

INTALIO

Tempo

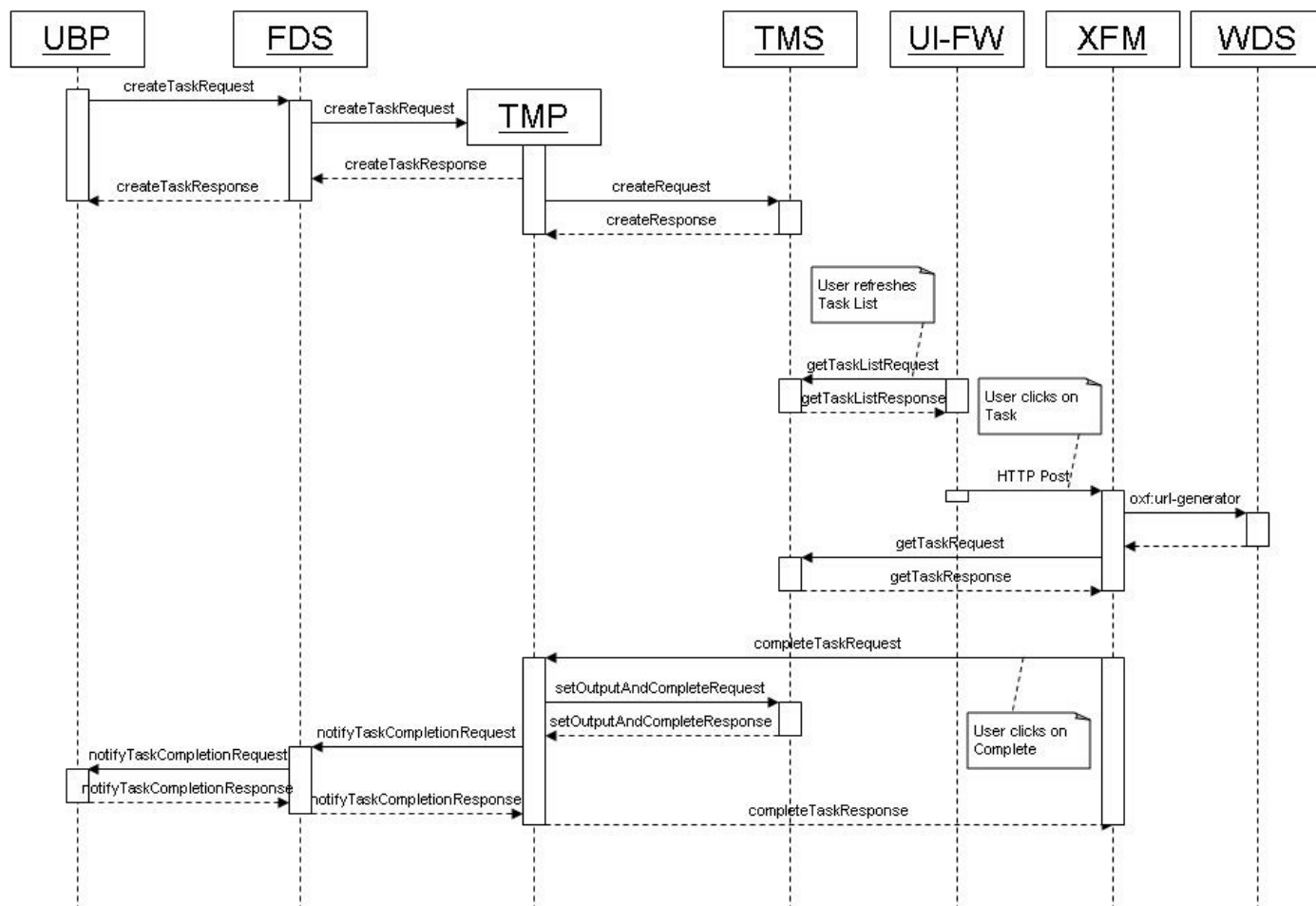
Log In 

View Attachments (1) Info

 Browse SpaceAdded by [Jacques-Alexandre Gerber](#), last edited by [Jacques-Alexandre Gerber](#) on Nov 21, 2006

Creating And Completing a Task

This page explains how the different Tempo components are used when creating and completing a task. This is represented by the sequence diagram below:



In the above diagram, the acronyms mean:

- UBP: User Business Process. That's the process that create the task. It typically is a BPEL process, but in fact could be any application. It makes a web service call to create the task, and provides a web service operation to complete the task.
- FDS: Form Dispatcher Service. Acts as a proxy between UBP and TMP. See below why it's needed.
- TMP: Task Manager Process. The BPEL process responsible for managing the task lifecycle. It is instantiated by the reception of the createTaskRequest process from FDS.
- TMS: Task Management Services. This is a web service providing the workflow actions, persisting tasks in the database and handling security (through the Security Framework, not shown in this diagram).
- UI-FW: UI Framework. Shows the task list and forward that form management to the form manager (XFM).
- XFM: XForms Manager. This is one implementation of a Form Manager, for XForms forms.
- WDS: Workflow Deployment Service. Manages the storage and access to the forms.

Note that creating and completing a tasks are completely separate. The Task ID is used to correlate the createTaskResponse message with the notifyTaskCompletionRequest message. The Task ID is generated by TMP.

In this page, to make it simpler, we assume there is no chained execution. TMP has additional logic to handle chained execution.

Creating a Task

This starts by a web service call to FDS. FDS then routes the request to TMP. TMP then creates the task by calling TMS.

FDS is required in order to manage the "any process" to "one TMP" relationship, which cannot be done in BPEL. From TMP perspective, the createTaskRequest message can only be sent by one uniquely defined partnerLink. Every different UBP would be a different partnerLink. To solve this problem, FDS is a servlet that can handle the any-to-one relationship by accepting all createTaskRequest messages going to /fds/workflow and dynamically changing the namespace to be what TMP expects and then routing it to TMP.

When TMP returns the createTaskResponse, it is also sent to /fds/workflow and therefore captured by FDS, which changes the namespace back to what it was. The namespace is found in the createTaskResponse message and was mapped from the createTaskRequest. This is why the namespace is required to be mapped in the UBP when creating a task. It allows to be passed in the createTaskResponse back to FDS so that FDS can build the createTaskResponse message that a specific UBP instance is

waiting for.

Completing a Task

When a user logs in or refreshes the task list, the UI-FW makes a call to TMS to retrieve the updated task list. When clicking on a task, the request is routed to the corresponding Form Manager. At the moment, UI-FW posts only to XFM, but it could be extended to dynamically select a form manager based on the task properties. XFM then calls TMS to get all the task details. It also calls WDS to retrieve the actual form. With all this information, XFM renders the form with its input data, as well as the button to let the user complete the task or any other workflow action XFM would support.

When the user clicks the Complete button, XFM calls TMP to complete the task. Note that here FDS does not need to be involved as it's always the same XFM that calls the same service offered by TMP. This call however is correlated thanks to the task ID.

Then TMP changes the task output and its state by calling TMS.

Then it forwards the task output back to the UBP through FDS. Here again, FDS needs to be involved for the same reason. The Task ID is used for correlation.

Finally, TMP sends the responds back to XFM after receiving the response from UBP.

Comments

Site powered by a free **Open Source Project / Non-profit License** ([more](#)) of [Confluence - the Enterprise wiki](#).
[Learn more or evaluate Confluence for your organisation](#).

Powered by [Atlassian Confluence](#), the [Enterprise Wiki](#). (Version: 1.4.1 Build:#212 Jun 02, 2005) - [Bug/feature request](#) - [Contact Administrators](#)