Oracle® Manufacturing

Implementing Oracle E-Records in Discrete Manufacturing Guide

Release 11*i*

Part No. B16387-01

August 2005

ORACLE

Oracle Manufacturing Implementing Oracle E-Records in Discrete Manufacturing Guide, Release 11*i*

Part No. B16387-01

Primary Author: Laurel Dale

Contributing Author: Aravindan Sankaramurthy

Contents

B Device History Record Example

C Oracle E-Records Event Data for Discrete Manufacturing

Index

Send Us Your Comments

Oracle Manufacturing Implementing Oracle E-Records in Discrete Manufacturing Guide, Release 11*i*
Part No. B16387-01

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, please indicate the title and part number of the documentation and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: appsdoc_us@oracle.com
- FAX: 650-506-7200 Attn: Oracle Supply Chain Management Documentation Manager
- Postal service:
  Oracle Supply Chain Management Documentation Manager
  Oracle Corporation
  500 Oracle Parkway
  Redwood Shores, CA 94065
  USA

If you would like a reply, please give your name, address, telephone number, and electronic mail address (optional).

If you have problems with the software, please contact your local Oracle Support Services.

Preface

Intended Audience

Welcome to Release 11*i* of the *Oracle Manufacturing Implementing Oracle E-Records in Discrete Manufacturing Guide*.

This guide assumes you have read the *Oracle E-Records Implementation Guide* and are familiar with Oracle Discrete Manufacturing Applications.

See Related Documents on page x for more Oracle Applications product information.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at http://www.oracle.com/accessibility/ .

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Structure

1 Introduction

This guide supplements the *Oracle E-Records Implementation Guide*, providing Oracle E-Records implementation information that is specific to the Oracle Discrete Manufacturing Applications.

This chapter discusses using Oracle E-Records with discrete manufacturing business processes to support compliance of Food and Drug Administration (FDA) regulations, specifically the FDA Code of Federal Regulations (CFR) Title 21 Part 11, otherwise known as 21 CFR Part 11.

2 Oracle E-Records Setup and Process Exceptions in Discrete Manufacturing

This chapter provides an example of how to set up a transaction to capture e-records and e-signatures. It also discusses exceptions to the standard setup steps and setup process.

3 Implementation Considerations

This chapter discusses strategies for you to consider using during Oracle E-Records implementation.

A Windows and Navigation Paths

This appendix lists each window referred to in the *Oracle Manufacturing: Implementing Oracle E-Records in Discrete Manufacturing Guide* as well as the associated navigator path for each window by responsibility. It also lists the navigation path(s) for every Oracle E-Records enabled discrete manufacturing event.

B Device History Record Example

This appendix shows collated and printed e-records created for the Device History Record Example, page 3-26.

C Oracle E-Records Event Data for Discrete Manufacturing

The tables in this appendix detail the seed data entered to support each discrete manufacturing application integrated with Oracle E-Records.

Related Documents

Oracle E-Records Implementation Guide

This guide explains the purpose of Oracle E-Records as well as how to implement it. Use Oracle E-Records when you have organizations that want to maintain electronic documents or need the ability to electronically sign those documents, ensuring that the appropriate personnel have reviewed and approved them.

Implementing Oracle Approvals Management

This guide explains how to define business rules governing the process for approving transactions in Oracle Applications where Oracle Approvals Management has been integrated.

Oracle Engineering User's Guide

This guide enables your engineers to utilize the features of Oracle Engineering to quickly introduce and manage new designs into production. Specifically, this guide details how to quickly and accurately define the resources, materials and processes necessary to implement changes in product design.

Oracle Inventory User's Guide

This guide describes how to define items and item information, perform receiving and inventory transactions, maintain cost control, plan items, perform cycle counting and physical inventories, and set up Oracle Inventory.

Oracle Bills of Material User's Guide

This guide describes how to create various bills of materials to maximize efficiency, improve quality and lower cost for the most sophisticated manufacturing environments. By detailing integrated product structures and processes, flexible product and process definition, and configuration management, this guide enables you to manage product details within and across multiple manufacturing sites.

Oracle Work In Process User's Guide

This guide describes how Oracle Work in Process provides a complete production management system. Specifically, this guide describes how discrete, repetitive, assemble-to-order, project, flow, and mixed manufacturing environments are supported.

Oracle Quality User's Guide

This guide describes how Oracle Quality can be used to meet your quality data collection and analysis needs. This guide also explains how Oracle Quality interfaces with other Oracle Manufacturing applications to provide a closed loop quality control system.

Oracle Quality Implementation Guide

This guide explains how to implement Oracle Quality's nonconformance, disposition, and corrective action solution based on seeded templates.

Oracle Shipping Execution User's Guide

This guide describes how to set up Oracle Shipping Execution to process and plan your trips, stops and deliveries, ship confirmation, query shipments, determine freight cost and charges to meet your business needs.

Oracle Purchasing User's Guide

This guide describes how to create and approve purchasing documents, including requisitions, different types of purchase orders, quotations, RFQs, and receipts. This guide also describes how to manage your supply base through agreements, sourcing rules and approved supplier lists. In addition, this guide explains how you can automatically create purchasing documents based on business rules through integration with Oracle Workflow technology, which automates many of the key procurement processes.

Oracle Applications Flexfields Guide

Order Management implementation team, as well as for users responsible for the ongoing maintenance of Oracle Applications product data. This guide also provides information on creating custom reports on flexfields data.

Do Not Use Database Tools to Modify Oracle Applications Data

Oracle STRONGLY RECOMMENDS that you never use SQL*Plus, Oracle Data Browser, database triggers, or any other tool to modify Oracle Applications data unless otherwise instructed.

Oracle provides powerful tools you can use to create, store, change, retrieve, and maintain information in an Oracle database. But if you use Oracle tools such as SQL*Plus to modify Oracle Applications data, you risk destroying the integrity of your data and you lose the ability to audit changes to your data.

Because Oracle Applications tables are interrelated, any change you make using an Oracle Applications form can update many tables at once. But when you modify Oracle Applications data using anything other than Oracle Applications, you may change a row in one table without making corresponding changes in related tables. If your tables get out of synchronization with each other, you risk retrieving erroneous information and you risk unpredictable results throughout Oracle Applications.

When you use Oracle Applications to modify your data, Oracle Applications automatically checks that your changes are valid. Oracle Applications also keeps track of who changes information. If you enter information into database tables using database tools, you may store invalid information. You also lose the ability to track who has changed your information because SQL*Plus and other database tools do not keep a record of changes.

1

Introduction

This guide supplements the *Oracle E-Records Implementation Guide*, providing Oracle E-Records implementation information that is specific to the Oracle Discrete Manufacturing Applications.

This chapter discusses using Oracle E-Records with discrete manufacturing business processes to support compliance of Food and Drug Administration (FDA) regulations, specifically the FDA Code of Federal Regulations (CFR) Title 21 Part 11, otherwise known as 21 CFR Part 11.

This chapter covers the following topics:

- Using Oracle E-Records to Support Compliance of 21 CFR Part 11
- Discrete Manufacturing Business Flows Using Oracle E-Records

Using Oracle E-Records to Support Compliance of 21 CFR Part 11

The Food and Drug Administration (FDA) protects the public health by regulating the food and drug industries. In 1997, the FDA issued new regulations for quality systems using computerized software in the FDA Code of Federal Regulations (CFR) Title 21 Part 11, otherwise known as 21 CFR Part 11. The regulations strive to assure that computerized records are safe, secure, and as accurate as a paper based system.

According to 21 CFR Part 11, the software application's role in the data management process is to guarantee and substantiate that the manufactured product data is electronically captured, manipulated, extracted, and coded during the manufacturing of the product. All subject data and definitional objects (metadata) must have a complete audit trail.

In general, 21 CFR Part 11 describes the requirements that must be met when using electronic records (e-records) and electronic signatures (e-signatures), but does not describe where they are required. The FDA left the definition of where to use electronic records and which signatures to apply as electronic signatures to the discretion of the medical device manufacturers who use software applications as part of their quality management system. The predicate rule used by the medical device manufacturers is FDA 21 CFR Part 820. Oracle Discrete Manufacturing Applications enable electronic records and signatures for certain business events (see: Oracle E-Records Enabled Transactions Summary, page 1-14), as well as give users ways to choose which business events must meet 21 CFR Part 11 requirements, by using the Oracle E-Records product. Refer to the *Oracle E-Records Implementation Guide* for information on how to tailor Oracle E-Records to meet your needs.

Depending on the type of business event, an electronic signature is either online or deferred. An online signature requires that you enter a valid signature before saving the event or transaction. A deferred signature enables you to save the event in a Pending Approval state before obtaining a final signature. Many times a deferred signature requires a workflow approval, so the event is saved before full approval is received back from the workflow process.

Certain business events also include any attachment made to the business event object in the e-record. For example, if a drawing depicting an engineering change is attached to an engineering change order, you can review the drawing in the e-record for the engineering change order.

The e-records and e-signatures (ERES) framework is a central tool designed to achieve 21 CFR Part 11 compliance for the necessary Oracle Applications business events. The ERES framework uses common Oracle Applications components such as Oracle Workflow business events, the XML Gateway, and others.

In order to enable electronic records and signatures for Oracle Discrete Manufacturing Applications, you must follow the steps listed in Implementing E-Records, *Oracle E-Records Implementation Guide*. Otherwise, no e-records or e-signatures are required. The discrete manufacturing applications that include transactions enabled (once you perform the implementation steps) to use electronic records and signatures are:

- Oracle Engineering
- Oracle Inventory
- Oracle Bills of Material
- Oracle Work in Process
- Oracle Quality
- Oracle Shipping
- Oracle Purchasing

Discrete Manufacturing Business Flows Using Oracle E-Records

The Oracle Discrete Manufacturing Applications use Oracle E-Records in the following business flows:

- Design Transfer
- Design Change
- Demand to Build
- Nonconformance Management
- Corrective Actions Processing
- Procure to Pay

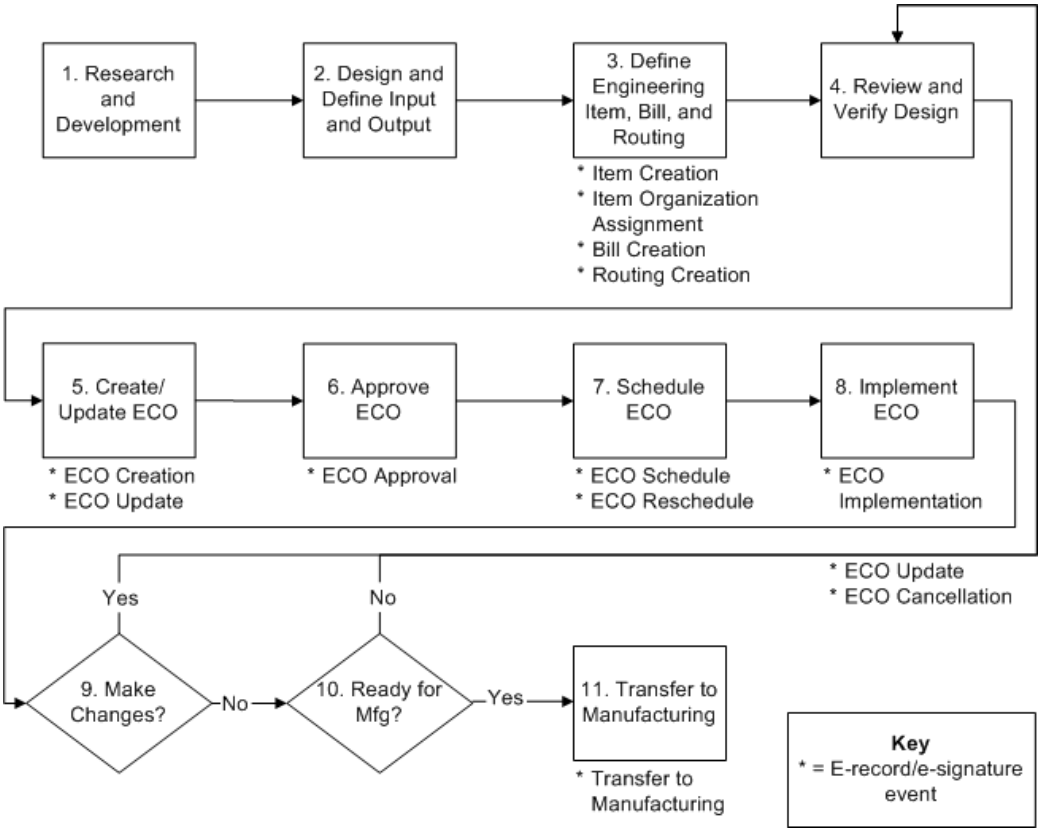
The following sections describe each business flow and the discrete manufacturing transactions enabled in Oracle E-Records within each flow.

Related Topics

Navigation Paths For All Oracle E-Records-Enabled Discrete Manufacturing Business Events, page A-3

Design Transfer

The following diagram illustrates an example business flow of a new or changed item as it moves from the design phase to manufacturing. Each numbered step is explained in the table following the diagram.



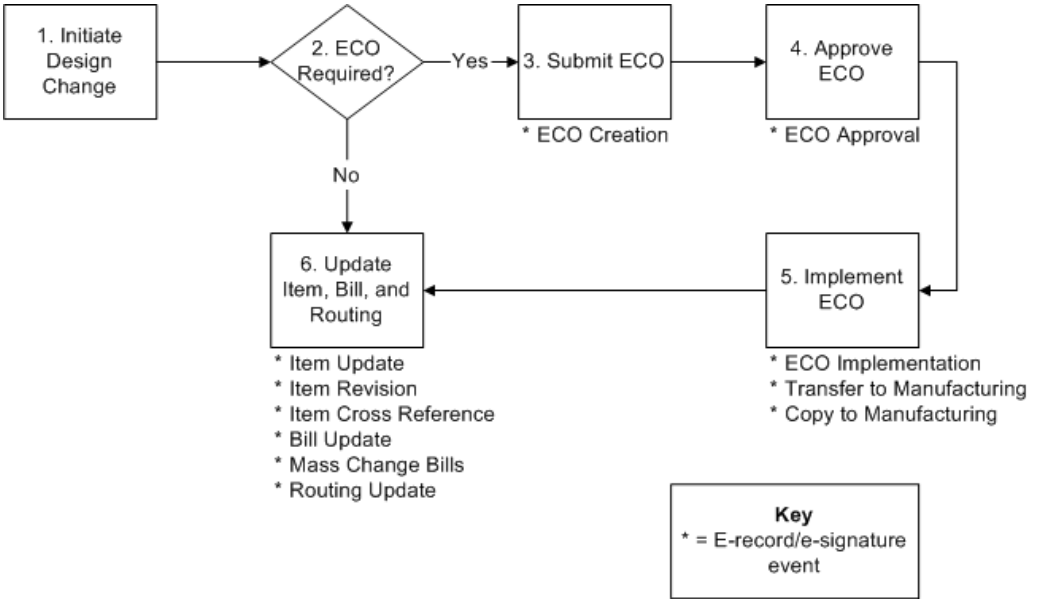
Design Transfer Business Flow Steps

Step	Name	Description	E-Record Enabled Events
1	Research and Development	Develop a new prototype item. Occurs outside of Oracle Applications (OA).	No transactions
2	Design and Define Input and Output	Finalize the prototype item specifications, components, and bill. Occurs outside of OA.	No transactions
3	Define Engineering Item, Bill, and Routing	Create an engineering item, bill, and routing. Assign the item to an organization. Occurs within OA.	Item Creation Item Organization Assignment and the child event Item Creation (for the organization): Bill Creation Routing Creation

Step	Name	Description	E-Record Enabled Events
4	Review and Verify Design	Review and verify the design of the item. Occurs outside of OA.	No transactions
5	Create/Update ECO	If changes to the engineering item, bill, or routing are needed, then create or update an engineering change order. Occurs within OA.	ECO Creation ECO Update
6	Approve ECO	Before an ECO is implemented, the ECO must be approved by the list of approvers. Occurs within OA.	ECO Approval
7	Schedule ECO	Engineering updates the date to implement the ECO. Occurs within OA.	ECO Schedule ECO Reschedule
8	Implement ECO	Make the changes specified in the ECO. Occurs within OA.	ECO Implementation and child events: • Transfer to Manufacturing • Copy to Manufacturing • Bill Creation • Bill Update • Routing Creation • Routing Update
9	Make Changes?	Are any ECO changes necessary before transferring the item, bill, or routing to manufacturing? Yes: Review and Verify Design No: Ready for Manufacturing?	Yes: ECO Update ECO Cancellation No: No transactions
10	Ready for Mfg?	Is the item, bill, or routing ready for transferring to manufacturing? Yes: Transfer the item, bill, or routing to manufacturing. No: Review and verify design.	Yes: No transactions No: ECO Update ECO Cancellation
11	Transfer to Manufacturing	Change from an engineering to a production item, bill, or routing. Occurs within OA.	Transfer to Manufacturing

Design Change

The following diagram illustrates an example business flow of a changed item, bill, or routing as it goes through the engineering change order (ECO) process to manufacturing. Each numbered step is explained in the table following the diagram.

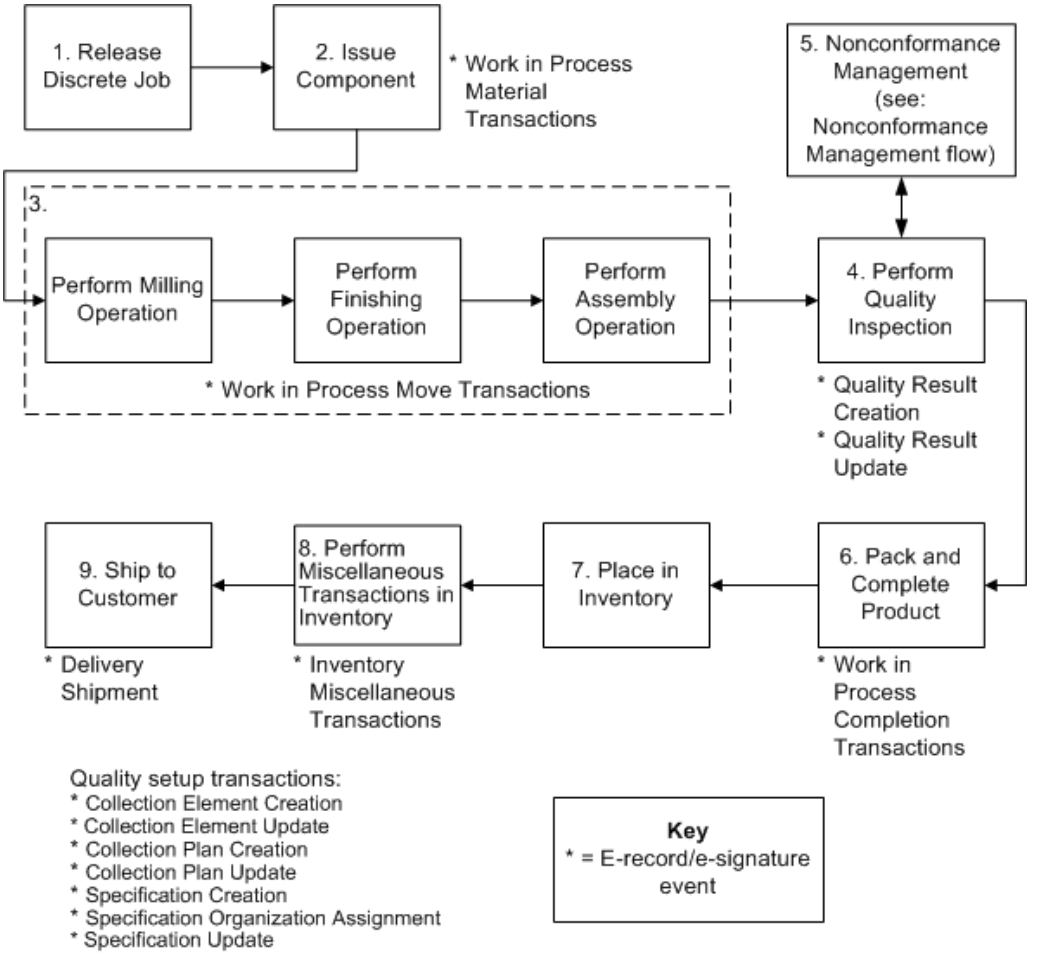

1. Initiate Design Change
2. ECO Required?
Yes
3. Submit ECO
* ECO Creation
4. Approve ECO
* ECO Approval
No
6. Update Item, Bill, and Routing
* Item Update
* Item Revision
* Item Cross Reference
* Bill Update
* Mass Change Bills
* Routing Update
5. Implement ECO
* ECO Implementation
* Transfer to Manufacturing
* Copy to Manufacturing
Key
* = E-record/e-signature event

Design Change Business Flow Steps

Step	Name	Description	E-Record Enabled Events
1	Initiate Design Change	Propose changes to an item, bill, or routing. Occurs outside of Oracle Applications (OA).	No transactions
2	ECO Required?	Is an ECO necessary? Occurs outside of OA. Yes: Submit ECO. No: Update Item, Bill, and Routing.	No transactions
3	Submit ECO	Create an ECO. Occurs within OA.	ECO Creation
4	Approve ECO	Before an ECO is executed, the ECO must be approved by the list of approvers. Occurs within OA.	ECO Approval
5	Implement ECO	Marks each revised item's status as 'implemented'. Occurs within OA.	ECO Implementation
6	Update Item, Bill, and Routing	Update the manufacturing item, bill, and routing with the proposed engineering changes. Occurs within OA.	Transfer to Manufacturing Copy to Manufacturing and child events: • Item Creation • Bill Creation • Routing Creation Item Update Item Revision Item Cross Reference Bill Update Mass Change Bills and the child event ECO Creation Routing Update

Demand to Build

The following diagram illustrates an example business flow of a discrete job released to build a product. As the product is built, it moves through a series of manufacturing operations (milling, finishing, assembly) and quality inspections before it is packed, placed in inventory, then shipped to customers. Each numbered step is explained in the table following the diagram.


1. Release Discrete Job
2. Issue Component
* Work in Process Material Transactions
5. Nonconformance Management (see: Nonconformance Management flow)
3.
Perform Milling Operation
Perform Finishing Operation
Perform Assembly Operation
* Work in Process Move Transactions
4. Perform Quality Inspection
* Quality Result Creation
* Quality Result Update
9. Ship to Customer
* Delivery Shipment
8. Perform Miscellaneous Transactions in Inventory
* Inventory Miscellaneous Transactions
7. Place in Inventory
6. Pack and Complete Product
* Work in Process Completion Transactions
Quality setup transactions:
* Collection Element Creation
* Collection Element Update
* Collection Plan Creation
* Collection Plan Update
* Specification Creation
* Specification Organization Assignment
* Specification Update
Key
* = E-record/e-signature event

Demand to Build Business Flow Steps

Step	Name	Description	E-Record Enabled Events
1	Release Discrete Job	Release a discrete job from planning. Occurs within Oracle Applications (OA).	No transactions
2	Issue Component	Issue a component from inventory. Occurs outside of OA.	Work in Process Material Transactions
3	Perform Milling Operation Perform Finishing Operation Perform Assembly Operation	Move materials between manufacturing operations. Occurs within OA.	Work in Process Move Transactions (If quality results are collected as part of this event, then the e-record includes the quality information.)
4	Perform Quality Inspection	Inspect manufactured materials for quality defects. If quality defects are present, go to the Nonconformance Management flow, page 1-9. If no quality defects exist, go to Pack and Complete Product. Occurs within OA.	Quality Result Creation Quality Result Update
5	Nonconformance Management	Manage nonconforming material. Occurs within OA.	See the Nonconformance Management flow, page 1-9, for a list of transactions.
6	Pack and Complete Product	Complete an assembly. Occurs within OA.	Work in Process Completion Transactions (If quality results are collected as part of this event, then the e-record includes the quality information.)
7	Place in Inventory	Move the assembly to inventory. Occurs within OA.	No transactions
8	Perform Miscellaneous Transactions in Inventory	Move the finished product to inventory. Occurs within OA.	Inventory Miscellaneous Transactions
9	Ship to Customer	Ship the finished product to the customer.	Delivery Shipment

Note: If you choose to collect e-signatures for the following events, you can only collect them in online mode (not deferred).

- Work in Process Material Transactions
- Work in Process Move Transactions
- Work in Process Completion Transactions
- Inventory Miscellaneous Transactions

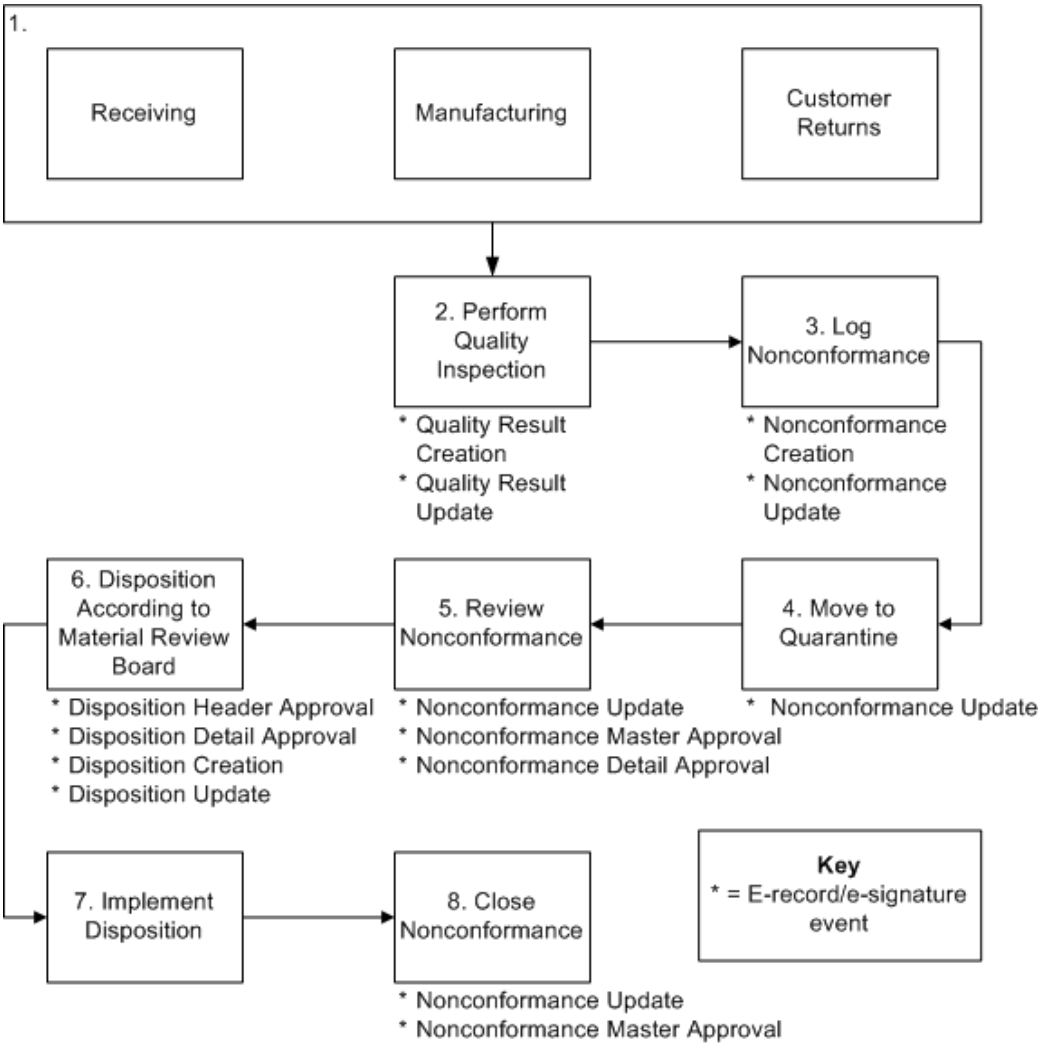
To perform the Demand to Build process, you must also perform the following Oracle Quality setup transactions. You can choose to enable these transactions for use with Oracle E-Records.

- Collection Element Creation

- Collection Element Update
- Collection Plan Creation
- Collection Plan Update
- Specification Creation
- Specification Organization Assignment
- Specification Update

Nonconformance Management

The following diagram illustrates an example business flow for managing nonconforming material. Each numbered step is explained in the table following the diagram.



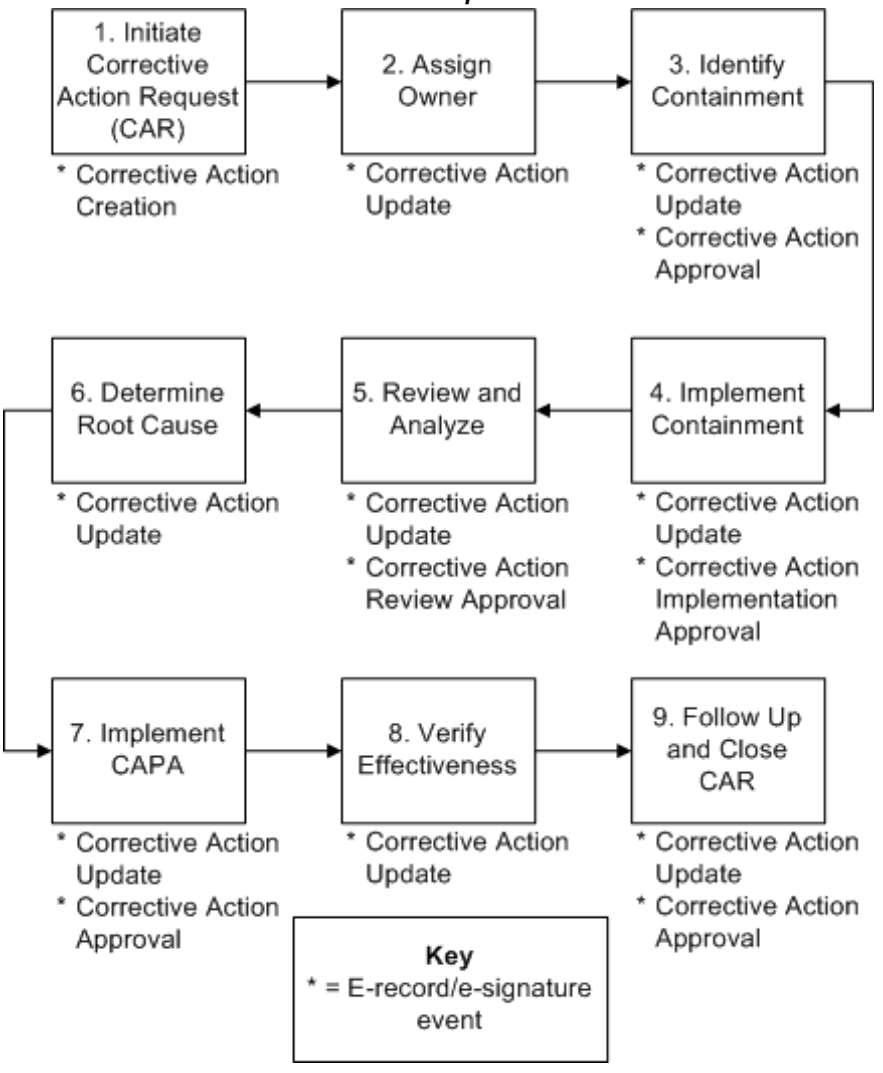
Nonconformance Management Business Flow Steps

Step	Name	Description	E-Record Enabled Events
1	Receiving Manufacturing Customer Returns	Receive supplies, manufacture items or assemblies, and accept customer sales order returns. Occurs within Oracle Applications (OA).	No transactions
2	Perform Quality Inspection	Perform a quality inspection on supplies received, items or assemblies manufactured, and customer returns. Occurs within OA.	Quality Result Creation Quality Result Update
3	Log Nonconformance	Create a nonconformance. Occurs within OA.	Nonconformance Creation Nonconformance Update
4	Move to Quarantine	Move the item or assembly to a quarantined inventory area. Occurs within OA.	Nonconformance Update
5	Review Nonconformance	Enter additional details about the nonconformance. Occurs within OA.	Nonconformance Update Nonconformance Master Approval Nonconformance Detail Approval
6	Disposition According to Material Review Board	Disposition nonconforming material according to the Material Review Board's instructions (rework, return to vendor, scrap, etc.). Occurs within OA.	Disposition Header Approval Disposition Detail Approval Disposition Creation Disposition Update
7	Implement Disposition	Confirm that the disposition occurred. Occurs within OA.	No transactions
8	Close Nonconformance	Close the nonconformance. Occurs within OA.	Nonconformance Update Nonconformance Master Approval

Corrective Actions Processing

The following diagram illustrates an example business flow of implementing actions designed to improve quality. Each numbered step is explained in the table following the diagram.

Corrective Actions Business Flow Example



Note: The following table explains each step in detail.

Corrective Actions Business Flow Steps

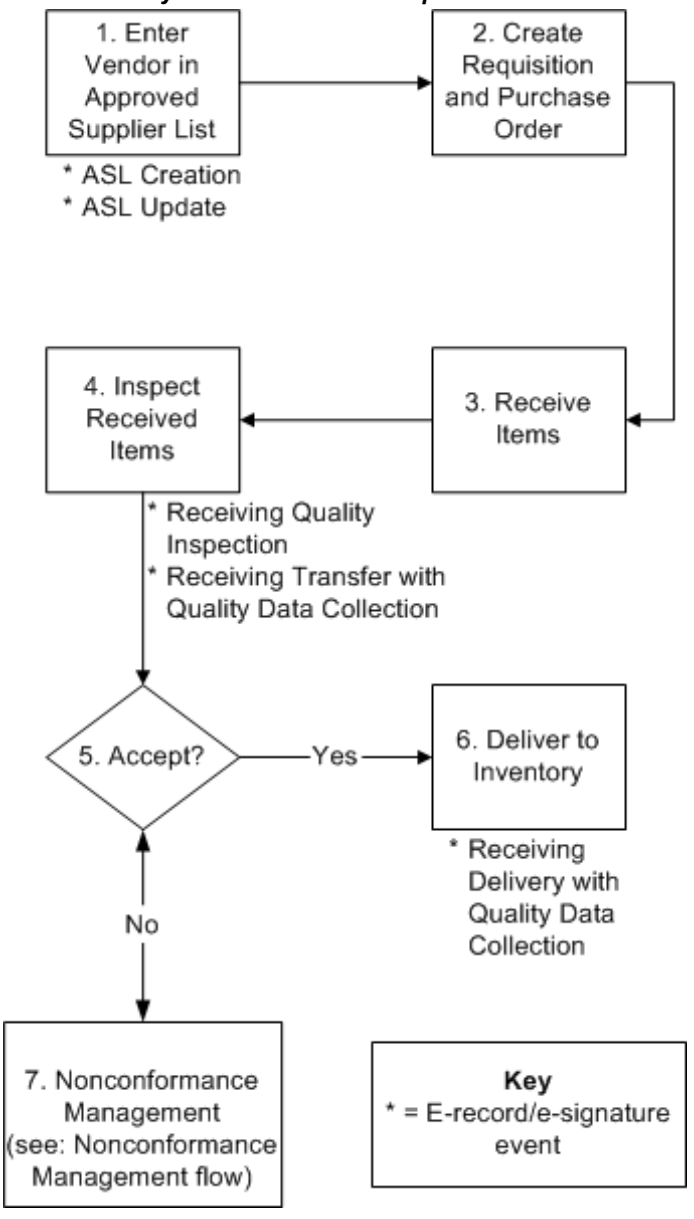
Step	Name	Description	E-Record Enabled Events
1	Initiate Corrective Action Request (CAR)	Create a corrective action request. Occurs within Oracle Applications (OA).	Corrective Action Creation
2	Assign Owner	Assign an owner to the CAR. Occurs within OA.	Corrective Action Update
3	Identify Containment	Identify containment action. Occurs within OA.	Corrective Action Update Corrective Action Approval, if Request eSignature collection element = Yes (see: Corrective Action Approval, *Oracle Quality User's Guide*).
4	Implement Containment	Implement containment action and obtain approvals if required. Occurs within OA.	Corrective Action Update Corrective Action Implementation Approval, if Request eSignature collection element = Yes (see: Corrective Action Implementation Approval, *Oracle Quality User's Guide*).
5	Review and Analyze	Review the corrective actions taken. Occurs within OA.	Corrective Action Update Corrective Action Review Approval, if Request eSignature collection element = Yes (see: Corrective Action Review Approval, *Oracle Quality User's Guide*).
6	Determine Root Cause	Determine the main cause of the problem. Occurs outside of OA.	Corrective Action Update
7	Implement CAPA	After identifying the corrective and preventive actions, implement them. Occurs within OA.	Corrective Action Update Corrective Action Approval, if Request eSignature collection element = Yes (see: Corrective Action Approval, *Oracle Quality User's Guide*).
8	Verify Effectiveness	Verify that the corrective action effectively solved the main cause of the quality problem. Occurs within OA.	Corrective Action Update
9	Follow Up and Close CAR	Perform any tasks identified in the previous step as necessary to resolve the quality problem. Occurs outside of OA. Close the CAR. Occurs within OA.	Corrective Action Update Corrective Action Approval, if Request eSignature collection element = Yes (see: Corrective Action Approval, *Oracle Quality User's Guide*).

Procure to Pay

The following diagram illustrates an example business flow of purchasing, then receiving supplies needed for manufacturing. Each numbered step is explained in the table following the diagram.

Important: You must implement Oracle Quality and set the profile option QA:PO Inspection to Oracle Quality in order to capture e-records and e-signatures in Oracle Purchasing.

Procure to Pay Business Flow Example



Note: The following table explains each step in detail.

Procure to Pay Business Flow Steps

Step	Name	Description	E-Record Enabled Events
1	Enter Vendor in Approved Supplier List	Enter the vendor in the Approved Supplier List (ASL). Occurs within Oracle Applications (OA).	ASL Creation ASL Update
2	Create Requisition and Purchase Order	Create a requisition. Once the requisition is approved, convert it to a purchase order. Occurs within OA.	No transactions
3	Receive Items	Receive the purchased items from the supplier. Occurs within OA.	No transactions
4	Inspect Received Items	Perform a quality inspection of the received items. Occurs within OA.	Receiving Quality Inspection Receiving Transfer with Quality Data Collection
5	Accept?	Accept the received items into inventory? Occurs within OA. Yes: Deliver to Inventory No: Nonconformance Management	No transactions
6	Deliver to Inventory	Put the received items in inventory.	Receiving Delivery with Quality Data Collection
7	Nonconformance Management (see: Nonconformance Management flow, page 1-9)	Manage purchased materials with quality defects. Occurs within OA.	See: Nonconformance Management flow

Oracle E-Records Enabled Transactions Summary

The following tables summarize the discrete manufacturing events seeded in Oracle E-Records. If an event includes an attachment, then the e-record for the event contains the attachment, also.

> **Tip:** The tables show the default e-signature setting for each event. You can modify the seeded event settings as needed.

Oracle Engineering

Event	Online or Deferred E-signature	Child Events/ Other
ECO Creation	None	N/A
ECO Update	None	N/A
ECO Approval	Deferred	Oracle Approvals Management is not used to request ECO Approvals. The person who creates or updates the ECO specifies which approval list to use.
ECO Implementation	None	Transfer to Manufacturing Copy to Manufacturing Bill Creation Bill Update Routing Creation Routing Update
ECO Schedule	None	N/A
ECO Reschedule	None	N/A
ECO Cancellation	None	N/A
Transfer to Manufacturing	Online	Item Creation Bill Creation Routing Creation
Copy to Manufacturing	Online	N/A

See E-records and E-signatures for Oracle Engineering, *Oracle Quality User's Guide* for detailed information regarding each event.

Oracle Inventory

Event	Online or Deferred E-signature	Child Events/ Other
Item Creation	Online	N/A
Item Update	Online	N/A
Item Organization Assignment	Online	Item Creation (within the organization)
Item Revision	Online	N/A
Item Cross Reference	None	N/A
Miscellaneous Transactions	Online	N/A

See E-records and E-signatures for Oracle Inventory, *Oracle Quality User's Guide* for detailed information regarding each event.

Oracle Bills of Material

Event	Online or Deferred E-signature	Child Events/ Other
Bill Creation	Online	N/A
Bill Update	Online	N/A
Routing Creation	Online	N/A
Routing Update	Online	N/A
Mass Change Bills	Online	ECO Creation

See E-records and E-signatures for Oracle Bills of Material, *Oracle Quality User's Guide* for detailed information regarding each event.

Oracle Work in Process

Event	Online or Deferred E-signature	Child Events/ Other
WIP Material Transactions	Online	N/A
Move Transactions	Online	If quality results are collected as part of this event, then the e-record includes the quality information.
Completion Transactions	Online	If quality results are collected as part of this event, then the e-record includes the quality information.

See E-records and E-signatures for Oracle Work in Process, *Oracle Quality User's Guide* for detailed information regarding each event.

Oracle Quality

Event	Online or Deferred E-signature	Child Events/ Other
Collection Element Creation	None	N/A
Collection Element Update	None	N/A
Collection Plan Creation	Online	N/A
Collection Plan Update	Online	N/A
Specification Creation	Deferred	Initial status is Draft. Initiate approval by selecting Tools > Request e-signature approval from the Specifications window menu (see: Manufacturing and Distribution Manager Windows and Navigation Paths, page A-3).
Specification Organization Assignment	Online	Specification Creation (within the organization)
Specification Update	Online	N/A

Event	Online or Deferred E-signature	Child Events/ Other
Quality Result Creation	Online	Collect an e-signature for every result row or an e-signature for all result rows. Use the Record Option field in the Collection Plans window to collect e-signatures by row or by collection.
Quality Result Update	Online	Collect an e-signature for every result row or an e-signature for all result rows. Use the Record Option field in the Collection Plans window to collect e-signatures by row or by collection.
Nonconformance Creation	None	N/A
Nonconformance Update	None	N/A
Nonconformance Header Approval	Deferred	Initiated when entering nonconformance results if you enter Yes in the eSignature Required collection element field (see: Nonconformance Header Approval and Detail Approval, *Oracle Quality User's Guide*).
Nonconformance Detail Approval	Deferred	Initiated when entering nonconformance results if you enter Yes in the eSignature Required collection element field (see: Nonconformance Header Approval and Detail Approval, *Oracle Quality User's Guide*).
Disposition Creation	None	N/A
Disposition Update	None	N/A
Disposition Header Approval	Deferred	Initiated when entering disposition results if you enter Yes in the eSignature Required collection element field (see: Disposition Header Approval and Detail Approval, *Oracle Quality User's Guide*).
Disposition Detail Approval	Deferred	Initiated when entering disposition results if you enter Yes in the eSignature Required collection element field (see: Disposition Header Approval and Detail Approval, *Oracle Quality User's Guide*).
Corrective Action Creation	None	N/A
Corrective Action Update	None	N/A
Corrective Action Approval	Deferred	Initiated when entering corrective action results if you enter Yes in the eSignature Required collection element field (see: Corrective Action Approval, *Oracle Quality User's Guide*).

Event	Online or Deferred E-signature	Child Events/ Other
Corrective Action Review Approval	Deferred	Initiated when entering corrective action review results if you enter Yes in the eSignature Required collection element field (see: Corrective Action Review Approval, *Oracle Quality User's Guide*).
Corrective Action Implementation Approval	Deferred	Initiated when entering corrective action implementation results if you enter Yes in the eSignature Required collection element field (see: Corrective Action Implementation Approval, *Oracle Quality User's Guide*).

See E-records and E-signatures for Oracle Quality, *Oracle Quality User's Guide* for detailed information regarding each event.

Oracle Shipping

Event	Online or Deferred E-signature	Child Events/ Other
Delivery Shipment	Not applicable	A concurrent program generates e-records for this transaction, with no option for enabling e-signatures. This event can only occur if Oracle Quality is implemented.

See E-records and E-signatures for Oracle Shipping, *Oracle Quality User's Guide* for detailed information regarding each event.

Oracle Purchasing

Event	Online or Deferred E-signature	Child Events/ Other
ASL Creation	None	N/A
ASL Update	None	N/A
Receiving Quality Inspection	Online	Applicable only if Oracle Quality is implemented.
Receiving Transfer with Quality Data Collection	None	Applicable only if Oracle Quality is implemented.
Receiving Delivery with Quality Data Collection	None	Applicable only if Oracle Quality is implemented.

See E-records and E-signatures for Oracle Purchasing, *Oracle Quality User's Guide* for detailed information regarding each event.

2

Oracle E-Records Setup and Process Exceptions in Discrete Manufacturing

This chapter provides an example of how to set up a transaction to capture e-records and e-signatures. It also discusses exceptions to the standard setup steps and setup process.

This chapter covers the following topics:

- Setting Up E-Records: A Discrete Manufacturing Example
- Setup and Process Exceptions

Setting Up E-Records: A Discrete Manufacturing Example

Implementing E-records, *Oracle E-Records Implementation Guide* provides detailed instructions on how to set up any transaction for use with Oracle E-Records. In many discrete manufacturing business scenarios, much of the transactional data used in these setup steps has been seeded for you. The exceptions to the standard Oracle E-Records setup steps and process flow (see: Online Flow, *Oracle E-Records Implementation Guide* and Deferred Flow, *Oracle E-Records Implementation Guide*) include:

- ECO Approval (Oracle Engineering), page 2-20
- Nonconformance, Disposition, and Corrective Actions (Oracle Quality), page 2-29
- Delivery Shipment (Oracle Shipping), page 2-32
- Miscellaneous Transactions (Oracle Inventory), page 2-32
- All transactions performed using the Oracle Applications (OA) Framework, page 2-19

The following example uses the Bills of Material Creation transaction (see: Creating a Bill of Material, *Oracle Bills of Material User's Guide*) to demonstrate how to enable a discrete manufacturing transaction for use with Oracle E-Records.

Example: Implementing Oracle E-Records for a Medical Device Manufacturer
A medical device manufacturer plans to implement Oracle E-Records for their manufacturing process. They decide to use the Bills of Material Creation transaction as a model to describe the setup steps required. Other business events in their manufacturing process require similar setup steps.

Requirements

- Collect an e-record and e-signatures when creating a new bill of material in manufacturing organization M1.
- Collect an e-record only for new bills of material created in all other organizations.

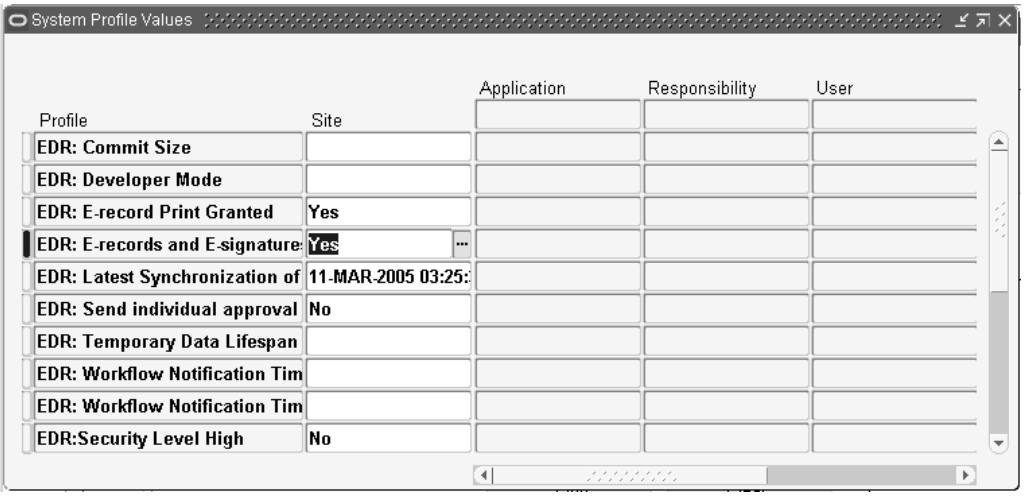
- Representatives from marketing, manufacturing, and product development must approve a new bill of material created in organization M1. The system must allow signers to be added or deleted when submitting the document for signatures. (see: Setting Up Ad Hoc Signers, *Oracle E-Records Implementation Guide*).

How to enable the Oracle E-Records profile options:

Before enabling a specific transaction for Oracle E-Records, decide how to set the profile options described in Enabling Profile Options, *Oracle E-Records Implementation Guide*. The following steps show how to define one of the profile options, EDR: E-records and E-signatures.

1. Navigate to the Find System Profile Values window (see: System Adminstrator Windows and Navigation Paths, page A-1).
2. Find and select the profile **EDR: E-records and E-signatures**.
3. Enter **Yes** for the appropriate level (Site, Application, or Responsibility).

