

# Task 85 RIPOUT Design Report

Author: John M. Slick, DXC [jslick2@dx.com](mailto:jslick2@dx.com) 615-826-8752

## Document Change History

DATE	DESCRIPTION	REVISION #
09Aug2019	Draft	1.0
17Sep2019	Draft	1.1
01Oct2019	Draft	1.2
16Oct2019	Draft	1.3

Overview.....	5
Supporting Documents.....	5
Ripout Workflow Overview.....	5
Routing.....	5
Status.....	6
Signing.....	6
Ripout Record Identifiers.....	6
XREF and Sequence.....	6
XREF.....	7
XREF_SEQ.....	7
Attribute Audit Trail.....	7
SSCI.....	8
Ripout List Number.....	8
Ripout List Number Increment.....	9
Ripout Creation.....	9
Ripout from Scratch .....	9
Ripout from Seed.....	10
Ripout Workflow Progression.....	11
Routing Steps: Open and Next.....	11
Input Validation.....	13
Red Mat Change Process.....	13
Red Mat Change Approval.....	13

Red Mat Validations.....	15
Red Mat Processing .....	15
Attachments.....	19
EBDT Attachments Code.....	19
EBDT File Attachment Code.....	20
Comments.....	23
JSON / Java Binding.....	24
Ripout Web Service Packaging.....	24
Web Service Security .....	24
HTTP Standard Methods.....	24
Enumerations .....	25
Resource Discovery.....	25
Ripout API.....	26
API Endpoints.....	27
Ripout Object Model.....	38
RipoutService.....	38
RipoutSession.....	38
RipoutSessionEJB.....	38
RipoutPO.....	38
JAXB.....	38
RipoutDao.....	39
7GRipoutReader.....	39
RipoutComments.....	39
Ripout Data Model.....	40

Assigning XREF (Primary Key).....	42
Assigning Ripout List Number.....	43
Workflow Routings.....	44
Nuclear Routings.....	46
Appendix 1 – Database Tables.....	47
Ripout Record.....	47
Ripout Record Attributes.....	50
Oracle Tables.....	64
Ripout_Comment.....	64
Ripout_Signature.....	64
Ripout_Routings.....	64
RedMat.....	64
Ripout_List.....	65
Item.....	65
Trade_Boundary.....	65
Controls Required.....	65
Retest Requirements.....	65
Database Connection.....	65
Project Documentation.....	66
Appendix 2 – Customer Comments.....	67
Footnotes.....	69

## Overview

This document describes high-level design of the future Ripout application to be hosted within Electronic Desktop (EBDT). DXC will develop the Ripout server-side application and data logic, and will publish an API for use by a client web application to be developed by Electric Boat.

The existing Tandem business process will cease Ripout functionality, and Ripout records will no longer be managed in Oracle 7G. Prior to deployment of Ripout on EBDT, all in-process 7G Tandem Ripout records will be put into a back log. Subsequent to deployment of Ripout, the back log records will be manually created by EB, using the new application. There is no plan for DXC to migrate in-process Ripout records into the new application. Ripout records will be created, managed and progressed through the ripout workflow via a client web application in conjunction with the ripout API exposed by EBDT.

## *Supporting Documents*

Details beyond the scope of this document may be found in the footnotes and also in the Ripout project repository: <sup>1</sup>

<\\us-ct-eb01\ebdepts\ENG.604.IPDE\IPDE Program\IPDE Business\Task Folders\Work Control Process-Flow Folder\Ripouts>

## Ripout Workflow Overview

For further details of work flow, see **Ripouts Detail Requirements** listed under heading **Project Documents**, below.

## *Routing*

The EBDT Ripout application will represent a workflow of routing steps, indicating which type of ripout user and AD role (trades, reviewers, quality, etc) is currently involved with the progression. For details about the association between AD groups and Ripout routings see document “Ripout Detail Requirements” in the Ripout project repository.

Each routing must be completed in a specific order and will require particular fields to be filled in. The ripout application server will validate that the user’s authorization role is appropriate to the current routing step, and that required input data was submitted and that each routing step is called in correct order.

## ***Status***

As the workflow progresses, the status will change. Several routing steps may progress under a given status. Routings and status will also vary between nuclear and non-nuclear ripout workflows. A detailed flow chart of workflow and status exists in the project document, Work Control Process Flow Ripouts MDD.pdf.

When the user is working on a particular routing step, s/he will provide input data required by the workflow.<sup>2</sup> The current routing and status will dictate which client input fields are read-only or editable, required or optional. It is the client application's responsibility to display, enable, disable and validate input fields on the client user interface. There will be no state management over multiple client-server requests.

## ***Signing***

When user is done editing a workflow step, then the client application will submit an HTTP request to a ripout REST API to apply the user's edits and apply the user's digital signature and user's optional comment to the completed step. Signing will be accomplished by EBDT ripout logic and shall include user's badge, name, department number, date, and time.

## **Ripout Record Identifiers**

The ripout Oracle record will require a primary key for internal use. The ripout client application may identify an instance of a ripout workflow by XREF, or perhaps by a combination of XREF, XREF\_SEQ, and RIPOUT\_NUMBER.

## ***XREF and Sequence***

XREF and Sequence are fundamental identifiers in the application. Customer has suggested a six-character XREF value as the unique identifier of each ripout record (pending DBA approval). Sequence number is intended to denote a contiguous workflow progression for an XREF. Therefore, a ripout process may be repeated or interrupted. If the workflow is repeated or is interrupted by a "Red Mat" proposed change, then the parent ripout record is duplicated and assigned the next sequence number, then the workflow proceeds again.

Ripout application will provide a means to display the history of attribute data for each ripout (XREF) under a given sequence number (XREF\_SEQ). The system shall also allow a user a view all sequences of an XREF. The system shall allow a user to modify attribute data but only on the latest XREF Sequence.

## ***XREF***

System-generated value unique to a ripout workflow. All ripout records created relative to the workflow will have the same XREF value. XREF may not be the ripout record primary key, but part of a secondary type of composite key.

## ***XREF\_SEQ***

System-generated value to uniquely identify a run of the workflow for a particular XREF. When a ripout workflow is repeated or is interrupted by a “Red Mat” proposed change, the ripout record is duplicated (retains XREF but XREF\_SEQ sequence is incremented by one) then the workflow proceeds again.

The ripout client application will provide a means to display the history of attribute data for each ripout XREF under a given XREF\_SEQ. The system shall also allow a user a view all sequences of an XREF. The system shall allow the modification of attribute data but only on the latest XREF Sequence.

## ***Attribute Audit Trail***

WRKCONT-1.7.1-005

The ripout service will assemble a history of attribute changes for each XREF. Given the ID of a ripout record and the XREF\_SEQ number, the service will respond with a JSON hierarchical representation of the ripout record, signature and comments. Roughly:

ripout\_record 1

status: NPUB

routing: Lead Trade – Approval

attributes [...]

signature

comment

ripout\_record 2

status: STEC

routing: Test Department - Test Controls

attributes [...]

signature  
comment  
...etc.

## ***SSCI***

SSCI (Ship System Code Index) identifies a procedural document which describes what is to be inspected, analogous to a Test Inspection Report. A *system* is an equipment module contained within a *frame* placed in a uniquely identifiable frame *bay*.

A ship will have one or more associated SSCI's. An SSCI relates to only one system.

An SSCI is not bound to a particular ship, but may be applied to different ships at different times.

An SSCI will specify one or more ripout procedures, which may be performed over several days. Workers might remove (rip out) frames 100 to 105; next week, frames 120 to 125.

SSCI will be defined by the Ripout Writer because Originator may not know the SSCI.

## ***Ripout List Number***

When Ripout Writer signs off, the system will assign a Ripout List Number with format: "RO" + "SSCI Number" + " " + "001".

Example: "RO82110 001". SSCI is Ship System Code Index Number (entered by ripout writer).

The first record created for the Ship + SSCI number shall be assigned as List = "001". Each sequential ripout record created for that Ship + SSCI number shall be assigned a list number incremented by one. Example: "RO82110 002".

After a ship is delivered (PSA [Post Shakedown Availability] / New Construction field [entered by Ripout Writer] = PSA) the system shall assign the ripout list number as 'RO' + "SSCI Number" + "B" + "001".

Example: "RO82110B001"

Per Jay McKernan 03Oct2019 discussion:

Ripout List number should specify "B" only if the ripout type is "enhanced"; otherwise no "B" or other character.

Note: If the Ship is on HOLD (aka LOCK), the Ripout List Number cannot be assigned and should be recorded on the ripout record as 271. Determine lock status by a query to CPR table. Ripout list number is unrelated to XREF\_SEQ.



## ***Ripout List Number Increment***

An SSCI may call for multiple ripout procedures. Each ripout workflow shall have a unique number, created as follows.

1. Originator creates a new record having unique 6-character xref, xref\_seq = 001, and limited other attributes.
2. Ripout Writer pulls originator's new record.
3. Ripout Writer defines SSCI and other limited attributes.
4. Writer saves record, system assigns ripout list number 001.

If the SSCI specifies multiple ripouts, then...

5. Ripout Writer pulls same originator record.
6. Ripout Writer defines same SSCI as in step 3, and adds limited attributes.
7. Writer saves record; system assigns ripout list number 002.

XREF now relates to two ripouts under same SSCI.

## **Ripout Creation**

A new ripout record may be created via an HTTP request to the ripout service. The record can be created from nothing ("scratch") or based on existing ripout record).

### ***Ripout from Scratch***

WRKCONT-1.7.1-116

To create a new ripout, a user in role EBROOriginator will fill in required and optional fields on the ripout web client user interface. Upon form submission the ripout server-side logic will validate that the user is authenticated and has role EBROOriginator; otherwise server will respond with HTTP 401 (unauthorized) and a limited error message.

If user is authorized, the service will validate that required fields are non-null, otherwise the server will respond with 400 (bad request) and a message stating which fields are required. When all required fields are submitted, service will generate a new Oracle ripout record. The new record will contain system-generated values for XREF, XREF\_SEQ, origination date, and will also include the values submitted by user. The new record will also contain:

ripout status = **DRFT**

open routing step = **Ripout Writer**

next routing = **Lead Trade – Approval**

The service will respond with 200 and the Ripout record in JSON form.

The four discrete drawing fields shown below are based on customer-provided mockup screens. The Ripout service will return a list of drawing field data, for the client application to parse and display.

