NTier Workflow Review

Neil Gordon

Contents

1 Version Info

1.1 Document Version 0.05

2006/01/19

1.2 Compiere 2.5.2d

Information is specific to 2.5.2d and with some changes will apply to earlier or later versions.

2 Important notes

- Always run *cache reset* after changing anything on the workflow
- Close the window you are working with and reopen, for the changes to become active
- Always remember to regenerate the model classes (GenerateModel) if you change the table structure.
- Users involved in workflow should be linked to a business partner. The business partner, under the tab 'Employee' , should have both 'Sales Rep' and 'Employee' ticked. The ***user*** record should belong to client 0 (system) if you wish the user to be a 'Workflow Responsible'

2.1 Conditional Transitions - Boolean

It is worth noting up front that when comparing with boolean on conditionals, 'true' and 'false' must be used *NOT* 'Y' and 'N'.

See the screenshot for details

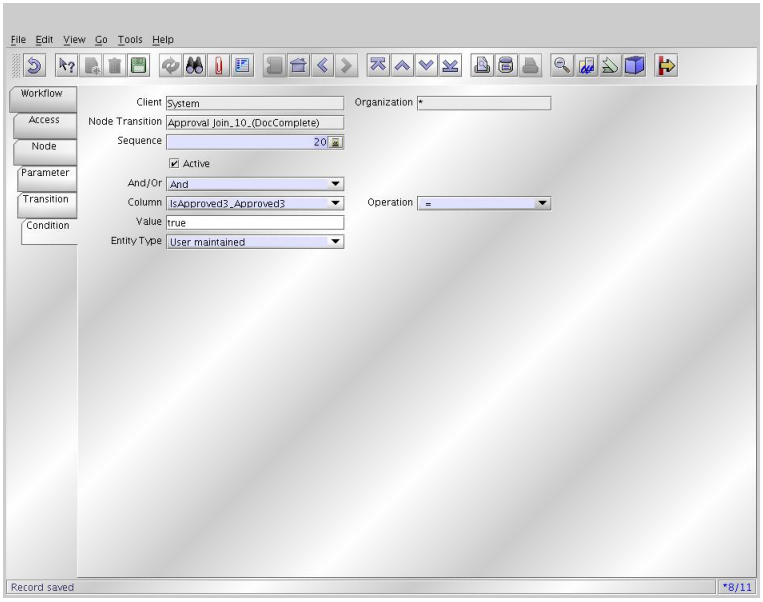

Figure 2.1: Conditions - Use 'true' and 'false' for boolean

3 Defining a new table which implements DocAction

In order to see what is going on behind the scenes a new table was created ZZ_WF. This table initially only has two fields - DocStatus and DocAction.

3.1 Goal

The purpose of this section is to define a new table with a simple workflow which simply changes the document status to 'In Progress'

3.2 ZZ_WF Table

3.2.1 Data Dictionary

Table Definition

Header
Details
Reference for DocAction incorrect!
There is an error in the screenshot!
You must use the '_DocumentStatus' reference NOT the 'DocAction' as is shown in C_Order. If you change this regenerate the model in Compiere!

Window definition (view)

Window Header
Window details
Note that the 'DocStatus' field should be set to read-only

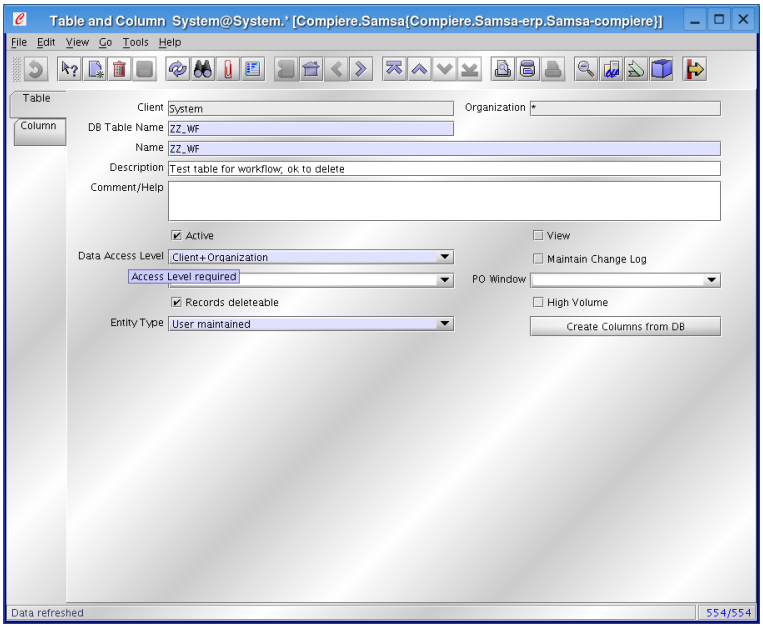

Figure 3.1: Data dictionary - Header

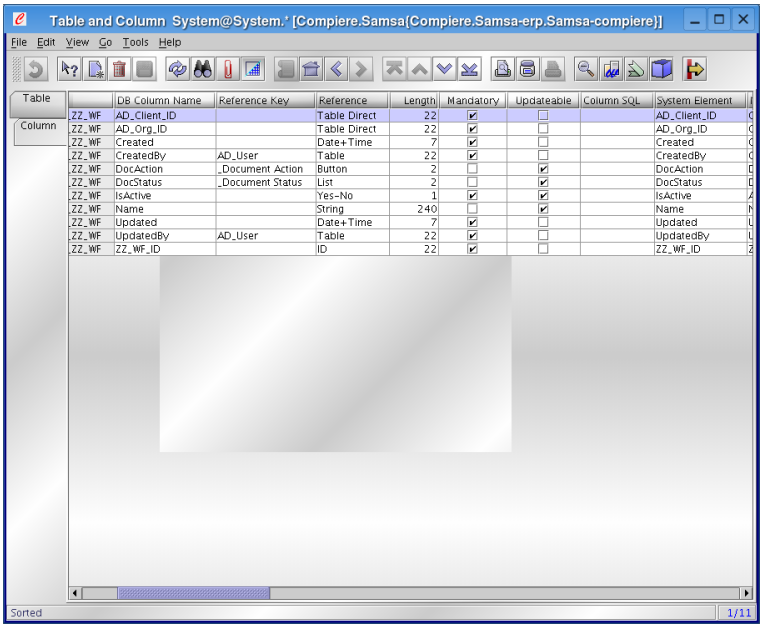

Figure 3.2: Table def - Details

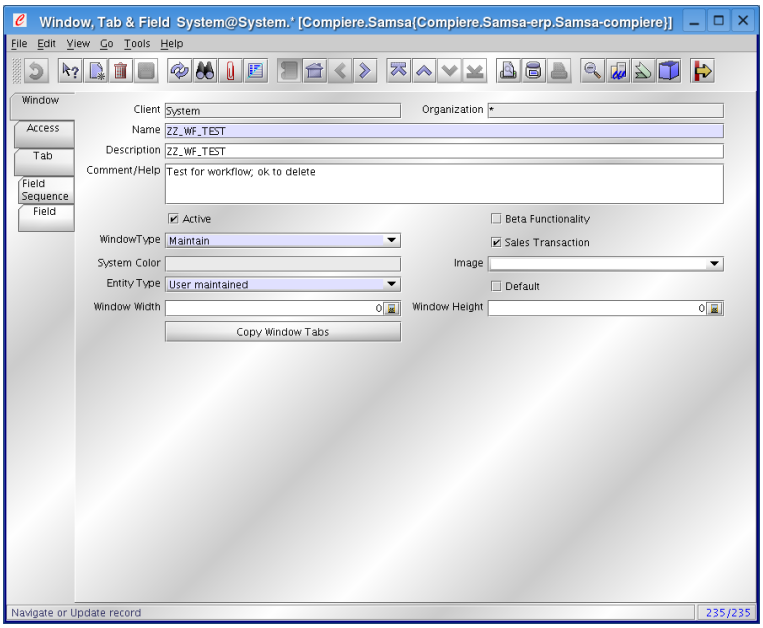

Figure 3.3:

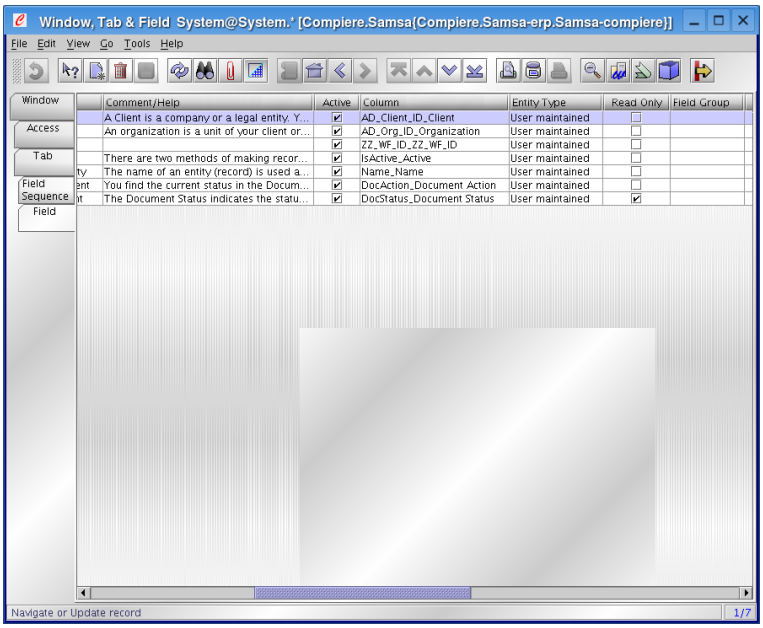

Figure 3.4: Window details - fields of interest

3.3 What happens next?

When you click complete you get the expected window with 'Complete' and 'Void' populated. However clicking on the Complete does nothing???

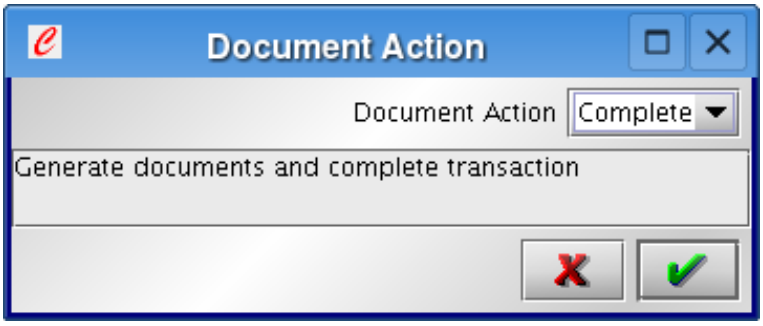

Figure 3.5: DocAction popup

3.3.1 The mistake

No process (report & process) was defined on the table definition for the field 'DocAction'

3.4 The class responsible for populating the list of choices...

Is VDocAction

3.5 Defining a report & process

Define a report & process and add it to the data dictionary.

The only field necessary is the Workflow field (the workflow should already be defined).

3.5.1 The process attached to DocAction

Attach it in the table definition to the DocAction field

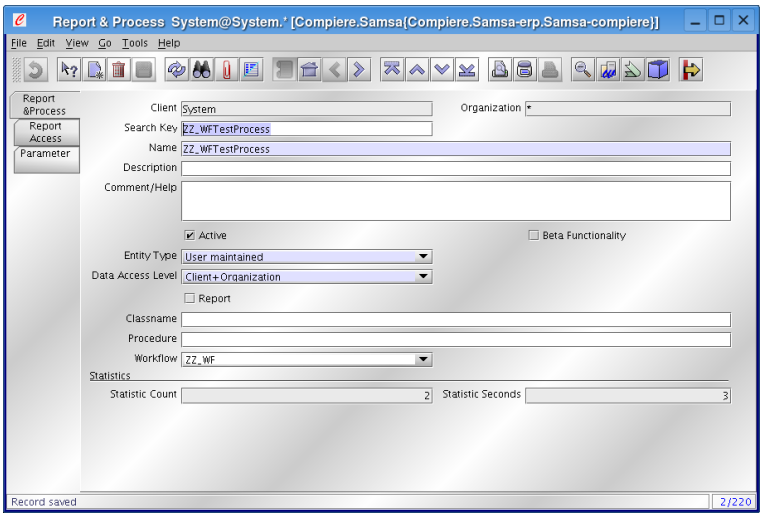

Figure 3.6: ZZ_WFTestProcess

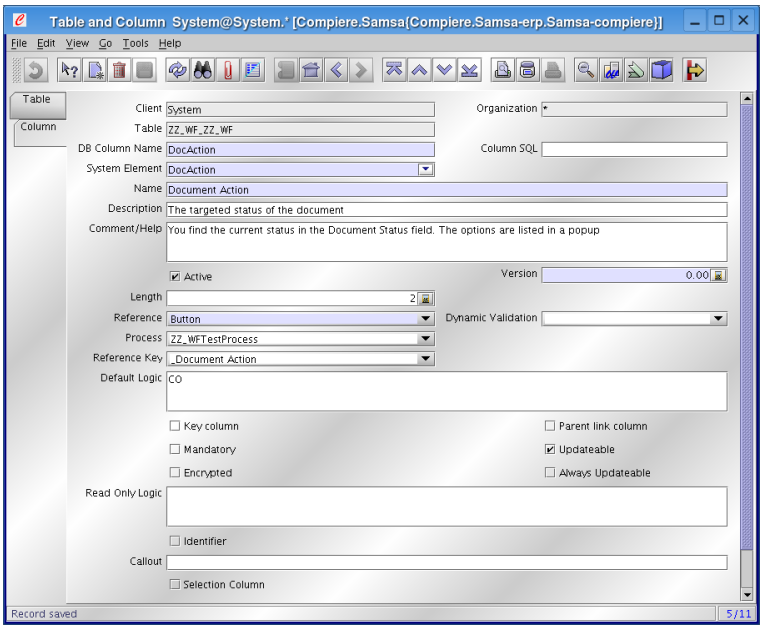

Figure 3.7: The dummy process attached to DocAction

3.5.2 The workflow

The workflow is a simple one step process that simply 'Prepares' the document

Workflow Editor

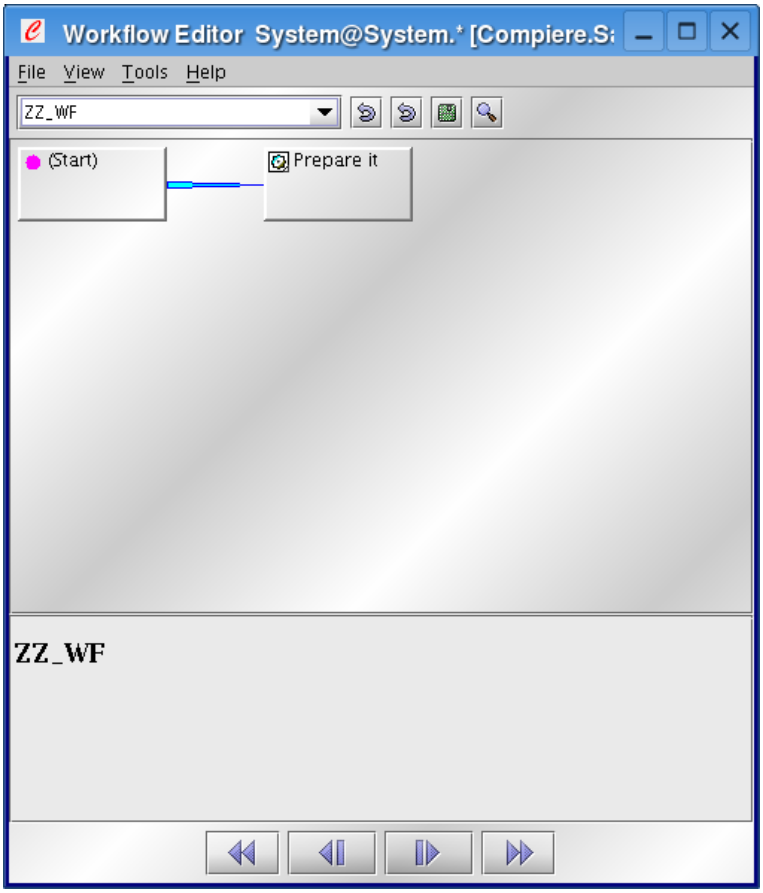

Figure 3.8:

