

Workflow Lifecycle 2

Also see the [document operation screen documentation](#), particularly the part about viewing the XML we give workflow.

Workflow Event Handling

Documents often perform business logic in response to certain workflow actions. For example, maintenance documents will modify database records in the database after all approvals are completed. Documents that post to the general ledger will mark pending entries as approved after all approvals are done.

Route Status Changes

`org.kuali.core.document.Document.handleRouteStatusChange()` is invoked by the KNS post processor when the document's route status (e.g. initiated, enroute, final, etc.) changes. The document's **new** status is accessible via the document header's workflow document (see [sample code](#)). The workflow document implements `org.kuali.core.workflow.service.KualiWorkflowDocument`, and its methods to retrieve a document's status begin with "statels".

The following code is from the base implementation of all GL posting documents.

From GeneralLedgerPostingDocumentBase.java

```
public class GeneralLedgerPostingDocumentBase extends LedgerPostingDocumentBase
implements GeneralLedgerPostingDocument {
    public void handleRouteStatusChange() {
        super.handleRouteStatusChange();
        if (getDocumentHeader().getWorkflowDocument().stateIsProcessed()) {
            changeGeneralLedgerPendingEntriesApprovedStatusCode();
        }
        else if (getDocumentHeader().getWorkflowDocument().stateIsCanceled() ||
getDocumentHeader().getWorkflowDocument().stateIsDisapproved()) {
            removeGeneralLedgerPendingEntries();
        }
    }
}
```

As we can see in the above example, if the document has just become processed, then the GL pending entries are marked as approved. If it has just been canceled or disapproved, then they are removed.



Since there is only one [MaintenanceDocument](#) implementation (`MaintenanceDocumentBase`) and custom behavior is possible only through [Maintainables](#), `Maintainable` provides a `handleRouteStatusChange` hook to which `MaintenanceDocumentBase` passes the `DocumentHeader`.

Route Level Changes

`org.kuali.core.document.Document.handleRouteLevelChange(DocumentRouteLevelChangeVO)` is invoked by the KNS post processor when the document's route level changes.

The `DocumentRouteLevelChangeVO` parameter contains information about the new and old route level, route node name, and route node ID.

From CorrectionDocument.java

```
public class CorrectionDocument extends TransactionalDocumentBase implements
AmountTotaling {
    private static final Integer WORKGROUP_APPROVAL_ROUTE_LEVEL = new Integer(1);

    public void handleRouteLevelChange(DocumentRouteLevelChangeVO change) {
        super.handleRouteLevelChange(change);
        if (WORKGROUP_APPROVAL_ROUTE_LEVEL.equals(change.getNewRouteLevel())) {
            // execute some logic
        }
    }
}
```

As we can see in the above code, this method will execute some logic when the document enters the workgroup approval route level. Different documents may have different integers representing the route level ID for each level.



Since there has been no need for MaintenanceDocuments to deal with route level changes thus far, there is no hook for this on Maintainable.

Synchronous vs. Asynchronous Routing

When running synchronously, all workflow actions happen synchronously. However, when running asynchronously (as KFS does), some actions still happen synchronously. Since we are running the workflow engine in embedded mode, all synchronous actions occur within our transactions without us having to do any additional work.

Here's the breakdown of which actions status and route node transitions are synchronous vs. asynchronous when running embedded workflow asynchronously (regardless of whether or not a standalone workflow server is in use):

- **Synchronous**
 - INITIATED -> SAVED
 - INITIATED -> ENROUTE
 - SAVED -> ENROUTE
 - ENROUTE -> DISAPPROVED
 - ENROUTE -> CANCELLED
- **Asynchronous**
 - ENROUTE -> PROCESSED
 - ENROUTE -> FINAL
 - All Route Node/Level transitions

Unable to render {include} The included page could not be found.