#### RIPOUT CREATION FIELDS

Originator Required Fields	Originator Optional Fields
Work Order Number	Drawing 2
Authorizing Document Number	Drawing 2 Rev
Authorizing Document Type	Drawing 2 Applicable Pages
Drawing 1	Drawing 3
Drawing 1 Rev	Drawing 3 Rev
Drawing 1 Applicable Pages	Drawing 3 Applicable Pages
System/Component	Drawing 4
Trade Boundary, including task description	Drawing 4 Rev
	Drawing 4 Applicable Pages
	Controls Required

#### 1 RIPOUT CLIENT MOCKUP: ORIGINATOR VIEW

Active Step: Originator View		Required Field			
		Optional Field			
		Cannot Edit Field			
Ripout Detail (tab - )		Ripout Detail Add (tab - active)			
XREF	SEQ	Nuclear		Ripout_NO	
Originator Name	Originator Badge	Originator Date		Status	
Work Order	WPPN	System/Component		PSA/New	
Authorizing Document Type	Authorizing	Ship		Lead Trade	
Drawing 1	Drawing 1 Rev	Drawing 1 Applicable		OR/SEP	
Drawing 2	Drawing 2 Rev	Drawing 2 Applicable			
Drawing 3	Drawing 3 Rev	Drawing 3 Applicable			
Drawing 4	Drawing 4 Rev	Drawing 4 Applicable			
Trade Boundary					

#### *Ripout from Seed*

WRKCONT-1.7.1-117

A user in role EBROOriginator will use the client UI controls to specify that a new ripout is to be created by copying specific values from an existing ripout record. User will specify the number of the ripout seed record and will submit the ripout create request.

The ripout server-side logic will validate that the user is authorized for the EBROOriginator role, or otherwise will return 401 (unauthorized) with a limited error message. If user is authorized, the service will reply with a JSON representation of the *seed record*, limited to the columns used during unseeded ripout record creation (see table “*Ripout Creation Fields*” above).

User may edit the returned values, but *may not change the nuclear choice* from that of the seed record. User will make edits and submit the web page form to create the new ripout record. The server will confirm the restriction of an identical nuclear choice as the seed record. Server will validate that required fields are non-null then create the record. Service will respond with code 200 and the new ripout record in JSON form.

## Ripout Workflow Progression

### *Routing Steps: Open and Next*

The ripout record will store open routing step according to the nuclear and non-nuclear routing definitions. Tables of Ripout Status, Routing Steps, and Active Directory groups are available elsewhere in this document.

The service response may optionally embed workflow routing and status as hints to the client application.

This contrived example is only a suggestion:

```
{
  "ripoutNumber" : 7741B007,
  "routings" : {
    "routingPrev" : "Originator",
    "routingOpen" : "Ripout Writer",
    "routingNext" : "Lead Trade - Approval"
  },
  "status" : {
    "statusPrv" : null,
    "statusCurrent" : "DRFT",
    "statusNext" : "NPUB"
  }
}
```

When the client application attempts to modify a ripout resource, the server logic will validate that the routing is at the correct sequential position, and that the user has the proper role to modify or sign the resource.

Example 1: [workflow: non-nuclear, open routing step 12: (status NPUB, routing QAI – Piping Review, **user role EBROQAI**)].

In this example, it would be invalid for routing step 22 to sign because:

(a) 22 is not the next routing after the current step, 12;

(b) 22's role, EBROSafetyTechMechanical, is invalid for status NPUB.

Routing #	Routing Name	7G Status
12	QAI – Piping Review	NPUB
13	QAI – Mechanical Review	NPUB
14	QAI – Structural Review	NPUB
15	QAI – Electrical Review	NPUB
16	QAI – Supervisor Review	NPUB
17	QAE – Pre-Issue Review	NPUB
18	Test Department - Test Controls	STEC
19	Lead Trade – Seawater Concurrence	STEC
20	Ship Safety Officer - WIP	STEC
21	Lead Trade – WIP Concurrence	STEC
22	Safety Tech - Mechanical	STEC
23	Safety Tech - Electrical	STEC

The server will disallow resource requests or signings which are not synchronized with the workflow, and will respond with an HTTP code and error message. However, *all signature roles* can call a ripout API to add comments to any ripout sequence or XREF regardless of whether the ripout workflow is open or closed (see *WRKCONT-1.7.1-010 through 014*).

### ***Informing Client of Required User Input***

When the server responds to a valid resource request, it will send the response in JSON format. JSON objects in the response which are applicable to the current open routing will include attributes “**required**” : **true** and “**editable**” : **true** to indicate that user input is required to complete the routing step. If a field is not required but is marked as editable, then the field is **optional** for the routing step. The client application is responsible for reading these attributes and formatting the user interface accordingly.

```
{
  "ripoutNumber" : {
    "editable" : false,
    "value": "7741B007"
  },
  "tradeBoundary" : {
    "editable" : true,
    "required" : true,
```

```

        "type" : String
        "value" : "Dept 425 to do..."
    },
    "refDrawing1" {
        "required" : false,
        "editable" : true,
        "type" : String,
        "value" : ""
    },
}

```

## ***Input Validation***

When the routing step is submitted, the server will validate that all required fields for the step are present in the request, are non-null, and are of proper type and within range. When the step is signed, the system will update the ripout record with the supplied values, mark the routing step as complete, insert the signature and optional comment on the respective tables, then reply with 200.

## **Red Mat Change Process**

Refer to section K of project document [Ripouts Detail Requirements V5 3-6-19.docx](#).

Red Mat refers to the process of changing attributes of the Ripout record during an in-process ripout work flow. When user initiates a change, the client app will make specific input fields editable. User will select the change type...

- Work Scope
- Re\_Test
  - Options:
  - Test Trade Re-Test Provided
  - Engineering Provided Re-Test

... and will also edit the fields requiring a change and provide a descriptive comment. When the web form is submitted to the red mat service method, the system will perform the logic described under heading "Red Mat Processing" below.

## ***Red Mat Change Approval***

A proposed change requires several approval signatures. When the proposed change is fully approved, work can proceed as usual.

Example: user initiates a change to drawing 1 field, from H821-0202 to H821-5555:

ship #	786
authorizing document type	ER
drawing 1	H821-0202
drawing 2	
drawing 3	
drawing 4	

As described below, after a change has been initiated but is not fully accepted, the status of the active ripout record under the new sequence will be PRTL. When the ripout service returns a PRTL ripout record, the response will include information about which routings must review and sign for the change. The client app will determine how to display this routing and signature information. The following ripout UI mockups illustrate a progression of approvals. 'R' indicates that the proposed change is to be reviewed by the respective routing.

Lead Trade	Test Trade	Inspecting Trade	D461	Re-Test Trade	Ship's Force	Safety Tech Mechanical	Safety Tech Electrical	Safety Tech Electronics
R	R	R	R	R	R	R	R	R
Lead Trade	Test Trade	Inspecting Trade	D461	Re-Test Trade	Ship's Force	Safety Tech Mechanical	Safety Tech Electrical	Safety Tech Electronics

Accept Change

'A' indicates review and signature by a routing:

## 2 LEAD TRADE HAS ACCEPTED THE PROPOSED CHANGE

A	R	R	R	R	R	R	R	R
Lead Trade	Test Trade	Inspecting Trade	D461	Re-Test Trade	Ship's Force	Safety Tech Mechanical	Safety Tech Electrical	Safety Tech Electronics

Accept Change

## 3 ALL REQUIRED ROUTINGS HAVE SIGNED OFF ON THE PROPOSED CHANGE

A	A	A	A	A	A	A	A	A
Lead Trade	Test Trade	Inspecting Trade	D461	Re-Test Trade	Ship's Force	Safety Tech Mech	Safety Tech Electrical	Safety Tech Electronics

						anical		
--	--	--	--	--	--	--------	--	--

#### 4 RED MAP CHANGE ACCEPTED

ship #	786
authorizing document type	ER
drawing 1	H821-5555
drawing 2	
drawing 3	
drawing 4	

### ***Red Mat Validations***

Server should validate the following items. If validations fail, reply with 400 and limited explanation.

A red mat change cannot be initiated when Ripout status is at DRFT, LOCK, NPUB, STEC or CNCL.

Only users in roles Lead Trade, Test Trade, QAE or Super User can initiate a red mat change.

A red mat change can only be initiated upon the latest XREF Sequence.

The red mat change request may address any ripout attribute except: XREF, Sequence, Nuclear, Ripout Number, Originator Name, Originator Badge, Originator Date, Status, Ship #, SSCI, Lead Trade.

The requested change must include a non-null user comment object in the request.

### ***Red Mat Processing***

When the proposed changes and comment are submitted, the service will perform the following steps:

Add new routing step "Change Initiated" before the current open routing step and after the last signed (or partial/kicked) step.

Create a new sequence (example, if user is on XREF ABCDEF SEQ 01, create SEQ 02).

Update the status of the current ripout record to CHNG; do not apply the proposed changes to this record.

Create a new ripout record based on the submitted web form.

Set the new record status to PRTL; set the new records sequence to the new sequence value.

Insert the comment using the new next sequence number.

During a red mat change, attachments may be edited (WRKCONT-1.7.1-035).

Once required signatures (listed under *Red Mat Signatures* below) are re-signed and the open routing step is Lead Trade – Work Approval, set the ripout record status to WORK, indicating that Red Mat Initiate Change process is complete.

[EB explain: How does open routing get back to Lead Trade – Work Approval?]

## Red Mat Signatures

- a. If the record contains any of the following routing steps and they have been signed off, then copy the signatures from XREF Sequence n over to XREF Sequence n+1:
  - a.i. DRFT Ripout Writer
  - a.ii. NPUB Ship's Management – Approval
  - a.iii. NPUB Engineering Initial Review
  - a.iv. NPUB Engineering Supervisor Review
  - a.v. NPUB Planning
  - a.vi. STEC Lead Trade – Seawater Concurrence
  - a.vii. STEC Ship Safety Officer - WIP
  - a.viii. STEC Lead Trade – WIP Concurrence
  - a.ix. STEC Safety Tech - Mechanical
  - a.x. STEC Safety Tech - Electrical
  - a.xi. STEC Safety Tech - Electronics
  - a.xii. STEC Ship's Force – Description of Work
- b. If the record contains any of the following routing steps and they have been signed off, *do not copy* the signatures from XREF Sequence 01 over to XREF Sequence 02. These routing steps need to be signed again on Sequence 02:



- b.i. NPUB Lead Trade - Approval
- b.ii. NPUB QAE – Special Emphasis Review
- b.iii. NPUB NQCE – Nuclear Interface
- b.iv. NPUB QAI – Piping Review
- b.v. NPUB QAI – Mechanical Review
- b.vi. NPUB QAI – Structural Review
- b.vii. NPUB QAI – Electrical Review
- b.viii. NPUB QAI – Supervisor Review
- b.ix. NPUB QAE – Pre-Issue Review
- b.x. STEC Test Department - Test Controls
- b.xi. WORK Lead Trade – Perform Work
- b.xii. INSP Reinstallation Inspection - Piping
- b.xiii. INSP Reinstallation Inspection - Mechanical
- b.xiv. INSP Reinstallation Inspection - Structural
- b.xv. INSP Reinstallation Inspection - Electrical
- b.xvi. RFRT Retest Trade
- b.xvii. RFRT Ship's Force – Re-Test Acceptance
- b.xviii. RTON Test Director (Salary) – System Restored to Normal
- b.xix. RTON Inspecting Trade – System Restored to Normal (Hourly)

- b.xx. RTON Inspecting Trade – Ripout Restored to Normal (Salary)
- b.xxi. INST Close UNSATs
- b.xxii. INST Ship's Force – Reinstallation Inspection
- b.xxiii. AUDT QAE - Closeout Review
- b.xxiv. AUDT SUPSHIP - Final Review
- b.xxv. AUDT Ship's Force – Reinstallation Inspection

## Attachments

Requirements traced from documents:

- Ripouts Detail Requirements V5 3-6-19.docx, Work\_Control\_Process\_Flow
- Ripouts\_MDD 12.10.18 - team meeting comments.pdf

**WRKCONT-1.7.1-007:** user can add attachments using same process as 7G UNSATs.

- Any user who is authorized to sign the open routing step can add an attachment.
- Attachments must be added before the user signs.
- Need to attach files and associate them to each XREF.
- Need to store upload user, date/time stamp
- Document Types PDF, excel, word documents

**WRKCONT-1.7.1-007:** User can delete attached files.

To delete an attachment, user must be in in AD group “Attachment Deletion”.