Workflow definition (main tab)

Note: Next node is '(Start)'

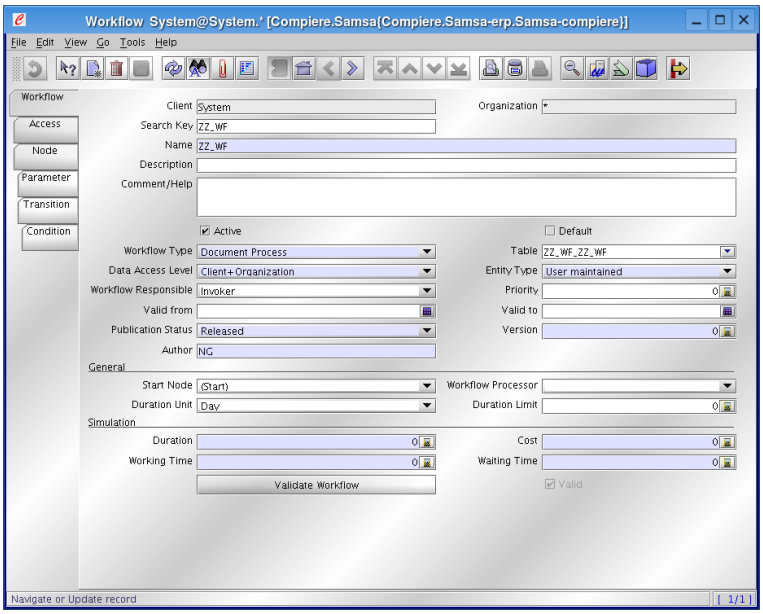

Figure 3.9: Workflow definition (main tab)

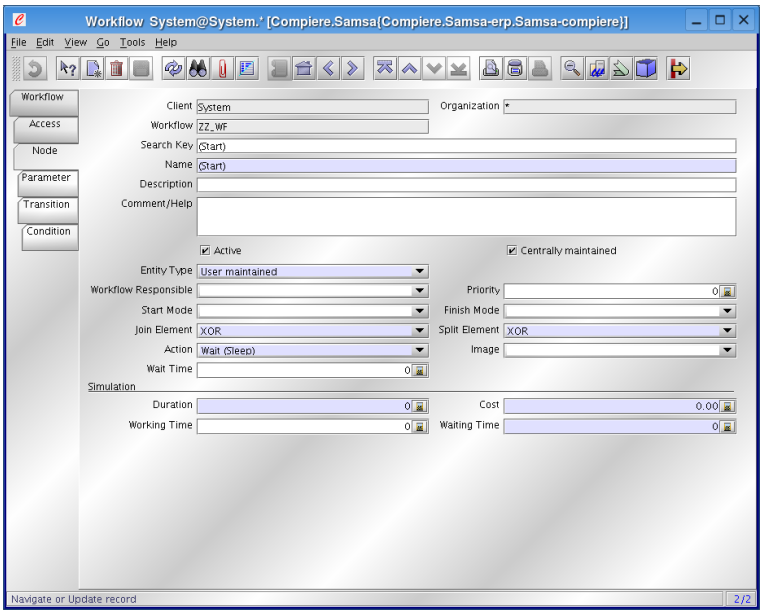

Figure 3.10: Workflow node start

Workflow Node: (Start)

Transition: (PrepareIt)

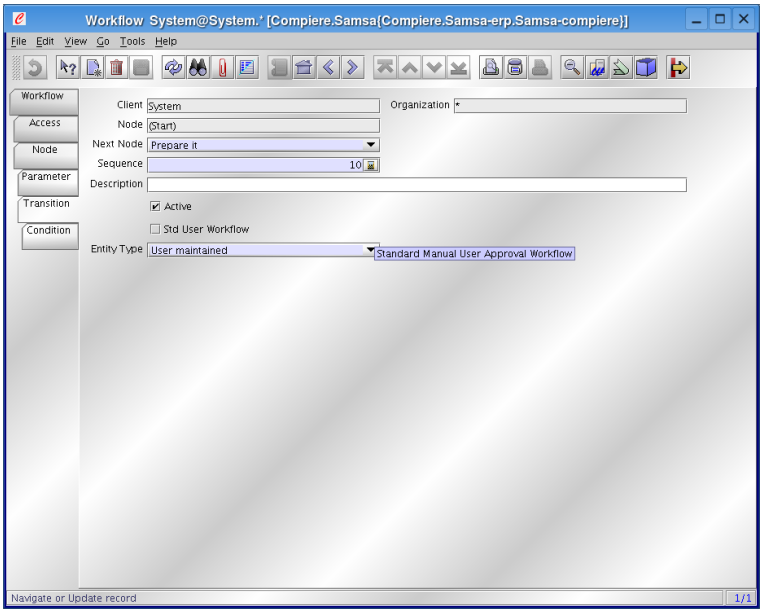

Figure 3.11: Transition: (PrepareIt)

Workflow Node: (PrepareIt)

Transition from PrepareIt

Empty

3.6 Defining the PO class

The java object defined to support this workflow needs to be called 'MWF' (in 2.5.2) and 'MZZWF' (in 2.5.1).

NB: The MWF class must implement the DocAction interface.

e.g.:

public class MWF extends X_ZZ_WF implements DocAction

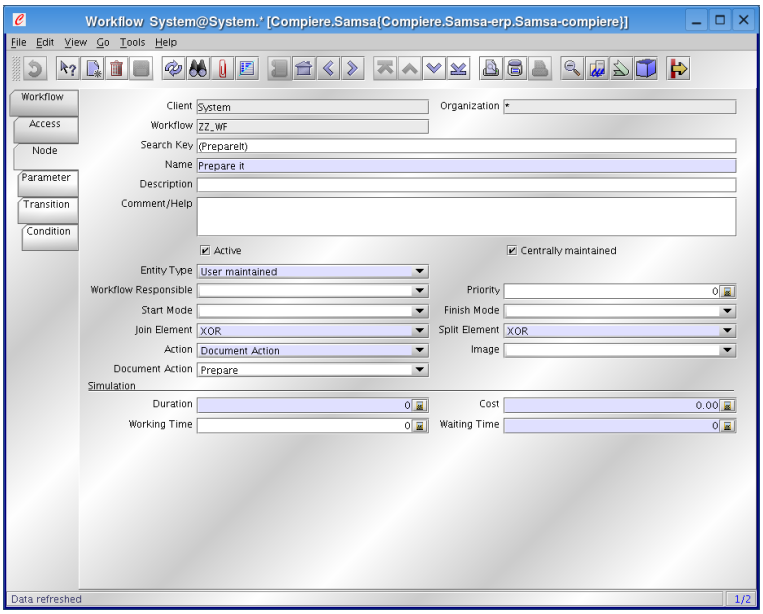

Figure 3.12: Workflow Node: (PrepareIt)

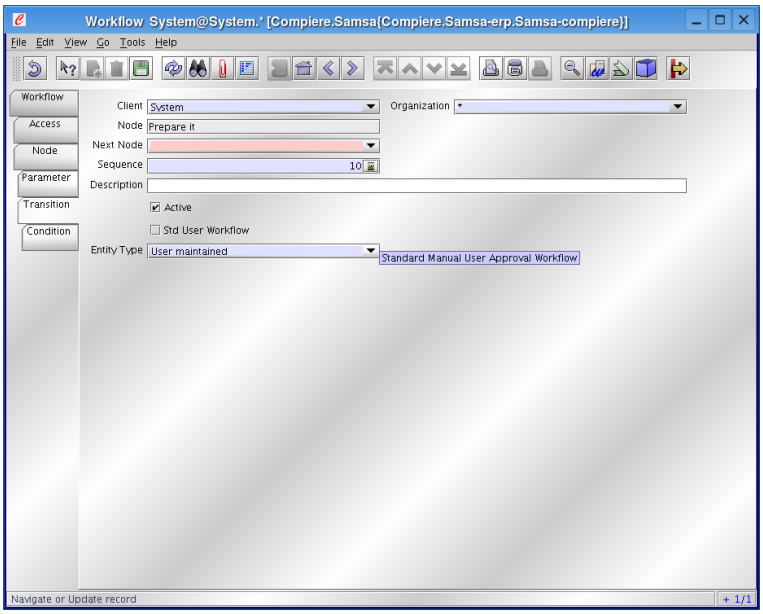

Figure 3.13: Transition from PrepareIt

3.7 Defining the 'processIt' method in your PO class

In the MOrder class 'processIt' is defined as follows:

```
public boolean processIt (String processAction)
{
m_processMsg = null;
DocumentEngine engine = new DocumentEngine (this, getDocStatus());
return engine.processIt (processAction, getDocAction(), log);
} // processIt
```

We define the same method in our MWF class. This is the method that is *first called* when a document action is called against a PO.

3.8 Defining the prepareIt method

This is the method ultimately being called by processIt.

We are expected to return the docstatus in this method as follows:

```
public String prepareIt() {
log.fine('prepareIt');
System.out.println('prepareIt');
return DocAction.STATUS_InProgress;
}
```

NB: The field DocAction will only accept valid values per the reference list which was defined on the table field!

3.9 Defining the getProcessMsg() in your PO

You can set the process msg elsewhere in your code. It will display on the toolbar, e.g. when the 'prepareit' has completed.

```
/**
* Get Process Message
* @return clear text error message
*/
public String getProcessMsg()
{
return m_processMsg;
} // getProcessMsg
```

3.10 PrepareIt - Success!

If everything has been setup correctly to this point clicking on 'Complete' will trigger a workflow which calls 'processIt' which then calls your 'prepareIt' method which returns 'In Progress' which is the document status.

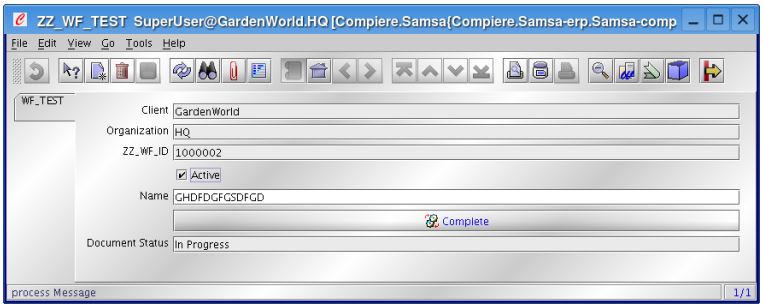

Figure 3.14: Prepare has run successfully

3.11 Workflow activity

Shows the detailed steps

3.11.1 Header

Notice that the process message has been appended to the text

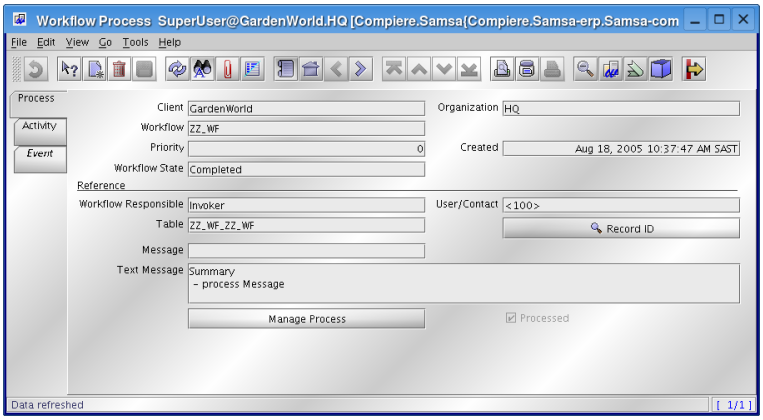

Figure 3.15: The activity header

3.11.2 Detail - Start

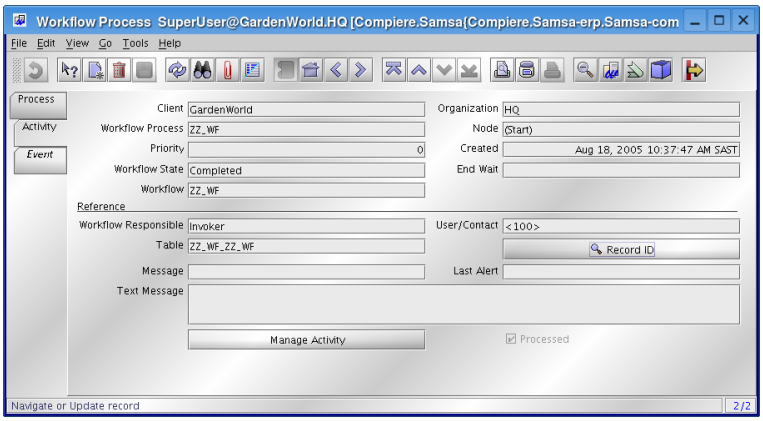

Figure 3.16: Detail - Start

3.11.3 Detail - PrepareIt

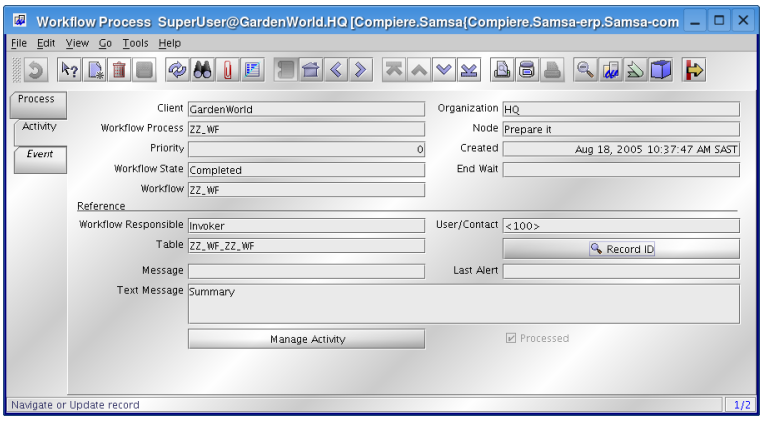

Figure 3.17: Detail - Prepare It

4 Extending the previous example

We will now extend the previous example to include Approval.

Add the column 'ApprovalAmt' to the ZZ_WF table. Also add a column and element (yes/no) called IsApproved2. Add both of these fields to the window as well.