   Tip: Oracle recommends setting the profile option value at the Site level.



How to Enable the Workflow Business Event Subscription:

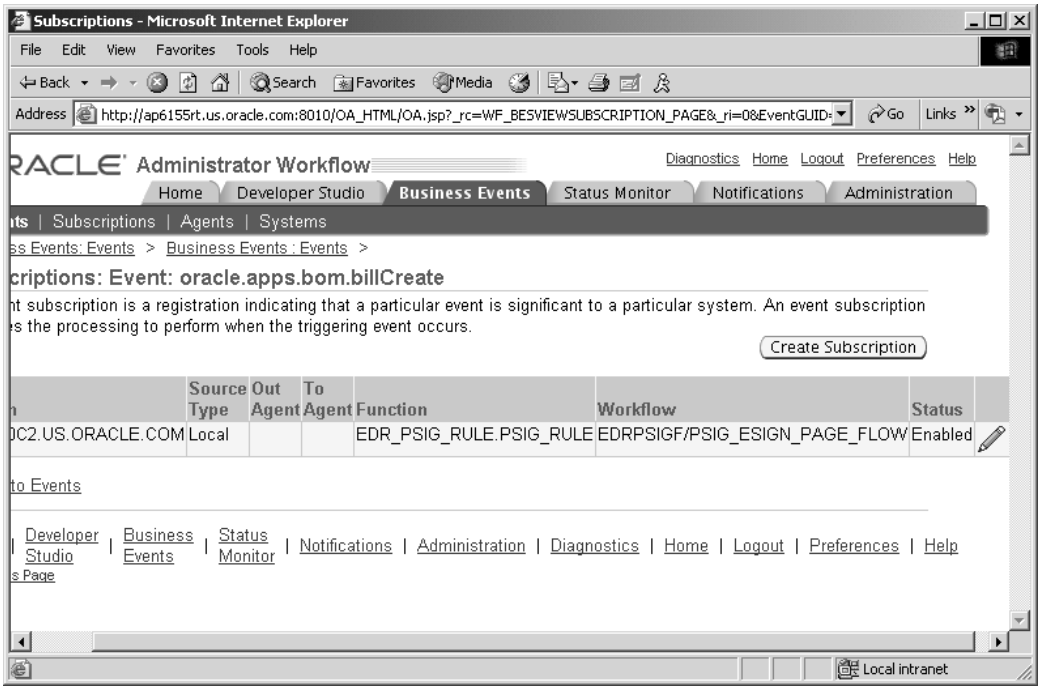
Enabling the subscription lets users receive workflow notifications related to a business event. For this scenario, enable the business event BOM Bill of Materials Create so users can receive a workflow notification requesting an e-signature.

1. Navigate to the Business Events page (see: Workflow Administrator Pages and Navigation Paths, page A-1).
2. Search for the business event that you want to enable. Refer to Oracle E-Records Event Data for Discrete Manufacturing, page C-1, for a listing of seeded business events.
3. Enable the business events that you plan to use by choosing Update (the pencil icon) for each business event.

   Tip: If you do not see the Update button, then you do not have administrative privileges and the system administrator needs to enable the business events. To verify this, select the Administration

tab. An asterisk (*) in the Workflow System Administrator field indicates that all users have been granted administrative privileges.

4. Enable the subscription for each business event that you plan to use by choosing the Subscription button for each business event.



How to Set Up Oracle Approvals Management:

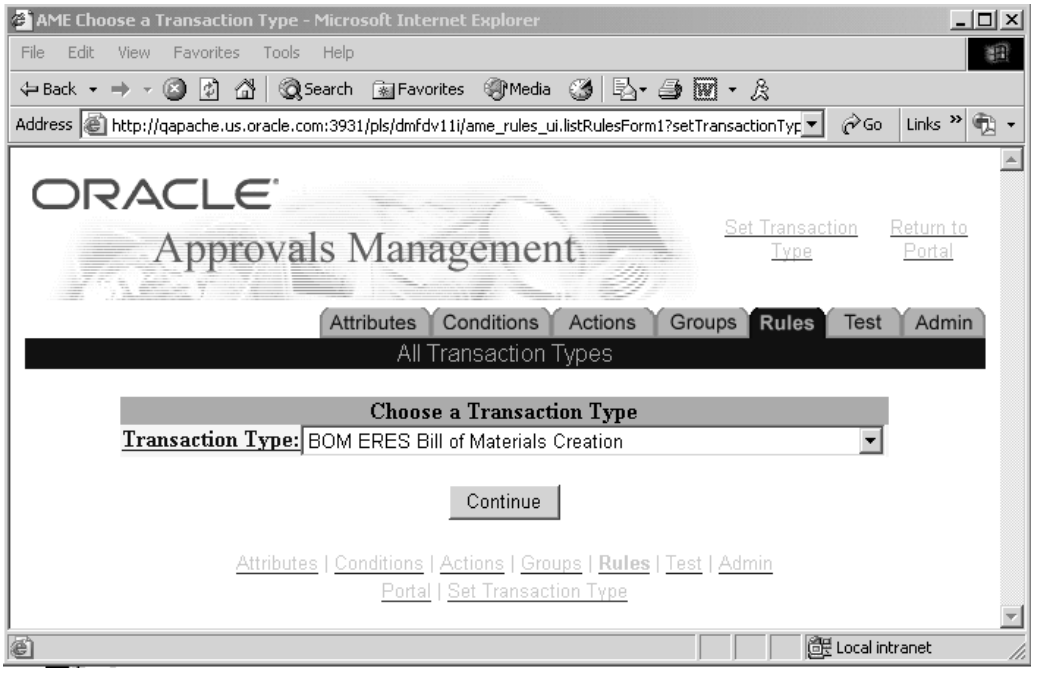
Use Oracle Approvals Management to define the rules for when to require e-records and e-signatures.

This example follows the setup steps explained in Setting Up Approvals Management, *Oracle E-Records Implementation Guide*. Refer to these steps as well as *Implementing Oracle Approvals Management* at http://metalink.oracle.com (search for Metalink Note #282529.1) for more information about setting up Oracle Approvals Management.

Creating Transaction Attributes

Use transaction attributes to define your rule criteria. In this example, verify that an organization attribute exists (so you can define a condition stating that the organization equals M1 later).

1. Navigate to Oracle Approvals Management (see: Approvals Management Application Administrator Windows and Navigation Paths, page A-2), then select Set Transaction Type.
2. In the Transaction Type field, select BOM ERES Bill of Materials Creation from the list of values.

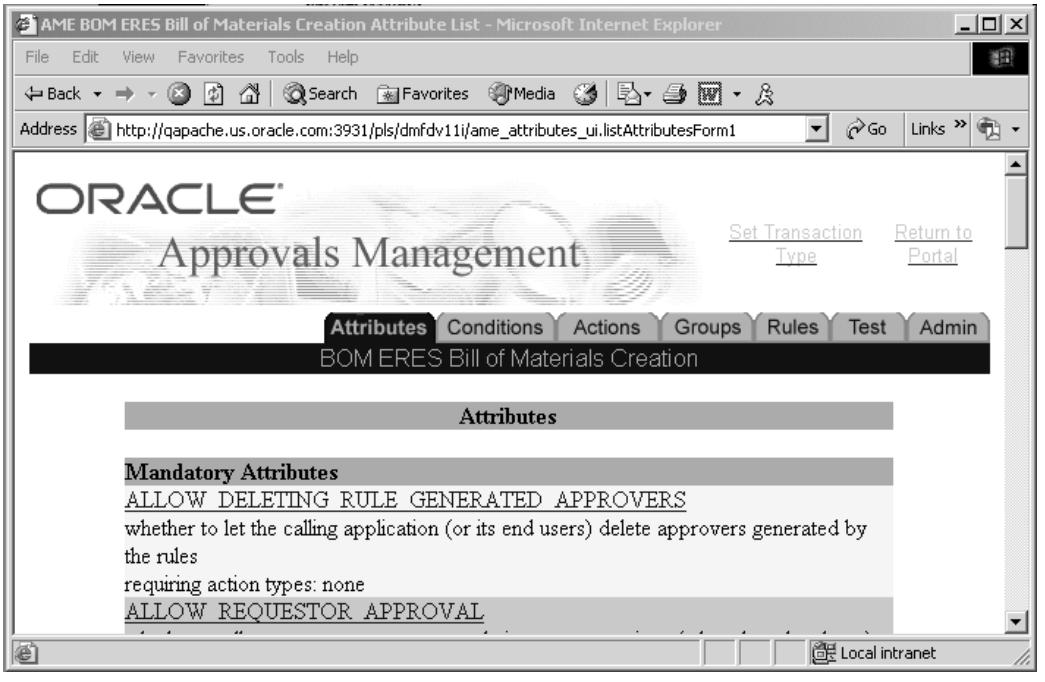


3. Choose Continue.

   This action selects the transaction type.

4. Select the Attributes tab to review the list of seeded attributes for the transaction.

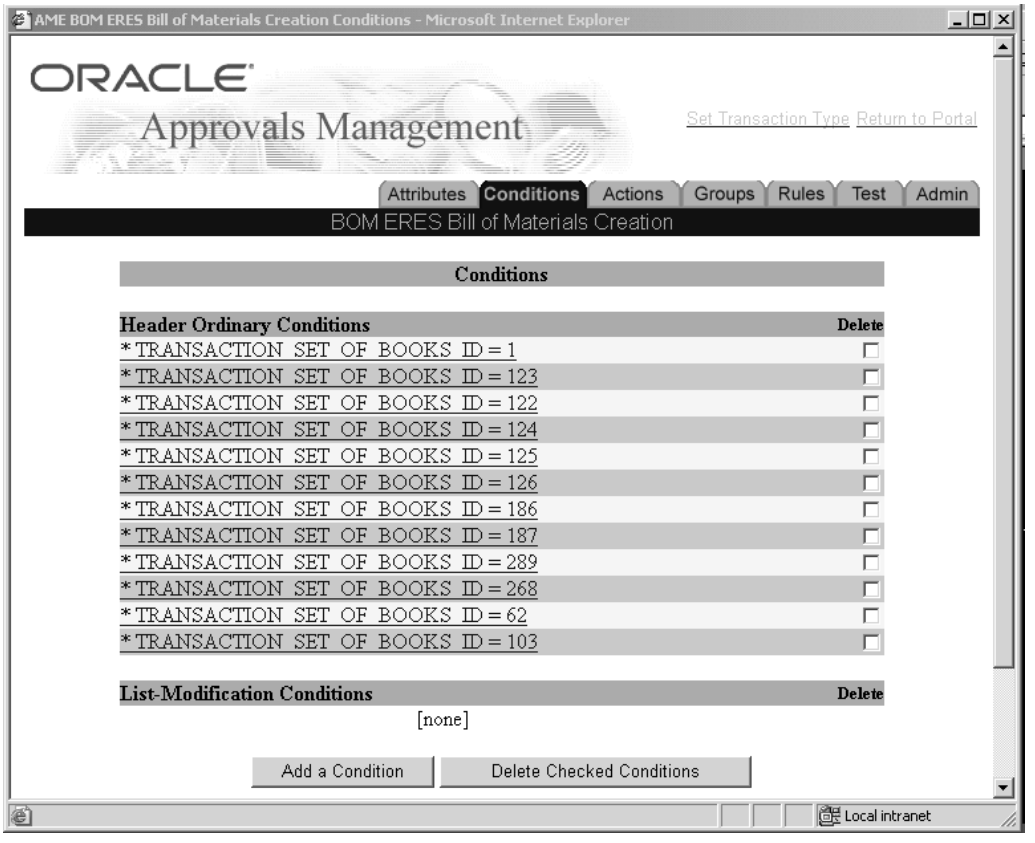
   Notice the ORGANIZATION_CODE attribute. Use this attribute later to build a condition.



Creating Conditions

When you create a condition, you define when an action can occur. For this example, collect an e-record and e-signature when the condition of a new bill of material created within organization M1 is met.

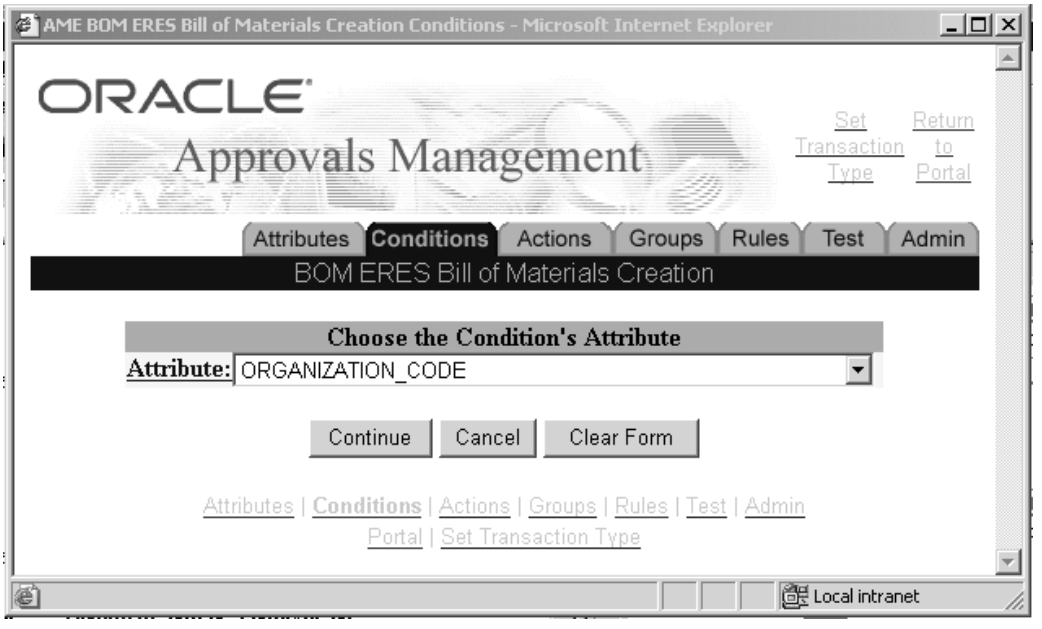
5. Select the Conditions tab, then choose Add a Condition.



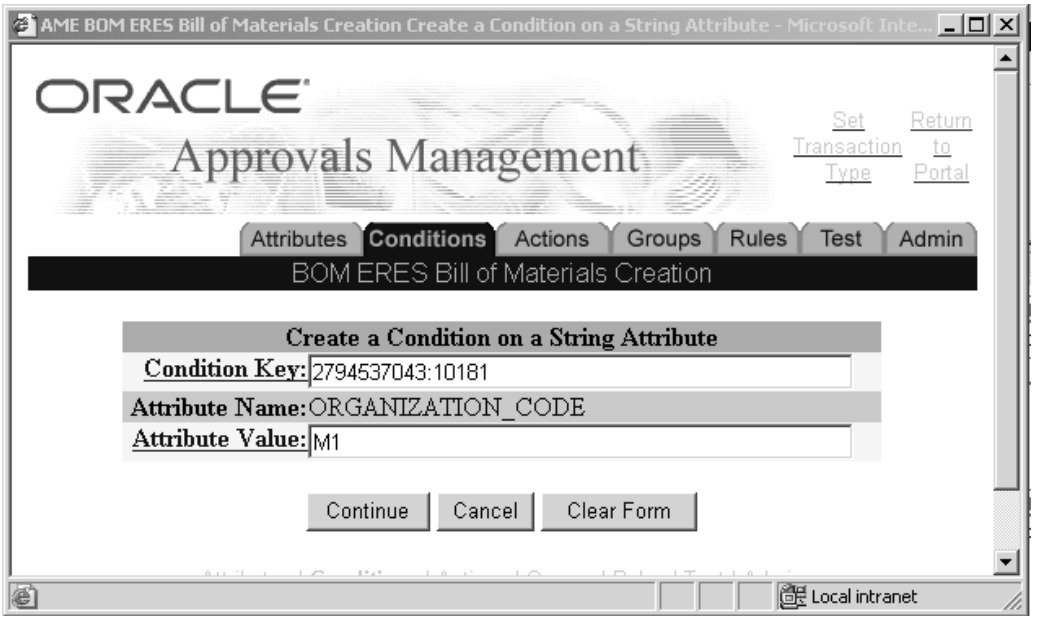
6. Select a Condition Type, then choose Continue.

   For more information about Condition Types, see: Condition Types, *Implementing Oracle Approvals Management* at http://metalink.oracle.com. Search for Metalink Note #282529.1.

7. Select ORGANIZATION_CODE from the Attribute list of values. Choose Continue.



8. The condition key automatically generates. Enter the organization code M1 in the Attribute Value field.



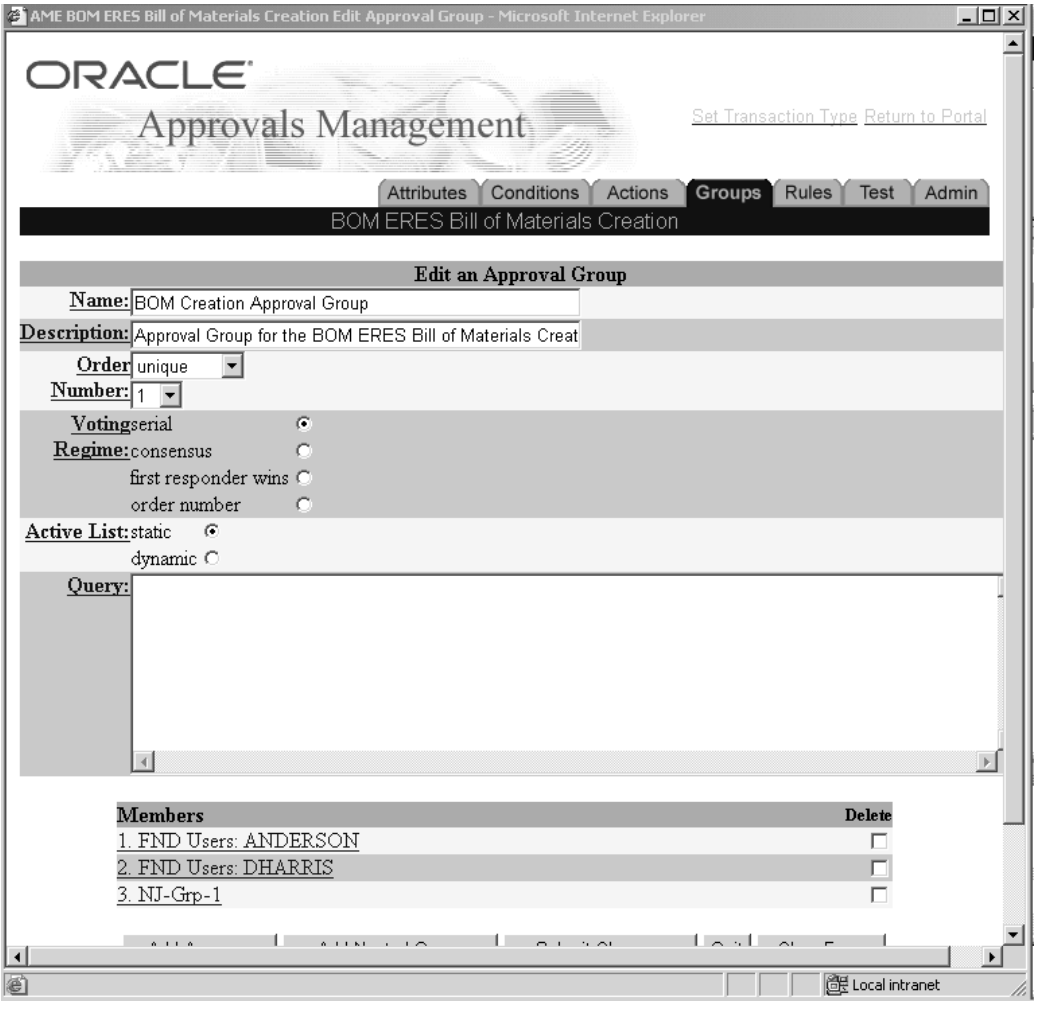
9. Choose Continue, then Quit.

   You receive a confirmation message stating that the condition was created.

Creating Approval Groups

10. For this example, use a previously defined Approval Group named BOM Creation Approval Group.

    For more information about creating Approval Groups, see: Approval Groups, *Implementing Oracle Approvals Management*, at http://metalink.oracle.com. Search for Metalink Note #282529.1.

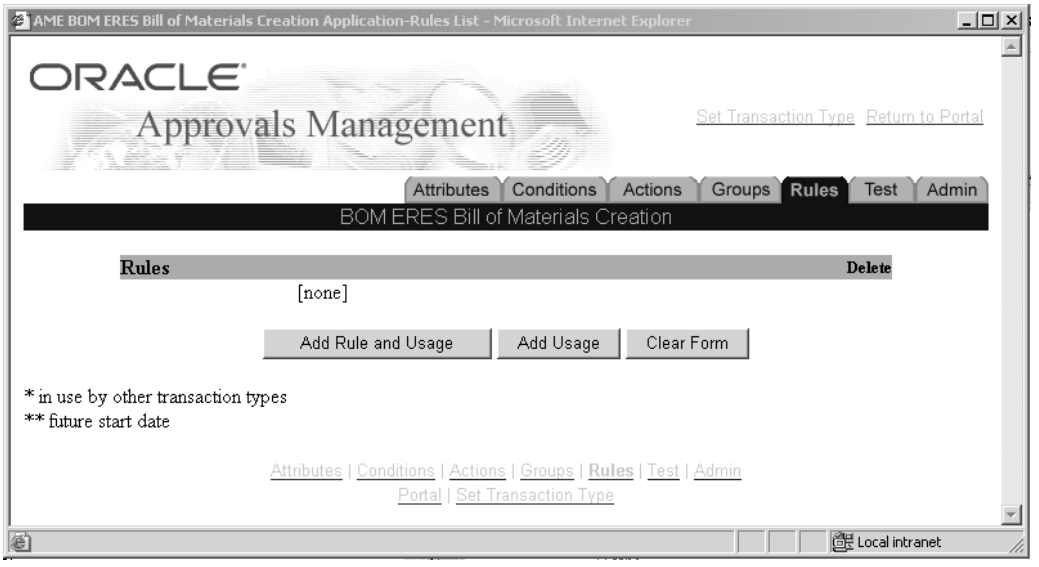


Defining Approval Rules

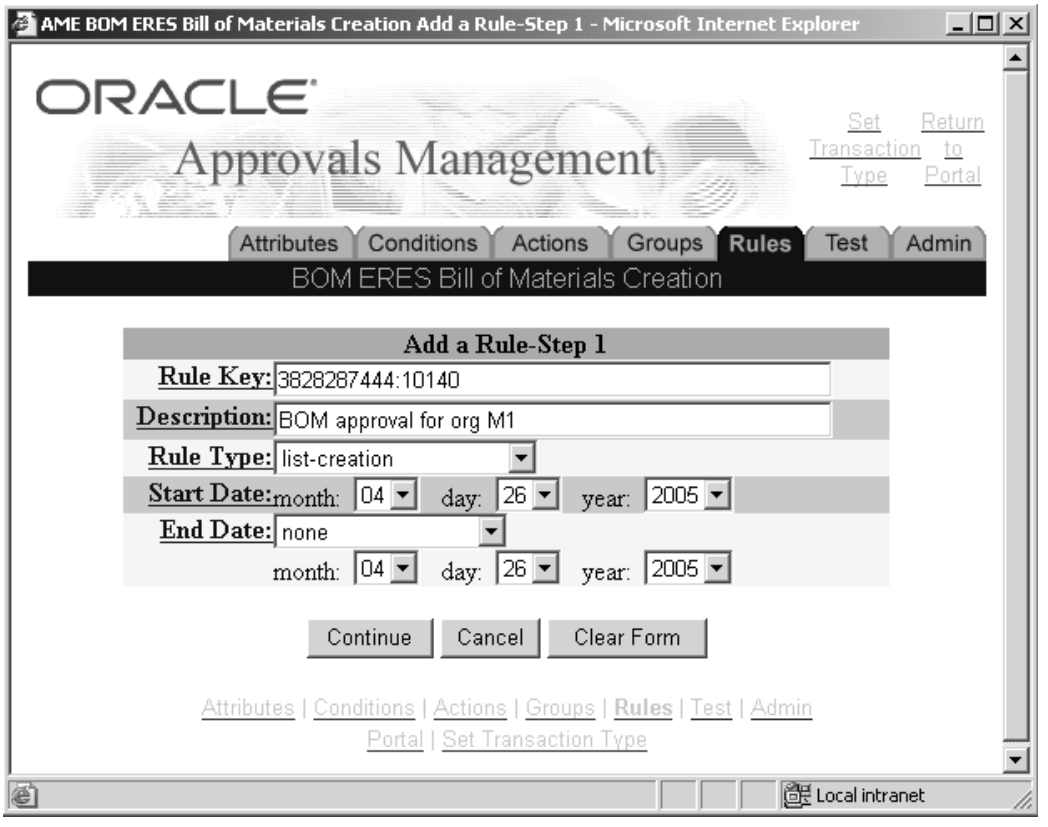
Rules associate one or more conditions with an approval in an if-then statement. The rule is the culmination of all the previous Oracle Approval Management setup steps.

For more information about creating rules, see: Rules, *Implementing Oracle Approvals Management*, at http://metalink.oracle.com. Search for Metalink Note #282529.1.

11. Select the Rules tab, then choose Add Rule and Usage.



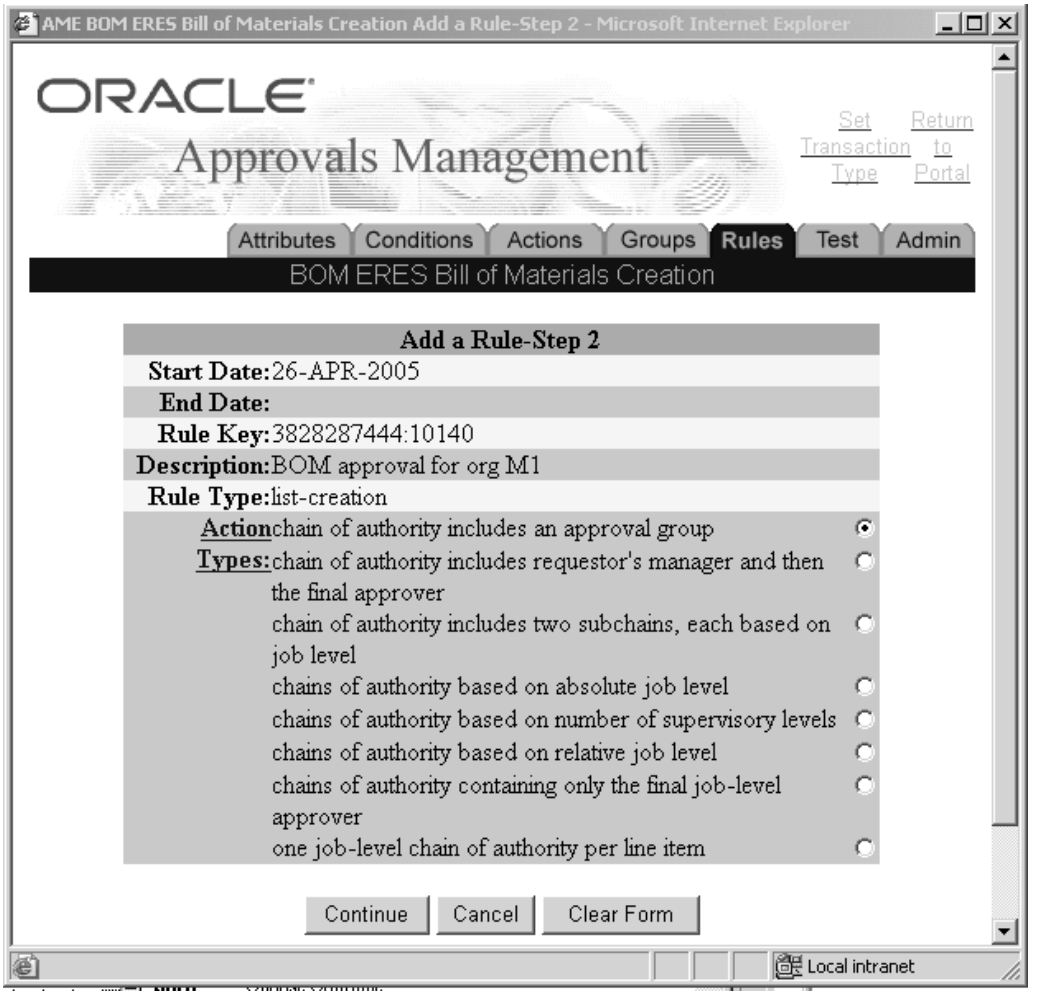
12. Enter these values in the following three fields:
    - Description: BOM approval for org M1
    - Rule Type: list-creation
    - Start Date: today's date



13. Choose Continue.

14. Select the Action Type for the approval.

    Select the following Action Type: chain of authority includes an approval group.



15. Choose Continue.

16. Select the approval group for this rule in the approval-group chain of authority Actions field.

    For this example, select the BOM Creation Approval Group.

17. Choose Continue.
18. Select **none** for Subordinate Item Class.

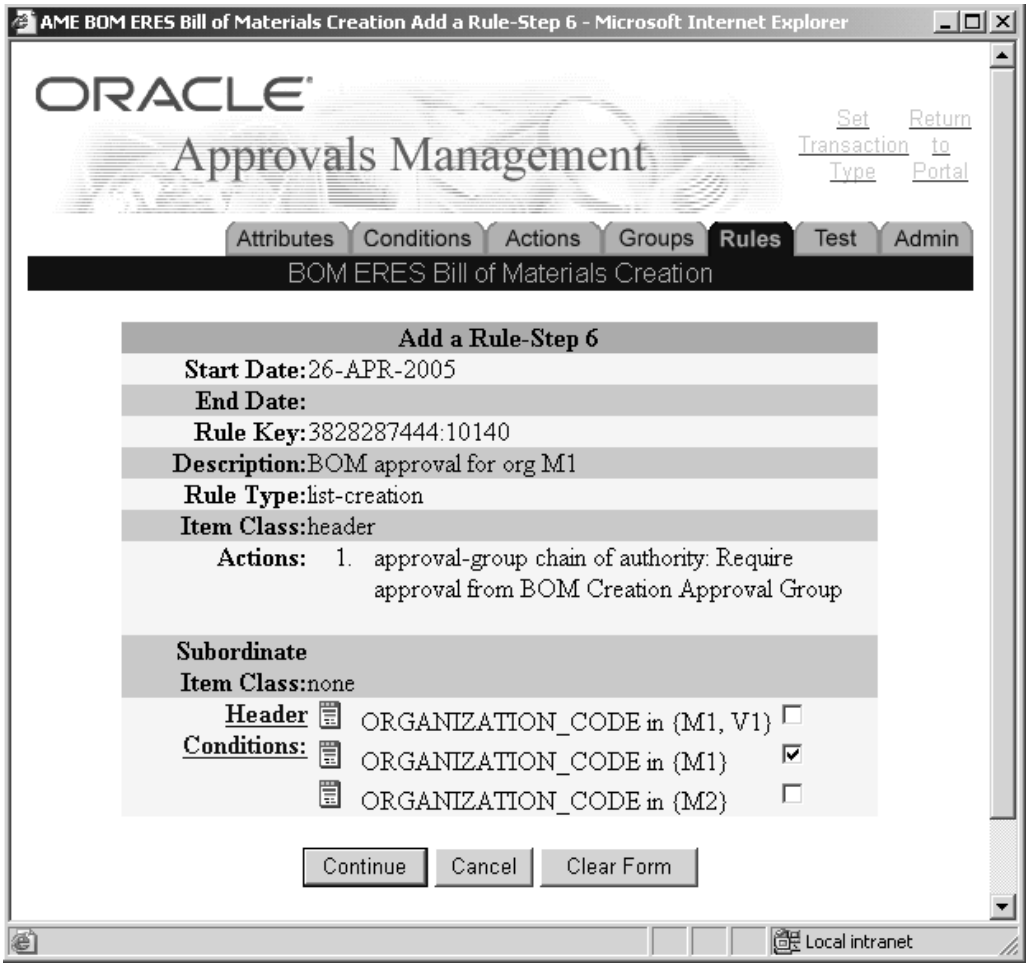
19. Choose Continue.
20. Select the header level condition attribute for the rule, then choose Continue.

    For this example, select ORGANIZATION_CODE.

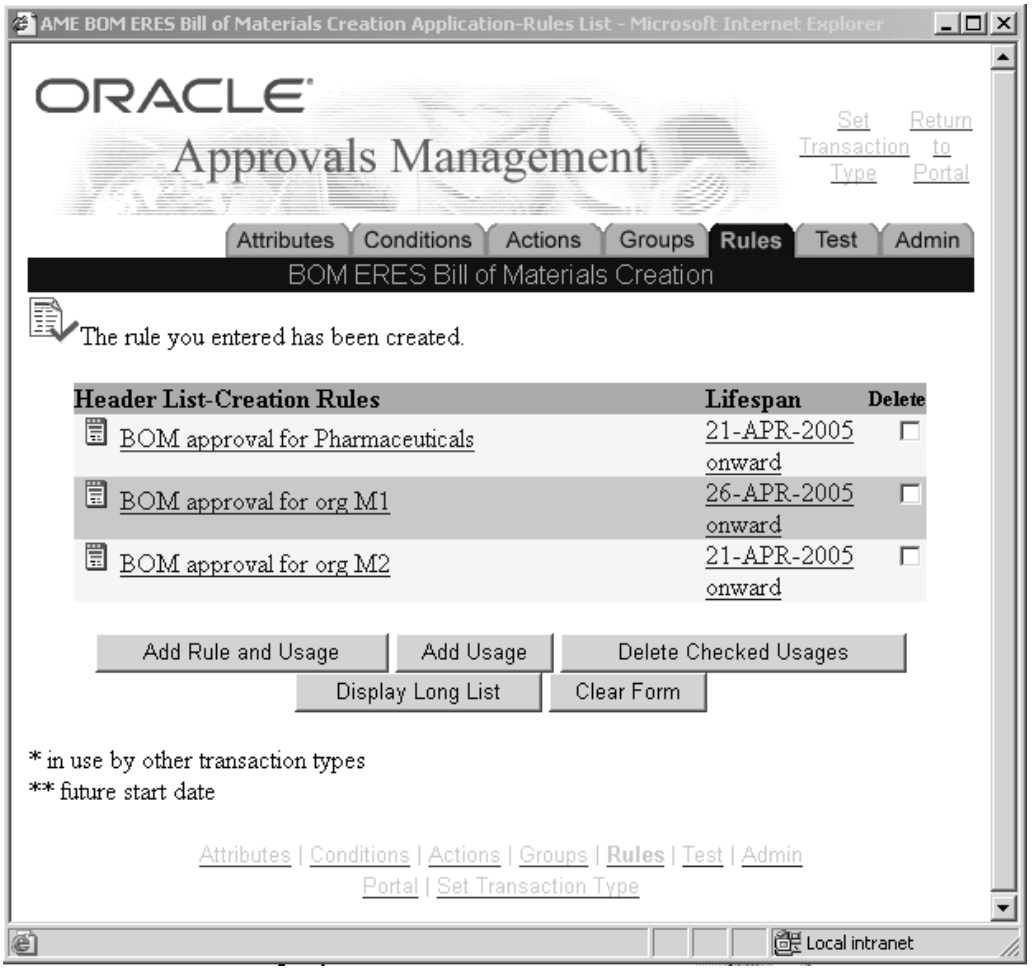
21. Select the condition you want to use in the rule evaluation, then choose Continue.

    For this example, select ORGANIZATION_CODE in {M1}.

    Note: If you select more than one condition, all of the conditions must be met for the rule to take effect.



View the rule you created.



How to Set Up the Configuration Variables:

Use configuration variables to define the variables for each transaction enabled for use with Oracle E-Records. Oracle E-Records requires a minimum of the following variables for each transaction:

- EREC_REQUIRED
- EREC_STYLE_SHEET
- EREC_STYLE_SHEET_VER
- ESIG_REQUIRED

Add additional variables, such as CHANGE_SIGNERS (see: Setting Up Ad Hoc Signers, *Oracle E-Records Implementation Guide*), if desired.

Configuration variables consist of two parts:

- Transaction variables
- Rule variables

Transaction variables contain the default variables for a transaction. Rule variables override the transaction variables for a transaction that meets the rule criteria defined in Oracle Approvals Management.

You only need to define those rule variables that are necessary to override default transaction variables. For example, there is no need to define a rule variable for EREC_REQUIRED when the value equals Y for both the transaction and the rule.

For detailed information regarding configuration variables, refer to Setting Up the Configuration Variables, *Oracle E-Records Implementation Guide*.

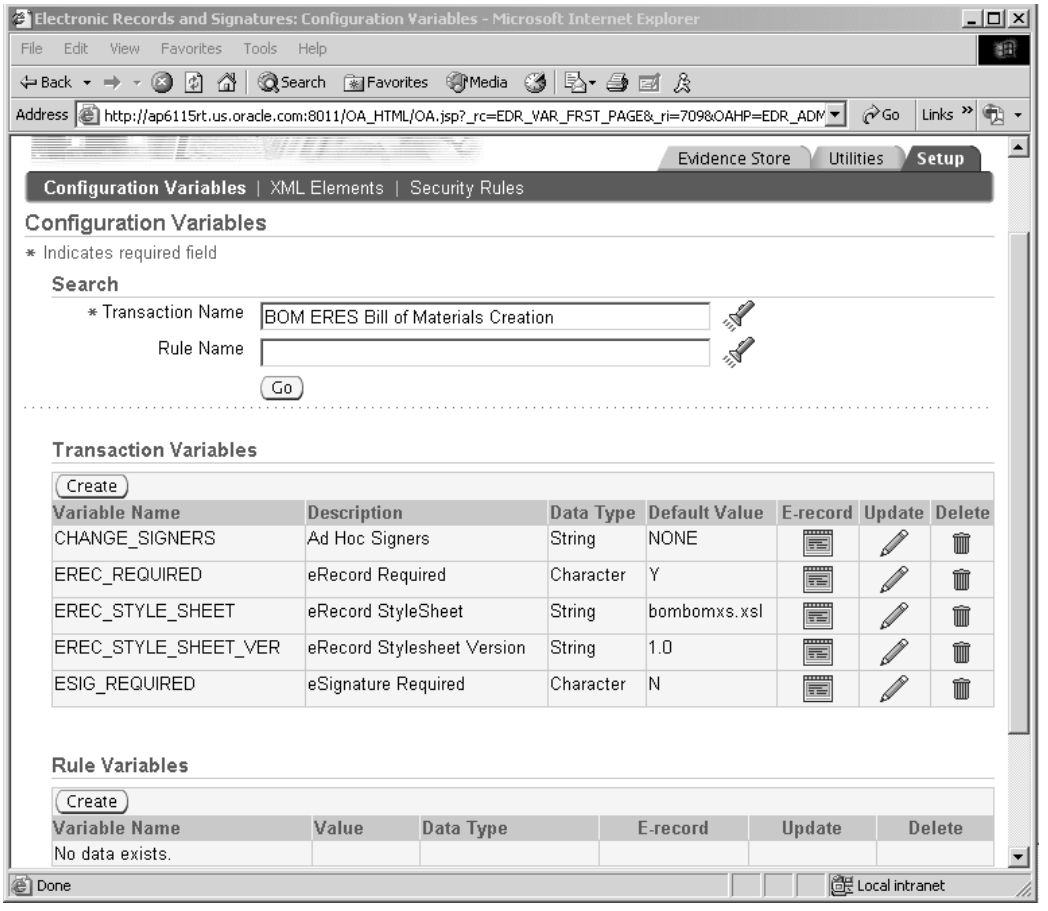
1. Navigate to the Configuration Variables page (see: ERES Administrator Windows and Navigation Paths, page A-2).
2. Search for the Transaction Name BOM ERES Bill of Materials Creation.

   Review the Default Values for the transaction and modify them if necessary.

   For this example, set the variable default values as follows:

   - EREC_REQUIRED = Y
   - EREC_STYLE_SHEET = bombomxs.xsl
   - EREC_STYLE_SHEET_VER = 1.0
   - ESIG_REQUIRED = N
3. Add additional variables as needed.

   In this example, add CHANGE_SIGNERS = NONE (see: Setting Up Ad Hoc Signers, *Oracle E-Records Implementation Guide*). Add this variable to meet the requirement for M1 of allowing signers to be added or deleted prior to submitting the document for approval.. The default value of NONE does not allow signers to be added or deleted. You can add a rule variable with CHANGE_SIGNERS = ALL to override the default value.



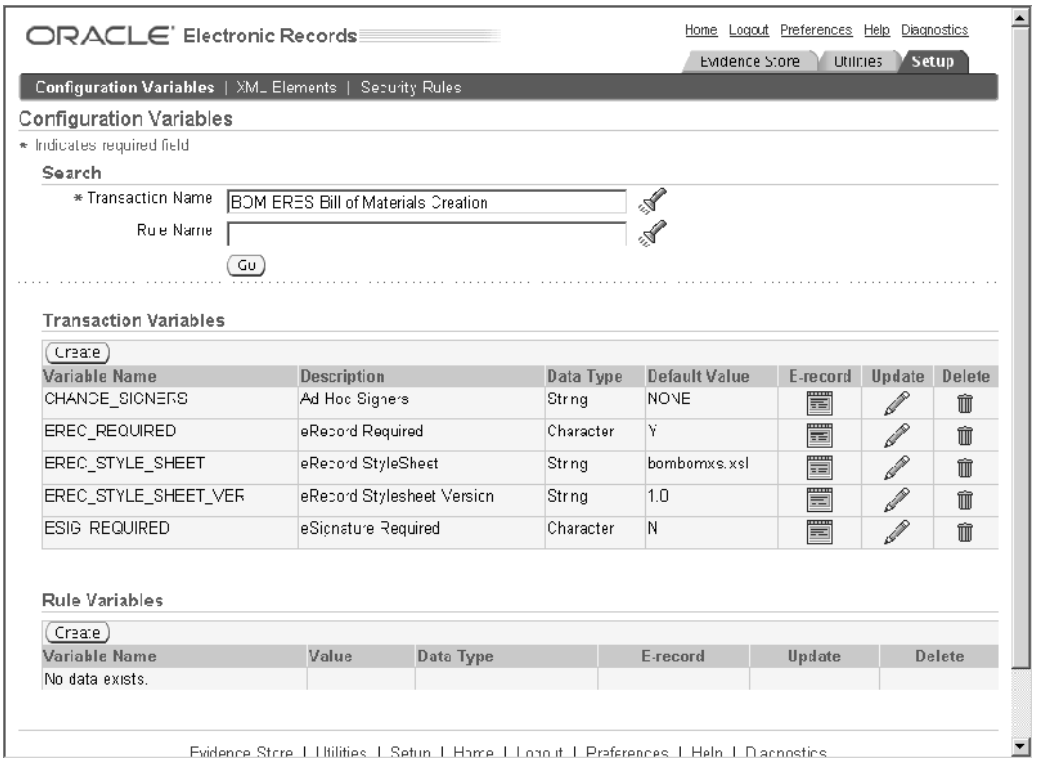
How to Set Up a Rule Variable

The requirements for organization M1 differ from the other manufacturing organizations. M1 requires e-signatures for each new bill of material created. Define a rule variable to meet these requirements.

4. Enter BOM ERES Bill of Materials Creation in the Transaction Name field. Leave the Rule Name field blank. Choose Go.

   This shows all rules associated with the transaction. Currently, there are no rules associated with the BOM ERES Bill of Materials Creation transaction.

   If no rules are defined, then the default transaction variables apply to all transactions.



5. Choose Search (flashlight icon) for the Rule Name field.

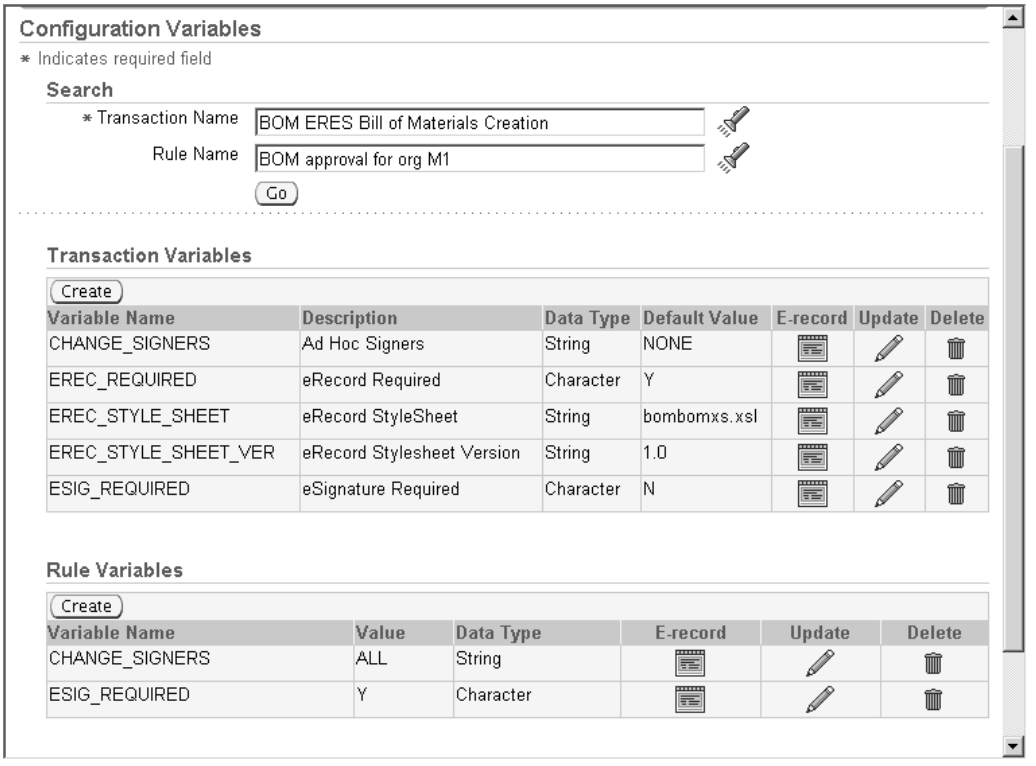
   In the Search and Select: Rule Name page, choose Go. The rule you defined earlier, BOM approval for org M1, appears. Select this rule.

6. Underneath the Rule Variables title, choose Create.

7. Search for and select the following input variables individually:

   - ESIG_REQUIRED. Enter Y in the Variable Value field.
   - CHANGE_SIGNERS. Enter ALL in the Variable Value field.

   Now, when a BOM ERES Bill of Materials Creation transaction is entered for organization M1, the system requires an e-signature and allows a list of signers to be modified before submitting the document for signatures.



Setup and Process Exceptions

The following discrete manufacturing transactions require setup and process steps that differ from those listed in Implementing E-records, *Oracle E-Records Implementation Guide* and discussed throughout the Oracle E-Records Implementation Guide.

- ECO Approval (Oracle Engineering)

  (see: Creating an ECO, *Oracle Engineering User's Guide*)

- All Nonconformance, Disposition, and Corrective Action Approvals (Oracle Quality)

  (see: *Oracle Quality Implementation Guide*)

- Delivery Shipment (Oracle Shipping)

  (see: Closing a Delivery, *Oracle Shipping Execution User's Guide*)

- Miscellaneous Transactions (Oracle Inventory)

  (see: Performing Miscellaneous Transactions, *Oracle Inventory User's Guide*)

All discrete manufacturing transactions only integrate with Oracle E-Records when the transaction is entered using Oracle Forms. Oracle E-Records cannot create e-records or e-signatures when discrete manufacturing transactions or data are entered using the Oracle Applications (OA) Framework or a mobile application such as Oracle Mobile Supply Chain Applications (MSCA) or Oracle Warehouse Management System (WMS).

> **Tip:** This tip is for Oracle Quality users who use the Copy Collection Plan program (An OA Framework program. See: Collection Plan Templates, *Oracle Quality User's Guide* for more information.). When copying a parent-child collection plan structure, check the Create Plans as Disabled box. Enable the new plan structure using the Collection Plan

window. This ensures that Oracle E-Records can create e-records and e-signatures for the new plan structure.