Attachment Deletion group: EBRODeleteAttachments

Modifications will be handled by delete and add

### ***EBDT Attachments Code***

MDD section “Q. Attachments” states that files will be stored in Imagesite (the software which stores LCAD and IPDE documents). Since attachments are not deliverables and are transient, the ripout service will use existing 7g attachment logic in EBDT which already supports file attachment upload, download, add and delete. See these packages:

com.gdeb.comp.eb7g  
com.gdeb.task.eb7g  
com.gdeb.task.pcm  
com.gdeb.comp.common.doclink.ejb

The Ripout developer should study these and other packages for further insight into how to leverage the logic for Ripout use. For example, file upload logic can be found here:

com.gdeb.comp.eb7g.EB7GItem.uploadAttachment(EB7GAttachment)  
com.gdeb.task.eb7g.EB7GHttpClient.executeFileUpload(EB7GAttachment)

The EB7GHttpClient mentioned above is not a web browser. It is an EBDT class which wraps org.apache.commons.httpclient.HttpClient, a convenience library for HTTP transport methods. The Ripout service will define a POST endpoint consuming MediaType.MULTIPART\_FORM\_DATA for uploading file attachments.

### ***EBDT File Attachment Code***

*This section is preliminary as of 29Aug2019 and needs revision.*

The EB7GAttachments dialog presents a list of attachment objects. Since the dialog has no capability to add attachments, it appears that attachment data was added from Tandem via the inbound message queue shown below in processInboundMessage().

Attachments dialog uses EB7GItem (a BusinessObject specialization) to pass information . Dialog calls methods on EB7GSessionProxy (façade for EB7GSessionEJB). EB7GSessionEJB contains an instance of EB7GItemDAO which performs view, delete, update, and kick operations.

This project would add logic to the ejb and dao to upload binary file attachments (formats: PDF, Excel, Word).

### **View Attachment List**

```
com.gdeb.comp.eb7g.EB7GAttachmentsDialog:
  .EB7GAttachmentsPanel: // ui components
    +onPrint()
    +onView()
      +httpClient.executeFileDownload() to get URL and pop a browser window in ebdt
    +onDelete()
      +FWElectronicSignatureDialog() // verify user can delete

520: EB7GSessionProxy.deleteAttachment(...)
3783: Oracle call eb7g.documents.filedelete(?,?,?)
```

### Logical path to saving an attachment binary object:

EB7GKickDialog

357: EB7GItem.kick(...)

1644: EB7GSessionProxy.kick(...)

338: EB7GSessionEJB.saveBFile(...)

864: EB7GSessionEJB.addAttachment(...)

Uses EB7GAttachment to contain parameters of the object to attach. Perhaps can use this as a model to define a RipoutAttachment class.

EB7GSessionEJB

+kick(...EB7GAttachment attObj)

358: saveBFile(EB7GItem item, EB7GAttachment attObj,...)

...

864: saveBFile(EB7GItem item, EB7GAttachment attObj,...)

...

878: EB7GItemDAO.addAttachment(... EB7GAttachment attObj);

EB7GItemDAO.addAttachment(...):

## 3746: stmt = con.prepareCall("begin eb7g.documents.fileput(?,?,?,?);end;");

## 3752: stmt.setBinaryStream(6, iStream, (int)file.length());

### Add attachment from inbound MQ

+processInboundMessage

+ processInboundMessage()

642: EB7GItemDAO.addExtFromTandem( con, tranMsg );

+ EB7GItemDAO:

1407 addExtFromTandem()

...

```
3730 addAttachment(..., EB7GAttachment attObj)
3746: stmt = con.prepareCall("begin eb7g.documents.fileput(?,?,?,?,?);end;");
3752: stmt.setBinaryStream(6, iStream, (int)file.length());
```

## Comments

User comments will be stored in a dedicated comment table, apart from the signature table.

wrkcont-1.7.1-009: Comment is optional during routing signature

wrkcont-1.7.1-012: Comment can be added for any sequence number.

wrkcont-1.7.1-013: Comment can be added for XREF, open or closed.

wrkcont-1.7.1-014: Comment is required during partial signing.

wrkcont-1.7.1-015: Comment is required during kick back signing.

A comment is required during Red Mat initiation.

## JSON / Java Binding

The Ripout client application will produce and consume JSON. To expose Java objects as JSON, the Ripout application will use the JAXB API implementation, “MOXy”, included in the EclipseLink library distributed with WebLogic 12.x. For details, see:

<https://docs.oracle.com/middleware/1221/toplink/develop-document-bindings/json>

Optionally, Ripout service could use the JAX-RS API for JSON/Java binding. The Java executable library reference implementation of JAX-RS is “Jersey”. If using Jersey, then WebLogic Server supports JAX-RS as follows: <sup>3</sup>

<b>WebLogic 12.2.1.3</b>
JAX-RS-2.0 (aka Jersey 2.x) No shared library required.
Jersey 2.22.4

## Ripout Web Service Packaging

The Ripout services module will be packaged as a Web application in a .war file. Related ripout application classes are packaged in WEB-INF/classes or WEB-INF/lib and required libraries are packaged in WEB-INF/lib. Ripout.war will be built by the EBDT build script, bundled into the EBDT EAR file and deployed to WebLogic Server. For further information, see the Servlet specification regarding web application packaging, and Weblogic Server middleware documentation regarding deployment. <sup>4</sup>

### *Web Service Security*

Security configuration of the Ripout web service will be accomplished by standard Java web application security declarations in the Ripout web.xml deployment descriptor. Application server-side resource access control will be provided by legacy RBAC services in EBDT.

### *HTTP Standard Methods*

Ripout web services will mimic server-side CRUD operations over HTTP by specifying URI’s combined with standard HTTP methods:

HTTP Method	Server-Side Function
-------------	----------------------



GET	Retrieve existing resource
POST	Create a new resource
PUT	Update an existing resource
DELETE	Delete a resource

## ***Enumerations***

Enumerated constants will be declared in all-caps. These constants will be used in server-side code. If they also appear in the response body they will be accompanied by a full expansion so that client will not need to interpret the abbreviations.

```
enum STATUS {
    DRFT, NPUB, STEC, WORK, INSP, RFRT, INST, RTON, AUDT, CLSD, CNCL
}
```

## ***Resource Discovery***

The Ripout client application should have no prior knowledge of the service API or the different steps involved in the ripout workflow. HATETOAS (Hypermedia as the Engine of Application State) is an application architectural style which supports the use of hypermedia links in the response content so that the client can dynamically navigate to the appropriate resource by traversing the hypermedia links. To avoid the need for hard-coded links in client code and to support client discovery of Ripout workflow routing steps, resource relationships and resource links may be embedded within the service response body.

This example is an excerpt of a hypothetical response from GET `https://ebdt.ebnet.gdeb.com/ripout/7741B007`.

Here, the embedded links can be discovered and used by the client application.

```
{
  "_links": {
    "self": {
      "href": "/ripout/7741B007"
    },
    "signatures": {
      "description" : "Return list of signatures",
      "href" : "7741B007/signatures",
      "type" : "GET"
    }
  },
  "attachments": {
    "description" : "Return list of attachments",
    "href" : "7741B007/attachments",
    "type" : "GET"
  }
}
```

```
}  
}
```

## Ripout API

The Ripout application programming interface will follow RESTful design principles:

- Client display of resources shall be independent of and unaware of server implementation.
- A service should be self-describing; client shall need no prior knowledge or out-of-band information other than the URI and media types.
- Client-Server interaction shall be stateless.
- State transitions shall be driven by client requests on ripout API endpoints, or driven by application of hyperlinks embedded within the endpoint's response.<sup>5</sup>
- Response data shall not contain server-side implementation details, i.e., server-side data types, database keys, etc.
- API shall not expose typed resources with significance to the client, other than media type and relation names.
- API shall not define fixed resource names or hierarchies, but shall use path variables for flexibility.
- URI's shall model ripout business capabilities.
- URI's shall be designed in a flexible, clear and uniform manner.
- URI's may model database CRUD operations or may declare custom methods where necessary.<sup>6</sup>

The ripout server will not store any state about the client session; the client is responsible for managing application state on the client side.

Each ripout API request must contain all information necessary for the server to understand the request and must not presume any stored context information on the server.

All state necessary to handle a single ripout resource request shall be contained *within* the request: the state may be within part of the URI, it may be in the query parameters, in the request body, in the headers or in combination of these.

The ripout service will perform application logic and then return a JSON representation of the server's current version of the requested resource. Therefore, when the client holds a representation of a resource, then the client has enough information to further modify the resource by calling other URI's.

## API Endpoints

The following Ripout use cases and endpoints are derived from Task 85 project document **Ripouts Attachment 3 – Use Cases.xlsx**.

These URI's describe the client view of business cases without describing any server-side implementation.

Provide a default end point to handle unmatched requests. This example from ElcaddService.java.

```
@GET
@Path("/{s:.}*}")
public Response defaultGetEntry() {
    log.info("Default GET Entry");
    return errorRsp("An Error Occurred", "The server has not found anything matching
the Requested URI.");
}

@POST
@Path("/{s:.}*}")
public Response defaultPostEntry() {
    log.info("Default POST Entry");
    return errorRsp("An Error Occurred", "The server has not found anything matching
the Requested URI.");
}
```

Req #	JAXB Class	Use Case Name	Summary	URI Endpoint	Data	Return	Note
1  WRKCON T-1.7.1-	L	Create a Ripout from	Allow user to write a new ripout from scratch (no fields are pre-filled for the	POST /ripout	Post data: {ripoutType: [nuclear   nonnuclear], authorizingDocumentType], authorizingDocumentNumber,	JSON ripout record	

016		scratch	Ripout Originator)		shipNumber, drawingNumber, tradeBoundary}		
2	L	Create a Ripout from seed	Ripout Originator and Writer fields are pre-filled by service.  Ripout nuclear type cannot be changed from original seed type.	POST /ripout/{ripoutNumber}	{ripoutNumber}: URI param identifying the ripout seed record, from which to populate new ripout record.	JSON ripout record	
WRKCON T-1.7.1-017 WRKCON T-1.7.1-018							
3.1	R	Initiate a Red Mat change	While the Ripout is at WORK status, allow a user to make an attribute change that must be approved by others before proceeding with work.	POST /ripout/{ripoutNumber}/redmat/initiateChange	Request data: {comment}		See document:  "Attachment 4 - Red Mat Change Process. docx"
3.2	L	Cancel a Red Mat change	Red mat indicates a proposed change to a ripout record field. Others must sign off before the change is permanently applied to the ripout record.	DELETE /ripout/{ripoutNumber}/redmat	Reply with ripout record version prior to suggested red mat change.		

