a script where you can create tasks and connect them to the subject.

You can look into triggers : http://community.southpawtech.com/tactic-setup_setup_project-automation-triggers.html

Create subject: http://community.southpawtech.com/python-client-api_general_insert.html

Connect subjects: http://community.southpawtech.com/python-client-api_general_connect_subjects.html

The example below might be helpful and JS script code will roughly be something similar:

```
var search_type = 'project/table_name';
var subject = server.insert(search_type, data);
// Create a corresponding task.
task_type = "sthpw/task";
var task = server.insert(task_type, task_data, { parent_key.__search_key__ });
server.connect_subjects(subject, task, {context: "main_task"});
```

Question: Load modes for CustomLayoutWdg ?

Answer: We added two load modes to the CustomLayoutWdg available in Tactic 4.1+

```
<element view="my_view" load="async"/>
<element view="my_view2" load="async"/>
<element view="my_view3" load="async"/>
<element view="my_view4" load="async"/>

<element view="my_view" load="sequence"/>
<element view="my_view2" load="sequence"/>
<element view="my_view3" load="sequence"/>
<element view="my_view4" load="sequence"/>
```

With "async", all will be loaded simultaneously. With "sequence", they will be loaded sequentially, meaning one load will not start until the other finishes. You can mix and match:

```
<element view="my_view"/>
<element view="my_view2" load="async"/>
<element view="my_view3" load="sequence"/>
<element view="my_view4" load="sequence"/>
```

Here, #1, will be loaded with the page, #2 and 3 will be loaded together (the first sequence item starts immediately with async) and then #4 is loaded after #3 is finished.

Question: When you use expression in the column definition, can you pass subject information explicitly?

Answer: Yes, if the mode_expr is "relative". The parent_wdg of a column is TableLayoutWdg and for a row in an edit pop up, the parent_wdg is EditWdg. You can pass in a search_key explicitly if there is no parent_wdg and search_key(search key doesn't exist for insert_search_key).

In the source code we have something like:

```
if mode_expr == 'relative':
    subjects = my.subjects
    if not subjects:
        parent_wdg = my.get_parent_wdg()
        if parent_wdg:
            # use the search_key as a starting point if applicable
            sk = parent_wdg.kwargs.get('search_key')
            if sk:
                subjects = [Search.get_by_search_key(sk)]
            else:
                sk = my.kwargs.get('search_key')
                if sk:
                    subjects = [Search.get_by_search_key(sk)]
```

Question: Can we pass subject information through the column definition as an argument or parameter to the js script being called in it?

Answer: Not at the moment

Question: In the edit mode, can you call a JS script?

Answer: Yes. Pass the script location as an argument

```
<element name="element_name" edit_script="JS Script Location">
  <display class="SelectWdg">
    <empty>true</empty>
  </display>
</element>
```

Question: Can you retrieve subject informations in the script called through edit_script ?

Answer: We can't use bvr.search_key because the source code doesn't pass it in. This is a limitation at the moment. Passing bvr information here would be difficult. There are many places where this function is called, each with different behaviors.

However, the "cell" is available to this method, so you can do:

```
var row = spt.table.get_row_by_cell(cell);
var search_key = row.getAttribute("spt_search_key");
```

There are inconsistencies where it's basically random what a script receives. At some point, we should conglomerate and standardize all of that.

Question: In the view mode, can you call a JS script for a button widget?

Answer: Yes. Pass the script location as

```
<script_path>js file location</script_path>
```

Question: Can you retrieve subject information in the script passed as script_path in view mode for button widget?

Answer: Yes using var search_key = bvr.search_key in JS script

Question: How can you add constraints to fields using Tactic?

Answer: For input validation refer to page 65 of the developer docs: http://community.southpawtech.com/sites/default/files/-doc_tactic-developer_0.pdf

1.2 Expression Related Questions:

Question: How to get the project url ex: https://VM_address/tactic/project_name?

Answer: \$PROJECT_URL is a python variable that will be evaluated by the server, so you can this directly.

Question: How can we have a combination of several columns from a row as a label expression.