4.1 Screenshot of modified screen

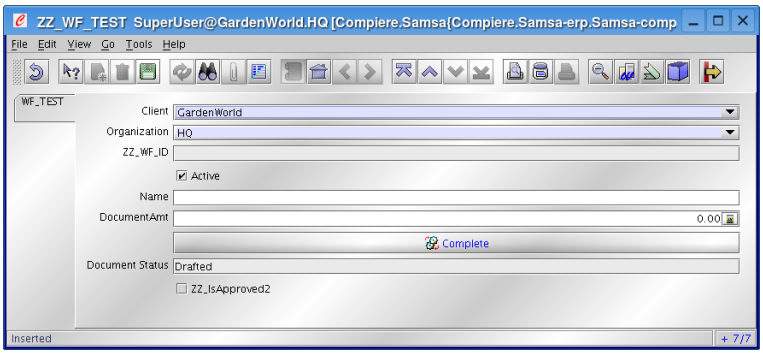

Figure 4.1: Modified test screen with amounts and approval checkbox

4.2 Excercise

Change the workflow for the ZZ_WF table as per the workflow editor screenshot attached. If the user enters an amount over 100 then approval must occur otherwise it is automatically completed

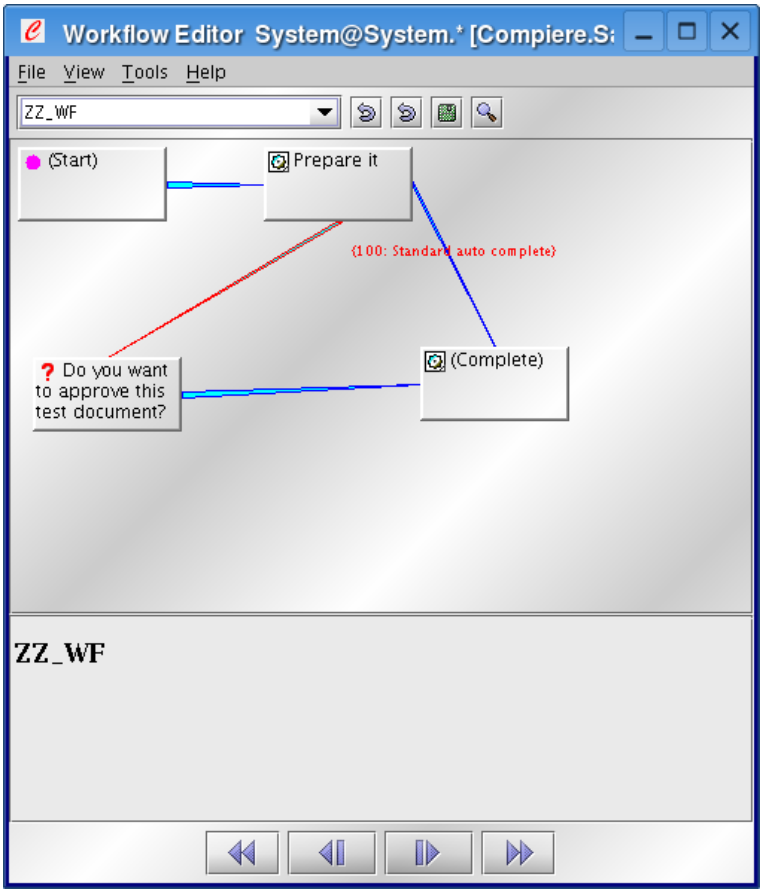
Workflow Editor System@System.* [Compiere.S:
File View Tools Help
ZZ_WF
(Start)
Prepare it
(100: Standard auto complete)
Do you want to approve this test document?
(Complete)
ZZ_WF

Figure 4.2:

4.2.1 Modify code in MWF model class

You will need to modify completeIt as follows:

```
public String completeIt() {
log.fine('completeIt');
System.out.println('CompleteIt');
return DocAction.ACTION_Complete;
}
```

4.2.2 Final product

- The authorizer will need to approve the doc
- The document will now be 'Completed'

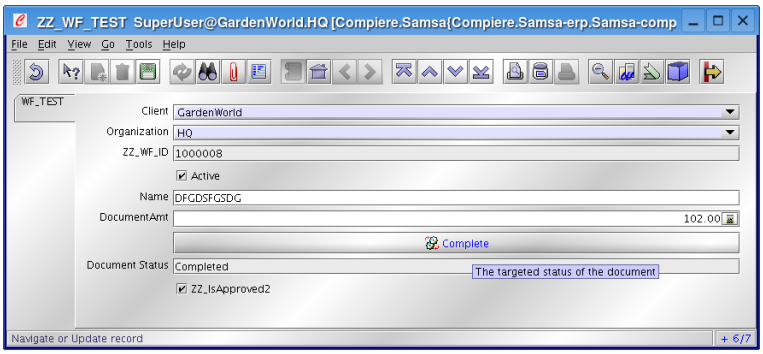

Figure 4.3: After authorizing the doc is completed automatically

5 Building again on the previous example: Two approvers

5.1 Goal

We wish to have two approvers in parallel. No code modification should be made. Only once both approvers have approved the doc, then the document is completed.

5.2 Columns required

ZZ_IsApproved2 and ZZ_IsApproved3 should be added to the table ZZ_WF. We avoid using IsApproved column because it is suspected there is some logic in Compiere using this field. (Still to be discovered)

5.3 Workflow Editor: Overview

If the labels are setup on the transitions then the workflow editor becomes more descriptive. You need to login as system.

Tip: When you have reorganized the layout to look good you can click the 'save' button on the toolbar it will save the organization

5.4 Nodes - Description

- (Start) start node 'Sleeps' and then transitions directly to the (Prepare) Node (no other transitions defined for 'Start')
- (PrepareIt) calls the document action Prepare. If successful, then 3 transitions are processed (with the 'split' set to 'AND' so that all are

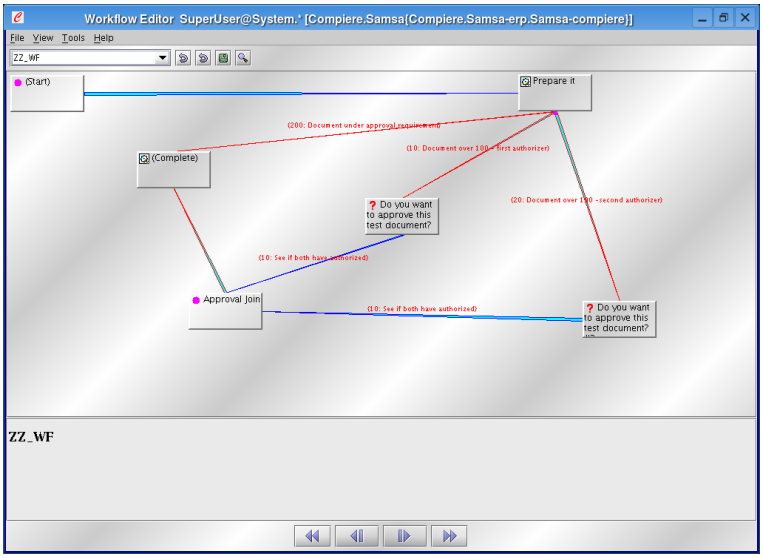

Figure 5.1: Workflow Editor: Overview

processed). First approval, second approval, and (Complete). The first two approvals have conditions attached on whether the document amount is greater than 100. The last one (Complete) will only process if the document amount is less than 100. This provides auto-completion of the document under a certain value.

- Do you want to approve this test document? (#1) - is a user choice bound to IsApproved2. If the user selects YES then the transition is to 'Approval Join'
- Do you want to approve this test document? (#2) - is a user choice bound to IsApproved3. If the user selects YES then the transition is to 'Approval Join'
- 'Approval Join' sleeps and then transitions to 'Complete', conditionally , if both IsApproved2 and IsApproved3 are set. So it will be executed when either of the documents are approved but it will only transition to Complete when both flags are set

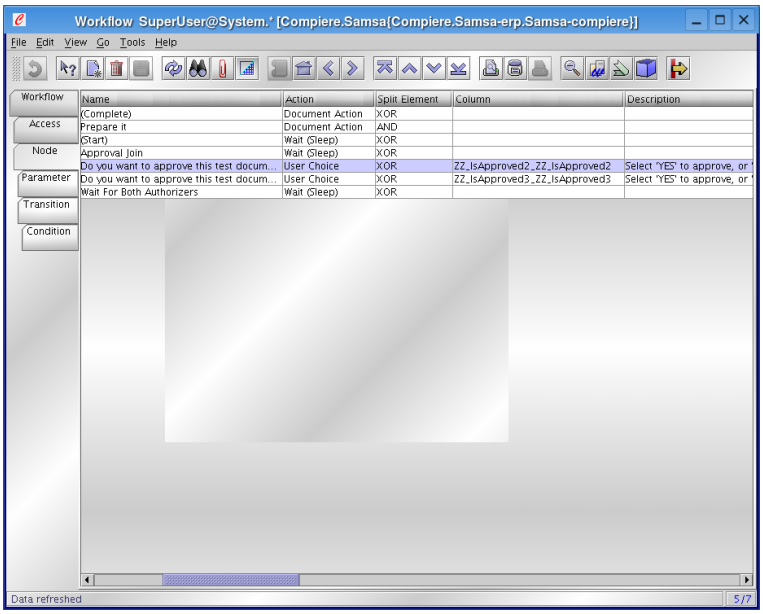

Figure 5.2: Nodes - Two Parallel Approvers

6 Analysis of the Requisition example workflow

The requisition workflow is used as an example workflow within Compiere. They have chosen it because it has a one document type to one table relationship (IE M_Requisition). A workflow based on another table e.g. C_Order would be more complex because it stores both SO and PO.

6.1 Compiere user manual 252c - Reference

To be read together with this analysis:

Refer page 21 under section 'Common Functions and Commands: Workflow' (page 51 on Adobe) entitled 'Document Process Workflow' for for step by step details on this process from a user's perspective.

6.2 compiere.com description of workflow

Business Process Management

Workflow Management Coalition Workflow is usually defined as 'Steps involving People' - Business Process Management is usually defined as 'Workflow and System Activities'.

Compiere fully supports Business Process Management (BPM) and is based on the Workflow Management Coalition and OMG standards. In the following, we use the term Workflow to include BPM capabilities.

In contrast to all known ERP and CRM applications, Workflow is not 'on top' of the application, but Compiere is based on Workflow. The Compiere Workflow Engine is Compiere's core transaction management. That means that all processes in Compiere are automatically workflow enabled and easy to extend and modify. As workflow is completely integrated, Compiere workflows are easier to maintain and provide much more functionality then the external or ad-on workflow offerings of other ERP and CRM vendors.

Compiere Workflow

Types of Workflow

Compiere provides three types of workflow:

* General Workflow - Provides guidance and step-by-step instructions for achieving a task. Examples: Setup Wizards or Month End procedures. A user starts them from the menu.

* Document Process Workflow - Started when processing any document. You would extend these workflow type for approval situations. Example: Special approval for orders over a certain amount.

* Document Value Workflow - The workflow is automatically started when any entity fulfils a user defined condition. Example: Start credit approval for a new Business Partner.

Node Actions, Transitions

A Compiere Workflow Node (Step) can have the following Actions

* Automatic Process - Any Process, Report, Task, Workflow, Document Action,

planned: Web Services, etc.

* User Action - Any Window, Form - User needs to confirm Completion

* Set Variable - Any Column to Constant or Variable

* User Choice - Any Choice (e.g. approval), List selection

* Wait (Sleep) - can also be used for Start/End/etc.

Transition between nodes can optionally have conditions. Multiple transitions from a node allow parallel processing. This allows to model very complex scenarios.

Approvals, Responsible

You can define your own approval hierarchy or use the defined orgraization hierarchy.

A workflow responsible can be a human (specific or invoker), a group (role), or (supervisor) of an organization. Different workflow nodes/steps can have a different workflow responsibles.

Priority, Escalation, Alerts

Compiere provides dynamic priority management, allowing to use the workflow engine for call center routing and priority based customer support.

You can define escalation rules due to inactivity and send alerts to the workflow resonsible and/or the supervisor.

6.3 Workflow type: Document Process

This workflow example is based on 'Document Process'. It takes place when 'Process' is clicked on the requisition (or document).

6.4 How the workflow is activated from Compiere

6.4.1 DocAction column on table

This is the button 'Process' which is clicked on the Requisition form. This is linked to a 'Report and Process', which should actually be called 'Report, Process and Workflow'.

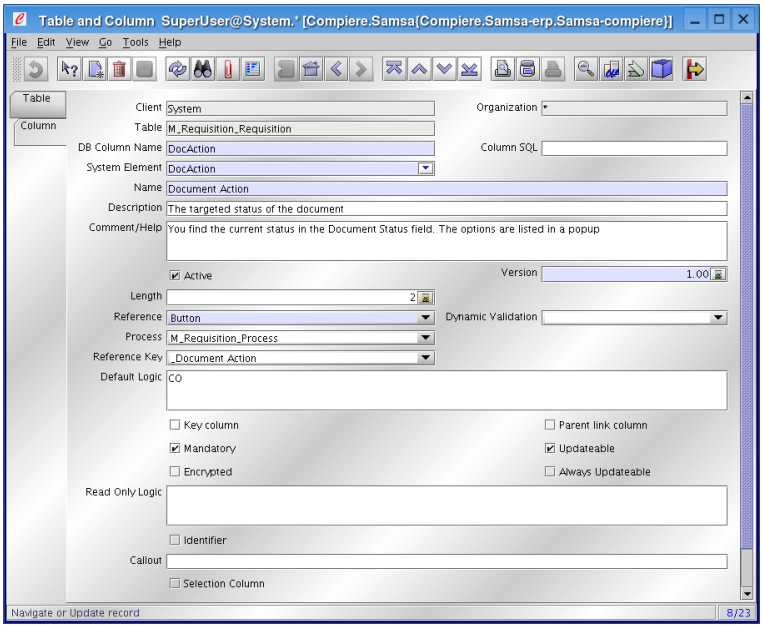

Figure 6.1:

Report And Process

- This process activates the workflow. All it needs is the field 'Workflow' to be defined in order to work. None of the other fields (classname, procedure name,etc) should be defined, as they are ignored
- Search key: M_Requisition_Process
- Name: Process Requisition
- Workflow: Process_Requisition
- ID=273

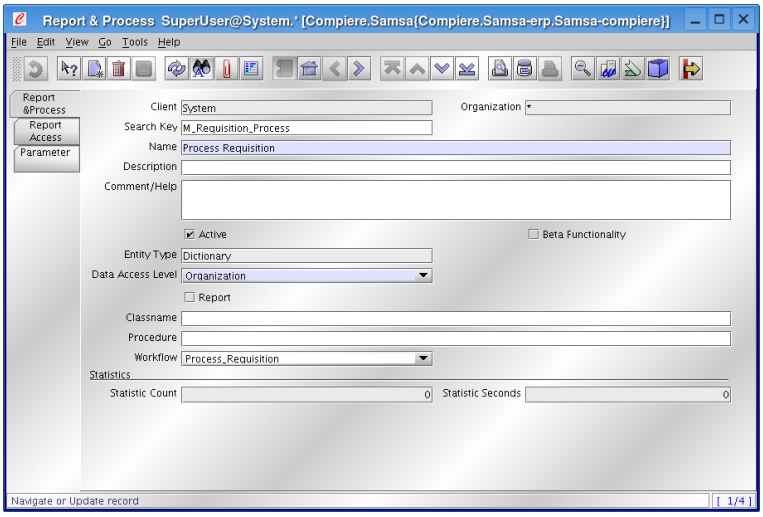

Figure 6.2:

6.5 Workflow Editor: Requisition

Compiere provides a graphical tool for viewing a workflow. Choose the 'Workflow Editor' from the menu

6.6 Screenshot of the requisition example workflow

The 'User/Contact' field is used as the 'Workflow Responsible'

6.6.1 Who is responsible for authorizing?

NB: The person responsible for this workflow is the person under User/Contact. *This is detailed in the user manual.*

TODO: Add screenshots of the Role to explain why the ultimate workflow responsible here is GardenAdmin

Difference between the Requisition and the Purchase Order: The requisition appears to display User Roles. The purchase order displays user/contacts for the selected business partner (see User/Contact tab on Business Partner). So we cannot expect the two to behave identically so far as roles go.

