Chapter 5

Information Systems Development Methodology

for the

Environmental Screening Tool

Table of Contents

Environmental Screening Tool

Design Stage

Design Document

Draft User Manual

Draft Test Plan

Updated Project Schedule

Project Review Issue Lists

Project Review Management Reports

Project Revision Log

Stage-end Walkthrough Forms

Design Document

Table of Contents

Tables

Figures

Attachments

Appendices

The appendices referenced in this Design Document are provided following Chapter 9 of this ISDM Documentation submittal.

The Environmental Screening Tool (EST) is a fundamental component of the new Efficient Transportation Decision Making (ETDM) Process. As such, its development occurred while the new business process was being defined. This produced a very flexible environment in which the process could be refined to take advantage of technology, and the technology could be easily adjusted as process details were defined. It also presented the team with the challenge of developing a complex application while the work process requirements were still evolving. The team addressed this challenge by designing for change and developing the application incrementally in a series of modules using an evolving prototype model for the development methodology. This is a life-cycle model in which the system is developed in increments so that it can be modified in response to customer feedback. Unlike other types of prototyping, the prototype code is not discarded; instead, it evolves into the code that is ultimately delivered. In the EST, the database design emphasizes flexibility so that the application can be easily adapted as the process is adjusted. Each module was developed by starting with the basic requirements and adding complexity as the process was refined. This allowed frequent opportunities for the Steering Committee and potential users to review and respond to the application as it was being developed. The end result is a toolbox of customized applications that support the ETDM Process. Because of the need to respond rapidly as system requirements were defined, the prototype served as a de facto design specification, which was later documented in the user handbook. Although the initial release of the EST was well received, it was anticipated that modification would be needed to reflect further refinement of the new business process after it had been used during the first year. Based on feedback from users during the first year of implementation, a new integrated design of the EST was developed to improve the graphical user interface, code maintainability, and user work flow. The new design also takes advantage of technology advancements and upgrades made available since the conception of the project.

This design document reflects the updated design of the secure EST Web site application.

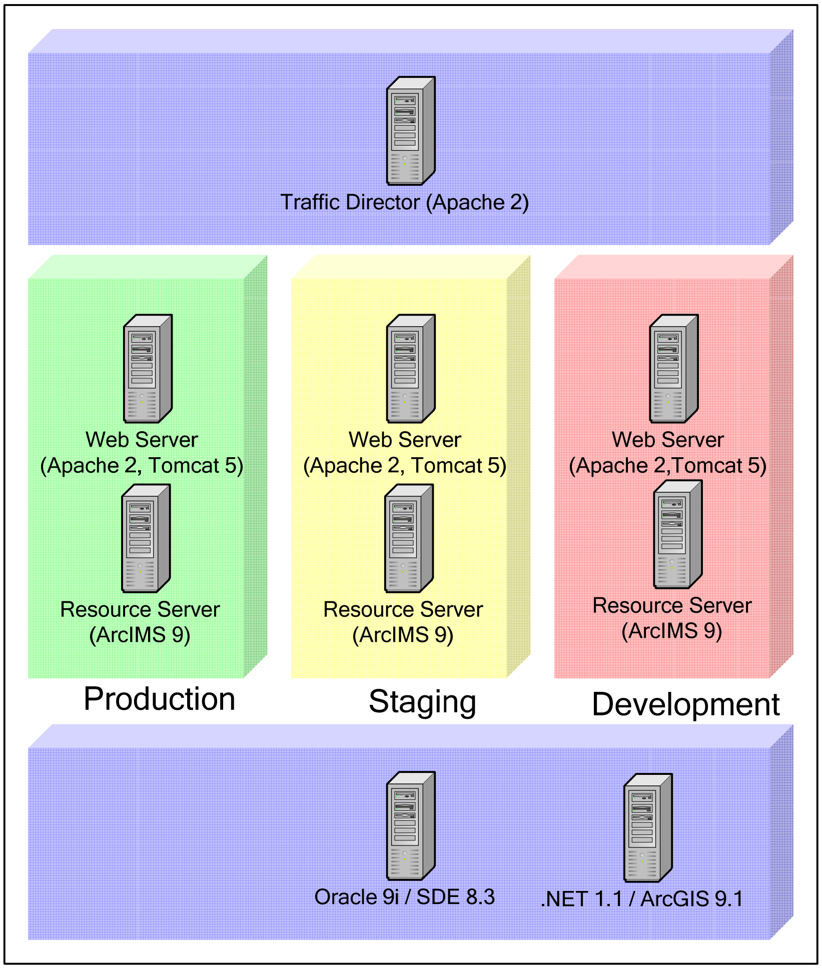
Chapter 1 Preliminary Design (Updated 7/31/2009)

1.1 Conceptual High Level Architecture

This section includes a description of the overall application architecture.

1.1.1 Server Configuration (Updated 11/30/2007)

Figure 1-1 Server Configuration



The above diagram (**Figure 1-1**) represents the server configuration for the Environmental Screening Tool. In this configuration, 7 server boxes are utilized (not including the firewall, which is not shown in this illustration). The configuration is as follows:

1. Traffic Director: Runs Apache 2.0 for purposes of accepting all incoming port ‘80’ and port ‘443’ traffic which comes through the firewall. SSL (port 443) traffic is encrypted and decrypted here. Traffic is then directed to one of the web servers (production, staging, or development) depending on the host name requested. This direction is accomplished via a proxy module (mod_proxy) installed on Apache.

2. Production Web Server: Runs Apache 2.0 and Tomcat 5.5 (With Java1.5). Live server that users access via ‘www.fla-etat.org’. Dependent on the production ArcIMS server and the Oracle 9i/SDE 8.3 server.

3. Production ArcIMS Server: Runs ArcIMS 9 and is dependent on the Oracle 9i/SDE 8.3 server.

4. Staging Web Server: Runs the same software as the production web server and is almost always completely identical to it except for minor naming configuration differences. This server is to be used to test **all** new deployments before they are deployed to the production web server. This

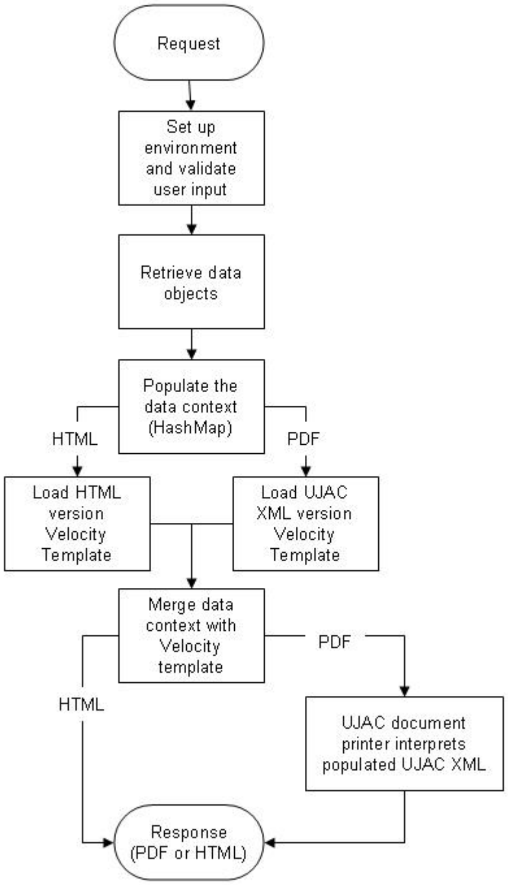
server may be configured to connect to any ArcIMS and any Oracle database server on the network, depending on the nature of the deployment to be tested. This server is also to be used to test application software upgrades prior to performing the same upgrade on the production web server.

5. Staging ArcIMS Server: Runs ArcIMS 9 and is dependent on the Oracle 9i/SDE 8.3 server.
6. Development Web Server: Runs the same software as the staging web server. This server is used to test latest development code before migrating to the staging server.
7. Development ArcIMS Server: Runs ArcIMS 9 and is dependent on the Oracle 9i/SDE 8.3 server.
8. Database Server: Runs Oracle 9i currently. This server may be used for production, staging, and development databases so long as proper care is taken in the use of different schemas to ensure that production databases do not become corrupted due to development or staging procedures.
9. .NET / ArcGIS server. Runs feature editing services and hardcopy map generation services.

1.1.2 Object Class Model

The high level graphical representation of the class model used with the Environmental Screening Tool is provided in **Appendix B (Volume 6)**.

1.1.3 Web Application Flow (Updated 11/30/2007)


Request
Set up environment and validate user input
Retrieve data objects
Populate the data context (HashMap)
HTML
PDF
Load HTML version Velocity Template
Load UJAC XML version Velocity Template
Merge data context with Velocity template
PDF
HTML
UJAC document printer interprets populated UJAC XML
Response (PDF or HTML)

The flow chart to the left illustrates the typical flow of information within the EST web application.

1. A request is sent to the Apache HTTP server. If the request is for a Servlet URL, the request is then forwarded to Apache Tomcat via the JK Tomcat Connector.
2. All necessary resources are made accessible to the Servlet via the various scopes (Application, Session, Request, Response). This includes file paths, database connections, and context objects. Specific variables to be handled by this Servlet are checked for validity.
3. Oracle SQL is abstracted from the application via Hibernate. Data objects are returned instead of a JDBC ResultSet.
4. Data Requested from the database is processed and placed into a tree of objects (data context), implemented as a HashMap.
5. A set of Velocity templates predefines the layout within which data from the data context is to be placed. Velocity is used to create text-based output of either HTML or XML format. When the user's request is for HTML output, a web browser viewable page is generated by the Velocity template being merged with the data context. When the user's request is for PDF output, first a UJAC-compatible XML file must be generated. This XML file is generated in the same manner as the above HTML output, but uses an XML format defined by the UJAC specification. This XML

output is then processed through UJAC which in turn uses the iText PDF output engine. Since UJAC is also capable of accepting a data context and handling some additional dynamic formatting options, the data context is passed to UJAC in addition to having already been passed through Velocity previously.

6. The response is sent back through Tomcat and then through Apache and finally through the user's web browser. In the case of PDF output, HTTP headers for the Adobe Acrobat type are set. The web browser will launch Adobe Acrobat automatically to view the output.

1.2 Graphical User Interface Design *(Updated 01/31/2008)*

This section describes the graphical user interface (GUI), including application menus, forms, reports, and static pages. It includes standards used to develop these elements and example graphics depicting each type of element.

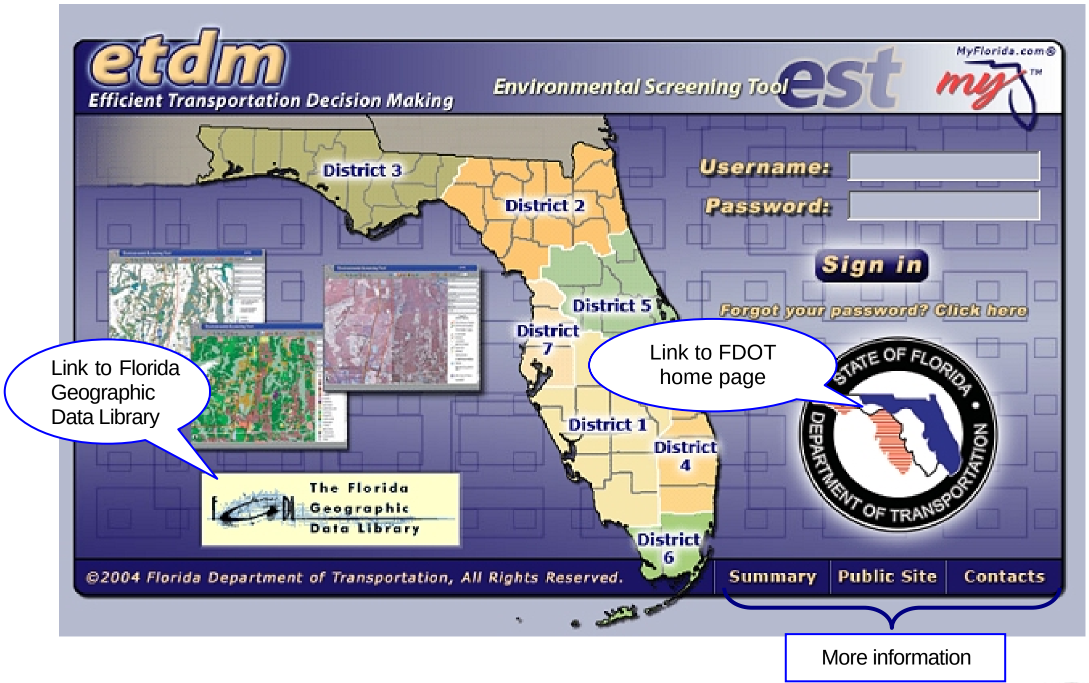
Log-on Screen (Splash Page)

The EST is a secure Web application. The log-on screen, or splash page, contains fields to log on to the Environmental Screening Tool, as well as several links to obtain more information:

- Summary - Link to a general overview of the Environmental Screening Tool
- Public Site - Link to the Public Access site home page
- Contacts – Link to a list of names, phone numbers, and email addresses of people to contact for more information about the EST and ETDM process
- Florida Geographic Data Library logo - Link to the Florida Geographic Data Library (FGDL) home page
- State of Florida Department of Transportation (FDOT) logo - Link to the FDOT home page

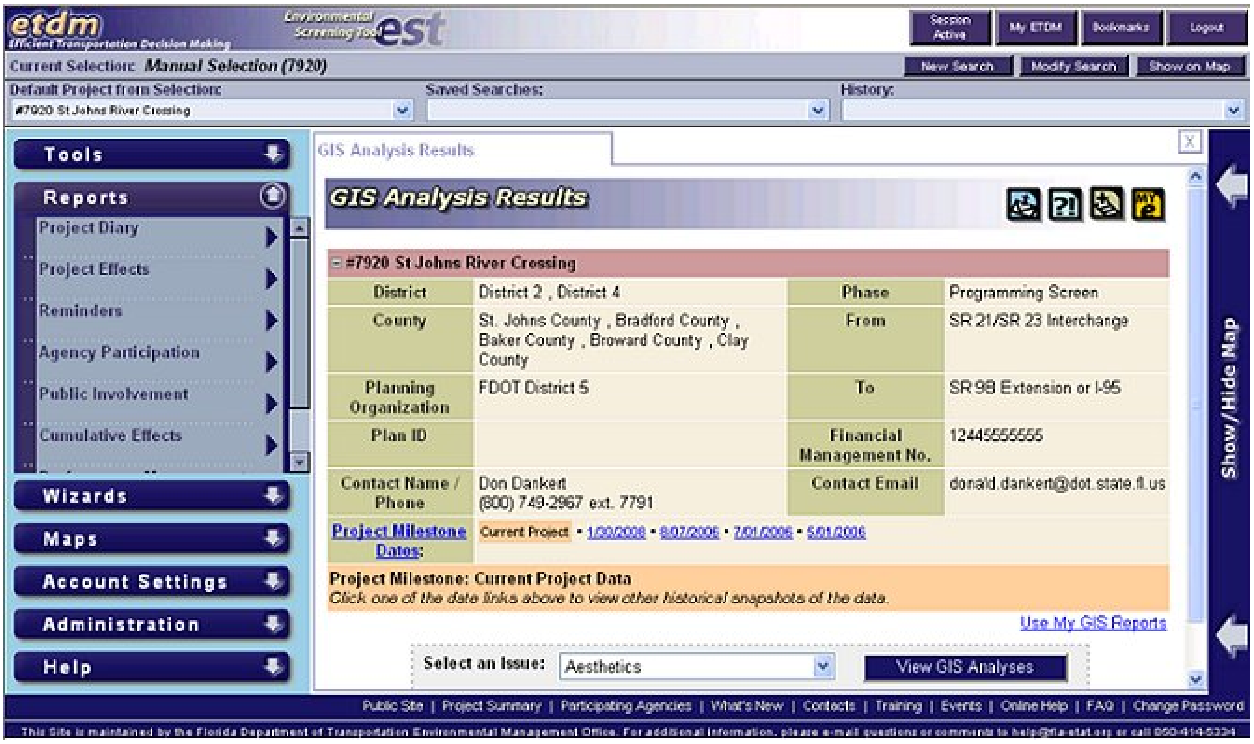
The splash page is illustrated in **Figure 1-2**.