			The suggested change will be stored in a table separate from the ripout record. When all approvals have been captured, the change will be permanently applied to the ripout record.				
4	L	Cancel a Ripout	A ripout record with a prior status of AUDN or AUDT may achieve this status to indicate the ripout is canceled. This action causes the item record reserved for ripout processing to be deleted. No further activity may occur.	<b>PUT /ripout/{ripoutNumber}/cancel?reason=reason</b>	Request data: <pre>{   reason: [     noWorkPerformed       superseded&amp;supercededby={supersedingRipoutNumber}   ] }</pre>		If reason is that ripout is superseded, then provide that ripout number.
5	L	Delete a Ripout	After a Ripout number has been assigned, allow user to delete Ripout	<b>DELETE /ripout/{ripoutNumber}</b>			
6	W	Initiate a Stop Work on a	A ripout with a prior status of STEC, WORK may achieve this status to indicate that all WORK has been	<b>PUT /ripout/{ripoutNumber}/?action=stop</b>	See section M of project document Ripouts Detail Requirements V5 3-6-19.docx		

		Ripout	stopped and will resume at a later date.				
7	W	Initiate a Resume Work on a Ripout	A ripout with a prior status of STOP may achieve this status to indicate that all WORK can resume	<b>PUT /ripout/{ripoutNumber}/?action=resume</b>	See section M of project document Ripouts Detail Requirements V5 3-6-19.docx		
8, 9	L	Search for an existing Ripout (in work or closed)	Search for Ripout by the following field(s): XREF Ripout # Ship System (SSCI #) Nuclear or Non-Nuclear	<b>GET /ripout?xref={xref}&amp;ship={sip}&amp;ripoutNumber={ripoutNumber}&amp;drawingNumber={drawingNumber}&amp;authDocType={authDocType}&amp;ssci={ssci}&amp;status={status}&amp;leadTrade={leadTrade}&amp;openStep={step}&amp;nuclear={[nuclear   non-nuclear   enhanced]],workOrder={workOrder}</b>	Query parameters will vary, server must check each parameter for non-null to build DB query.	JSON ripout record	Ripout MDD lists 5 criteria; Ripout detail req's doc lists 11 criteria.
10	L	Open a closed ripout (see note)	Ripout is closed and needs to be opened back up for a specific reason.	<b>GET /ripout/{ripoutNumber}reopen?reason={reason}</b>	Request data: {reason} one of: [ reTest, attributeChange, oqeChange, attachmentChange]	JSON ripout record	Consolidates requirements #10: wrong

		column).					re-test, #11: wrong attrib data, #12: wrong OQE, #13: wrong attachm ents)
17		View attribute data	Top portion of ripout form	<b>GET /ripout/{ripoutNumber}/attributes</b>	n/a	<b>JSON ripout attributes</b>	
18	I	View routing information	A list of nuclear or non-nuclear routing steps.	<b>GET /ripout/{ripoutNumber}/routings&amp;nuclear=true false</b>	n/a	<b>JSON routing steps</b>	
19	S	View signatures of who signed the ripout before	User's name, badge, department and the signature date, time (5/25/17 16:01) along with signature type	<b>GET /ripout/{ripoutNumber}/signatures</b>	n/a	<b>JSON list of signatures</b>	

		e them					
20.0  WRKCON T-1.7.1- 015	C	View comm ents on ripout	Allow user to view existing comments anytime	<b>GET /ripout/ {ripoutNumber}/comm ents</b>	n/a		
20.2  WRKCON T-1.7.1- 009 WRKCON T-1.7.1- 010 WRKCON T-1.7.1- 012 WRKCON T-1.7.1- 013 WRKCON T-1.7.1- 014	C	Add comm ent.	Whether ripout is closed or open, user may add comment to any sequence, regardless of current (latest) sequence.				



21 WRKCON T-1.7.1- 005	W	View history of attributes for each XREF by sequence		GET /ripout/{ripoutNumber}/history		List of Ripout attributes for each sequence of an XREF.	
22	A	View attachments	Allow user to download to view locally anytime	GET /ripout/{ripoutNumber}/attachments		JSON list of attachments or URLs pointing to location of attachments.	
23.1	W	User signs active step	User agrees with attribute data and wants to sign off for their active step	POST /ripout/{ripoutNumber}/signature			User signing information is in ripout record.

							<p>Server logic knows current routing step.</p> <p>Insert signature record.</p>
23.2 WRKCON T-1.7.6-015	W	Sign and Kick	The system shall capture user name, badge, and department, date & time. The system shall mark signature with K, close open routing step and open the previous sequential routing step. Comment is required.	<b>POST /ripout/{ripoutNumber}/kick</b>	n/a	<b>303 location header pointing to ripout resource with for previous routing step.</b>	

23.3 Ripouts Detail Requirements v3-6-19, section P. Signatures	W	Partial Sign	User may partially sign if has correct role during open step.	<b>POST /ripout/{ripoutNumber}/signaturePartial</b>			
48.1 WRKCON T-1.7.6-007	A	Allow ripout writer to attach highlighted copies of applicable pages from a drawing, that ultimately helps the foreman accom	Attachment capability	<b>POST /ripout/{ripoutNumber}/attachments</b>	<b>Post data: multipart/form-data</b>  <b>Service consumes MediaTypd.MULTIPART_FORM_DATA</b>		<b>user may attach and upload pages from a drawing .</b>

		plish work					
48.2 WRKCON T-1.7.6- 008	A	Allow user to delete attach ments , using same attach ment proce ss as 7G UNSA Ts.					
WRKCON T-1.7.1- 026	I	Get refere nce links.	Links to SSP attachments: Retest forms, Test form, Tagouts, WIP's, NDT sheets, QAL sheet, Re-Entry Briefing Sheet	<b>GET /ripout/ {ripoutNumber}/refLink s</b>			

Ripouts Detail Requirements v3-6-19, section S. Notifications	I	Email Notification	Need Email notification to all users within a role when a step is open to the role. When a routing step is available to be signed for, the system shall send an email to the all users who are in that role.	No Ripout endpoint. This is for ripout processing by server logic.			
Ripouts Detail Requirements v3-6-19, section R.	I	Get list of ripouts	Users need list of ripout numbers by status and active step for user. Need to see active step, lead trade, nuclear/non-nuclear.				

## **Ripout Object Model**

Ripout services will employ the “FW2” EB Desktop framework model. FW2 defines business behaviors in the Business Object and provides persistence and utility services via a Proxy + Stateless EJB + DAO abstraction. The UML diagram below is based on the legacy BusinessObject persistence model within EBDT. Limited sample methods and properties are merely hints in this diagram. For details about endpoints and object properties and methods, see the sections for endpoints and ripout record.

### ***RipoutService***

Exposes ripout service endpoints for client use. Decorated with JAX-RS annotations declaring RESTful properties. Looks up the RipoutSession by JNDI name, causing WebLogic container to instantiate a RipoutSessionEJB.

### ***RipoutSession***

Defines JNDI\_HOME used by WebLogic to create an instance of the Ripout EJB.  
Declares abstract methods for implementation in RipoutSessionEJB.

### ***RipoutSessionEJB***

Implements abstract methods declared in RipoutSession with supporting business logic for ripout database transactions.  
Maintains a reference to a RipoutPO and a reference to a RipoutDao.  
Passes RipoutPO reference to RipoutDao and calls various DML methods on RipoutPO.  
Performs marshalling between RipoutPO and RipoutService.

### ***RipoutPO***

Persistent object representing a single ripout record and its child record dependencies.

### ***JAXB***

Maps Java objects to JSON and XML representation for client use. Recommend using EclipseLink 2.4.2 or greater, which supports the output of a grouping element around collection data: ‘widgets’: [‘widget-01’, ‘widget-02’]  
See @XmlElementWrapper annotation. Also see WebLogic documentation for details regarding the application-specific deployment of JAXB-supporting JAR files which WebLogic includes by default.

## ***RipoutDao***

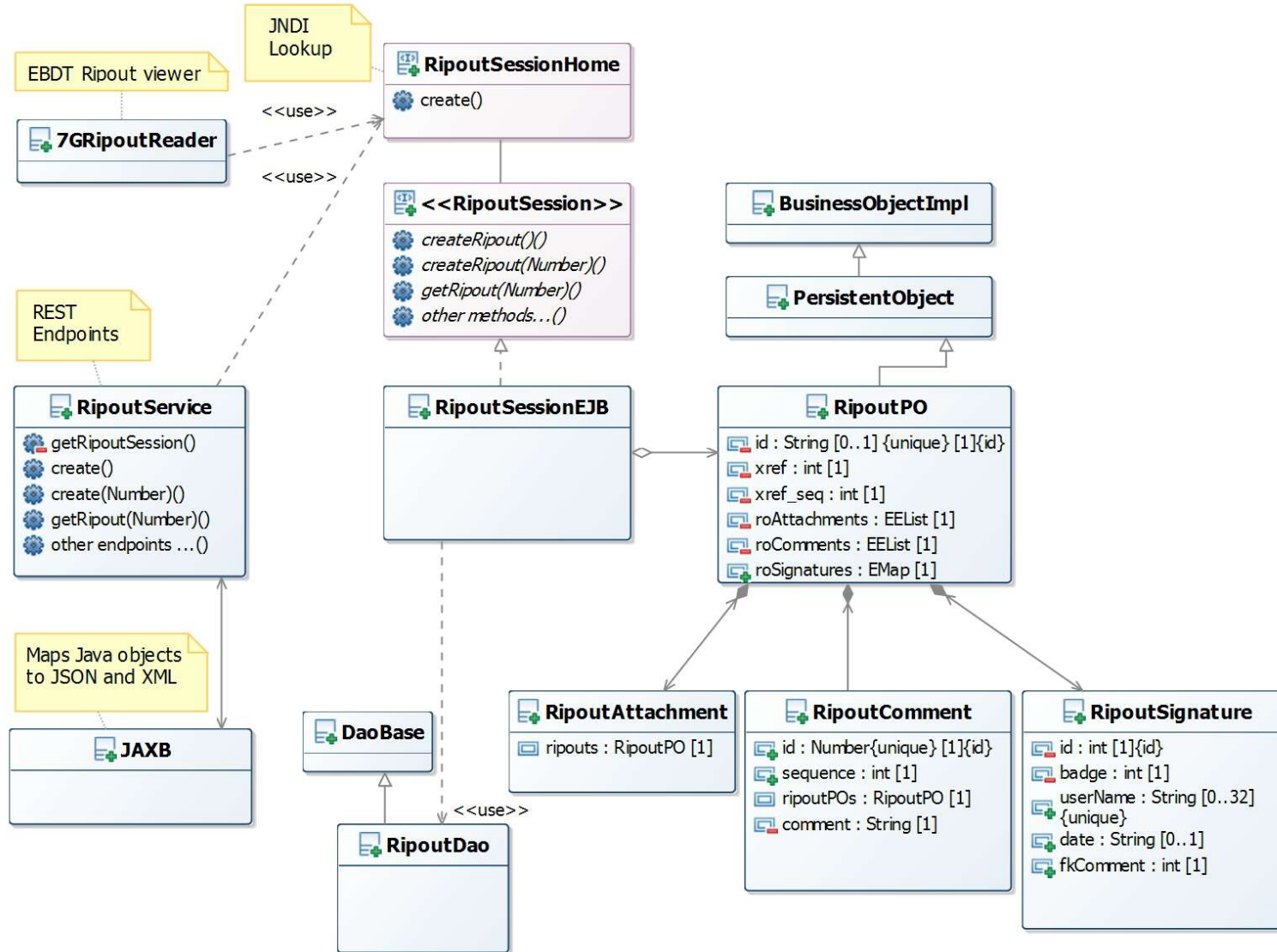
Performs DML operations on a RipoutPO. Contains predefined queries or query fragments. May contain various Java Map instances (signatureMap, attachmentMap, commentMap) for reading objects from result sets.