Setting up the ECO Approval transaction:

The ECO Approval process requires an additional setup step in Oracle Engineering because this process is not defined in Oracle Approvals Management. You must associate the Change Type and Priority used by the ECO with the ERES Approval Process in Oracle Workflow. See Defining ECO Types, *Oracle Engineering User's Guide* for more information about how to associate a Change Type and Priority with an Oracle Workflow process. The following example explains the setup and process steps required to:

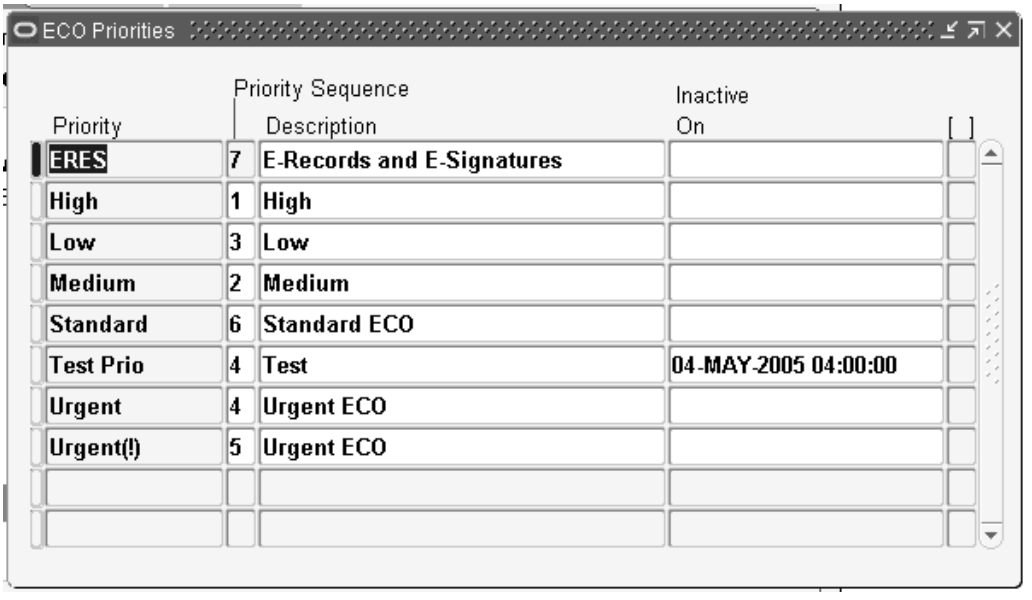
1. Generate an e-record when an engineering change order (ECO) is created.
2. Generate an e-record that requires an e-signature when an ECO is submitted for approval.

Example: Approving an ECO for Manufacturing Using Oracle E-Records

A medical device manufacturer requires e-records each time a new ECO is created and e-signatures once the new ECO is submitted for approval in organization M1. The following steps explain how to accomplish this.

1. Navigate to the ECO Priorities window (see: Manufacturing and Distribution Manager Windows and Navigation Paths, page A-3).

   Enter a priority to use for those ECOs requiring e-signatures. See: Defining ECO Priorities, *Oracle Engineering User's Guide*.



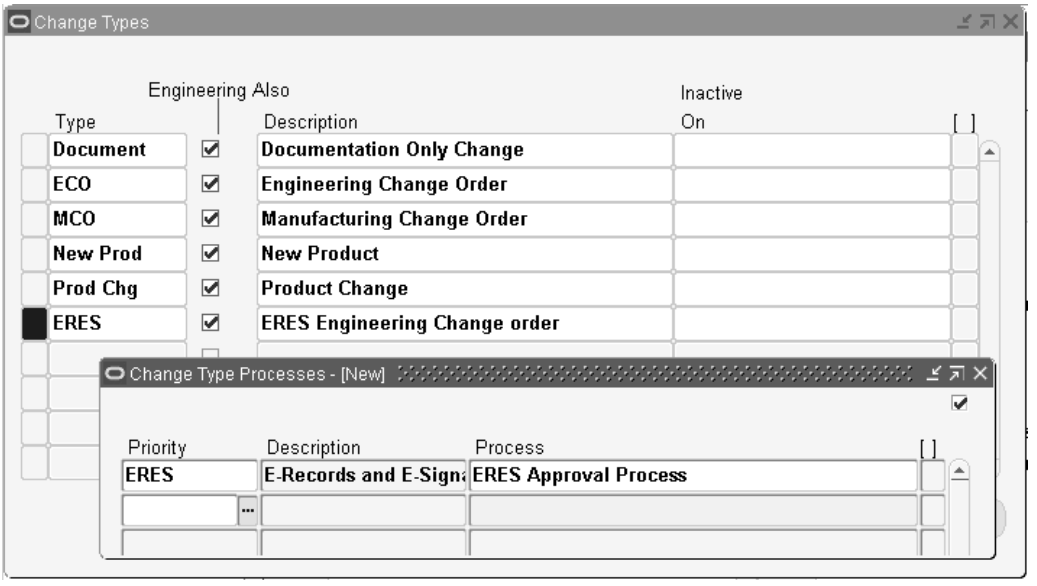
2. Navigate to the Change Types window (see: Manufacturing and Distribution Manager Windows and Navigation Paths, page A-3).

   Enter a change type for those ECOs requiring e-signatures. See: Defining ECO Types, *Oracle Engineering User's Guide*.

   In this example, enter ERES in the Change Type field.

3. Navigate to the Change Type Processes window (see: Manufacturing and Distribution Manager Windows and Navigation Paths, page A-3).

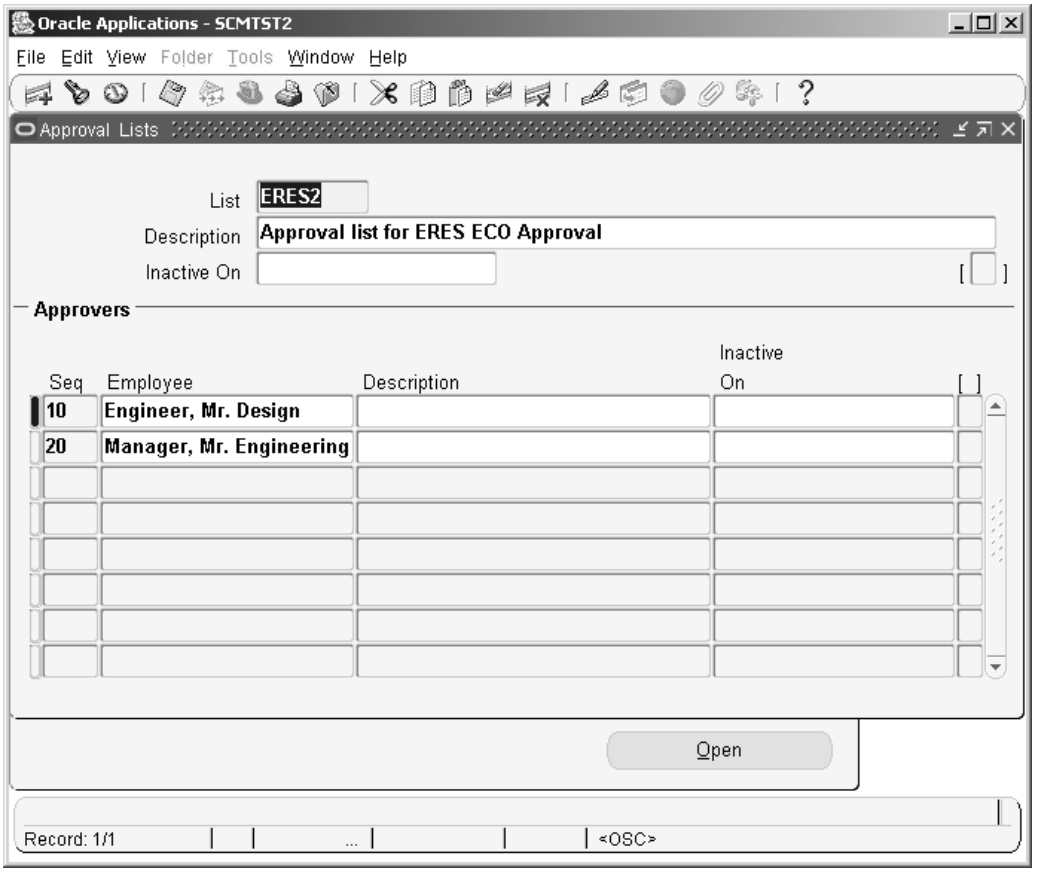
Link the change type and the priority you defined in the previous steps to the ERES Approval Process workflow.



4. Navigate to the Approval Lists window (see: Manufacturing and Distribution Manager Windows and Navigation Paths, page A-3).

   Create a list of the employees responsible for approving new and updated engineering change orders (see: Defining ECO Approval Lists, *Oracle Engineering User's Guide*).

   In this example, create a list named ERES2.



Set Up the ENG ERES ECO Creation Transaction in Oracle Approvals Management

Note: The setup necessary to generate an e-record and request an e-signature when approving an ECO is complete. The remaining steps in **Set Up the ENG ERES ECO Creation Transaction in Oracle Approvals Management** and **Dene the Conguration Variables for the ENG ERES ECO Creation Transaction** complete the setup necessary to generate an e-record when creating a new ECO.

5. Navigate to the Oracle Approvals Management page (see: Approvals Management Application Administrator Windows and Navigation Paths, page A-2).
6. Set the transaction type to ENG ERES ECO Creation.
7. Verify that the following attributes exist:
   - ECO TYPE
   - ORGANIZATION_CODE
8. Add the following conditions:
   - ECO TYPE = ERES
   - ORGANIZATION_CODE = M1
9. Add an approval group.

   For this example, use an existing approval group, Engineering Approval Group.

For more information about creating Approval Groups, see: Approval Groups, *Implementing Oracle Approvals Management*, at http://metalink.oracle.com. Search for Metalink Note #282529.1.

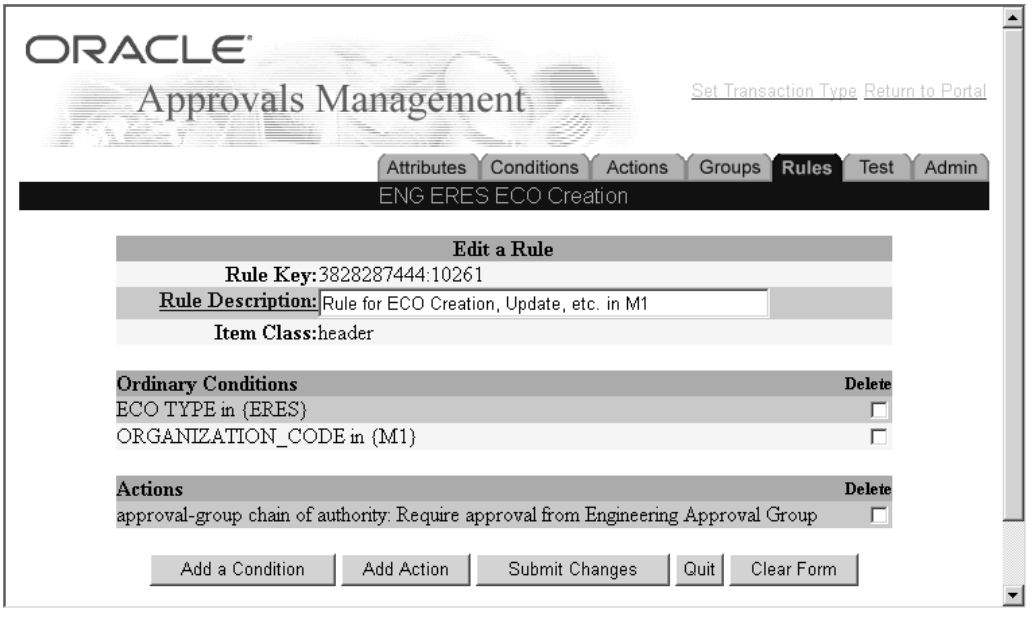
10. Add an approval rule.

    For this example, enter the following information:

    - Description = Rule for ECO Creation, Update, etc. in M1
    - Require approval from the Engineering Approval Group

      Note: The default transaction variables for the ECO Creation transaction include ESIG_REQUIRED equal to N. Therefore, no approval request for e-signatures is made.

    - Select the following header conditions:
      - ECO TYPE in {ERES}
      - ORGANIZATION_CODE in {M1}



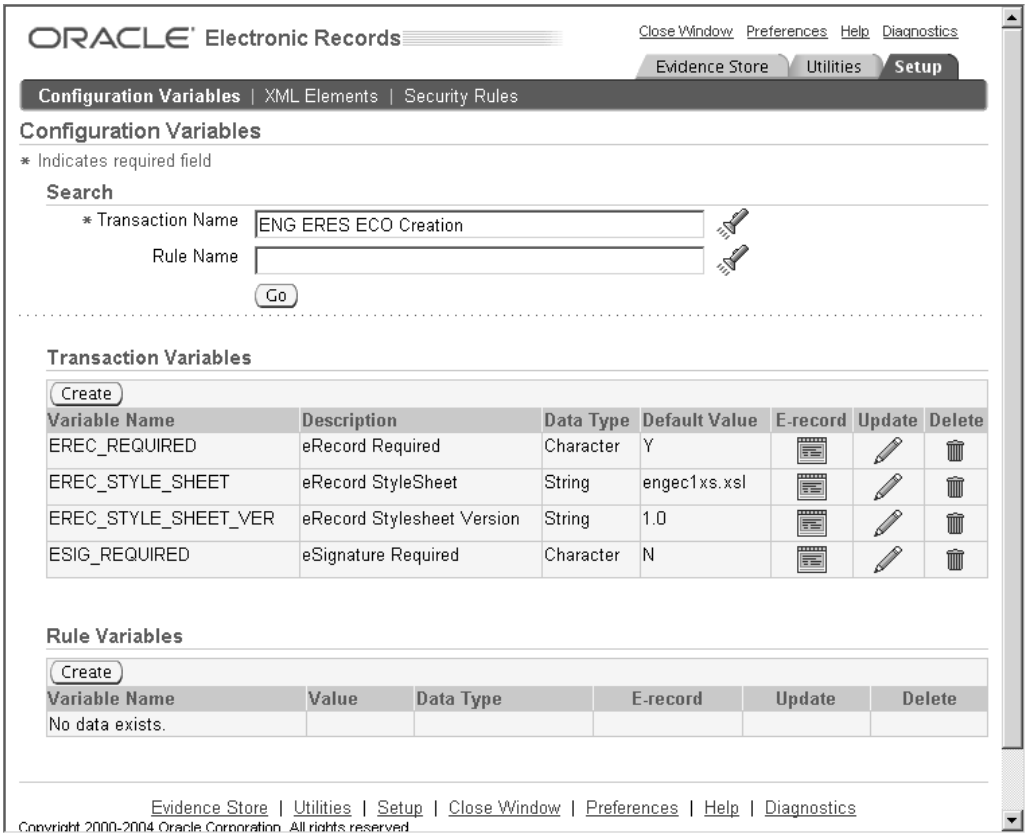
Define the Configuration Variables for the ENG ERES ECO Creation Transaction

11. Navigate to the Configuration Variables page (see: ERES Administrator Windows and Navigation Paths, page A-2). Search for and select the transaction name ENG ERES ECO Creation, then choose Go.

    The transaction variables with their default values appear.

12. Add, change, or delete the transaction and rule variables as needed.

    Use the transaction variable default values for this example.



Enter and submit an ECO for approval

The remaining steps test the scenario set up in the previous steps. The following should occur:

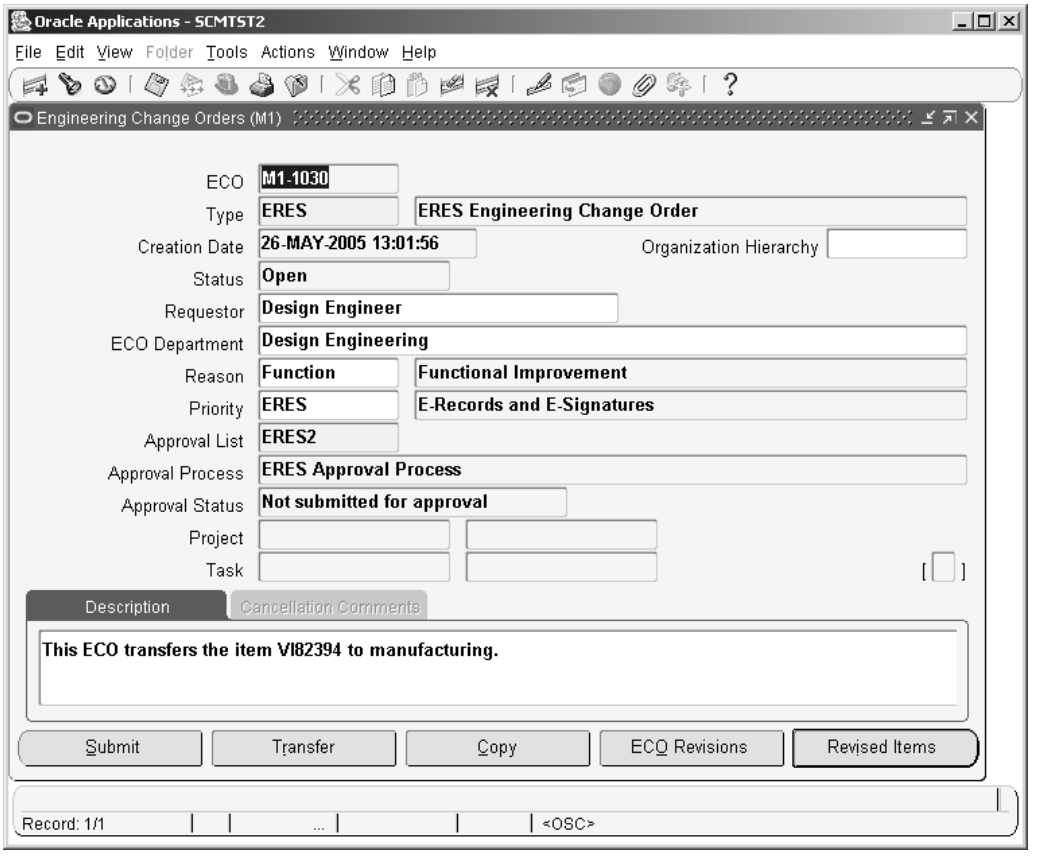
1. An e-record generates when an engineering change order (ECO) is created.
2. An e-record requiring an e-signature is created when an ECO is submitted for approval.

13. Navigate to the Engineering Change Orders window (see: Manufacturing and Distribution Manager Windows and Navigation Paths, page A-3) and enter an ECO (see: Creating an ECO, *Oracle Engineering User's Guide*).

    For this example, enter the following:

    - ECO = M1-1030
    - Type = ERES
    - Requestor = Design Engineer
    - Priority = ERES
    - Approval List = ERES2

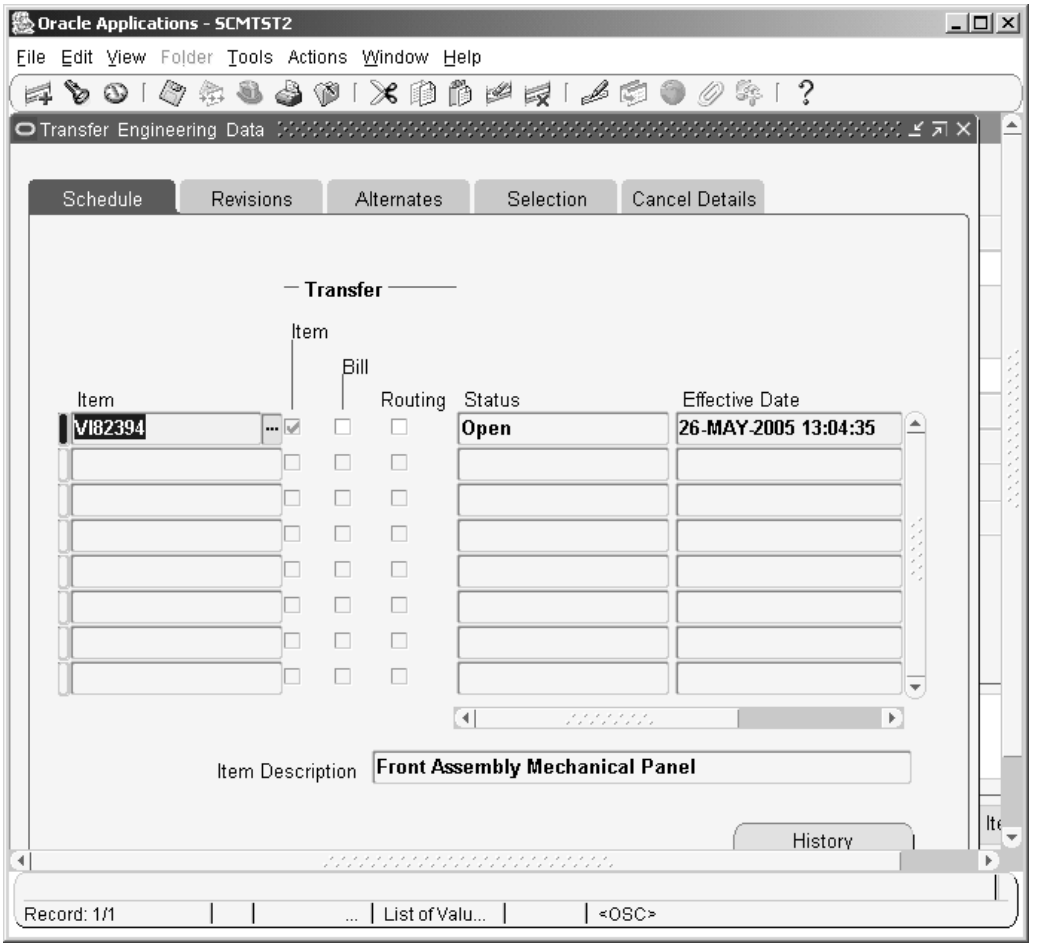
    Notice that the Type and Priority values default the Approval Process of ERES Approval Process. The Approval List is chosen by the requestor of the ECO. For all other transactions, the approval group is automatically applied according to the transaction's applicable rule variables.



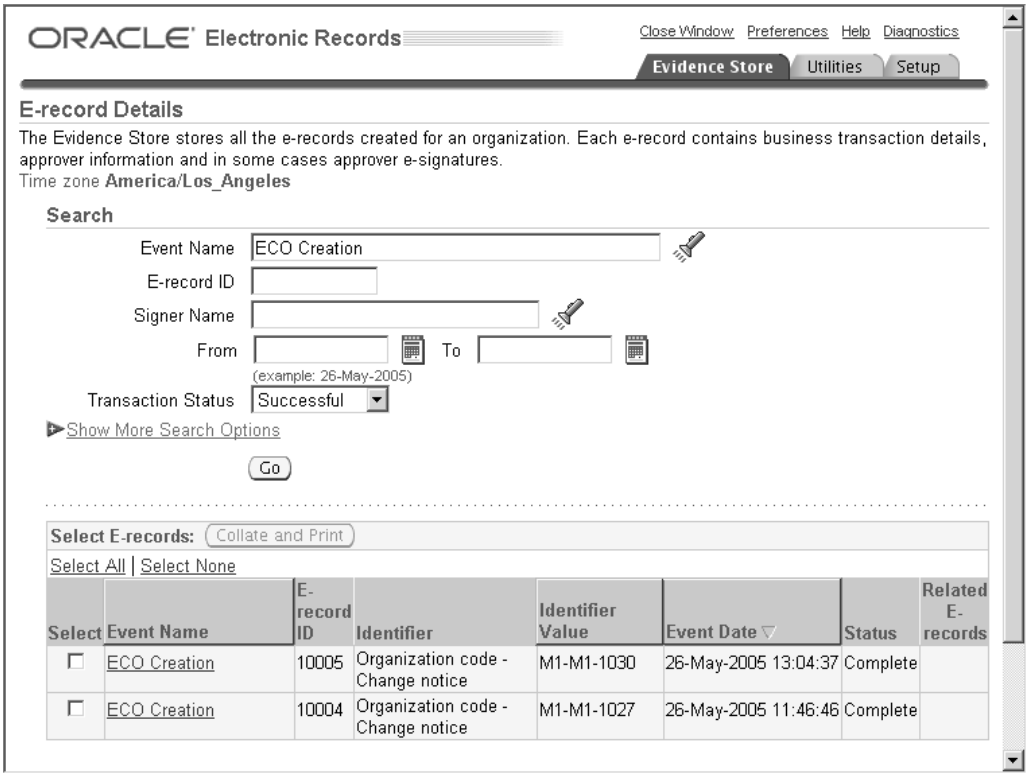
14. For this example, choose Transfer.

    This ECO transfers the item VI82394 to manufacturing.

    Save your work.



15. Optional. Navigate to the Evidence Store. Search for and view the ECO Creation e-record you created.



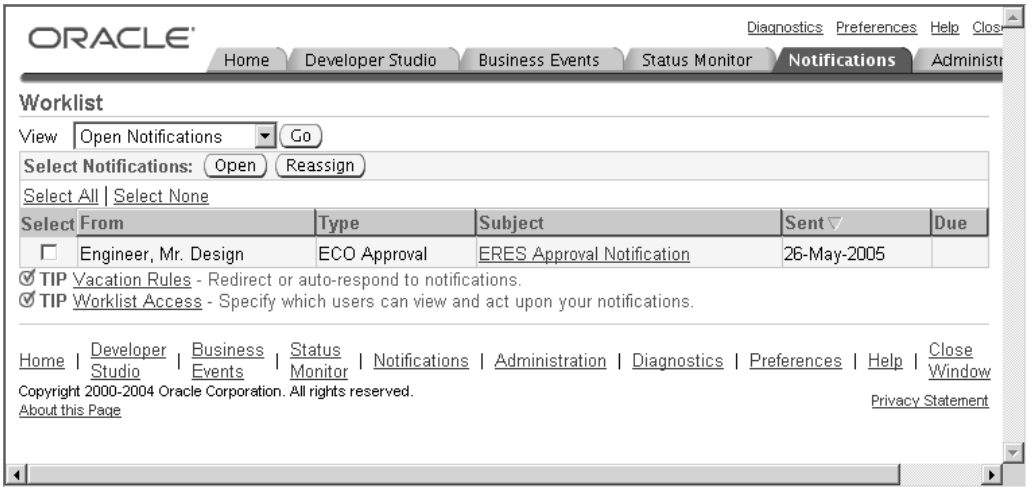
16. Navigate to the ECO window (see: Manufacturing and Distribution Manager Windows and Navigation Paths, page A-3) and find ECO M1-1030.

    Choose Submit.

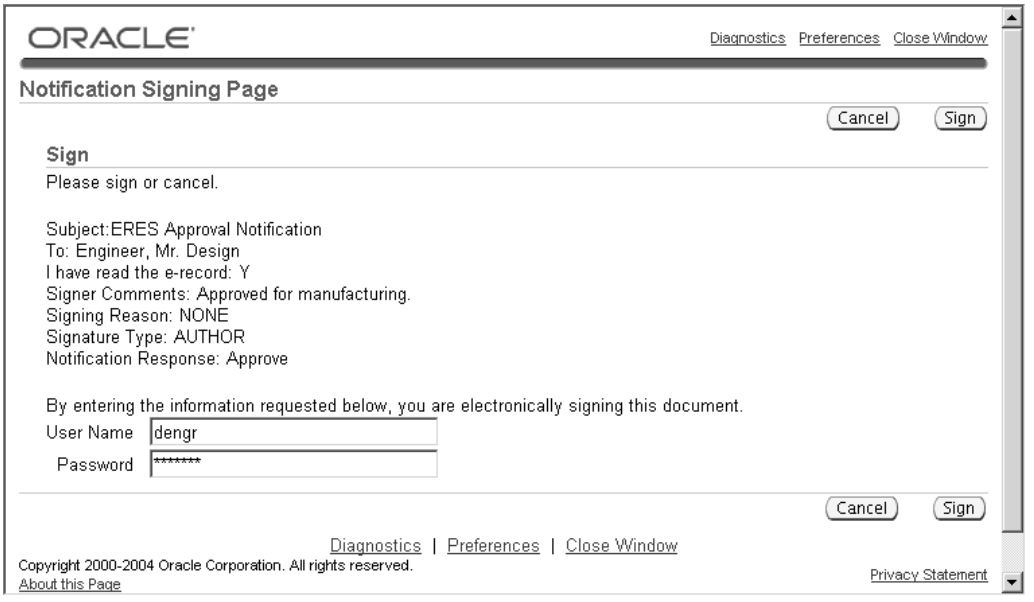
    This submits the ECO for approval. Notice that the Approval Status changes to Approval requested.

Approve the ECO in Oracle Workflow

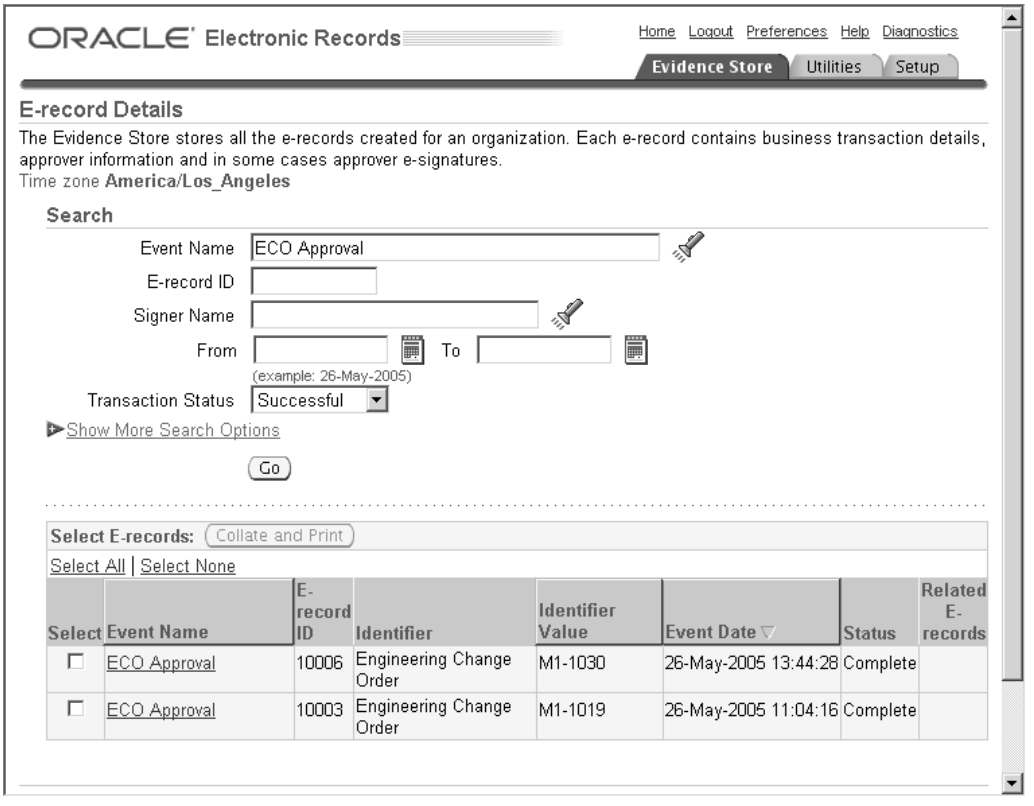
17. Navigate to the Worklist page (see: Workflow Administrator Pages and Navigation Paths, page A-1) . Select the approval notification.



18. Review the approval notification and choose Approve, Reject, or Request Information after entering your response.
19. Sign the approval notification.

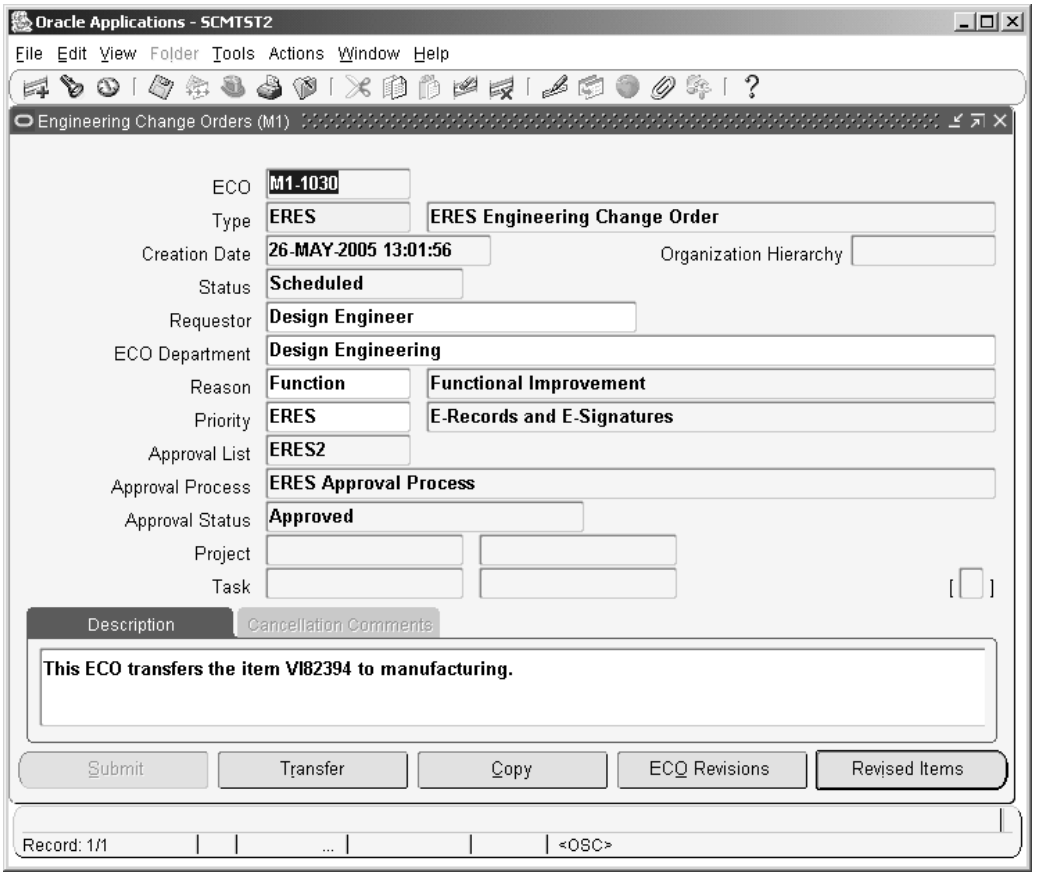


20. Navigate to the Evidence Store page (see: ERES Administrator Windows and Navigation Paths, page A-2). View the e-record once all approvals are obtained (see: Evidence Store, *Oracle E-Records Implementation Guide*).



21. Navigate to the ECO window (see: Manufacturing and Distribution Manager Windows and Navigation Paths, page A-3) and find ECO M1-1030.

    Notice that the Approval Status is now Approved.



Setting Up and Processing Nonconformances, Dispositions, and Corrective Actions:

Nonconformance, disposition, and corrective action collection plans (see: *Oracle Quality Implementation Guide*) require one additional setup step if you intend to collect e-signatures and two exceptions to the standard process of collecting e-records and e-signatures.

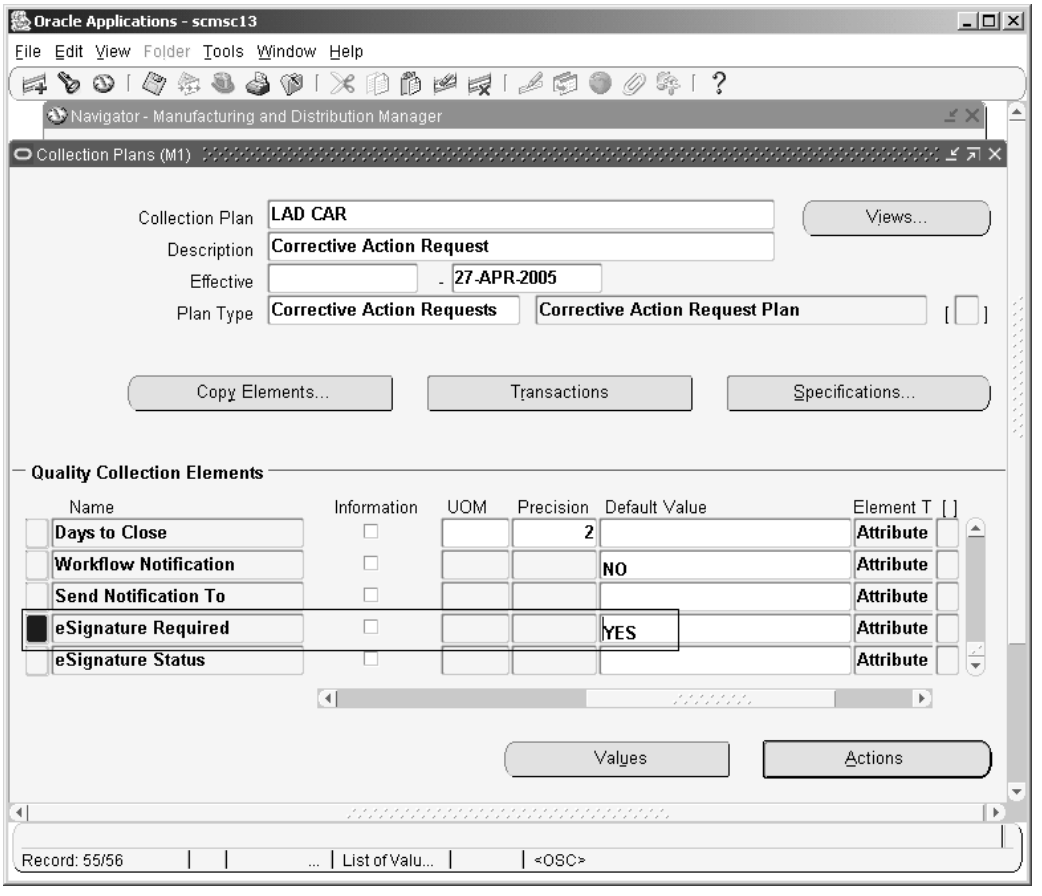
Setup Exception - Setting the eSignature Required Collection Element to Yes

If you intend to collect deferred e-signatures for nonconformance, disposition, or corrective action collection plans, enable, display, and set the eSignature Required collection element to Yes. For information on how to require e-signatures for nonconformance, disposition, and corrective action requests under certain conditions, see: Enforcing Nonconformance, Disposition, and Corrective Action Request Approvals, page 3-20.

1. Navigate to the Collection Plans window (see: Manufacturing and Distribution Manager Windows and Navigation Paths, page A-3).
2. Find the collection plan you want to enable for e-signatures.

3. Scroll down to the eSignature Required and eSignature Status collection elements.
4. Check the Enabled and Displayed boxes for each element.
5. For the eSignature Required element, enter Yes in the Default Value field.

   Enter Yes in the Default Value field only if you always want someone to sign the respective nonconformance, disposition or corrective action record. Otherwise, leave the Default Value field blank.



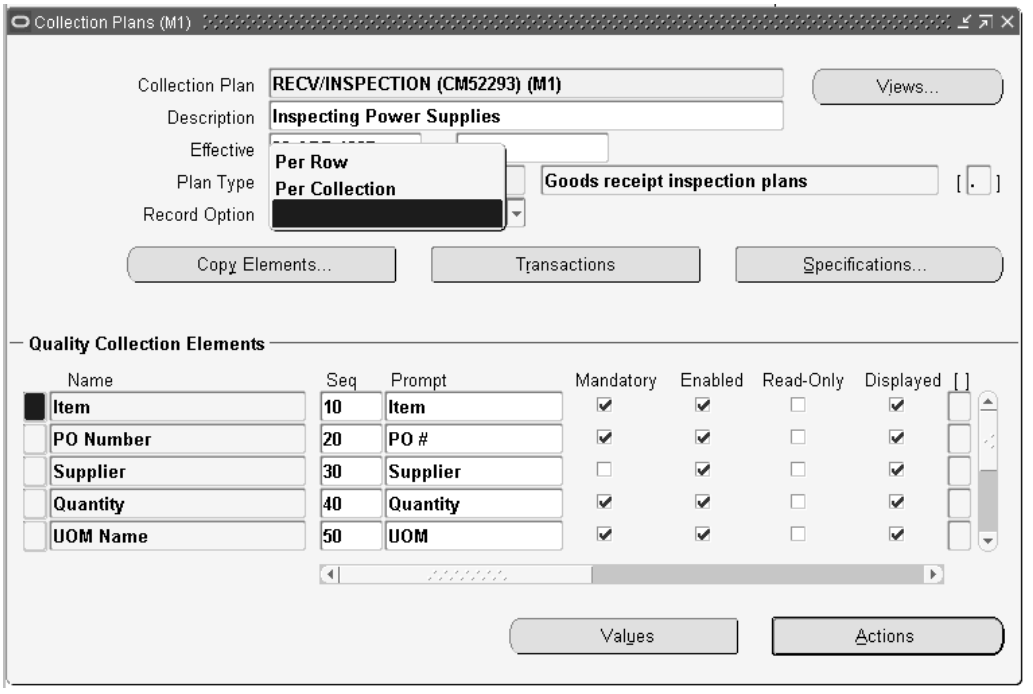
Process Exception - Capturing an E-Record for Nonconformance, Disposition, and Corrective Action Plans

When the system captures an e-record for a nonconformance, disposition, or corrective action collection plan, the e-record includes information from all of the associated child plans, too. Refer to Parent-Child Collection Plans, *Oracle Quality User's Guide* for more information.

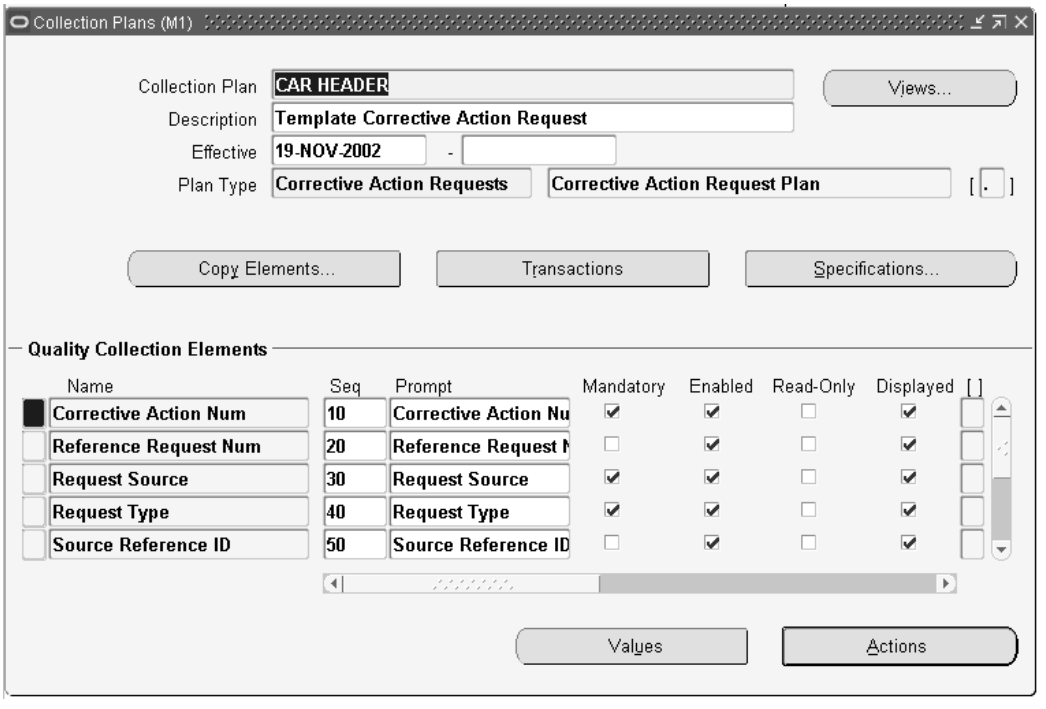
Process Exception - No E-Record Signing Option for Nonconformance, Disposition, and Corrective Action Plans

For nonconformance, disposition, and corrective action request collection plans, you can only enter your e-signature by row, not by collection. For other collection plans, you have the option of signing by row or by collection.

The following picture of the Collection Plans window shows a collection plan with the Record Option field selected:



The following picture of the Collection Plans window shows a corrective action collection plan. Notice that the Record Option field is not visible. The Record Option field is not visible for nonconformance, disposition, and corrective action request collection plans.



Capturing E-Records for Shipments:

Oracle Shipping generates e-records using a concurrent program. This concurrent program, Quality Shipping ERES Collector, generates e-records for all outbound deliveries with a status of Closed (see: Closing a Delivery, *Oracle Shipping Execution User's Guide*). You can run this program as needed or your system administrator can set it up to run on a periodic basis (for example, nightly). Generate e-records for a delivery date range by entering dates into the program's parameters. You can view the e-record in the Evidence Store.

Approving and Rejecting Miscellaneous Transactions:

The Oracle Inventory Miscellaneous Transactions window (see: Performing Miscellaneous Transactions, *Oracle Inventory User's Guide*) can contain many line items. If you approve (or reject) all line items during the e-signature process, you can save all line items at once. However, if you want to reject some line items and approve the rest, you must perform the rejection and the approval in separate actions. For example, approve some of the line items, then open the Miscellaneous Transactions window again, search for the line items to reject, and reject them.

Related Topics

Setting Up Oracle Approvals Management, *Oracle E-Records Implementation Guide*

Setting Up the Configuration Variables, *Oracle E-Records Implementation Guide*

3

Implementation Considerations

This chapter discusses strategies for you to consider using during Oracle E-Records implementation.

This chapter covers the following topics:

- Using Additional Attributes
- Modifying a Stylesheet to Include Descriptive Flexfields
- Enforcing Nonconformance, Disposition, and Corrective Action Request Approvals
- Capturing Non-Seeded Collection Elements in Nonconformance, Disposition, and Corrective Action E-Records for History Collection Plans
- Creating a Device History Record

Using Additional Attributes

Oracle provides seeded attributes for each discrete manufacturing transaction, but you may have special circumstances that require the addition of a new attribute. The following example explains in detail how to define and use a new attribute.

Example of Using a Newly Defined Attribute:

Manufacturing plant M1 requires an e-record for each bill of material created for an engineering item. Bills of material created for manufacturing items require both an e-record and an e-signature, though. Use the Eng Item field value in the Bills of Material window to determine if an e-signature is necessary.

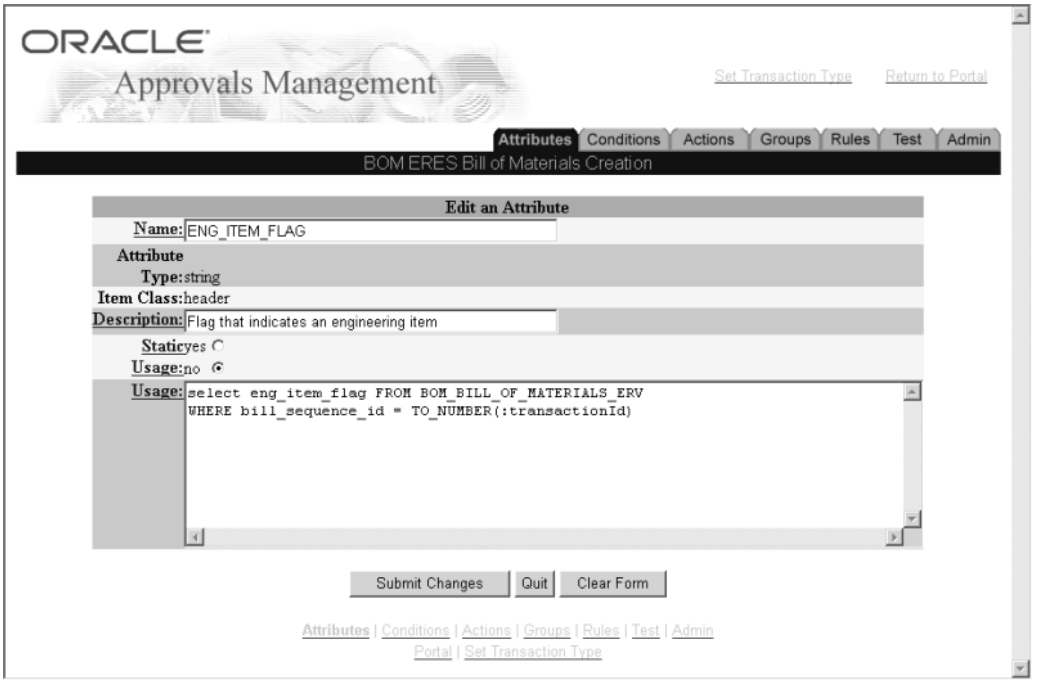
The following steps describe how to define a new attribute and use it to enforce the requirement explained above. For a more detailed setup example, see: Setting Up E-Records: A Discrete Manufacturing Example, page 2-1.

1. Navigate to Oracle Approvals Management (see: Approvals Management Application Administrator Windows and Navigation Paths, page A-2).
2. Set the Transaction Type to BOM ERES Bill of Materials Creation.
3. Select the Attributes tab.