Figure 1-2 EST Log-on Page


etdm
Efficient Transportation Decision Making
Environmental Screening Tool
est
MyFlorida.com®
my
Username:
Password:
Sign in
Forgot your password? Click here
District 3
District 2
District 5
District 7
District 1
District 4
District 6
The Florida Geographic Data Library
STATE OF FLORIDA DEPARTMENT OF TRANSPORTATION
©2004 Florida Department of Transportation, All Rights Reserved.
Summary
Public Site
Contacts
Link to Florida Geographic Data Library
Link to FDOT home page
More information

After logging on to the site, the core navigation elements on the EST Web site include a top tool bar, project navigation bar, collapsible left-side menu, and bottom tool bar. Navigation among multiple open pages is accomplished through the use of tabs and slide-out maps. **Figure 1-3** shows an overview of the navigational elements as used with a report.

Figure 1-3 Overview of Navigational Components



Current Selection: *Manual Selection (7920)*

Default Project from Selection: #7920 St Johns River Crossing

Saved Searches:

History:

GIS Analysis Results

#7920 St Johns River Crossing

District	District 2 , District 4	Phase	Programming Screen
County	St. Johns County , Bradford County , Baker County , Broward County , Clay County	From	SR 21/SR 23 Interchange
Planning Organization	FDOT District 5	To	SR 98 Extension or I-95
Plan ID		Financial Management No.	12445555555
Contact Name / Phone	Don Dankert (800) 749-2967 ext. 7791	Contact Email	donald.dankert@dot.state.fl.us
Project Milestone Dates:	Current Project • 1/30/2008 • 8/07/2006 • 7/01/2006 • 5/01/2006		

Project Milestone: Current Project Data

Click one of the date links above to view other historical snapshots of the data.

Use My GIS Reports

Select an Issue: Aesthetics

View GIS Analyses

This Site is maintained by the Florida Department of Transportation Environmental Management Office. For additional information, please e-mail questions or comments to help@fla-etat.org or call 850-414-5334

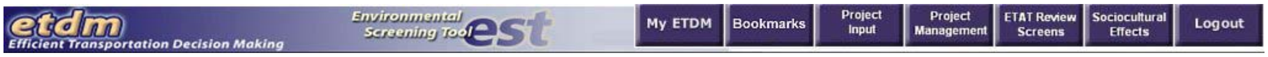
Top Tool Bar

The top tool bar provides quick access to certain pages and functions in the EST. Initially, it maintained access to forms and reports through the existing modules. After approximately four to six months, access through the modules was removed as users became accustomed to new methods of navigation to forms and reports through the new left-side menu.

Phase 1. Access to Frequently Used Pages and Existing Modules

In the initial release of the converted GUI, access to forms and reports will remain available through the existing EST modules via menus on the top tool bar. The only access point to be removed from the existing top tool bar is the "Admin Tools" menu, which is used by the development team. The existing "Home" button will be replaced with "My ETDM." The "Log Out" button, currently at the bottom right of the screen, will move to the top tool bar, at the right-most position. **Figure 1-4** depicts the top tool bar as implemented in Phase 1.

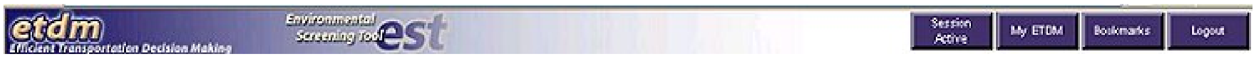
Figure 1-4 Phase 1 Top Tool Bar



Phase 2. Quick Links

In the Phase 2 implementation of the new GUI, the existing links to the modules will be removed from the top tool bar. **Figure 1-5** depicts the top tool bar to be implemented in Phase 2.

Figure 1-5 Phase 2 Top Tool Bar


etdm
Efficient Transportation Decision Making
Environmental Screening Tool
est
Session Active
My ETDM
Bookmarks
Logout

Top Tool Bar Components

- Session Active - Indicates the status of the current EST Session. FDOT security standards require the application to time out after 15 minutes of inactivity. After 13 minutes of inactivity, the EST browser warns the user that the session will soon time out, and the message in the Session Active button changes to "Session Expiring." After 15 minutes of inactivity, the EST browser warns the user that the session has expired, and the message in the Session Active button changes to "Session Expired." The Session Active button also resets the timeout period. The user can reset the timeout to 15 minutes at any time by clicking or moving the mouse over the Session Active button.

- My ETDM – Returns the user to the welcome screen of the EST. This is a page or group of pages that greet the user after initially logging on to the site, or whenever the My ETDM button is clicked. The primary page displays up to four mini-reports selected by the user. Initially, users may select from the items listed in **Table 1-1**. Each user can also choose to open up to five additional form(s), report(s) or static page(s) of their choice. The available selections for these additional pages are listed in **Table 1-2**. The mini-reports and pages are identified by the user from an option found under "Account Settings" in the collapsible left-side menu. A form will allow the user to select one or more mini-reports and/or pages from a checklist of available pages. The list will include some, but not all, pages referenced on any of the navigation bars (left, top or bottom). Pages that are in the middle of a multi-step process may be excluded. The total number of pages added to My ETDM may change, depending on performance considerations. Until the user sets preferences for My ETDM, the main page will include a generic welcome page that provides a site overview and explains how to set preferences for My ETDM. When more than one page opens simultaneously, a row of tabs will appear across the top of the pages to show the open pages. The user navigates among the pages by clicking on the tabs. If the page is the result of a database query, the most recently used query will be used to open the page. Otherwise, default values will be used. The user may change the selection criteria for the query, if needed.

Table 1-1 Mini-Report Selections

Static Information Pages	Reports
What's New (most recent posting)	Agency Resource Issue Matrix (user's agency and issues)
Contacts (appropriate ETDM Coordinator and help desk)	Outstanding Summary Reports (for user's district)
Training (calendar)	Projects Flagged for Dispute (for user's district and no finalized summary report)
Events page (upcoming events)	Projects in Dispute Resolution Process (for user's district)
Frequently Asked Questions (recently added)	ETAT Contact List (for user's ETAT or another choice if in more than one)
	Show Analysis Queue (number of projects in queue)
	Projects Needing ETDM Q/A (for user's district)
	Training Calendar and Registration (list of upcoming events with link to registration page)
	ETAT Review Status Report (for district)

Static Information Pages	Reports
	Finalized Summary Report (newly posted)
	Projects Needing Review (only those not reviewed by user)
	ETAT Comments: Project Effects (in user's jurisdiction, with degree of effect > moderate and no finalized summary report)
	ETAT Comments: Purpose and Need (not understood or not accepted and no finalized summary report)

Table 1-2 Additional Pages Selections

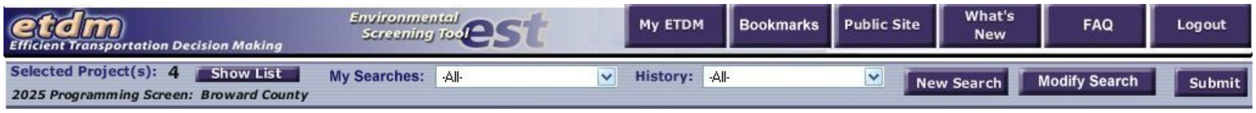
Static Information Pages	Reports
Project Summary	Agency Resource Issue Matrix
Participating Agencies	Agency Review Report
What's New	Outstanding Summary Reports
Contacts - main page	Projects Flagged for Dispute
Training - main page	Projects in Dispute Resolution Process
Events page	ETAT Contact List
Frequently Asked Questions - main page	Show Analysis Queue
	Projects Needing ETDM Q/A
	On-line Help
	ETAT Review Status Report
	Finalized Summary Report
	Projects Needing Review
	ETAT Comments: Ad Hoc

- Bookmarks – Used to quickly open pages on the EST site that the user frequently visits. This pull-down menu allows users to add the current page to their list of Bookmarks and lets them open a page by selecting it from the list. These pages may also be identified using the "Manage Bookmarks" option in the collapsible left-side menu, which will allow them to select/deselect one or more pages from a check list of available pages that are referenced on any of the navigation bars (top, left or bottom). The total number of pages added to Bookmarks will not be limited in number.

Project Navigation Bar

Immediately below the Top Tool Bar, a series of tools will be available for project navigation. This data navigation bar will serve as the primary method of selecting ETDM projects or regions. The current selection will be used to determine the availability of left-side menu items, as well as the data that will be displayed or prompted for when menu items are selected. This will provide the ability to the user to move from section to section without having to repeatedly search for a project. The Project Navigation Bar will be implemented during Phase 1 and will remain available through final implementation. The Project Navigation Bar is shown in **Figure 1-6**.

Figure 1-6 Project Navigation Bar with Top Tool Bar



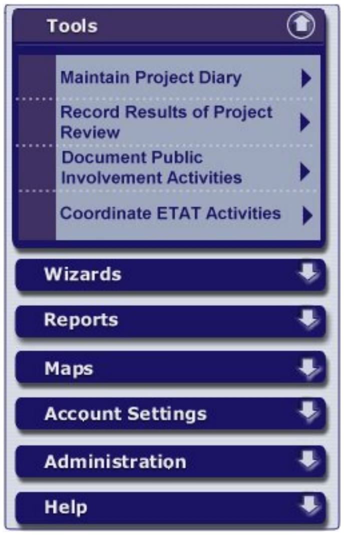
Selected Project(s): 4
2025 Programming Screen: Broward County
My Searches: -All-
History: -All-

Project Navigation Tool Bar Components

- Project List – A pull-down menu of selected projects including County, District, ETDM#, Project Name, ETDM Phase, ETDM Status, and Date of Last ETAT Review. Projects are added to this list from the "New Search" wizard button. The current search result list can be modified using either the New Search or Modify Search buttons.
- Saved Searches - A pull-down list of user-defined selections of projects and project groups. Projects are added to this list from the "New Search" wizard button and removed from this list using the "Modify Search" button.
- History - A system-defined select list of recently accessed projects and project groups.
- New Search – Opens a project selection tool that allows the user to search for and select individual or multiple projects. This selection will be used to determine the starting point whenever the user subsequently activates functions from the Collapsible Left-side Menu. The user may also choose to add the selection to the "Saved Searches" pull-down list. The different selection types are as follows:
  1. Manual Selection: The user selects a project or groups of projects by entering the ETDM number for the project. The user then indicates whether the project or group of projects will be used only during the current session or saved as a selected group. If "Save my search as a selected group" is selected, the selection will be added to the "Saved Searches" pull-down list.
  2. Power Search: The Power Search feature is used to select a group of projects, based on a region and other criteria (Planning Organization, Plan Year, ETDM Status and ETDM Phase). After the user selects the criteria from pull-down menus, a list of all projects matching those criteria is displayed. The user can then edit the list by un-checking the box next to the project name, and indicate whether the project or group of projects will be used only during the current session or saved as either a dynamic query or a selected group.
- Modify Search – A button that opens the project look-up tool, loaded with the selection criteria that define the search. The search criteria can be modified and saved as a new search or as a modification of the existing saved search.
- Show on Map – A button that opens the map viewer zoomed into the currently selected project.

Collapsible Left-side Menu

The left-side menu of the screen will serve as the main entry point to all of the application's forms and reports. Functionality is grouped into major categories based on function type. The major categories will be represented as sliding bars (similar to some versions of Microsoft Outlook and Outlook Express). Each category contains a list of functions (forms, reports, etc). When the list of functions on the user menus exceeds 20 items, they are grouped into sub-categories. Sub-categories will expand to the right when the cursor hovers over the name. Features that are not available to users because of their assigned role will not appear on the menu. The entire menu will collapse to hide it from view, if necessary for report navigation.


Tools
Maintain Project Diary
Record Results of Project Review
Document Public Involvement Activities
Coordinate ETAT Activities
Wizards
Reports
Maps
Account Settings
Administration
Help

Update to Collapsible Left-side Menu System (11/01/2006)

The left-side menu is implemented on the client-side using Javascript, dynamic HTML, and CSS. The server side is implemented using an XML configuration file and a custom XSL transformation.

Tools

The left-side menu tools are used primarily for data entry related functions, which includes all input provided throughout the life cycle of a project. **Table 1-3** lists the options initially available in the Tool category, with a reference to the existing forms.

Table 1-3 Tool Options with Reference to Initial Function

Tool Name and Sub-category	Old Name on Menu or Button
Maintain Project Diary	
Add Alternative Description	Add Alternative Corridor (Project Alternative form)
Add/Modify Plan Summaries	Input System Plan Summaries
Add Project Features to Map	Digitize (on input map viewer)
Assign Project Manager	Assign Project Manager
Attach Documents	Attach Project Documents
Create New Project Record	Create Project (Project Description short form)
Extract Project Features from SHS Map	Add From FDOT Base map (on input map viewer)
Identify Required Permits	Assign Project Permits
Identify Required Technical Studies	Assign Project Technical Studies
Update Commitments/Responses	Update Project Commitments and Responses
Update Project Description	Update Project Details (Project Description long form)
Update Status and Phase	Set Project Status and Phase
Update Segment Description	Show Segment Description (Project Segment form)
Upload GIS Files	Transfer GIS Files - instructions and Upload Utility Form
Record Results of Project Review	
Describe Direct Effects	Direct Effects
Describe Secondary & Cumulative Effects	Secondary & Cumulative Effects
Class of Action Determination	Class of Action Determination
Review Purpose & Need Statement	ETAT Review - Purpose & Need
Summarize ETAT Review Screen	Generate Summary Report
Document Public Involvement Activities	
Add/Modify Community Inventory	Community Characteristics Inventory
Add/Modify Community Focal Point	Community Focal Point Description
Record Community-Desired Features	CLC Desired Project Features
Summarize/Modify Public Comments	Update Summary of Public Comment
Coordinate ETAT Activities	
Allow Comments after Review Period	Grant Agency Overdue Role
Extend ETAT Review Period	Grant Project ETAT Review Extension
Funding Agreement Forms	Participating Agencies (links to forms)
Notify ETAT to Review Projects	ETAT Notification (called from Set Project Status and Phase)
Send Email to User Groups	Send Email
Track Dispute Resolution Activities	Record Dispute Resolution

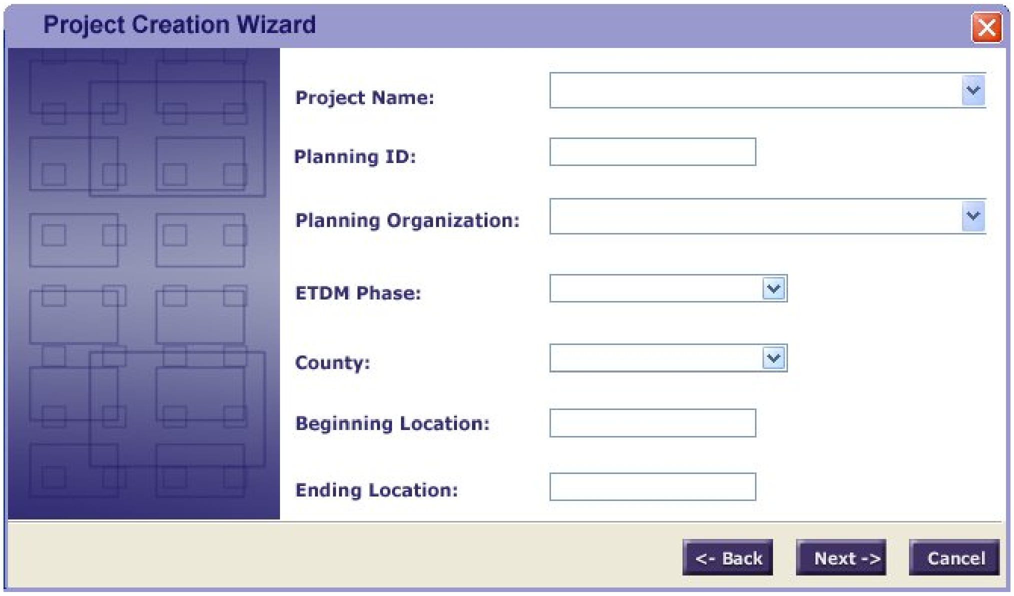
Wizards

Used for more complex input and query functionality, these features will combine tools and reports to step a user through a sequential process. Initially the following Wizards will be available:

- ETAT Review Purpose & Need
- Create ETDM Project
- Summarize ETAT Review Screen

A sample graphic depiction of an input form used in a wizard is shown below as **Figure 1-7**:

Figure 1-7 Project Creation Wizard



Project Creation Wizard

Project Name:	
Planning ID:	
Planning Organization:	
ETDM Phase:	
County:	
Beginning Location:	
Ending Location:	

The Wizards that are currently available (as of 09/30/2007) are:

- Perform ETAT Review
- Create ETDM Project
- Generate Summary Report
- Update ETDM Project
- Set Up User Preferences
- Send ETAT Notifications
- Prepare Invoice

Reports

The Reports menu will initially include standard reports and queries, as listed in **Table 1-4**.