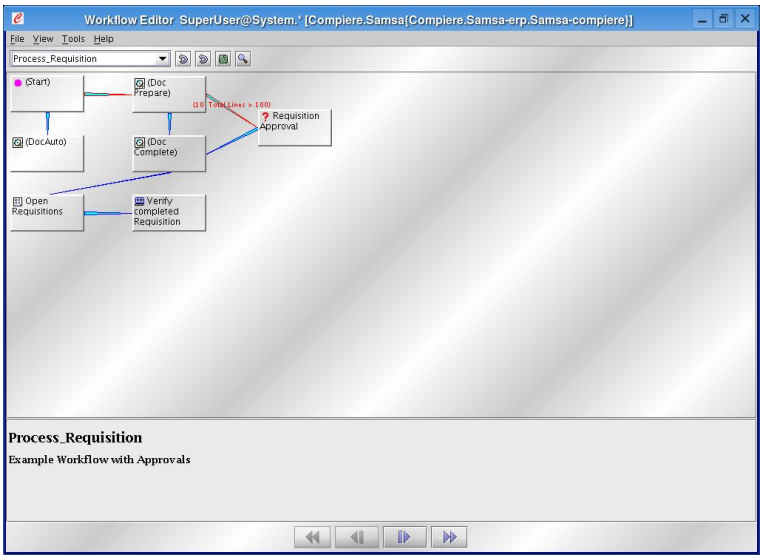

Figure 6.3:

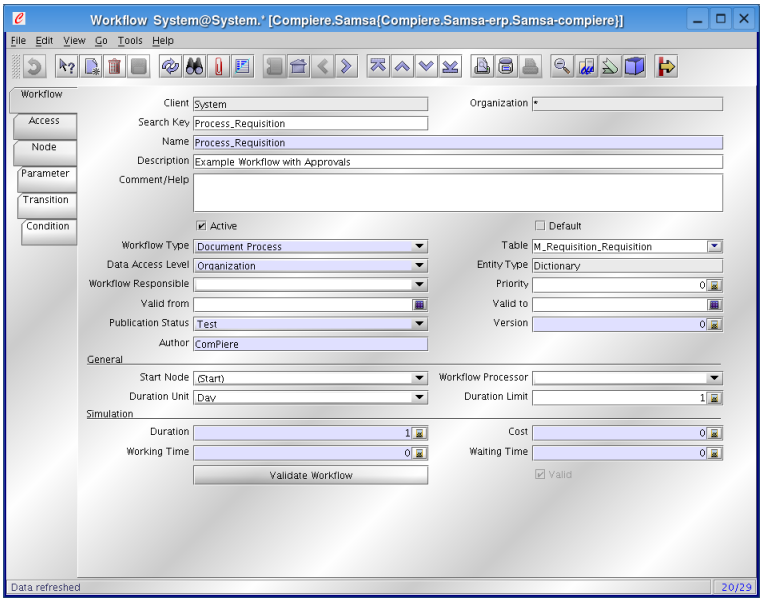

Figure 6.4:

6.7 Start node: (start)

On this node definition, the Start Node is set to '(Start)'

6.8 Node: (Start)

The start node of this workflow is set to '(Start)'.

The 'Split' field defines whether all or just one of the transitions will be executed. XOR means that only the first valid transition is executed, and 'AND' means all the transitions are executed.

Currently my theory is that it works like this: With 'Action' being set to 'Wait (sleep)' the process sleeps for 0 seconds (ie does nothing) then goes into the 'Transition' phase.

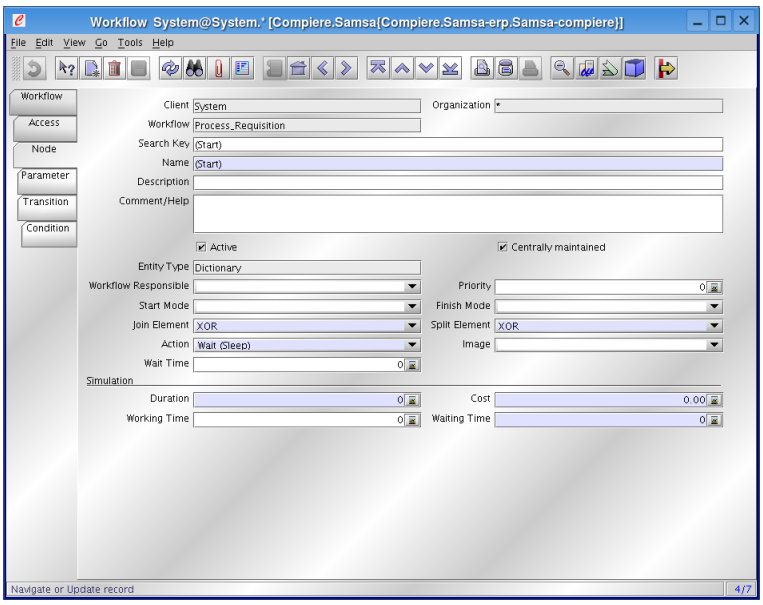

Figure 6.5:

6.8.1 Transition: (Start)

The transition is defined for the (Start) node and can be one of (DocPrepare) or (DocAuto).

The sequence no. for '(DocPrepare)' is 10 and the sequence number for '(DocAuto)' is 100. 'Std User Workflow' is ticked for Sequence 10.

'(DocAuto)' is the node which is processed if none of the other nodes are valid. For example if the user clicks 'Void' this is the node which actually causes the Voiding to take place. If the '(DocPrepare)' transition is valid then '(DocAuto)' is not executed. This is because of the '(Start)' node being set to 'XOR'.

There are no Conditions defined on either of the transitions, so based on the sequence numbers (10 and 100) the (DocPrepare) Node is executed first. Because 'Split' is XOR the node 100 will not be executed if the '(DocPrepare)' node is executed. Since the 'Std User Workflow' tickbox is ticked , this node will ONLY be executed if the Action is 'COMPLETE', and some other conditions which are still undefined in this document. (This has been determined from the source code). The source code needs to be analysed further to determine this.

NB: It looks **Prepare** is not a valid action for a Requisition so the title of '(DocPrepare)' is misleading? A Requisition can only be Completed or Voided.

- Doc prepare is sequenced first before doc auto - Doc prepare is flagged as 'std user workflow'

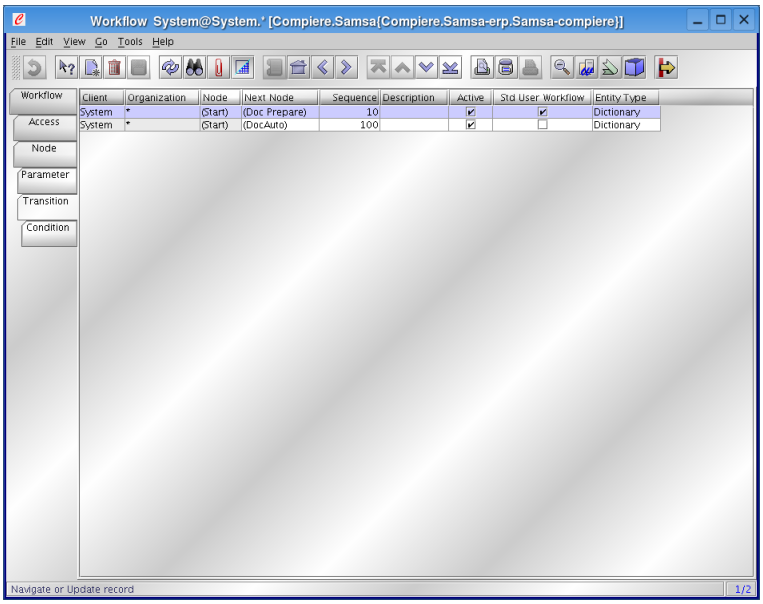

Figure 6.6:

Node: (DocPrepare)

The node (DocPrepare) is defined as 'Action' and 'Document Prepare'. This will be called by the transition 10 which is valid for 'Complete'. There is no Prepare on the requisition.

The current theory is that Prepare is called (does nothing on the requisition) then Transition begins for this node.

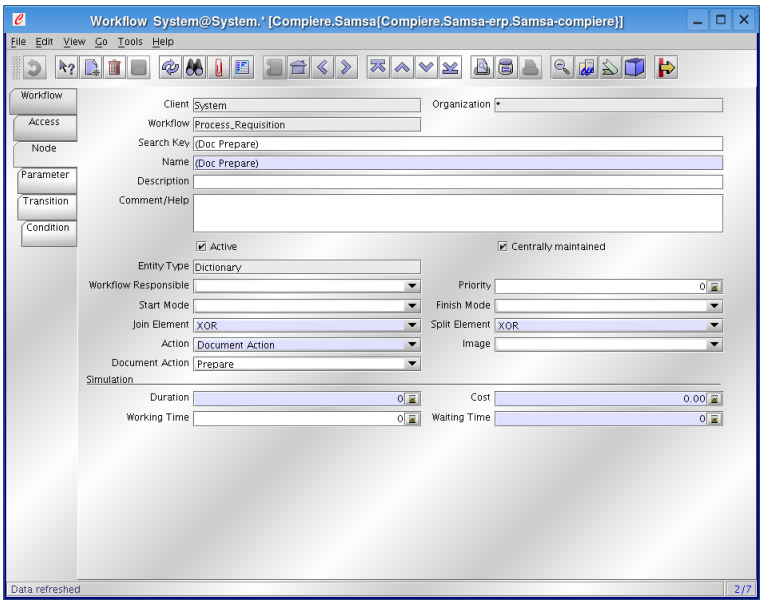

Figure 6.7:

Transition: (DocPrepare)

Two transitions are defined on (DocPrepare): Requisition Approval and (DocComplete). The sequence numbers indicate that 'Requistion Approval' takes place first. However there is a Condition attached to (DocPrepare) which tests the TOTALLINES column of M_Requisition. If the TOTALLINES is > 100 then the 'Requistion Approval' node is executed. Because the node is set to 'XOR' then when approval is required the following node is not executed.

Condition: (DocPrepare) Requisition Approval

If totallines>100

Requisition Approval Node

This node has only been entered if the previous condition was met. The 'action' on the node is set to 'user choice' and the column is 'IsApproved'. So we require some kind of column behind the user's action.

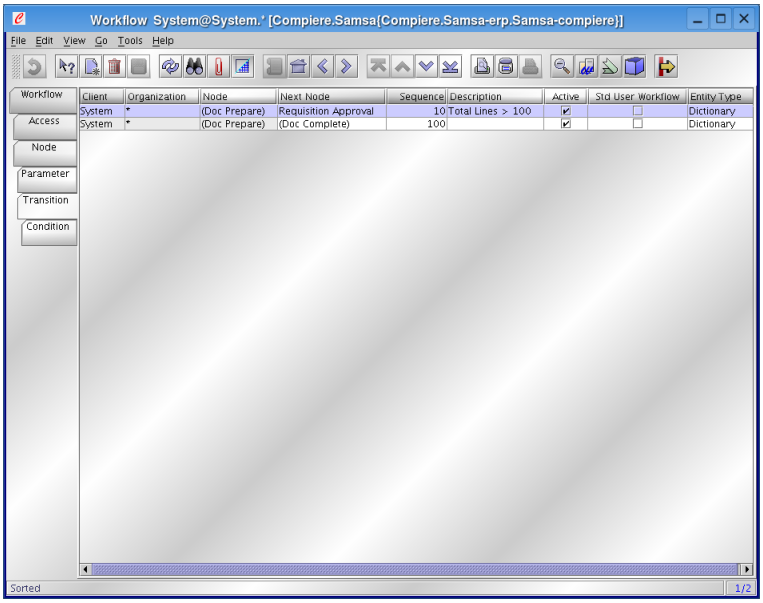

Figure 6.8:

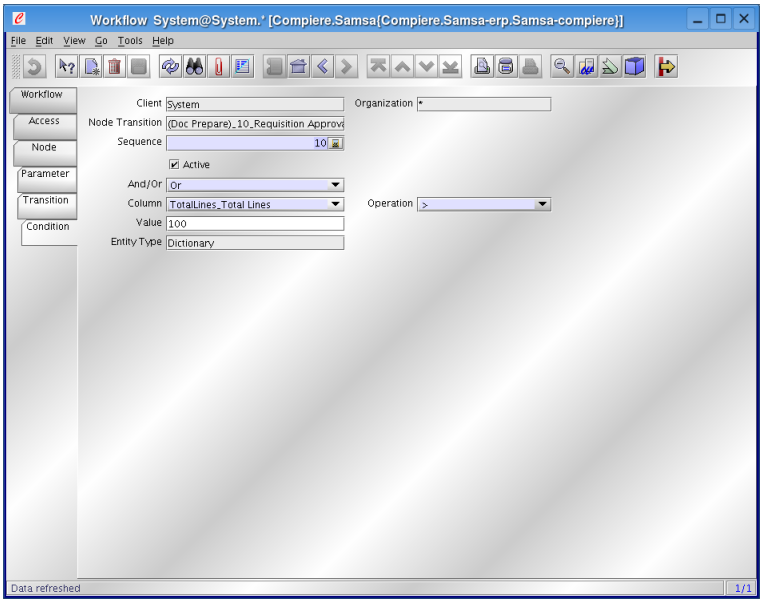

Figure 6.9:

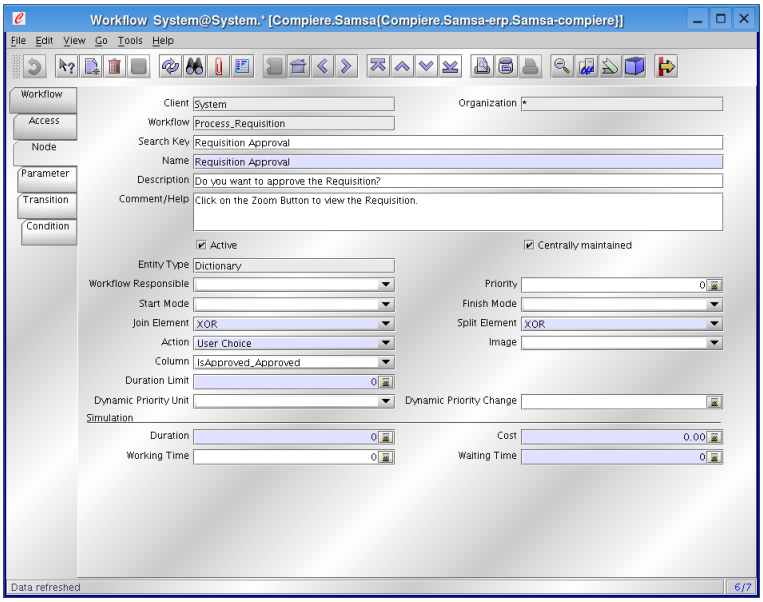

Figure 6.10:

Transition: Requistion Approval

This transition defines that the next step should be '(DocComplete)' ie if the supervisor approved the transaction the document must complete.

Goto (DocComplete) node

Node: (DocComplete)

This node is entered standardly if the condition for approval (>100) was not met. Otherwise it is entered from the successful completion of the Requisition Approval Node.

'Action' is defined as 'Document Action' and the 'Document Action' is defined as 'Complete'

Transition: (DocComplete)

Node: Open Requistions

This node takes place after successful completion of a document and is intended to demo some additional workflow functionality.

Transition: Open Requisitions

Node: Verify Completed Requisitions

This node takes place after successful completion of a document and is intended to demo some additional workflow functionality.