   Review the list of attributes to see if the attribute you need exists. If not, choose Add Attribute.
4. Use the following values when adding the attribute:
   - Item Class = header

- Name = ENG_ITEM_FLAG
- Attribute Type = string
- Description = Flag that indicates an engineering item
- Static Usage = no
- Usage = select eng_item_flag FROM BOM_BILL_OF_MATERIALS_ERV WHERE bill_sequence_id = TO_NUMBER(:transactionID)

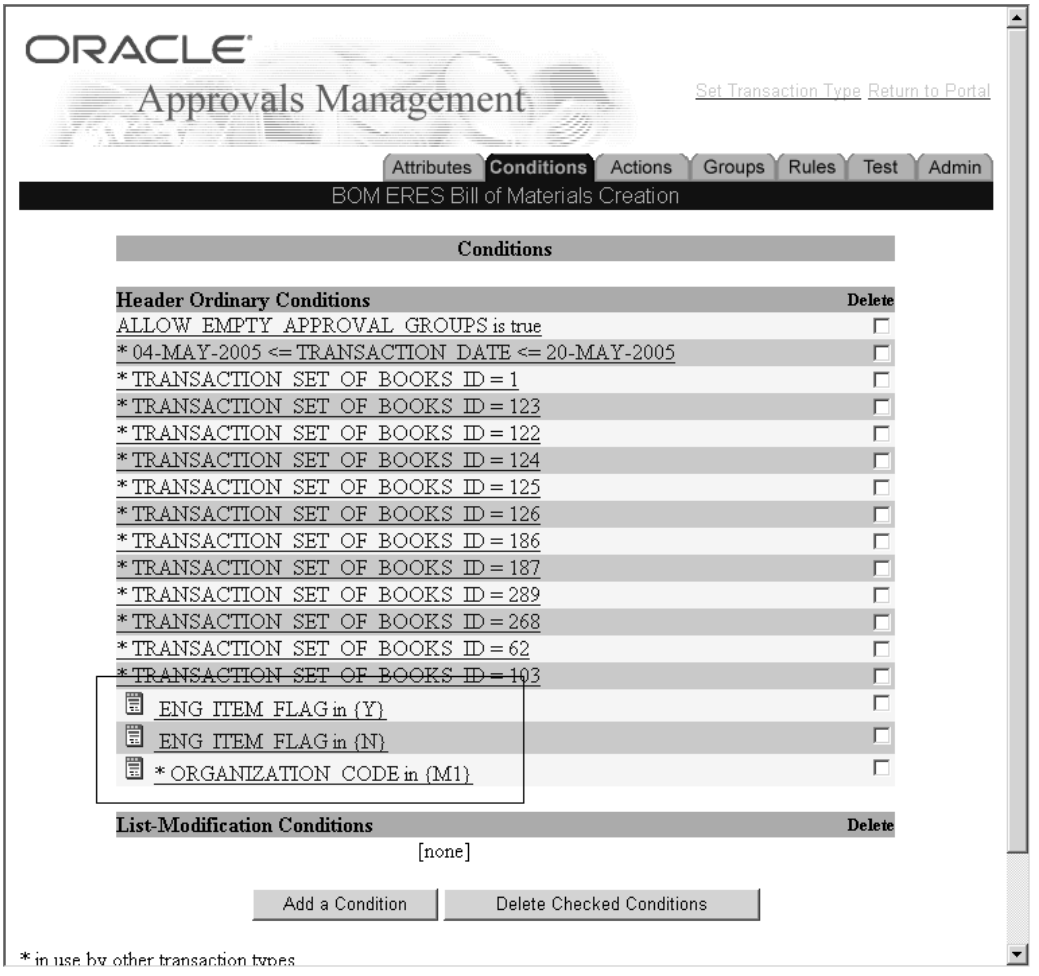
Note: This SQL string selects the Eng Item field value.



5. Select the Conditions tab.

   Create two new conditions using the following values. A condition for Organization Code = M1 already exists.

Field Name	Condition 1 Field Values	Condition 2 Field Values
Attribute	ENG_ITEM_FLAG	ENG_ITEM_FLAG
Attribute Value	Y	N



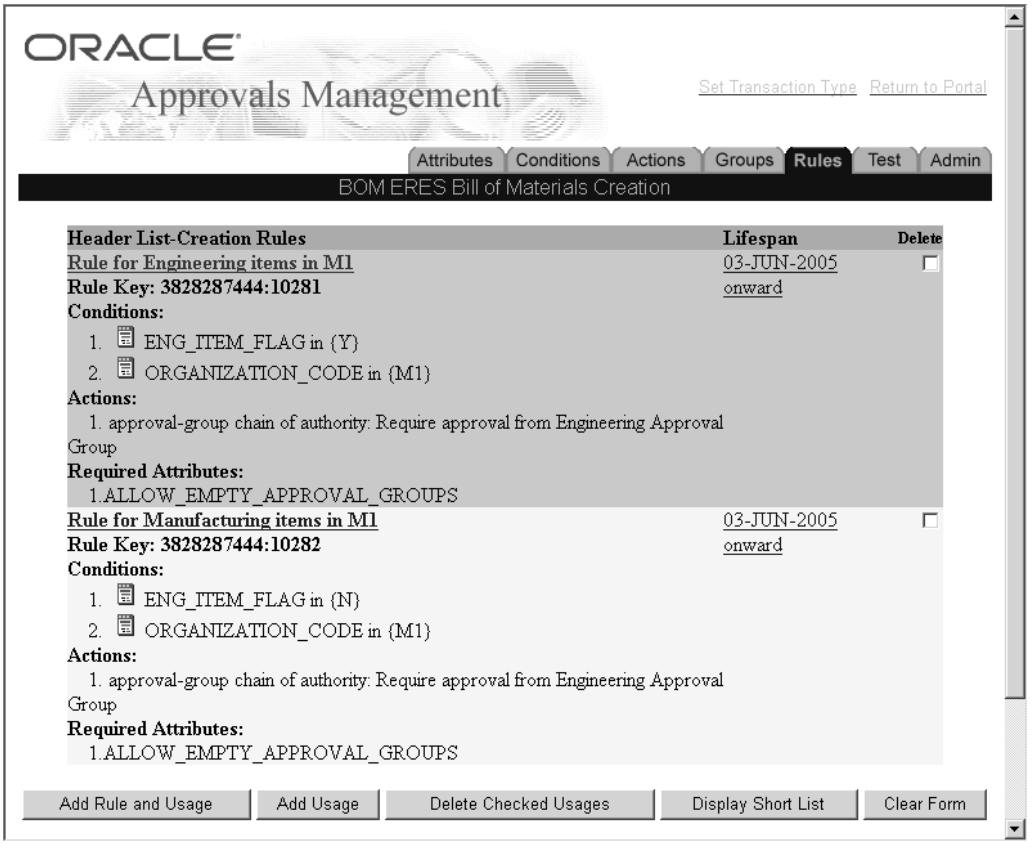
6. Select the Rules tab.

   Add two new rules using the following values.

   - If no field value is given in the table below, then use the field's default value.
   - These rules use a predefined approval group, Engineering Approval Group. For more information about creating Approval Groups, see: Approval Groups, *Implementing Oracle Approvals Management*, at http://metalink.oracle.com. Search on Metalink Note #282529.1.

Field Name	Rule 1 Field Values	Rule 2 Field Values
Description	Rule for Engineering items in M1	Rule for Manufacturing items in M1
Action Types	chain of authority includes an approval group	chain of authority includes an approval group
approval-group chain of authority Actions	Require approval from Engineering Approval Group	Require approval from Engineering Approval Group
Header Attributes	ENG_ITEM_FLAG ORGANIZATION_CODE	ENG_ITEM_FLAG ORGANIZATION_CODE
Header Conditions	ENG_ITEM_FLAG in {Y} ORGANIZATION_CODE in {M1}	ENG_ITEM_FLAG in {N} ORGANIZATION_CODE in {M1}

Note: When defining an approval rule, you must specify an approval group, even if you do not need to collect e-signatures (approvals) for the transaction. Oracle E-Records ignores the approval group if the configuration or variable rule does not require an e-signature.



Define the Rule Variables for the BOM ERES Bill of Materials Creation Transaction

7. Navigate to the Configuration Variables page (see: ERES Administrator Windows and Navigation Paths, page A-2). Search on the BOM ERES Bill of Materials Creation transaction.

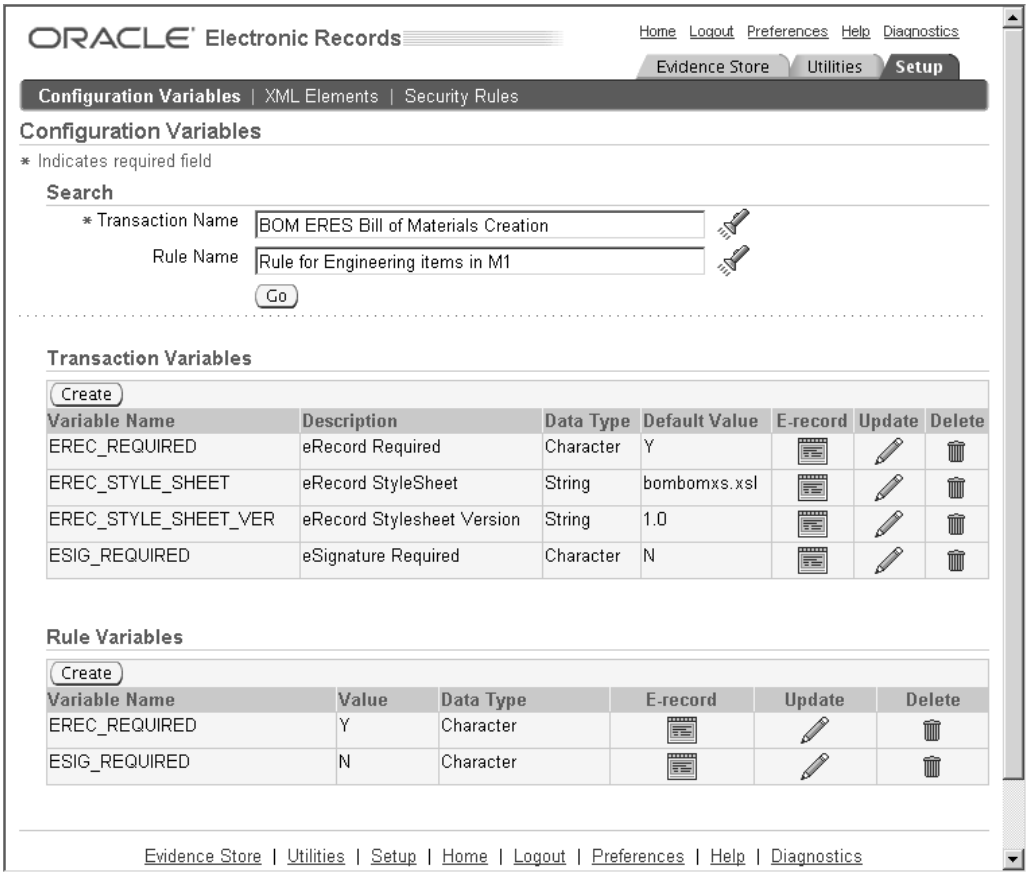
   The default values for the transaction variables follow:

   - EREC_REQUIRED = Y
   - EREC_STYLE_SHEET = bombomxs.xsl
   - EREC_STYLE_SHEET_VER = 1.0
   - ESIG_REQUIRED = N

8. Search on the BOM ERES Bill of Materials Creation transaction with the Rule for Engineering items in M1 rule.

9. Create two rule variables:

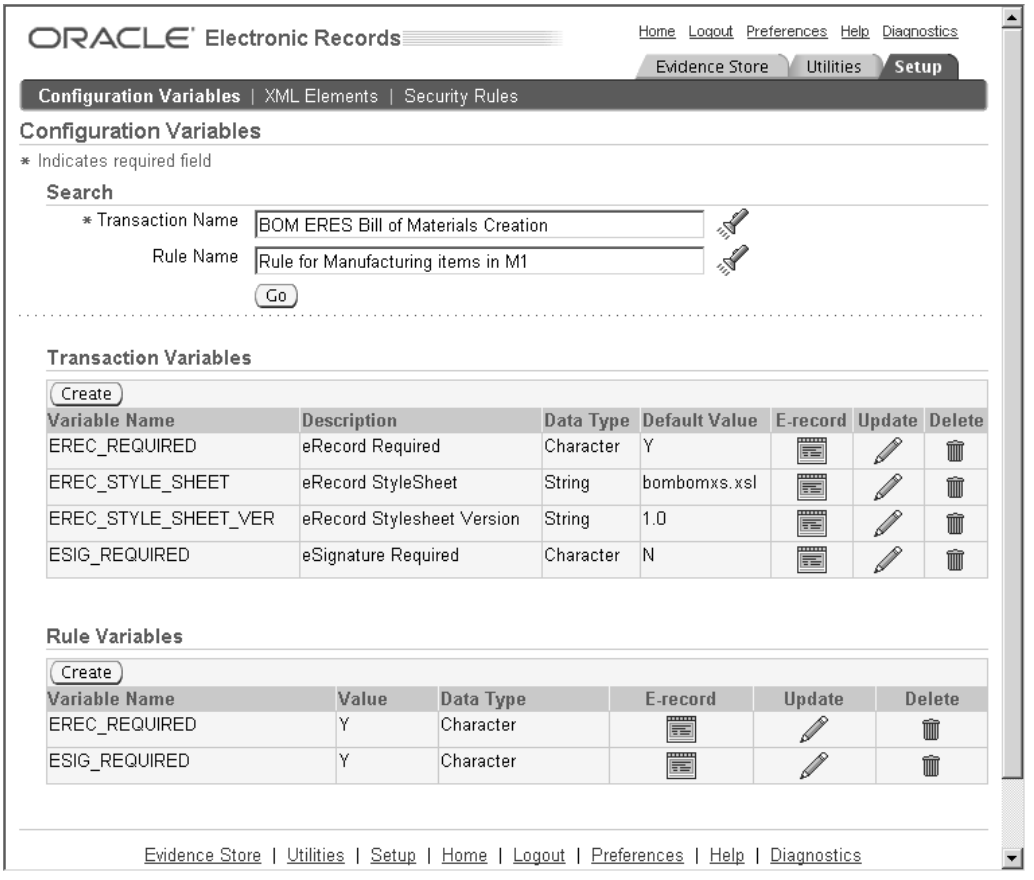
   - EREC_REQUIRED = Y
   - ESIG_REQUIRED = N



Note: Since the transaction variable default values for EREC_REQUIRED and ESIG_REQUIRED are the same, it is not necessary to create the above two rule variables. However, if the transaction variable default values were ever to change, this

guarantees that the variables applied for the rule conditions remain the same.

10. Search on the BOM ERES Bill of Materials Creation transaction with the Rule for Manufacturing items in M1 rule.
11. Create two rule variables:
    - EREC_REQUIRED = Y
    - ESIG_REQUIRED = Y

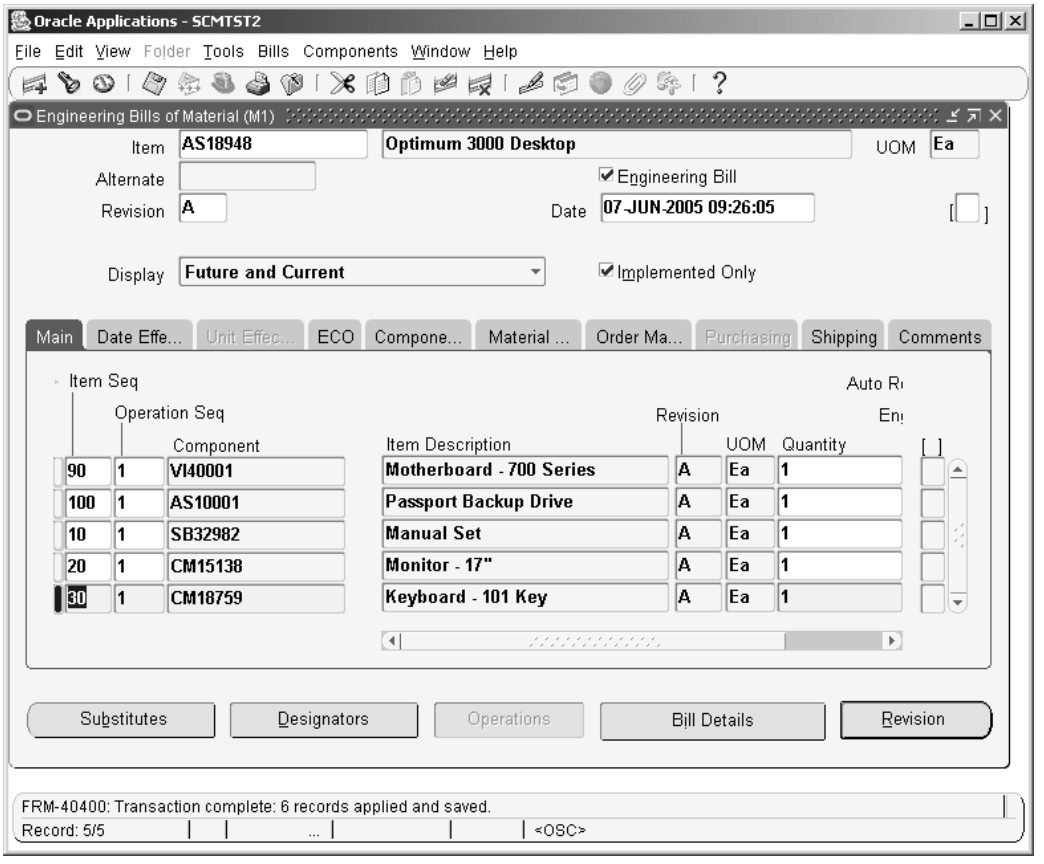


Enter an Engineering Bill of Material

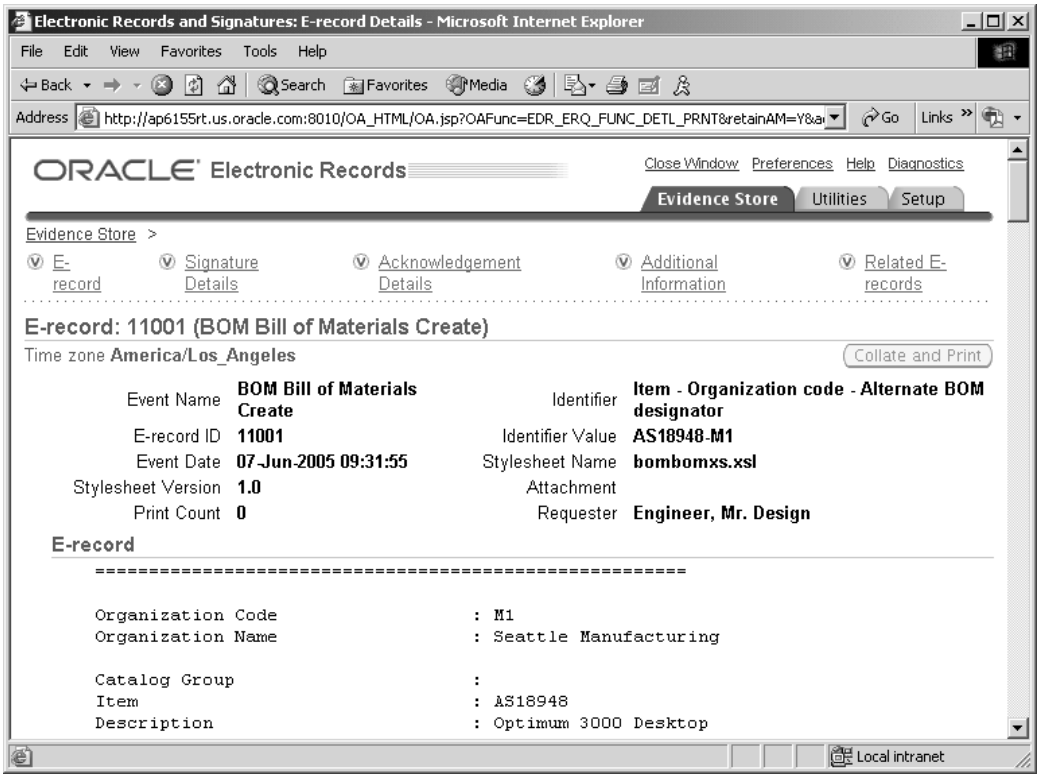
12. Navigate to the Engineering Bills of Material window (see: Manufacturing and Distribution Manager Windows and Navigation Paths, page A-3).

    Enter a bill of material for an engineering (prototype) item.

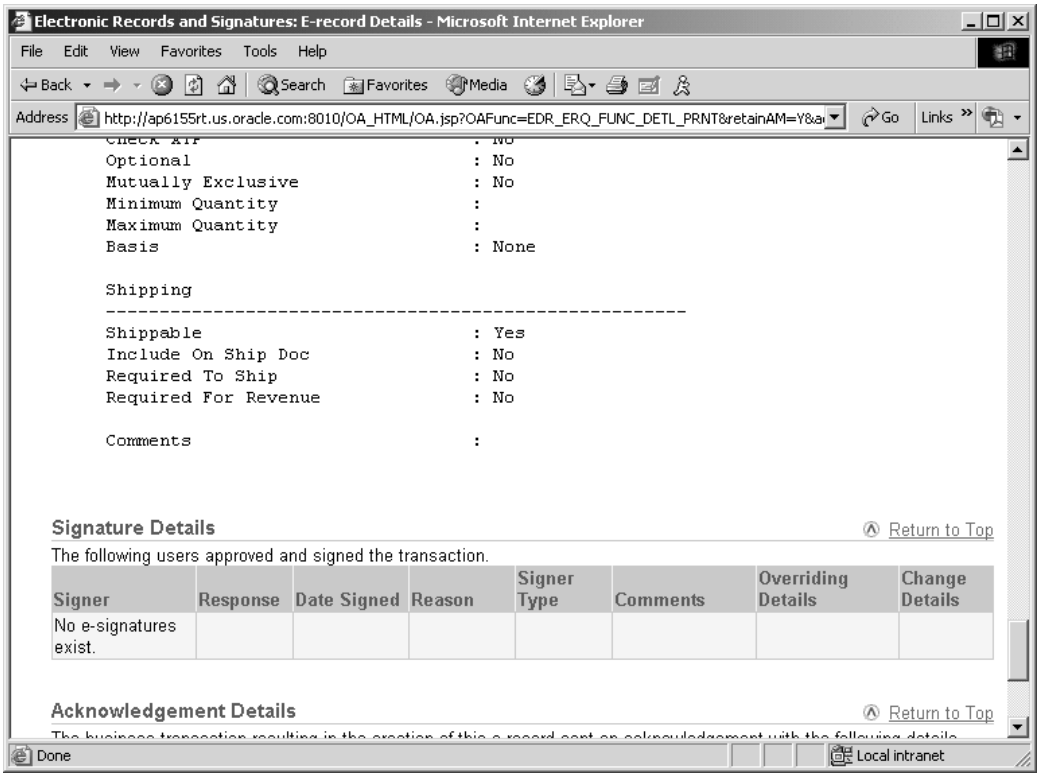
    Note: Verify that an item is an engineering item by ensuring that the Engineering Item check box in the Engineering Item window, Bills of Material tab, is checked.



13. Optional. Navigate to the Evidence Store (see: ERES Administrator Windows and Navigation Paths, page A-2). Search for and view the ECO Creation e-record you created.



Notice that this e-record does not include e-signatures.

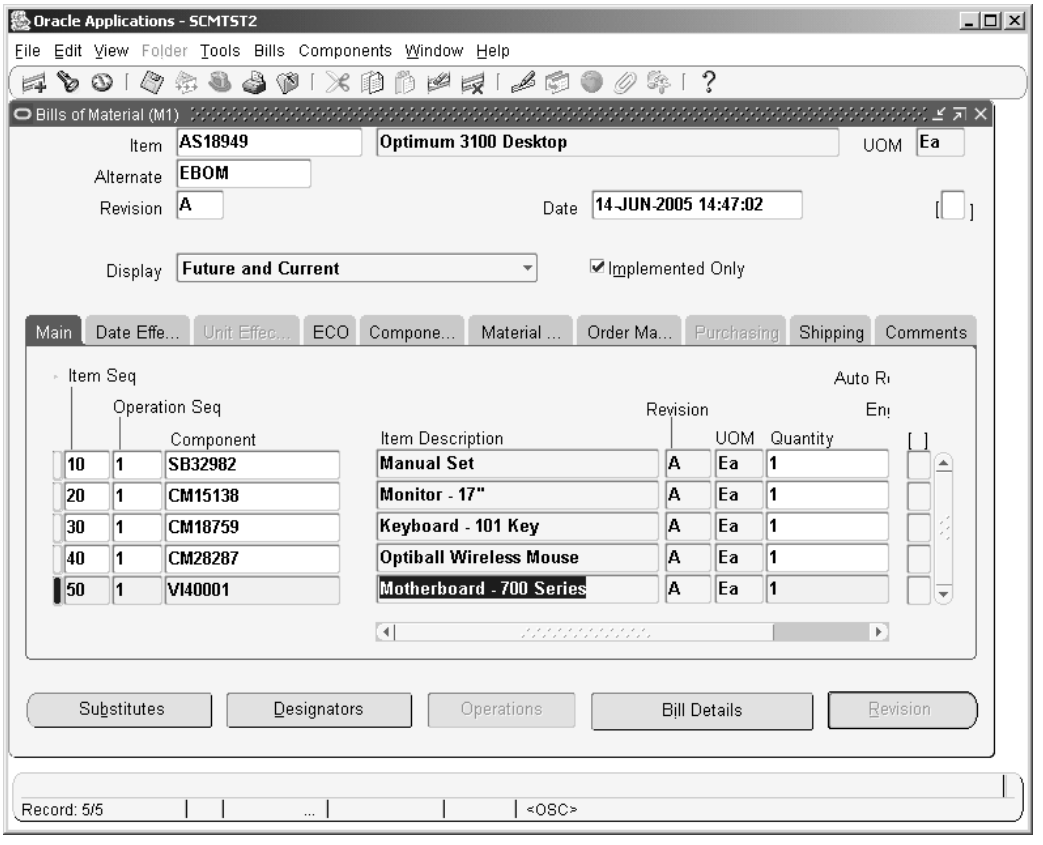


Enter a Manufacturing Bill of Material

14. Navigate to the Bills of Material window (see: Manufacturing and Distribution Manager Windows and Navigation Paths, page A-3).

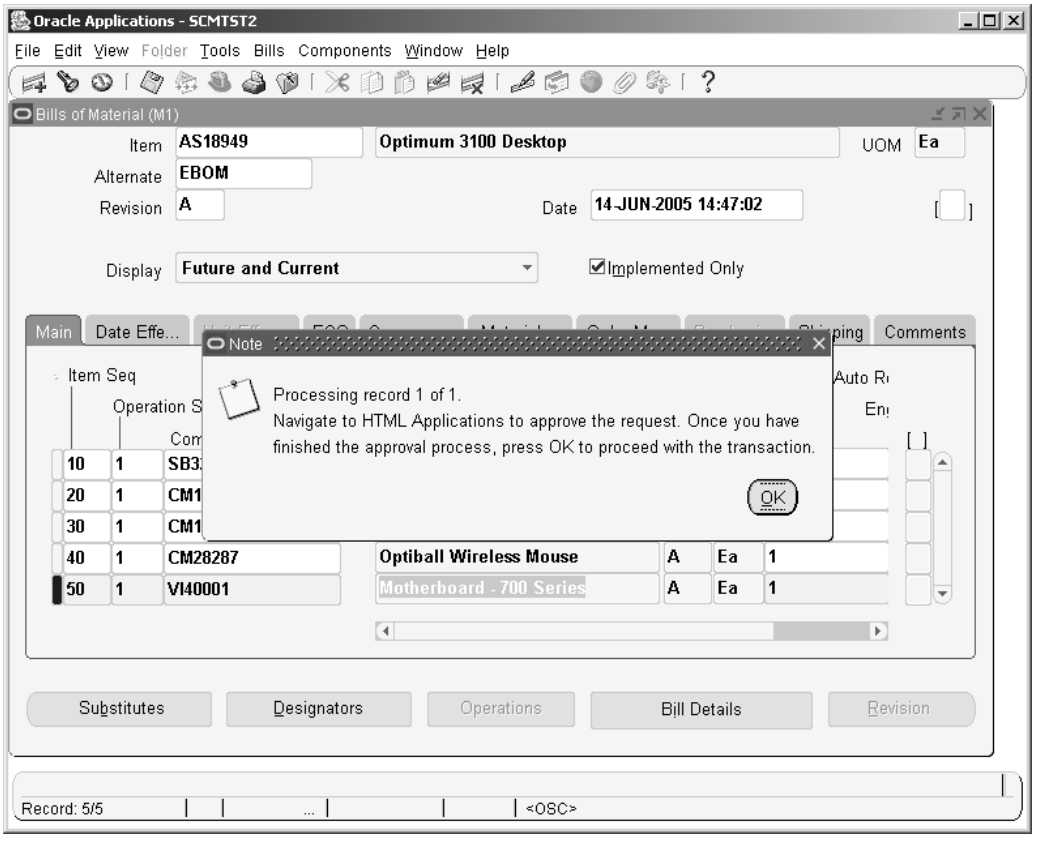
    Enter a bill of material for a manufacturing item.

    Note: Verify that an item is a manufacturing item by ensuring that it exists in the Organization Item window.



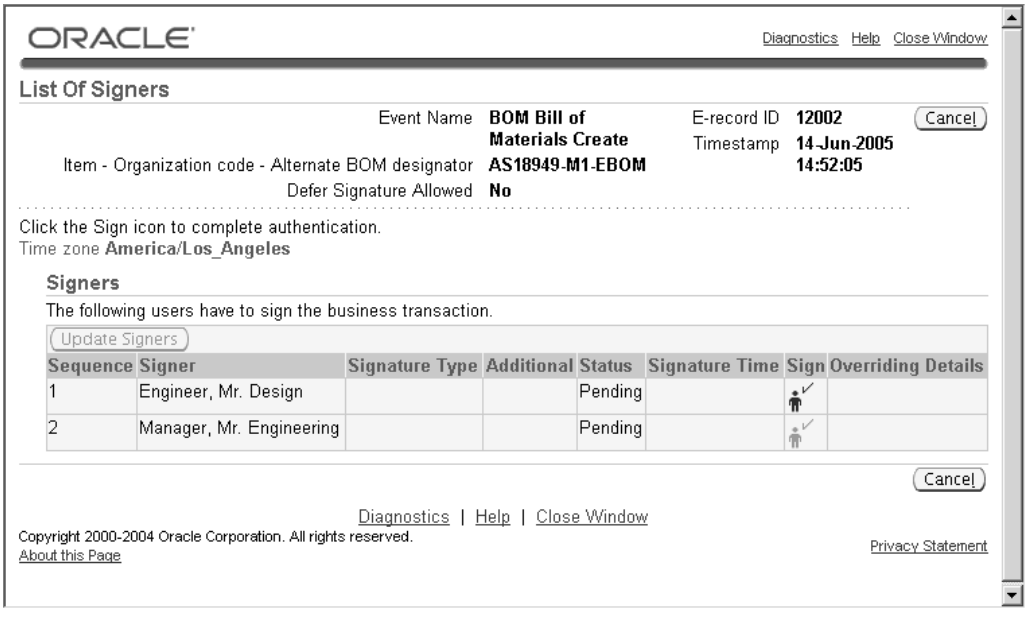
The following online e-signature request appears:

Note: Navigate to HTML Applications to approve the request. Once you have finished the approval process, press OK to proceed with the transaction.



15. The List of Signers page opens.

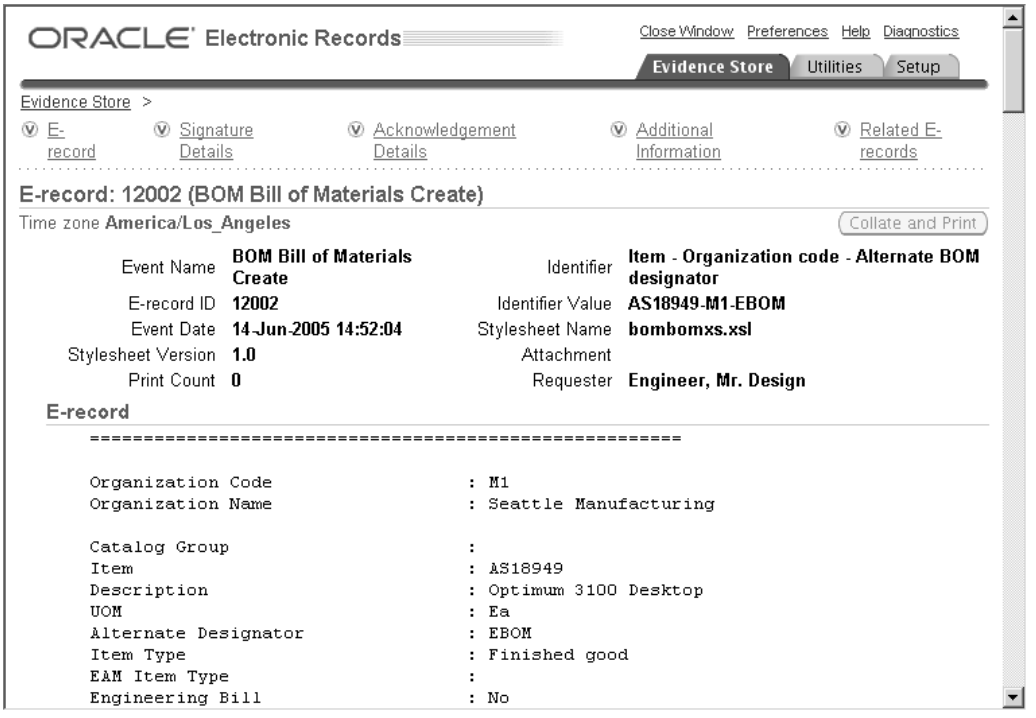
    Obtain all required e-signatures.



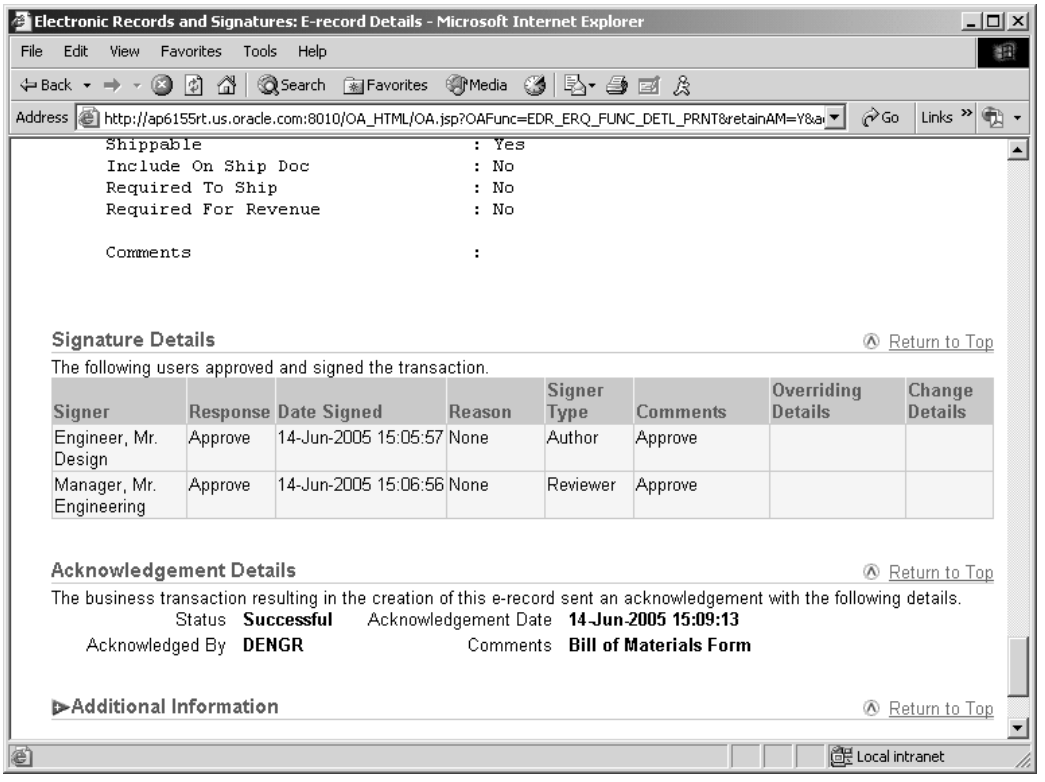
16. Return to the Bills of Material window. Choose OK to proceed with the transaction.

The transaction is saved.

17. Optional. Navigate to the Evidence Store (see: ERES Administrator Windows and Navigation Paths, page A-2). Search for and view the BOM Bill of Materials Create e-record you created.



Notice that this e-record contains e-signatures.



Modifying a Stylesheet to Include Descriptive Flexfields

You can modify the standard stylesheets provided with Oracle E-Records if you want to include descriptive flexfield values in your e-records. The following example adds a descriptive flexfield to the Collection Plans window, then modifies the stylesheet for the QA ERES Collection Plan Creation transaction so that the e-record displays the descriptive flexfield value.

Example:

Create a Descriptive Flexfield

1. Navigate to the Descriptive Flexfield Segments window (see: Application Developer Windows and Navigation Paths, page A-3).

   In this example, define a descriptive flexfield for the Collection Plans window using the following values:

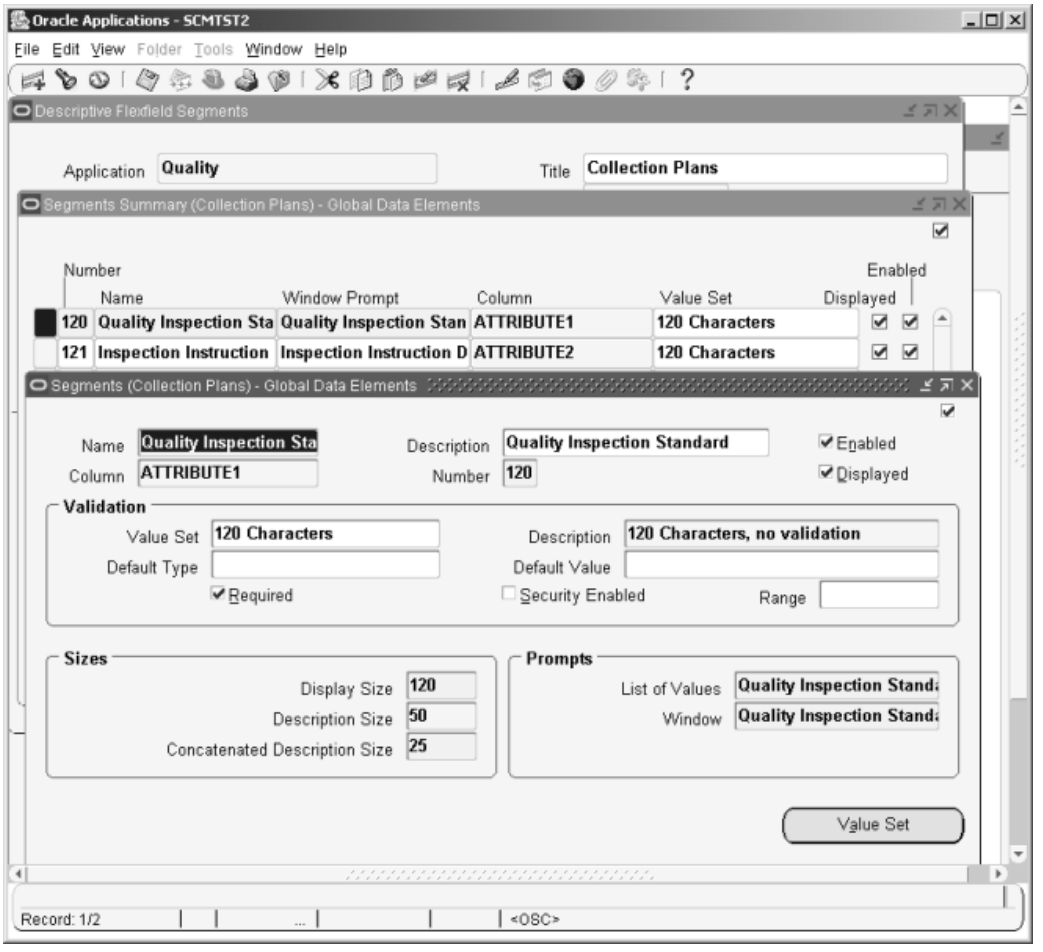
   - Application = Quality
   - Title = Collection Plans

   See Defining Descriptive Flexfield Structures, *Oracle Applications Flexfields Guide*, for more information about setting up descriptive flexfields.

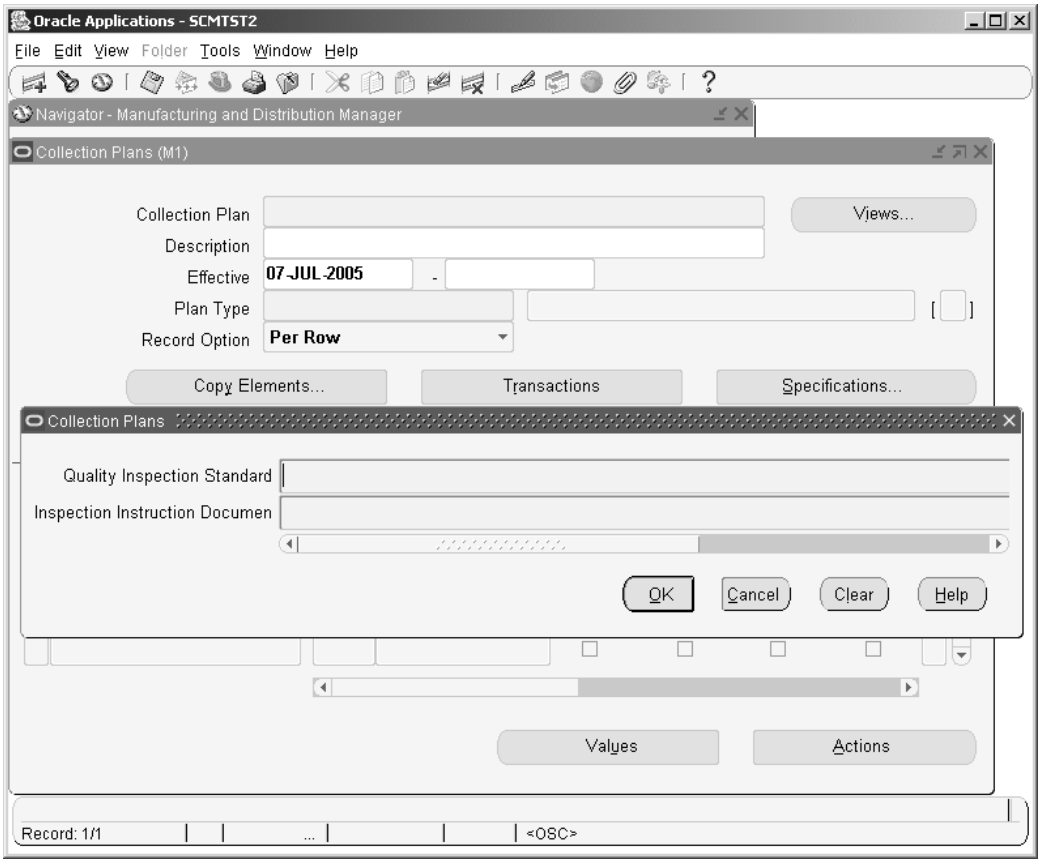
2. Choose Segments, then New.

   In the Segments window, define the descriptive flexfield using the following values:

Number	Name	Description	Column	Value Set
120	Quality Inspection Standard	Quality Inspection Standard	ATTRIBUTE1	120 Characters
121	Inspection Instruction Document	Inspection Instruction Document	ATTRIBUTE2	120 Characters



3. Navigate to the Collection Plans window (see: Manufacturing and Distribution Manager Windows and Navigation Paths, page A-3). Verify that the new descriptive flexfield segments, Quality Inspection Standard and Inspection Instruction Document, exist.



Modify the Stylesheet

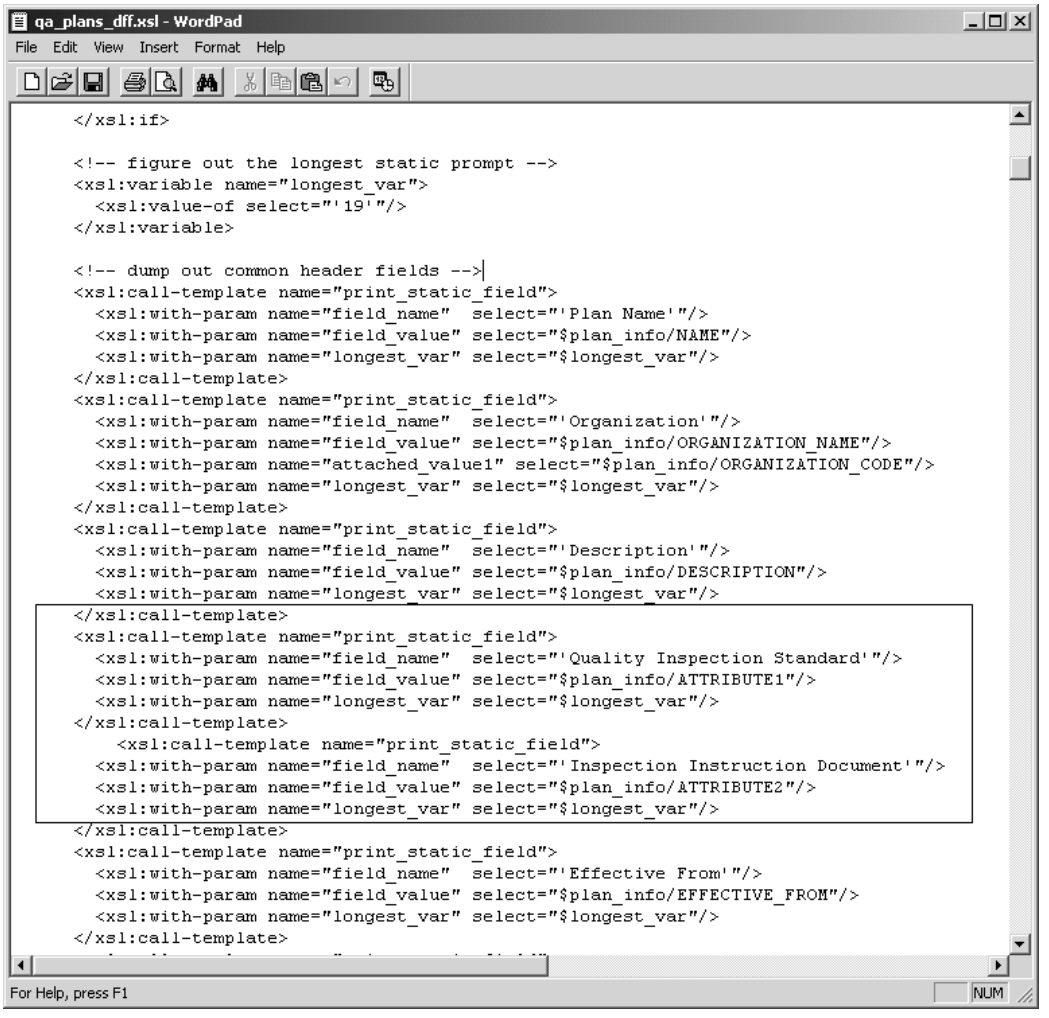
4. Locate and download the stylesheet you want to modify.

   The stylesheet for the QA ERES Collection Plan Creation transaction provided with Oracle E-Records is named qa_plans.xsl.

   > **Tip:** Download the provided stylesheets from $APPL_TOP/<product directory>/11.5.0/html. $APPL_TOP is an environment variable and depends on your environment. In this example, download qa_plans.xsl from $APPL_TOP/qa/11.5.0/html.

5. Use a text editor program, such as Microsoft Notepad or WordPad, to change the stylesheet.

   In this example, add two new lines, between Description and Effective From, named Quality Inspection Standard and Inspection Instruction Document. Select the ATTRIBUTE1 and ATTRIBUTE2 columns since these are the columns where the descriptive flexfield segments are mapped.



6. Save the stylesheet with a new name.

   In this example, the new name is qa_plans_dff.xsl.