Table 1-4 Reports with Reference to Existing Function

Reports	Old Name on Menu or button
Project Diary	
Community-Desired Features	Community Desired Features
Dispute Resolution Activity Log	Project Dispute Resolution Log
List of Technical Studies	Technical Studies Report
Project Commitments/Responses	Project Commitments and Responses
Project Description	Project Description
Status of GIS Analysis	Show Analysis Queue
Transportation Plan Summary Report	System Plan Summary Report
Project Effects	
Agency Comments - Project Effects	ETAT Comments: Project Effects & ETAT Comments: Ad Hoc Query
Agency Comments - Purpose & Need	ETAT Comments: Purpose and Need & ETAT Comments: Ad Hoc Query
Countywide GIS Summary Report	Countywide GIS Summary Report
GIS Analysis Results	GIS Analysis Results
GIS Analysis History	GIS Analysis History
Screening Summary Report	Finalized Summary Report
Reminders	
Projects Flagged for Dispute	Projects Flagged for Dispute
Projects in Dispute Resolution	Projects in Dispute Resolution Process
Projects Awaiting ETDM QA/QC	Projects Needing ETDM Quality Assurance Review
Un-finalized Summary Reports	Outstanding Summary Reports
Agency Participation	
Agency Annual Reports	ETDM Annual Reports and Program Reviews
Agencies and Associated Resource Issues	Agency Resource Issue Matrix
Agency Review Report	Agency Review Report
Agency Web Sites	Participating Agencies
ETAT Contact List	ETAT Contact List
ETAT Review Status Report	ETAT Review Status Report
Projects Needing Review	Projects Needing Review

Maps

The Maps menu is used to access various types of maps, both interactive and hard copy. Selection of these options would open a search page where a user could specify/update the criteria to be applied to the map.

- View Interactive Maps
- Edit Map Features
- Print Hard Copy Maps

Account Settings

This menu includes the following functions to set user preferences, change password, and other user-specific option management:

- Change Password
- Update Contact Information
- Customize My ETDM Page
- Manage My Searches List
- Manage Bookmarks
- Set Default Layers on Maps
- Change Notification Settings

Administration

This menu is used by the development team for system administrative functionality. Tools available on the "Admin Tools" will be included in this category. Initially, these tools will include:

- Add New Admin Tool
- ETAT Review Totals by Resource Issue [Note: previously named "Agency Issue Matrix"]
- Agency Resource Issue Matrix
- Agency Review Report
- Analysis Log Parser
- Analysis Results
- Change Log
- Change Password
- Copy Updated Analyses
- Create EST User
- Delete Alternative
- Delete Community Point
- Delete Community Polygon
- Delete Project
- EST Design Diagrams
- EST Server Configuration
- EST Technologies
- EST User Report
- ETDM Project Status Report

- HelpDesk Tracker
- Mass Emailer
- New Data Viewer
- Performance Measure Tool
- Task Manager
- Test Digitizer
- Training Report
- View Admin Tool Details [Note: this option links to current Admin Tools view.]
- Write Permissions Rules

Help

Includes the system help and other help-related topics (i.e., FAQ), as shown in **Table 1-5**.

Table 1-5 Help Functions with Reference to Existing Pages

Help	Old Name on Menu or Button
About EST	Project Summary
Contact Us	Contacts
Frequently Asked Questions	FAQ
EST Help	Help
ETDM Library	Organize links to documents found scattered on multiple static info pages
ETDM Meetings and Conferences	Events
Hands-On Training Calendar and Registration	Training

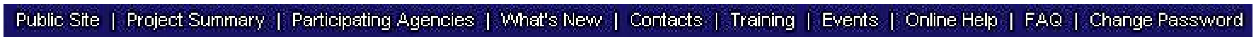
Rules for Determining Menu Item Placement

- If a function updates the database, it should be placed under the Tools section.
- If a function or set of functions can be better presented with a multiple step ”wizard,” a wizard should be written and placed into the Wizards section to supplement the original function(s).
- If a function provides a report, or a query form which results in a report, it should be placed under the Reports section.
- If a function pertains solely to a user’s specific preferences or settings, it should be placed under the Account Settings section.
- If a function is to be used only by the EST development team, it should be placed under the Administration section.
- If a function provides information-only pages that are related to help or topics about the EST or the ETDM Process, it should be placed under the Help section.
- Mapping functions will be placed into the Maps section.

Bottom Tool Bar

The options on the bottom of the EST home page provide links for additional information. Users will click on the text to navigate to these sections. **Figure 1-8** depicts the Bottom Tool Bar.

Figure 1-8 Bottom Tool Bar


Public Site | Project Summary | Participating Agencies | What's New | Contacts | Training | Events | Online Help | FAQ | Change Password

Bottom Tool Bar Components

- Public Access Web Site - Enables read-only access to the ETDM database, serving as a source of information for the public concerning projects.
- Project Summary - Provides an overview of the Environmental Screening Tool as found in Section 1 of the Environmental Screening Tool Handbook. In Phase 2, this page will be renamed “About EST.”
- Participating Agencies - Includes links to the Web sites of each of the 25 agencies which signed the ETDM Memorandum of Understanding. Forms used by agencies with funding agreements are also found here.
- What’s New (will move to top tool bar in Phase 2) - Provides the latest information about enhancements made to the EST.
- Contacts - Lists the phone numbers and email addresses for technical support in using the EST. It also provides contact information for the ETDM Coordinators, CLCs, and ETAT representatives for each District.
- Training - Provides the calendar and materials for hands-on EST training.
- Events – Lists announcements and materials for meetings and conferences related to ETDM and the EST.
- On-line Help – Opens the Internet version of the Environmental Screening Tool Handbook.
- FAQ (will move to top tool bar in Phase 2) - The Frequently Asked Questions (FAQ) page lists answers to common questions about the ETDM Process and the Environmental Screening Tool.
- Change Password (will be available under “Account Settings” in Phase 2) – Enables users to create a new password.

Navigating Among Open Windows

When a user selects an option from one of the navigation bars or menus, the feature opens within the main EST window, between the navigation bars. By default, when a new option is selected, the current page closes and the new feature opens in the main EST window. Alternatively, the user may select the “New Tab” option to open the new feature on a new page, keeping the previously selected pages open. When opening multiple forms or reports in this manner, a tab will appear at the top of the page, similar to a folder within a filing cabinet. The user switches between these pages by clicking on the tab. To close these pages, the user must click on the “Close Page” button which appears at the top of the page. When project selections change in the Project Navigation Bar, the change, by default, only affects the currently active page (i.e. the page on top). The “Update All” option may be selected if the user wants the project selection changes to apply to all tabbed pages. If “Update All” is not selected, the project selection lists for all but the top tabbed page remain the same as when each page was originally opened.

Maps will be opened and moved out of the way by the use of a slide-out window, accessed from the right side of the page. Maps will retain the most recently used data selections and map extent unless the user clicks the

“Show on Map” button to refresh the map using new project selections. For a graphical depiction of tabs and the map slide-out bar, see **Figure 1-2**.

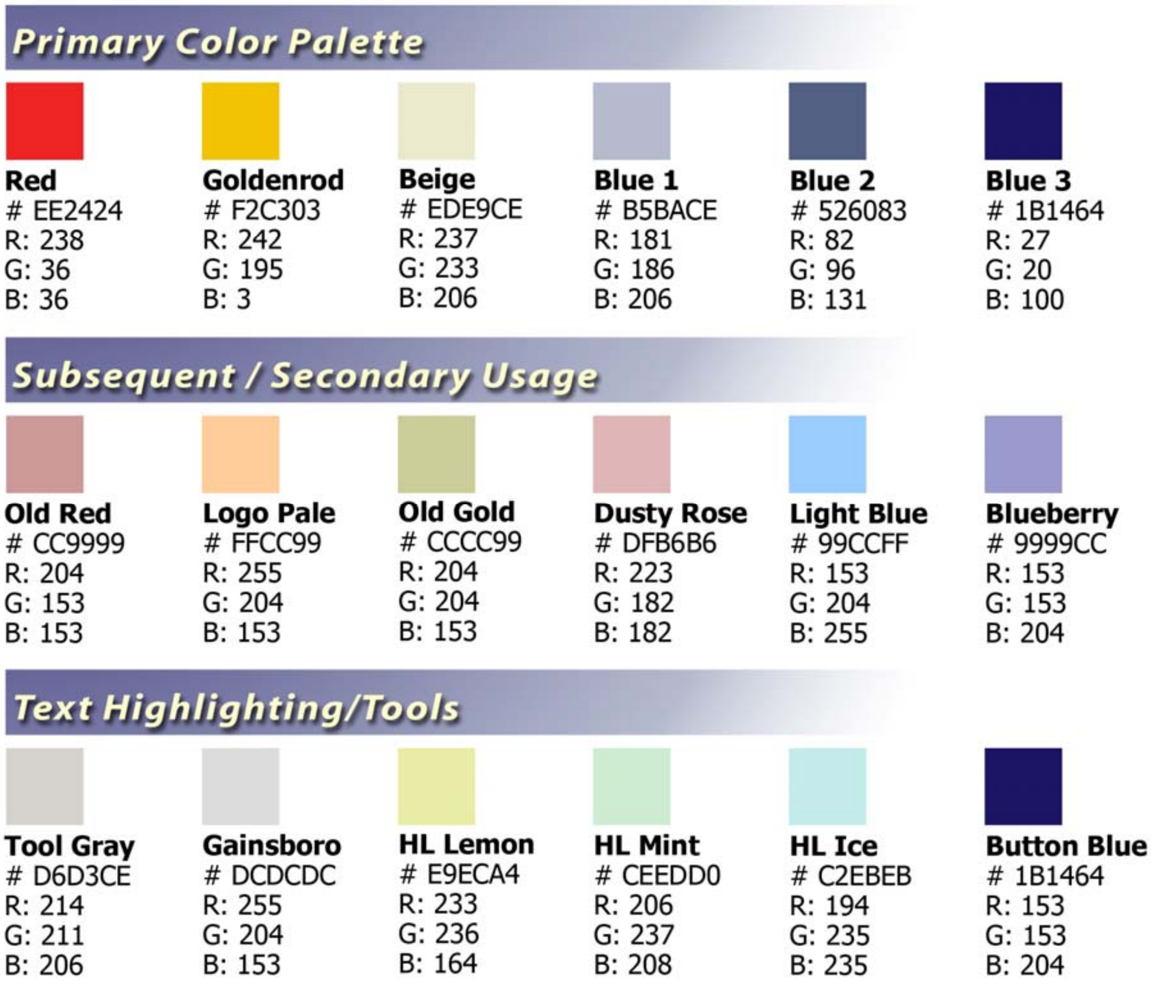
1.2.1 Graphical User Interface Standards (Updated 7/31/2009)

This section identifies the standards used to create a uniform style throughout the EST application.

Colors

The primary color palette chosen for the GUI are three shades of blue with accents of beige, goldenrod, and red. Blues were chosen to complement the FDOT logo. Beige and goldenrod were selected as accents because they are both very complementary to various shades of blue. Red was chosen to complement the “MyFlorida” logo. A secondary color palette was developed for use on pages that require more than the six primary colors. They were selected to complement and accent the primary palette. Both palettes are shown in **Figure 1-9** below.

Figure 1-9 EST Color Palette



Primary Color Palette

Red	Goldenrod	Beige	Blue 1	Blue 2	Blue 3
# EE2424	# F2C303	# EDE9CE	# B5BACE	# 526083	# 1B1464
R: 238	R: 242	R: 237	R: 181	R: 82	R: 27
G: 36	G: 195	G: 233	G: 186	G: 96	G: 20
B: 36	B: 3	B: 206	B: 206	B: 131	B: 100

Subsequent / Secondary Usage

Old Red	Logo Pale	Old Gold	Dusty Rose	Light Blue	Blueberry
# CC9999	# FFCC99	# CCCC99	# DFB6B6	# 99CCFF	# 9999CC
R: 204	R: 255	R: 204	R: 223	R: 153	R: 153
G: 153	G: 204	G: 204	G: 182	G: 204	G: 153
B: 153	B: 153	B: 153	B: 182	B: 255	B: 204

Text Highlighting/Tools

Tool Gray	Gainsboro	HL Lemon	HL Mint	HL Ice	Button Blue
# D6D3CE	# DCDCDC	# E9ECA4	# CEEDD0	# C2EBEB	# 1B1464
R: 214	R: 255	R: 233	R: 206	R: 194	R: 153
G: 211	G: 204	G: 236	G: 237	G: 235	G: 153
B: 206	B: 153	B: 164	B: 208	B: 235	B: 204

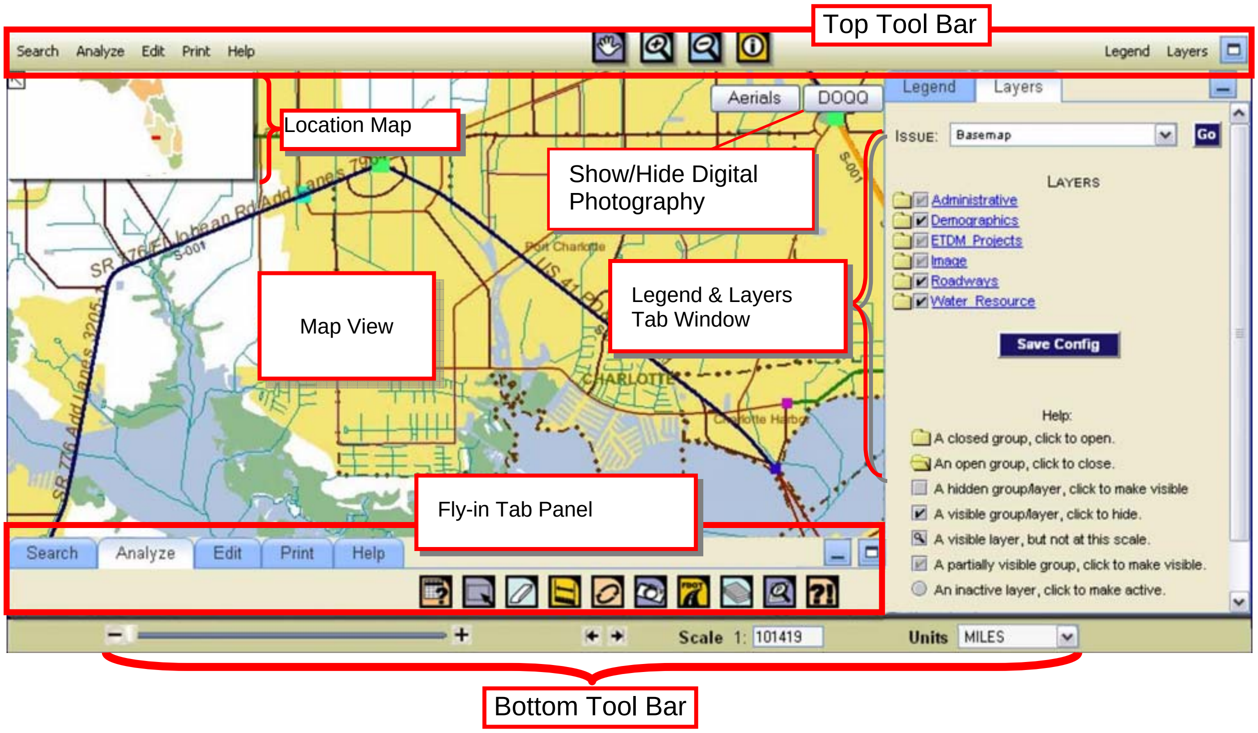
Styles

Styles within the EST application are created using two methods. Primarily, a standard Cascading Style Sheets (CSS) document is referenced on each Web page, providing a single location for changing the aesthetics of the site. The CSS document establishes styles for standard HTML tag types (such as tables and headers). However, CSS alone cannot create alternating styles. Because of this, alternating styles, in particular alternating colors, must be specified within each page. A number of custom Velocity functions have been developed to achieve alternating styles. These Velocity functions make it easy for a programmer to load a series of CSS style classes, and easily deploy them within a page. In summary, CSS and Velocity helper functions are used to create the styles of the EST application.