Ends Here

The workflow is complete

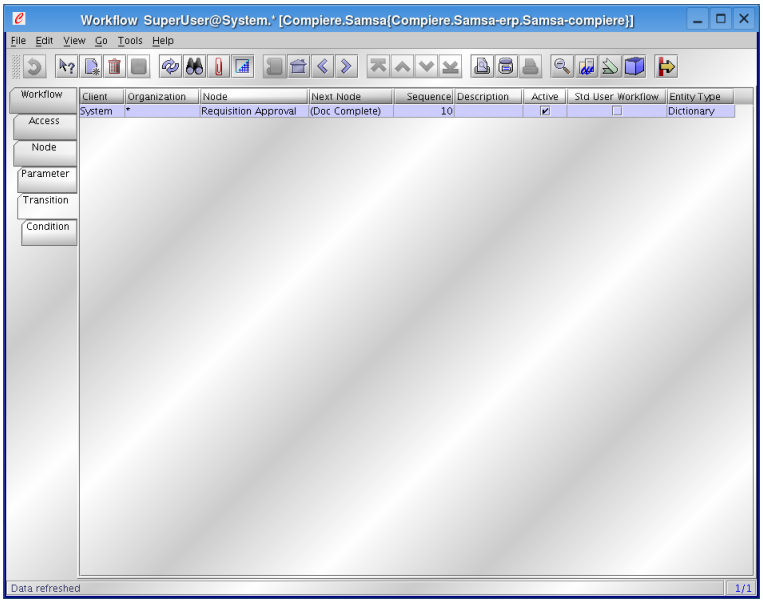

Figure 6.11:

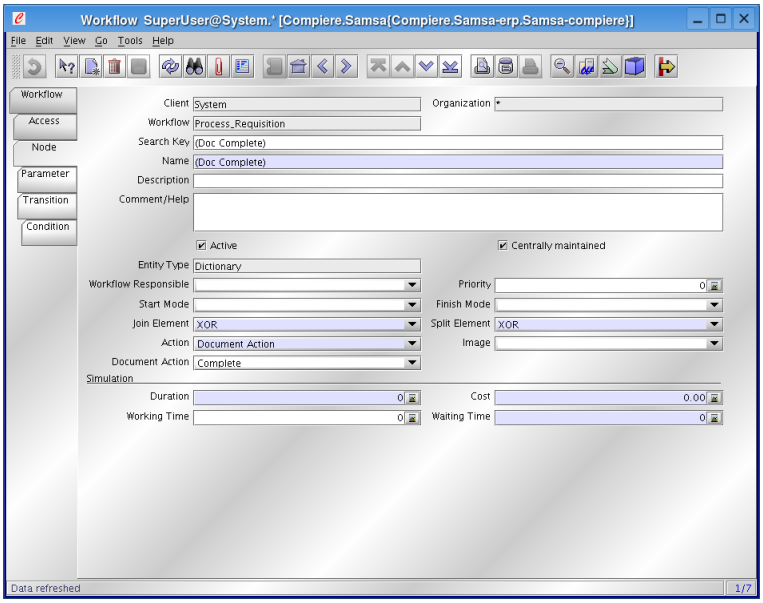

Figure 6.12:

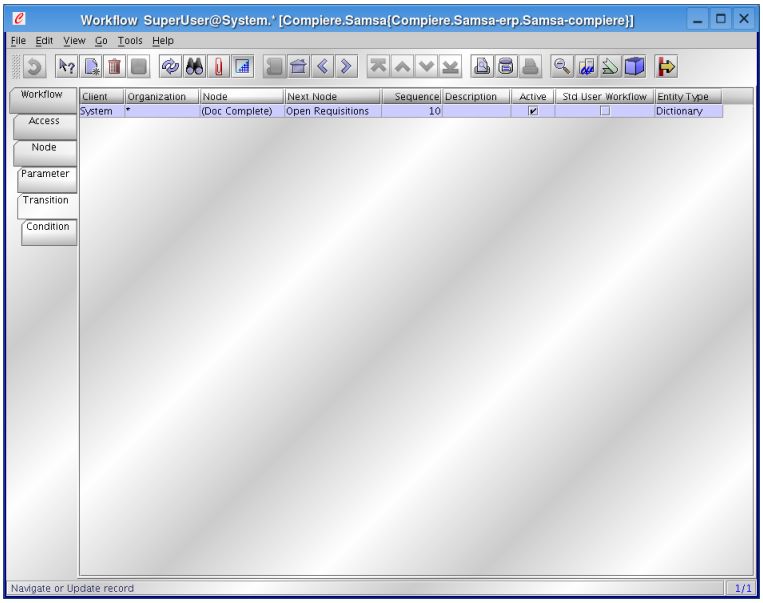

Figure 6.13:

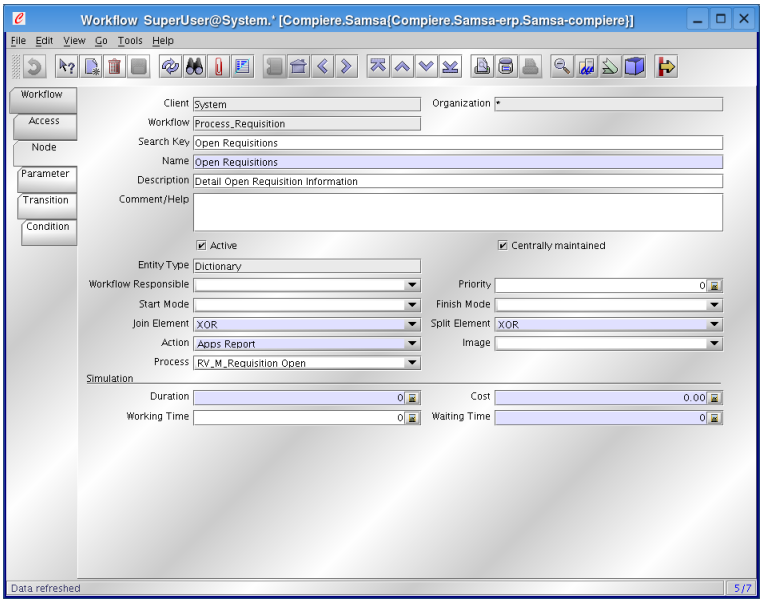

Figure 6.14:

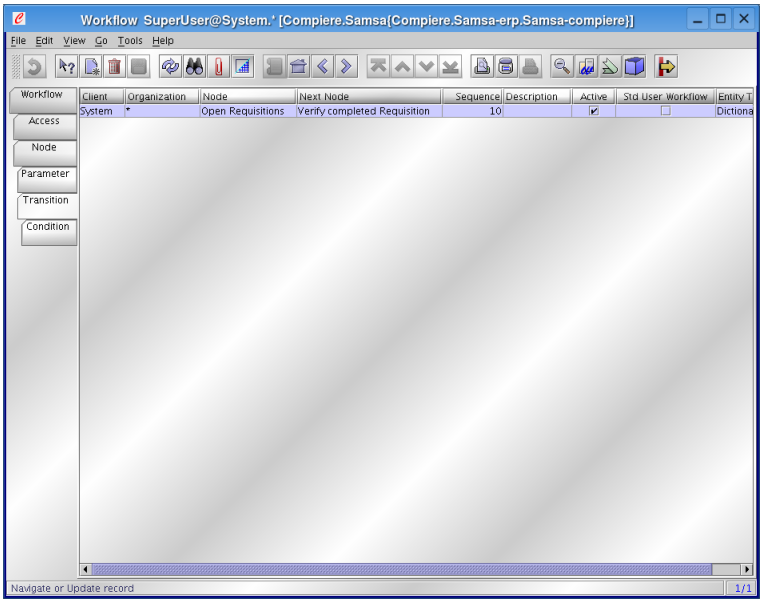

Figure 6.15:

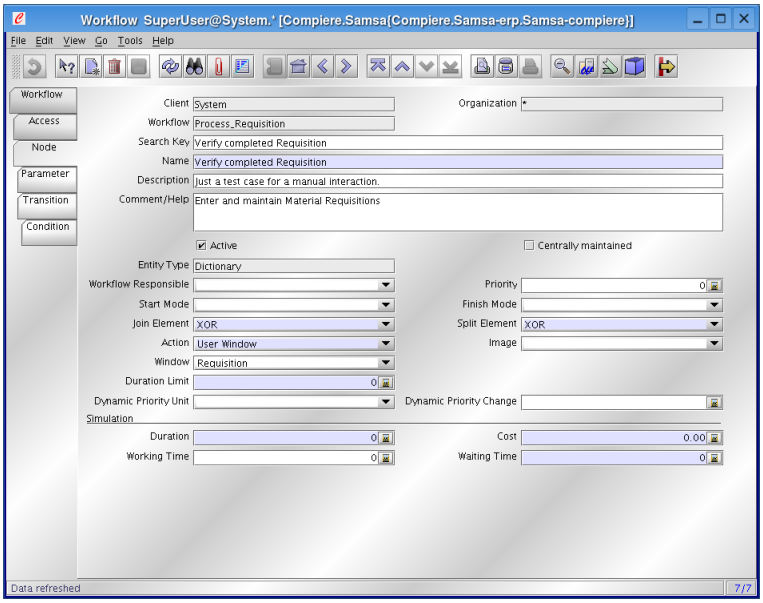

Figure 6.16:

Node: (DocAuto)

This node handles any 'standard' document actions e.g. Voiding which are not covered by the other workflows.

- Action= 'Document Action' = NONE

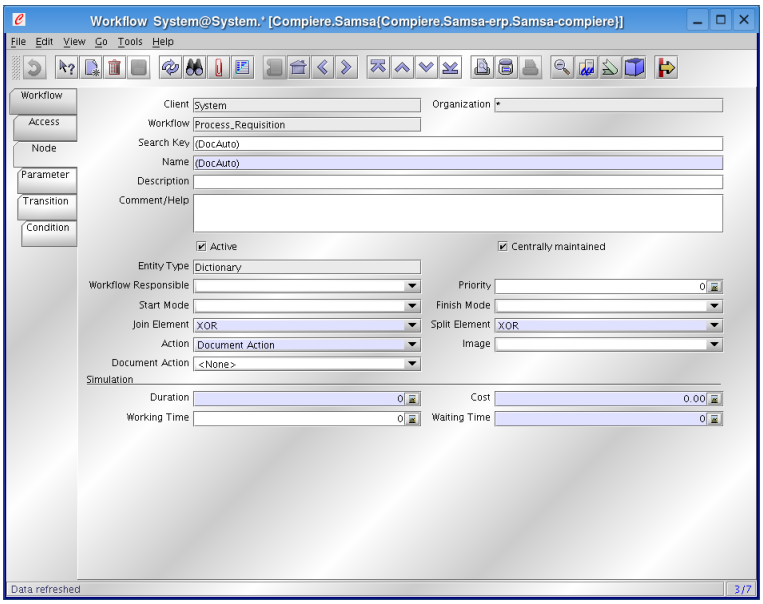

Figure 6.17:

Ends here

6.9 Bugs/problems encountered

6.9.1 Workflow tab at top of Compiere Window

Doesn't appear to refresh properly. Sometimes you need to start another Compiere option before the workflow appears there.

6.9.2 Removing transition leaves condition behind?

Caution/To Be Confirmed:

It appears that deleting a Transition does not always properly remove the associated condition.

6.9.3 When making changes to the workflow..

Be careful to exit your document window (e.g. Purchase Order) and then enter again. Otherwise the changes may not be effective (cache?)

Cache reset (Compiere menu option) might be the best thing to do after your changes.

6.10 Debugging Workflow - Dialog With Jorg

This should give some insight as to Jorg's strategy in general. Everything that is learnt is from trial and error and inference. Otherwise you must attend the training. See below:

Date: 2005-05-07 20:53

Sender: jjankeSourceForge.net Subscriber and DonorAccepting Donations

Logged In: YES
user_id=87038
Sorry, I can't debug this w/o dialing in. It's a setup issue
- i.e.
your responibility - there are enough working examples in the
system.
Date: 2005-05-06 23:18
Sender: alinv
Logged In: YES
user_id=23294
I attached a screenshot with what Workflow editor thinks about
my workflow. I also should mention that at the time I opened
this request, there were only 2 nodes and the split/join was
XOR (as opposed to AND now), with the same result
Thank you
Date: 2005-05-06 02:51

Sender: jjankeSourceForge.net Subscriber and DonorAccepting Donations

Logged In: YES
user_id=87038
Validate is very restrictive at the moment.
You probably have the transitions wrong - to check that,
that's what the Workflow editor is for.
Date: 2005-05-05 00:56
Sender: alinv

Logged In: YES
user_id=23294
Please tell me what's wrong with my workflow, because both
Workflow Editor and Validate Workflow process say it's ok.
Thank you
Date: 2005-05-05 00:10
Sender: jjankeSourceForge.net Subscriber and DonorAccepting Donations
Logged In: YES
user_id=87038
Sorry, seems that you have set up your worflow not
correctly. Check with the Workflow Editor
Date: 2005-05-05 00:10
Sender: jjankeSourceForge.net Subscriber and DonorAccepting Donations
Logged In: YES
user_id=87038
Sorry, seems that you have set up your worflow not
correctly. Check with the Workflow Editor
Date: 2005-05-05 00:05
Sender: alinv
Logged In: YES
user_id=23294
Please have a look at the attached logs. They are about the
same 3-nodes workflow. In one case it is run without calling
loadNodes() in MWorkflow constructor (the default) and only the
first node is evaluated (trace_wf_wo_loadnodes.txt). In the
second case, all the nodes are evaluated, after loadNodes() is
explicitly called (trace_wf_w_loadnodes.txt).
Date: 2005-05-04 23:20
Sender: jjankeSourceForge.net Subscriber and DonorAccepting Donations
Logged In: YES
user_id=87038
Cannot replicate. Suggest that you attend the technical
training.

7 From a users and an authorizer's perspective

7.1 The initial requisition

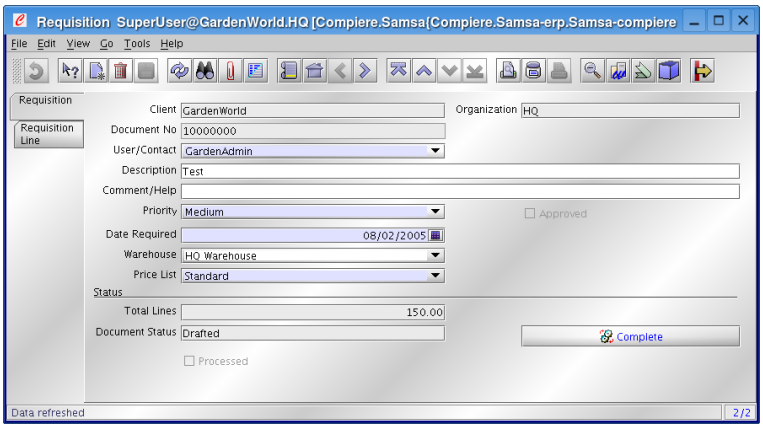

Figure 7.1:

7.2 Suspended

7.3 Workflow history

The admin had to answer Yes/No

7.4 The approved requisition

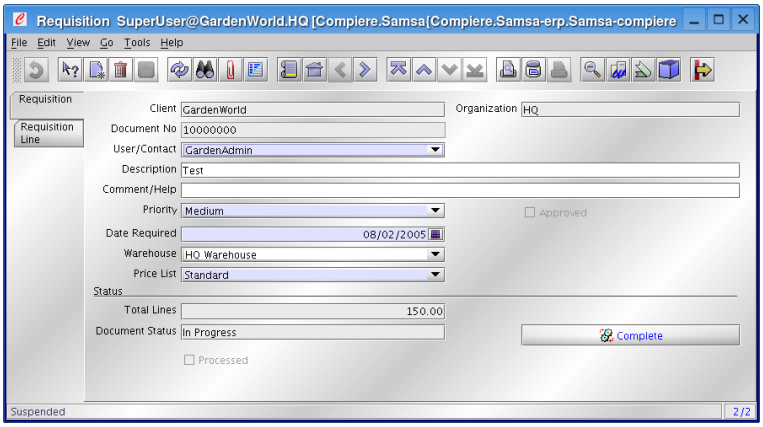

Figure 7.2:

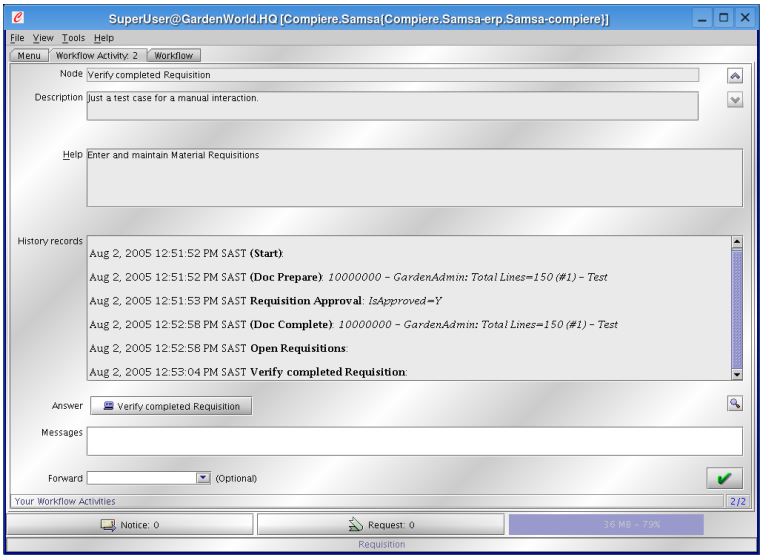

Figure 7.3:

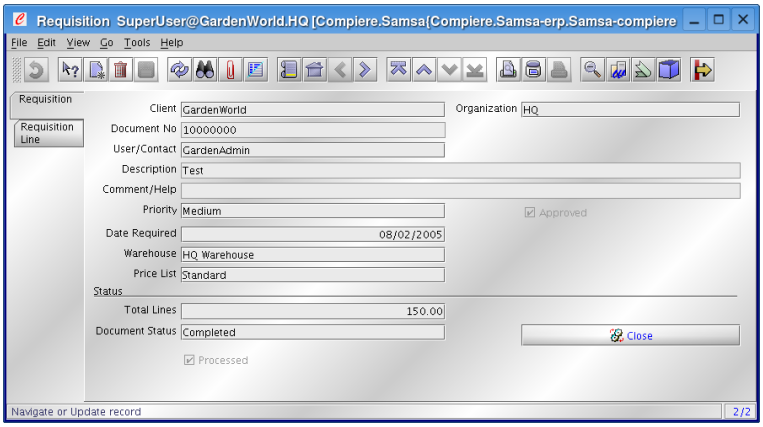
Requisition SuperUser@GardenWorld.HQ [Compiere.Samsa{Compiere.Samsa-erp.Samsa-compiere
File Edit View Go Tools Help
Requisition
Requisition Line
Client GardenWorld
Organization HQ
Document No 10000000
User/Contact GardenAdmin
Description Test
Comment/Help
Priority Medium
Approved
Date Required 08/02/2005
Warehouse HQ Warehouse
Price List Standard
Status
Total Lines 150.00
Document Status Completed
Close
Processed
Navigate or Update record
2/2

Figure 7.4:

8 Whats happening in the background

- *Log output with log level set to FINE has given the best information as to what is happening in the heart of the workflow engine. Clear the console in eclipse (set the debugging console window to show unlimitted output) just before hitting 'Complete'*
- Reading the classes mentioned here also give insight as to how the process works.