Upload the New Stylesheet

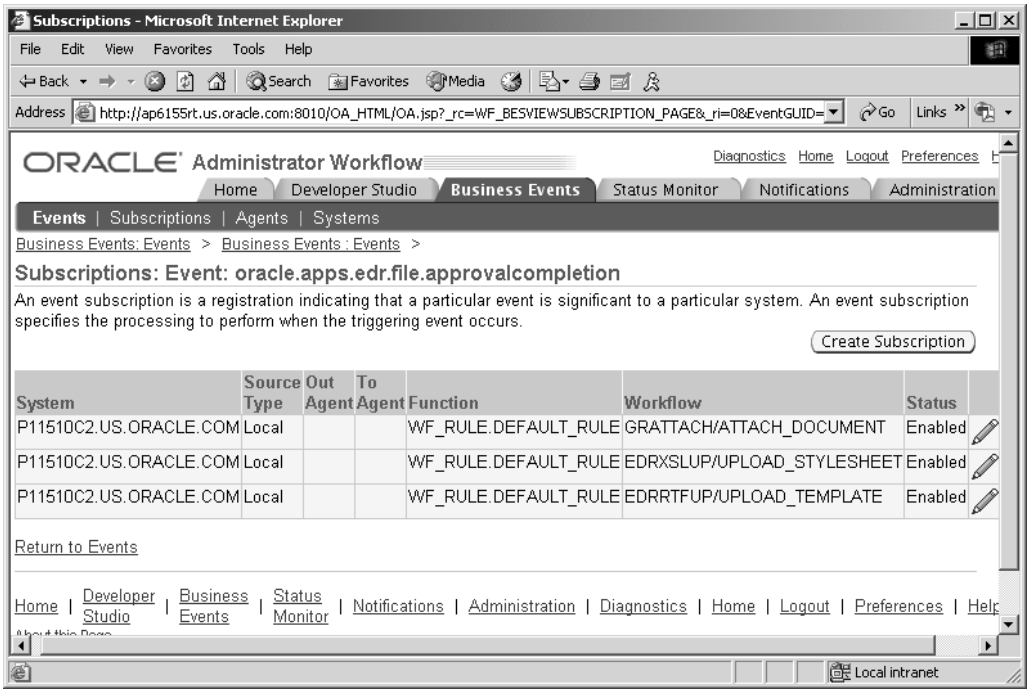
7. Navigate to the Subscriptions page (see: Workflow Administrator Pages and Navigation Paths, page A-1).

   To upload the new stylesheet, you must enable the subscriptions to two business events:

   - ERES File Approval Event (oracle.apps.edr.file.approve)
   - EDR File Approval Complete (oracle.apps.edr.file.approvalcompletion)

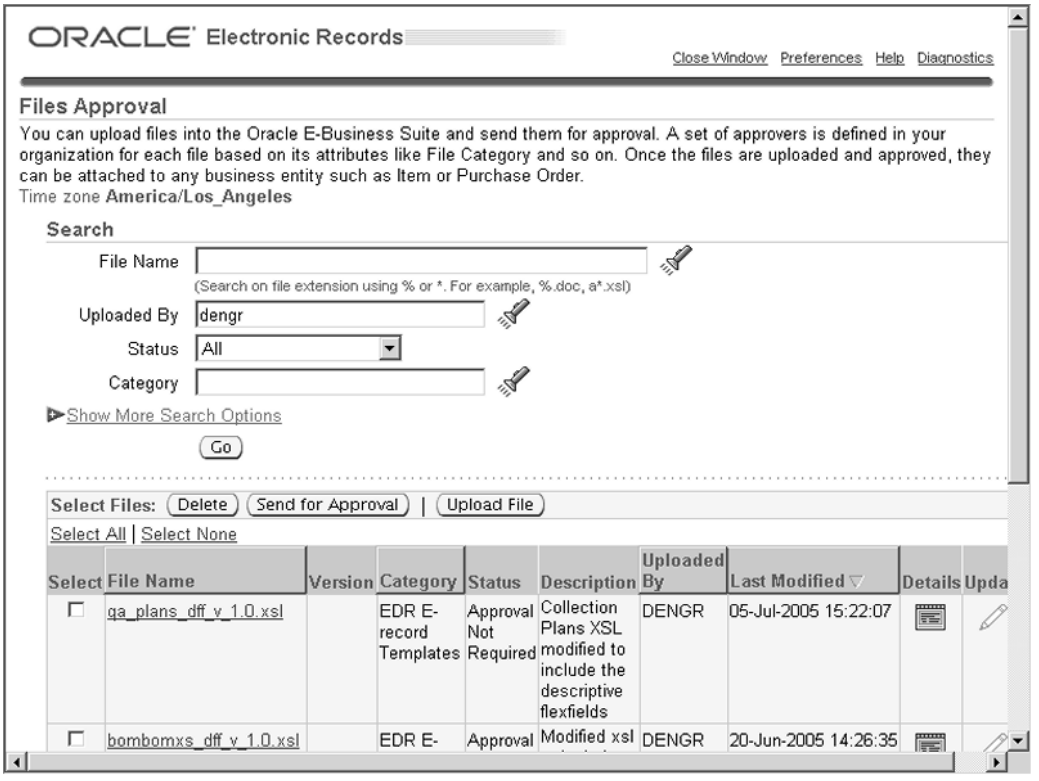
   For the EDR File Approval Complete event, enable the subscription to the UPLOAD_TEMPLATE workflow process.

   For the ERES File Approval Event, enable the subscription to the PSIG_ESIGN_PAGE_FLOW workflow process.



8. Navigate to the Files Approval page (see: Approvals Management Application Administrator Windows and Navigation Paths, page A-2). Choose Upload File.
9. On the File Upload page, select the following field values for this example:
   - Category = EDR E-record Templates
   - Product = QA
   - Template Type = XSL
   - File = qa_plans_dff.xsl
   - If File Exists = Version Existing File
   - Version Label = 1.0
   - Description = Collection Plans XSL modified to include the descriptive flexfields
10. Choose Apply.

    You can now view the new file, qa_plans_dff_v_1.0.xsl, in the Files Approval page. Notice the status of New.



11. Select the row containing qa_plans_dff_v_1.0.xsl, then choose Send for Approval.

    Notice that the status changes to Approval Not Required.

    Tip: If you require an approval for the new stylesheet, you must set up an approval rule for the EDR ERES File Approval transaction in Oracle Approvals Management. See Setting Up E-Records: A Discrete Manufacturing Example, page 2-1, for an example of how to set up an approval.

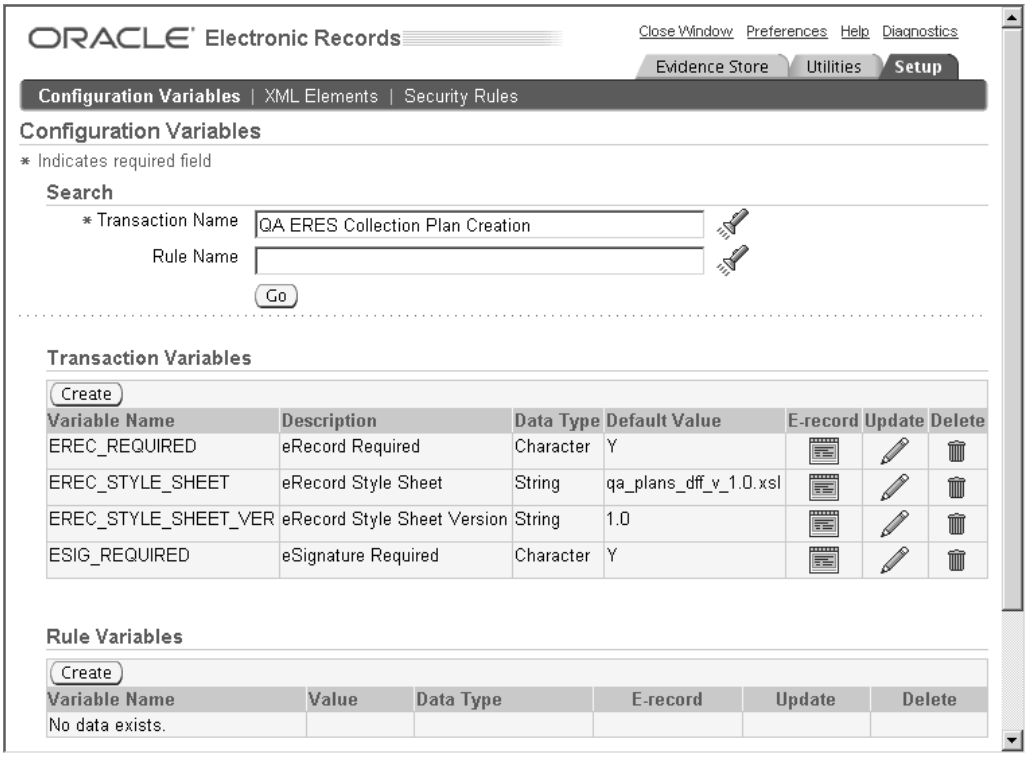
Link the New Stylesheet to a Transaction

12. Navigate to the Configuration Variables page (see: ERES Administrator Windows and Navigation Paths, page A-2). Search for the transaction using the new stylesheet.

    For this example, find the configuration variables for the QA ERES Collection Plan Creation transaction.

13. Update the variable EREC_STYLE_SHEET with the new stylesheet.

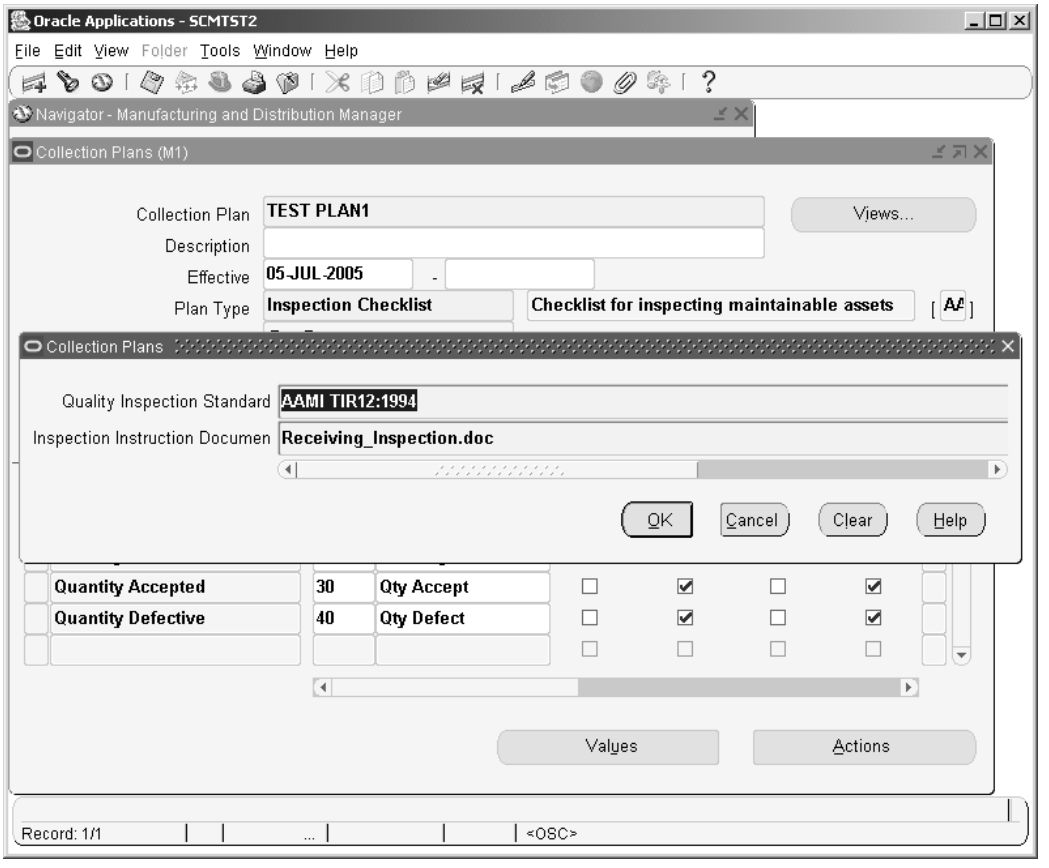
    Enter qa_plans_dff_v_1.0.xsl in the Default Value field.



Create an E-record Using the New Stylesheet

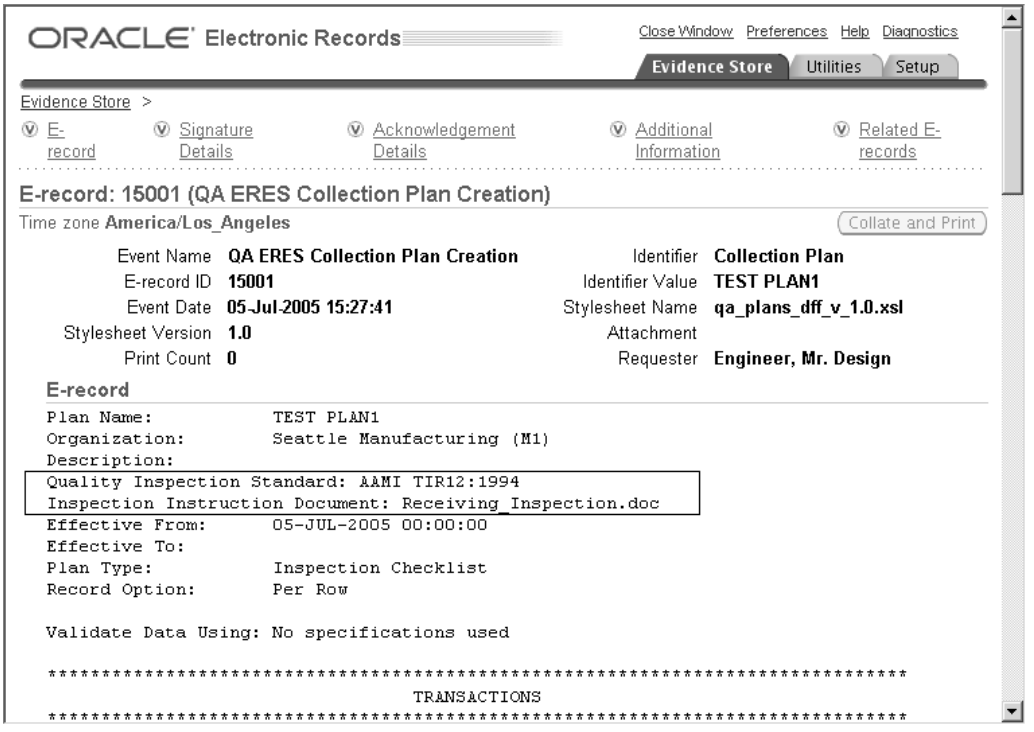
14. Navigate to the Collection Plans window (see: Manufacturing and Distribution Manager Windows and Navigation Paths, page A-3).

    Enter a collection plan.



15. Navigate to the Evidence Store (see: ERES Administrator Windows and Navigation Paths, page A-2) to view the e-record created.

    Notice the descriptive flexfield lines **Quality Inspection Standard** and **Inspection Instruction Document** in the header of the QA ERES Collection Plan Creation e-record.



Related Topics

Uploading Documents, *Oracle E-Records Implementation Guide*

Uploading Documents into the Evidence Store, *Oracle E-Records Implementation Guide*

Managing Templates Using iSign, *Oracle E-Records Implementation Guide*

Setting Up E-Records: A Discrete Manufacturing Example, page 2-1

Enforcing Nonconformance, Disposition, and Corrective Action Request Approvals

If nonconformances, dispositions, and corrective action requests require an e-signature, the user must enter Yes in the eSignature Required collection element when entering quality results (see: Setup Exception - Setting the eSignature Required Collection Element to Yes, page 2-29). You can use the **Assign a value to a collection element** action to automatically assign Yes in the eSignature Required collection element result field based on values entered in other collection element result fields, such as Nonconformance Type or Nonconformance Status, for example. This eliminates the need for the user to initiate the e-signature process. For more information on using the assign a value action, see: Defining Assign A Value Actions: User-Defined Formulas, *Oracle Quality User's Guide*.

Capturing Non-Seeded Collection Elements in Nonconformance, Disposition, and Corrective Action E-Records for History Collection Plans

The stylesheet provided for use with all of the Oracle Quality nonconformance, disposition, and corrective action transactions is based on the seeded collection

plan templates (see: Seeded Template Collection Plan Details, *Oracle Quality Implementation Guide*). This stylesheet, qa_ncm.xsl, enables e-records for nonconformance, disposition, and corrective action transactions to capture only the data entered in the seeded collection plan template collection elements.

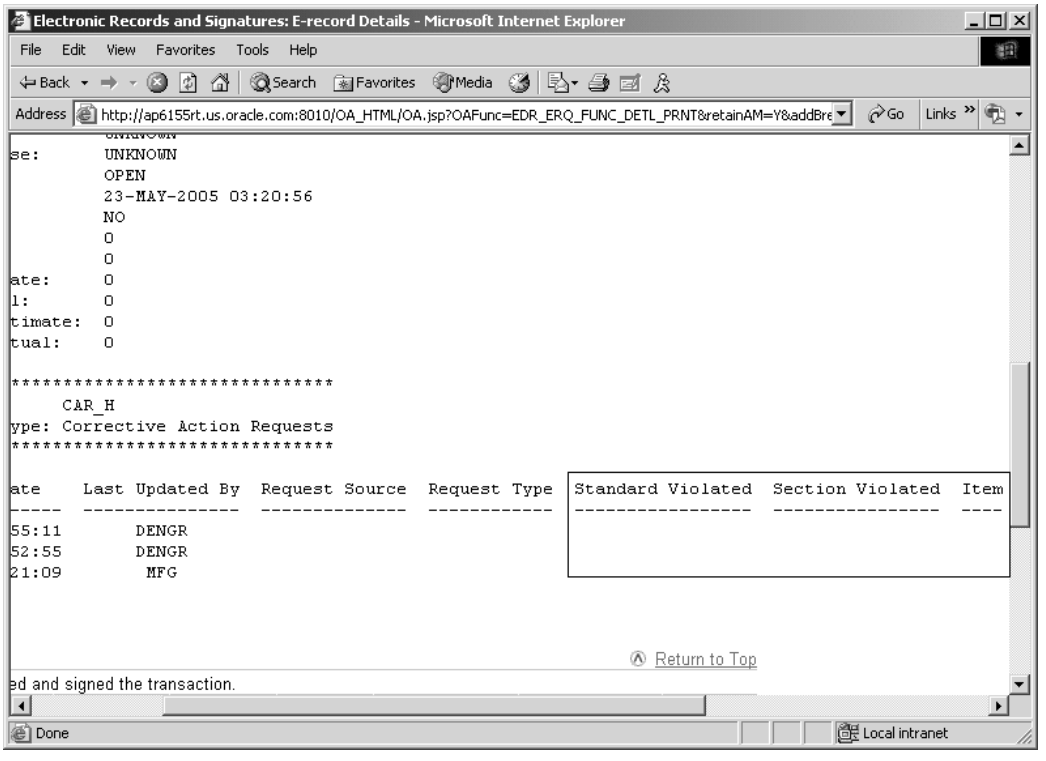
Most users copy, then modify the seeded collection plan templates to meet their needs. These modifications could include the addition of new collection elements to the nonconformance, disposition, and corrective action collection plans. To capture the new collection element data in an e-record, you must modify the seeded stylesheet, qa_ncm.xsl. The process of modifying this stylesheet is similar to the process described in Modifying a Stylesheet to Include Descriptive Flexfields, page 3-12.

The following example describes how to include non-seeded collection elements that are displayed in the history plans in the nonconformance, disposition, and corrective action e-records.

Example:

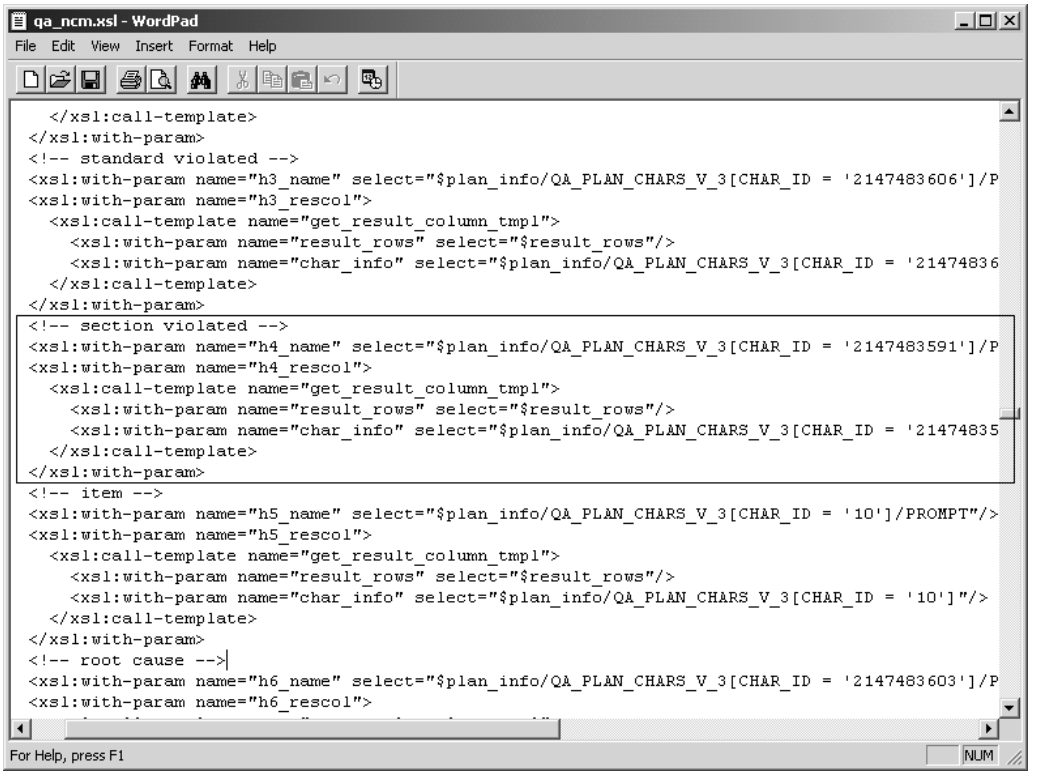
Create a modified versions of the Oracle Quality nonconformance, disposition, and corrective action seeded collection plan template (see: Seeded Template Collection Plan Details, *Oracle Quality Implementation Guide*). The modified collection plans include the non-seeded collection element Cause Code for the corrective action request history collection plan. Remove the seeded collection element Section Violated.

While reviewing nonconformance, disposition, and corrective action e-records, notice that the e-records are not capturing Cause Code, which should appear between the Standard Violated and Item collection elements. Instead, the Section Violated collection element appears in that location of the e-record.



Modify the Stylesheet

1. Follow the instructions in Modify the Stylesheet, page 3-14, to modify the qa_ncm.xsl stylesheet.
2. Search for **section violated**. Notice that the CHAR_ID for the Section Violated collection element equals 2147483591. The CHAR_ID is a unique identifier for a collection element. The qa_ncm.xsl stylesheet uses CHAR_ID to identify the collection elements to include in the history portion of the e-record.



3. Replace the Section Violated CHAR_ID with the Cause Code CHAR_ID, 105.

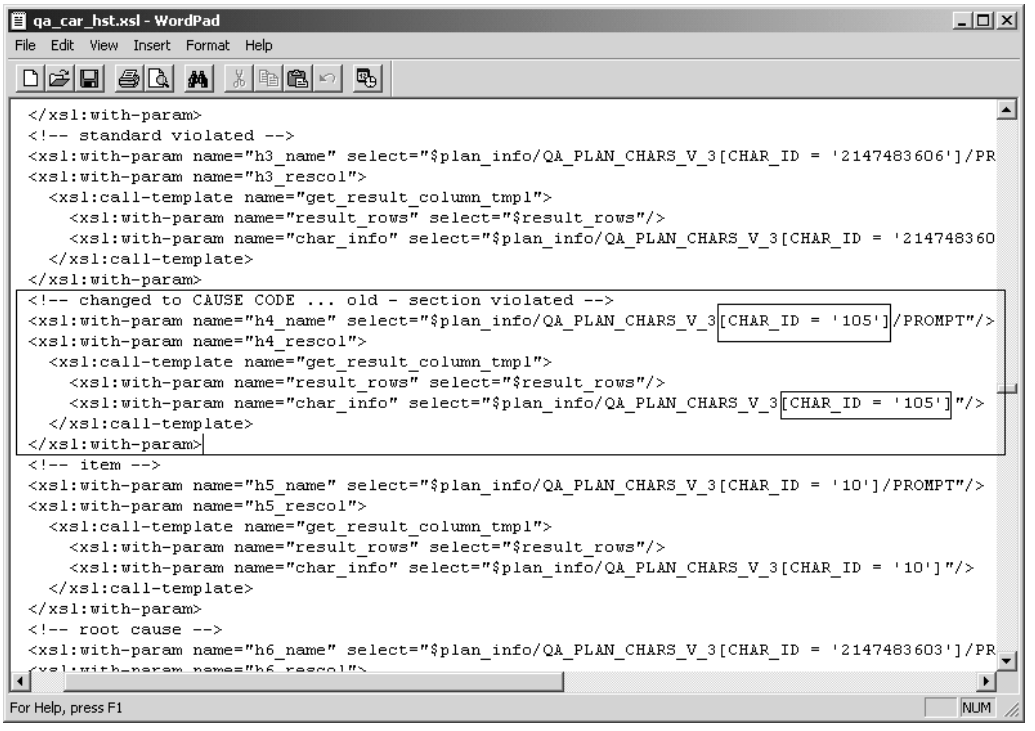
   In this example, save the modified stylesheet as qa_car_hst.xsl.

   Tip: Follow these steps to find the CHAR_ID value of a collection element:

   1. Navigate to a collection element field. You can find the field by navigating to the Collection Plans, Enter Quality Results, or Update Quality Results window (see: Manufacturing and Distribution Manager Windows and Navigation Paths, page A-3), then finding the collection plan containing the collection element.
   2. With the cursor in the collection element field, select (M) Help > Diagnostics > Examine.
   3. Choose the Field list of values button and select CHAR_ID from the list.

      The Value field now displays the CHAR_ID value.

You can also find the CHAR_ID value by using SQL to query the CHAR_ID from the QA_CHARS table.



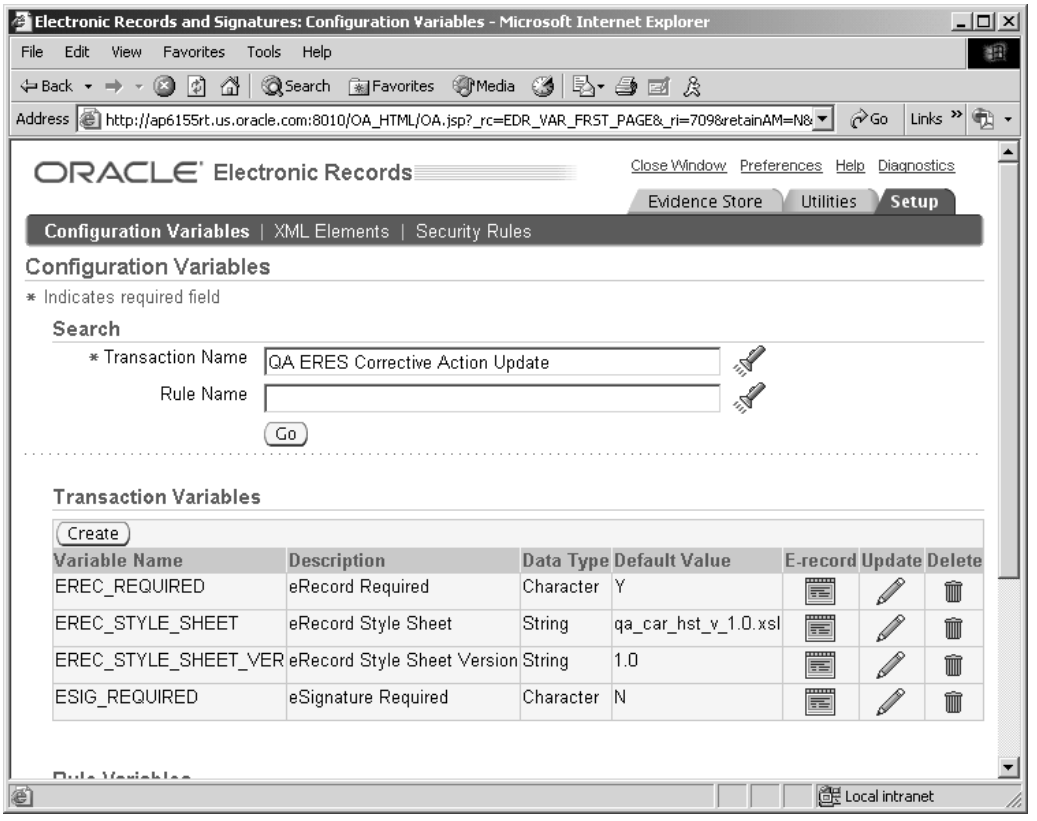
Upload the New Stylesheet

4. Follow the instructions in Upload the New Stylesheet, page 3-15, to upload the qa_car_hst.xsl stylesheet.

   Once you upload qa_car_hst.xsl, the name changes to qa_car_hst_v_1.0.xsl.

Link the New Stylesheet to a Transaction

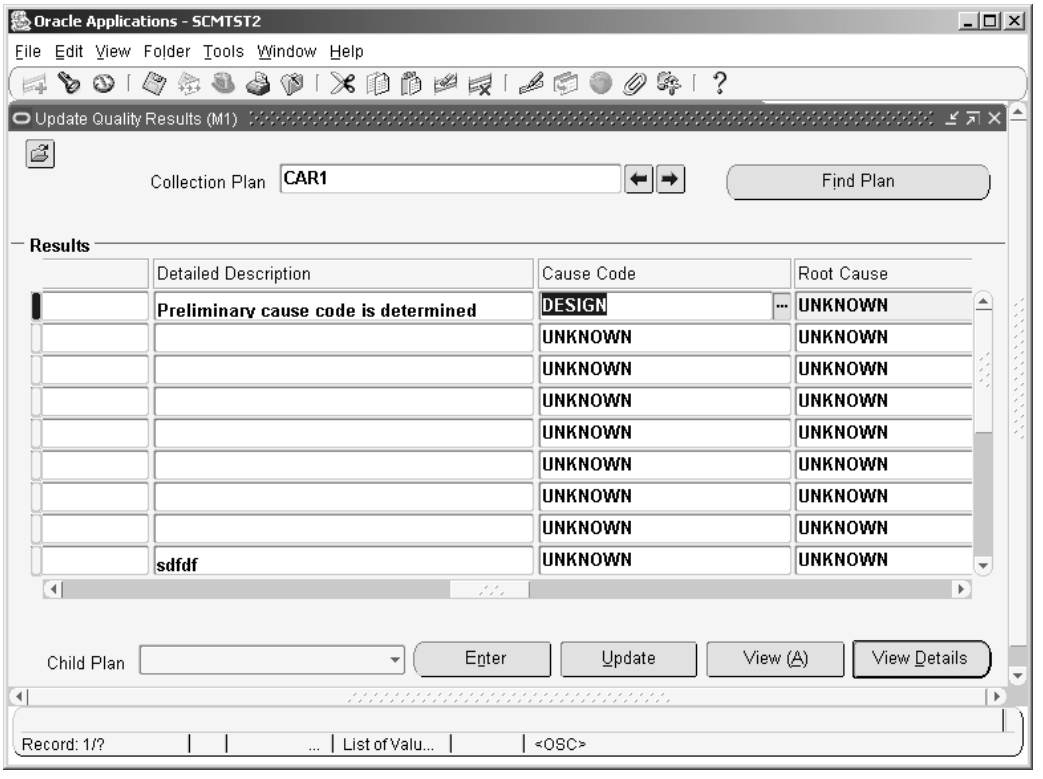
5. Follow the instructions in Link the New Stylesheet to a Transaction, page 3-17, to link the qa_car_hst_v_1.0.xsl stylesheet to all nonconformance, disposition, and corrective action transactions.



Create an E-record Using the New Stylesheet

For this example, update a corrective action request.

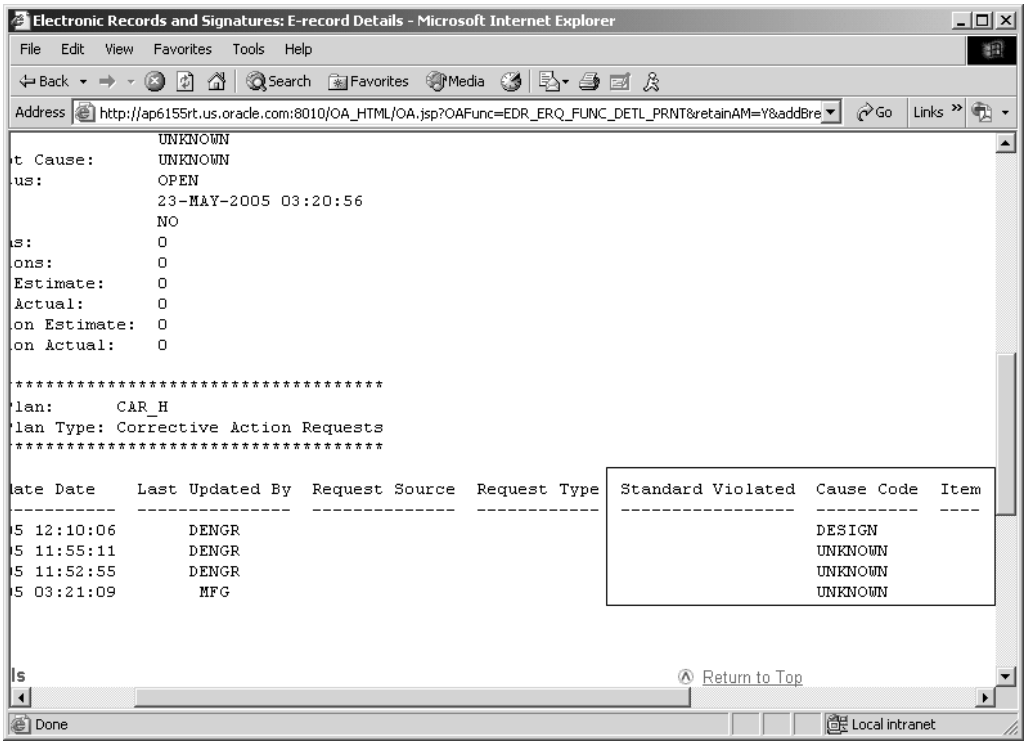
6. Navigate to the Update Quality Results window for corrective action requests (see: Manufacturing and Distribution Manager Windows and Navigation Paths, page A-3).



7. Update quality results in the CAR1 collection plan, including entering Design in the Cause Code collection element.

8. Navigate to the Evidence Store (see: ERES Administrator Windows and Navigation Paths, page A-2) to view the e-record created, E-record ID 16003.

   Notice that the Cause Code column, with the value of Design in the first row, now appears where the Section Violated column previously appeared in the QA ERES Corrective Action Update e-record.



Creating a Device History Record

The medical device industry requires device history records to track an item's manufacturing history. A device history record compiles all of the manufacturing records for an item. You can create a document that functions as a device history record by collating the required e-records, then printing them together in one document. Refer to Printing E-records, *Oracle E-Records Implementation Guide*, for instructions on collating and printing e-records.

> **Note:** The specific requirements for a device history record can vary. The method for creating a device history record described below may not meet all requirements.

Device History Record Example

A medical device company manufactures serial controlled assembly Z1000 at manufacturing plant M1. Z1000 is assembled from two lot controlled items, Z3000 (Supply Type = Push), and another component, Z2000 (Supply Type = Assembly Pull). Perform a quality check of Z1000 after manufacturing completion.

In order to print a document that can function as a device history record for this example, perform the following prerequisites:

- Ensure that you have authority to print e-records. The system administrator must grant you authority using the profile option EDR: E-record Print Granted (see: Enabling Profile Options, *Oracle E-Records Implementation Guide*).
- Enable the E-Records functionality for the following three transactions (see: Implementing E-Records, *Oracle E-Records Implementation Guide*, E-records and E-signatures for Oracle Work in Process, *Oracle Quality User's Guide*, and Setting Up E-Records: A Discrete Manufacturing Example, page 2-1):

    - WIP Job Material Transaction
    - WIP Job Assembly Move
    - WIP Job Assembly Completion
- Create a quality collection plan and associate it with the WIP Completion transaction (see: Quality Data Collection During Transactions, *Oracle Quality User's Guide*).

Now, perform the following steps:

1. Create and release a discrete job to manufacture Z1000 in manufacturing plant M1 (see: Defining Discrete Jobs Manually, *Oracle Work in Process User's Guide* and Releasing Discrete Jobs, *Oracle Work in Process User's Guide*).

   In this example, job 127931 was created.

2. Issue the Push component Z3000 to the job (see: Issuing and Returning All Push Components, *Oracle Work in Process User's Guide*).

   In this example, e-record ID 1004, viewable in the Evidence Store (see: Evidence Store, *Oracle E-Records Implementation Guide*), shows the details of this transaction, the lot information, and the e-signatures.

3. Perform an interoperation move transaction (see: Performing Move Transactions, *Oracle Work in Process User's Guide*).

   In this example, move the partially assembled Z1000 from operation 10 to operation 20. This transaction automatically issues Z2000 into work in process from inventory since Z2000 has a supply type of Assembly Pull. E-record ID 1005 shows the details of this transaction, the lot information, and the e-signatures.

4. Complete the job of assembling Z1000 and collect the mandatory quality results (see: Completing and Returning Assemblies, *Oracle Work in Process User's Guide* and Entering Quality Results Directly, *Oracle Quality User's Guide*).

   In this example, e-record ID 1006 shows the details of this transaction, including lot and serial information and e-signatures.

5. In the Evidence Store, search for these three e-records (1004, 1005, and 1006). Search using criteria that finds all three e-records at once. For example, search on a date range that includes all three e-records (see: Evidence Store, *Oracle E-Records Implementation Guide*).

6. Select the three e-records and choose Collate and Print (see: Printing E-records, *Oracle E-Records Implementation Guide*).

   View the printed document for this example at Example of Collated and Printed E-Records, page B-1.

A

Windows and Navigation Paths

This appendix lists each window referred to in the *Oracle Manufacturing: Implementing Oracle E-Records in Discrete Manufacturing Guide* as well as the associated navigator path for each window by responsibility. It also lists the navigation path(s) for every Oracle E-Records enabled discrete manufacturing event.

This appendix covers the following topics:

- System Adminstrator Windows and Navigation Paths
- Workflow Administrator Pages and Navigation Paths
- Approvals Management Application Administrator Windows and Navigation Paths
- ERES Administrator Windows and Navigation Paths
- iSignatures User Pages and Navigation Paths
- Manufacturing and Distribution Manager Windows and Navigation Paths
- Application Developer Windows and Navigation Paths
- Navigation Paths For All Oracle E-Records Enabled Discrete Manufacturing Business Events

System Adminstrator Windows and Navigation Paths

Although your system administrator may have customized your navigator, typical navigational paths are presented in the following table:

Note: [B] indicates a button and [M] indicates a menu.

Window Name	Navigation Path
System Profile Values	Profile > System

Workflow Administrator Pages and Navigation Paths

Although your system administrator may have customized your navigator, typical navigational paths are presented in the following table:

Note: [B] indicates a button and [M] indicates a menu.

Window or Page Name	Navigation Path
Worklist	Worklist
Business Events	Administrator Workflow > Business Events
Subscriptions	Administrator Workflow > Business Events [B] Subscription column icon

Approvals Management Application Administrator Windows and Navigation Paths

Although your system administrator may have customized your navigator, typical navigational paths are presented in the following table:

Note: [B] indicates a button and [M] indicates a menu.

Window or Page Name	Navigation Path
Oracle Approvals Management	Approvals Management Application Administrator > Approvals
Worklist	Workflow > Worklist

ERES Administrator Windows and Navigation Paths

Although your system administrator may have customized your navigator, typical navigational paths are presented in the following table:

Note: [B] indicates a button and [M] indicates a menu.

Window or Page Name	Navigation Path
Configuration Variables	Administration Tasks > Setup
Evidence Store	Administration Tasks > Evidence Store

iSignatures User Pages and Navigation Paths

Although your system administrator may have customized your navigator, typical navigational paths are presented in the following table:

Note: [B] indicates a button and [M] indicates a menu.

Page Name	Navigation Path
Files Approval	Files Approval
File Upload	Files Approval [B] Upload Files

Manufacturing and Distribution Manager Windows and Navigation Paths

Although your system administrator may have customized your navigator, typical navigational paths are presented in the following table:

Note: [B] indicates a button and [M] indicates a menu.

Window or Page Name	Navigation Path
Approval Lists	Engineering > Setup > Approval Lists
Bills of Material	Bills of Materials > Bills > Bills
Change Types	Engineering > Setup > Change Types
Change Type Processes	Engineering > Setup > Change Types [B] Processes
Collection Plans	Quality > Setup > Collection Plans
Copy Collection Plans	Quality > Setup > Copy Collection Plans
ECO Priorities	Engineering > Setup > Priorities
Engineering Bills of Material	Engineering > Prototypes > Bills > Bills
Engineering Item	Engineering > Prototypes > Items > Organization Items
Organization Item	Inventory > Items > Organization Items
Specifications	Quality > Setup > Specifications
Update Quality Results	Quality > Corrective Action > Update Corrective Action Request

Application Developer Windows and Navigation Paths

Although your system administrator may have customized your navigator, typical navigational paths are presented in the following table:

Note: [B] indicates a button and [M] indicates a menu.

Window or Page Name	Navigation Path
Descriptive Flexfield Segments	Flexfield > Descriptive > Segments
Segments	Flexfield > Descriptive > Segments [B] Segments [B] New

Navigation Paths For All Oracle E-Records Enabled Discrete Manufacturing Business Events

Use the Manufacturing and Distribution Manager responsibility when performing these business events. Although your system administrator may have customized your navigator, typical navigational paths are presented in the following table:

Note: [B] indicates a button and [M] indicates a menu.

Oracle Engineering Business Events

Business Event	Navigation Path
ECO Creation / ECO Update	Engineering > ECOs > ECOs
ECO Approval	ECO Approval Workflow
ECO Implementation	Engineering > ECOs > ECOs [M] Tools > Implement
ECO Schedule	Engineering > ECOs > ECOs [M] Tools > Schedule
ECO Reschedule	Engineering > ECOs > ECOs [M] Tools > Reschedule
ECO Cancellation	Engineering > ECOs > ECOs [M] Tools > Cancel
Transfer to Manufacturing	1. Engineering > Prototypes > Transfer To Manufacturing 2. Engineering > Prototypes > Items > Master/Organization Items [M] Tools > Engineering Transfer 3. Engineering > Prototypes >Bills > Bills [M] Tools > Transfer Bill 4. Engineering > Prototypes > Routings > Routings [M] Tools > Transfer Routing 5. Through ECO Approval (the ECO Approval captures the e-signature)
Copy to Manufacturing	Engineering > Prototypes > Copy to Manufacturing

Oracle Inventory Business Events

Business Event	Navigation Path
Item Creation Item Update	1. Inventory > Items > Master Items 2. Engineering > Prototypes > Items > Master Items
Item Organization Assignment	1. Inventory > Items > Master Items [M] Tools > Organization Assignment 2. Engineering > Prototypes > Items > Master Items > Organization Assignment
Item Revision	1. Inventory > Items > Master/Organization Items [M] Tools > Revisions 2. Engineering > Prototypes > Items > Master/Organization Items [M]Tools > Revisions
Item Cross Reference	Inventory > Items > Master/Organization Items [M}Tools > Cross References

Oracle Bills of Material

Business Event	Navigation Path
Bill Creation	Bills of Materials > Bills > Bills
Bill Update	Engineering > Prototypes > Bills > Bills
Routing Creation	Bills of Materials > Routings > Routings
Routing Update	Engineering > Prototypes > Routings > Routings
Mass Change Bills	Engineering > ECOs > Mass Changes Bills of Materials > Bills > Mass Changes

Oracle Work in Process

Business Event	Navigation Path
WIP Material Transactions	WIP > Material Transactions > WIP Material Transactions
Move Transactions	WIP > Move Transactions > Move Transactions
Completion Transactions	WIP > Material Transactions > Completion Transactions

Oracle Quality

Business Event	Navigation Path
Collection Element Creation Collection Element Update	Quality > Setup > Collection Elements
Collection Plan Creation Collection Plan Update	Quality > Setup > Collection Plans
Specification Creation Specification Update	Quality > Setup > Specifications
Specification Organization Assignment	Quality > Setup > Specifications [B] Org Assignment
Quality Result Creation	Quality > Results > Entry > Enter Quality Results
Quality Result Update	Quality > Results > Entry > Update Quality Results
Nonconformance Creation	Quality > Nonconformances > Enter Nonconformances
Nonconformance Update	Quality > Nonconformances > Update Nonconformances
Nonconformance Master Approval Nonconformance Detail Approval	Quality > Nonconformances > Enter Nonconformances Quality > Nonconformances > Update Nonconformances
Disposition Creation	Quality > Dispositions > Enter Dispositions
Disposition Update	Quality > Dispositions > Update Dispositions
Disposition Header Approval	Quality > Dispositions > Enter Dispositions
Disposition Detail Approval	Quality > Dispositions > Update Dispositions
Corrective Action Creation	Quality > Corrective Action > Enter Corrective Action Request
Corrective Action Update	Quality > Corrective Action > Update Corrective Action Request
Corrective Action Request Approval	Quality > Corrective Action > Enter Corrective Action Request Quality > Corrective Action > Update Corrective Action Request
Corrective Action Review Approval Corrective Action Implementation Approval	Quality > Corrective Action > Enter Corrective Action Request Quality > Corrective Action > Update Corrective Action Request

Oracle Shipping

Business Event	Navigation Path
Delivery Shipment	Quality > Reports > Submit Requests > Quality Shipping ERES Collector

Oracle Purchasing

Business Event	Navigation Path
ASL Creation ASL Update	Purchasing > Supply Base > Approved Supplier List PO Approval workflow
Receiving Quality Inspection	Purchasing > Receiving > Receiving Transactions
Receiving Transfer with Quality Data Collection	Purchasing > Receiving > Receiving Transactions
Receiving Delivery with Quality Data Collection	Purchasing > Receiving > Receiving Transactions

B

Device History Record Example

This appendix shows collated and printed e-records created for the Device History Record Example, page 3-26.

This appendix covers the following topics:

- Example of Collated and Printed E-Records

Example of Collated and Printed E-Records

The following document provides an example of collated and printed e-records. For an explanation of how this example document was created, see Device History Record Example, page 3-26.

E-record: 1004 (WIP Job Material Transaction)

Time zone **America/Los_Angeles**

Event Name	WIP Job Material Transaction	Identifier	Job / Transaction Type
E-record ID	1004	Identifier Value	127931 / WIP component issue
Event Date	22-SEP-2004 14:19:54	Stylesheet Name	wipmtlxs.xsl
Requester	Smith, Mr. Jonathan	Stylesheet Version	1

Signature Details

The following users approved and signed the transaction.

Signer	Response	Date Signed	Reason	Signer Type	Comments	Overriding Details	Change Details
Smith, Mr. Jonathan	Approve	22-SEP-2004 14:21:17	None	Author	Verified the correct component lot(s) and quantities have been issued		

Acknowledgement Details

The business transaction resulting in the creation of this e-record sent an acknowledgement with the following details.

Status	Successful
Acknowledged By	WIP
Acknowledgement Date	22-SEP-2004 14:21:25
Comments	

E-record

Organization Code:	M1
Organization Name:	Seattle Manufacturing
Assembly:	Z1000
Assembly Description:	Hybrid Tinned Assy
Job:	127931
Job Description:	
Job Type:	Standard
Bill Revision:	A
Alternate Bill Designator:	
Routing Revision:	A
Alternate Routing Designator:	
Project Number:	
Task Number:	
Transaction Type:	WIP component issue
Transaction Date:	22-SEP-2004 14:19:35

TRANSACTION LINE

Component:	Z3000
Component Description:	Hybrid Assembly
Component Revision:	
Subinventory:	Stores
Locator:	
Operation Sequence Number:	10
Operation Code:	
Department Code:	ASSEMBLY
Job Required Quantity:	3
Transaction Quantity:	3
UOM:	Ea
Reason:	
Reference:	

LOT INFORMATION

Lot Number:	S00107
Expiration Date:	05-NOV-2004 13:22:07
Qty:	3

E-record: 1005 (WIP Job Assembly Move)

Time zone **America/Los_Angeles**

Event Name	**WIP Job Assembly Move**	Identifier	**Job / Transaction Type**
E-record ID	**1005**	Identifier Value	**127931 / Move transaction**
Event Date	**22-SEP-2004 14:22:56**	Stylesheet Name	**wipmovxs.xsl**
Requester	**Smith, Mr. Jonathan**	Stylesheet Version	**1**

Signature Details

The following users approved and signed the transaction.