Map Viewers

The map viewer is an online, interactive mapping application that provides customized tools and maps to help evaluate potential effects for proposed projects. It also provides authorized users with input tools for digitizing Community Characteristics Inventory (CCI) and transportation project records. The map can be accessed using the "Show/Hide Map" bar, the "Show on map" button in the Project Search bar, or from links in other EST pages (Project Description report, for example). An example layout of the interactive map viewer is shown below in **Figure 1-10**.

Figure 1-10 Interactive Map Viewer



Help:
A closed group, click to open.
An open group, click to close.
A hidden group/layer, click to make visible
A visible group/layer, click to hide.
A visible layer, but not at this scale.
A partially visible group, click to make visible.
An inactive layer, click to make active.

GUI Component Guidelines

In addition to best practices, as defined in the ISDM and the STO Web Design Standards, the following guidelines will be used throughout the EST application:

- Reports which the user has sufficient privileges to modify the contents of may be presented to these users with an “edit” button, where appropriate.
- Forms which require a specific individual project to be selected should assume the currently selected project via the top bar, or in cases where multiple projects are selected via the top bar, provide a select list which only contains the projects from the group, with the most recently accessed project from the list automatically selected.
- When users open forms and reports which require selection criteria that have not been specified yet, the user should be prompted with the options to make the selection, rather than just receiving an error message telling them to set the criteria. Any report that results from a query will present the user with an interface to update the search criteria.
- Selection criteria other than project (such as issue or agency) should be displayed on search results pages, and re-used where appropriate.
- All reports will have a “PDF” button which creates an Adobe Acrobat PDF file of the report.
- All reports will have a “Print” button which sends a printer-friendly version of the report to a printer. Printed reports will include a title page with any selection criteria identified.
- All pages will have a “Help” button which links to the appropriate section of the on-line user handbook, which contains instructions on how to use the feature.
- All pages will have a “Feedback” button to send messages to the development team.
- Error messages will be presented as a user-friendly error report, and not as a stack trace. The error report will indicate that an error occurred, and that the development team has been automatically notified of the error. The user is also presented with a feedback form in case they want to provide additional information about the sequence of events that led to the error.
- Titles on reports and forms will match the titles as they appear on the menu that opens them. All menus that call the same page will use the same title.
- Users may sort simple tables by any of the table columns by clicking on the name of the column.

Chapter 2 Detailed Design (Updated 11/30/2007)

2.1 Conceptual Low Level Architecture

2.1.1 Hardware Configuration

This section lists the server software used for the EST Web site.

Host	OS	CPU (GHz)	RAM (MB)
Production Web Server	Red Hat AS3	Xeon 3.0 * 2	2048
Production Resource Server	Red Hat ES3	P4 2.66	2048
Stage Web Server	Red Hat AS3	Xeon 3.0 * 2	2048
Stage Resource Server	Red Hat AS3	P4 2.66	2048
Development Web Server	Red Hat AS4	Xeon 2.8 * 2	1024
Development Resource Server	Red Hat AS3	Xeon 3.0 * 2	2048
.NET / ArcGIS Application Server	Windows 2003	Xeon 2.66	1024
Traffic Director Server	Red Hat ES3	P4 3.0	1024
Database Server	Solaris 2.8	Sun Ultra III 900 * 2	2048

2.1.2 Software Configuration

This section describes the web application software used for the EST Web site.

Apache HTTP Server version 2.0.59

http://httpd.apache.org

Apache HTTP Server is installed on the front end ‘traffic’ server which is the initial entry point for all HTTP and HTTPS traffic. Apache HTTP Server is utilized here with mod_proxy and mod_ssl to handle all SSL encryption/decryption and proxy to the internal production, staging, and development servers.

Apache HTTP Server is installed on the production, staging, and development servers to serve static content. In these installations, mod_jk2 is used to direct web application requests to Tomcat (see below) and mod_proxy is used to proxy requests for “/output” to its assigned ArcIMS server.

Apache HTTP Server is installed on all ArcIMS servers in order to serve images that were generated by the ArcIMS map service.

Apache Jakarta Tomcat version 5.5.16

http://jakarta.apache.org/tomcat

Tomcat is installed on the production, staging, and development servers in order to serve the core web application. Several libraries (jar files - described later) have been installed in order to increase the functionality of the programming environment. The built-in ‘CGI Servlet’ is currently activated and in use in order to support some legacy Perl programs. Eventually the CGI Servlet will be disabled once all of the EST Perl programs have been migrated over to the new Java-based application architecture for the EST.

Apache Velocity 1.4 (jar files) and Apache Velocity Tools 1.2 (jar files)

http://jakarta.apache.org/velocity

Apache Velocity and Apache Velocity Tools jar files are installed on all of the EST's Tomcat installations. The Apache Velocity libraries provides an engine for fast and easy data binding with custom templates and is part of the overall new Java-based application architecture for the EST.

Hibernate 3.0.3 (jar files)

http://www.hibernate.org

Hibernate jar files are installed on all of the EST's Tomcat installations. The Hibernate library provides an engine for object relational mapping and object persistence. In short, Hibernate is used to create a natural relationship between a Java application and the data that it is to access. Hibernate is paired up with Velocity in that Hibernate is used to retrieve data as objects which can be readily passed directly to a Velocity template with little to no manipulation required.

UJAC 0.9.21 with prepackaged iText (jar files)

http://ujac.sourceforge.net

iText is a PDF generation library for Java. UJAC is a library which works on top of iText to convert an easier to create XML file (formatted with special UJAC tags) into a PDF file. UJAC XML can be easily generated via a Velocity template just as Velocity can be used to generate HTML. Both UJAC and iText are installed on all of the EST's Tomcat installations.

Oracle JDBC Drivers (jar files)

The Oracle JDBC Drivers which are packaged with Oracle 9i are installed on all of the EST's Tomcat servers to allow communication between the web application and the Oracle databases installed within the EST network. These drivers may be updated as the EST database is updated to Oracle 10g.

ESRI ArcIMS 9.0.1 Java Connector (jar files) and ESRI ArcIMS 9.0.1 Servlet Connector (jar files)

The Java Connector and Servlet Connector jar files which are packaged with ArcIMS 9.0.1 are installed on all of the EST's Tomcat servers to allow communication between the web application and the ArcIMS servers. These jars may be updated as the ArcIMS servers are upgraded to ArcIMS 9.0.1.

2.1.3 Class Method and Message Documentation

Class properties, methods and messages for the EST are documented using Javadoc at:

https://dev.fla-etat.org/est/javadoc/

Javadoc is a tool for generating documentation in HTML format from doc comments in source code. It is produced by Sun as part of the Java 2 SDK. A copy of the Javadoc program documentation for the EST is included in **Appendix D (Volume 6)**. In addition, a graphical representation of the class model can be found in **Appendix B (Volume 6)**.

2.2 Conceptual Data Model

The conceptual data model is located in the EST Requirements document. This shows the relationships between the logical entities required in the EST. The relationships among the database tables are included in the Schema Design Diagram found in **Appendix C (Volume 6)**. Data elements are defined in the Data Dictionary located in **Appendix A (Volume 5)**.

2.3 Schema Design Diagram (Physical Data Model)

The Schema Design Diagram is located in **Appendix C (Volume 6)**.

2.4 Data Dictionary

The Data Dictionary is included in **Appendix A (Volume 5)**.

2.5 Data Validation Procedures, Referential Integrity Rules, Approaches to Enforcing Business Rules (Updated 11/30/2007)

Constraints are created in the database to enforce a business rule and to specify relationships between various tables. Business rules are also enforced by using database triggers and within the application code. Integrity constraints prevent invalid data from being entered into the database. Oracle allows the creation of five types of integrity constraints which are discussed below.

2.5.1 Database Constraints (Updated 11/30/2007)

Database constraints are of the following five types. A detailed listing of database constraints is provided in **Appendix A (Volume 5)**.

NOT NULL

Prevents NULL values from being entered into the column. These types of constraints are defined on a single column.

CHECK

Checks whether the condition specified in the constraint is satisfied for the values in the column specified.

UNIQUE

Ensures that there are no duplicate values for the column(s) specified. Every value or set of values for the column(s) is unique within the table.

PRIMARY KEY

A column or set of non-null columns that uniquely identifies each row of the table. A table can have only one primary key constraint.

FOREIGN KEY

Establishes a parent-child relationship between tables by using common columns (i.e. a foreign key in one table is a primary key in a related table).

2.5.2 Application Code (Updated 11/30/2007)

Within the application source code, business rules are enforced with program logic. For example, user's roles and privileges are queried to determine whether a request to access or update data is allowed. Detailed information about the business rules enforced on each page is provided in **Appendix D (Volume 6)**.

2.5.3 Database Triggers and Functions (Updated 11/30/2007)

Database triggers are pieces of PL/SQL code that are executed as part of certain database transactions, as defined by the trigger. Database functions and procedures are pieces of PL/SQL code that are executed when called by the application or database. The PL/SQL code is used to enforce some business rules in the database; for example, records that are user-editable have a time stamp indicating the date and time of the last edit to the record. The types of triggers, functions, and procedures used in the EST database are described below. A detailed listing of triggers, functions, and procedures is provided in **Appendix A (Volume 5)**.

Primary Key Triggers

On insert to a table, the associated trigger creates the primary key code from the next value in the stored sequence.

Timestamp Triggers

On insert and/or update, the trigger sets a column to the current date and time.

History Triggers

On update and/or delete, the trigger copies the record from the table to a history table.

Database Functions and Procedures

These are programs within the database that are used to perform specific actions.

Miscellaneous Triggers

A number of other triggers are used in the database to perform specific actions. In pseudo-code, these triggers can be described as *On EVENT to* TABLE *{COLUMN} {WHEN CLAUSE}, fire TRIGGER to perform ACTION*.

2.6 Data Migration and Transformation (Updated 4/30/2008)

Data migration and transformation includes plans for taking existing data and transforming/migrating that data into the correct values/format according to the latest design of the application. If information exists in electronic format which will be converted for use by the new or updated application design, information about data conversion/archiving will also be included. A cross-reference may be used to indicate how the existing data will be used to update the new data elements. If the latest application version is replacing an existing one which will be disabled, any data which will not be retained or archived, and therefore not easily accessed, should be explicitly listed.

The EST does not replace an existing application. When new versions of the application are released, any data transformations needed are performed programmatically at that time. Changes to the production data model are documented in **Attachment A** of this Design Document. This documentation includes the programs used to transform the database during the GUI conversion conducted in August 2005, for example.

The EST application relies heavily on the use of GIS data sources from multiple agencies. The data collection and processing procedures for these data sets are described in the EST Requirements Document, section 7.4 Data Collection Methodologies.

Chapter 3 Interface Design

3.1 Type of Interface to be Implemented

The Environmental Screening Tool is an Internet-accessible application that provides tools to input and update information about transportation projects, perform standardized analyses, gather and report comments about potential project effects, and provide information to the public. The Environmental Screening Tool brings together information about a project and provides analytical and visualization tools that help synthesize and communicate that information. It is used throughout the ETDM process to:

- Integrate data from multiple sources into an easy to use, standard format
- Analyze the effects of proposed projects on the human and natural environment
- Communicate information effectively among Environmental Technical Advisory Team (ETAT) representatives and to the public
- Store and report results of the ETAT review effectively and efficiently
- Maintain project records, including commitments and responses, throughout the project life cycle

The Environmental Screening Tool integrates Internet mapping technology, relational database management systems and geographic information systems (GIS) using software described in Section 2.1.

3.2 Characteristics of Individual Data Elements

Data elements are described in the Data Dictionary found in **Appendix A (Volume 5)**.

3.3 Characteristics of Data Element Assemblies

Data element assemblies include the reports, forms, and programs which make up the EST application. The characteristics of each program are documented within the code itself. The objects accessed by these programs are documented in **Appendix D (Volume 6)**.

3.4 Characteristics of Communication Methods

The EST uses standard Internet communication methods supported by Apache and Tomcat (see section 2.1).

3.5 Characteristics of Protocols

The EST is a secure web site that uses standard Internet protocols supported by Apache and Tomcat (see section 2.1).

3.6 Other Characteristics

There are no other characteristics which need to be described.

Table of Contents

This attachment to the Design Document for the Environmental Screening Tool (EST) includes documentation of Data Migration/Transformation procedures used for the EST.

A.1 Graphical User Interface (GUI) Conversion, August 2005

The following programs were used to transform the database during the GUI conversion conducted in August 2005:

```
*******************************
IMPORT DEV2 TABLES SANS RECORDS
*******************************
imp <admin>/<pass>3r@etdmsde.fla-etat.org indexfile=2005081901etdmdev2.sql.0
file=etdmsde_etdmdev2_20050819.dmp constraints=N indexes=Y grants=N fromuser=etdmdev2
touser=etdmv3stage

    sed 's/ LOGGING/ NOLOGGING/g'  2005081901etdmdev2.sql.0 > 2005081901etdmdev2.sql.1
    sed 's/ETDMDEV2/ETDMv3stage/g'  2005081901etdmdev2.sql.1 > 2005081901etdmdev2.sql.2
    sed 's/.*rows$//g'             2005081901etdmdev2.sql.2 > 2005081901etdmdev2.sql.3
    sed 's/.*CONNECT .*//g'        2005081901etdmdev2.sql.3 > 2005081901etdmdev2.sql.4
    sed 's/REM  //g'               2005081901etdmdev2.sql.4 > 2005081901etdmdev2.sql

    sqlplus etdmv3stage/guiv3@etdmsde.fla-etat.org @2005081901etdmdev2.sql

******************
ALL RECS FROM DEV2 - approx. 40 mins
******************

imp <admin>/<pass>3r@etdmsde.fla-etat.org ignore=Y indexes=N constraints=N grants=N
file=etdmsde_etdmdev2_20050819.dmp log=etdmsde_etdmdev2_20050819.log fromuser=etdmdev2
touser=etdmv3stage log=etdmdev2_20050819_recs.log;date

**************************************
TRUNCATE MOST TBLS TO REMOVE DEV2 RECS -- 4 mins
**************************************
-- as system
-- select statement returns 179 table names (08/22/05).

 declare

    sqlStmt varchar2(150);

    cursor c is
    select table_name from all_tables where owner = 'ETDMv3stage'
      and (
      ( table_name like '%MILEST%' and table_name not like '%TYPES' )
      or table_name like 'OLD%'
      or table_name like '%TEMP%'
      or table_name like '%HIST%'
      or table_name like '%ANALYSIS%REPORT%'
      or table_name like '%PROJECT%'
      or table_name like '%ORG_USER%'
      or table_name like '%PURPOSE_NEED%'
      or table_name in (select table_name
         from all_tables where owner = 'ETDMV2')
      )
      and table_name not in ('T_ANALYSIS_TYPES','T_ANALYSIS_FIELDS','T_USER_TYPES')
      or table_name in ('T_REVIEW_EXTENSION','T_ANALYSIS_AREAS','AT_REGIONS_PROJECTS','T_USERS');
 begin
```

```
    for rec in c
    loop
      sqlStmt := 'truncate table ETDMv3stage.' || rec.table_name;
      execute immediate sqlStmt;
    end loop;

end;

     [Reserve dev2 records for the following tables]

  AT_ORG_REGIONS
  AT_REGIONCATEGORIES
  AT_REGIONCATEGORIES_REGIONS
  AT_REGIONS_REGIONCATEGORIES
  T_ANALYSIS_BUFFERS
  T_ANALYSIS_FIELDS
  T_ANALYSIS_TYPES
  T_AUTH_ROLES
  T_COMM_POLY_EDIT
  T_COMM_POLY_NEW
  T_COMM_POLY_RESULTS
  T_EFFECTS
  T_ETAT_LIBRARY
  T_ETDM_STATUSES
  T_ISSUE_GROUPS
  T_MILESTONE_TYPES
  T_REGIONCATEGORIES
  T_REGIONS
  T_REGIONTYPES
  T_USER_PREFS
  T_USER_TYPES
  T_WHATS_NEW

  22 rows.

********************************************
delete deprecated tables (e.g., OLD_*, Z_*)
********************************************

done.

********************
disable all triggers
********************

--etdmdev2
connect etdmdev2/
SQL> select trigger_name from user_triggers where status='DISABLED';

  TRIGGER_NAME
  ------------------------------
  PK_ISSUE_TRIG
  PK_PN_REVIEW_TRIG
  PK_PROJECT_REVIEWS_TRIG
  PK_REVIEW_EXTENSION_TRIG
  T_PN_REVIEW_HIST_TRIG_NEW
  T_PROJECT_RESPONSE_HISTORY
  T_PROJECT_SEGS_HISTORY
  T_PROJECT_SEG_COORDS_TERMINII
  T_REVIEWERS_EMAIL_INFO
```

```
   9 rows selected.

--etdmv3stage
connect etdmv3stage/*****
declare
   sqlStmt varchar2(200);
   cursor c is select distinct table_name from user_triggers;

begin
   for rec in c
   loop
      sqlStmt := 'alter table ' || rec.table_name || ' disable all triggers';
      execute immediate sqlStmt;
   end loop;
end;

**********************************************
EXPORT OF SCHEMA CONTAINING ONLY DEV2 OBJECTS
**********************************************
done. finished quickly.

********************************************
INSERT RECS FROM PROD ORGS AND USERS TABLES
********************************************

   insert into etdmv3stage.t_orgs select * from etdmv2.t_orgs order by PK_ORG;
   see solidago:~/transform_users.sql


*****************************************
INSERT RECS FROM PROD PRJ AND ALT TABLES
*****************************************

   insert into etdmv3stage.t_projects select PK_PROJECT,
   PRJNAME,PLAN_ID,FK_ETDM_STAGE,USERNAME,TIME_STAMP,
   FROM_FACILITY,TO_FACILITY,FEDERAL_AID,STATE_FINANCIAL,
   PRJ_TIMESTAMP,PRJ_LOCKED,FEATURE_TYPE,FK_ORG,null,null
   from etdmv2.t_projects
   where pk_project not in (select pk_project
      from etdmv3stage.t_projects)
   order by pk_project;


drop sequence etdmv3stage.t_project_alt_sequence;
create sequence etdmv3stage.t_project_alt_sequence;
alter trigger etdmv3stage.t_project_alt.PK_PROJECT_ALT_TRIG enable ;

   insert into etdmv3stage.t_project_alt select
   PK_PRJ_ALT,FK_PROJECT,COST,MODE_ROADWAY,MODE_TRANSIT,MODE_BIKE,
   MODE_PEDESTRIAN,LENGTH,XMIN,XMAX,YMIN,YMAX,PRJALT_FROM_FACILITY,
   PRJALT_TO_FACILITY,LOCAL_ID,TIME_STAMP,MODE_TOBEDETERMINED,
   MODE_RAIL,FK_STATUS,EDIT_REVIEW_CYCLE,
   CURRENT_REVIEW_START,REVIEW_DURATION,REVIEW_LOCKED,FEATURE_TYPE,
   USERNAME,FK_ALT_TYPE,FIHS,RDWYID,BMP,EMP,NEW_ANALYSIS_RUN,ORIG_STATUS,
   null,null,0,null from etdmv2.t_project_alt
   where (fk_project,pk_prj_alt) not in (select fk_project,pk_prj_alt
      from etdmv3stage.t_project_alt)
   order by fk_project,pk_prj_alt;

******************************
ENABLE ALL TEMP_POP% TRIGGERS
******************************
```

```
declare
   cursor c is
      select 'alter trigger etdmv3stage.' || trigger_name ||
      ' enable' as sqlstmt
      from all_triggers where  owner = 'ETDMv3stage'
      and trigger_name like 'TEMP_POP%';
begin
   for rec in c
   loop
      execute immediate rec.sqlstmt;
   end loop;
end;
/
```