## ***7GRipoutReader***

Interface for serving read-only Ripout records in EBDT.

## ***RipoutComments***

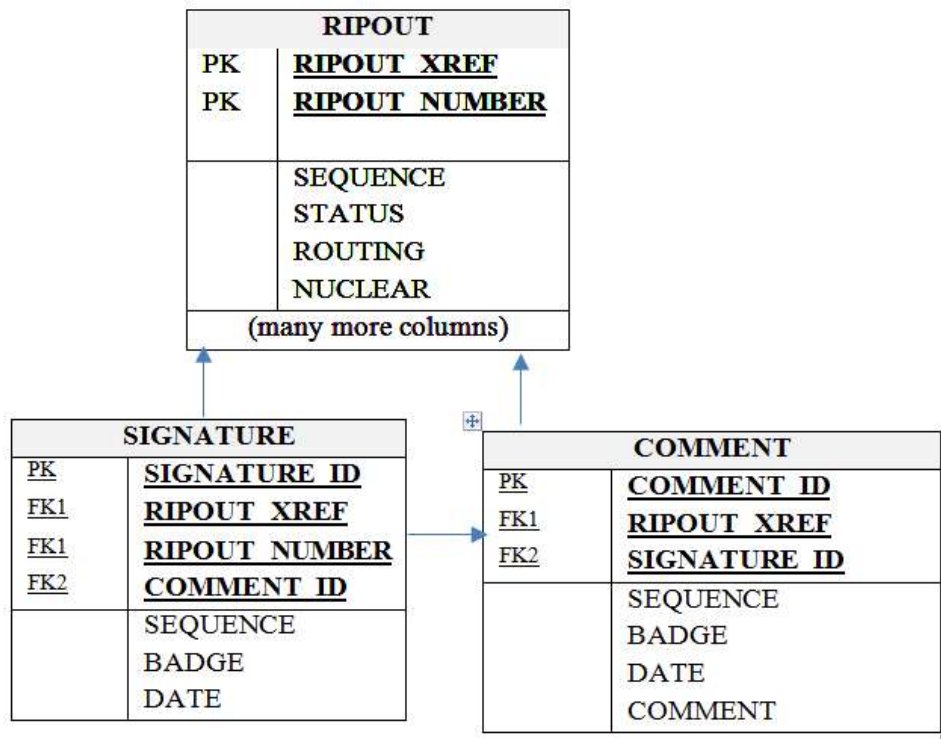
**extends** BusinessObjectCollectionImpl see EB7GComment



## Ripout Data Model

Simplified relational model, does not consider ancillary tables such as ripout list, trade boundary, etc.





- Signature *may* include a comment.
- Comment *may* be added without a signature.
- Routings tables for lookup only.]

NON-NUCLEAR ROUTING LIST	
<u>PK</u>	<u>ROUTING ID</u>
	ROUTING_NUMBER
	ROUTING_NAME
	ROUTING_STATUS

NUCLEAR ROUTINGLIST	
<u>PK</u>	<u>ROUTING ID</u>
	ROUTING_NUMBER
	ROUTING_NAME
	ROUTING_STATUS

The follow details are from project document: Ripouts Detail Requirements V5 3-6-19.docx

### ***Assigning XREF (Primary Key)***

- After the Ripout Originator (first step in the process) enters their data and then saves, need to assign a unique system ID (primary key) to each ripout in order to insert the record into the Oracle table
  - When a ripout is created, the system shall assign a unique XREF (increase sequentially from the previous XREF #)
  - Format will be numeric, 6 characters
- Need to include a sequence number for each XREF
  - When a ripout is created, the system shall assign a sequence number, starting with sequence = 01
  - *JBelliveau example: if six routings into workflow then they have to startover, they use same xref, but new sequence. In ers there is "deprogressing" a record. If someone changes a critical field, then they saved everything to that point on that record; they make a new record and increment the sequenece and can change anything, but have an audit trail.*
- Need to see a history of attribute data (audit trail) for each XREF by sequence
  - The system shall allow a user a view all sequences of an XREF
- Need to allow the user to modify data on the latest XREF Sequence only
  - The system shall allow a user to modify attribute data on the latest XREF Sequence only.  
The system shall not allow a user to modify attribute data on XREF Sequences that are not the latest.

## ***Assigning Ripout List Number***

Describes creation of successive ripout workflow records for a given ripout XREF.

- wrkcont-1.7.1-033: Need to assign Ripout List #
  - The system shall assign a Ripout List number after the Ripout Writer (2<sup>nd</sup> step) step signs off.  
(wrkcont-1.6-020: not allowed to created ripout list number if ship is on hold aka “lock”).
  - The system shall assign the Ripout List number as "RO" + "SSCI Number" + " " + "001".
    - Example: "RO82110 001"
    - SSCI = Ship System Code Index number (field entered by Ripout Writer)
  - The first record created for that Ship and SSCI Number shall be assigned as List = "001".  
Each sequential record created for that Ship and SSCI Number shall be assigned a List number incremented by 1.
    - Example: "RO82110 002"

After a ship is delivered (PSA [Post Shakedown Availability]/New Construction field [entered by Ripout Writer] = PSA) system shall assign the Ripout List number as "RO" + "SSCI Number" + "B" + "001". Need to add a table for the software to check delivery

Example: "RO82110B001"

## Workflow Routings

### *Non-Nuclear Routings*

Routing #	Non-Nuclear Routing Name	7G Status
1	Ripout Writer	DRFT
2	Lead Trade - Approval	NPUB
3	Ship's Management - Approval	NPUB
4	NQCE - Nuclear Interface	NPUB
5	QAE – Special Emphasis Review	NPUB
6	Engineering Initial Review	NPUB
7	Engineering Supervisor Review	NPUB
8	QAE – Reentry Control Document Number	NPUB
9	Planning - Blue Tags	NPUB
10	QAI – Piping Review	NPUB
11	QAI – Mechanical Review	NPUB
12	QAI – Structural Review	NPUB
13	QAI – Electrical Review	NPUB
14	QAI – Supervisor Review	NPUB
15	QAE – Pre-Issue Review	NPUB
16	Test Department - Test Controls	STEC
17	Lead Trade – Seawater Concurrence	STEC
18	Ship Safety Officer - WIP	STEC
19	Lead Trade – WIP Concurrence	STEC
20	Safety Tech - Mechanical	STEC
21	Safety Tech - Electrical	STEC
22	Safety Tech - Electronics	STEC
23	Ship's Force – Description of Work	STEC
24	Lead Trade – Perform Work	WORK

25	Inspecting Trade – Piping Inspection	INSP
26	Inspecting Trade – Mechanical Inspection	INSP
27	Inspecting Trade – Structural Inspection	INSP
28	Inspecting Trade – Electrical Inspection	INSP
29	QAI – Piping Inspection	INSP
30	QAI – Mechanical Inspection	INSP
31	QAI – Structural Inspection	INSP
32	QAI – Electrical Inspection	INSP
33	Re-Test Trade	RFRT
34	Ship's Force – Re-Test Acceptance	RFRT
35	Test Director (Salary) – System Restored to Normal	RTON
36	Inspecting Trade – System Restored to Normal (Hourly)	RTON
37	Inspecting Trade – System Restored to Normal (Salary)	RTON
38	Close UNSATs	INST
39	QAE – Closeout Review	AUDT
40	SUPSHIP – Final Review	AUDT
41	Test Engineering – Embedded Testing Required?	AUDT
42	Nuclear Ship's Management - Approval	STEC
43	Nuclear Test – Controls Required	STEC

## ***Nuclear Routings***

<b>Routing #</b>	<b>Nuclear Routing Name</b>	<b>7G Status</b>
1	Ripout Writer	DRFT
2	Lead Trade - Approval	NPUB
3	Ship's Management - Approval	NPUB
4	NQCE - Nuclear Interface	NPUB
5	QAE – Special Emphasis Review	NPUB
6	Nuclear Ship's Management - Approval	STEC
7	Nuclear Test – Controls Required	STEC
8	Safety Tech - Electrical	STEC
9	Safety Tech - Mechanical	STEC
10	Ship's Force – Description of Work	STEC
11	Nuclear Test - WIP	STEC
12	Ship's Force - WIP	STEC
13	Lead Trade – WIP Concurrence	STEC
14	Lead Trade – Perform Work	WORK
15	NQCI – Reinstallation Inspection	INSP
16	Ship's Force – (2) Installation Only	INSP
17	Nuclear Test – Re-test Req'd 1	RFRT
18	Nuclear Test – Re-test Req'd 2	RFRT
19	Nuclear Test –Ready for NCQI Retest Inspection	RFRT
20	NQCI – Re-Test Inspection	RFRT
21	Nuclear Re-Test Completed	RFRT
22	Close UNSATs	RFRT
23	Ship's Force – (3)	RFRT
24	Re-Test Accepted	RFRT
25	Nuclear Test – Restore to Normal	RTON

## Appendix 1 – Database Tables

Data definitions subject to change with DBA involvement.