Signer	Response	Date Signed	Reason	Signer Type	Comments	Overriding Details	Change Details
Smith, Mr. Jonathan	Approve	22-SEP-2004 14:23:48	None	Author	Moved the job to last operation and backflushed required components		

Acknowledgement Details

The business transaction resulting in the creation of this e-record sent an acknowledgement with the following details.

Status	**Successful**
Acknowledged By	**WIP**
Acknowledgement Date	**22-SEP-2004 14:23:57**
Comments	

E-record

```
Organization Code:                M1
Organization Name:                Seattle Manufacturing

Assembly:                         Z1000
Assembly Description:             Hybrid Tinned Assy
Job:                              127931
Job Description:
Job Type:                         Standard
Bill Revision:                    A
Alternate Bill Designator:
Routing Revision:                 A
Alternate Routing Designator:
Project Number:
Task Number:

Transaction Type:                 Move transaction

From Operation Sequence Number:   10
From Operation Sequence Code:
From Department:                  ASSEMBLY
From Intraoperation Step:         Queue
To Operation Sequence Number:     20
To Operation Sequence Code:
To Department:                    TESTING
To Intraoperation Step:           To move

Transaction Quantity:             3
Overcompleted Quantity:
UOM:                              Ea
Transaction Date:                 22-SEP-2004 14:22:29

Reason:
Reference:
```

```
BACKFLUSH COMPONENT
==========================
Transaction Type:                 WIP component issue
Component:                        Z2000
Component Description:            Solder, Bar, SN63E
Component Revision:

Subinventory:                     RIP
Locator:
Operation Sequence Number:        10
Operation Code:
Department Code:                  ASSEMBLY
Job Required Quantity:            3
Transaction Quantity:             3
UOM:                              Ea
Reason:
Reference:

LOT INFORMATION
==========================
Lot Number:                       S00106
Expiration Date:                  25-NOV-2004 13:21:40
Qty:                              3
```

E-record: 1006 (WIP Job Assembly Completion)

Time zone **America/Los_Angeles**

Event Name	WIP Job Assembly Completion	Identifier	Job / Transaction Type
E-record ID	1006	Identifier Value	127931 / WIP Assy Completion
Event Date	22-SEP-2004 14:26:20	Stylesheet Name	wipcmpxs.xsl
Requester	Smith, Mr. Jonathan	Stylesheet Version	1

Signature Details

The following users approved and signed the transaction.

Signer	Response	Date Signed	Reason	Signer Type	Comments	Overriding Details	Change Details
Smith, Mr. Jonathan	Approve	22-SEP-2004 14:27:16	None	Reviewer	Final assembly inspected and verified per specifications		

Acknowledgement Details

The business transaction resulting in the creation of this e-record sent an acknowledgement with the following details.

Status	**Successful**
Acknowledged By	**WIP**
Acknowledgement Date	**22-SEP-2004 14:27:21**
Comments	

E-record

Organization Code:	M1
Organization Name:	Seattle Manufacturing
Assembly:	Z1000
Assembly Description:	Hybrid Tinned Assy
Job:	127931
Job Description:	
Job Type:	Standard
Bill Revision:	A
Alternate Bill Designator:	
Routing Revision:	A
Alternate Routing Designator:	
Project Number:	
Task Number:	
Transaction Type:	WIP Assy Completion
Transaction Date:	22-SEP-2004 14:24:22
Job Quantity:	3
UOM:	Ea

ASSEMBLY COMPLETION DETAILS

Kanban Number:	
Subinventory:	Stores
Locator:	
Quantity:	1
UOM:	Ea
Reason:	
Reference:	

```
LOT INFORMATION
==========================
Lot Number:                    S00111
Expiration Date:               22-OCT-2004 00:00:00
Qty:                           1

SERIAL INFORMATION
==========================
Serial Number:                 SJM092204-01
Parent Serial Number:
Manufacturer"s Serial Number:


**********************************************************************
**********
QUALITY RESULTS
**********************************************************************
**********

*******************************************
Collection Plan:      SJM WIP COMPLETION Z1000
Collection Plan Type: WIP Inspection
*******************************************

(SJM WIP COMPLETION Z1000) - Result Row 1/1
-------------------------------------------
Specification Used: None

Item:              Z1000
Job:               127931
Qty Tested:        1
Qty Passed:        1


Qty Failed:        0
% Passed:          100
% Failed:          0
Employee:          Black, Mr. Chris
Dept:              ASSEMBLY
Comments:          Inspected and verified final assembly is per
design and functional specifications
```

C

Oracle E-Records Event Data for Discrete Manufacturing

The tables in this appendix detail the seed data entered to support each discrete manufacturing application integrated with Oracle E-Records.

This appendix covers the following topics:

- Event Data for Oracle Engineering
- Event Data for Oracle Inventory
- Event Data for Oracle Bills of Material
- Event Data for Oracle Work in Process
- Event Data for Oracle Quality
- Event Data for Oracle Purchasing
- Event Data for Oracle Shipping

Event Data for Oracle Engineering

Oracle Workflow Seeded Data

Events

Name	Display Name	Description	Status	Owner Name	Owner Tag
oracle.apps.eng.ecoApproval	ECO Approval	ERES Event for ECO Approval	Enabled	Oracle Engineering	ENG
oracle.apps.eng.ecoCreate	ECO Creation	ERES Event for ECO Create	Enabled	Oracle Engineering	ENG
oracle.apps.eng.ecoUpdate	ECO Update	ERES Event for ECO Update	Enabled	Oracle Engineering	ENG
oracle.apps.eng.ecoSchedule	ECO Schedule	ERES Event for ECOSchedule	Enabled	Oracle Engineering	ENG
oracle.apps.eng.ecoReschedule	ECO Reschedule	ERES Event for ECOReschedule	Enabled	Oracle Engineering	ENG
oracle.apps.eng.ecoCancellation	ECO Cancellation	ERES Event for ECOCancellation	Enabled	Oracle Engineering	ENG
oracle.apps.eng.eco Implementation	ECO Implementation	ERES Event for ECO Implementation	Enabled	Oracle Engineering	ENG
oracle.apps.eng.transferTo Manufacturing	Transfer To Manufacturing	ERES Event for Transfer To Manufacturing	Enabled	Oracle Engineering	ENG
oracle.apps.eng.copyTo Manufacturing	Copy To Manufacturing	ERES Event for Copy To Manufacturing	Enabled	Oracle Engineering	ENG
oracle.apps.eng.massChangeBill	Mass Change Bills	ERES Event for Mass Change Bills	Enabled	Oracle Engineering	ENG

Event Key

Event Name	User Event Key (Identifier)
oracle.apps.eng.ecoApproval	Engineering Change Order
oracle.apps.eng.ecoCreate	Engineering Change Order
oracle.apps.eng.ecoUpdate	Engineering Change Order
oracle.apps.eng.ecoSchedule	Engineering Change Order
oracle.apps.eng.ecoReschedule	Engineering Change Order
oracle.apps.eng.ecoCancellation	Engineering Change Order
oracle.apps.eng.ecoImplement	Engineering Change Order
oracle.apps.eng.transferTo Manufacturing	Organization, Item
oracle.apps.eng.copyToManufacturing	Organization, Item
oracle.apps.eng.massChangeBill	Organization, Change Notice

Event Subscription

All events subscriptions have the same values for the following fields:

- System = HM001
- Source Type = Local
- Phase = 0
- Status = Disabled
- Rule Data = Key
- Rule Function = EDR_PSIG_RULE.PSIG_RULE
- Priority = Normal

Event Filter	Parameters
oracle.apps.eng.eco.approval	EDR_XML_MAP_CODE=oracle.apps.eng. ecoGeneric EDR_AME_TRANSACTION_TYPE= oracle.apps.eng.ecoApproval
oracle.apps.eng.eco.create	EDR_XML_MAP_CODE=oracle.apps.eng. ecoGeneric EDR_AME_TRANSACTION_TYPE= oracle.apps.eng.ecoCreate
oracle.apps.eng.eco.update	EDR_XML_MAP_CODE=oracle.apps.eng. ecoGeneric EDR_AME_TRANSACTION_TYPE= oracle.apps.eng.ecoUpdate
oracle.apps.eng.eco.schedule	EDR_XML_MAP_CODE=oracle.apps.eng. ecoGeneric EDR_AME_TRANSACTION_TYPE= oracle.apps.eng.ecoSchedule
oracle.apps.eng.eco.reschedule	EDR_XML_MAP_CODE=oracle.apps.eng. ecoGeneric EDR_AME_TRANSACTION_TYPE= oracle.apps.eng.ecoReschedule
oracle.apps.eng.eco.cancellation	EDR_XML_MAP_CODE = oracle.apps.eng. eco. genericEDR_AME_TRANSACTION_TYPE = oracle.apps.eng.eco.cancellation
oracle.apps.eng.eco.implement	EDR_XML_MAP_CODE=oracle.apps.eng. ecoGeneric EDR_AME_TRANSACTION_TYPE= oracle.apps.eng.ecoImplement oracle.apps.eng. copyToManufacturing=IGNORE_SIGNATURE oracle.apps.eng.transferToManufacturing= IGNORE_SIGNATURE oracle.apps.inv. itemRevisionEntry=IGNORE_SIGNATURE oracle.apps.bom.billUpdate=IGNORE_S IGNATURE oracle.apps.bom.billCreate=IGNORE_ SIGNATURE oracle.apps.bom.routingUpdate= IGNORE_SIGNATURE oracle.apps.bom. routingCreate=IGNORE_SIGNATURE
oracle.apps.eng.transferTo Manufacturing	EDR_XML_MAP_CODE=oracle.apps.eng. manufact EDR_AME_TRANSACTION_TYPE= oracle.apps.eng.transferToManufacturing
oracle.apps.eng.copyTo Manufacturing	EDR_XML_MAP_CODE=oracle.apps.eng. manufact EDR_AME_TRANSACTION_TYPE= oracle.apps.eng.copyToManufacturing oracle.apps. inv.itemCreate=IGNORE_SIGNATURE oracle. apps.bom.billCreate=IGNORE_SIGNATURE oracle.apps.bom.routingCreate=IGNORE_S IGNATURE
oracle.apps.eng.massChangeBill	EDR_XML_MAP_CODE=oracle.apps.eng. massChangeBill EDR_AME_TRANSACT ION_TYPE=oracle.apps.eng.massChangeBill oracle.apps.eng.ecoCreate=IGNORE_SIGNATURE

Oracle Approvals Management Seeded Data

Transaction Type

All transaction types listed below belong to the Oracle Engineering application.

Transaction Type ID	Transaction Type Description	Line Item Id Query String
oracle.apps.eng.ecoCreate	ENG ERES ECO Creation	N/A
oracle.apps.eng.ecoUpdate	ENG ERES ECO Update	N/A
oracle.apps.eng.ecoSchedule	ENG ERES ECO Schedule	N/A
oracle.apps.eng.ecoReschedule	ENG ERES ECO Reschedule	N/A
oracle.apps.eng.ecoCancellation	ENG ERES ECO Cancellation	N/A
oracle.apps.eng.ecoImplementation	ENG ERES ECO Implementation	N/A
oracle.apps.eng.transferToManufacturing	ENG ERES Transfer to Manufacturing	select category_id from mtl_item_categories where inventory_item_id = (select distinct INVENTORY_ITEM_ID from ENG_REVISED_ITEMS_TEMP where temp_id = :transactionId) and organization_id = (select distinct ORGANIZATION_ID from ENG_REVISED_ITEMS_TEMP where temp_id = :transactionId) order by category_id
oracle.apps.eng.copyToManufacturing	ENG ERES Copy to Manufacturing	select category_id from mtl_item_categories where inventory_item_id = (select distinct INVENTORY_ITEM_ID from ENG_REVISED_ITEMS_TEMP where temp_id = :transactionId) and organization_id = (select distinct ORGANIZATION_ID from ENG_REVISED_ITEMS_TEMP where temp_id = :transactionId) order by category_id
oracle.apps.eng.massChangeBills	ENG ERES Mass Change Bills	N/A

Transaction Attributes

Although the user can define their own attributes for the transaction types listed above, the more commonly used attributes are seeded. The following attributes apply to all Oracle Engineering transaction types. Attributes specific to certain transaction types are listed in later tables, by transaction type.

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Mandatory Attribute	ALLOW_DELETING_RULE_GENERATED_APPROVERS	boolean	whether to let the calling application (or its end users) delete approvers generated by the rules	Yes	false
Mandatory Attribute	ALLOW_REQUESTOR_APPROVAL	boolean	whether to allow requestors to approve their own transactions (when the rules do so)	Yes	false
Mandatory Attribute	AT_LEAST_ONE_RULE_MUST_APPLY	boolean	whether to require that at least one rule apply to each transaction	Yes	false
Mandatory Attribute	EFFECTIVE_RULE_DATE	date	the date that determines which rules are active	Yes	N/A
Mandatory Attribute	EVALUATE_PRIORITIES_PER_ITEM	boolean	whether to evaluate rule priorities per item under strict item evaluation	Yes	false
Mandatory Attribute	REJECTION_RESPONSE	string	how AME responds to a rejection	Yes	STOP_ALL_ITEMS
Mandatory Attribute	USE_RESTRICTIVE_ITEM_EVALUATION	boolean	whether to require that the same item satisfy all item conditions in a given rule	Yes	false
Mandatory Attribute	USE_WORKFLOW	boolean	whether OAM should log exceptions to the Workflow context stack	Yes	true
Mandatory Attribute	WORKFLOW_ITEM_KEY	string	the transaction's Workflow item key	Yes	N/A
Mandatory Attribute	WORKFLOW_ITEM_TYPE	string	the transaction's Workflow item type	Yes	N/A
Non-mandatory Header Attribute	ALLOW_EMPTY_APPROVAL_GROUPS	boolean	whether to allow approval groups to have no members	Yes	false

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-mandatory Header Attribute	INCLUDE_ALL_JOB_LEVEL_APPROVERS	boolean	whether to include all approvers at a given job level	Yes	false
Non-mandatory Header Attribute	TRANSACTION_DATE	date	date transaction occurred	No	SELECT ame_util.versionDateToString(CREATION_DATE) FROM ENG_ENG_CHANGES_INTERFACE_V WHERE organization_id = TO_NUMBER(SUBSTR(:transactionId,1,INSTR(:transactionId,'-') -1)) AND change_notice = SUBSTR(:transactionId,INSTR(:transactionId,'-')+1,(LENGTH(:transactionId) - INSTR(:transactionId,'-')))
Non-mandatory Header Attribute	TRANSACTION_GROUP_ID	number	business-group ID in which transaction occurred	Yes	N/A
Non-mandatory Header Attribute	TRANSACTION_REQUESTOR_PERSON_ID	number	person ID of person initiating transaction, if any	Yes	N/A
Non-mandatory Header Attribute	TRANSACTION_REQUESTOR_USER_ID	number	user ID of user initiating transaction, if any	No	SELECT last_updated_by FROM ENG_ENG_CHANGES_INTERFACE_V WHERE organization_id = TO_NUMBER(SUBSTR(:transactionId,1,INSTR(:transactionId,'-') -1)) AND change_notice = SUBSTR(:transactionId,INSTR(:transactionId,'-')+1, (LENGTH(:transactionId) - INSTR(:transactionId,'-')))
Non-mandatory Header Attribute	TRANSACTION_SET_OF_BOOKS_ID	number	set-of-books ID in which transaction occurred	Yes	N/A

Seeded transaction attributes for the following transaction types:

- ENG ERES ECO Creation (oracle.apps.eng.ecoCreate)
- ENG ERES ECO Update (oracle.apps.eng.ecoUpdate)
- ENG ERES ECO Implementation (oracle.apps.eng.ecoImplementation)
- ENG ERES ECO Schedule (oracle.apps.eng.ecoSchedule)

- ENG ERES ECO Reschedule (oracle.apps.eng.ecoReschedule)
- ENG ERES ECO Cancellation (oracle.apps.eng.ecoCancellation)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	ECO	string	ECO. Typically used for an ECO range, for example, trigger an event when an ECO is between ABC0001 and ABC1000	No	select change_notice from ENG_ENGINEERING_CHANGES where change_id = to_number(:transactionId)
Non-Mandatory Header Attribute	ECO DEPARTMENT	string	ECO Department. For example, triggers an event when the ECO Department = Design Engineering.	No	SELECT ECO_DEPARTMENT FROM ENG_ENGINEERING_CHANGES_V WHERE change_id = to_number(:transactionId)
Non-mandatory Header Attribute	ECO TYPE	string	ECO Type. For example, trigger an event when the ECO Type = New Product.	No	select ecotvl.type_name from ENG_ENGINEERING_CHANGES eec, ENG_CHANGE_ORDER_TYPES_VL ecotvl where eec.change_id = to_number(:transactionId) and eec.CHANGE_ORDER_TYPE_ID = ecotvl.CHANGE_ORDER_TYPE_ID
Non-mandatory Header Attribute	JOB_LEVEL_NON_DEFAULT_STARTING_POINT_PERSON_ID	number	person ID of non-default first approver for job-level authority approval types	No	select null from dual
Non-mandatory Header Attribute	ORGANIZATION_CODE	string	Organization Code.	No	select organization_code FROM mtl_parameters where organization_id = (select organization_id FROM ENG_ENGINEERING_CHANGES WHERE change_id = to_number(:transactionId))
Non-mandatory Header Attribute	PRIORITY	string	Priority. For example, trigger an event when Priority = Urgent.	No	SELECT PRIORITY_CODE FROM ENG_ENGINEERING_CHANGES WHERE change_id = to_number(:transactionId)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-mandatory Header Attribute	PROJECT	string	Project. Use when under project/task control.	No	SELECT PROJECT_NUMBER FROM ENG_ENGINEERING_CHANGES_V WHERE change_id = to_number(:transactionId)
Non-mandatory Header Attribute	TASK	string	Task	No	SELECT TASK_NUMBER FROM ENG_ENGINEERING_CHANGES_V WHERE change_id = to_number(:transactionId)
Non-mandatory Header Attribute	TOP_SUPERVISOR_PERSON_ID	number	person ID of the top person in the HR supervisory hierarchy	Yes	N/A
Non-mandatory Header Attribute	TRANSACTION_ORG_ID	number	org ID in which transaction occurred	Yes	N/A

Seeded transaction attributes for the following transaction types:

- ENG ERES Transfer to Manufacturing (oracle.apps.eng.transferToManufacturing)
- ENG ERES Copy to Manufacturing (oracle.apps.eng.copyToManufacturing)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	JOB_LEVEL_NON_DEFAULT_STARTING_POINT_PERSON_ID	number	person ID of non-default first approver for job-level authority approval types	No	select null from dual
Non-Mandatory Header Attribute	ORGANIZATION_CODE	string	Organization Code	No	SELECT ORGANIZATION_CODE FROM ENG_REVISED_ITEMS_TEMP WHERE TEMP_ID = :transactionId
Non-Mandatory Header Attribute	REVISED_ITEM	string	ITEM	No	SELECT ITEM_NUMBER FROM ENG_REVISED_ITEMS_TEMP WHERE TEMP_ID = :transactionId
Non-Mandatory Header Attribute	TOP_SUPERVISOR_PERSON_ID	number	person ID of the top person in the HR supervisory hierarchy	Yes	N/A
Non-Mandatory Header Attribute	TRANSACTION_ORG_ID	number	org ID in which transaction occurred	No	select organization_id from qa_plans where plan_id = :transactionId
Non-Mandatory Line Item Attribute	ITEM CATEGORY	string	The category set defined for functional area 'Order Mgmt' in Inventory	No	select concatenated_segments from MTL_CATEGORIES_B_KFV where category_id in (select category_id from mtl_item_categories where inventory_item_id = (select distinct INVENTORY_ITEM_ID from ENG_REVISED_ITEMS_TEMP where temp_id = :transactionId) and organization_id = (select distinct ORGANIZATION_ID from ENG_REVISED_ITEMS_TEMP where temp_id = :transactionId)) order by category_id

Seeded transaction attributes for the following transaction types:

- ENG ERES Mass Change Bills (oracle.apps.eng.massChangeBills)

Attribute Category	Attribute Name	At-tribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	JOB_ LEVEL_ NON_ DEFAULT_ STARTING_ POINT_ PERSON_ID	number	person ID of non-default first approver for job-level authority approval types	No	select null from dual
Non-Mandatory Header Attribute	ORGAN IZATION_ CODE	string	Organization Code	No	select organization_code FROM mtl_parameters where organization_id = to_number(substrb(: transactionId,1, instrb(: transactionId,'-') -1))
Non-Mandatory Header Attribute	TOP_SU PERVISOR_ PERSON_ID	number	person ID of the top person in the HR supervisory hierarchy	Yes	N/A
Non-Mandatory Header Attribute	TRANSACT ION_ORG_ ID	number	org ID in which transaction occurred	No	select organization_id from qa_plans where plan_id = :transactionId

Define Rules/Associate Approvers

No rules or approvers are seeded. As part of the implementation, you must complete these activities (see: Setting Up Oracle Approvals Management, *Oracle E-Records Implementation Guide*).

Oracle E-Records Seeded Data: Generic Query Attributes

Use generic query attributes to search for e-records and their associated documents in the Evidence Store. For instructions on how to search, see Evidence Store, *Oracle E-Records Implementation Guide*. For all attributes listed below:

Query Element Type = Generic

ECO Approval

Application Code	XML Element Tag	Display Name	Comments
ENG	CHANGE_NOTICE	ECO	Used to query ECO-related events.
ENG	CHANGE_ORDER_TYPE	ECO Type	N/A
ENG	ECO_DEPARTMENT	ECO Department	N/A
ENG	REVISION	ECO Revision	N/A
ENG	PROJECT_NUMBER	Project	N/A
ENG	TASK_NUMBER	Task	N/A
ENG	REVISED_ITEM_NO	Item	For revised items, items transferred to manufacturing, and new manufacturing items copied to manufacturing.
ENG	NEW_ITEM_REVISION	Item Revision	For new item revisions.
ENG	NEW_ROUTING_REVISION	Routing Revision	For new routing revision.
ENG	ALTERNATE_BOM_DESIGNATOR	Alternate Designator	N/A
ENG	STANDARD_OPERATION_CODE	Operation Code	For a standard operation in a routing change.
BOM	RESOURCE_CODE	Resource Code	N/A
ENG	COMPONENT_ITEM_NO	Component	For components and substitute components of a bill change.

ECO Implementation/Cancellation/Schedule/Reschedule

Application Code	XML Element Tag	Display Name	Comments
ENG	CHANGE_NOTICE	ECO	Used to query ECO-related events.
ENG	CHANGE_ORDER_TYPE	ECO Type	N/A
ENG	ECO_DEPARTMENT	ECO Department	N/A
ENG	REVISION	ECO Revision	N/A
ENG	PROJECT_NUMBER	Project	N/A
ENG	TASK_NUMBER	Task	N/A

Transfer to Manufacturing

Application Code	XML Element Tag	Display Name	Comments
ENG	NEW_ITEM_REVISION	Item Revision	For new item revisions.
ENG	NEW_ROUTING_REVISION	Routing Revision	For new routing revisions.
ENG	ALTERNATE_BOM_DESIGNATOR	Alternate Designator	N/A
ENG	CHANGE_NOTICE	ECO	N/A

Copy to Manufacturing

Application Code	XML Element Tag	Display Name	Comments
ENG	NEW_ITEM_REVISION	Item Revision	For new item revisions.
ENG	NEW_ROUTING_REVISION	Routing Revision	For new routing revisions.
ENG	ALTERNATE_BOM_DESIGNATOR	Alternate Designator	N/A
ENG	CHANGE_NOTICE	ECO	N/A

Mass Change Bills

Application Code	XML Element Tag	Display Name	Comments
ENG	ALTERNATE_BOM_DESIGNATOR	Alternate Designator	N/A

Event Data for Oracle Inventory

Oracle Workflow Seeded Data

Events

Name	Display Name	Description	Status	Owner Name	Owner Tag
oracle.apps.inv.itemCreatel	INV ERES Item Creation	INV ERES Item Creation	Enabled	Oracle Inventory	INV
oracle.apps.inv.itemUpdate	INV ERES Item Update	INV ERES Item Update	Enabled	Oracle Inventory	INV
oracle.apps.inv.itemRevisionEntry	INV ERES Item Revision Entry	INV ERES Item Revision Entry	Enabled	Oracle Inventory	INV
oracle.apps.inv.itemCrossRefEntry	INV ERES Item Cross Reference Entry	INV ERES Item Cross Reference Entry	Enabled	Oracle Inventory	INV
oracle.apps.inv.itemOrgAssignment	INV ERES Item Organization Assignment	INV ERES Item Organization Assignment	Enabled	Oracle Inventory	INV

Event Key

Event Name	User Event Key (Identifier)
oracle.apps.inv.itemCreate	Organization, Item
oracle.apps.inv.itemUpdate	Organization, Item
oracle.apps.inv.itemRevisionEntry	Organization, Item
oracle.apps.inv.itemCrossRefEntry	Item
oracle.apps.inv.itemOrgAssignment	Organization, Item

Event Subscription

All events subscriptions have the same values for the following fields:

- System = HM001
- Source Type = Local
- Phase = 0
- Status = Disabled
- Rule Data = Key
- Rule Function = EDR_PSIG_RULE.PSIG_RULE
- Priority = Normal

Event Filter	Parameters
oracle.apps.inv.itemCreate	EDR_XML_MAP_CODE=inviditm EDR_AME_TRANSACTION_TYPE=oracle.apps.inv.itemCreate
oracle.apps.inv.itemUpdate	EDR_XML_MAP_CODE=inviditm EDR_AME_TRANSACTION_TYPE=oracle.apps.inv.itemUpdate oracle.apps.inv.itemUpdate=IGNORE_SIGNATURE
oracle.apps.inv.itemRevisionEntry	EDR_XML_MAP_CODE=invidrev EDR_AME_TRANSACTION_TYPE=oracle.apps.inv.itemRevisionEntry
oracle.apps.inv.itemCrossRefEntry	EDR_XML_MAP_CODE=invidxrf EDR_AME_TRANSACTION_TYPE=oracle.apps.inv.itemCrossRefEntry
oracle.apps.inv.itemOrgAssignment	EDR_XML_MAP_CODE=invidasn EDR_AME_TRANSACTION_TYPE=oracle.apps.inv.itemOrgAssignment oracle.apps.inv.itemCreate=IGNORE_SIGNATURE

Oracle Approvals Management Seeded Data

Transaction Type

All transaction types listed below belong to the Oracle Inventory application.

Transaction Type ID	Transaction Type Description	Line Item Id Query String
oracle.apps.inv.itemCreate	INV ERES Item Creation	select category_id from mtl_item_categories where inventory_item_id = to_number(substrb(:transactionId,(instrb(:transactionId,'-') +1))) and organization_id = to_number(substrb(:transactionId,1, instrb(:transactionId,'-') -1)) order by category_id
oracle.apps.inv.itemUpdate	INV ERES Item Update	select category_id from mtl_item_categories where inventory_item_id = to_number(substrb(:transactionId,(instrb(:transactionId,'-') +1))) and organization_id = to_number(substrb(:transactionId,1, instrb(:transactionId,'-') -1)) order by category_id
oracle.apps.inv.itemRevisionEntry	INV ERES Item Revision Entry	select category_id from mtl_item_categories where inventory_item_id = to_number(substrb(:transactionId,(instrb(:transactionId,'-') +1))) and organization_id = to_number(substrb(:transactionId,1, instrb(:transactionId,'-') -1)) order by category_id
oracle.apps.inv.itemCrossRefEntry	INV ERES Item Cross Reference Entry	select distinct category_id from mtl_item_categories where inventory_item_id = to_number(:transactionId) order by category_id
oracle.apps.inv.itemOrgAssignment	INV ERES Item Organization Assignment	select category_id from mtl_item_categories where inventory_item_id = to_number(substrb(:transactionId,(instrb(:transactionId,'-') +1))) and organization_id = to_number(substrb(:transactionId,1, instrb(:transactionId,'-') -1)) order by category_id

Transaction Attributes

Although the user can define their own attributes for the transaction types listed above, the more commonly used attributes are seeded. The following attributes apply to all Oracle Inventory transaction types. Attributes specific to certain transaction types are listed in later tables, by transaction type.

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Mandatory Attribute	AL-LOW_DELETING_RULE_GENERATED_APPROVERS	boolean	whether to let the calling application (or its end users) delete approvers generated by the rules	Yes	false

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Mandatory Attribute	ALLOW_ REQUESTOR_ APPROVAL	boolean	whether to allow requestors to approve their own transactions (when the rules do so)	Yes	false
Mandatory Attribute	AT_LEAST_ ONE_RULE_ MUST_APPLY	boolean	whether to require that at least one rule apply to each transaction	Yes	false
Mandatory Attribute	EFFECTIVE_ RULE_DATE	date	the date that determines which rules are active	Yes	N/A
Mandatory Attribute	EVALUATE_ PRIORITIES_ PER_ITEM	boolean	whether to evaluate rule priorities per item under strict item evaluation	Yes	false
Mandatory Attribute	REJECTION_ RESPONSE	string	how AME responds to a rejection	Yes	STOP_ALL_ ITEMS
Mandatory Attribute	USE_RESTR ICTIVE_ ITEM_ EVALUAT ION	boolean	whether to require that the same item satisfy all item conditions in a given rule	Yes	false
Mandatory Attribute	USE_ WORKFLOW	boolean	whether OAM should log exceptions to the Workflow context stack	Yes	true
Mandatory Attribute	WORK- FLOW_ITEM_ KEY	string	the transaction's Workflow item key	Yes	N/A
Mandatory Attribute	WORK- FLOW_ITEM_ TYPE	string	the transaction's Workflow item type	Yes	N/A
Non-mandatory Header Attribute	ALLOW_ EMPTY_A PPROVAL_ GROUPS	boolean	whether to allow approval groups to have no members	Yes	false

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-mandatory Header Attribute	INCLUDE_ALL_JOB_LEVEL_APPROVERS	boolean	whether to include all approvers at a given job level	Yes	false
Non-Mandatory Header Attribute	JOB_LEVEL_NON_DEFAULT_STARTING_POINT_PERSON_ID	number	person ID of non-default first approver for job-level authority approval types	No	select null from dual
Non-mandatory Header Attribute	TRANSACTION_DATE	date	date transaction occurred	No	select ame_util.versionDateToString(sysdate) from dual
Non-mandatory Header Attribute	TRANSACTION_GROUP_ID	number	business-group ID in which transaction occurred	Yes	N/A
Non-mandatory Header Attribute	TRANSACTION_REQUESTOR_PERSON_ID	number	person ID of person initiating transaction, if any	Yes	N/A
Non-mandatory Header Attribute	TRANSACTION_REQUESTOR_USER_ID	number	user ID of user initiating transaction, if any	No	select fnd_global.user_id from dual
Non-mandatory Header Attribute	TRANSACTION_SET_OF_BOOKS_ID	number	set-of-books ID in which transaction occurred	Yes	N/A

Seeded transaction attributes for the following transaction types:

- NV ERES Item Creation (oracle.apps.inv.itemCreate)
- INV ERES Item Update (oracle.apps.inv.itemUpdate)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	INVENTORY_ITEM	string	Inventory item. Use individually or as a range. For example, trigger an event when an item is between ALUM0001 and ALUM1000.	No	select item FROM mtl_system_items_er3_v where inventory_item_id =to_number(substrb(:transactionId,(instrb(:transactionId,'-') +1)))

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	IS_BOM_ ALLOWED_ ITEM	string	Whether the Inventory Item is a BOM Allowed Item. For example, trigger the event only when BOM Allowed = Yes.	No	select bom_enabled_ flag from mtl_ system_items_b where inventory_item_id = to_number(substrb(: transactionId,(instrb(: transactionId,'-') +1))) and organization_id = to_number(substrb(: transactionId,1, instrb(: transactionId,'-') -1))
Non-mandatory Header Attribute	IS_BUILD_ IN_WIP_ ITEM	string	Whether the Inventory Item is a Build in WIP Item	No	select build_in_ wip_flag from mtl_ system_items_b where inventory_item_id = to_number(substrb(: transactionId,(instrb(: transactionId,'-') +1))) and organization_id = to_number(substrb(: transactionId,1, instrb(: transactionId,'-') -1))
Non-mandatory Header Attribute	IS_ CUSTOMER_ ORDER_ITEM	string	Whether the Inventory Item is a Customer Order item	No	select customer_ order_flag from mtl_ system_items_b where inventory_item_id = to_number(substrb(: transactionId,(instrb(: transactionId,'-') +1))) and organization_id = to_number(substrb(: transactionId,1, instrb(: transactionId,'-') -1))
Non-Mandatory Header Attribute	IS_ENG INEERING_ ITEM	string	Whether the Inventory Item is a Engineering item	No	select eng_item_flag from mtl_system_ items_b where inventory_item_id = to_number(substrb(: transactionId,(instrb(: transactionId,'-') +1))) and organization_id = to_number(substrb(: transactionId,1, instrb(: transactionId,'-') -1))
Non-Mandatory Header Attribute	IS_ INVENTORY_ ITEM	string	Whether the Item is a Inventory Item	No	select inventory_ item_flag from mtl_ system_items_b where inventory_item_id = to_number(substrb(: transactionId,(instrb(: transactionId,'-') +1))) and organization_id = to_number(substrb(: transactionId,1, instrb(: transactionId,'-') -1))

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	IS_PURCHASING_ITEM	string	Whether the Inventory Item is a Purchasing Item	No	select purchasing_item_flag from mtl_system_items_b where inventory_item_id = to_number(substrb(:transactionId,(instrb(:transactionId,'-') +1))) and organization_id = to_number(substrb(:transactionId,1, instrb(:transactionId,'-') -1))
Non-Mandatory Header Attribute	IS_SHIPPABLE_ITEM	string	Whether the Inventory Item is a Shippable Item	No	select shippable_item_flag from mtl_system_items_b where inventory_item_id = to_number(substrb(:transactionId,(instrb(:transactionId,'-') +1))) and organization_id = to_number(substrb(:transactionId,1, instrb(:transactionId,'-') -1))
Non-Mandatory Header Attribute	IS_STOCKABLE_ITEM	string	Whether the Inventory Item is a Stockable Item	No	select stock_enabled_flag from mtl_system_items_b where inventory_item_id = to_number(substrb(:transactionId,(instrb(:transactionId,'-') +1))) and organization_id = to_number(substrb(:transactionId,1, instrb(:transactionId,'-') -1))
Non-Mandatory Header Attribute	IS_TRANSACTABLE_ITEM	string	Whether the Inventory Item is a Transactable Item	No	select mtl_transactions_enabled_flag from mtl_system_items_b where inventory_item_id = to_number(substrb(:transactionId,(instrb(:transactionId,'-') +1))) and organization_id = to_number(substrb(:transactionId,1, instrb(:transactionId,'-') -1))
Non-Mandatory Header Attribute	ITEM_CATALOG_GROUP	string	Catalog group for the inventory item	No	select item_catalog_group from mtl_system_items_er3_v where inventory_item_id =to_number(substrb(:transactionId,(instrb(:transactionId,'-') +1)))

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-mandatory Header Attribute	ORGANIZATION_CODE	string	Organization Code.	No	select organization_code FROM mtl_parameters where organization_id = to_number(substrb(:transactionId,1, instrb(:transactionId,'-') -1))
Non-mandatory Header Attribute	TOP_SUPERVISOR_PERSON_ID	number	person ID of the top person in the HR supervisory hierarchy	Yes	N/A
Non-mandatory Header Attribute	TRANSACTION_ORG_ID	number	org ID in which transaction occurred	Yes	N/A
Non-mandatory Line Item Attribute	ITEM_CATEGORY	string	Item Category	No	select concatenated_segments from MTL_CATEGORIES_B_KFV where category_id in (select category_id from mtl_item_categories where inventory_item_id = to_number(substrb(:transactionId,(instrb(:transactionId,'-') +1))) and organization_id = to_number(substrb(:transactionId,1, instrb(:transactionId,'-') -1))) order by category_id

Seeded transaction attributes for the following transaction types:

- INV ERES Item Revision Entry (oracle.apps.inv.itemRevisionEntry)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	INVENTORY_ITEM	string	Inventory item. Use individually or as a range. For example, trigger an event when an item is between ALUM0001 and ALUM1000.	No	select item FROM mtl_system_items_er3_v where inventory_item_id =to_number(substrb(:transactionId,(instrb(:transactionId,'-') +1)))

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	IS_ENG INEERING_ ITEM	string	Whether the Inventory Item is a Engineering item	No	select eng_item_flag from mtl_system_ items_b where inventory_item_id = to_number(substrb(: transactionId,(instrb(: transactionId,'-') +1))) and organization_id = to_number(substrb(: transactionId,1, instrb(: transactionId,'-') -1))
Non-Mandatory Header Attribute	ITEM_ CATALOG_ GROUP	string	Catalog group for the inventory item	No	select item_catalog_ group from mtl_ system_items_er3_v where inventory_ item_id =to_ number(substrb(: transactionId,(instrb(: transactionId,'-') +1)))
Non-mandatory Header Attribute	ORGANIZAT ION_CODE	string	Organization Code.	No	select organization_ code FROM mtl_ parameters where organization_id = to_number(substrb(: transactionId,1, instrb(: transactionId,'-') -1))
Non-mandatory Header Attribute	TOP_SU PERVISOR_ PERSON_ID	number	person ID of the top person in the HR supervisory hierarchy	Yes	N/A
Non-mandatory Header Attribute	TRANSACT ION_ORG_ID	number	org ID in which transaction occurred	Yes	N/A
Non-mandatory Line Item Attribute	ITEM_ CATEGORY	string	Item Category	No	select concatenated_ segments from MTL_ CATEGORIES_B_KFV where category_id in (select category_ id from mtl_item_ categories where inventory_item_id = to_number(substrb(: transactionId,(instrb(: transactionId,'-') +1))) and organization_id = to_number(substrb(: transactionId,1, instrb(: transactionId,'-') -1))) order by category_id

Seeded transaction attributes for the following transaction types:

- INV ERES Item Cross Reference Entry (oracle.apps.inv.itemCrossRefEntry)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	INVENTORY_ ITEM	string	Inventory item. Use individually or as a range. For example, trigger an event when an item is between ALUM0001 and ALUM1000.	No	select item FROM mtl_system_items_er3_ v where inventory_ item_id = to_number(: transactionId)
Non-Mandatory Header Attribute	ITEM_ CATALOG_ GROUP	string	Catalog group for the inventory item	No	select item_catalog_ group from mtl_ system_items_er3_ v where inventory_ item_id =to_number(: transactionId)
Non-mandatory Header Attribute	TRANSACT ION_ORG_ID	number	org ID in which transaction occurred	Yes	N/A
Non-mandatory Line Item Attribute	ITEM_ CATEGORY	string	Item Category	No	select concatenated_ segments from MTL_ CATEGORIES_B_KFV where category_id in (select distinct category_id from mtl_ item_categories where inventory_item_id = to_ number(:transactionId)) order by category_id

Seeded transaction attributes for the following transaction types:

- INV ERES Item Organization Assignment (oracle.apps.inv.itemOrgAssignment)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	INVENTORY_ITEM	string	Inventory item. Use individually or as a range. For example, trigger an event when an item is between ALUM0001 and ALUM1000.	No	select item FROM mtl_system_items_er3_v where inventory_item_id = to_number(substrb(:transactionId,(instrb(:transactionId,'-') +1)))
Non-Mandatory Header Attribute	ITEM_CATALOG_GROUP	string	Catalog group for the inventory item	No	select item_catalog_group from mtl_system_items_er3_v where inventory_item_id = to_number(substrb(:transactionId,(instrb(:transactionId,'-') +1)))
Non-mandatory Header Attribute	ORGANIZATION_CODE	string	Organization Code.	No	select organization_code from mtl_parameters where organization_id = to_number(substrb(:transactionId,1, instrb(:transactionId,'-') -1))
Non-mandatory Header Attribute	TOP_SUPERVISOR_PERSON_ID	number	person ID of the top person in the HR supervisory hierarchy	Yes	N/A
Non-mandatory Header Attribute	TRANSACTION_ORG_ID	number	org ID in which transaction occurred	Yes	N/A
Non-mandatory Line Item Attribute	ITEM_CATEGORY	string	Item Category	No	select concatenated_segments from MTL_CATEGORIES_B_KFV where category_id in (select category_id from mtl_item_categories where inventory_item_id = to_number(substrb(:transactionId,(instrb(:transactionId,'-') +1))) and organization_id = to_number(substrb(:transactionId,1, instrb(:transactionId,'-') -1))

Define Rules/Associate Approvers

No rules or approvers are seeded. As part of the implementation, you must complete these activities (see: Setting Up Oracle Approvals Management, *Oracle E-Records Implementation Guide*).

Oracle E-Records Seeded Data: Generic Query Attributes

Use generic query attributes to search for e-records and their associated documents in the Evidence Store. For instructions on how to search, see Evidence Store, *Oracle E-Records Implementation Guide*. For all attributes listed below:

Query Element Type = Generic

Item Creation and Item Update

Application Code	XML Element Tag	Display Name	Comments
INV	ITEM_CATALOG_GROUP	Catalog Group	N/A
INV	ITEM	Item	For items and new items.
INV	INVENTORY_ITEM_FLAG	Inventory Item Flag	Shared by item creation and item update events.
INV	STOCK_ENABLED_FLAG	Stockable	Shared by item creation and item update events.
INV	MTL_TRANSACTIONS_ENABLED_FLAG	Transactable	Shared by item creation and item update events.
INV	BOM_ENABLED_FLAG	BOM Allowed	Shared by item creation and item update events.
INV	PURCHASING_ITEM_FLAG	Purchased	Shared by item creation and item update events.
INV	BUILD_IN_WIP_FLAG	Build in WIP	Shared by item creation and item update events.
INV	CUSTOMER_ORDER_FLAG	Customer Ordered	Shared by item creation and item update events.

Item Revision

Application Code	XML Element Tag	Display Name	Comments
INV	ITEM_CATALOG_GROUP	Catalog Group	N/A
INV	ITEM	Item	N/A
INV	ITEM_REVISION	Item Revision	N/A
ENG	ECO	ECO	Use this query element only when an Item Revisions e-record is part of an ECO.

Item Organization Assignment

Application Code	XML Element Tag	Display Name	Comments
INV	ITEM_CATALOG_GROUP	Catalog Group	N/A
INV	ITEM	Item	N/A

Event Data for Oracle Bills of Material

Oracle Workflow Seeded Data

Events

Name	Display Name	Description	Status	Owner Name	Owner Tag
oracle.apps.bom.billCreate	BOM Bill of Materials Create	ERES Event for BOM Bill of Materials Create	Enabled	Oracle Bills of Material	BOM
oracle.apps.bom.billUpdate	BOM Bill of Materials Update	ERES Event for BOM Bill of Materials Update	Enabled	Oracle Bills of Material	BOM
oracle.apps.bom.routingCreate	BOM Routing Create	ERES Event for BOM Routing Create	Enabled	Oracle Bills of Material	BOM
oracle.apps.bom.routingUpdate	BOM Routing Update	ERES Event for BOM Routing Update	Enabled	Oracle Bills of Material	BOM

Event Key

Event Name	User Event Key (Identifier)
Oracle.apps.bom.billCreate	Organization, Item, Alternate BOM Designator
Oracle.apps.bom.billUpdate	Organization, Item, Alternate BOM Designator
Oracle.apps.bom.routingCreate	Organization, Item, Alternate Routing Designator
Oracle.apps.bom.routingUpdate	Organization, Item, Alternate Routing Designator

Event Subscription

All events subscriptions have the same values for the following fields:

- System = HM001
- Source Type = Local
- Phase = 0

- Status = Disabled
- Rule Data = Key
- Rule Function = EDR_PSIG_RULE.PSIG_RULE
- Priority = Normal

Event Filter	Parameters
oracle.apps.bom.billCreate	EDR_XML_MAP_CODE=oracle.apps.bom.bill EDR_AME_TRANSACTION_TYPE=oracle.apps.bom.billCreate oracle.apps.bom.billUpdate=IGNORE_SIGNATURE
oracle.apps.bom.billUpdate	EDR_XML_MAP_CODE=oracle.apps.bom.bill EDR_AME_TRANSACTION_TYPE=oracle.apps.bom.billUpdate oracle.apps.bom.billUpdate=IGNORE_SIGNATURE
oracle.apps.bom.routingCreate	EDR_XML_MAP_CODE=oracle.apps.bom.routing EDR_AME_TRANSACTION_TYPE=oracle.apps.bom.routingCreate
oracle.apps.bom.routingUpdate	EDR_XML_MAP_CODE=oracle.apps.bom.routing EDR_AME_TRANSACTION_TYPE=oracle.apps.bom.routingUpdate

Oracle Approvals Management Seeded Data

Transaction Type

All transaction types listed below belong to the Oracle Bills of Material application.

Transaction Type ID	Transaction Type Description	Line Item Id Query String
Oracle.apps.bom.billCreate	BOM ERES Bill of Materials Creation	select category_id from mtl_item_ categories where inventory_item_id = (select assembly_item_id from BOM_BILL_OF_MATERIALS WHERE bill_sequence_id = TO_NUMBER(: transactionId)) AND organization_id = (select organization_id from BOM_BILL_ OF_MATERIALS WHERE bill_sequence_ id = TO_NUMBER(:transactionId)) order by category_id
Oracle.apps.bom.billUpdate	BOM ERES Bill of Materials Update	select category_id from mtl_item_ categories where inventory_item_id = (select assembly_item_id from BOM_BILL_OF_MATERIALS WHERE bill_sequence_id = TO_NUMBER(: transactionId)) AND organization_id = (select organization_id from BOM_BILL_ OF_MATERIALS WHERE bill_sequence_ id = TO_NUMBER(:transactionId)) order by category_id
Oracle.apps.bom. routingCreate	BOM ERES Operational Routing Creation	select category_id from mtl_item_ categories where inventory_item_id = (select assembly_item_id from bom_ operational_routings WHERE routing_ sequence_id = TO_NUMBER(:transaction Id)) AND organization_id = (select organization_id from bom_operational_ routings WHERE routing_sequence_id = TO_NUMBER(:transactionId)) order by category_id
Oracle.apps.bom. routingUpdate	BOM ERES Operational Routing Update	select category_id from mtl_item_ categories where inventory_item_id = (select assembly_item_id from bom_ operational_routings WHERE routing_ sequence_id = TO_NUMBER(:transaction Id)) AND organization_id = (select organization_id from bom_operational_ routings WHERE routing_sequence_id = TO_NUMBER(:transactionId)) order by category_id

Transaction Attributes

Although the user can define their own attributes for the transaction types listed above, the more commonly used attributes are seeded. The following attributes apply to all Oracle Bills of Material transaction types. Attributes specific to certain transaction types are listed in later tables, by transaction type.