****
COPY RECORDS FROM ETDMSTAGE
****

```
insert into etdmv3stage.t_analysis_issues select * from etdmstage.t_analysis_issues;
```

10min.+?

```
   insert into etdmv3stage.t_analysis_report select
   FK_PROJECT, FK_PRJ_ALT,0,FK_ANALYSIS_TYPE, fk_buffer_distance, SORT_ID,
   ROUTINE_START_DATE, dk_report_date, RECORD_TYPE, FIELD1,FIELD2,FIELD3,FIELD4,
   FIELD5,FIELD6,FIELD7,FIELD8,FIELD9,FIELD10,FIELD11,FIELD12,FIELD13,FIELD14,
   TOTAL_NUM_FEATURES,TOTAL_FEAT_ACRES from etdmstage.z_analysis_report_new
   order by dk_report_date,sort_id;
```

15min.

```
   insert into etdmv3stage.t_analysis_report_sce select
   FK_PROJECT, FK_PRJ_ALT,0,FK_ANALYSIS_TYPE, fk_buffer_distance, SORT_ID,
   ROUTINE_START_DATE, dk_report_date, RECORD_TYPE, FIELD1,FIELD2,FIELD3,FIELD4,
   FIELD5,FIELD6,FIELD7,FIELD8,FIELD9,FIELD10,FIELD11,FIELD12,FIELD13,FIELD14,
   TOTAL_NUM_FEATURES,TOTAL_FEAT_ACRES from etdmstage.z_sce_analysis_report
   order by dk_report_date,sort_id;
```