### *Ripout Record*

Delivery database: RIPOUT.RIPOUT\_RECORD

<i>Mock</i>	<i>COLUMN_NAME</i>	<i>DATA_TYPE</i>	<i>N U L L A B L E</i>	<i>DATA_DEFAULT</i>	<i>COLUMN_I D</i>
X	XREF	NUMBER(38, 0)	No	(null)	1
X	XREF_SEQ	NUMBER(38, 0)	No	(null)	2
X	NUCLEAR	CHAR(1 BYTE)	No	''	3
	USER_ID	CHAR(15 BYTE)	No	''	4
X	USER_BADGE	CHAR(9 BYTE)	No	''	5
X	CREATE_DATE	DATE	No	sysdate	6
X	RIPOUT_STATUS	CHAR(4 BYTE)	No	''	7
X	WORK_ORDER	CHAR(15 BYTE)	No	''	8
X	WPPN	CHAR(15 BYTE)	No	''	9
X	WO_SHIP	CHAR(3 BYTE)	No	''	10
X	WO_LEAD_TRADE	CHAR(3 BYTE)	No	''	11
X	AUTH_DOC_TYPE	CHAR(4 BYTE)	No	(null)	12
X	AUTH_DOC_NUM	CHAR(4 BYTE)	No	(null)	13
x	SYSTEM_COMP	CHAR(1 BYTE)	No	(null)	14
X	TRADE_BOUNDARY	CHAR(200 BYTE) verify size	No	(null)	15
x	DRAWING1	CHAR(15 BYTE)	No	''	16
x	DRAWING1_REV	CHAR(2 BYTE)	No	''	17
x	DRAWING1_PGS	CHAR(9 BYTE) verify length	No	''	18
x	DRAWING2	CHAR(15 BYTE)	No	''	19
x	DRAWING2_REV	CHAR(2 BYTE)	No	''	20
x	DRAWING2_PGS	CHAR(9 BYTE) verify length	No	''	21
x	DRAWING3	CHAR(15 BYTE)	No	''	22

<b>Mock</b>	<b>COLUMN_NAME</b>	<b>DATA_TYPE</b>	<b>N U L L A B L E</b>	<b>DATA_DEFAULT</b>	<b>COLUMN_I D</b>
x	DRAWING3_REV	CHAR(2 BYTE)	No	''	23
x	DRAWING3_PGS	CHAR(9 BYTE) verify length	No	''	24
x	DRAWING4	CHAR(15 BYTE)	No	''	25
x	DRAWING4_REV	CHAR(2 BYTE)	No	''	26
x	DRAWING4_PGS	CHAR(9 BYTE) verify length	No	''	27
	CONTROLS_REQ	CHAR(1 BYTE)	No	''	28
x	SSCI_NUM	CHAR(5 BYTE) VERIFY	No	''	29
X	WRK_SCP_PIPING	CHAR(1 BYTE)	No	''	30
X	WRK_SCP_MECHAN	CHAR(1 BYTE)	No	''	31
X	WRK_SCP_STRUCT	CHAR(1 BYTE)	No	''	32
X	WRK_SCP_ELECTRIC	CHAR(1 BYTE)	No	''	33
x	OPER_RESP	CHAR(1 BYTE)	No	''	34
x	LEVEL_1	CHAR(1 BYTE)	No	''	35
x	SUBSAFE	CHAR(1 BYTE)	No	''	36
x	DSS_SOC	CHAR(1 BYTE)	No	''	37
x	SFCC	CHAR(1 BYTE)	No	''	38
x	FLY_BY_WIRE	CHAR(1 BYTE)	No	''	39
40X	OCT	CHAR(3 BYTE) Verify size	No	''	40
X	WAD	CHAR(3 BYTE)	No	''	41
X	LOCATION	CHAR(4 BYTE)	No	''	42
X	LEVEL	CHAR(1 BYTE)	No	''	43
X	P_S_C	CHAR(1 BYTE)	No	''	44
X	FRAME	CHAR(7 BYTE) Verify size	No	''	45
X	REASON	CHAR(50 BYTE) Verify size	No	''	46
X	BLUETAG_REQ	CHAR(1 BYTE)	No	''	47
X	NDT_REQ	CHAR(1 BYTE)	No	''	48
X	VEND_WRK_PERFORM	CHAR(1 BYTE)	No	''	49
50x	TEST_TRADE	CHAR(3 BYTE)	No	'275'	50
X	RETEST_REQ	CHAR(1 BYTE)	No	''	51



<b>Mock</b>	<b>COLUMN_NAME</b>	<b>DATA_TYPE</b>	<b>N U L L A B L E</b>	<b>DATA_DEFAULT</b>	<b>COLUMN_I D</b>
X	RETEST_REQUIREMENTS	CHAR(20 BYTE)Verify size	No	''	52
X	RETEST_TRADE1	CHAR(3 BYTE)	No	''	53
X	RETEST_TRADE2	CHAR(3 BYTE)	No	''	54
X	RETEST_TRADE3	CHAR(3 BYTE)	No	''	55
X	TANK_CLS_REQ	CHAR(1 BYTE)	No	''	56
X	KEY_EVENT	CHAR(3 BYTE)	No	''	57
X	REF_DRAWING1	CHAR(15 BYTE)	No	''	58
X	REF_DRAWING1_REV	CHAR(2 BYTE)	No	''	59
60x	REF_DRAWING1_PGS	CHAR(9 BYTE)Verify size	No	''	60
X	REF_DRAWING2	CHAR(15 BYTE)	No	''	61
X	REF_DRAWING2_REV	CHAR(2 BYTE)	No	''	62
X	REF_DRAWING2_PGS	CHAR(9 BYTE)Verify size	No	''	63
X	REF_DRAWING3	CHAR(15 BYTE)	No	''	64
X	REF_DRAWING3_REV	CHAR(2 BYTE)	No	''	65
X	REF_DRAWING3_PGS	CHAR(9 BYTE)Verify size	No	''	66
X	REF_DRAWING4	CHAR(15 BYTE)	No	''	67
X	REF_DRAWING4_REV	CHAR(2 BYTE)	No	''	68
X	REF_DRAWING4_PGS	CHAR(9 BYTE)Verify size	No	''	69
70x	INSPECTING_TRADE	CHAR(3 BYTE)	No	''	70
X	TAG_REC_SHEET_REQ	CHAR(1 BYTE)	No	''	71
X	NUCLEAR_INTERFACE	CHAR(1 BYTE)	No	''	72
X	REC_SEQ_NUM_REQ	CHAR(1 BYTE)	No	''	73
X	RECORD_SEQ_NUM	CHAR(2 BYTE)Verify size	No	''	74
X	QAL_CERT_REQ	CHAR(1 BYTE)	No	''	75
	QAL_CERTIFICATION	CHAR(8)Check May be an attachment	No	(null)	76
X	TAGOUT_REQ	CHAR(1 BYTE)	No	''	77
X	HAZ_ENERG_TAGOUT_REQ	CHAR(1 BYTE)	No	''	78
X	WHO_HAS_OPCON	CHAR(10 BYTE)Verify size	No	''	79

<i>Mock</i>	<i>COLUMN_NAME</i>	<i>DATA_TYPE</i>	<i>N U L L A B L E</i>	<i>DATA_DEFAULT</i>	<i>COLUMN_I D</i>
80x	WIP_REQ	CHAR(1 BYTE)	No	''	80
X	WIP_NUM	CHAR(1 BYTE)No info available on size	No	''	81
X	RADCON_REQ	CHAR(1 BYTE)	No	''	82
(x)	SAFETY_TECH_MECH	CHAR(1 BYTE)	No	''	83
(x)	SAFETY_TECH_ELEC	CHAR(1 BYTE)	No	''	84
(x)	SAFETY_TECH_TRNX	CHAR(1 BYTE)	No	''	85
X	SEAWATR_CON_SYS	CHAR(1 BYTE)	No	''	86
X	EMBED_TEST_REQ	CHAR(1 BYTE)	No	''	87
88X	RIPOUT_LIST_NUM	CHAR(11 BYTE)	No	''	88

### *Ripout Record Attributes*

<i>Attribute name</i>	<i>information</i>	<i>Type</i>	<i>Required (user input)</i>	<i>Editable state</i>	<i>Validating</i>
Ripout_state	internal use to hold current state index	Short	na	none	
NUCLEAR		Boolean	true	Originating	
ORIGINATOR_NAME	stored from Context on EJB Create	String	na	none	
ORIGINATOR_BADGE	stored from Context on EJB Create	String	na	none	
ORIGINATOR_DATE	stored from current Date on EJB Create	String	na	none	

<b><i>Attribute name</i></b>	<b><i>information</i></b>	<b><i>Type</i></b>	<b><i>Required (user input)</i></b>	<b><i>Editable state</i></b>	<b><i>Validating</i></b>
RIPOUT_STATUS	internally updated on current state signoff action	String	na	none	
WORK_ORDER		String	true	Originating Ripout Writer	cannot be blank verify that wo exists in system and status < C
WPPN	internally derived on lookup keyed to Work_Order	String	na	none	
SHIP	internally derived on lookup keyed to Work_Order	String	na	none	
LEAD_TRADE	internally derived on lookup keyed to Work_Order	String	na	none	
AUTH_DOC_TYPE		String	true	Originating Ripout Writer	cannot be blank
AUTH_DOC_NUMBER		String	true	Originating Ripout Writer	cannot be blank
DRAWING1		String	true	Originating Ripout Writer QAE - Special Emphasis Review QAE - Pre-Issue Review	cannot be blank
DRAWING1_REV		String	true	Originating Ripout Writer QAE - Special Emphasis Review QAE - Pre-Issue Review	cannot be blank
DRAWING1_APPL_PGS		String	true	Originating Ripout Writer QAE - Special Emphasis Review QAE - Pre-Issue Review	cannot be blank

<b><i>Attribute name</i></b>	<b><i>information</i></b>	<b><i>Type</i></b>	<b><i>Required (user input)</i></b>	<b><i>Editable state</i></b>	<b><i>Validating</i></b>
SYSTEM_COMP		String	true	Originating Ripout Writer	cannot be blank
TRADE_BOUNDARY		String	true	Originating Ripout Writer QAE - Special Emphasis Review QAI - Piping Review QAI - Mechanical Review QAI - Structural Review QAI - Electrical Review QAI - Supervisor Review QAE - Pre-Issue Review	cannot be blank
DRAWING2		String	false	Originating Ripout Writer QAE - Special Emphasis Review QAE - Pre-Issue Review	
DRAWING2_REV		String	false	Originating Ripout Writer QAE - Special Emphasis Review QAE - Pre-Issue Review	
DRAWING2_APPL_PGS		String	false	Originating Ripout Writer QAE - Special Emphasis Review QAE - Pre-Issue Review	
DRAWING3		String	false	Originating Ripout Writer QAE - Special Emphasis Review QAE - Pre-Issue Review	

<b><i>Attribute name</i></b>	<b><i>information</i></b>	<b><i>Type</i></b>	<b><i>Required (user input)</i></b>	<b><i>Editable state</i></b>	<b><i>Validating</i></b>
DRAWING3_REV		String	false	Originating Ripout Writer QAE - Special Emphasis Review QAE - Pre-Issue Review	
DRAWING3_APPL_PGS		String	false	Originating Ripout Writer QAE - Special Emphasis Review QAE - Pre-Issue Review	
DRAWING4		String	false	Originating Ripout Writer QAE - Special Emphasis Review QAE - Pre-Issue Review	
DRAWING4_REV		String	false	Originating Ripout Writer QAE - Special Emphasis Review QAE - Pre-Issue Review	
DRAWING4_APPL_PGS		String	false	Originating Ripout Writer QAE - Special Emphasis Review QAE - Pre-Issue Review	