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Mandatory Attribute	AL-LOW_DELET ING_RULE_ GENERATED_ APPROVERS	boolean	whether to let the calling application (or its end users) delete approvers generated by the rules	Yes	false
Mandatory Attribute	ALLOW_ REQUESTOR_ APPROVAL	boolean	whether to allow requestors to approve their own transactions (when the rules do so)	Yes	false
Mandatory Attribute	AT_LEAST_ ONE_RULE_ MUST_APPLY	boolean	whether to require that at least one rule apply to each transaction	Yes	false
Mandatory Attribute	EFFECTIVE_ RULE_DATE	date	the date that determines which rules are active	Yes	N/A
Mandatory Attribute	EVALUATE_ PRIORITIES_ PER_ITEM	boolean	whether to evaluate rule priorities per item under strict item evaluation	Yes	false
Mandatory Attribute	REJECTION_ RESPONSE	string	how AME responds to a rejection	Yes	STOP_ALL_ ITEMS
Mandatory Attribute	USE_RESTR ICTIVE_ ITEM_ EVALUAT ION	boolean	whether to require that the same item satisfy all item conditions in a given rule	Yes	false
Mandatory Attribute	USE_ WORKFLOW	boolean	whether OAM should log exceptions to the Workflow context stack	Yes	true
Mandatory Attribute	WORK-FLOW_ITEM_ KEY	string	the transaction's Workflow item key	Yes	N/A

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Mandatory Attribute	WORK-FLOW_ITEM_TYPE	string	the transaction's Workflow item type	Yes	N/A
Non-mandatory Header Attribute	ALLOW_EMPTY_APPROVAL_GROUPS	boolean	whether to allow approval groups to have no members	Yes	false
Non-mandatory Header Attribute	INCLUDE_ALL_JOB_LEVEL_APPROVERS	boolean	whether to include all approvers at a given job level	Yes	false
Non-mandatory Header Attribute	JOB_LEVEL_NON_DEFAULT_STARTING_POINT_PERSON_ID	number	person ID of non-default first approver for job-level authority approval types	No	select null from dual
Non-mandatory Header Attribute	TOP_SUPERVISOR_PERSON_ID	number	person ID of the top person in the HR supervisory hierarchy	Yes	N/A
Non-mandatory Header Attribute	TRANSACTION_DATE	date	person ID of the top person in the HR supervisory hierarchy	No	SELECT ame_util.versionDateToString(CREATION_DATE) FROM bom_bill_of_materials WHERE bill_sequence_id = TO_NUMBER(:transactionId)
Non-mandatory Header Attribute	TRANSACTION_GROUP_ID	number	business-group ID in which transaction occurred	Yes	N/A
Non-mandatory Header Attribute	TRANSACTION_ORG_ID	number	org ID in which transaction occurred	Yes	N/A
Non-mandatory Header Attribute	TRANSACTION_REQUESTOR_PERSON_ID	number	person ID of person initiating transaction, if any	Yes	N/A

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-mandatory Header Attribute	TRANSACTION_REQUESTOR_USER_ID	number	user ID of user initiating transaction, if any	No	SELECT last_updated_by FROM bom_bill_of_materials WHERE bill_sequence_id = TO_NUMBER(:transactionId)
Non-mandatory Header Attribute	TRANSACTION_SET_OF_BOOKS_ID	number	set-of-books ID in which transaction occurred	Yes	N/A

Seeded transaction attributes for the following transaction types:

- BOM ERES Bill of Materials Creation (Oracle.apps.bom.billCreate)
- BOM ERES Bill of Materials Update (Oracle.apps.bom.billUpdate)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	ALTERNATE_DESIGNATOR	string	Alternate Designator. Triggers the event for certain alternate bill designators or for the primary bill, for example, when the Alternate Designator is null.	No	SELECT alternate_bom_designator FROM bom_bill_of_materials WHERE bill_sequence_id = TO_NUMBER(:transactionId)
Non-Mandatory Header Attribute	ASSEMBLY_ITEM	string	Assembly Item.	No	select assembly_item_no FROM BOM_BILL_OF_MATERIALS_ERV WHERE bill_sequence_id = TO_NUMBER(:transactionId)
Non-Mandatory Header Attribute	ORGANIZATION_CODE	string	Organization Code	No	select organization_code FROM mtl_parameters where organization_id = (select organization_id from BOM_BILL_OF_MATERIALS WHERE bill_sequence_id = TO_NUMBER(:transactionId))
Non-mandatory Line Item Attribute	ITEM CATEGORY	string	The category set defined for functional area 'Order Mgmt' in Inventory	No	select concatenated_segments from MTL_CATEGORIES_B_KFV where category_id in (select category_id from mtl_item_categories where inventory_item_id = (select assembly_item_id from BOM_BILL_OF_MATERIALS WHERE bill_sequence_id = TO_NUMBER(:transactionId)) AND organization_id = (select organization_id from BOM_BILL_OF_MATERIALS WHERE bill_sequence_id = TO_NUMBER(:transactionId))) order by category_id

Seeded transaction attributes for the following transaction types:

- BOM ERES Operational Routing Creation (Oracle.apps.bom.routingCreate)
- BOM ERES Operational Routing Update (Oracle.apps.bom.routingUpdate)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	ALTERNATE_DES IGNATOR	string	Alternate Designator. Triggers an event for certain alternate routing designators or triggers an event only for aprimary routing, for example, when the . Alternate Designator is null.	No	SELECT alternate_routing_designator FROM bom_operational_routings WHERE routing_sequence_id = to_number(:transaction Id)
Non-Mandatory Header Attribute	ASSEMBLY_ITEM	string	Assembly Item.	No	select ASSEMBLY_ITEM FROM BOM_OPERATIONAL_ROUTINGS_ERV WHERE routing_sequence_id = TO_NUMBER(:transactionId)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	ORGANIZATION_CODE	string	Organization Code	No	select organization_code FROM mtl_parameters where organization_id = (select organization_id from bom_operational_routings WHERE routing_sequence_id = TO_NUMBER(:transactionId))
Non-mandatory Line Item Attribute	ITEM CATEGORY	string	The category set defined for functional area 'Order Mgmt' in Inventory	No	select concatenated_segments from MTL_CATEGORIES_B_KFV where category_id in (select category_id from mtl_item_categories where inventory_item_id = (select assembly_item_id from bom_operational_routings WHERE routing_sequence_id = TO_NUMBER(:transactionId)) AND organization_id = (select organization_id from bom_operational_routings WHERE routing_sequence_id = TO_NUMBER(:transactionId))) order by category_id

Define Rules/Associate Approvers

No rules or approvers are seeded. As part of the implementation, you must complete these activities (see: Setting Up Oracle Approvals Management, *Oracle E-Records Implementation Guide*).

Oracle E-Records Seeded Data: Generic Query Attributes

Use generic query attributes to search for e-records and their associated documents in the Evidence Store. For instructions on how to search, see Evidence Store, *Oracle E-Records Implementation Guide*. For all attributes listed below:

Query Element Type = Generic

Bill Creation/Bill Update

Application Code	XML Element Tag	Display Name	Comments
BOM	ASSEMBLY_ITEM_NO	Item	N/A
BOM	ASSEMBLY_ITEM_REVISION	Item Revision	N/A
BOM	ALTERNATE_BOM_DESIGNATOR	Alternate Designator	N/A
BOM	COMPONENT_ITEM_NO	Component	For bill components and substitute components.

Routing Creation/Routing Update

Application Code	XML Element Tag	Display Name	Comments
BOM	ASSEMBLY_ITEM	Item	N/A
BOM	CURRENT_REVISION	Item Revision	N/A
BOM	ALTERNATE_ROUTING_DESIGNATOR	Alternate Designator	N/A
BOM	PROCESS_REVISION	Routing Revision	N/A
BOM	STANDARD_OPERATION_CODE	Standard Operation Code	N/A
BOM	RESOURCE_CODE	Resource Code	For resources and alternate resource codes.

Event Data for Oracle Work in Process

Oracle Workflow Seeded Data

Events

Name	Display Name	Description	Status	Owner Name	Owner Tag
oracle.apps.wip.job.assembly.complete	WIP Job Assembly Completion	WIP job assembly completion or assembly return	Enabled	Oracle Work in Process	WIP
oracle.apps.wip.job.assembly.move	WIP Job Assembly Move	WIP job operation move	Enabled	Oracle Work in Process	WIP
oracle.apps.wip.job.material.transact	WIP Job Material Transaction	WIP material issue, material return, negative issue, or negative return for a discrete job	Enabled	Oracle Work in Process	WIP

Event Key

Event Name	User Event Key (Identifier)
oracle.apps.wip.job.assembly.complete	Job, Transaction Type
oracle.apps.wip.job.assembly.move	Job, Transaction Type
oracle.apps.wip.job.material.transact	Job, Transaction Type

Event Subscription

All events subscriptions have the same values for the following fields:

- System = HM001
- Source Type = Local
- Phase = 0
- Status = Disabled
- Rule Data = Key
- Rule Function = EDR_PSIG_RULE.PSIG_RULE
- Priority = Normal

Event Filter	Parameters
oracle.apps.wip.job.assembly.complete	EDR_XML_MAP_CODE=wipcmpmp EDR_AME_TRANSACTION_TYPE=oracle.apps.wip.job.assembly.complete
oracle.apps.wip.job.assembly.move	EDR_XML_MAP_CODE=wipmovmp EDR_AME_TRANSACTION_TYPE=oracle.apps.wip.job.assembly.move
oracle.apps.wip.job.material.transact	EDR_XML_MAP_CODE=wipmtlmp EDR_AME_TRANSACTION_TYPE=oracle.apps.wip.job.material.transact

Oracle Approvals Management Seeded Data

Transaction Type

All transaction types listed below belong to the Oracle Work in Process application.

Transaction Type ID	Transaction Type Description	Line Item Id Query String
oracle.apps.wip.job.assembly.complete	WIP ERES Job Assembly Completion	select transaction_id from mtl_material_transactions where transaction_set_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and transaction_source_id = substr(:transactionId, (instr(:transactionId, '-')+1), (instr(:transactionId, '-', 1, 2)- (instr(:transactionId, '-')+1))) and transaction_type_id in (44,17) order by transaction_id
oracle.apps.wip.job.assembly.move	WIP ERES Job Assembly Move	N/A
oracle.apps.wip.job.material.transact	WIP ERES Job Material Transaction	select transaction_id from mtl_material_transactions where transaction_set_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and transaction_source_id = substr(:transactionId, (instr(:transactionId, '-')+1)) order by transaction_id

Transaction Attributes

Although the user can define their own attributes for the transaction types listed above, the more commonly used attributes are seeded. The following attributes apply to all Oracle Work in Process transaction types. Attributes specific to certain transaction types are listed in later tables, by transaction type.

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Mandatory Attribute	AL-LOW_DELET ING_RULE_ GENERATED_ APPROVERS	boolean	whether to let the calling application (or its end users) delete approvers generated by the rules	Yes	false
Mandatory Attribute	ALLOW_ REQUESTOR_ APPROVAL	boolean	whether to allow requestors to approve their own transactions (when the rules do so)	Yes	false
Mandatory Attribute	AT_LEAST_ ONE_RULE_ MUST_APPLY	boolean	whether to require that at least one rule apply to each transaction	Yes	false
Mandatory Attribute	EFFECTIVE_ RULE_DATE	date	the date that determines which rules are active	Yes	N/A
Mandatory Attribute	EVALUATE_ PRIORITIES_ PER_ITEM	boolean	whether to evaluate rule priorities per item under strict item evaluation	Yes	false
Mandatory Attribute	REJECTION_ RESPONSE	string	how AME responds to a rejection	Yes	STOP_ALL_ ITEMS
Mandatory Attribute	USE_RESTR ICTIVE_ ITEM_ EVALUAT ION	boolean	whether to require that the same item satisfy all item conditions in a given rule	Yes	false
Mandatory Attribute	USE_ WORKFLOW	boolean	whether OAM should log exceptions to the Workflow context stack	Yes	true
Mandatory Attribute	WORK-FLOW_ITEM_ KEY	string	the transaction's Workflow item key	Yes	N/A

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Mandatory Attribute	WORK-FLOW_ITEM_TYPE	string	the transaction's Workflow item type	Yes	N/A
Non-mandatory Header Attribute	ALLOW_EMPTY_APPROVAL_GROUPS	boolean	whether to allow approval groups to have no members	Yes	false
Non-mandatory Header Attribute	INCLUDE_ALL_JOB_LEVEL_APPROVERS	boolean	whether to include all approvers at a given job level	Yes	false
Non-mandatory Header Attribute	JOB_LEVEL_NON_DEFAULT_STARTING_POINT_PERSON_ID	number	person ID of non-default first approver for job-level authority approval types	No	select null from dual
Non-mandatory Header Attribute	TOP_SUPERVISOR_PERSON_ID	number	person ID of the top person in the HR supervisory hierarchy	Yes	N/A
Non-mandatory Header Attribute	TRANSACTION_GROUP_ID	number	business-group ID in which transaction occurred	Yes	N/A
Non-mandatory Header Attribute	TRANSACTION_ORG_ID	number	org ID in which transaction occurred	Yes	N/A
Non-mandatory Header Attribute	TRANSACTION_REQUESTOR_PERSON_ID	number	person ID of person initiating transaction, if any	Yes	N/A
Non-mandatory Header Attribute	TRANSACTION_SET_OF_BOOKS_ID	number	set-of-books ID in which transaction occurred	Yes	N/A

Seeded transaction attributes for the following transaction types:

- WIP ERES Job Material Transaction (oracle.apps.wip.job.material.transact)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-mandatory Header Attribute	TRANSACTION_DATE	date	date transaction occurred	No	select ame_util.versionDateToString(LAST_UPDATE_DATE) from mtl_material_transactions where transaction_set_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and transaction_source_id = substr(:transactionId, (instr(:transactionId, '-')+1)) and rownum = 1
Non-mandatory Header Attribute	TRANSACTION_REQUESTOR_USER_ID	number	user ID of user initiating transaction, if any	No	select fnd_number.number_to_canonical(last_updated_by) from mtl_material_transactions where transaction_set_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and transaction_source_id = substr(:transactionId, (instr(:transactionId, '-')+1)) and rownum = 1
Non-Mandatory Header Attribute	WIP_ASSEMBLY	string	Approval based on assembly	No	select msik.concatenated_segments from mtl_system_items_kfv msik, mtl_material_transactions mmt, wip_discrete_jobs wdj where mmt.transaction_set_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and mmt.transaction_source_id = substr(:transactionId, (instr(:transactionId, '-')+1)) and mmt.transaction_source_id = wdj.wip_entity_id and wdj.organization_id = msik.organization_id and wdj.primary_item_id = msik.inventory_item_id and rownum = 1

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	WIP_JOB	string	Approval based on job.	No	select we.wip_entity_name from wip_entities we, mtl_material_transactions mmt where mmt.transaction_set_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and mmt.transaction_source_id = substr(:transactionId, (instr(:transactionId, '-')+1)) and we.wip_entity_id = mmt.transaction_source_id and rownum = 1
Non-Mandatory Header Attribute	WIP_JOB_PROJECT	string	Approval based on job's project	No	select pjm_project.all_proj_idtonum(wdj.project_id) from mtl_material_transactions mmt, wip_discrete_jobs wdj where mmt.transaction_set_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and mmt.transaction_source_id = substr(:transactionId, (instr(:transactionId, '-')+1)) and mmt.transaction_source_id = wdj.wip_entity_id and rownum = 1
Non-mandatory Header Attribute	WIP_JOB_TASK	string	Approval based on job's task	No	select pjm_project.all_task_idtonum(wdj.task_id) from mtl_material_transactions mmt, wip_discrete_jobs wdj where mmt.transaction_set_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and mmt.transaction_source_id = substr(:transactionId, (instr(:transactionId, '-')+1)) and mmt.transaction_source_id = wdj.wip_entity_id and rownum = 1

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-mandatory Header Attribute	WIP_MATERIAL_TRANSACTION_TYPE	string	Txn Type: WIP component issue,WIP Component Return,WIP Neg Comp Issue,WIP Neg Comp Return	No	select transaction_type_name from mtl_material_transactions mmt, mtl_transaction_types type where mmt.transaction_set_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and mmt.transaction_source_id = substr(:transactionId, (instr(:transactionId, '-')+1)) and type.transaction_type_id = mmt.transaction_type_id and rownum = 1
Non-mandatory Line Item Attribute	WIP_COMPONENT	string	Approval based on component item	No	select msik.concatenated_segments from mtl_system_items_kfv msik, mtl_material_transactions mmt where mmt.transaction_set_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and mmt.transaction_source_id = substr(:transactionId, (instr(:transactionId, '-')+1)) and mmt.transaction_id in (select transaction_id from mtl_material_transactions where transaction_set_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and transaction_source_id = substr(:transactionId, (instr(:transactionId, '-')+1))) and mmt.organization_id = msik.organization_id and mmt.inventory_item_id = msik.inventory_item_id order by mmt.transaction_id

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-mandatory Line Item Attribute	WIP_DEPARTMENT	string	Approval based on department	No	select bd.department_code from mtl_material_transactions mmt, bom_departments bd where mmt.transaction_set_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and mmt.transaction_source_id = substr(:transactionId, (instr(:transactionId, '-')+1)) and mmt.transaction_id in (select transaction_id from mtl_material_transactions where transaction_set_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and transaction_source_id = substr(:transactionId, (instr(:transactionId, '-')+1))) and bd.department_id = mmt.department_id order by mmt.transaction_id
Non-mandatory Line Item Attribute	WIP_LOCATOR	string	Approval based on locator	No	select inv_project.get_locator(locator_id, organization_id) from mtl_material_transactions mmt where mmt.transaction_id in (select transaction_id from mtl_material_transactions where transaction_set_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and transaction_source_id = substr(:transactionId, (instr(:transactionId, '-')+1))) order by mmt.transaction_id

Attribute Category	Attribute Name	At-tribute Type	Description	Static Usage	Usage
Non-mandatory Line Item Attribute	WIP_OP_ SEQ_NUM	number	Approval based on operation seq num	No	select fnd_number. number_to_ canonical(mmt. operation_seq_num) from mtl_material_ transactions mmt where mmt.transaction_set_id = substr(:transaction Id, 1, (instr(:transaction Id, '-')-1)) and mmt. transaction_source_id = substr(:transaction Id, (instr(:transaction Id, '-')+1)) and mmt. transaction_id in (select transaction_id from mtl_ material_transactions where transaction_set_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and transaction_ source_id = substr(: transactionId, (instr(: transactionId, '-')+1))) order by mmt. transaction_id
Non-mandatory Line Item Attribute	WIP_SUB INVENTORY	string	Approval based on subinventory	No	select mmt.subinventory_ code from mtl_material_ transactions mmt where mmt.transaction_id in (select transaction_id from mtl_material_ transactions where transaction_set_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and transaction_ source_id = substr(: transactionId, (instr(: transactionId, '-')+1))) order by mmt. transaction_id

Seeded transaction attributes for the following transaction types:

- WIP ERES Job Assembly Move (oracle.apps.wip.job.assembly.move)

Attribute Category	Attribute Name	At-tribute Type	Description	Static Usage	Usage
Non-mandatory Header Attribute	TRANSACTION_DATE	date	date transaction occurred	No	select ame_util.versionDateToString(LAST_UPDATE_DATE) from wip_move_transactions where transaction_id = :transactionId
Non-mandatory Header Attribute	TRANSACTION_REQUESTOR_USER_ID	number	user ID of user initiating transaction, if any	No	select fnd_number.number_to_canonical(last_updated_by) from wip_move_transactions where transaction_id = :transactionId
Non-Mandatory Header Attribute	WIP_ASSEMBLY	string	Approval based on assembly	No	select msik.concatenated_segments from mtl_system_items_kfv msik, wip_move_transactions wmt where wmt.transaction_id = :transactionId and wmt.primary_item_id = msik.inventory_item_id and wmt.organization_id = msik.organization_id
Non-Mandatory Header Attribute	WIP_ASSEMBLY_LOCATOR	string	Approval based on locator for assembly completion/return	No	select inv_project.get_locator(wdj.completion_locator_id, wdj.organization_id) from wip_move_transactions wmt, wip_discrete_jobs wdj where wmt.transaction_id = :transactionId and wmt.wip_entity_id = wdj.wip_entity_id
Non-Mandatory Header Attribute	WIP_ASSEMBLY_SUB INVENTORY	string	Approval based on subinventory for assembly completion/return	No	select wdj.completion_subinventory from wip_move_transactions wmt, wip_discrete_jobs wdj where wmt.transaction_id = :transactionId and wmt.wip_entity_id = wdj.wip_entity_id
Non-Mandatory Header Attribute	WIP_FROM_DEPARTMENT	string	Approval based on department from which assembly was moved	No	select fm_department_code from wip_move_transactions_v wmt where wmt.transaction_id = :transactionId

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	WIP_FROM_INTRAOPERATION_STEP	string	Approval based on intraoperation step from which assembly was moved	No	select wmt.fm_intraoperation_step_meaning from wip_move_transactions_v wmt where wmt.transaction_id = :transactionId
Non-Mandatory Header Attribute	WIP_FROM_OPERATION_CODE	string	Approval based on operation code from which assembly was moved	No	select fm_operation_code from wip_move_transactions_v wmt where wmt.transaction_id = :transactionId
Non-Mandatory Header Attribute	WIP_FROM_OPERATION_SEQ_NUM	number	Approval based on operation sequence from which assembly was moved	No	select fnd_number.number_to_canonical(fm_operation_seq_num) from wip_move_transactions wmt where wmt.transaction_id = :transactionId
Non-Mandatory Header Attribute	WIP_JOB	string	Approval based on job.	No	select we.wip_entity_name from wip_move_transactions wmt, wip_entities we where wmt.transaction_id = :transactionId and wmt.wip_entity_id = we.wip_entity_id
Non-Mandatory Header Attribute	WIP_JOB_PROJECT	string	Approval based on job's project	No	select pjm_project.all_proj_idtonum(wdj.project_id) from wip_move_transactions wmt, wip_discrete_jobs wdj where wmt.transaction_id = :transactionId and wmt.wip_entity_id = wdj.wip_entity_id
Non-mandatory Header Attribute	WIP_JOB_TASK	string	Approval based on job's task	No	select pjm_project.all_task_idtonum(wdj.task_id) from wip_move_transactions wmt, wip_discrete_jobs wdj where wmt.transaction_id = :transactionId and wmt.wip_entity_id = wdj.wip_entity_id
Non-mandatory Header Attribute	WIP_MOVE_TRANSACTION_TYPE	string	Transaction type: Move transaction, Move and completion transaction, Return and move transaction	No	select wip_move_validator.move_txn_type(wmt.transaction_id) from wip_move_transactions wmt where wmt.transaction_id = :transactionId

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-mandatory Line Item Attribute	WIP_TO_DEPARTMENT	string	Approval based on department to which assembly was moved	No	select to_department_code from wip_move_transactions_v wmt where wmt.transaction_id = :transactionId
Non-mandatory Line Item Attribute	WIP_TO_INTRAOPERATION_STEP	string	Approval based on intraoperation step to which assembly was moved	No	select wmt.to_intraoperation_step_meaning from wip_move_transactions_v wmt where wmt.transaction_id = :transactionId
Non-mandatory Line Item Attribute	WIP_TO_OPERATION_CODE	string	Approval based on operation code to which assembly was moved	No	select to_operation_code from wip_move_transactions_v wmt where wmt.transaction_id = :transactionId
Non-mandatory Line Item Attribute	WIP_TO_OPERATION_SEQ_NUM	number	Approval based on operation sequence to which assembly was moved	No	select fnd_number.number_to_canonical(to_operation_seq_num) from wip_move_transactions wmt where wmt.transaction_id = :transactionId

Seeded transaction attributes for the following transaction types:

- WIP ERES Job Assembly Completion (oracle.apps.wip.job.assembly.complete)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-mandatory Header Attribute	TRANSACTION_DATE	date	date transaction occurred	No	select ame_util.versionDateToString(LAST_UPDATE_DATE) from mtl_material_transactions where transaction_set_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and transaction_source_id = substr(:transactionId, (instr(:transactionId, '-')+1), (instr(:transactionId, '-', 1, 2)- (instr(:transactionId, '-')+1))) and transaction_type_id in (44,17) and rownum = 1

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-mandatory Header Attribute	TRANSACTION_REQUESTOR_USER_ID	number	user ID of user initiating transaction, if any	No	select fnd_number.number_to_canonical(last_updated_by) from mtl_material_transactions where transaction_set_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and transaction_source_id = substr(:transactionId, (instr(:transactionId, '-')+1), (instr(:transactionId, '-', 1, 2)- (instr(:transactionId, '-')+1))) and transaction_type_id in (44,17) and rownum = 1
Non-Mandatory Header Attribute	WIP_ASSEMBLY	string	Approval based on assembly	No	select msik.concatenated_segments from mtl_system_items_kfv msik, mtl_material_transactions mmt where transaction_set_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and transaction_source_id = substr(:transactionId, (instr(:transactionId, '-')+1), (instr(:transactionId, '-', 1, 2)- (instr(:transactionId, '-')+1))) and mmt.inventory_item_id=msik.inventory_item_id and mmt.organization_id = msik.organization_id and mmt.transaction_type_id in (44,17) and rownum = 1
Non-Mandatory Header Attribute	WIP_COMPLETION_TRANSACTION_TYPE	string	Transaction type: WIP Assembly Completion,WIP Assembly Return	No	select transaction_type_name from mtl_material_transactions mmt, mtl_transaction_types type where transaction_set_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and transaction_source_id = substr(:transactionId, (instr(:transactionId, '-')+1), (instr(:transactionId, '-', 1, 2)- (instr(:transactionId, '-')+1))) and type.transaction_type_id = mmt.transaction_type_id and mmt.transaction_type_id in (44,17) and rownum = 1

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	WIP_JOB	string	Approval based on job.	No	select we.wip_entity_name from wip_entities we, mtl_material_transactions mmt where transaction_set_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and transaction_source_id = substr(:transactionId, (instr(:transactionId, '-')+1), (instr(:transactionId, '-', 1, 2)- (instr(:transactionId, '-')+1))) and we.wip_entity_id = mmt.transaction_source_id and mmt.transaction_type_id in (44,17) and rownum = 1
Non-Mandatory Header Attribute	WIP_JOB_PROJECT	string	Approval based on job's project	No	select pjm_project.all_proj_idtonum(wdj.project_id) from mtl_material_transactions mmt, wip_discrete_jobs wdj where transaction_set_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and transaction_source_id = substr(:transactionId, (instr(:transactionId, '-')+1), (instr(:transactionId, '-', 1, 2)- (instr(:transactionId, '-')+1))) and mmt.transaction_source_id = wdj.wip_entity_id and rownum = 1
Non-mandatory Header Attribute	WIP_JOB_TASK	string	Approval based on job's task	No	select pjm_project.all_task_idtonum(wdj.task_id) from mtl_material_transactions mmt, wip_discrete_jobs wdj where transaction_set_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and transaction_source_id = substr(:transactionId, (instr(:transactionId, '-')+1), (instr(:transactionId, '-', 1, 2)- (instr(:transactionId, '-')+1))) and mmt.transaction_source_id = wdj.wip_entity_id and rownum = 1

Attribute Category	Attribute Name	At-tribute Type	Description	Static Usage	Usage
Non-mandatory Line Item Attribute	WIP_ LOCATOR	string	Approval based on locator	No	select inv_project.get_ locator(mmt.locator_ id, mmt.organization_ id) from mtl_material_ transactions mmt where transaction_id in (select transaction_id from mtl_ material_transactions where transaction_set_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and transaction_ source_id = substr(: transactionId, (instr(: transactionId, '-')+1), (instr(:transactionId, '-', 1, 2)- (instr(:transactionId, '-')+1))) and transaction_ type_id in (44,17)) order by mmt.transaction_id
Non-mandatory Line Item Attribute	WIP_SUB INVENTORY	string	Approval based on subinventory	No	select mmt.subinventory_ code from mtl_material_ transactions mmt where transaction_id in (select transaction_id from mtl_ material_transactions where transaction_set_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and transaction_ source_id = substr(: transactionId, (instr(: transactionId, '-')+1), (instr(:transactionId, '-', 1, 2)- (instr(:transactionId, '-')+1))) and transaction_ type_id in (44,17)) order by mmt.transaction_id

Define Rules/Associate Approvers

No rules or approvers are seeded. As part of the implementation, you must complete these activities (see: Setting Up Oracle Approvals Management, *Oracle E-Records Implementation Guide*).

Oracle E-Records Seeded Data: Generic Query Attributes

Use generic query attributes to search for e-records and their associated documents in the Evidence Store. For instructions on how to search, see Evidence Store, *Oracle E-Records Implementation Guide*. For all attributes listed below:

Query Element Type = Generic

WIP Job Material Transaction

Application Code	XML Element Tag	Display Name	Comments
WIP	JOB	Job	N/A
WIP	JOB_TYPE	Job Type	N/A
WIP	TRANSACTION_TYPE	Transaction Type	N/A

WIP Job Assembly Move

Application Code	XML Element Tag	Display Name	Comments
WIP	ASSET_GROUP	Asset Group	N/A
WIP	ASSET_NUMBER	Asset Number	N/A
WIP	OP_STEP	Intraoperation Step	N/A
WIP	JOB	Job	N/A
WIP	JOB_TYPE	Job Type	N/A
WIP	MOVE_TRANSACTION_TYPE	Move Transaction Type	N/A
WIP	SCRAP_ACCOUNT_NUMBER	Scrap Account Number	N/A
WIP	TO_DEPARTMENT	To Department Code	N/A
WIP	TO_OP_STEP	To Intraoperation Step	N/A
WIP	TO_OP_CODE	To Operation Sequence Code	N/A
WIP	TO_OP_SEQ_NUM	To Operation Sequence Number	N/A
WIP	TRANSACTION_TYPE	Transaction Type	N/A

WIP Job Assembly Completion

Application Code	XML Element Tag	Display Name	Comments
WIP	ASSET_GROUP	Asset Group	N/A
WIP	ASSET_NUMBER	Asset Number	N/A
WIP	JOB	Job	N/A
WIP	JOB_TYPE	Job Type	N/A
WIP	TO_DEPARTMENT	To Department Code	N/A
WIP	TO_OP_SEQ_NUM	To Operation Sequence Number	N/A
WIP	TRANSACTION_TYPE	Transaction Type	N/A

Event Data for Oracle Quality

Oracle Workflow Seeded Data

Events

Name	Display Name	Description	Status	Owner Name	Owner Tag
oracle.apps.qa.element.create	QA ERES Collection Element Creation	ERES Event for the Creation of a Collection Element in Oracle Quality	Enabled	Oracle Quality	QA
oracle.apps.qa.element.update	QA ERES Collection Element Update	ERES Event for the Update of a Collection Element in Oracle Quality	Enabled	Oracle Quality	QA
oracle.apps.qa.plan.create	QA ERES Collection Plan Creation	ERES Event for the Creation of a Collection Plan in Oracle Quality	Enabled	Oracle Quality	QA
oracle.apps.qa.plan.update	QA ERES Collection Plan Update	ERES Event for the Update of a Collection Plan in Oracle Quality	Enabled	Oracle Quality	QA
oracle.apps.qa.spec.create	QA ERES Specification Creation	ERES Event for the Creation of a Specification in Oracle Quality	Enabled	Oracle Quality	QA
oracle.apps.qa.spec.update	QA ERES Specification Update	ERES Event for the Update of a Specification in Oracle Quality	Enabled	Oracle Quality	QA
oracle.apps.qa.result.create	QA ERES Result Creation	ERES Event for the Creation of a Result in Oracle Quality	Disabled	Oracle Quality	QA

Name	Display Name	Description	Status	Owner Name	Owner Tag
oracle.apps.qa.result.update	QA ERES Result Update	ERES Event for the Update of a Result in Oracle Quality	Enabled	Oracle Quality	QA
oracle.apps.qa.ncm.create	QA ERES Nonconformance Creation	ERES Event for the Creation of a Nonconformance in Oracle Quality	Enabled	Oracle Quality	QA
oracle.apps.qa.ncm.update	QA ERES Nonconformance Update	ERES Event for the Update of a Nonconformance in Oracle Quality	Enabled	Oracle Quality	QA
oracle.apps.qa.ncm.master.approve	QA ERES Nonconformance Master Approval	ERES Event for the Approval of a Nonconformance Master in Oracle Quality	Enabled	Oracle Quality	QA
oracle.apps.qa.ncm.detail.approve	QA ERES Nonconformance Detail Approval	ERES Event for the Approval of a Nonconformance Detail in Oracle Quality	Enabled	Oracle Quality	QA
oracle.apps.qa.disp.create	QA ERES Disposition Creation	ERES Event for the Creation of a Disposition in Oracle Quality	Enabled	Oracle Quality	QA
oracle.apps.qa.disp.update	QA ERES Disposition Update	ERES Event for the Update of a Disposition in Oracle Quality	Enabled	Oracle Quality	QA
oracle.apps.qa.disp.header.approve	QA ERES Disposition Header Approval	ERES Event for the Approval of a Disposition Header in Oracle Quality	Enabled	Oracle Quality	QA
oracle.apps.qa.disp.detail.approve	QA ERES Disposition Detail Approval	ERES Event for the Approval of a Disposition Detail in Oracle Quality	Enabled	Oracle Quality	QA
oracle.apps.qa.car.create	QA ERES Corrective Action Creation	ERES Event for the Creation of a Corrective Action in Oracle Quality	Enabled	Oracle Quality	QA
oracle.apps.qa.car.update	QA ERES Corrective Action Update	ERES Event for the Update of a Corrective Action in Oracle Quality	Enabled	Oracle Quality	QA
oracle.apps.qa.car.approve	QA ERES Corrective Action Approval	ERES Event for the Approval of a Corrective Action in Oracle Quality	Enabled	Oracle Quality	QA
oracle.apps.qa.car.review.approve	QA ERES Corrective Action Review Approval	ERES Event for the Approval of a Corrective Action Review in Oracle Quality	Enabled	Oracle Quality	QA

Name	Display Name	Description	Status	Owner Name	Owner Tag
oracle.apps.qa.car.impl.approve	QA ERES Corrective Action Implementation Approval	ERES Event for the Approval of a Corrective Action Implementation in Oracle Quality	Enabled	Oracle Quality	QA
oracle.apps.qa.spec.org.assign	QA ERES Specification Org Assignment	ERES Event for the Assignment of a Specification to one or more Organizations in Oracle Quality	Enabled	Oracle Quality	QA

Event Key

Note: You can only define a single event key for use with Oracle E-Records, but many of the Oracle Quality events require composite keys. The system creates composite keys for Oracle E-Records calls by combining multiple fields to form a unique key. This unique key identifies a single entity by concatenating the fields together with a hyphen delimiter (for example: key1-key2). The fields used to create the composite key are presented below in a comma separated list in the User Event Key column.

Event Name	User Event Key (Identifier)
oracle.apps.qa.element.create	Collection Element ID
oracle.apps.qa.element.update	Collection Element ID
oracle.apps.qa.plan.create	Plan Name
oracle.apps.qa.plan.update	Plan Name
oracle.apps.qa.result.create	Plan Name - Collection Id - Occurrence
oracle.apps.qa.result.update	Plan Name - Collection Id - Occurrence
oracle.apps.qa.ncm.create	Plan Name - Collection Id - Occurrence
oracle.apps.qa.ncm.update	Plan Name - Collection Id - Occurrence
oracle.apps.qa.ncm.master.approve	Plan Name - Collection Id - Occurrence
oracle.apps.qa.ncm.detail.approve	Plan Name - Collection Id - Occurrence
oracle.apps.qa.disp.create	Plan Name - Collection Id - Occurrence
oracle.apps.qa.disp.update	Plan Name - Collection Id - Occurrence
oracle.apps.qa.disp.header.approve	Plan Name - Collection Id - Occurrence
oracle.apps.qa.disp.detail.approve	Plan Name - Collection Id - Occurrence
oracle.apps.qa.car.create	Plan Name - Collection Id - Occurrence
oracle.apps.qa.car.update	Plan Name - Collection Id - Occurrence
oracle.apps.qa.car.approve	Plan Name - Collection Id - Occurrence
oracle.apps.qa.car.review.approve	Plan Name - Collection Id - Occurrence
oracle.apps.qa.car.impl.approve	Plan Name - Collection Id - Occurrence
oracle.apps.qa.spec.org.assign	Specification Name
oracle.apps.qa.spec.create	Specification Name
oracle.apps.qa.spec. update	Specification Name

Event Subscription

All events subscriptions have the same values for the following fields:

- System = HM001
- Source Type = Local
- Phase = 0
- Status = Disabled
- Rule Data = Key
- Rule Function = EDR_PSIG_RULE.PSIG_RULE
- Priority = Normal

Event Filter	Parameters
oracle.apps.qa.element.create	EDR_XML_MAP_CODE =qa_elements EDR_AME_TRANSACTION_TYPE =oracle.apps.qa.element.create
oracle.apps.qa.element.update	EDR_XML_MAP_CODE =qa_elements EDR_AME_TRANSACTION_TYPE =oracle.apps.qa.element.update
oracle.apps.qa.plan.create	EDR_XML_MAP_CODE =qa_plans EDR_AME_TRANSACTION_TYPE =oracle.apps.qa.plan.create
oracle.apps.qa.plan.update	EDR_XML_MAP_CODE =qa_plans EDR_AME_TRANSACTION_TYPE =oracle.apps.qa.plan.update
oracle.apps.qa.spec.create	EDR_XML_MAP_CODE = qa_specs EDR_AME_TRANSACTION_TYPE =oracle.apps.qa.spec.create
oracle.apps.qa.spec.update	EDR_XML_MAP_CODE = qa_specs EDR_AME_TRANSACTION_TYPE =oracle.apps.qa.spec.update
oracle.apps.qa.result.create	EDR_XML_MAP_CODE =qa_results EDR_AME_TRANSACTION_TYPE =oracle.apps.qa.result.create
oracle.apps.qa.result.update	EDR_XML_MAP_CODE =qa_results EDR_AME_TRANSACTION_TYPE =oracle.apps.qa.result.update
oracle.apps.qa.ncm.create	EDR_XML_MAP_CODE = qa_ncm EDR_AME_TRANSACTION_TYPE =oracle.apps.qa.ncm.create
oracle.apps.qa.ncm.update	EDR_XML_MAP_CODE =qa_ncm EDR_AME_TRANSACTION_TYPE =oracle.apps.qa.ncm.update
oracle.apps.qa.ncm.master.approve	EDR_XML_MAP_CODE =qa_ncm EDR_AME_TRANSACTION_TYPE =oracle.apps.qa.ncm.master.approve
oracle.apps.qa.ncm.detail.approve	EDR_XML_MAP_CODE =qa_ncm EDR_AME_TRANSACTION_TYPE =oracle.apps.qa.ncm.detail.approve
oracle.apps.qa.disp.create	EDR_XML_MAP_CODE =qa_ncm EDR_AME_TRANSACTION_TYPE =oracle.apps.qa.disp.create
oracle.apps.qa.disp.update	EDR_XML_MAP_CODE =qa_ncm EDR_AME_TRANSACTION_TYPE =oracle.apps.qa.disp.update
oracle.apps.qa.disp.header.approve	EDR_XML_MAP_CODE =qa_ncm EDR_AME_TRANSACTION_TYPE =oracle.apps.qa.disp.header.approve

Event Filter	Parameters
oracle.apps.qa.disp.detail.approve	EDR_XML_MAP_CODE =qa_ncm EDR_AME_TRANSACTION_TYPE =oracle.apps.qa.disp.detail.approve
oracle.apps.qa.car.create	EDR_XML_MAP_CODE =qa_ncm EDR_AME_TRANSACTION_TYPE =oracle.apps.qa.car.create
oracle.apps.qa.car.update	EDR_XML_MAP_CODE =qa_ncm EDR_AME_TRANSACTION_TYPE =oracle.apps.qa.car.update
oracle.apps.qa.car.approve	EDR_XML_MAP_CODE =qa_ncm EDR_AME_TRANSACTION_TYPE =oracle.apps.qa.car.approve
oracle.apps.qa.car.review.approve	EDR_XML_MAP_CODE =qa_ncm EDR_AME_TRANSACTION_TYPE =oracle.apps.qa.car.review.approve
oracle.apps.qa.car.impl.approve	EDR_XML_MAP_CODE =qa_ncm EDR_AME_TRANSACTION_TYPE =oracle.apps.qa.impl approve
oracle.apps.qa.spec.org.assign	EDR_XML_MAP_CODE=qa_spec_org_assignments EDR_AME_TRANSACTION_TYPE=oracle.apps.qa.spec.org.assign oracle.apps.qa.spec.create=IGNORE_SIGNATURE oracle.apps.qa.spec.update=IGNORE_SIGNATURE

Oracle Approvals Management Seeded Data

Transaction Type

All transaction types listed below belong to the Oracle Quality application.

Transaction Type ID	Transaction Type Description	Line Item Id Query String
oracle.apps.qa.element.create	QA ERES Collection Element Creation	N/A
oracle.apps.qa.element.update	QA ERES Collection Element Update	N/A
oracle.apps.qa.plan.create	QA ERES Collection Plan Creation	N/A
oracle.apps.qa.plan.update	QA ERES Collection Plan Update	N/A
oracle.apps.qa.spec.create	QA ERES Specification Creation	N/A
oracle.apps.qa.spec.update	QA ERES Specification Update	N/A

Transaction Type ID	Transaction Type Description	Line Item Id Query String
oracle.apps.qa.result.create	QA ERES Result Creation	select distinct mic.category_id from mtl_item_categories mic, (select organization_id, item_id from qa_results where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', -1)+1)), OCCURRENCE)) qr where qr.organization_id = mic.organization_id and qr.item_id = mic.inventory_item_id order by category_id asc
oracle.apps.qa.result.update	QA ERES Result Update	select distinct mic.category_id from mtl_item_categories mic, (select organization_id, item_id from qa_results where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', -1)+1)), OCCURRENCE)) qr where qr.organization_id = mic.organization_id and qr.item_id = mic.inventory_item_id order by category_id asc
oracle.apps.qa.ncm.create	QA ERES Nonconformance Creation	select distinct mic.category_id from mtl_item_categories mic, (select organization_id, item_id from qa_results where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)) qr where qr.organization_id = mic.organization_id and qr.item_id = mic.inventory_item_id order by category_id asc

Transaction Type ID	Transaction Type Description	Line Item Id Query String
oracle.apps.qa.ncm.update	QA ERES Nonconformance Update	select distinct mic.category_id from mtl_item_categories mic, (select organization_id, item_id from qa_results where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)) qr where qr.organization_id = mic.organization_id and qr.item_id = mic.inventory_item_id order by category_id asc
oracle.apps.qa.ncm.master.approve	QA ERES Nonconformance Master Approval	select distinct mic.category_id from mtl_item_categories mic, (select organization_id, item_id from qa_results where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)) qr where qr.organization_id = mic.organization_id and qr.item_id = mic.inventory_item_id order by category_id asc
oracle.apps.qa.ncm.detail.approve	QA ERES Nonconformance Detail Approval	select distinct mic.category_id from mtl_item_categories mic, (select organization_id, item_id from qa_results where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)) qr where qr.organization_id = mic.organization_id and qr.item_id = mic.inventory_item_id order by category_id asc

Transaction Type ID	Transaction Type Description	Line Item Id Query String
oracle.apps.qa.disp.create	QA ERES Disposition Creation	select distinct mic.category_id from mtl_item_categories mic, (select organization_id, item_id from qa_results where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)) qr where qr.organization_id = mic.organization_id and qr.item_id = mic.inventory_item_id order by category_id asc
oracle.apps.qa.disp.update	QA ERES Disposition Update	select distinct mic.category_id from mtl_item_categories mic, (select organization_id, item_id from qa_results where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)) qr where qr.organization_id = mic.organization_id and qr.item_id = mic.inventory_item_id order by category_id asc
oracle.apps.qa.disp.header.approve	QA ERES Disposition Header Approval	select distinct mic.category_id from mtl_item_categories mic, (select organization_id, item_id from qa_results where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)) qr where qr.organization_id = mic.organization_id and qr.item_id = mic.inventory_item_id order by category_id asc

Transaction Type ID	Transaction Type Description	Line Item Id Query String
oracle.apps.qa.disp.detail.approve	QA ERES Disposition Detail Approval	select distinct mic.category_id from mtl_item_categories mic, (select organization_id, item_id from qa_results where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)) qr where qr.organization_id = mic.organization_id and qr.item_id = mic.inventory_item_id order by category_id asc
oracle.apps.qa.car.create	QA ERES Corrective Action Creation	select distinct mic.category_id from mtl_item_categories mic, (select organization_id, item_id from qa_results where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)) qr where qr.organization_id = mic.organization_id and qr.item_id = mic.inventory_item_id order by category_id asc
oracle.apps.qa.car.update	QA ERES Corrective Action Update	select distinct mic.category_id from mtl_item_categories mic, (select organization_id, item_id from qa_results where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)) qr where qr.organization_id = mic.organization_id and qr.item_id = mic.inventory_item_id order by category_id asc

Transaction Type ID	Transaction Type Description	Line Item Id Query String
oracle.apps.qa.car.approve	QA ERES Corrective Action Approval	select distinct mic.category_id from mtl_item_categories mic, (select organization_id, item_id from qa_results where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)) qr where qr.organization_id = mic.organization_id and qr.item_id = mic.inventory_item_id order by category_id asc
oracle.apps.qa.car.review.approve	QA ERES Corrective Action Review Approval	select distinct mic.category_id from mtl_item_categories mic, (select organization_id, item_id from qa_results where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)) qr where qr.organization_id = mic.organization_id and qr.item_id = mic.inventory_item_id order by category_id asc

Transaction Type ID	Transaction Type Description	Line Item Id Query String
oracle.apps.qa.car.impl.approve	QA ERES Corrective Action Implementation Approval	select distinct mic.category_id from mtl_item_categories mic, (select organization_id, item_id from qa_results where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)) qr where qr.organization_id = mic.organization_id and qr.item_id = mic.inventory_item_id order by category_id asc
oracle.apps.qa.spec.org.assign	QA ERES Specification Org Assignment	select child_spec_id from qa_spec_org_assignments_v where spec_id = :transactionId and assign_flag = 1 order by child_spec_id asc

Transaction Attributes

Although the user can define their own attributes for the transaction types listed above, the more commonly used attributes are seeded. The following attributes apply to all Oracle Quality transaction types. Attributes specific to certain transaction types are listed in later tables, by transaction type.