[per lex:]
v- ANALYSIS_REPORT TABLES
v- T_ANALYSIS_ISSUES
v- NO REPORT HISTORY ****

****
Perform 'simple' copies from etdmv2 tables
****

-id tables with no diffs between v2 and etdmv3stage;
-recs from these can be copied easily insert..select *.

```
--ANALYSIS TABLES (~10 min)
begin
   dbms_stats.gather_schema_stats(ownname=>'ETDMv3stage',cascade=>TRUE);
end;

--script to copy recs from etmdv2 to remaining tables.
--where table structure is dissimilar table recs will not be copied.
--quick.
-----------------------------------
set pagesize 0
set linesize 200
set trimspool on
spool deleteme.sql
select 'insert into etdmv3stage.' || table_name ||
' select * from etdmv2.' || table_name || ';'
from all_tables
where owner = 'ETDMv3stage'
and num_rows = 0
/
spool off
set linesize 80
!egrep 'insert|;' deleteme.sql > copy_from_etdmv2_actual.sql
@copy_from_etdmv2_actual.sql
---------------------------

--ANALYZE TABLES AGAIN TO SEE WHAT REMAINS (12 min)
begin
  dbms_stats.gather_schema_stats(ownname=>'ETDMv3stage',cascade=>TRUE);
end;

--ID tables still needing records
SQL>  select table_name from all_tables where owner = 'ETDMv3stage'
      and num_rows = 0
      and table_name not like '%MILEST%'
      and table_name not in (select table_name from all_tables where owner = 'ETDMV2' and num_rows = 0)
      order by table_name;
--66 rows.

************************************************************
COPY RECORDS FROM ETDMV2 66 TABLES RETURNED BY QUERY ABOVE
************************************************************

--T_ANALYSIS_NOTIFICATION
alter trigger etdmv3stage.pk_analysis_notification_trig enable;
drop sequence etdmv3stage.t_analysis_notification_seq;
create sequence etdmv3stage.t_analysis_notification_seq;

insert into etdmv3stage.t_analysis_notification select
FK_PROJECT,TIME_STAMP,FK_PRJ_ALT,NOTIFY,AGENCY,null,null,null
from etdmv2.t_analysis_notification
--where (fk_project,fk_prj_alt) in (select fk_project,pk_prj_alt
--from etdmv3stage.t_project_alt)
order by fk_project,fk_prj_alt,time_stamp;

--T_ANALYSIS_QUEUE
insert into etdmv3stage.t_analysis_queue select
SORTID,Q_PROJECTID,Q_ALTID,null,ANALYSIS_MODE,CPU,QUE_STATUS,MILES,QUE_FLAG,
ANALYSIS_TIME,QUE_FLAG2,ANALYSIS_TIME2,TIME_STAMP,ERROR_MSG,null
from etdmv2.t_analysis_queue;

SQL> insert into etdmv3stage.t_component_status select
  2  PK_COMPONENT_STATUS, NAME, STATUS, HOST, null
  3  from etdmv2.t_component_status;

SQL>de
  1  insert into etdmv3stage.t_dispute_log
  2  select
PK_DISPUTE_LOG,FK_PROJECT,FK_ISSUE,FK_PLAN_ORG,FK_USERID,LOG_ITEM_DESCRIPTION,LOG_ITE
M_DATE,
  3  LOG_ITEM_URL,null,null,null
  4* from etdmv2.t_dispute_log

SQL> insert into etdmv3stage.t_dot_districts select * from etdmdev2.t_dot_districts;

SQL> 1  insert into etdmv3stage.t_effects_summary
```

```
  2   select
FK_ISSUE,FK_PROJECT,FK_ETDM_STAGE,FK_USERID,FK_SUMMARY_EFFECT,SUMMARY_TEXT,TECH_ST
UDIES,STATUS_FINAL,FK_PRJ_ALT
  3* ,FK_CALC_EFFECT,SUMMARY_REPORT_ID,PK_EFFECTS_SUMMARY,TIME_STAMP,nullSQL> SQL>
SQL>  4 from etdmv2.t_effects_summary;

SQL> insert into etdmv3stage.T_EFFECTS_SUMMARY_HISTORY (
  2
FK_ISSUE,FK_PROJECT,FK_ETDM_STAGE,FK_USERID,FK_SUMMARY_EFFECT,SUMMARY_TEXT,TECH_ST
UDIES,STATUS_FINAL,FK_PRJ_ALT,FK_CALC_EFFECT,SUMMARY_REPORT_ID,FK_EFFECTS_SUMMARY,TI
ME_STAMP)
  3  select * from etdmv2.T_EFFECTS_SUMMARY_HISTORY
  4  /

SQL> insert into etdmv3stage.t_email_history (
  2  RECIPIENTS,SUBJECT,BODY,TIMESTAMP,SENDER,ATTACHMENTS,PK_EMAIL_HISTORY
  3  )
  4  select * from etdmv2.t_email_history
  5  ;

SQL> drop sequence etdmv3stage.t_enviro_determ_seq;
SQL> create sequence etdmv3stage.t_enviro_determ_seq;
SQL> alter trigger etdmv3stage.t_enviro_determination_pk enable;
SQL> insert into etdmv3stage.t_user_roles select * from etdmdev2.t_user_roles where
  2  fk_user_type = 'FDOT ETDM Coordinator Primary';
SQL> 1 insert into etdmv3stage.T_ENVIRO_DETERMINATION (
  2
FK_PROJECT,COE,USCG,FWS,EPA,FK_ACTION,FK_OUTREACH,SECTION_4F,SECTION_106,ESA,PUB_HEAR
ING_DATE,FK_TRANSCRIPT,FK_CERT_OPPORTUNITY,FK_LEAD_FED_ORG,FHWA,FRA,FTA,NMFS,NONE)
  3  select * from etdmv2.T_ENVIRO_DETERMINATION;

SQL> insert into etdmv3stage.t_etat_acronyms
  2  select * from etdmdev2.t_etat_acronyms;

SQL> insert into etdmv3stage.t_etat_glossary
  2  select * from etdmdev2.t_etat_glossary;

SQL> insert into etdmv3stage.t_funding_source_lookup (PK_FUNDING_SOURCE)
  2  select * from etdmv2.t_funding_source_lookup;

SQL> insert into etdmv3stage.t_issues
  2  select * from etdmdev2.t_issues;

SQL> insert into etdmv3stage.t_permits
  2  (PK_PERMIT_ID,ITEM_TYPE,ITEM_TEXT)
  3  select * from etdmv2.t_permits;

truncate table etdmv3stage.T_PURPOSE_NEED_REVIEW_NEW;
SQL> alter trigger etdmv3stage.PK_PN_REVIEW_TRIG enable;
SQL> drop sequence etdmv3stage.PK_PN_REVIEW_SEQ;
SQL> create sequence etdmv3stage.PK_PN_REVIEW_SEQ;

--FROM TRANSFORM.sql
/* Populate new purpose/need review table
 *
 */
INSERT INTO etdmv3stage.T_PURPOSE_NEED_REVIEW_NEW (FK_PROJECT, COMMENTS,
    APPROVED_TEXT, PN_REVIEW_DATE, FK_ORG_USER)
SELECT FK_PROJECT, COMMENTS, APPROVED_TEXT, PN_REVIEW_DATE, PK_ORG_USER
FROM etdmv3stage.T_PURPOSE_NEED_REVIEW PN, etdmv3stage.T_USERS U, etdmv3stage.T_ORG_USERS
OU
```

```
WHERE PN.fk_userid = U.user_id
AND U.pk_user = OU.fk_user
AND PN.fk_review_org = OU.fk_org
AND PN.approved_text IS NOT NULL;

/* Disable pk trigger on pn review table
 * Not strictly necessary - but if trigger is enabled, table will need TEMP_ID NUMBER(20) NULL added
 */
ALTER TRIGGER "ETDMDEV2"."PK_PN_REVIEW_TRIG" DISABLE

SQL> alter trigger ETDMv3stage.PK_PRJ_CONSISTENCY_TRIG enable;
SQL> drop sequence ETDMv3stage.T_PRJ_CONSISTENCY_SEQ;
SQL> create sequence ETDMv3stage.T_PRJ_CONSISTENCY_SEQ;
SQL> insert into ETDMv3stage.t_prj_consistency
  2  ( FK_PROJECT,FK_PLAN_REG_NAME,CONSISTENT)
  3  select * from etdmv2.t_prj_consistency

SQL> 1 insert into etdmv3stage.t_projects_deleted (
  2
PK_PROJECT,PRJNAME,PLAN_ID,FK_ETDM_STAGE,USERNAME,TIME_STAMP,FROM_FACILITY,TO_FACILITY,FK_ORG
  3  )
  4  select * from etdmv2.t_projects_deleted;

SQL> l
  1   insert into etdmv3stage.T_PROJECT_ALT_HISTORY
  2    select
  3    PK_PRJ_ALT,FK_PROJECT,COST,MODE_ROADWAY,MODE_TRANSIT,MODE_BIKE,MODE_PEDESTRIAN,
  4    LENGTH,XMIN,XMAX,YMIN,YMAX,PRJALT_FROM_FACILITY,PRJALT_TO_FACILITY,LOCAL_ID,
  5
TIME_STAMP,MODE_TOBEDETERMINED,MODE_RAIL,FK_STATUS,EDIT_REVIEW_CYCLE,CURRENT_REVIEW_START,
  6
REVIEW_DURATION,REVIEW_LOCKED,FEATURE_TYPE,USERNAME,FK_ALT_TYPE,FIHS,RDWYID,BMP,EMP,
NEW_ANALYSIS_RUN
  7  ,0,0,null
  8    from etdmv2.T_PROJECT_ALT_HISTORY;

SQL> drop sequence etdmv3stage.T_PROJECT_CLOBS_SEQ
SQL> create sequence etdmv3stage.T_PROJECT_CLOBS_SEQ;
SQL> alter trigger etdmv3stage.PK_PROJECT_CLOBS enable;
SQL> insert into etdmv3stage.t_project_clobs (
  2
FK_PROJECT,PK_PRJCLBHISTORY_DATE,PURPOSE_NEED,PROJ_PUB_CMT_SMRY,DESCRIPTION_SMRY,DESIRED_FEATURES_CLC)
  3  select * from etdmv2.t_project_clobs;

SQL> 1  insert into etdmv3stage.t_project_clobs_history (
  2
FK_PROJECT,PK_PRJCLBHISTORY_DATE,PURPOSE_NEED,PROJ_PUB_CMT_SMRY,DESCRIPTION_SMRY,DESIRED_FEATURES_CLC)
  3  select * from etdmv2.t_project_clobs_history;

SQL> insert into etdmv3stage.t_project_commitment (
  2
PK_PROJECT_COMMITMENT,FK_PROJECT,COMMIT_DATE,COMMIT_TEXT,COMMIT_ORG,FK_USERID,COMMIT_TEXT_CLOB,TIME_STAMP)
  3  select * from etdmv2.t_project_commitment;
SQL> alter trigger etdmv3stage.PK_PROJECT_CONTACT_TRIG enable;
SQL> drop sequence etdmv3stage.T_PROJECT_CONTACT_SEQ;
SQL> create sequence etdmv3stage.T_PROJECT_CONTACT_SEQ
SQL> insert into etdmv3stage.t_project_contact (
  2  FK_PROJECT,NAME,ADDRESS,EMAIL,PHONE )
  3  select * from etdmv2.t_project_contact;

SQL> insert into etdmv3stage.t_project_docs
  2  ( FK_PROJECT,PATH_URL,DOC_DESCRIPTION,DOC_DATE,PK_PROJECT_DOC,DOC_TYPE )
  3  select * from etdmv2.t_project_docs;

SQL> alter trigger etdmv3stage.PK_PROJECT_EXTENSION_TRIG enable;
SQL> drop sequence etdmv3stage.T_PROJECT_EXTENSION_SEQ;
SQL> create sequence etdmv3stage.T_PROJECT_EXTENSION_SEQ
SQL> insert into etdmv3stage.t_project_extension (
  2  FK_PROJECT,FK_PRJ_ALT,COMMENTS)
  3  select FK_PROJECT,FK_PRJ_ALT,COMMENT_TEXT from etdmv2.t_project_extension;

insert into etdmv3stage.t_projects_history
select
FK_PROJECT_HISTORY,PRJHIST_PRJNAME,PRJHIST_PLAN_ID,PRJHIST_ETDM_STAGE,PRJHIST_USERNA
ME,
PK_PRJHISTORY_DATE,PRJHIST_FROM_FACILITY,PRJHIST_TO_FACILITY,PRJHIST_FEDERAL_AID,
PRJHIST_STATE_FINANCIAL,PRJ_TIMESTAMP,PRJ_LOCKED,FEATURE_TYPE,FK_ORG,null,0
from etdmv2.t_project_history;

SQL> insert into etdmv3stage.t_project_reviews ( FK_ISSUE,
  2
FK_REVIEW_ORG,FK_PROJECT,FK_ETDM_STAGE,AGENCY_INVOLVEMENT,REVIEW_TEXT,FK_USERID,RE
VIEW_DATE,
  3  FK_REVIEW_STATUS,PK_REVIEW,FK_EFFECT,FK_PRJ_ALT )
  4  select * from etdmv2.t_project_reviews;

SQL> insert into etdmv3stage.t_project_response (PK_RESPONSE,RESPONSE_DATE,RESPONSE_TEXT,
  2  FK_REVIEW,RESPONSE_ORG,LAST_UPDATED,FK_ETDM_STAGE,PLAN_YEAR)
  3  select * from etdmv2.t_project_response;

SQL> insert into etdmv3stage.t_project_reviews_draft (
  2
FK_ISSUE,FK_REVIEW_ORG,FK_PROJECT,FK_ETDM_STAGE,AGENCY_INVOLVEMENT,REVIEW_TEXT,FK_U
SERID,
  3  REVIEW_DATE,FK_REVIEW_STATUS,PK_REVIEW,FK_EFFECT,FK_PRJ_ALT,COPY_TO_PRODUCTION)
  4  select * from etdmv2.t_project_reviews_draft;

SQL> insert into etdmv3stage.t_project_reviews_history
(FK_ISSUE,FK_REVIEW_ORG,FK_PROJECT,FK_ETDM_STAGE,
  2
AGENCY_INVOLVEMENT,REVIEW_TEXT,FK_USERID,REVIEW_DATE,FK_REVIEW_STATUS,PK_REVIEW,FK_
EFFECT,FK_PRJ_ALT)
  3  select * from etdmv2.t_project_reviews_history;

SQL> insert into etdmv3stage.t_project_segments
(PK_SEGMENT,FK_PROJECT,FK_PRJ_ALT,DATE_STAMP,SEGMENT_ID,CURR_YEAR,CURR_AADT,CURR_LA
NES,CURR_CONFIG,INTRM_YEAR,INTRM_AADT,INTRM_LANES,INTRM_CONFIG,PLAN_YEAR,COST_AADT,C
OST_LANES,
  2
COST_CONFIG,NEED_AADT,NEED_LANES,NEED_CONFIG,STREET_NAME,FROM_STREET,TO_STREET,LEN
GTH,FK_JURISDICTION,
  3
URB_SERVICE,TCEA,FK_ROADWAY_FCLASS,BASEMAP_SEGMENT_ID,RDWYID,BEGPT,ENDPT,USERNAME,
SEGMENT_ORDER,
  4  LOCAL_ID,SEG_LOCAL_ID,SOURCE,EDIT_REVIEW_CYCLE)
  5  select * from etdmv2.t_project_segments;
```

```
SQL> insert into etdmv3stage.t_project_segment_coords
(FK_SEGMENT,FK_PROJECT,FK_PRJ_ALT,SORT_ORDER_ID,X,Y,TERMINII)
  2  select * from etdmv2.t_project_segment_coords;

SQL> 1  insert into etdmv3stage.t_project_segments_history
  2  (
fk_Segment,FK_PROJECT,FK_PRJ_ALT,DATE_STAMP,SEGMENT_ID,CURR_YEAR,CURR_AADT,CURR_LANE
S,CURR_CONFIG,INTRM_YEAR,INTRM_AADT,INTRM_LANES,INTRM_CONFIG,PLAN_YEAR,COST_AADT,COS
T_LANES,COST_CONFIG,NEED_AADT,NEED_LANES,NEED_CONFIG,STREET_NAME,
  3
FROM_STREET,TO_STREET,LENGTH,FK_JURISDICTION,URB_SERVICE,TCEA,FK_ROADWAY_FCLASS,BAS
EMAP_SEGMENT_ID,RDWYID,BEGPT,ENDPT,USERNAME,EDIT_REVIEW_CYCLE)
  4  select * from etdmv2.t_project_seg_history;

SQL> insert into etdmv3stage.t_req_permits
(FK_REVIEW,FK_PERMIT_ID,FK_PROJECT,FK_USERID,TIMESTAMP,PK_REQ_PERMIT,PERMIT_CONDITIONS
) select * from etdmv2.t_req_permits;

SQL>  insert into etdmv3stage.t_req_permits_history
(FK_REVIEW,FK_PERMIT_ID,FK_PROJECT,FK_USERID,TIMESTAMP,fK_REQ_PERMIT,PERMIT_CONDITIONS)
select * from etdmv2.t_req_permits_history;

SQL>  insert into etdmv3stage.t_req_tech_studies (
   FK_REVIEW,FK_TECHID,FK_PROJECT,FK_USERID,TIMESTAMP,
   PK_REQ_TECH_STUDY,TECH_STUDY_CONDITIONS)
   select * from etdmv2.t_req_tech_studies;

SQL>  insert into etdmv3stage.t_req_tech_studies_history (
   FK_REVIEW,FK_TECHID,FK_PROJECT,FK_USERID,TIMESTAMP,
   FK_REQ_TECH_STUDY,TECH_STUDY_CONDITIONS)
   select * from etdmv2.t_req_tech_studies_history;

SQL> l
  1*  insert into etdmv3stage.t_segments_modified (FK_SEGMENT,FK_PRJ_ALT,FK_PROJECT,PK_MOD_DATE)
select * from etdmv2.t_segments_modified
SQL> a  where (fk_project, fk_prj_alt) in (select fk_project,pk_prj_alt
from etdmv3stage.t_project_alt);

SQL> 1 insert into etdmv3stage.t_segment_funding
(FK_SEGMENT,FK_PROJECT,FK_PRJ_ALT,FK_FUNDING_SOURCE,AMOUNT) select * from
etdmv2.t_segment_funding where (fk_project, fk_prj_alt) in (select fk_project,pk_prj_alt
  2* from etdmv3stage.t_project_alt)SQL> ;

SQL> insert into etdmv3stage.t_signatures
(FK_PROJECT,FK_USER_ID,REVIEW_STATUS,DETERM_DATE,EXPLANATION,TIME_STAMP) select * from
etdmv2.t_signatures;

SQL> insert into etdmv3stage.t_signatures_history
(FK_PROJECT,FK_USER_ID,REVIEW_STATUS,DETERM_DATE,EXPLANATION,TIME_STAMP) select * from
etdmv2.t_signatures_history;

SQL> insert into etdmv3stage.t_tech_studies (PK_TECHID,ITEM_TYPE,ITEM_TEXT) select * from
etdmv2.t_tech_studies;

SQL> insert into etdmv3stage.t_tran_plan
(FK_ROAD_SYSTEM,PLAN_YEAR,SYSTEM_NEEDS,SYSTEM_PUB_CMT_SMRY,SUMMARY_SACIE_FILE,FK_P
LAN_ORG,DESCRIPT,TIME_STAMP,PK_PLAN,FK_ORG)
select * from etdmv2.t_tran_plan;

SQL> insert into etdmv3stage.t_tran_plan_history (
  2  FK_ROAD_SYSTEM,PLAN_YEAR,
  3
SYSTEM_NEEDS,SYSTEM_PUB_CMT_SMRY,SUMMARY_SACIE_FILE,DESCRIPT,TIME_STAMP,FK_PLAN,FK_
ORG)
  4  select * from etdmv2.t_tran_plan_history;

-- Add recs for new org_users (that were created by temp_pop triggers)
INSERT INTO T_ORG_USER_AUTHORITY (FK_ORG_USER, FK_AUTH_ROLE, START_DATE)
SELECT PK_ORG_USER, PK_AUTH_ROLE, sysdate
FROM T_ORG_USERS OU, T_USERS U, T_USER_ROLES UR, T_AUTH_ROLES AR
WHERE OU.FK_USER = U.PK_USER
AND U.USER_ID = UR.FK_USERID
AND UR.FK_USER_TYPE = AR.NAME  2    3    4    5    6
and (pk_org_user,pk_auth_role) not in (select fk_org_user,fk_auth_role from T_ORG_USER_AUTHORITY);

alter trigger PK_REVIEW_EXTENSION_TRIG enable;
truncate table t_review_extension;
drop sequence  T_REVIEW_EXTENSION_SEQ;
create sequence T_REVIEW_EXTENSION_SEQ;

INSERT INTO T_REVIEW_EXTENSION (FK_PROJECT_ALT, START_DATE, END_DATE, EXPIRED,
FK_ORG_USER_GRANTED_TO)
SELECT PK_PROJECT_ALT, START_DATE, END_DATE, EXPIRED, PK_ORG_USER
FROM T_PROJECT_ALT alt, AT_PROJECT_EXTENSION ext, T_USERS u, T_ORG_USERS ou, t_projects p
WHERE alt.FK_PROJECT = ext.FK_PROJECT
and p.pk_project = alt.fk_project
and p.fk_org = ou.fk_org
AND ext.FK_USERID = u.USER_ID
AND ou.FK_USER = u.PK_USER;

-- add not null constraint to t_review_extension for fk_org_user_granted_to (no validate)

---------------------------------------
--ANALYZE TABLES (~5 min)
begin
   dbms_stats.gather_schema_stats(ownname=>'ETDMv3stage',cascade=>TRUE);
end;

v-populate [follow milestones2.txt] [t_analysis_report took 30 min.]

v-analyze tables with 0 records.  troubleshoot any id'ed problems.

-sync_* script, begin with adding constraints?  then proceed to end.

   ### 3. Add Constraints
      ### 3.1 Add Check Constraints
      sqlplus $admin/$pass@etdmsde.fla-etat.org @addChecks $fruser $touser $pass addchecks.log

   ### 4. Import all other constraints
      ### 4.1 Generate SQL to create constraints

      imp <admin>/<pass>3r indexfile=2005081901etdmdev2_2.sql.0 indexes=N constraints=Y grants=N
file=etdmsde_etdmdev2_20050819.dmp fromuser=etdmdev2 touser=etdmv3stage

      ### 4.2 Edit SQL before executing
      sed 's/ LOGGING/ NOLOGGING/g'  2005081901etdmdev2_2.sql.0 > 2005081901etdmdev2_2.sql.1
      sed 's/ETDMDEV2/ETDMv3stage/g'  2005081901etdmdev2_2.sql.1 > 2005081901etdmdev2_2.sql.2
      sed 's/.*rows$//g'            2005081901etdmdev2_2.sql.2 > 2005081901etdmdev2_2.sql.3
      sed 's/.*CONNECT .*//g'       2005081901etdmdev2_2.sql.3 > 2005081901etdmdev2_2.sql.4
      sed 's/REM  //g'              2005081901etdmdev2_2.sql.4 > 2005081901etdmdev2_2.sql.5
      sed 's/CREATE TABLE.*//g'      2005081901etdmdev2_2.sql.5 > 2005081901etdmdev2_2.sql
      ### 4.3 Execute SQL to create constraints
      sqlplus etdmv3stage/**** @2005081901etdmdev2_2.sql 2005081901etdmdev2_2.log
```

v-update sequences
   * pk_objects.txt
   * tables: pkmax, pk_objects

v-fix invalid views,triggers,procedures, etc.

v-create like triggers 'estdata' triggers. (imp rows=N ignore=Y).

v-enable triggers
v-updated JAVA apis, changed T_INPUT_* to T_PROJECT, changed pass and username, added FK_PROJECT_ALT

```
   ### 5. Update JAVA APIs
   dropjava -u $touser/$v PointLinePoly
   dropjava -u $touser/$v PointPolygon
   loadjava -u $touser/$v $JDIR/PointLinePoly.class
   loadjava -u $touser/$v $JDIR/PointPolygon.class
```

v-grant some permissions

```
connect <admin>/<pass>
grant select on estdatav3****.s_segments to etdmv3*;
grant update on estdatav3****.s_segments to etdmv3*;
grant select on estdatav3****.s_terminii to etdmv3*;
grant select on estdatav3****.s_points to etdmv3*;
grant select on estdatav3****.s_polygons to etdmv3*;
```

v-check_synonyms. correct schema referenced? permissions in place?

v-make copy of etdmv3stage.

```
exp <admin>/<pass>3r owner=etdmv3stage file=etdmv3stage_20050906.dmp log=etdmv3stage_20050906.log consistent=Y
```

v-capture baseline of etdmv3stage (09072005)

v-exp estdata triggers and constraints

```
exp estdatav3stage/*** file=estdatav3stage_20050907.dmp log=estdatav3stage_20050907.log consistent=Y rows=N triggers=Y constraints=Y tables=\(S_ANALYSIS_AREAS,S_COMM_POLY,S_COMM_PT,S_POINTS,S_POLYGONS,S_SCE_ANALYSIS_AREAS,S_SEGMENTS,S_SEGMENTS_MODIFICATIONS,S_SEGMENTS_REVIEWED,S_TERMINII\)
```

```
imp <admin>/<pass> fromuser=estdatav3stage touser=estdatav3stage file=estdatav3stage_20050907.dmp log=deleteme.log rows=N ignore=Y
```

create synonym for t_project_alt, and t_analysis_areas under estdata* schema
grant select & update on t_project_alt & t_analysis_areas to estdata* schema
compile all invalid triggers
delete invalid spcol_del* triggers where tied to estdatav3stage schema.