Answer: You can do that by using a single expression(works for both absolute and relative mode and can have as many column values) For example:

```
@SUBSTITUTE((%0.d - %s ....),@GET(table_name/column_name), @GET(table_name/column_name) ↔
, ...)
```

Question: How can you create an expression column that uses the data from the current row and matches this against the full database?

Answer: 1. Send the id of the current subject in the current table table to get a certain column and look for columns related to that in another table Table2

```
@COUNT(oss_assets/table2['base_product', @GET(oss_assets/table1['id',$SUBJECT_ID].col_name) ↔
])
```

Note: Only available on Tactic 4.1+

However this expression will still run in absolute mode and it will be way slower and will only work for table expressions. . Here in this case, you don't need to mention the mode. But its use will still be limited as it doesn't have the capability of dealing with the case you mentioned(matching first five characters). It works for matching only equal strings.

1. Better way is to use python widget and add a script path and write a python script for that.

Question: How to get a list of parents for a list of task subjects using API?

Answer: To get a list of tasks:

```
tasks = server.eval("@SUBJECT(sthpw/task['search_type','film/shot?project=XYZ'])")
```

1. To get a list of parents corresponding to the list of tasks:

```
parents = server.eval("@SUBJECT(parent)", tasks)
```

1. However for single task , you can also use:

```
server.get_parent(task.get('__search_key__'))
```

Avoid looping to get the list of parents for a task list. Use the previous expression for better speed in this case. This is buggy at the moment and doesn't work correctly for list of different task subjects.

1.3 Tactic UI Related Questions

Question: How can we get all task search_key using trigger insertlsthpw/task?

Answer: The trigger is run for every trigger, so you don't need a list. It is passed in through the "input" variable. There is an example here:

<http://community.southpawtech.com/content/how-write-python-trigger-tactic-script-editor>

Task.py does have some convenience functions, but it is not necessarily called to create every task. The Task class is a wrapper class derived from SObject. Many functions will just use the lower level SObject functionality to create tasks. Modifying the Task.py class is not going to get what you want because these just contain helper functions that are sometimes convenient to use. A trigger is the proper place to do what you want to do because it will be called every time an event occurs. In this case "insertlsthpw/task" will be called **every** time a task is inserted. You can get the info the need as specified in the doc link above.

Question: How do you add a new button to the task view(which is in asset and shot of project vfx), but in "Manage sTypes" ?

Answer: You can do that in any task view, like project tasks, right click on the column header and choose "create new column ". You have a choice for button widget there.

Question: How can you customize weekends?

Answer: As of Tactic 4.1, we only support customizing the display for weekend in the Gantt Element Widget and you can't modify the number of working days. Its hard coded to 5days default. For display, you can pick the start of the week using the Edit Column Manager and it will automatically set the two previous days as weekend. So say you choose start day as Monday, then weekends are Saturday and Sunday. If you choose start day as Wednesday, then weekend is Monday and Tuesday.

image

Work Element Widget(work hour column in the table) and has nothing to do with Gantt Element Widget (task Schedule column). However, the work element widget still doesn't support the flexible weekends and its not linked to Gantt Element widget in any way. Its only the Gantt Element widget that we modified for customizing weekends. So if you modify the weekends in the Gantt element widget, they won't be reflected in the Work Element Widget. Unfortunately, at this point we don't support it for Work Element Widget.

Question: Is there is any translation for other languages for TACTIC UI?

Answer: There are two interfaces in Tactic:

1. **Administrative interface:** This interface supports only English at the moment and it requires some work to "internationalize" in order to accommodate other languages.
2. **Interface built using Custom layout editor (available in TACTIC 4.0+):** This interface can be themed with any look and feel in any language.

Question: Is there a way to clone/copy a project in Tactic?

Answer: Yes, you can use Template projects. If a project wasn't made a template at the time of creation follow the steps below:

→ Go under Projects and select the Template option under Is Template column for the project you wish to clone and hit save → Go under the Template Projects and update the Template by Clicking the Update icon under Update column. → The project you made the template of should appear in the option. Use the this project template when you create the new project(http://community.southpawtech.com/tactic-setup_setup_create-new-project.html). It will be a copy of that project.