8.1 Document Void Output

When a requisition is **voided** the following output gives insight as to what process is taking place and what the workflow is doing:::

15:23:46.333 APanel.dataStatusChanged: 3/6 [11]
15:23:46.333 APanel.setStatusLine: Record saved [11]
15:23:46.333 MTable.dataSavePO: fini [11]
15:23:46.333 MTab.setCurrentRow: Row=2 - fire=false [11]
15:23:46.333 GridController.dynamicDisplay: (MTab #0 Requisition (641))
[11]
15:23:46.333 GridController.dynamicDisplay: (MTab #0 Requisition (641))
- fini - complete [11]
15:23:46.333 ProcessCtl.process: WindowNo=3 - ProcessInfo[DocAction,Process_ID=273,Rec
[11]
15:23:46.333 MPInstance.saveNew: AD_PInstance - AD_PInstance_ID=1000148
[11]
15:23:46.333 ProcessParameter.initDialog: [11]
15:23:46.333 AWindow.setBusy: AWindow_322 - false [11]
15:23:46.333 ProcessCtl.run: AD_PInstance_ID=1000148, Record_ID=1000011
[39]

15:23:46.333 AGlassPane.setBusyTimer: AGlassPane.setBusyTimer - 5
[39]
15:23:46.333 VLookup.focusGained: AD_User_ID - Start Count=2, Selected=GardenAdmin [11]
15:23:46.333 MLookup.fillComboBox: AD_User.AD_User_ID: fillComboBox - #2 - ms=0 [11]
15:23:46.333 VLookup.focusGained: AD_User_ID - Update Count=2, Selected=GardenAdmin [11]
15:23:46.333 VLookup.focusGained: AD_User_ID - Selected Count=2, Selected=GardenAdmin [11]
15:23:46.333 ProcessCtl.startWorkflow: 115 - ProcessInfo[Process Requisition,Process_ID=273,AD_PInstance_ID=1000148,Record_ID=1000011,Error=false,Summary=,
[39]
15:23:46.333 M_Table.getPOclass: Found: org.compiere.model.MRequisition
[39]
15:23:46.333 MWFProcess.saveNew: AD_WF_Process - AD_WF_Process_ID=1000026
[39]
15:23:46.333 MWFProcess.startWork: AD_WF_Node_ID=181 [39]
15:23:46.333 MWFProcess.setWFState: OR [39]
15:23:46.333 MWFProcess.saveUpdate: AD_WF_Process.AD_WF_Process_ID=1000026
[39]
15:23:46.333 MWFActivity.saveNew: AD_WF_Activity - AD_WF_Activity_ID=1000072
[39]
15:23:46.333 MWFEventAudit.saveNew: AD_WF_EventAudit - AD_WF_EventAudit_ID=100
[39]
15:23:46.333 MWFActivity.run: Node=MWFNode[181-(Start),Action=Sleep:WaitTime=0]
[40]
15:23:46.333 MWFActivity.setWFState: ON->OR, Msg=null [40]
15:23:46.333 MWFActivity.saveUpdate: AD_WF_Activity.AD_WF_Activity_ID=1000072
[40]
15:23:46.333 MWFEventAudit.saveUpdate: AD_WF_EventAudit.AD_WF_EventAudit_ID=10
[40]
15:23:46.333 MWFProcess.checkActivities: (115) - OR [40]
15:23:46.333 MWFActivity.performWork: MWFNode[181-(Start),Action=Sleep:WaitTime=0]
[WF_1123076972606] [40]
15:23:46.333 MWFActivity.performWork: Sleep:WaitTime=0 [40]
15:23:46.333 MWFActivity.setWFState: OR->CC, Msg=null [40]
15:23:46.333 MWFActivity.saveUpdate: AD_WF_Activity.AD_WF_Activity_ID=1000072
[40]
15:23:46.333 MWFEventAudit.saveUpdate: AD_WF_EventAudit.AD_WF_EventAudit_ID=10
[40]

15:23:46.333 MWFProcess.checkActivities: (115) - OR [40]
15:23:46.333 MWFProcess.startNext: Last=MWFActivity[1000072,Node=181,State=CC,AD
08-03 15:49:32.0] [40]
15:23:46.333 MWFActivity.saveUpdate: AD_WF_Activity.AD_WF_Activity_ID=1000072
[40]
15:23:46.333 M_Table.getPOclass: Found: org.compiere.model.MRequisition
[40]
15:23:46.333 MWFNodeNext.isValidFor: isValidFor =NO= StdUserWF
- Status=DR - Action=VO [40]
15:23:46.333 MWFNodeNext.isValidFor: isValidFor #0 MWFNodeNext[100:Node=181-
>Next=182,#0] [40]
15:23:46.333 MWFActivity.saveNew: AD_WF_Activity - AD_WF_Activity_ID=1000073
[40]
15:23:46.333 MWFEventAudit.saveNew: AD_WF_EventAudit - AD_WF_EventAudit_ID=1000
[40]
15:23:46.333 MWFActivity.run: Node=MWFNode[182-(DocAuto),Action=DocumentAction=
] [41]
15:23:46.333 MWFActivity.setWFState: ON->OR, Msg=null [41]
15:23:46.333 MWFActivity.saveUpdate: AD_WF_Activity.AD_WF_Activity_ID=1000073
[41]
15:23:46.333 MWFEventAudit.saveUpdate: AD_WF_EventAudit.AD_WF_EventAudit_ID=10
[41]
15:23:46.333 MWFProcess.checkActivities: (115) - OR [41]
15:23:46.333 MWFActivity.performWork: MWFNode[182-(DocAuto),Action=DocumentActio
] [WF_1123076972761] [41]
15:23:46.333 MWFActivity.performWork: DocumentAction=– [41]
15:23:46.333 M_Table.getPOclass: Found: org.compiere.model.MRequisition
[41]
15:23:46.333 Trx.start: **** WF_1123076972761 [41]
15:23:46.333 MRequisition.processIt: **** Action=VO (Prc=–/Doc=VO)
MRequisition[1000011-gfhfgh,Status=DR,Action=VO] [41]
15:23:46.333 MRequisition.voidIt: voidIt - MRequisition[1000011-gfhfgh,Status=DR,Action=V
[41]
15:23:46.333 MRequisition.closeIt: closeIt - MRequisition[1000011-gfhfgh,Status=DR,Action=
[41]
15:23:46.333 MRequisitionLine.saveUpdate: [WF_1123076972761] - M_RequisitionLine.M_Req
[41]
15:23:46.333 MRequisitionLine.updateHeader: updateHeader [41]
15:23:46.333 MRequisition.saveUpdate: [WF_1123076972761] - M_Requisition.M_Requisition_I
[41]

15:23:46.333 MRequisition.processIt: **** Action=VO - Success=true
[41]
15:23:46.333 MRequisition.saveUpdate: [WF_1123076972761] - M_Requisition.M_Requisition_I
[41]
15:23:46.333 MWFActivity.setWFState: OR->CC, Msg=gfhfgh - Garde-
nAdmin: Total Lines=0.0 (#1) - hgfh [41]
15:23:46.333 MWFActivity.saveUpdate: AD_WF_Activity.AD_WF_Activity_ID=1000073
[41]
15:23:46.333 MWFEventAudit.saveUpdate: AD_WF_EventAudit.AD_WF_EventAudit_ID=10
[41]
15:23:46.333 MWFProcess.checkActivities: (115) - OR [41]
15:23:46.333 MWFProcess.startNext: Last=MWFActivity[1000073,Node=182,State=CC,AD_
08-03 15:49:32.0] [41]
15:23:46.333 MWFActivity.saveUpdate: AD_WF_Activity.AD_WF_Activity_ID=1000073
[41]
15:23:46.333 MWFProcess.setWFState: CC [41]
15:23:46.333 MWFProcess.saveUpdate: AD_WF_Process.AD_WF_Process_ID=1000026
[41]
15:23:46.333 Trx.commit: **** WF_1123076972761 [41]
15:23:46.333 Trx.close: WF_1123076972761 [41]
15:23:46.333 MWorkflow.startWait: Completed [39]
15:23:46.333 AWindow.setBusy: AWindow_322 - false [11]
15:23:46.333 MTab.dataRefresh: #0 - row=2 [11]
15:23:46.333 MTable.dataRefresh: Row=2 [11]
15:23:46.333 MTable.dataIgnore: Nothing to ignore [11]
15:23:46.333 MTab.dataStatusChanged: #0 - DataStatusEvent - Refreshed
: 0/6 [11]
15:23:46.333 MTab.fireDataStatusChanged: DataStatusEvent - Refreshed
: 3/6 [11]
15:23:46.333 GridController.dataStatusChanged: (MTab #0 Requisition
(641)) Col=0: DataStatusEvent - Refreshed : 3/6 [11]
15:23:46.333 GridController.dynamicDisplay: (MTab #0 Requisition (641))
[11]
15:23:46.333 GridController.dynamicDisplay: (MTab #0 Requisition (641))
- fini - complete [11]
15:23:46.333 APanel.dataStatusChanged: 3/6 [11]
15:23:46.333 APanel.setStatusLine: Data refreshed [11]
15:23:46.333 MTab.setCurrentRow: Row=2 - fire=true [11]
15:23:46.333 VLookup.setValue: DocStatus=VO [11]
15:23:46.333 MTab.fireDataStatusChanged: DataStatusEvent - Refreshed
: 3/6 [11]

15:23:46.333 GridController.dataStatusChanged: (MTab #0 Requisition (641)) Col=0: DataStatusEvent - Refreshed : 3/6 [11]
15:23:46.333 GridController.dynamicDisplay: (MTab #0 Requisition (641)) [11]
15:23:46.333 GridController.dynamicDisplay: (MTab #0 Requisition (641)) - fini - complete [11]
15:23:46.333 APanel.dataStatusChanged: 3/6 [11]
15:23:46.333 APanel.setStatusLine: Data refreshed [11]
15:23:46.333 GridController.dynamicDisplay: (MTab #0 Requisition (641)) [11]
15:23:46.333 GridController.dynamicDisplay: (MTab #0 Requisition (641)) - fini - complete [11]
15:23:46.333 APanel.setStatusLine: Completed [11]

8.2 Complete - what happens?

This is a semi-correct analysis of the flow in the source code as to what happens when 'Complete' is pressed on the Requisition. It also indicates which classes should be read in order to understand/diagnose the workflow.

8.2.1 Trace

> VDocAction.actionPerformed vda.setVisible(true); if (!vda.getStartProcess()) return; startWOasking = true; vda.dispose(); APanel - Start Process

Starts M_Requisition_Process (id273)

273 0 0 Y 2004-04-10 17:40:21.0 0 2000-01-02 00:00:00.0 0 M_Requisition_Process Process Requisition (null) (null) 1 D (null) N N (null) (null) 0 0 (null) Process_Requisition 115 N

ProcessCtl.java
This process only has AD_Workflow_ID defined
if (AD_Workflow_ID > 0)
{
startWorkflow (AD_Workflow_ID);
unlock();
return;
}
startWorkFlow
Code:

MWorkflow wf = MWorkflow.get (Env.getCtx(), AD_Workflow_ID);
MWFProcess wfProcess = wf.startWait(m_pi); // may return null
started = wfProcess != null;
}
MWFProcess.java
MWorflow: wf.startWait(m_pi)
MWFProcess process = start(pi);
start
retValue = new MWFProcess (this, pi);
retValue.save();
retValue.startWork();
WFSTATE_NotStarted
XOR: Only the first valid
// only the first valid if XOR
if (MWFNode.SPLITELEMENT_XOR.equals(split))
return true;
AND: All the items
startWork
MWFActivity activity = new MWFActivity (this, AD_WF_Node_ID);
new Thread(activity).start();
MWFActivity.run
performWork
true:STATE_Completed
false: STATE_Suspended
sleep
wait time=0: true
wait time!=0: false
If workflow is defined no other attribute (e.g. ClassName) is considered

8.3 Classes/Methods of interest:WorkflowProcessor

Runs every 2 minutes

8.3.1 wakeUp()

Wakes up processes that are 'Suspended' and sleeping. It gets this info from the AD_WF_Activity table.