****
CREATE MILESTONE RECORDS
****

```
connect etdmv3test/guiv3

/* Enable milestone PK triggers */

SQL> declare
  2     sqlStmt varchar2(100);
  3     cursor c is select trigger_name from user_triggers
  4        where trigger_name like '%PK%' and table_name like '%MILEST%';
  5  begin
  6  for rec in c
```

```
loop
   sqlStmt := 'alter trigger ' || rec.trigger_name || ' enable';
   execute immediate sqlStmt;
end loop;
end;
/

truncate table t_milestones;
drop sequence T_MILESTONE_SEQ;
create sequence T_MILESTONE_SEQ;
******************************** ETAT_REVIEW_START ****************************************************
/ * combine the following two select statements... */

declare
cursor c is select distinct pk_project_alt, fk_project, pk_prj_alt, current_review_start
   from t_project_alt
   where current_review_start is not null
   and fk_status = 'ETAT Review';
begin
for rec in c
loop
   insert into t_milestones_retro
   (fk_milestone_type, time_stamp, fk_project_alt, fk_project, alt_id)
   values (1,rec.current_review_start, rec.pk_project_alt,rec.fk_project,rec.pk_prj_alt);
end loop;
end;
/

declare
cursor c is select distinct fk_project_alt, fk_project, pk_prj_alt, current_review_start
   from t_project_alt_history
   where current_review_start is not null
   and fk_status = 'ETAT Review';
begin
for rec in c
loop
   insert into t_milestones_retro
   (fk_milestone_type, time_stamp, fk_project_alt, fk_project, alt_id)
   values (1,rec.current_review_start, rec.fk_project_alt,rec.fk_project,rec.pk_prj_alt);
end loop;
end;
/

******************************** ETAT_REVIEW_END **************************************

declare
last_status varchar2(50) := 'ETAT Review';
cursor c is
  select pk_project_alt, fk_project, pk_prj_alt, time_stamp, fk_status
  from t_project_alt
  union
  select fk_project_alt as pk_project_alt, fk_project, pk_prj_alt, time_stamp, fk_status
  from t_project_alt_history
  order by pk_project_alt, time_stamp;
begin
for rec in c
loop
   if (rec.fk_status = 'ETAT Review Complete' and last_status = 'ETAT Review') then
      insert into t_milestones_retro
         (fk_milestone_type, time_stamp, fk_project_alt, fk_project, alt_id)
         values (2,rec.time_stamp,rec.pk_project_alt,
      rec.fk_project,rec.pk_prj_alt);
   end if;
   last_status := rec.fk_status;
end loop;
end;
/

***************** SUMMARY FINALIZED (Mike's SQL, added fk_project, pk_prj_alt) ***************************

insert into t_milestones_retro
   (FK_MILESTONE_TYPE, TIME_STAMP, FK_PROJECT_ALT, fk_project, alt_id)
select 3, sum_rpt.finalize_date, alt.pk_project_alt, alt.fk_project, alt.pk_prj_alt
from t_summary_report sum_rpt,
(select fk_project, fk_prj_alt, summary_report_id
   from t_effects_summary
   group by fk_project, fk_prj_alt, summary_report_id) eff_sum,
t_project_alt alt
where sum_rpt.pk_summary_report = eff_sum.summary_report_id
and eff_sum.fk_prj_alt = alt.pk_prj_alt
and eff_sum.fk_project = alt.fk_project
and sum_rpt.finalize_date is not null;

******** copy from t_milestones_retro (temp table) to t_milestones (permanent table) *****

truncate table t_milestones;
drop sequence T_MILESTONE_SEQ;
create sequence T_MILESTONE_SEQ;

insert into t_milestones select * from t_milestones_retro order by time_stamp;

************************** QA/QC needed on MILESTONES *********************************

-- update t_milestones set qaqc = null;

declare
   last_type number := 1;
   last_alt number := 0;
   cursor c is select pk_milestone, fk_project_alt, fk_milestone_type from t_milestones
      order by fk_project_alt, time_stamp;
begin
for rec in c
loop
   if ((last_alt = rec.fk_project_alt and rec.fk_milestone_type-last_type != 1)
      or
      (last_alt != rec.fk_project_alt and rec.fk_milestone_type != 1)) then
      update t_milestones set qaqc = 'Y' where pk_milestone = rec.pk_milestone
      and fk_milestone_type != 1;
   end if;
   last_type := rec.fk_milestone_type;
   last_alt := rec.fk_project_alt;
end loop;
end;
/
********************** ID test projects ****************************

update t_milestones set test_project = 'Y' where fk_project in (
   select pk_project from v_test_projects );

*********************** View results ******************
alter session set nls_date_format = 'dd-mon-yy hh24:mi:ss';

select fk_project_alt,fk_milestone_type,qaqc,time_stamp from t_milestones
```

```
order by fk_project_alt,time_stamp;

select fk_project_alt,fk_milestone_type,qaqc,time_stamp from t_milestones
where fk_project_alt not in (
   select fk_project_alt from t_milestones where qaqc = 'Y'
)
order by fk_project_alt,time_stamp;

********************** Populate T_PROJECT_MILESTONES ********************************
/* There should be one t_project_milestones record per t_milestones record.
Likewise, there should be one t_project_alt_milestones record per t_milestones record.
After initial population of the alt ms and prj ms tables, some records were missing due
a lack of records in the parent and history tables with a time_stamp <= ms timestamp.
I think the lack of matching records was caused by the specs on timestamp triggers.  They
were set to fire for all fields except the time_stamp field.  They have been modified
to fire on update of *any* field.  Adding missing records--note timestamps will be more
recent than ms timestamp.*/

truncate table t_project_milestones;
drop sequence  T_PROJECT_MILESTONES_SEQ;
create sequence  T_PROJECT_MILESTONES_SEQ;

declare
   recExists number(1);
   cursor c is select pk_milestone, fk_project, time_stamp from t_milestones
      --where fk_project_alt not in (
      --select fk_project_alt from t_milestones where qaqc = 'Y')
      order by time_stamp;
begin
for rec in c
loop
   select count(*) into recExists from v_chrono_projects
      where (pk_project,time_stamp) in (
          select rec.fk_project,max(time_stamp)
          from v_chrono_projects
          where pk_project = rec.fk_project
          and time_stamp <= rec.time_stamp);
   if (recExists > 0) then
    insert into t_project_milestones select rec.pk_milestone,
      PK_PROJECT,
      PRJNAME,
      PLAN_ID,
      FK_ETDM_STAGE,
      nvl(USERNAME,'lc_mccain'),
      TIME_STAMP,
      FROM_FACILITY,
      TO_FACILITY,
      FEDERAL_AID,
      STATE_FINANCIAL,
      PRJ_TIMESTAMP,
      PRJ_LOCKED,
      FEATURE_TYPE,
      FK_ORG,
      TEMP_ID,
      FK_ORG_USER,
      null
      from v_chrono_projects
      where (pk_project,time_stamp) in (
          select rec.fk_project,max(time_stamp)
          from v_chrono_projects
          where pk_project = rec.fk_project
```

```
           and time_stamp <= rec.time_stamp);
   else
    insert into t_project_milestones select rec.pk_milestone,
      PK_PROJECT,
      PRJNAME,
      PLAN_ID,
      FK_ETDM_STAGE,
      nvl(USERNAME,'lc_mccain'),
      TIME_STAMP,
      FROM_FACILITY,
      TO_FACILITY,
      FEDERAL_AID,
      STATE_FINANCIAL,
      PRJ_TIMESTAMP,
      PRJ_LOCKED,
      FEATURE_TYPE,
      FK_ORG,
      TEMP_ID,
      FK_ORG_USER,
      null
      from v_chrono_projects
      where (pk_project,time_stamp) in (
          select rec.fk_project,min(time_stamp)
          from v_chrono_projects
          where pk_project = rec.fk_project);
   end if;
end loop;
end;
/

********************** Populate T_PROJECT_ALT_MILESTONES *****************************

truncate table t_project_alt_milestones;
drop sequence T_PRJ_ALT_MILESTONES_SEQ;
create sequence T_PRJ_ALT_MILESTONES_SEQ;

/* USE the following only if UNQ constraint violated */
--update t_project_alt set time_stamp = time_stamp + 1/86400
--where (pk_project_alt,time_stamp) in (select fk_project_alt,time_stamp
--from t_project_alt_history);

-- there were still issues with null fk_org_users.  so disable fk_org_user constraint,
-- inserted records as shown below.  will fix nulls then re-enable constaint.

declare
   cursor c is select pk_milestone, fk_project_alt, time_stamp
      from t_milestones
      where fk_project_alt not in (
      select fk_project_alt from t_milestones where qaqc = 'Y');
begin
for rec in c
loop
   insert into t_project_alt_milestones select distinct rec.pk_milestone,
      PK_PRJ_ALT,
      FK_PROJECT,
      COST,
      MODE_ROADWAY,
      MODE_TRANSIT,
      MODE_BIKE,
      MODE_PEDESTRIAN,
      LENGTH,
      XMIN,
      XMAX,
      YMIN,
      YMAX,
      PRJALT_FROM_FACILITY,
      PRJALT_TO_FACILITY,
      LOCAL_ID,
      TIME_STAMP,
      MODE_TOBEDETERMINED,
      MODE_RAIL,
      FK_STATUS,
      EDIT_REVIEW_CYCLE,
      CURRENT_REVIEW_START,
      REVIEW_DURATION,
      REVIEW_LOCKED,
      FEATURE_TYPE,
      USERNAME,
      FK_ALT_TYPE,
      FIHS,
      RDWYID,
      BMP,
      EMP,
      NEW_ANALYSIS_RUN,
      ORIG_STATUS,
      PK_PROJECT_ALT,
      TEMP_ID,
      FK_ETDM_STATUS,
      FK_ORG_USER,
      null,
      null
      from v_chrono_alternatives
      where (pk_project_alt,time_stamp) in (
          select rec.fk_project_alt,max(time_stamp)
          from v_chrono_alternatives
          where pk_project_alt = rec.fk_project_alt
          and time_stamp <= rec.time_stamp);
end loop;
end;
/

/* remove prj milestones where no records (non-qaqc) exist in the alt milestones table */

delete from t_project_milestones  where fk_milestone not in (select fk_milestone from t_project_alt_milestones);

********************** Populate T_PROJECT_CLOB_MILESTONES ********************************

declare
   cursor c is select pk_milestone, fk_project, time_stamp from t_milestones
      where pk_milestone in (
      select fk_milestone from t_project_milestones);
begin
for rec in c
loop
   insert into t_project_clob_milestones select rec.pk_milestone,
      FK_PROJECT,
      PK_PRJCLBHISTORY_DATE,
      PURPOSE_NEED,
      PROJ_PUB_CMT_SMRY,
      DESCRIPTION_SMRY,
      DESIRED_FEATURES_CLC,
      TEMP_ID,
    null
    from v_chrono_prj_clobs
    where (fk_project,PK_PRJCLBHISTORY_DATE) in (
      select rec.fk_project,max(PK_PRJCLBHISTORY_DATE)
      from v_chrono_prj_clobs
      where fk_project = rec.fk_project
      and PK_PRJCLBHISTORY_DATE <= rec.time_stamp);
end loop;
end;
/

********************** Populate T_PROJECT_SEGMENT_MILESTONES ********************************

truncate table t_project_segment_milestones;
drop sequence T_PRJ_SEG_MILESTONES_SEQ;
create sequence T_PRJ_SEG_MILESTONES_SEQ;

declare
  cursor c is select pk_milestone, fk_project_alt, time_stamp from t_milestones
    where fk_project_alt not in (
    select fk_project_alt from t_milestones where qaqc = 'Y');
begin
for rec in c
loop
  insert into t_project_segment_milestones select rec.pk_milestone,
    PK_SEGMENT,
    FK_PROJECT,
    FK_PRJ_ALT,
    DATE_STAMP,
    SEGMENT_ID,
    CURR_YEAR,
    CURR_AADT,
    CURR_LANES,
    CURR_CONFIG,
    INTRM_YEAR,
    INTRM_AADT,
    INTRM_LANES,
    INTRM_CONFIG,
    PLAN_YEAR,
    COST_AADT,
    COST_LANES,
    COST_CONFIG,
    NEED_AADT,
    NEED_LANES,
    NEED_CONFIG,
    STREET_NAME,
    FROM_STREET,
    TO_STREET,
    LENGTH,
    FK_JURISDICTION,
    URB_SERVICE,
    TCEA,
    FK_ROADWAY_FCLASS,
    BASEMAP_SEGMENT_ID,
    RDWYID,
    BEGPT,
    ENDPT,
    nvl(USERNAME,'lc_mccain'),
    SEGMENT_ORDER,
    LOCAL_ID,
    SEG_LOCAL_ID,
    SOURCE,
    EDIT_REVIEW_CYCLE,
    FK_PROJECT_ALT,
    TEMP_ID,
    null,
    null
    from v_chrono_segments
    where (fk_project_alt,date_stamp) in (
      select rec.fk_project_alt,max(date_stamp)
      from v_chrono_segments
      where fk_project_alt = rec.fk_project_alt
      and date_stamp <= rec.time_stamp);
end loop;
end;
/

********************** Populate T_SEGMENT_FUNDING_MILESTONES *********************************

cannot populate.
no time_stamp field.

********************** Populate T_SIGNATURE_MILESTONES *********************************
truncate table T_SIGNATURE_MILESTONES;
drop sequence  T_SIGNATURE_MILESTONES_SEQ;
create sequence T_SIGNATURE_MILESTONES_SEQ ;
declare
  cursor c is select pk_milestone, fk_project, time_stamp from t_milestones
    where pk_milestone in (
    select fk_milestone from t_project_milestones);
begin
for rec in c
loop
  insert into t_signature_milestones select rec.pk_milestone,
    FK_PROJECT,
    FK_USER_ID,
    REVIEW_STATUS,
    DETERM_DATE,
    EXPLANATION,
    TIME_STAMP,
    null,
    FK_ORG_USER,
    FK_ENVIRO_DETERMINATION,
    null
    from v_chrono_signatures
    where (fk_project,time_stamp) in (
      select rec.fk_project,max(time_stamp)
      from v_chrono_signatures
      where fk_project = rec.fk_project
      and time_stamp <= rec.time_stamp);
end loop;
end;
/
********************** Populate T_PROJECT_COMMIT_MILESTONES *********************************
truncate table t_project_commit_milestones;
drop sequence T_PRJ_COMMIT_MILESTONES_SEQ;
create sequence T_PRJ_COMMIT_MILESTONES_SEQ;

declare
  cursor c is select pk_milestone, fk_project, time_stamp from t_milestones
    where pk_milestone in (
    select fk_milestone from t_project_milestones);
```

```
begin
for rec in c
loop
   insert into t_project_commit_milestones select rec.pk_milestone,
      PK_PROJECT_COMMITMENT,
      FK_PROJECT,
      COMMIT_DATE,
      COMMIT_TEXT,
      COMMIT_ORG,
      FK_USERID,
      COMMIT_TEXT_CLOB,
      TIME_STAMP,
      FK_ORG_USER,
      TEMP_ID,
      null,
      null
      from v_chrono_commitments
      where (fk_project,time_stamp) in (
          select rec.fk_project,max(time_stamp)
          from v_chrono_commitments
          where fk_project = rec.fk_project
          and time_stamp <= rec.time_stamp);
end loop;
end;
/

********************** Populate T_PROJECT_REVIEW_MILESTONES ********************************

truncate table T_PROJECT_REVIEW_MILESTONES;
drop sequence  T_PRJ_REV_MILESTONES_SEQ;
create sequence  T_PRJ_REV_MILESTONES_SEQ;

declare
   cursor c is select pk_milestone, fk_project, alt_id, time_stamp from t_milestones
      where fk_project_alt not in (
      select fk_project_alt from t_milestones where qaqc = 'Y');
begin
for rec in c
loop
   insert into t_project_review_milestones select rec.pk_milestone,
      FK_ISSUE,
      FK_REVIEW_ORG,
      FK_PROJECT,
      FK_ETDM_STAGE,
      AGENCY_INVOLVEMENT,
      REVIEW_TEXT,
      FK_USERID,
      REVIEW_DATE,