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Mandatory Attribute	AL-LOW_DELETING_RULE_GENERATED_APPROVERS	boolean	whether to let the calling application (or its end users) delete approvers generated by the rules	Yes	false
Mandatory Attribute	ALLOW_REQUESTOR_APPROVAL	boolean	whether to allow requestors to approve their own transactions (when the rules do so)	Yes	false
Mandatory Attribute	AT_LEAST_ONE_RULE_MUST_APPLY	boolean	whether to require that at least one rule apply to each transaction	Yes	false

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Mandatory Attribute	EFFECTIVE_ RULE_DATE	date	the date that determines which rules are active	Yes	N/A
Mandatory Attribute	EVALUATE_ PRIORITIES_ PER_ITEM	boolean	whether to evaluate rule priorities per item under strict item evaluation	Yes	false
Mandatory Attribute	REJECTION_ RESPONSE	string	how AME responds to a rejection	Yes	STOP_ALL_ ITEMS
Mandatory Attribute	USE_RESTR ICTIVE_ ITEM_ EVALUAT ION	boolean	whether to require that the same item satisfy all item conditions in a given rule	Yes	false
Mandatory Attribute	USE_ WORKFLOW	boolean	whether OAM should log exceptions to the Workflow context stack	Yes	true
Mandatory Attribute	WORK- FLOW_ITEM_ KEY	string	the transaction's Workflow item key	Yes	N/A
Mandatory Attribute	WORK- FLOW_ITEM_ TYPE	string	the transaction's Workflow item type	Yes	N/A
Non-mandatory Header Attribute	ALLOW_ EMPTY_A PPROVAL_ GROUPS	boolean	whether to allow approval groups to have no members	Yes	false
Non-mandatory Header Attribute	INCLUDE_ ALL_JOB_ LEVEL_AP PROVERS	boolean	whether to include all approvers at a given job level	Yes	false
Non-mandatory Header Attribute	TRANSACT ION_DATE	date	date transaction occurred	No	select ame_util. versionDateToString(CREAT ION_DATE) from qa_chars where char_id = :transaction Id

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-mandatory Header Attribute	TRANSACTION_GROUP_ID	number	business-group ID in which transaction occurred	Yes	N/A
Non-mandatory Header Attribute	TRANSACTION_REQUESTOR_PERSON_ID	number	person ID of person initiating transaction, if any	Yes	N/A
Non-mandatory Header Attribute	TRANSACTION_REQUESTOR_USER_ID	number	user ID of user initiating transaction, if any	No	select Last_updated_by from qa_chars where char_id = :transaction Id
Non-mandatory Header Attribute	TRANSACTION_SET_OF_BOOKS_ID	number	set-of-books ID in which transaction occurred	Yes	N/A

Seeded transaction attributes for the following transaction types:

- QA ERES Collection Element Creation (oracle.apps.qa.element.create)
- QA ERES Collection Element Update (oracle.apps.qa.element.update)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_ELEMENT_NAME	string	Approval based on the Collection Element Name	No	select name from qa_chars_v where char_id = :transactionId
Non-Mandatory Header Attribute	QA_ELEMENT_TYPE	string	Approval based on the Collection Element Type	No	select char_type_meaning from qa_chars_v where char_id = :transactionId
Non-mandatory Header Attribute	TOP_SUPERVISOR_PERSON_ID	number	person ID of the top person in the HR supervisory hierarchy	Yes	N/A

Seeded transaction attributes for the following transaction types:

- QA ERES Collection Plan Creation (oracle.apps.qa.plan.create)
- QA ERES Collection Plan Update (oracle.apps.qa.plan.update)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_ORGANIZATION_CODE	string	Approval based on the Organization Code	No	select organization_code from qa_eres_plans_v where plan_id = :transactionId
Non-Mandatory Header Attribute	QA_PLAN_NAME	string	Approval based on the Collection Plan Name	No	select name from qa_eres_plans_v where plan_id = :transactionId
Non-Mandatory Header Attribute	QA_PLAN_TYPE	string	Approval based on the Collection Plan Type	No	select plan_type_meaning from qa_eres_plans_v where plan_id = :transactionId
Non-Mandatory Header Attribute	TOP_SUPERVISOR_PERSON_ID	number	person ID of the top person in the HR supervisory hierarchy	Yes	N/A
Non-Mandatory Header Attribute	TRANSACTION_ORG_ID	number	org ID in which transaction occurred	No	select organization_id from qa_plans where plan_id = :transactionId

Seeded transaction attributes for the following transaction types:

- QA ERES Specification Creation (oracle.apps.qa.spec.create)
- QA ERES Specification Update (oracle.apps.qa.spec.update)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_ CUSTOMER	string	Approval based on the Customer	No	select customer_name from qa_eres_specs_ v where spec_id = : transactionId
Non-Mandatory Header Attribute	QA_ITEM	string	Approval based on the Item	No	select item from qa_eres_ specs_v where spec_id = :transactionId
Non-Mandatory Header Attribute	QA_ITEM_ CATEGORY	string	Item Category of the Specification	No	select category_name from qa_eres_specs_ v where spec_id = : transactionId
Non-Mandatory Header Attribute	QA_ITEM_ CATEGORY_ SET	string	Approval based on the Item's Category Set	No	select category_set_ name from qa_eres_ specs_v where spec_id = :transactionId
Non-Mandatory Header Attribute	QA_ORGAN IZATION_ CODE	string	Approval based on the Organization Code	No	select organization_code from qa_eres_specs_ v where spec_id = : transactionId
Non-Mandatory Header Attribute	QA_SPECIF ICATION_TY PE	string	Approval based on the Specification Type	No	select assignment_type_ meaning from qa_eres_ specs_v where spec_id = :transactionId
Non-Mandatory Header Attribute	QA_SUPPL IER	string	Approval based on the Supplier	No	select vendor_name from qa_eres_specs_v where spec_id = :transactionId
Non-Mandatory Header Attribute	TOP_SU PERVISOR_ PERSON_ID	number	person ID of the top person in the HR supervisory hierarchy	Yes	N/A
Non-Mandatory Header Attribute	TRANSACT ION_ORG_ID	number	org ID in which transaction occurred	No	select organization_id from qa_specs where spec_id = :transactionId

Seeded transaction attributes for the following transaction types:

- QA ERES Result Creation (oracle.apps.qa.result.create)
- QA ERES Result Update (oracle.apps.qa.result.update)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_ASSET_NUMBER	string	Approval based on the Asset Number	No	select qa_eres_pkg.get_result_column_value(:transactionId, 163) from dual
Non-Mandatory Header Attribute	QA_COMPONENT_ITEM	string	Approval based on the Component Item	No	select qa_eres_pkg.get_result_column_value(:transactionId, 60) from dual
Non-Mandatory Header Attribute	QA_ITEM	string	Approval based on the Item	No	select qa_eres_pkg.get_result_column_value(:transactionId, 10) from dual
Non-Mandatory Header Attribute	QA_LOT_NUMBER	string	Approval based on the Lot Number	No	select qa_eres_pkg.get_result_column_value(:transactionId, 16) from dual
Non-Mandatory Header Attribute	QA_ORGANIZATION_CODE	string	Approval based on the Organization Code	No	select organization_code from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', -1)+1)), OCCURRENCE) and rownum < 2
Non-Mandatory Header Attribute	QA_PLAN_NAME	string	Approval based on the Collection Plan Name	No	select name from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', -1)+1)), OCCURRENCE) and rownum < 2

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_PLAN_TYPE	string	Approval based on the Collection Plan Type	No	select plan_type from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', -1)+1)), OCCURRENCE) and rownum < 2
Non-Mandatory Header Attribute	QA_SERIAL_NUMBER	string	Approval based on the Serial Number	No	select qa_eres_pkg.get_result_column_value(:transactionId, 17) from dual
Non-Mandatory Header Attribute	QA_SUPPLIER	string	Approval based on the Supplier	No	select qa_eres_pkg.get_result_column_value(:transactionId, 26) from dual
Non-Mandatory Header Attribute	TOP_SUPERVISOR_PERSON_ID	number	person ID of the top person in the HR supervisory hierarchy	Yes	N/A

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	TRANSACTION_ORG_ID	number	org ID in which transaction occurred	No	select organization_id from qa_results where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', -1)+1)), OCCURRENCE) and rownum < 2
Non-Mandatory Line Attribute	QA_ALL_ITEM_CATEGORIES	string	Approval based on all possible Categories for an Item	No	select concatenated_segments from mtl_categories_kfv where category_id in (select distinct mic.category_id from mtl_item_categories mic, (select organization_id, item_id from qa_results where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', -1)+1)), OCCURRENCE)) qr where qr.organization_id = mic.organization_id and qr.item_id = mic.inventory_item_id) order by category_id

Seeded transaction attributes for the following transaction types:

- QA ERES Nonconformance Creation (oracle.apps.qa.ncm.create)
- QA ERES Nonconformance Update (oracle.apps.qa.ncm.update)
- QA ERES Nonconformance Master Approval (oracle.apps.qa.ncm.master.approve)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_ASSET_NUMBER	string	Approval based on the Asset Number	No	select asset_number from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	QA_COMPONENT_ITEM	string	Approval based on the Component Item	No	select comp_item from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_COMPONENT_LOT_NUMBER	string	Approval based on the Component Lot Number	No	select comp_lot_number from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	QA_COMPONENT_SERIAL_NUMBER	string	Approval based on the Component Serial Number	No	select comp_serial_number from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_ITEM	string	Approval based on the Item	No	select item from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	QA_LOT_NUMBER	string	Approval based on the Lot Number	No	select lot_number from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_NONCONFORMANCE_PRIORITY	string	Approval based on the Non-conformance Priority	No	select nonconform_priority from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transaction Id, (instr(:transaction Id, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transaction Id, (instr(:transaction Id, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transaction Id), (instr(:transaction Id, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	QA_NONCONFORMANCE_SEVERITY	string	Approval based on the Nonconformance Severity	No	select nonconform_severity from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transaction Id, (instr(:transaction Id, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transaction Id, (instr(:transaction Id, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transaction Id), (instr(:transaction Id, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_NONCONFORMANCE_SOURCE	string	Approval based on the Non-conformance Source	No	select nonconformance_source from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transaction Id, (instr(:transaction Id, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transaction Id, (instr(:transaction Id, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transaction Id), (instr(:transaction Id, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	QA_NONCONFORMANCE_TYPE	string	Approval based on the Noncon-formance Type	No	select nonconformance_type from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transaction Id, (instr(:transaction Id, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transaction Id, (instr(:transaction Id, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transaction Id), (instr(:transaction Id, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_ORGANIZATION_CODE	string	Approval based on the Organization Code	No	select organization_code from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transaction Id, (instr(:transaction Id, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transaction Id, (instr(:transaction Id, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transaction Id), (instr(:transaction Id, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	QA_PLAN_NAME	string	Approval based on the Collection Plan Name	No	select name from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transaction Id, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transaction Id), (instr(:transaction Id, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_PLAN_TYPE	string	Approval based on the Collection Plan Type	No	select plan_type from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	QA_SERIAL_NUMBER	string	Approval based on the Serial Number	No	select serial_number from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_SUPPLIER	string	Approval based on the Supplier	No	select vendor_name from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	TOP_SUPERVISOR_PERSON_ID	number	person ID of the top person in the HR supervisory hierarchy	Yes	N/A

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	TRANSACTION_ORG_ID	number	org ID in which transaction occurred	No	select organization_id from qa_results where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Line Attribute	QA_ALL_ITEM_CATEGORIES	string	Approval based on all possible Categories for an Item	No	select concatenated_segments from mtl_categories_kfv where category_id in (select distinct mic.category_id from mtl_item_categories mic, (select organization_id, item_id from qa_results where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)) qr where qr.organization_id = mic.organization_id and qr.item_id = mic.inventory_item_id) order by category_id

Seeded transaction attributes for the following transaction types:

- QA ERES Nonconformance Detail Approval (oracle.apps.qa.ncm.approve.detail)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_ACTION_EXECUTED	string	Approval based on the Action Executed	No	select qa_eres_pkg.get_result_column_value(plan_id, collection_id, occurrence, 195) from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	QA_COMPONENT_ITEM	string	Approval based on the Component Item	No	select comp_item from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_COMPONENT_LOT_NUMBER	string	Approval based on the Component Lot Number	No	select comp_lot_number from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	QA_COMPONENT_SERIAL_NUMBER	string	Approval based on the Component Serial Number	No	select comp_serial_number from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_ITEM	string	Approval based on the Item	No	select item from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	QA_LOT_NUMBER	string	Approval based on the Lot Number	No	select lot_number from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_NONCONFORMANCE_SOURCE	string	Approval based on the Non-conformance Source	No	select nonconformance_source from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transaction Id, (instr(:transaction Id, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	QA_ORGANIZATION_CODE	string	Approval based on the Organization Code	No	select organization_code from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transaction Id, (instr(:transaction Id, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_PLAN_NAME	string	Approval based on the Collection Plan Name	No	select name from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	QA_PLAN_TYPE	string	Approval based on the Collection Plan Type	No	select plan_type from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_SERIAL_NUMBER	string	Serial Number of the Collection Result Row	No	select serial_number from qa_results_full_v select serial_number from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	TOP_SUPERVISOR_PERSON_ID	number	person ID of the top person in the HR supervisory hierarchy	Yes	N/A

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	TRANSACTION_ORG_ID	number	org ID in which transaction occurred	No	select organization_id from qa_results where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Line Attribute	QA_ALL_ITEM_CATEGORIES	string	Approval based on all possible Categories for an Item	No	select concatenated_segments from mtl_categories_kfv where category_id in (select distinct mic.category_id from mtl_item_categories mic, (select organization_id, item_id from qa_results where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)) qr where qr.organization_id = mic.organization_id and qr.item_id = mic.inventory_item_id) order by category_id

Seeded transaction attributes for the following transaction types:

- QA ERES Disposition Creation (oracle.apps.qa.disp.create)
- QA ERES Disposition Update (oracle.apps.qa.disp.update)
- QA ERES Disposition Header Approval (oracle.apps.qa.disp.header.approve)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_DISPOSITION	string	Approval based on the Disposition	No	select disposition from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	QA_DISPOSITION_SOURCE	String	Approval based on the Disposition Source	No	select disposition_source from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_ITEM	String	Approval based on the Item	No	select item from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	QA_ORGANIZATION_CODE	String	Approval based on the Organization Code	No	select organization_code from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_PLAN_NAME	String	Approval based on the Collection Plan Name	No	select name from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	QA_PLAN_TYPE	String	Approval based on the Collection Plan Type	No	select plan_type from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	TOP_SUPERVISOR_PERSON_ID	number	person ID of the top person in the HR supervisory hierarchy	Yes	N/A

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	TRANSACTION_ORG_ID	number	org ID in which transaction occurred	No	select organization_id from qa_results where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Line Attribute	QA_ALL_ITEM_CATEGORIES	string	Approval based on all possible Categories for an Item	No	select concatenated_segments from mtl_categories_kfv where category_id in (select distinct mic.category_id from mtl_item_categories mic, (select organization_id, item_id from qa_results where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)) qr where qr.organization_id = mic.organization_id and qr.item_id = mic.inventory_item_id) order by category_id

Seeded transaction attributes for the following transaction types:

- QA ERES Disposition Detail Approval (oracle.apps.qa.disp.detail.approve)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_COMPONENT_ITEM	string	Approval based on the Component Item	No	select comp_item from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	QA_COMPONENT_LOT_NUMBER	string	Approval based on the Component Lot Number	No	select comp_lot_number from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_COMPONENT_SERIAL_NUMBER	string	Approval based on the Component Serial Number	No	select comp_serial_number from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	QA_DISPOSITION	string	Approval based on the Disposition	No	select disposition from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_DISPOSITION_ACTION	String	Approval based on the Disposition Action	No	select disposition_action from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	QA_DISPOSITION_SOURCE	string	Approval based on the Disposition Source	No	select disposition_source from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)

Attribute Category	Attribute Name	At-tribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_ITEM	String	Approval based on the Item	No	select item from qa_ results_full_v where plan_id = substr(: transactionId, 1, (instr(: transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(: transactionId, '- ', 1, 2)-1) - (instr(: transactionId, '-', 1, 1)))) and occurrence = NVL(substr(: transactionId, (instr(: transactionId, '-', 1, 2)+1), (DECODE((instr(: transactionId, '-', 1, 3)-1), -1, length(: transactionId), (instr(: transactionId, '-', 1, 3)-1)) - (instr(: transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	QA_JOB	string	Approval based on the Job	No	select job_name from qa_results_full_v where plan_id = substr(: transactionId, 1, (instr(: transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(: transactionId, '- ', 1, 2)-1) - (instr(: transactionId, '-', 1, 1)))) and occurrence = NVL(substr(: transactionId, (instr(: transactionId, '-', 1, 2)+1), (DECODE((instr(: transactionId, '-', 1, 3)-1), -1, length(: transactionId), (instr(: transactionId, '-', 1, 3)-1)) - (instr(: transactionId, '-', 1, 2)))), OCCURRENCE)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_LOT_NUMBER	string	Approval based on the Lot Number	No	select lot_number from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	QA_ORGANIZATION_CODE	string	Approval based on the Organization Code	No	select organization_code from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_PLAN_NAME	string	Approval based on the Collection Plan Name	No	select name from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	QA_PLAN_TYPE	string	Approval based on the Collection Plan Type	No	select plan_type from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_PO_NUMBER	string	Approval based on the PO Number	No	select po_number from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	QA_SERIAL_NUMBER	string	Approval based on the Serial Number	No	select serial_number from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	TOP_SUPERVISOR_PERSON_ID	number	person ID of the top person in the HR supervisory hierarchy	Yes	N/A

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	TRANSACTION_ORG_ID	number	org ID in which transaction occurred	No	select organization_id from qa_results where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Line Attribute	QA_ALL_ITEM_CATEGORIES	string	Approval based on all possible Categories for an Item	No	select concatenated_segments from mtl_categories_kfv where category_id in (select distinct mic.category_id from mtl_item_categories mic, (select organization_id, item_id from qa_results where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)) qr where qr.organization_id = mic.organization_id and qr.item_id = mic.inventory_item_id) order by category_id

Seeded transaction attributes for the following transaction types:

- QA ERES Corrective Action Creation (oracle.apps.qa.car.create)
- QA ERES Corrective Action Update (oracle.apps.qa.car.update)
- QA ERES Corrective Action Approval (oracle.apps.qa.car.approve)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_ CUSTOMER	string	Approval based on the Customer	No	select customer_name from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transaction Id), (instr(:transaction Id, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	QA_DE PARTMENT	String	Approval based on the Department	No	select department from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transaction Id), (instr(:transaction Id, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_ITEM	String	Approval based on the Item	No	select item from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId,'-', 1, 3)-1), -1, length(:transaction Id), (instr(:transaction Id, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	QA_ORGANIZATION_CODE	String	Approval based on the Organization Code	No	select organization_code from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId,'-', 1, 3)-1), -1, length(:transaction Id), (instr(:transaction Id, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_PLAN_NAME	String	Approval based on the Collection Plan Name	No	select name from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	QA_PLAN_TYPE	String	Approval based on the Collection Plan Type	No	select plan_type from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_PO_NUMBER	String	Approval based on the PO Number	No	select po_number from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transaction Id), (instr(:transaction Id, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	QA_REQUEST_PRIORITY	String	Approval based on the Request Priority	No	select request_priority from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transaction Id), (instr(:transaction Id, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_REQUEST_SEVERITY	String	Approval based on the Request Severity	No	select request_severity from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transaction Id), (instr(:transaction Id, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	QA_REQUEST_SOURCE	String	Approval based on the Request Source	No	select request_source from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transaction Id), (instr(:transaction Id, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_SUPPLIER	String	Approval based on the Supplier	No	select vendor_name from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transaction Id), (instr(:transaction Id, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	TOP_SUPERVISOR_PERSON_ID	number	person ID of the top person in the HR supervisory hierarchy	Yes	N/A

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	TRANSACTION_ORG_ID	number	org ID in which transaction occurred	no	select organization_id from qa_results where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	QA_ALL_ITEM_CATEGORIES	String	Approval based on all possible Categories for an Item	No	select concatenated_segments from mtl_categories_kfv where category_id in (select distinct mic.category_id from mtl_item_categories mic, (select organization_id, item_id from qa_results where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)) qr where qr.organization_id = mic.organization_id and qr.item_id = mic.inventory_item_id) order by category_id

Seeded transaction attributes for the following transaction types:

- QA ERES Corrective Action Review Approval (oracle.apps.qa.car.approve.review)
- QA ERES Corrective Action Implementation Approval (oracle.apps.qa.car.approve.impl)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_ITEM	String	Approval based on the Item	No	select item from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	QA_ORGANIZATION_CODE	String	Approval based on the Organization Code	No	select organization_code from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_PLAN_NAME	String	Approval based on the Collection Plan Name	No	select name from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	QA_PLAN_TYPE	String	Approval based on the Collection Plan Type	No	select plan_type from qa_results_full_v where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transactionId, (instr(:transactionId, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transactionId), (instr(:transactionId, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_ REQUEST_ PRIORITY	String	Approval based on the Request Priority	No	select request_priority from qa_results_full_v where plan_id = substr(: transactionId, 1, (instr(: transactionId, '-')-1)) and collection_id = substr(: transactionId, (instr(: transactionId, '-', 1, 1)+1), ((instr(:transaction Id, '-', 1, 2)-1) - (instr(: transactionId, '-', 1, 1)))) and occurrence = NVL(substr(: transactionId, (instr(: transactionId, '-', 1, 2)+1), (DECODE((instr(: transactionId, '-', 1, 3)-1), -1, length(:transaction Id), (instr(:transaction Id, '-', 1, 3)-1)) - (instr(: transactionId, '-', 1, 2)))), OCCURRENCE)
Non-Mandatory Header Attribute	QA_ REQUEST_ SEVERITY	String	Approval based on the Request Severity	No	select request_severity from qa_results_full_v where plan_id = substr(: transactionId, 1, (instr(: transactionId, '-')-1)) and collection_id = substr(: transactionId, (instr(: transactionId, '-', 1, 1)+1), ((instr(:transaction Id, '-', 1, 2)-1) - (instr(: transactionId, '-', 1, 1)))) and occurrence = NVL(substr(: transactionId, (instr(: transactionId, '-', 1, 2)+1), (DECODE((instr(: transactionId, '-', 1, 3)-1), -1, length(:transaction Id), (instr(:transaction Id, '-', 1, 3)-1)) - (instr(: transactionId, '-', 1, 2)))), OCCURRENCE)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_ REQUEST_ SOURCE	String	Approval based on the Request Source	No	select request_source from qa_results_full_v where plan_id = substr(: transactionId, 1, (instr(: transactionId, '-')-1)) and collection_id = substr(: transactionId, (instr(: transactionId, '-', 1, 1)+1), ((instr(:transaction Id, '-', 1, 2)-1) - (instr(: transactionId, '-', 1, 1)))) and occurrence = NVL(substr(: transactionId, (instr(: transactionId, '-', 1, 2)+1), (DECODE((instr(: transactionId, '-', 1, 3)- 1), -1, length(:transaction Id), (instr(:transaction Id, '-', 1, 3)-1)) - (instr(: transactionId, '-', 1, 2)))), OCCURRENCE)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	TOP_SUPERVISOR_PERSON_ID	number	person ID of the top person in the HR supervisory hierarchy	Yes	N/A
Non-Mandatory Line Item Attribute	QA_ALL_ITEM_CATEGORIES	String	Approval based on all possible Categories for an Item	No	select concatenated_segments from mtl_categories_kfv where category_id in (select distinct mic.category_id from mtl_item_categories mic, (select organization_id, item_id from qa_results where plan_id = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and collection_id = substr(:transaction Id, (instr(:transaction Id, '-', 1, 1)+1), ((instr(:transactionId, '-', 1, 2)-1) - (instr(:transactionId, '-', 1, 1)))) and occurrence = NVL(substr(:transaction Id, (instr(:transaction Id, '-', 1, 2)+1), (DECODE((instr(:transactionId, '-', 1, 3)-1), -1, length(:transaction Id), (instr(:transaction Id, '-', 1, 3)-1)) - (instr(:transactionId, '-', 1, 2)))), OCCURRENCE)) qr where qr.organization_id = mic.organization_id and qr.item_id = mic.inventory_item_id) order by category_id

Seeded transaction attributes for the following transaction types:

- QA ERES Specification Org Assignment (oracle.apps.qa.spec.org.assign)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	QA_CUSTOMER	String	Approval based on the Customer	No	select customer_name from qa_eres_specs_v where spec_id = :transactionId
Non-Mandatory Header Attribute	QA_ITEM	String	Approval based on the Item	No	select item from qa_eres_specs_v where spec_id = :transactionId
Non-Mandatory Header Attribute	QA_ITEM_CATEGORY	String	Approval based on the Item's Category	No	select category_name from qa_eres_specs_v where spec_id = :transactionId
Non-Mandatory Header Attribute	QA_ITEM_CATEGORY_SET	String	Approval based on the Item's Category Set	No	select category_set_name from qa_eres_specs_v where spec_id = :transactionId
Non-Mandatory Header Attribute	QA_MASTER_SPEC_ORG_CODE	String	Approval based on the Master Specification's Organization Code	No	select organization_code from qa_eres_specs_v where spec_id = :transactionId
Non-Mandatory Header Attribute	QA_SPECIFICATION_TYPE	String	Approval based on the Specification Type	No	select assignment_type_meaning from qa_eres_specs_v where spec_id = :transactionId
Non-Mandatory Header Attribute	QA_SUPPLIER	String	Approval based on the Supplier	No	select vendor_name from qa_eres_specs_v where spec_id = :transactionId
Non-Mandatory Header Attribute	TOP_SUPERVISOR_PERSON_ID	Number	person ID of the top person in the HR supervisory hierarchy	Yes	N/A
Non-Mandatory Header Attribute	TRANSACTION_ORG_ID	Number	org ID in which transaction occurred	No	select organization_id from qa_specs where spec_id = :transactionId
Non-Mandatory Line Item Attribute	QA_ALL_ASSIGNED_SPEC_ORG_CODES	String	Approval based on the Organization Codes to which the Master Specification has been Assigned	No	select organization_code from qa_eres_specs_v where spec_id in (select child_spec_id from qa_spec_org_assignments_v where spec_id = :transactionId and assign_flag = 1) order by spec_id

Define Rules/Associate Approvers

No rules or approvers are seeded. As part of the implementation, you must complete these activities (see: Setting Up Oracle Approvals Management, *Oracle E-Records Implementation Guide*).

Oracle E-Records Seeded Data

Generic Query Attributes

XML Element Tag	Display Name
REVISION	Revision
COMP_ITEM	Component Item
PROJECT_NUMBER	Project
TASK_NUMBER	Task
JOB_NAME	Job
VENDOR_NAME	Supplier
CUSTOMER_NAME	Customer

Event Data for Oracle Purchasing

Oracle Workflow Seeded Data

Events

Name	Display Name	Description	Status	Owner Name	Owner Tag
oracle.apps.po.rcv.inspect	PO ERES Receiving Inspection	ERES Event for an Inspection in the Receiving component of Oracle Purchasing	Enabled	Oracle Purchasing	PO
oracle.apps.po.rcv.transfer	PO ERES Receiving Transfer	ERES Event for a Transfer in the Receiving component of Oracle Purchasing	Enabled	Oracle Purchasing	PO
oracle.apps.po.rcv.deliver	PO ERES Receiving Delivery	ERES Event for a Delivery in the Receiving component of Oracle Purchasing	Enabled	Oracle Purchasing	PO
oracle.apps.po.asl.create	PO ERES ASL Creation	ERES Event for the Creation of an ASL in Oracle Purchasing	Enabled	Oracle Purchasing	PO
oracle.apps.po.asl.update	PO ERES ASL Update	ERES Event for the Update of an ASL in Oracle Purchasing	Enabled	Oracle Purchasing	PO

Event Key

Note: You can only define a single event key for use with Oracle E-Records, but many of the Oracle Purchasing events require composite

keys. The system creates composite keys for Oracle E-Records calls by combining multiple fields to form a unique key. This unique key identifies a single entity by concatenating the fields together with a hyphen delimiter (for example: key1-key2). The fields used to create the composite key are presented below in a comma separated list in the User Event Key column.

Event Name	User Event Key (Identifier)
oracle.apps.po.rcv.inspect	Receipt Number
oracle.apps.po.rcv.transfer	Receipt Number
oracle.apps.po.rcv.deliver	Receipt Number
oracle.apps.po.asl.create	Organization, Supplier Site, Item/Commodity
oracle.apps.po.asl.update	Organization, Supplier Site, Item/Commodity

Event Subscription

All events subscriptions have the same values for the following fields:

- System = HM001
- Source Type = Local
- Phase = 0
- Status = Disabled
- Rule Data = Key
- Rule Function = EDR_PSIG_RULE.PSIG_RULE
- Priority = Normal

Event Filter	Parameters
oracle.apps.po.rcv.inspect	EDR_XML_MAP_CODE=po_eres_rcv_ transactions EDR_AME_TRANSACTION_TY PE=oracle.apps.po.rcv.inspect
oracle.apps.po.rcv.transfer	EDR_XML_MAP_CODE=po_eres_rcv_ transactions EDR_AME_TRANSACTION_TY PE=oracle.apps.po.rcv.transfer
oracle.apps.po.rcv.deliver	EDR_XML_MAP_CODE=po_eres_rcv_ transactions EDR_AME_TRANSACTION_TY PE=oracle.apps.po.rcv.deliver
oracle.apps.po.asl.create	EDR_XML_MAP_CODE=po_eres_asl EDR_AME_TRANSACTION_TYPE=oracle. apps.po.asl.create
oracle.apps.po.asl.update	EDR_XML_MAP_CODE=po_eres_asl EDR_AME_TRANSACTION_TYPE=oracle. apps.po.asl.update

Oracle Approvals Management Seeded Data

Transaction Type

All transaction types listed below belong to the Oracle Purchasing application.

Transaction Type ID	Transaction Type Description	Line Item Id Query String
oracle.apps.po.rcv.inspect	PO ERES Receiving Inspection	select mic.category_id from mtl_item_categories mic, (select organization_id, item_id from qa_eres_rcv_trans_interface_v where PARENT_TRANSACTION_ID = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and QA_COLLECTION_ID = substr(:transaction Id, (instr(:transactionId, '-', 1, 1)+1)) and rownum = 1) qrti where qrti.organization_id = mic.organization_id(+) and qrti.item_id = mic.inventory_item_id(+) order by category_id asc
oracle.apps.po.rcv.transfer	PO ERES Receiving Transfer	select mic.category_id from mtl_item_categories mic, (select organization_id, item_id from qa_eres_rcv_trans_interface_v where PARENT_TRANSACTION_ID = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and QA_COLLECTION_ID = substr(:transaction Id, (instr(:transactionId, '-', 1, 1)+1)) and rownum = 1) qrti where qrti.organization_id = mic.organization_id(+) and qrti.item_id = mic.inventory_item_id(+) order by category_id asc
oracle.apps.po.rcv.deliver	PO ERES Receiving Delivery	select mic.category_id from mtl_item_categories mic, (select organization_id, item_id from qa_eres_rcv_trans_interface_v where PARENT_TRANSACTION_ID = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and QA_COLLECTION_ID = substr(:transaction Id, (instr(:transactionId, '-', 1, 1)+1)) and rownum = 1) qrti where qrti.organization_id = mic.organization_id(+) and qrti.item_id = mic.inventory_item_id(+) order by category_id asc
oracle.apps.po.asl.create	PO ERES ASL Creation	select mic.category_id from mtl_item_categories mic, (select owning_organization_id, item_id from po_approved_supplier_list_v where asl_id = :transactionId) paslv where paslv.owning_ organization_id = mic.organization_id(+) and paslv.item_id = mic.inventory_item_id(+) order by category_id asc
oracle.apps.po.asl.update	PO ERES ASL Update	select mic.category_id from mtl_item_categories mic, (select owning_organization_id, item_id from po_approved_supplier_list_v where asl_id = :transactionId) paslv where paslv.owning_ organization_id = mic.organization_id(+) and paslv.item_id = mic.inventory_item_id(+) order by category_id asc

Transaction Attributes

Although the user can define their own attributes for the transaction types listed above, the more commonly used attributes are seeded. The following attributes apply to all Oracle Purchasing transaction types. Attributes specific to certain transaction types are listed in later tables, by transaction type.

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Mandatory Attribute	AL-LOW_DELET ING_RULE_ GENERATED_ APPROVERS	boolean	whether to let the calling application (or its end users) delete approvers generated by the rules	Yes	false
Mandatory Attribute	ALLOW_ REQUESTOR_ APPROVAL	boolean	whether to allow requestors to approve their own transactions (when the rules do so)	Yes	false
Mandatory Attribute	AT_LEAST_ ONE_RULE_ MUST_APPLY	boolean	whether to require that at least one rule apply to each transaction	Yes	false
Mandatory Attribute	EFFECTIVE_ RULE_DATE	date	the date that determines which rules are active	Yes	N/A
Mandatory Attribute	EVALUATE_ PRIORITIES_ PER_ITEM	boolean	whether to evaluate rule priorities per item under strict item evaluation	Yes	false
Mandatory Attribute	REJECTION_ RESPONSE	string	how AME responds to a rejection	Yes	STOP_ ALL_ ITEMS
Mandatory Attribute	USE_RESTR ICTIVE_ ITEM_ EVALUAT ION	boolean	whether to require that the same item satisfy all item conditions in a given rule	Yes	false
Mandatory Attribute	USE_ WORKFLOW	boolean	whether OAM should log exceptions to the Workflow context stack	Yes	true
Mandatory Attribute	WORK-FLOW_ITEM_ KEY	string	the transaction's Workflow item key	Yes	N/A
Mandatory Attribute	WORK-FLOW_ITEM_ TYPE	string	the transaction's Workflow item type	Yes	N/A
Non-mandatory Header Attribute	ALLOW_ EMPTY_A PPROVAL_ GROUPS	boolean	whether to allow approval groups to have no members	Yes	false
Non-mandatory Header Attribute	INCLUDE_ ALL_JOB_ LEVEL_AP PROVERS	boolean	whether to include all approvers at a given job level	Yes	false

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-mandatory Header Attribute	TOP_SUPERVISOR_PERSON_ID	number	person ID of the top person in the HR supervisory hierarchy	Yes	N/A
Non-mandatory Header Attribute	TRANSACTION_GROUP_ID	number	business-group ID in which transaction occurred	Yes	N/A
Non-mandatory Header Attribute	TRANS-ACTION_REQUESTOR_PERSON_ID	number	person ID of person initiating transaction, if any	Yes	N/A
Non-mandatory Header Attribute	TRANSACTION_SET_OF_BOOKS_ID	number	set-of-books ID in which transaction occurred	Yes	N/A

Seeded transaction attributes for the following transaction types:

- PO ERES Receiving Inspection (oracle.apps.po.rcv.inspect)
- PO ERES Receiving Transfer (oracle.apps.po.rcv.transfer)
- PO ERES Receiving Delivery (oracle.apps.po.rcv.deliver)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	PO_ERES_CATALOG_GROUP	string	Approval based on the Catalog Group	No	select micgk.concatenated_segments from MTL_ITEM_CATALOG_GROUPS_KFV micgk, MTL_SYSTEM_ITEMS msi, qa_eres_rcv_trans_interface_v qertiv where qertiv.PARENT_TRANSACTION_ID = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and qertiv.QA_COLLECTION_ID = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1)) and rownum = 1 and qertiv.item_id = msi.inventory_item_id(+) and qertiv.organization_id = msi.organization_id(+) and msi.item_catalog_group_id = micgk.item_catalog_group_id(+)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	PO_ERES_ CUSTOMER	string	Approval based on the Customer	No	select customer from qa_ eres_rcv_trans_interface_ v where PARENT_ TRANSACTION_ID = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and QA_COLLECTION_ID = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1)) and rownum = 1
Non-Mandatory Header Attribute	PO_ERES_ CUSTOMER_ ITEM	string	Approval based on the Customer Item	No	select customer_item from qa_eres_rcv_trans_ interface_v where PARENT_ TRANSACTION_ID = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and QA_COLLECTION_ID = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1)) and rownum = 1
Non-Mandatory Header Attribute	PO_ERES_ HAZARD_ CLASS	string	Approval based on the Hazard Class	No	select hazard_class from qa_ eres_rcv_trans_interface_ v where PARENT_ TRANSACTION_ID = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and QA_COLLECTION_ID = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1)) and rownum = 1
Non-Mandatory Header Attribute	PO_ERES_ITEM	string	Approval based on the Item	No	select item from qa_eres_ rcv_trans_interface_v where PARENT_TRANSACTION_ ID = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and QA_COLLECTION_ID = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1)) and rownum = 1
Non-Mandatory Header Attribute	PO_ERES_ITEM_ REVISION	string	Approval based on the Item Revision	No	select item_revision from qa_eres_rcv_trans_ interface_v where PARENT_ TRANSACTION_ID = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and QA_COLLECTION_ID = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1)) and rownum = 1

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	PO_ERES_ LOCATOR	string	Approval based on the Locator	No	select locator from qa_eres_ rcv_trans_interface_v where PARENT_TRANSACTION_ ID = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and QA_COLLECTION_ID = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1)) and rownum = 1
Non-Mandatory Header Attribute	PO_ERES_ PROJECT	string	Approval based on the Project	No	select project_number from qa_eres_rcv_trans_ interface_v where PARENT_ TRANSACTION_ID = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and QA_COLLECTION_ID = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1)) and rownum = 1
Non-Mandatory Header Attribute	PO_ERES_RECEI PT_ROUTING	string	Approval based on the Receipt Routing	No	select receipt_routing from qa_eres_rcv_trans_ interface_v where PARENT_ TRANSACTION_ID = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and QA_COLLECTION_ID = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1)) and rownum = 1
Non-Mandatory Header Attribute	PO_ERES_SUB INVENTORY	string	Approval based on the Subinventory	No	select subinventory from qa_eres_rcv_trans_ interface_v where PARENT_ TRANSACTION_ID = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and QA_COLLECTION_ID = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1)) and rownum = 1
Non-Mandatory Header Attribute	PO_ERES_SUPPL IER	string	Approval based on the Supplier	No	select supplier from qa_eres_ rcv_trans_interface_v where PARENT_TRANSACTION_ ID = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and QA_COLLECTION_ID = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1)) and rownum = 1

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-mandatory Header Attribute	PO_ERES_SUPPLIER_ITEM	string	Approval based on the Supplier Item	No	select supplier_item from qa_eres_rcv_trans_interface_v where PARENT_TRANSACTION_ID = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and QA_COLLECTION_ID = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1)) and rownum = 1
Non-mandatory Header Attribute	PO_ERES_TASK	string	Approval based on the Task	No	select task_number from qa_eres_rcv_trans_interface_v where PARENT_TRANSACTION_ID = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and QA_COLLECTION_ID = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1)) and rownum = 1
Non-mandatory Header Attribute	PO_ERES_TRANSACTION_TYPE	string	Approval based on the Transaction Type	No	select order_type from qa_eres_rcv_trans_interface_v where PARENT_TRANSACTION_ID = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and QA_COLLECTION_ID = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1)) and rownum = 1
Non-mandatory Header Attribute	PO_ERES_UN_NUMBER	string	Approval based on the UN Number	No	select un_number from qa_eres_rcv_trans_interface_v where PARENT_TRANSACTION_ID = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and QA_COLLECTION_ID = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1)) and rownum = 1
Non-mandatory Header Attribute	TRANSACTION_DATE	date	date transaction occurred	No	select ame_util.versionDateToString(CREATION_DATE) from QA_ERES_RCV_TRANS_INTERFACE_V where PARENT_TRANSACTION_ID = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and QA_COLLECTION_ID = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1)) and rownum = 1

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-mandatory Header Attribute	TRANSACTION_ ORG_ID	number	org ID in which transaction occurred	No	select organization_id from QA_ERES_RCV_TRANS_ INTERFACE_V where PARENT_TRANSACTION_ ID = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and QA_COLLECTION_ID = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1)) and rownum = 1
Non-mandatory Header Attribute	TRANSACTION_ REQUESTOR_ USER_ID	number	user ID of user initiating transaction, if any	No	select Last_updated_by from QA_ERES_RCV_TRANS_ INTERFACE_V where PARENT_TRANSACTION_ ID = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and QA_COLLECTION_ID = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1)) and rownum = 1
Non-mandatory Line Item Attribute	PO_ERES_ALL_ ITEM_CATEGOR IES	string	Approval based on all possible Categories for an Item	No	select concatenated_ segments from mtl_ categories_kfv where category_id in (select mic.category_id from mtl_ item_categories mic, (select organization_id, item_id from qa_eres_rcv_trans_ interface_v where PARENT_ TRANSACTION_ID = substr(:transactionId, 1, (instr(:transactionId, '-')-1)) and QA_COLLECTION_ID = substr(:transactionId, (instr(:transactionId, '-', 1, 1)+1)) and rownum = 1) qrti where qrti.organization_id = mic.organization_id(+) and qrti.item_id = mic. inventory_item_id(+)) order by category_id

Seeded transaction attributes for the following transaction types:

- PO ERES ASL Creation (oracle.apps.po.asl.create)
- PO ERES ASL Update (oracle.apps.po.asl.update)

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-Mandatory Header Attribute	PO_ERES_ITEM	string	Approval based on the Item	No	select item_num from PO_APPROVED_SUPPLIER_LIST_V where asl_id = :transactionId
Non-Mandatory Header Attribute	PO_ERES_MANUFACTURER	string	Approval based on the Manufacturer	No	select asl_manufacturer from PO_APPROVED_SUPPLIER_LIST_V where asl_id = :transactionId
Non-Mandatory Header Attribute	PO_ERES_SUPPLIER	string	Approval based on the Supplier	No	select vendor_name from PO_APPROVED_SUPPLIER_LIST_V where asl_id = :transactionId
Non-mandatory Header Attribute	PO_ERES_SUPPLIER_ITEM	string	Approval based on the Supplier Item	No	select primary_vendor_item from PO_APPROVED_SUPPLIER_LIST_V where asl_id = :transactionId
Non-mandatory Header Attribute	TRANSACTION_DATE	date	date transaction occurred	No	select ame_util.versionDateToString(CREATION_DATE) from PO_APPROVED_SUPPLIER_LIST_V where asl_id = :transactionId
Non-mandatory Header Attribute	TRANSACTION_ORG_ID	number	org ID in which transaction occurred	No	select owning_organization_id from PO_APPROVED_SUPPLIER_LIST_V where asl_id = :transactionId
Non-mandatory Header Attribute	TRANSACTION_REQUESTOR_USER_ID	number	user ID of user initiating transaction, if any	No	select Last_updated_by from PO_APPROVED_SUPPLIER_LIST_V where asl_id = :transactionId
Non-mandatory Line Item Attribute	PO_ERES_ALL_ITEM_CATEGORIES	string	Approval based on all possible Categories for an Item	No	select concatenated_segments from mtl_categories_kfv where category_id in (select mic.category_id from mtl_item_categories mic, (select owning_organization_id, item_id from po_approved_supplier_list_v where asl_id = :transactionId) paslv where paslv.owning_organization_id = mic.organization_id(+) and paslv.item_id = mic.inventory_item_id(+)) order by category_id

Define Rules/Associate Approvers

No rules or approvers are seeded. As part of the implementation, you must complete these activities (see: Setting Up Oracle Approvals Management, *Oracle E-Records Implementation Guide*).

Oracle E-Records Seeded Data

Use generic query attributes to search for e-records and their associated documents in the Evidence Store. For instructions on how to search, see Evidence Store, *Oracle*

E-Records Implementation Guide. The Query Element Type equals Generic for all attributes listed below.

PO ERES ASL Creation and Update

Application Code	XML Element Tag	Display Name
QA	VENDOR_NAME	Supplier

PO ERES Receiving Delivery, Inspection, and Transfer

Application Code	XML Element Tag	Display Name
PO	RECEIPT_ROUTING	Receipt Routing
PO	TRANSACTION_DATE	Transaction Date
PO	SUPPLIER	Supplier
PO	SUPPLIER_ITEM	Supplier Item
PO	CUSTOMER	Customer
PO	CUSTOMER_ITEM	Customer Item
PO	HAZARD_CLASS	Hazard Class
QA	VENDOR_NAME	Supplier
QA	JOB_NAME	Job
QA	CUSTOMER_NAME	Customer
QA	COMP_ITEM	Component Item

Event Data for Oracle Shipping

Oracle Workflow Seeded Data

Events

Name	Display Name	Description	Status	Owner Name	Owner Tag
oracle.apps.wsh.eres.delivery.shipment	WSH ERES Delivery Shipment	ERES Event for Deliveries in a Closed or In-Transit Status in Oracle Shipping	Enabled	Oracle Quality	QA

Event Key

Event Name	User Event Key (Identifier)
oracle.apps.wsh.eres.delivery.shipmen	Shipment Number

Event Subscription

All events subscriptions have the same values for the following fields:

- System = HM001
- Source Type = Local
- Phase = 0
- Status = Disabled
- Rule Data = Key
- Rule Function = EDR_PSIG_RULE.PSIG_RULE
- Priority = Normal

Event Filter	Parameters
oracle.apps.wsh.eres.delivery.shipment	EDR_XML_MAP_CODE=qa_wsh_eres_ delivery_shipment EDR_AME_TRANSACT ION_TYPE=oracle.apps.wsh.eres.delivery. shipment

Oracle Approvals Management Seeded Data

Transaction Type

All transaction types listed below belong to the Oracle Shipping application.

Transaction Type ID	Transaction Type Description	Line Item Id Query String
oracle.apps.wsh.eres.delivery. shipment	WSH ERES Delivery Shipment	select delivery_detail_id from qa_eres_wsh_deliverables_ v where delivery_id = :transactionId order by delivery_detail_id asc

Transaction Attributes

Although the user can define their own attributes for the transaction type listed above, the more commonly used attributes are seeded.

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Mandatory Attribute	AL- LOW_DELET ING_RULE_ GENERATED_ APPROVERS	boolean	whether to let the calling application (or its end users) delete approvers generated by the rules	Yes	false
Mandatory Attribute	ALLOW_ REQUESTOR_ APPROVAL	boolean	whether to allow requestors to approve their own transactions (when the rules do so)	Yes	false

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Mandatory Attribute	AT_LEAST_ ONE_RULE_ MUST_APPLY	boolean	whether to require that at least one rule apply to each transaction	Yes	false
Mandatory Attribute	EFFECTIVE_ RULE_DATE	date	the date that determines which rules are active	Yes	N/A
Mandatory Attribute	EVALUATE_ PRIORITIES_ PER_ITEM	boolean	whether to evaluate rule priorities per item under strict item evaluation	Yes	false
Mandatory Attribute	REJECTION_ RESPONSE	string	how AME responds to a rejection	Yes	STOP_ALL_ITEMS
Mandatory Attribute	USE_RESTR ICTIVE_ ITEM_ EVALUAT ION	boolean	whether to require that the same item satisfy all item conditions in a given rule	Yes	false
Mandatory Attribute	USE_ WORKFLOW	boolean	whether OAM should log exceptions to the Workflow context stack	Yes	true
Mandatory Attribute	WORK- FLOW_ITEM_ KEY	string	the transaction's Workflow item key	Yes	N/A
Mandatory Attribute	WORK- FLOW_ITEM_ TYPE	string	the transaction's Workflow item type	Yes	N/A
Non-mandatory Header Attribute	ALLOW_ EMPTY_A PPROVAL_ GROUPS	boolean	whether to allow approval groups to have no members	Yes	false
Non-mandatory Header Attribute	INCLUDE_ ALL_JOB_ LEVEL_AP PROVERS	boolean	whether to include all approvers at a given job level	Yes	false
Non-mandatory Header Attribute	TOP_SU PERVISOR_ PERSON_ID	number	person ID of the top person in the HR supervisory hierarchy	Yes	N/A

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-mandatory Header Attribute	TRANSACTION_DATE	date	date transaction occurred	No	select ame_util.versionDateToString(CREATION_DATE) from qa_eres_wsh_new_deliveries_v where delivery_id = :transactionId
Non-mandatory Header Attribute	TRANSACTION_GROUP_ID	number	business-group ID in which transaction occurred	Yes	N/A
Non-mandatory Header Attribute	TRANSACTION_ORG_ID	number	org ID in which transaction occurred	No	select organization_id from qa_eres_wsh_new_deliveries_v where delivery_id = :transactionId
Non-mandatory Header Attribute	TRANSACTION_REQUESTOR_PERSON_ID	number	person ID of person initiating transaction, if any	Yes	N/A
Non-mandatory Header Attribute	TRANSACTION_REQUESTOR_USER_ID	number	user ID of user initiating transaction, if any	No	select Last_updated_by from qa_eres_wsh_new_deliveries_v where delivery_id = :transactionId
Non-mandatory Header Attribute	TRANSACTION_SET_OF_BOOKS_ID	number	set-of-books ID in which transaction occurred	Yes	N/A
Non-mandatory Header Attribute	WSH_DELIVERY_NAME	string	Approval based on the Delivery Name	No	select delivery_name from qa_eres_wsh_new_deliveries_v where delivery_id = :transactionId
Non-mandatory Header Attribute	WSH_SHIP_TO_LOCATION	string	Approval based on the Ship To Location	No	select ultimate_ship_to from qa_eres_wsh_new_deliveries_v where delivery_id = :transactionId

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-mandatory Line Item Attribute	WSH_ CATALOG_ GROUPS	string	Approval based on all possible Catalog Groups for all Items in the Delivery	No	select micgk. concatenated_segments from MTL_ITEM_ CATALOG_GROUPS_ KFV micgk, MTL_ SYSTEM_ITEMS msi, (select organization_id, item_id from qa_eres_ wsh_deliverables_v where delivery_detail_ id in (select delivery_ detail_id from qa_eres_ wsh_deliverables_v where delivery_id = :transactionId) order by delivery_detail_id ASC) dl where dl. organization_id = msi. organization_id(+) and dl.item_id = msi. inventory_item_id(+) and msi.item_catalog_ group_id = micgk.item_ catalog_group_id(+)
Non-mandatory Line Item Attribute	WSH_ITEMS	string	Approval based on all Items in the Delivery	No	select item from qa_ eres_wsh_deliverables_ v where delivery_ detail_id in (select delivery_detail_ id from qa_eres_ wsh_deliverables_v where delivery_id = : transactionId) order by delivery_detail_id ASC
Non-mandatory Line Item Attribute	WSH_ITEM_ REVISIONS	string	Approval based on the Revisions of all Items in the Delivery	No	select item_revision from qa_eres_wsh_ deliverables_v where delivery_detail_id in (select delivery_ detail_id from qa_eres_ wsh_deliverables_v where delivery_id = : transactionId) order by delivery_detail_id ASC

Attribute Category	Attribute Name	Attribute Type	Description	Static Usage	Usage
Non-mandatory Line Item Attribute	WSH_ LOCATORS	string	Approval based on all Locators in the Delivery	No	select locator from qa_ eres_wsh_deliverables_ v where delivery_ detail_id in (select delivery_detail_ id from qa_eres_ wsh_deliverables_v where delivery_id = : transactionId) order by delivery_detail_id ASC
Non-mandatory Line Item Attribute	WSH_SUB INVENTOR IES	string	Approval based on all Subinventories in the Delivery	No	select subinventory from qa_eres_wsh_ deliverables_v where delivery_detail_id in (select delivery_ detail_id from qa_eres_ wsh_deliverables_v where delivery_id = : transactionId) order by delivery_detail_id ASC

Define Rules/Associate Approvers

No rules or approvers are seeded. As part of the implementation, you must complete these activities (see: Setting Up Oracle Approvals Management, *Oracle E-Records Implementation Guide*).

Oracle E-Records Seeded Data

There are no generic query attributes to use while searching for Oracle Shipping e-records.

Index

T

W