The query is as follows:
SELECT *
FROM AD_WF_Activity a

WHERE Processed='N' AND WFState='OS' – suspended
AND EndWaitTime > SysDate
AND AD_Client_ID=?
AND EXISTS (SELECT * FROM AD_Workflow wf
INNER JOIN AD_WF_Node wfn ON (wf.AD_Workflow_ID=wfn.AD_Workflow_ID)
WHERE a.AD_WF_Node_ID=wfn.AD_WF_Node_ID
AND wfn.Action='Z' – sleeping
AND wf.AD_WorkflowProcessor_ID IS NULL OR wf.AD_WorkflowProcessor_ID=?)
For each of these records we do:
MWFActivity activity = new MWFActivity (getCtx(), rs, null);
activity.setWFState (StateEngine.STATE_Completed);

Checks AD_WF_Activity

8.3.2 Run on server or on client?

Seems to run on the client

8.3.3 MWFNodeNext

This is an important class which decides which node to use next based on the Transition information.

isValidFor

if isStdUserWorkflow
if ...
!DocAction.ACTION_Complete.equals(docAction)
—— DocAction.STATUS_Completed.equals(docStatus)
—— DocAction.STATUS_WaitingConfirmation.equals(docStatus)
—— DocAction.STATUS_WaitingPayment.equals(docStatus)
—— DocAction.STATUS_Voided.equals(docStatus)
—— DocAction.STATUS_Closed.equals(docStatus)
—— DocAction.STATUS_Reversed.equals(docStatus))
Not Action=Complete
If VOID then returns false!!
As seen below, ACTION VOID is not valid for StdUserWorkFlow
if NOT isStdUserWorkflow
If has no conditions then TRUE

9 Background reading (summary of existing materials)

See the workflow.ppt document from Jorg for the complete materials

9.1 Workflow Overview

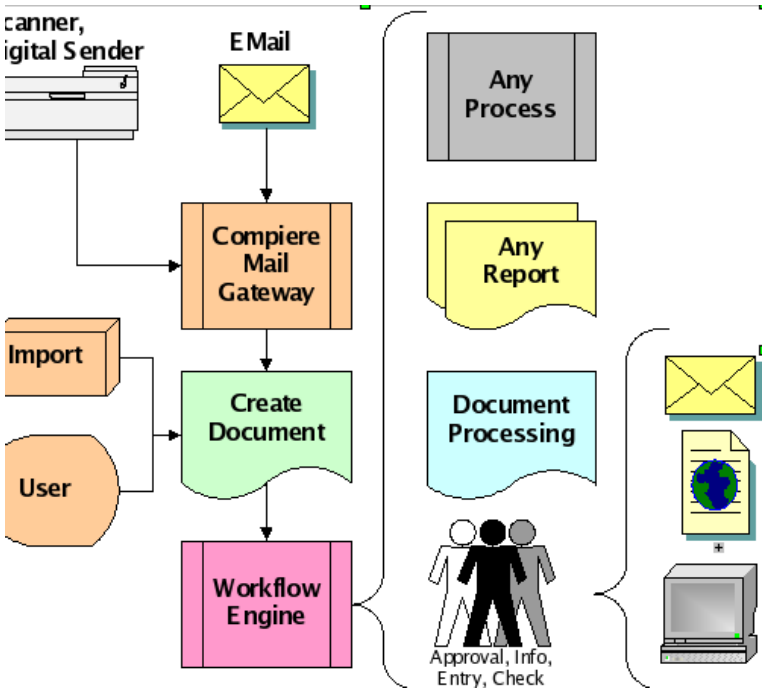

Figure 9.1:

9.2 Sequences

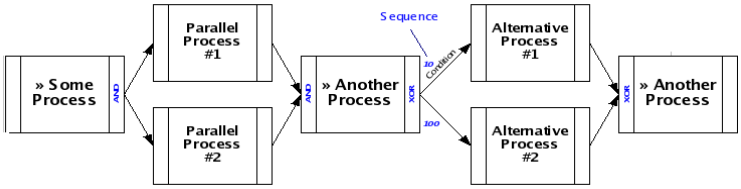

Figure 9.2:

9.3 Doc Status

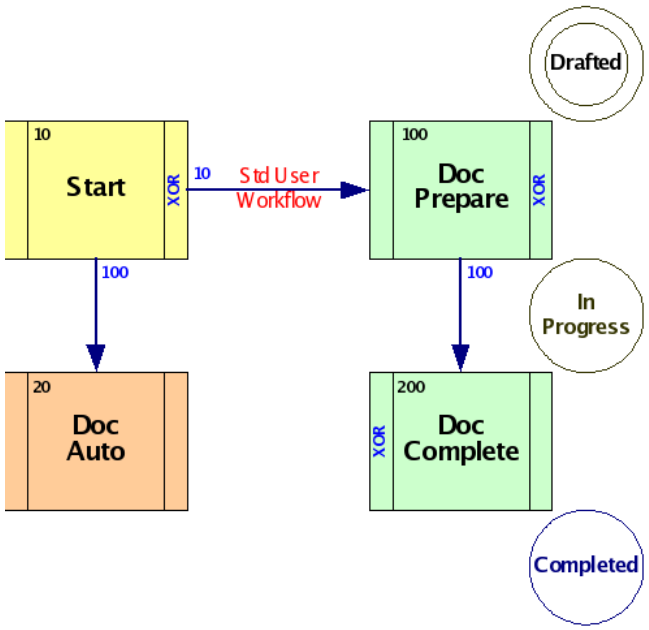

Figure 9.3:

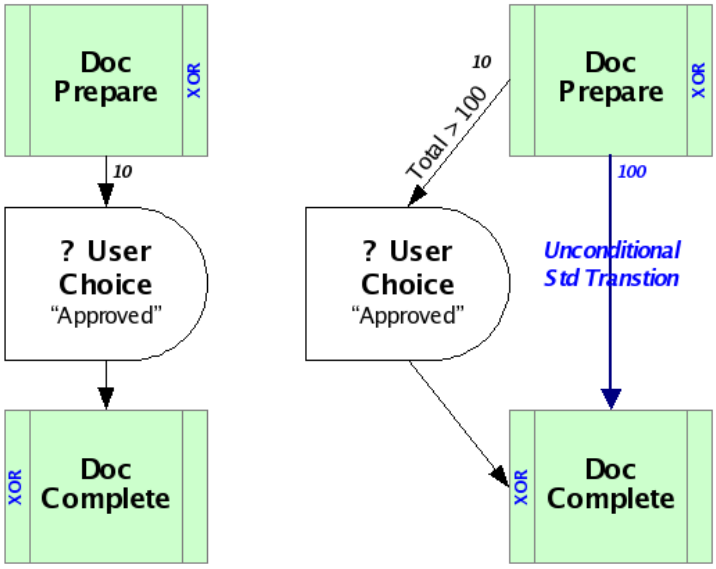

Figure 9.4:

9.4 Approval Standard

9.5 Doc Status and Action

9.6 Approval non standard

9.7 Text from the Compiere Presentation

Any kind of Approval

Some decisions between document status: Draft and Complete

Follow up ...

Order was processed ... inform people ... check completeness after one week, ...

etc.

Any formal/informal Process/Procedure

Based on Standards

Workflow Management Coalition

OMG Workflow Facility

Actual API

Based on Java API Notation

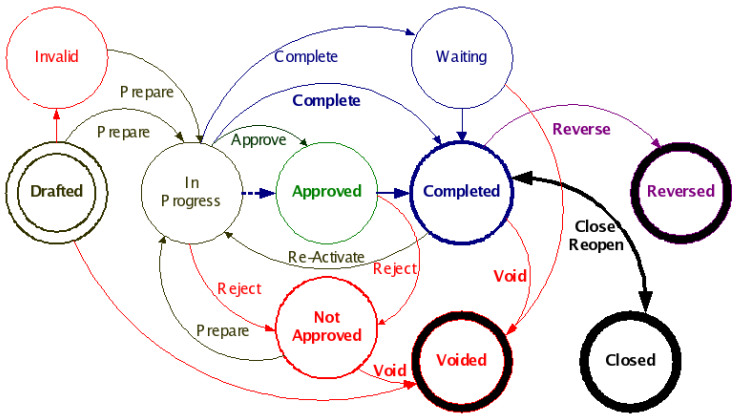

Figure 9.5:

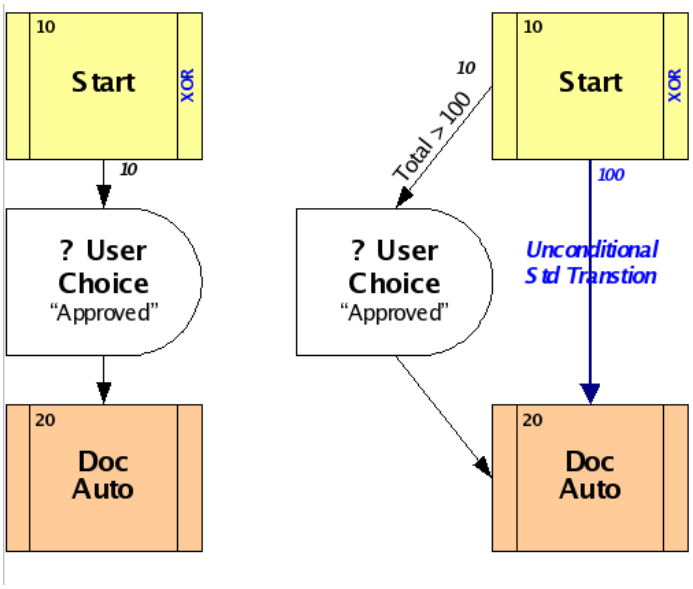

Figure 9.6:

For full standard compliance, the equivalent CORBA APIs needed
Workflow
Definition of Flow = Node & Transitions
Node
The piece of Work
Transition (Next)
Transition between Nodes
Process
Active Workflow
Activity
Active Node
Process can have multiple (parallel) Activities
History
Open
Not Running
Not Started
Suspended
Waiting User Input
Running
Closed
Completed
Aborted
Environment /
Setup Issue
Terminated
Execution Issue
Document Process Workflow
Started via Process Button
Process created immediately
Has a defined Start Context (Document)
E.g. Order 1234
Has a Responsible
Document Value Workflow
Started after (every) Save of Document
When Condition in Document is fulfilled
General Workflow (Step/Guidance)
Started & performed via Menu
"Online" Process until other User is responsible or System Activity*
Types
Human
A defined User

Invoker = the user who started it
If no User defined
Organization
The Supervisor of the Document Organization
Role
All Users who have that Role
System Resource*
If no Users are found, it is the Invoker
Defined on Workflow Level
Optionally overwritten on Node Level
User Action or User Choice
Available to ALL Responsible online
The First one "wins"
Email Reminder sent after expected Duration
Your "To Do" List
Directly or as Workflow Responsible
Via
Menu Tab
Window: Workflow Activity
Self Service
Approve, View, etc.
Zoom to Document
Menu Workflow Tab updated
Starting any Menu Item
Clicking on Memory Gauge
Process = Active Workflow Instances
View All Processes, Activities, History
Manage Process
Abort
New User or Workflow Responsible
Activity = Active Workflow Nodes
Ordered by Priority
View History
Manage Activity
Abort
New User or Workflow Responsible
Actual Completion Times
Workflow – Node
Average Completion Times
Workflow – Node
Prepare Simulation*

Set Simulation Data in Workflow / Node
Defining workflows
The actual Work Step
Actions (Types)
Automatic Process
Any Process, Report, Task, Workflow, Document Action
Later* also Web Services, etc.
User Action
Any Window, Form
User needs to confirm Completion
Set Variable
Any Column to Constant or Variable
User Choice
Any Choice, List (other are User Action)
Wait (Sleep)
Also for Start/End/etc.
If empty = Unconditional
"Standard User Workflow"
Predefined "Is this a Approval Situation"
Not if Document Status
Complete, Waiting, Closed, Voided, Closed, Reversed
Not if Document Action
None, Post, Invalidate, Unlock
Conditional
AND/OR
Variable/Constant OP Variable/Constant
Operation: =, !=, >, <, ≥, ≤, ~, >-<
Variable
Context @Variable@
Data Column of Persistent Object
Workflow transitions
If more than one Transition
Evaluated in Sequence
If more than one valid Out Transition
Split Element = XOR (default)
First Transition only
Split Element = AND
Parallel Execution of Activities
If more than one In Transition
Join Element = XOR (default)
First Arrival continues Workflow

Join Element = AND
Wait for all Activities to be complete
Workflow defined Duration Unit
e.g. Day, hour, minute, ..
Duration Limit
Optional Maximum Expected Time for User Actions
After that Email Reminder to Responsible ofActivity & Process
Inactivity Days (Workflow Processor)
Email Reminder to Responsible of Activity & Process
Priority = Optional Relative Number
Node Priority is starting point
Activity Priority
Sort Criteria of Activities
Optional dynamic Change
E.g. every hour waiting +10
Notify Workflow Processor Supervisor
If Priority exceeds Limit
Tasks
Continue Workflow after Wait Period
Send Alerts & Reminders
Increase dynamic Priority
Default System Level Processor
Workflows can have additional specific Processor
Processed then by both
Example: Order Processing every 5 minutes
User Action
In Activity
Open/Zoom to Document
Optionally enter Description/Comment
Mark as complete
User Choice
In Activity / Self-Service
Make Decision
Optionally enter Description/Comment
Email (Reminder)*
Reply and mark choice
Optionally enter Description/Comment
Workflow (Commit) Block
For Automatic Processes only
If Process fails, rollback all previous Processes
If Invoice creation fails, roll back Shipment

Workflow Context
Workflow maintains a Persistent Object
E.g. Order 1234
Nodes can change/add to the Context
Document Workflows
Predefined Workflows for Document processing (Owner=System)
Nodes (Processes)
Start – Doc Auto
To ease Workflow Setup
Doc Prepare
Data Consistency check, Reserve Inventory, ...
Doc Complete
(Assumes Document is approved)
Generation, etc.
Processed = true
Document Status & Action
Additional System Actions
None, Unlock, Invalidate, Post
Standard User Workflow (Transition)
"Guard" for manual (approval) Processes
Prevents that a Workflow Branch is used
Only for Complete Actions
Not when Status is "Closed" for users
Waiting – Completed – Voided – Reversed – Closed
Standard "DocAuto" Node performs any automatic Processes
Don't use when you write e.g. manual Processes
E.g.: for Reverse, Close, Open, Re-Activate, Void
Customizing Workflows:
Option 1 (for all Clients)
Add your Nodes/Transitions
Entity Type not Dictionary/Compiere
Deactivate Standard Transitions as required
Option 2 (for specific Clients/Orgs)*
Create (copy) your own Workflow
ADD something to the Workflow Key (!)
E.g. Process_Order -> Process_Order_Mine
Set as Default
Approvals
Add your Approval Nodes between DocCheck and DocProcess
Option 1
Create "User Choice" Node for Column Approved

Define Workflow Responsible for User/Role for that (specific) Approval Transition with first Sequence e.g.
Column GrandTotal > 1000
(add Currency if applicable)
Option 2 (Implicit)
Create "User Choice" Node for Column Approved
Select "Invoker" as Responsible
Create Transition(s)
Approval of void, etc
Approval Steps or Actions for "Non-Standard Workflow"
Voiding, Re-Activate, Processing, Reverse, etc.
Needs to be before "Doc Auto"
Approval fails:
First time, a Document is NOT approved
Document Status = Not Approved
Workflow State = Completed
Optionally: Make Changes
Document Status = In Progress / Drafted
Re-Submit
Alternative: Void
Approval hierarchy
Sequence
Supervisor of Invoker
Supervisor of Organization
Supervisor of Parent Organization
Etc.
First Supervisor found
If not sufficient Limit repeated for Supervisor of Supervisor
Implicit Approval/Routing
Define Approval Amounts on Role
The Invoker can optionally approve the document
Used when on Node no explicit Workflow Responsible defined
Invoker, Organization
If Sales Rep on a Document exists, the Sales Rep is used instead of the "real" Invoker
The first user assigned to a Role with a sufficient Approval Amount finally approves
Invoker is not asked to approve (is assumed)
Define Approval Amounts on Role
The Invoker can optionally approve the document
Used when on Node no explicit Workflow Responsible defined

Invoker, Organization

If Sales Rep on a Document exists, the Sales Rep is used instead of the "real" Invoker

The first user assigned to a Role with a sufficient Approval Amount finally approves

Invoker is not asked to approve (is assumed)

Dynamic changes

Dynamic Routing

Any User can forward/delegate a Workflow Activity to another user

This Suspends the Activity

Allows Ad-hoc modification of Workflow

Re-Assign Processes or Activity

To other User

To other Workflow Responsible

Workflow Processor

Wakeup after Sleep

Dynamic Priority Change

Alerts (EMails)

Over Priority Threshold

Over End Wait Time

Based also on Duration Limit of Node

Over Days of Inactivity

Remind Days

For sending next EMail

Notes

Workflow Definition Changes

Are currently not tracked

Only one Version

Change with active Processes

Will use new Nodes/Transition

If Node has no Transitions anymore

Assumes it is complete

External users

External Users

Integration of Business Partners

E.g.: Provide Service, Ship Goods, ...