<b>Attribute name</b>	<b>information</b>	<b>Type</b>	<b>Required (user input)</b>	<b>Editable state</b>	<b>Validating</b>
CNTLS_REQ	i.e. controls are required	Boolean	set true when current state is 'NQCE - Nuclear Interface' set true when current state is 'QAI - Piping Review' set true when current state is 'QAI - Mechanical Review' set true when current state is 'QAI - Structural Review' set true when current state is 'QAI - Electrical Review' set true when current state is 'QAI - Supervisor Review' set true when current state is 'QAE - Pre-Issue Review' set true when current state is 'Test Department - Test Controls' set true when current state is 'Retest Trade'	Originating Ripout Writer QAE - Special Emphasis Review NQCE - Nuclear Interface QAI - Piping Review QAI - Mechanical Review QAI - Structural Review QAI - Electrical Review QAI - Supervisor Review QAE - Pre-Issue Review Test Department - Test Controls Retest Trade	cannot be empty
SSCI_NUMBER		String	set true when current state is 'Ripout Writer'	Ripout Writer	cannot be empty
WORK_SCOPE_PIPING		Boolean	set true when current state is 'Ripout Writer'	Ripout Writer QAE - Special Emphasis Review	
WORK_SCOPE_MECHAN		Boolean	set true when current state is 'Ripout Writer'	Ripout Writer QAE - Special Emphasis Review	
WORK_SCOPE_STRUCT		Boolean	set true when current state is 'Ripout Writer'	Ripout Writer QAE - Special Emphasis Review	
WORK_SCOPE_ELECTRICAL		Boolean	set true when current state is 'Ripout Writer'	Ripout Writer QAE - Special Emphasis Review	
OR	i.e. Operations is Responsible	Boolean	set true when current state is 'QAE – Special Emphasis Review'	Ripout Writer QAE - Special Emphasis Review	

<b>Attribute name</b>	<b>information</b>	<b>Type</b>	<b>Required (user input)</b>	<b>Editable state</b>	<b>Validating</b>
LEVEL_1		Boolean	set true when current state is 'Ripout Writer' and NUCLEAR=True	Ripout Writer QAE - Special Emphasis Review	
SUBSAFE		Boolean	set true when current state is 'Ripout Writer' and NUCLEAR=True	Ripout Writer QAE - Special Emphasis Review	
DSS_SOC		Boolean	set true when current state is 'QAE - Special Emphasis Review'	Ripout Writer QAE - Special Emphasis Review	
SFCC		Boolean	set true when current state is 'QAE - Special Emphasis Review'	Ripout Writer QAE - Special Emphasis Review	
FBW	i.e. Fly By Wire	Boolean	set false when current state is 'QAE - Special Emphasis Review'	Originating Ripout Writer QAE - Special Emphasis Review	
OCT		String	set true when current state is 'Ripout Writer' and NUCLEAR=True set true when current state is 'Lead Trade - Approval'	Originating Ripout Writer Lead Trade - Approval	cannot be blank
WAD		String	set true when current state is 'Lead Trade - Approval' and NUCLEAR=True	Originating Ripout Writer Lead Trade - Approval	cannot be blank
LOCATION		String	set true when current state is 'Ripout Writer' set true when current state is 'Lead Trade - Approval'	Originating Ripout Writer Lead Trade - Approval	valid values (ship, shop)
LEVEL		String	set true when current state is 'Ripout Writer' and LOCATION = "SHIP" set true when current state is 'Lead Trade - Approval' and LOCATION = 'SHIP' and NUCLEAR=True	Originating Ripout Writer Lead Trade - Approval	value to be blank when LOCATION=SHOP value validates to a managed list of values

<b>Attribute name</b>	<b>information</b>	<b>Type</b>	<b>Required (user input)</b>	<b>Editable state</b>	<b>Validating</b>
P_S_C	i.e. p/s/c	String	set true when current state is 'Ripout Writer' and LOCATION = "SHIP" set true when current state is 'Lead Trade - Approval' and LOCATION = "SHIP" and NUCLEAR=True	Originating Ripout Writer Lead Trade - Approval	value to be blank when LOCATION=SHOP value validates to a managed list of values
FRAME		String	set true when current state is 'Ripout Writer' and LOCATION = "SHIP" set true when current state is 'Lead Trade - Approval' and LOCATION = "SHIP" and NUCLEAR=True	Originating Ripout Writer Lead Trade - Approval	value must be blank when LOCATION=SHOP
REASON		String	set true when current state is 'Ripout Writer' set true when current state is 'Lead Trade - Approval' and NUCLEAR=True	Originating Ripout Writer Lead Trade - Approval	
BTR	i.e. Blue Tags are Required	Boolean	set true when current state is 'Ripout Writer' set true when current state is 'QAE - Special Emphasis Review' set true when current state is 'QAI - Piping Review' set true when current state is 'QAI - Mechanical Review' set true when current state is 'QAI - Structural Review' set true when current state is 'QAI - Electrical Review' set true when current state is 'QAI - Supervisor Review' set true when current state is 'QAE - Pre-Issue Review' set true when current state is 'Lead Trade - Approval' and NUCLEAR=True	Originating Ripout Writer QAE - Special Emphasis Review QAI - Piping Review QAI - Mechanical Review QAI - Structural Review QAI - Electrical Review QAI - Supervisor Review QAE - Pre-Issue Review Planning - Blue Tags Lead Trade - Approval	



<b>Attribute name</b>	<b>information</b>	<b>Type</b>	<b>Required (user input)</b>	<b>Editable state</b>	<b>Validating</b>
NDT_REQ		Boolean	set true when current state is 'Ripout Writer' set true when current state is 'QAE - Special Emphasis Review' set true when current state is 'QAI - Piping Review' set true when current state is 'QAI - Mechanical Review' set true when current state is 'QAI - Structural Review' set true when current state is 'QAI - Electrical Review' set true when current state is 'QAI - Supervisor Review' set true when current state is 'QAE - Pre-Issue Review' set true when current state is 'Lead Trade - Approval' and NUCLEAR=True	Originating Ripout Writer QAE - Special Emphasis Review QAI - Piping Review QAI - Mechanical Review QAI - Structural Review QAI - Electrical Review QAI - Supervisor Review QAE - Pre-Issue Review Lead Trade - Approval	
VWP	i.e. Is Vendor Work to be performed?	Boolean	set true when current state is 'Ripout Writer' set true when current state is 'Lead Trade - Approval' and NUCLEAR=True	Originating Ripout Writer Lead Trade - Approval	
TEST_TRADE	Default Value "275"	String	set true when current state is 'Ripout Writer' set true when current state is 'Test Department - Test Controls' set true when current state is 'Lead Trade - Approval' and NUCLEAR=True	Originating Ripout Writer Test Department - Test Controls Lead Trade - Approval	value validates to NNS EB Lead Trade syntax) and any other constraints to be provided by EB
RETEST_REQ		Boolean	set true when current state = 'Test Department - Test Controls'	Test Department - Test Controls	
RETEST_REQUIREMENTS		String	set true when current state = 'Retest Trade'	Retest Trade Nuclear Test – Retest Required 1 Nuclear Test – Retest Required 2	

<b>Attribute name</b>	<b>information</b>	<b>Type</b>	<b>Required (user input)</b>	<b>Editable state</b>	<b>Validating</b>
RETEST_TRADE_1		String	set true when current state='Retest Trade' set true when state='Test Department-Test Controls' and Retest_Req=true	Test Department - Test Controls Retest Trade	value validates to NNS (EB Lead Trade syntax) and any rule provided by EB
RETEST_TRADE_2		String	false	Test Department - Test Controls Retest Trade	value validates to NNS (EB Lead Trade syntax) and any rule provided by EB
RETEST_TRADE_3		String	false	Test Department - Test Controls Retest Trade	value validates to NNS (EB Lead Trade syntax) and any rule provided by EB
TANK_CLSR_REQ	i.e. Tank Closure is Required	Boolean	Set true when current state is 'Ripout Writer' set true when current state is 'Lead Trade - Approval' and NUCLEAR=True	Originating Ripout Writer Lead Trade - Approval	
KEY_EVENT		String	Set true when current state is 'Nuclear Ship's Management - Approval'	NOTE: editing disallowed on any current state where 7Gstatus is 'DRFT' or 'CLSD'  Originating Nuclear Ship's Management - Approval Any state where user has Nuclear Ship's Management role and Current State 7Gstatus is not 'DRFT' or 'CLSD'	MDD requests Validation, EB to provide this rule
REF_DRAWING1		String	false	Originating Ripout Writer QAE - Special Emphasis Review Lead Trade - Approval	

<b><i>Attribute name</i></b>	<b><i>information</i></b>	<b><i>Type</i></b>	<b><i>Required (user input)</i></b>	<b><i>Editable state</i></b>	<b><i>Validating</i></b>
REF_DRAWING1_REV		String	false	Originating Ripout Writer QAE - Special Emphasis Review Lead Trade - Approval	
REF_DRAWING1_APPL_PGS		String	false	Originating Ripout Writer QAE - Special Emphasis Review Lead Trade - Approval	
REF_DRAWING2		String	false	Originating Ripout Writer QAE - Special Emphasis Review Lead Trade - Approval	
REF_DRAWING2_REV		String	false	Originating Ripout Writer QAE - Special Emphasis Review Lead Trade - Approval	
REF_DRAWING2_APP_PGS		String	false	Originating Ripout Writer QAE - Special Emphasis Review Lead Trade - Approval	
REF_DRAWING3		String	false	Originating Ripout Writer QAE - Special Emphasis Review Lead Trade - Approval	
REF_DRAWING3_REV		String	false	Originating Ripout Writer QAE - Special Emphasis Review Lead Trade - Approval	

<b>Attribute name</b>	<b>information</b>	<b>Type</b>	<b>Required (user input)</b>	<b>Editable state</b>	<b>Validating</b>
REF_DRAWING3_APP_PGS		String	false	Originating Ripout Writer QAE - Special Emphasis Review Lead Trade - Approval	
REF_DRAWING4		String	false	Originating Ripout Writer QAE - Special Emphasis Review Lead Trade - Approval	
REF_DRAWING4_REV		String	false	Originating Ripout Writer QAE - Special Emphasis Review Lead Trade - Approval	
REF_DRAWING4_APP_PGS		String	false	Originating Ripout Writer QAE - Special Emphasis Review Lead Trade - Approval	
INSPECTING_TRADE		String	set true when current state is 'Ripout Writer' and Nuclear = true set true when current state is 'Lead Trade - Approval' and NUCLEAR=True set true when current state is 'QAE - Special Emphasis Review'	Ripout Writer QAE - Special Emphasis Review Lead Trade - Approval	value validates to NNS EB Lead Trade syntax) and any rule provided EB
TAG_REC_SHEET_REQ		Boolean	set true when current state is 'QAE - Pre-Issue Review'	Ripout Writer QAE - Special Emphasis Review QAE - Pre-Issue Review	
NUCLEAR_INTERFACE		Boolean	set true when current state is 'QAE - Special Emphasis Review' set true when current state is 'NQCE - Nuclear Interface'	QAE - Special Emphasis Review NQCE - Nuclear Interface	

