

How To Create A New Workflow

▼ Details

This tutorial explains how to design and implement workflows into ERP5. We will design the workflow to associate with a Discussion Thread and a Discussion Post in our Discussion Threads module.

Agenda

- How to design a workflow
- How to implement workflows on ERP5 documents

▼ Details

A basic forum workflow diagram

▼ Details

Workflows are used to implement user interaction - validating objects, ordering, planning and confirming orders, etc. If you want to learn more:

[HowToUseAndDefineInteractionWorkflows](#)

The first thing to do when you want to implement a workflow on any ERP5 document is to **design** it. By designing it, you are making the goal of your development very clear and you will be able to use it to explain to the community how your module works.

A workflow is made of states and transitions. **States** are the status of your documents. **Transitions** are the possible actions you can take to change the state of your documents. As you can see on the diagram, transitions are **unilateral**.

This diagram explains how our Threads in the Discussion Threads Forum will work. When a user create a Thread, it will be put in **Draft** state. When he finishes editing it, he will be able to **Published** it. It will then be viewable by all the community. If the Thread is very interesting and pertinent it can be put on **Sticky** by an administrator. If one user creates a Thread but hide it, the user can still see his Thread and can reply but another user cannot see it. If it is useless or outdated an administrator will be able to **Close** it. We leave the possibility to re-open a Thread if an administrator makes a mistake. He will then use the **Unclose** action.

Portal Workflow Folder

▼ Details

All workflows can be manage in the **Portal Workflow Folder**. Therefore for this part we will mainly work in `INSTANCE_ID/erp5/portal_workflow/manage_main` page.

This tool will help you to **create, customize and use** your workflows in ERP5. When you access it, you see the list of all workflows being used in your instance.

The two important tabs to look at are the **Workflows tab**, which is used to assign a workflow to a type of object and the **Contents tab** which is used to store the workflows and access them.

Create a workflow

▼ Details

When you want to create a new workflow, you should be careful with its **name**: normally we always name them using the convention **"name_of_the_type_workflow"**. You should have already many example in the Contents tab.

In the Content tab of Portal Workflow Folder, click on **Add a new "ERP5-style pre-configured DCWorkflow"** and name it **"discussion_thread_workflow"**. You can then change the name of this new workflow to **"Discussion Thread Workflow"**.

Create a state

▼ Details

We will now **create the states in our workflow**. Go into your workflow by clicking on **"discussion_thread_workflow"**.

If you go on the **States tab** of the discussion_thread_workflow, you can see that the **Draft** state has already been created. This is the default state used when a document is created in ERP5. You can now add the five states left: **public, sticky, closed, hidden and cancelled**.

States can be added easily in the states tab, just give its **id** and click on **"Add"**.

After you added the state in **State tab**, please remember to **add Title** to each state by clicking on them, as shown in the slide. It is important because the states can not be shown in the user interface if they don't have title.

After adding the states, you will then be able to select it in the **Transitions tab**.

Once the five states are available in our workflow, we can start to **create the transitions in our workflow**.

Create a transition

▼ Details

For each transition in our workflow, we will create **two** transitions in the system. The first one is a **logic transaction**, for instance the "publish" transaction. This action will contain all the logics when calling this transition. It can be **directly called by some scripts in the system**. The second one is a **user transaction**, for example "publish_action" which is corresponding to "publish" transaction, it is the action which will be displayed in the user interface and which will be called **only if the user click on it**. It will then **call the first transition**.

We should follow the **naming convention**, that is why the transitions names has to be in **lower case**.

Security restriction can be set up on the user action, whereas the other action can only be called by a script.

Creating the transitions is easy. Go on the **Transitions tab** of the discussion_thread_workflow, fill in the id and click on **"Add"** as shown on the screenshot.

Logic transition

▼ Details

We will start by configuring the **logical transition** "publish". Access the transition by **clicking on its name**.