Note: Template projects don't copy the data, just the configuration. If you want to the full data, the best is to just dump the database and import it on another server. You shouldn't do tests on a production server.

1.4 Questions related to videos and file checkins

Question: Checkin files under vfx project and generate the thumbnail using the generate_video_thumbnail script using FFMPEG?

Answer: There are a few issues with generate_video_thumbnail script.

What to check for:

- 1) the thumb_path variable is hardcoded to a folder which doesn't exist. We generally use the "get_tmp_dir" from Tactic's Environment class (under pyasm/common) method to get a temporary folder to use for operations like this instead.
- 2) The path variable gets the incorrect path. add the mode="lib" argument to the get_path_from_snapshot function for the server path.
- 3) Even with the above, FFMPEG won't generate the temporary thumbnail because the script is buggy. You need go through FFMPEG documentation and fix the script.

Alternate solution is to not use the script. In that case, the checkin happens fine but the thumbnail has to be generated separately.

1.5 Tactic Table Related Questions

Question: How can you disable the edit option for a column?

Answer: For this there are 2 options:

- 1) For the column, Edit the column definition and in the edit tab, un-check edit which will disallow editing in the table and remove the code field from the edit view.
- 2) If you edit the column definition, in the edit tab make sure the widget is set to TextWdg. Then set the read_only option to true.

1.6 Source Code Related Questions

Question: Is there a way to get the immediate dependent task id/s using a task's id?

Answer: Yes

```
def get_dep(task):
    expr = "@SBJECT(sthpw/task['project_code','test777']['id','{}']).format(task)
    task_dict = server.eval(expr)[0]
    search_task = Search(Task)
    search_task.add_filter("id", task_dict['id'])
    task_objects = search_task.get_objects()
    task_object = task_objects[0]
    task_pipeline = task_object.get_pipeline()
    task_input_tasks = task_object.get_input_tasks()
    task_depend = []
    for each in task_input_tasks:
        task_depend.append(each.get_id())
    return task_depend
```

Question: Which module is called when a sign in is done?

Answer: Under pyasm/security you can find security.py and authenticate.py. The login_user function under the Security class in security.py logs in the user, authenticating using the TacticAuthenticate class in authenticate.py.

Question: If I have the ids of specific tasks of a project, how can I get the related notes?

Answer: Although the notes look at though they are connected directly, they are actually connected through the parent. They are then filtered by the same process as the task. So the basic example would be:

```
@SBJECT(parent.sthpw/note)
```

This will get all notes for the parent. To filter, it will depend on where they are using the expression but it will be something like:

```
@SBJECT(parent.sthpw/note['process', @GET(.process)])
```

The @GET(.process) in the filter will get the process value from the relative sObject (the task in this case) or If in code and the process is known:

```
@SBJECT(parent.sthpw/note['process', 'something'])
```

1.7 New features TACTIC 4.2:

i. Ability to set multiple sandboxes and multiple repo locations

We have added the ability to set multiple sandboxes and multiple repo locations. They can be set for any naming convention using the `sandbox_dir_alias` and `base_dir_alias` columns in the naming table. The values are "aliases" to the file-system, making it easy to systems people to remap to other base folders. The `sandbox_dir_alias` column can contain "|" separated values to provide an optional list of sandboxes to work in. Alternatively "all" will list all of the available paths and leaving it empty will use "default". It is backwards compatible so former configurations with just the value will work as before. You can see this optional selection in the check-in/out widget.

An example for config setting is shown below:

```
<asset_base_dir>{
  "default": "/home/tactic/assets",
  "second": "/home/tactic/assets2",
  "third": "/home/tactic/assets3"
}</asset_base_dir>

<web_base_dir>{
  "default": "/assets",
  "second": "/data/assets/second",
  "third": "/data/assets/third"
}</web_base_dir>

<win32_sandbox_dir>{
  "default": "C:/spt/sandbox",
  "fast": "C:/Data/Test/FAST",
  "slow": "C:/Data/Test/SLOW",
  "ftp": "C:/Data/Test/FTP",
  "remote": "C:/Data/Test/REMOTE"
}</win32_sandbox_dir>
```