<b>Attribute name</b>	<b>information</b>	<b>Type</b>	<b>Required (user input)</b>	<b>Editable state</b>	<b>Validating</b>
REC_SEQ_NO_REQ	i.e. Recorded Sequence Number Required?	Boolean	set true when current state is 'QAE - Special Emphasis Review'	Ripout Writer QAE - Special Emphasis Review Lead Trade - Approval	
RECD_SEQ_NO	i.e. Recorded Sequence Number	String	Set true when current state is 'QAE – Reentry Control Document Number'	QAE – Reentry Control Document Number	
QAL_CERT_REQ	i.e. QAL Certification Required	Boolean	set true when current state is 'QAI - Piping Review' set true when current state is 'QAI - Mechanical Review' set true when current state is 'QAI - Structural Review' set true when current state is 'QAI - Electrical Review' set true when current state is 'QAI - Supervisor Review'	QAI - Piping Review QAI - Mechanical Review QAI - Structural Review QAI - Electrical Review QAI - Supervisor Review	
QAL_CERTIFICATION		String	set true when current state is 'QAE - Pre-Issue Review'	QAE – Special Emphasis Review QAE - Pre-Issue Review	
TAGOUT_REQ		Boolean	set true when current state is 'Test Department - Test Controls' set true when current state is 'Nuclear Test - Controls Required'	Test Department - Test Controls Nuclear Test - Controls Required	
HAZ_ENERGY_TO_REQ	i.e. Hazardous Energy Tagout Required	Boolean	set true when current state is 'Test Department - Test Controls'	Test Department - Test Controls	
WHO_HAS_OPCON		String	set true when current state is 'Test Department - Test Controls' set true when current state is 'Retest Trade' set true when current state is 'Nuclear Test - Controls Required'	Test Department - Test Controls Retest Trade Nuclear Test - Controls Required	apply list of valid entries (ShipForce, EB)
WIP_REQ		Boolean	set true when current state is 'Test Department - Test Controls' set true when current state is 'Nuclear Test - Controls Required'	Test Department - Test Controls Nuclear Test - Controls Required	To validate on EJB - can proceed with update when new value is not equal to SEAWATER_CON_SYS

<b>Attribute name</b>	<b>information</b>	<b>Type</b>	<b>Required (user input)</b>	<b>Editable state</b>	<b>Validating</b>
WIP_NUMBER		String	set true when current state is 'Ship Safety Officer - WIP'	Ship Safety Officer - WIP Nuclear Test - WIP	EB to provide
RADCON_REQ	i.e. Radiological Controls Required	Boolean	set true when current state is 'Test Department - Test Controls'	Test Department - Test Controls	
SAFETY_TECH_MECH	i.e. Safety Tech Mechanical	Boolean	set true when current state is 'Test Department - Test Controls' set true when current state is 'Nuclear Test - Controls Required'	Test Department - Test Controls Nuclear Test - Controls Required	
SAFETY_TECH_ELEC	i.e. Safety Tech Electrical	Boolean	set true when current state is 'Test Department - Test Controls' set true when current state is 'Nuclear Test - Controls Required'	Test Department - Test Controls Nuclear Test - Controls Required	
SAFETY_TECH_ETRONIC	i.e. Safety Tech Electronics	Boolean	set true when current state is 'Test Department - Test Controls'	Test Department - Test Controls	
SEAWATER_CON_SYS	i.e. Sea Water Connected System	Boolean	set true when current state is 'Test Department - Test Controls'	Test Department - Test Controls	To validate on EJB - can proceed with update when new value is not equal to WIP_REQ
EMBED_TEST_REQ		Boolean	false	Test Engineering - Embedded Testing	
XREF	derived / assigned by Oracle on insert	Long	na	none	
XREF_SEQ	derived / assigned by Oracle on insert incremented internally on restart of ripout	Short	na	none	

<i><b>Attribute name</b></i>	<i><b>information</b></i>	<i><b>Type</b></i>	<i><b>Required (user input)</b></i>	<i><b>Editable state</b></i>	<i><b>Validating</b></i>
RIPOUT_LIST_NUMBER	<p>derived internally on EJB with formula</p> <p>Syntax: "RO" + SSCI_NUMBER + CONSTRUCTION_CODE + LIST_SEQ</p> <p>Notes CONSTRUCTION_CODE is blank or 'B', LIST_SEQ is 001 and can increment.</p> <p>Services</p>	String	na	none	

## ***Oracle Tables***

### **Ripout\_Comment**

- COMMENT\_RO\_XREF
- COMMENT\_RO\_XREF\_SEQ
- COMMENT\_COMMENT\_SEQ
- COMMENT\_TEXT
- COMMENT\_USER\_ID
- COMMENT\_BADGE
- COMMENT\_DATE\_TIME (apply as EST/EDT zone)

### **Ripout\_Signature**

- SIGNOFF\_XREF
- SIGNOFF\_XREF\_SEQ
- SIGNOFF\_STATE\_ID
- SIGNOFF\_STATUS (eligible value = S, P, K)
- SIGNOFF\_USER\_ID
- SIGNOFF\_BADGE
- SIGNOFF\_USER\_DEPT
- SIGNOFF\_DATE

### **Ripout\_Routings**

- ROUTING\_XREF\_ID
- ROUTING\_XREF
- ROUTING\_STATE\_ID
- ROUTING\_STATE\_NAME
- ROUTING\_NEXT\_NAME
- ROUTING\_SIGNATURE

### **RedMat**

- RIPOUT\_ID
- RIPOUT\_TARGET\_ATTRIBUTE -- name of the ripout table column for proposed change
- FK\_SIG\_LEAD\_TRADE – foreign key of record in signature table
- FK\_SIG\_TEST\_TRADE
- FK\_SIG\_INSPECTING\_TRADE
- FK\_SIG\_D461
- FK\_SIG\_RETEST\_TRADE
- FK\_SIG\_SHIPS\_FORCE
- FK\_SIG\_SAFETY\_TECH\_MECH
- FK\_SIG\_SAFETY\_TECH\_ELECTRICAL



## Appendex OR - Oracle Tables

- FK\_SIG\_SAFETY\_TECH\_ELECTRONICS
- APPROVED – BOOLEAN

### **Ripout\_List**

- Ripout\_List\_Number

### **Item**

EB to clarify

### **Trade\_Boundary**

Table listing trade boundaries for selection by client application.

? is this a defined list of boundaries that can be read from a lookup table?

### **Controls Required**

EB to clarify

### **Retest Requirements**

EB to clarify

### ***Database Connection***

DB connection and pool to be defined by DBA. Ripout DAO will use connection logic similar to:

```
Context context = new InitialContext();  
DataSource ds = (DataSource) context.lookup("ripoutPool");  
conn = ds.getConnection();
```

### ***Project Documentation***

Ripout Project Path: [\\us-ct-](#)

[eb01\ebdepts\ENG.604.IPDE\IPDE Program\IPDE Business\Task Folders](#)

\Work\_Control\_Process-Flow\_Folder\Ripouts

(Ripout project path)\System Design\Ripouts Detail Requirements V5 3-6-19.docx

(Ripout project path)\ Ripouts Attachment 7 Rev A 12-17-18- Ripouts Routings.xlsx

(Ripout project path)\ System Design\Ripout Routings and Users Access\MDD

\Ripouts Attachment 3 – Use Cases.xlsx.

(Ripout project path)\MDD\Work Control Process Flow Ripouts MDD 12.10.18 - team meeting comments.pdf.

## Appendix 2 – Customer Comments

Status – The server is still required to validate input.

It is not clear what “[no] support of other AJAX capability” implies. If there are GETs and POSTs and such, we can call it with AJAX. That’s a client-side thing.

Response: removed text about AJAX.

XREF and Sequence – It appears RIPOUT XREFS are different from Work Order/ERS XREFs. If these are different, should we really be calling the Ripout ID an XREF?

Response: ID to be defined with DBA help.

Ripout from Scratch – Why are there 4 discreet drawing fields? Why not a list of associated drawings? What if there are 5 applicable drawings? We can easily handle a general list of drawings/references.

Response: Ripout mockup shows 4 discreet drawing fields. Of course the API can return a dynamically generated list of drawings/references for the client to parse and display.

Ripout from Seed – We can do some validation / edit restriction client side. The server should confirm the restriction of an identical nuclear choice as the seed record, if that is a requirement. General rule for REST, never trust the client.

Response: add text regarding server validation of constant nuclear choice.

Not sure about page 20

Enumerations – “it’s the client’s responsibility to expand abbreviations for UI display” How would the client know what to display for a potentially new status (without a CR)? It would be better if the expanded abbreviations were also supplied or available.

Response: Modified that section.

Resource Discovery – There may be a (debatably better) design in which you wouldn’t go to different endpoints, say for different routing steps. (See, the ERS services).

Response: under advisement, will review ers services

Reqs 1 & 2 – Seems like this should be one service and we would identify if the create is from seed.

Response: Specifying a ripout number, **POST /ripout/{ripoutNumber}** seems sufficient to alert service to create from seed record identified by ripoutNumber.

Reqs 4 & 5 – Cancel vs Delete still open in my mind. Current design looks like Cancel puts the record into perpetual limbo. (Just deletes the workflow with no status change) Delete looks to be an “Erase from history” and it doesn’t look sufficiently defined who can have such power. Not sure that is the right thing to do. Looks dangerous the way it is written.

Response: need more study

## Appendix OR - Oracle Tables

Req 8, 9 – Need to add STATUS as a search parameter. Performing a multi-field search in a GET can occasionally be troublesome depending on how long the fields can get. POST might be safer.

update.)

Response: added STATUS param

Req. 18 –Why this there only non-nuc routing info. What about nuc?

Response: corrected omission, modified:

**GET /ripout/{ripoutNumber}/routings&nuclear=true|false**

DB – Should things like Drawings1 – 4 and Retest Trade 1 – 3 be associated tables.

Response: need information from DBA

## Footnotes

- 1 Ripout Project Path: [\\us-ct-eb01\ebdepts\ENG.604.IPDE\IPDE\\_Program\IPDE\\_Business\Task\\_Folders\Work\\_Control\\_Process-Flow\\_Folder\Ripouts](#)
- 2 Project documentation: (Ripout project path)\System Design\Ripout Screen Mock Ups\**Mock Ripouts Detail Screen.xlsx**
- 3 Oracle documentation: <https://docs.oracle.com/middleware/12213/wls/RESTF/intro-restful-service.htm>
- 4 WebLogic Server: <https://docs.oracle.com/middleware/12213/wls/RESTF>
- 5 Roy Fielding: <https://roy.gbiv.com/untangled/2008/res-apis-must-be-hypertext-driven>
- 6 Google Cloud: [https://cloud.google.com/apis/design/custom\\_methods](https://cloud.google.com/apis/design/custom_methods)