How

As Workflow Responsible

Dynamic Routing

Use Email reply or Web only

Planned

Workflow Process Simulation
Export/Import of Workflow Definition
Link to other Workflow Engines

10 Adding an approval option to the standard purchase order

10.1 Cannot Approve - No Approver

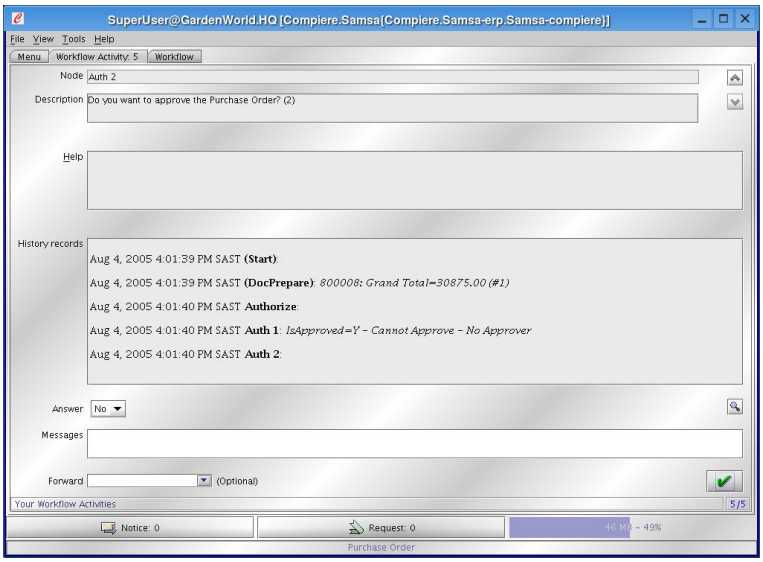

Figure 10.1:

10.1.1 User/Contact must be incorrect?

10.2

11 Example Workflow Responsible Setup

This example allows two different users to act as authorizers in the examples given in this document

11.1 Setting up Workflow Responsible

These are people or roles responsible for authorising workflows. This is the easiest way of setting up who is responsible. Other methods are Organization (still unclear) and Invoker which typically uses one of the fields from the document to decide who is responsible.

11.1.1 Update queries

The users who are 'Workflow Responsible' must belong to client 0.

These queries can assist or they can be setup as such in Compiere.

```
select * from ad_user a
where a.name like 'Oper%'
select * from ad_user a
where a.name like 'CFO1%'
update ad_user a set a.ad_client_id=0
where a.ad_user_id=1011303
```

11.1.2 Workflow responsible - CFO

11.1.3 Workflow responsible - Operations Executive Manager

11.2 Node: Approval Operations Executive

Note the workflow responsible - 'Operations Executive Manager'

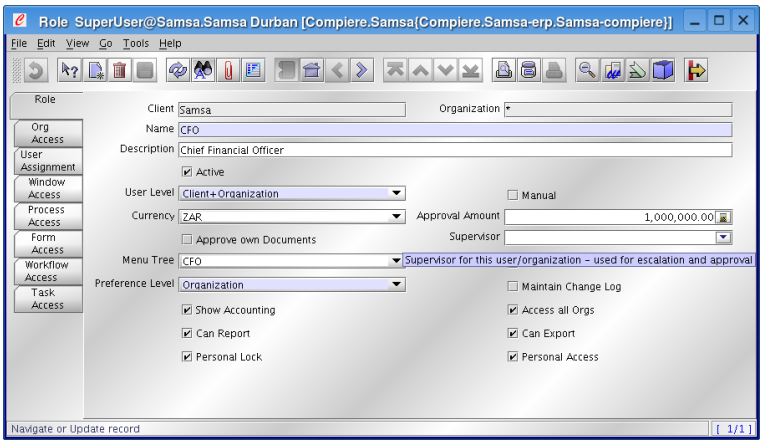

Figure 11.1: Workflow responsible - CFO

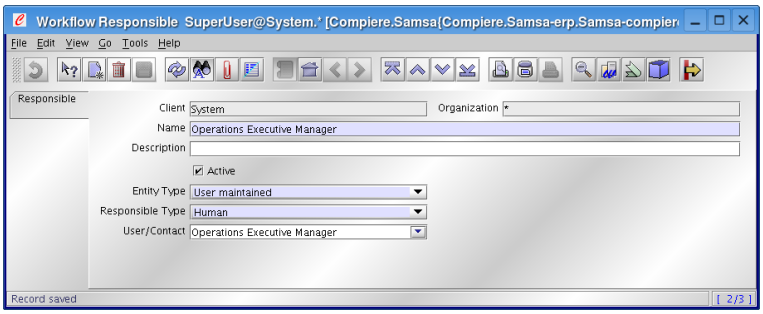

Figure 11.2: Workflow responsible - Operations Executive Manager

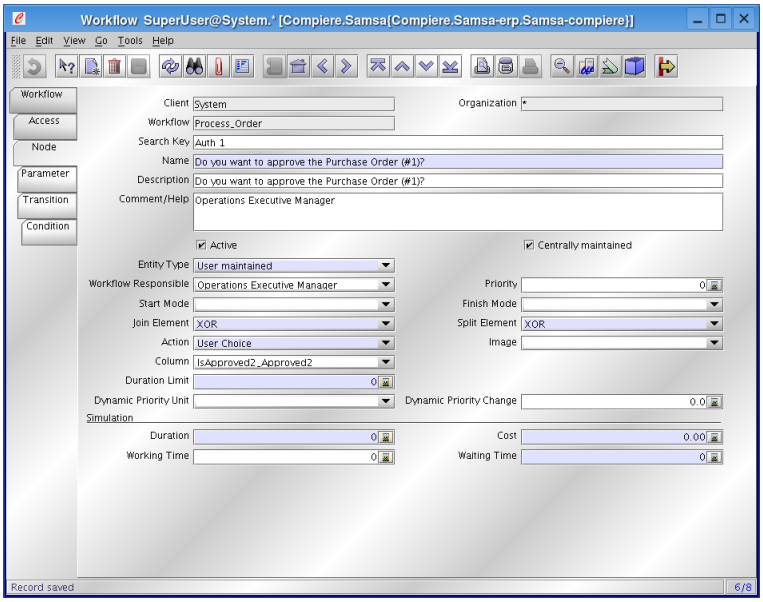

Figure 11.3: Note the workflow responsible - 'Operations Executive Manager'

11.3 Node: CFO Approval Node

Note the workflow responsible - 'CFO'

11.4 Demo

- Login as Port Officer Durban
- Create purchase order, grand total > R2000
- Complete document. Goes to suspended state.

11.4.1 Login

NB: The user is important, not the role that is selected.

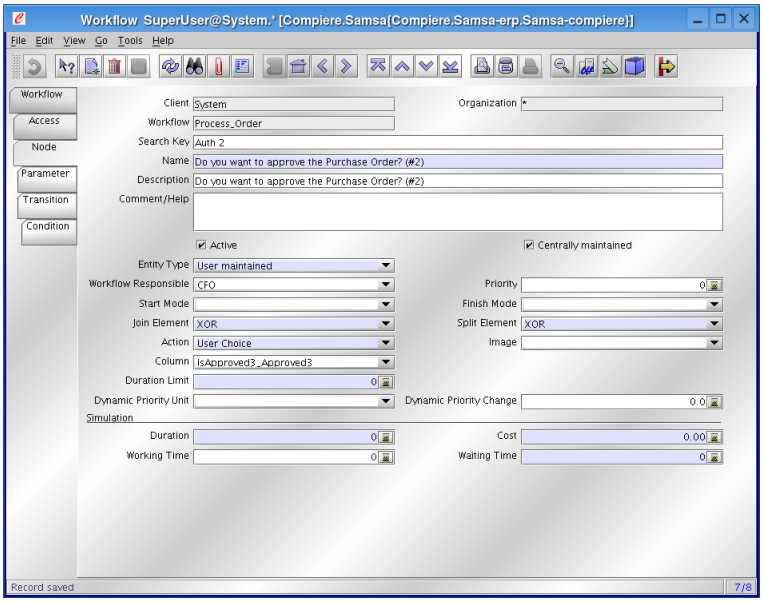

Figure 11.4: Approval Organization Node

11.4.2 Purchase Order

11.4.3 PO Line

11.4.4 Complete - Suspended

11.4.5 TODO: Add a section on the approver's actions

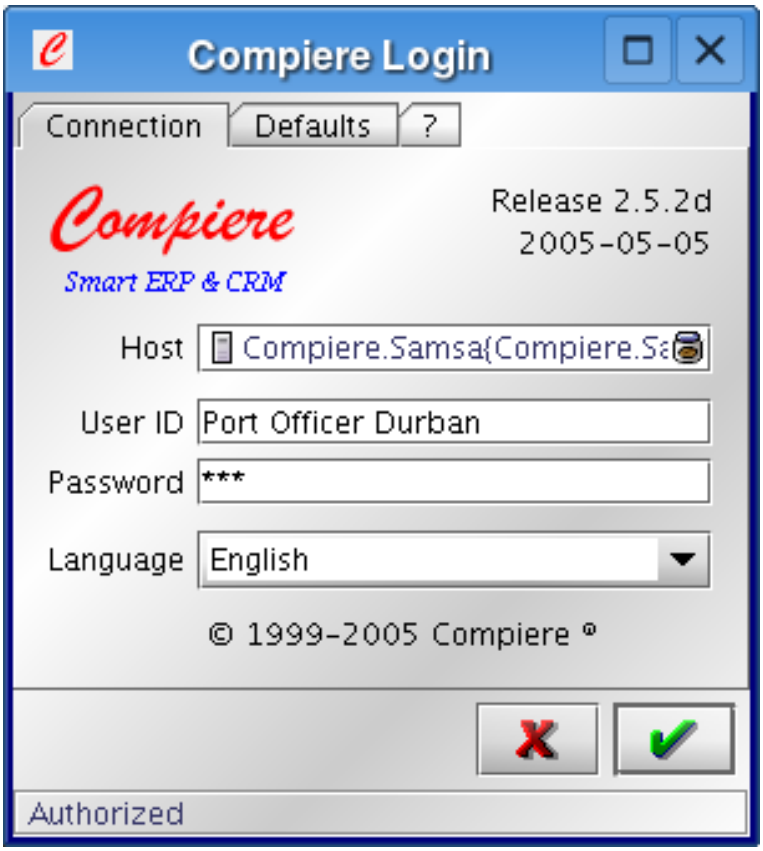

Figure 11.5: Login

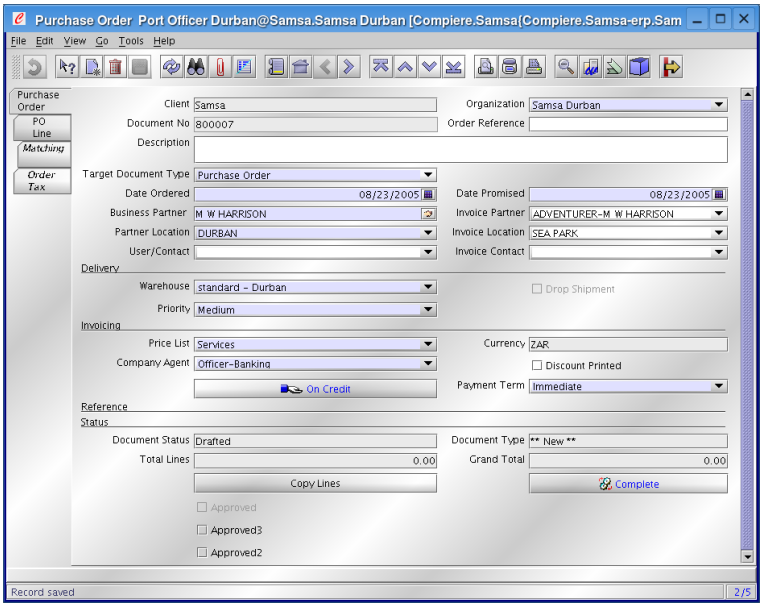

Figure 11.6: Purchase Order

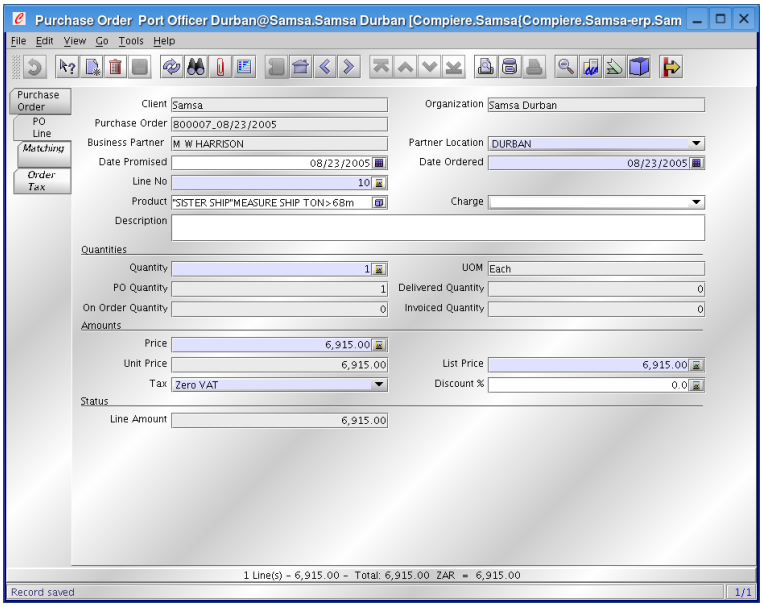

Figure 11.7: PO Line

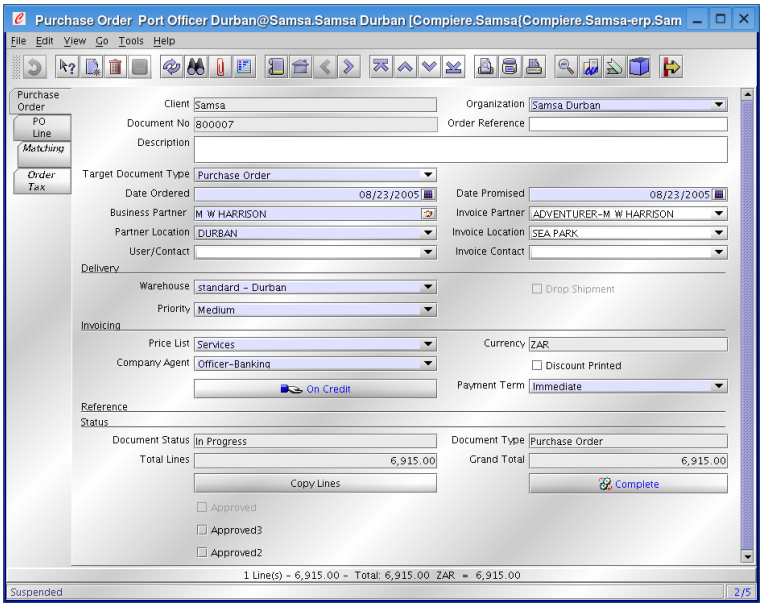

Figure 11.8: Complete - Suspended

12 Supervisor Setup

12.1 Organization Responsible

The user must be linked to a business partner otherwise won't show up in the list of users. (This is setup under Organization/Organisation Info/Supervisor). The business partner must be labled as an 'Employee' ***and*** 'Sales Rep'. The sales rep enables the item in the 'Company Agent' drop down lists

12.1.1 Organization Info showing the Supervisor field

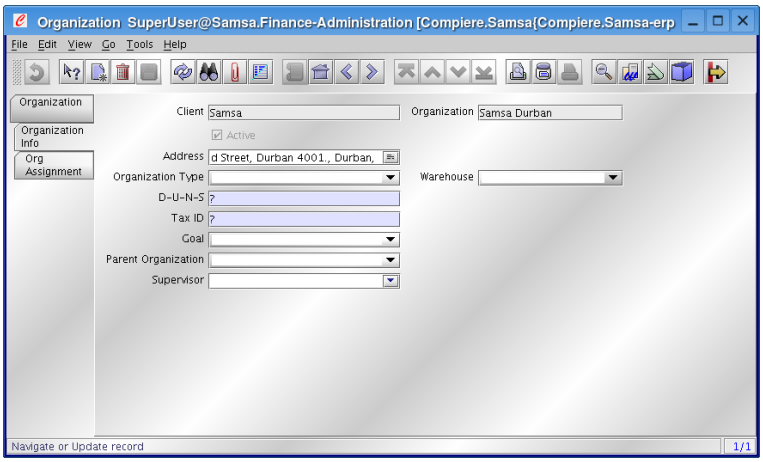

Figure 12.1: Organization Info showing the SuperVisor field

12.1.2 sql

This is for the info lookup for the org responsible:
(SELECT * FROM C_BPartner bp WHERE AD_User.C_BPartner_ID=bp.C_BPartner_ID AND (bp.IsEmployee='Y' OR bp.IsSalesRep='Y')))

12.2 Workflow responsible

Is based on users not on roles.

To test you have to logon as the user. Logging in as SuperUser and then selecting the role is not sufficient.