It is necessary to put the **Title** of the transition, do not forget it!

A workflow can be **triggered by three types of events**. The **"Automatic"** type is when the workflow will have its launch condition set in TALES (scenario correspond too, tales is the word use in ERP5). The **"Initiated by user action"** type is for transitions meant to be called from the "Action..." bar and the **"Initiated by WorkflowMethod"** means that only workflow methods will call this transition. In our case it can only be called from another method so you must check the **"Initiated by WorkflowMethod"** box.

In the **Destination State** field, you should put the state that this transition makes reference to. Here, the publish workflow should change the workflow state to public, so you must select **public** in the Destination State drop down menu.

Don't forget to click on "Save Changes" before navigating to a new page.

User transition

▼ Details

Next we should configure the **user transition** "publish_action".

First, don't forget to fill the **Title**. You can add a description for this transition.

Here we will choose the second trigger type **"Initiated by user action"**, so that user can invoke this transition from user interface.

In **"Script (after)"** property, you must select a logic transition to invoke, here **"publish"**, so that the workflow will allow user to publish a document. In general, a **"Initiated by user action" transition invokes a "Initiated by WorkflowMethod" transition** in "Script (after)" in order to change the workflowstate.

The transition "Initiated by WorkflowMethod" is always the one that is responsible to change the workflow state, never the "Initiated by user action". As the invoked logic transition "publish" will automatically change the workflow state to "public", so you don't need to select "public" in the **Destination State** drop down menu.

In order to show the publish action in the **"Action..." menu in user interface**, in addition to choose the trigger type, you still need to fill the **"Display in actions box"** values as listed below:

Name for the publish action: it will be displayed in the menu.

URL: it is used by the action, which must be:

`%(content_url)s/Base_viewWorkflowActionDialog?workflow_action=publish_action.`

It calls the default **workflow view dialogue** on the page.

Category: it is used to put the action in a specific section of the "Action..." menu (section titles are displayed in grey).

We can define here when the action will be viewable. But we will wait until we associate it with a state to do it.

Associate transitions with a state

▼ Details

Now that we have created transitions, we will associate them with a state.

In our example, according to the global workflow, the **Publish transition** must be available when the document is in **Draft state**. So you must go back in the **States tab of "discussion_thread_workflow"** and click on Draft. You should now see all transitions available as shown above. It allows you to choose what kind of transitions are accessible for this state. Please select the **check boxes of the "publish" and "publish_action" transitions**

After you associate the transition to a state, if the transition can be initiated by user action, it will appear in the "Action..." menu of every object which has this state.

Current states with transitions of your workflow

▼ Details

After all those manipulations, we can go back in **"discussion_thread_workflow" and click on the "States" tab**. You should have the same result as in the illustration above. It's time to discover a very useful tab which is the **"Graph" tab**, in the top right corner. If you click on it, you will see a graphic representation of the workflow you just configured.

In our example, you should have two states with two transitions defined as shown above.

At this step, if you haven't got error you can redo all the operation we made to create the remaining transitions such as cancel, stick, unstick, hide, close and unclose.

What you should have after

▼ Details

The above illustrations are the result of the configuration of each state and their related transition(s).

Don't forget to fill in the fields **Title**, Description and most importantly the field Name and URL in the section named **"Display in action box"**. Be careful to the associations between the states and the transitions.

Remember that Unstick and Unclose are just transitions, not states.

Assign the workflow

▼ Details

We now have fully created our workflow. The final step is to associate it with the proper document. In the **Workflows tab of portal_workflow/manage_main**, you can associate a workflow to a type of object.

We will associate the **Discussion Thread** with our newly created workflow. Add **",discussion_thread_workflow"** right after **"edit_workflow"** then click on **"Change"**. **Workflow fields are comma-separated lists of workflows.**

You can add many workflows to a single type of object, but be careful because the security and the workflow graph can become really complicated if you have states with the same names (and it increases security risks).