      FK_REVIEW_STATUS,
      PK_REVIEW,
      FK_EFFECT,
      FK_PRJ_ALT,
      FK_ORG_USER,
      FK_EFFECT_ID,
      FK_PROJECT_ALT,
      null,
      null
      from v_chrono_reviews
      where (fk_project,fk_prj_alt,review_date) in (
          select rec.fk_project,rec.alt_id,max(review_date)
          from v_chrono_reviews
```

```
       where fk_project = rec.fk_project
       and review_date <= rec.time_stamp);
end loop;
end;
/

********************** Populate T_PROJECT_RESPONSE_MILESTONES *********************************

/* if needed */
--alter session set nls_date_format = 'dd-mon-yy hh24:mi:ss';
--select pk_review,review_date from  t_project_reviews where (pk_review,review_date) in ( select
pk_review,review_date from t_project_reviews_history);
--update t_project_reviews set review_date = review_date+1/86400 where pk_review =  and review_date = '';

truncate table T_PROJECT_RESPONSE_MILESTONES;
drop sequence T_PRJ_RESPONSE_MILESTONES_SEQ;
create sequence T_PRJ_RESPONSE_MILESTONES_SEQ;

-- 0 records added.

declare
  cursor c is select pk_milestone, r.fk_project, pk_review, time_stamp
    from t_project_review_milestones r, t_milestones t
    where pk_milestone = fk_milestone;
begin
for rec in c
loop
  insert into t_project_response_milestones select
rec.pk_milestone,
    PK_RESPONSE,
    RESPONSE_DATE,
    RESPONSE_TEXT,
    FK_REVIEW,
    RESPONSE_ORG,
    LAST_UPDATED,
    FK_PROJECT,
    FK_ETDM_STAGE,
    PLAN_YEAR,
    null,
    null,
    null
    from v_chrono_responses
    where (fk_project,response_date) in (
       select rec.fk_project,max(response_date)
       from v_chrono_responses
       where fk_review = rec.pk_review
       and fk_project = rec.fk_project
       and response_date <= rec.time_stamp);
end loop;
end;
/

********************** Populate T_ENVIRO_DETERM_MILESTONES *********************************

could not add records.
no timestamp.

********************** Populate T_PN_REVIEW_NEW_MILESTONES *********************************

truncate table T_PN_REVIEW_NEW_MILESTONES;
drop sequence T_PN_REVIEW_MILESTONES_SEQ;
```

```
create sequence T_PN_REVIEW_MILESTONES_SEQ;

declare
  cursor c is select pk_milestone, fk_project, time_stamp from t_milestones
    where pk_milestone in (
    select fk_milestone from t_project_milestones);
begin
for rec in c
loop
  insert into t_pn_review_new_milestones select rec.pk_milestone,
    PK_PN_REVIEW,
    FK_PROJECT,
    COMMENTS,
    APPROVED_TEXT,
    PN_REVIEW_DATE,
    FK_ORG_USER,
    null,
    null
    from v_chrono_pn_review
    where (fk_project,PN_REVIEW_DATE) in (
      select rec.fk_project,max(PN_REVIEW_DATE)
      from v_chrono_responses
      where fk_project = rec.fk_project
      and PN_REVIEW_DATE <= rec.time_stamp);
end loop;
end;
/

********************** Populate T_EFFECTS_SUMMARY_MILESTONES *********************************

truncate table T_EFFECTS_SUMMARY_MILESTONES;
drop sequence T_EFF_SUM_MILESTONES_SEQ;
create sequence T_EFF_SUM_MILESTONES_SEQ;

declare
  cursor c is select pk_milestone, fk_project, alt_id, time_stamp from t_milestones
    where fk_project_alt not in (
    select fk_project_alt from t_milestones where qaqc = 'Y');
begin
for rec in c
loop
  insert into t_effects_summary_milestones select
    rec.pk_milestone,
    FK_ISSUE,
    FK_PROJECT,
    FK_ETDM_STAGE,
    FK_USERID,
    FK_SUMMARY_EFFECT,
    SUMMARY_TEXT,
    TECH_STUDIES,
    STATUS_FINAL,
    FK_PRJ_ALT,
    FK_CALC_EFFECT,
    SUMMARY_REPORT_ID,
    PK_EFFECTS_SUMMARY,
    TIME_STAMP,
    null,
    FK_PROJECT_ALT,
    null
    from v_chrono_effects
    where (fk_project,fk_prj_alt,time_stamp) in (
          select rec.fk_project,rec.alt_id,max(time_stamp)
          from v_chrono_effects
          where fk_project = rec.fk_project
          and time_stamp <= rec.time_stamp);
end loop;
end;
/

********************* Populate T_ANALYSIS_REPORT_MILESTONES *******************************

truncate table t_analysis_report_Milestones;  --[32 MINUTES]

declare
   cursor c is select pk_milestone, fk_project, alt_id, time_stamp from t_milestones
      where fk_project_alt not in (
      select fk_project_alt from t_milestones where qaqc = 'Y');
begin
for rec in c
loop
   insert into T_ANALYSIS_REPORT_milestones select
      rec.pk_milestone,
      FK_PROJECT,
      FK_PRJ_ALT,
      FK_PROJECT_ALT,
      FK_ANALYSIS_TYPE,
      BUFFER_DISTANCE,
      SORT_ID,
      ROUTINE_START_DATE,
      TIME_STAMP,
      RECORD_TYPE,
      FIELD1,
      FIELD2,
      FIELD3,
      FIELD4,
      FIELD5,
      FIELD6,
      FIELD7,
      FIELD8,
      FIELD9,
      FIELD10,
      FIELD11,
      FIELD12,
      FIELD13,
      FIELD14,
      TOTAL_NUM_FEATURES,
      TOTAL_FEAT_ACRES,
      null
      from T_ANALYSIS_REPORT
      where (fk_project,fk_prj_alt,ROUTINE_START_dATE) in (
          select rec.fk_project,rec.alt_id,max(ROUTINE_START_dATE)
          from T_ANALYSIS_REPORT
          where fk_project = rec.fk_project
          and ROUTINE_START_dATE <= rec.time_stamp);
end loop;
end;
/

/*********************
*TRANSFORM_USERS.SQL
*********************
*/
```

```
12:INSERT INTO T_USERS (USER_ID, PASSWORD, FIRST_NAME, MIDDLE_NAME, LAST_NAME)
44:INSERT INTO T_ORG_USERS (FK_ORG,
107:INSERT INTO T_ORG_USER_AUTHORITY (FK_ORG_USER, FK_AUTH_ROLE, START_DATE)
629:INSERT INTO T_PURPOSE_NEED_REVIEW_NEW (FK_PROJECT, COMMENTS, APPROVED_TEXT,
PN_REVIEW_DATE, FK_ORG_USER)
679:INSERT INTO T_ORG_USER_JURISDICTIONS (FK_REGION, FK_ORG_USER_AUTHORITY, ENABLED)
756:INSERT INTO T_REVIEW_EXTENSION (FK_PROJECT_ALT, START_DATE, END_DATE, EXPIRED,
FK_ORG_USER_GRANTED_TO)
838:INSERT INTO T_PROJECT_AUTH_ROLE VALUES (1, 'Project Manager');
893:INSERT INTO T_PROJECT_AUTHORITY (FK_PROJECT, FK_ORG_USER, FK_PROJECT_AUTH_ROLE,
START_DATE, EXPIRED, TIME_STAMP)

/* Enable PK insert trigger for Users table.
 * Also, recreate sequence.
 */
alter trigger "ETDMV3TEST"."PK_USER_TRIG" enable;
drop sequence ETDMV3TEST.T_USERS_SEQ;
create sequence ETDMV3TEST.T_USERS_SEQ;

/* Populate Users table from Reviewers table and commit
 *
 */
truncate table t_users;

insert into etdmv3test.t_reviewers select * from etdmv2.t_reviewers
order by pk_userid;

INSERT INTO etdmv3test.T_USERS (USER_ID, PASSWORD, FIRST_NAME, MIDDLE_NAME, LAST_NAME)
  SELECT PK_USERID, PASSWORD, FIRST_NAME, MIDDLE_NAME, LAST_NAME FROM
etdmv2.T_REVIEWERS;

COMMIT;

/* Data cleaning....assign a name to superuser
 */
UPDATE etdmv3test.t_users
SET first_name = 'Super', middle_name = 'D.', last_name = 'User'
WHERE user_id='superuser';
COMMIT;

/* Enable PK trigger for OrgUsers table and
 * recreate sequence.
 */
alter tRIGGER "ETDMV3TEST"."PK_ORG_USER_TRIG" enable;
drop sequence ETDMV3TEST.T_ORG_USERS_SEQ;
create sequence ETDMV3TEST.T_ORG_USERS_SEQ;

/* Data Cleaning...
 * Fix phone numbers that have extension in phone field instead of phone_ext and commit changes.
 * Prepares data for new field sizes.
 */
UPDATE etdmv3test.t_reviewers SET phone = substr(phone, 1, 12), phone_ext = substr(phone, 14, 12) WHERE
length(phone) > 15;

COMMIT;

/* Populate OrgUsers table from Reviewers table, Users table (for fk_user),
 * and User_Orgs association table (User_Orgs will be deprecated).
 * Commit changes.
 *
 */
```

```
truncate table etdmv3test.t_org_users;

INSERT INTO etdmv3test.T_ORG_USERS (FK_ORG,
    FK_USER,
    READ_ONLY,
    ENABLED,
    DIRECT_PHONE,
    DIRECT_PHONE_EXT,
    OFFICE_PHONE,
    OFFICE_PHONE_EXT,
    SUNCOM,
    CELL_PHONE,
    FAX,
    ADDRESS1,
    ADDRESS2,
    CITY,
    STATE,
    ZIP,
    EMAIL,
    ALT_EMAIL,
    EST_NOTICES,
    ETAT_NOTICES,
    RECEIVER,
    DEVELOPMENT_TEAM)
SELECT FK_ORG,
    PK_USER,
    READ_ONLY,
    DECODE(DISABLED, 'Y', 'N', 'Y'),
    PHONE,
    PHONE_EXT,
    OFFICE_PH,
    OFFICE_PH_EXT,
    SUNCOM,
    CELL,
    FAX,
    ADDRESS1,
    ADDRESS2,
    CITY,
    STATE,
    ZIP,
    EMAIL,
    ALT_EMAIL,
    EMAIL_NOTIFICATION,
    ETAT_NOTIFICATION,
    'N',
    DEVELOPMENT_TEAM
FROM etdmv3test.T_REVIEWERS R, etdmv3test.T_USERS U, etdmv2.T_USER_ORGS O
WHERE R.pk_userid = U.user_id
AND R.pk_userid = O.fk_userid

COMMIT;
/* Enable PK insert trigger for OrgUser Authority table &
 * recreate sequence.
 */
ALTER TRIGGER "ETDMV3TEST"."PK_ORG_USER_AUTHORITY_TRIG" enable;
drop sequence etdmv3test.T_ORG_USER_AUTHORITY_SEQ;
create sequence etdmv3test.T_ORG_USER_AUTHORITY_SEQ;

/* Populate OrgUser Authority table from OrgUsers table, Users table,
 * existing User Roles table and new Auth Roles table (T_User_Roles will be deprecated).
 * Commit changes.
 *
 */
truncate table etdmv3test.t_org_user_authority;

INSERT INTO etdmv3test.T_ORG_USER_AUTHORITY (FK_ORG_USER, FK_AUTH_ROLE, START_DATE)
SELECT PK_ORG_USER, PK_AUTH_ROLE, sysdate
FROM etdmv3test.T_ORG_USERS OU, etdmv3test.T_USERS U, etdmv2.T_USER_ROLES UR,
etdmv3test.T_AUTH_ROLES AR
WHERE OU.FK_USER = U.PK_USER
AND U.USER_ID = UR.FK_USERID
AND UR.FK_USER_TYPE = AR.NAME;

COMMIT;

SQL> select distinct username from t_project_alt where fk_org_user is null;

USERNAME
------------------------------
--16 users
--94 records
--username = '.' for 3 recs

update t_project_alt set username = 'superuser' where username = '.';

update t_project_alt set fk_org_user = null where fk_org_user is null;
--trigger temp_pop_fk_org_user populates fk_org_user :) :)

/* Create pk sequence for t_org_user_jurisdictions
 *
 */
drop SEQUENCE "ETDMV3TEST"."T_ORG_USER_JURISDICTIONS_SEQ";
create SEQUENCE "ETDMV3TEST"."T_ORG_USER_JURISDICTIONS_SEQ";

/* enable pk trigger on org user juris table
 *
 */
alter TRIGGER "ETDMV3TEST"."PK_ORG_USER_JURISDICTION_TRIG" enable;

/* Populate org user juris table - based on at_user_systems (does at_user_systems need to be maintained?)
 *
 */

truncate table etdmv3test.T_ORG_USER_JURISDICTIONS;

insert into etdmv3test.t_road_systems select * from etdmdev2.t_road_systems;

INSERT INTO etdmv3test.T_ORG_USER_JURISDICTIONS (FK_REGION, FK_ORG_USER_AUTHORITY,
ENABLED)
SELECT sys.FK_REGION, PK_ORG_USER_AUTHORITY, 'Y'
FROM etdmdev2.T_ROAD_SYSTEMS sys, etdmv2.AT_USER_SYSTEMS us, etdmv3test.T_USERS u,
etdmv3test.T_ORG_USERS ou,
   etdmv3test.T_ORG_USER_AUTHORITY oua
WHERE sys.PK_ROAD_SYSTEM = us.FK_ROAD_SYSTEM
AND us.FK_USERID = u.USER_ID
AND u.PK_USER = ou.FK_USER
AND ou.PK_ORG_USER = oua.FK_ORG_USER;

COMMIT;

/* Disable pk sequence trigger on org user juris table
 * Not strictly necessary - but if trigger is enabled, table will need TEMP_ID NUMBER(20) NULL added
 */
ALTER TRIGGER "ETDMV3TEST"."PK_ORG_USER_JURISDICTION_TRIG" DISABLE

/* Create Project Authority table
 * - Project authority ties a project authority role to a project.
 * - Allows granting authority by project, instead of by user's regions.
 * - For example, a project manager has specific rights on an assigned project,
 * but not on all projects in the user's region(s).
 * - Project authority is different than org user authority, which ties an org user
 * authority role to an org user and to a set of regions (via org user jurisdiction).
 *
 */

/* Create PK sequence Project Authority table
 *
 */
drop SEQUENCE "ETDMV3TEST"."T_PROJECT_AUTHORITY_SEQ";
CREATE SEQUENCE "ETDMV3TEST"."T_PROJECT_AUTHORITY_SEQ";

/* ENABLE PK on-insert trigger for Project Authority table
 *
 */
alter trigger "ETDMV3TEST"."PK_PROJECT_AUTHORITY_TRIG" enable;
drop SEQUENCE "ETDMV3TEST"."T_PROJECT_AUTHORITY_SEQ";
CREATE SEQUENCE "ETDMV3TEST"."T_PROJECT_AUTHORITY_SEQ";

/* Populate Project Authority table from Project Manager table and commit
 *
 */
truncate table ETDMV3TEST.AT_PROJECT_MANAGER;

insert into ETDMV3TEST.AT_PROJECT_MANAGER select * from ETDMV2.AT_PROJECT_MANAGER;

INSERT INTO etdmv3test.T_PROJECT_AUTHORITY (FK_PROJECT, FK_ORG_USER,
FK_PROJECT_AUTH_ROLE, START_DATE, EXPIRED, TIME_STAMP)
SELECT FK_PROJECT, PK_ORG_USER, 1, sysdate, 'N', sysdate
FROM etdmv3test.AT_PROJECT_MANAGER pm, etdmv3test.T_USERS u, etdmv3test.T_ORG_USERS ou
WHERE pm.FK_USERID = u.USER_ID
AND u.PK_USER = ou.FK_USER;
COMMIT;
  ************ Migrate Data to Milestone Tables *************

v  T_MILESTONES
v  Z_ANALYSIS_REPORT_MILESTONES
v  T_EFFECTS_SUMMARY_MILESTONES
v  T_PROJECT_ALT_MILESTONES
v  T_PROJECT_REVIEW_MILESTONES
v  T_PROJECT_RESPONSE_MILESTONES
v  T_SEGMENT_FUNDING_MILESTONES
*  T_PROJECT_SEGMENT_MILESTONES  (could not add records, no time_stamp)
v  T_PROJECT_CLOB_MILESTONES
*  T_ENVIRO_DETERM_MILESTONES  (could not add records, no time_stamp)
v  T_SIGNATURE_MILESTONES
v  T_PROJECT_COMMIT_MILESTONES
v  T_PROJECT_MILESTONES
v  T_PN_REVIEW_NEW_MILESTONES

***************************** Populate T_PROJECT_ALT_HISTORY.FK_PROJECT_ALT ***********************

SQL> alter table t_project_alt_history add fk_project_alt number(10);
```

```
Table altered.

SQL> l
  1  declare
  2  cursor c is select distinct fk_project, PK_PRJ_alt from t_project_alt_history;
  3  begin
  4  for rec in c
  5  loop
  6  update t_project_alt_history set fk_project_alt = ( select pk_project_alt from
  7  t_project_alt where fk_project = rec.fk_project and pk_prj_alt = rec.pk_prj_alt )
  8  where  fk_project = rec.fk_project and pk_prj_alt = rec.pk_prj_alt;
  9  end loop;
 10* end;
SQL> /

  1* select distinct fk_project, pk_prj_alt, fk_project_alt from t_project_alt_history where fk_project = 2791
SQL> /

FK_PROJECT PK_PRJ_ALT FK_PROJECT_ALT
---------- ---------- --------------
      2791          1             58

******************************* ETAT_REVIEW_START **************************************************
/ * combine the following two select statements... */

declare
cursor c is select distinct pk_project_alt, fk_project, pk_prj_alt, current_review_start from t_project_alt where
current_review_start is not null and fk_status = 'ETAT Review';
begin
for rec in c
loop
insert into t_milestones (fk_milestone_type, time_stamp, fk_project_alt, fk_project, alt_id)
values (1,rec.current_review_start, rec.pk_project_alt,rec.fk_project,rec.fk_prj_alt);
end loop;
end;
/

declare
cursor c is select distinct fk_project_alt, fk_project, pk_prj_alt, current_review_start from t_project_alt_history where
current_review_start is not null and fk_status = 'ETAT Review';
begin
for rec in c
loop
insert into t_milestones (fk_milestone_type, time_stamp, fk_project_alt, fk_project, alt_id)
values (1,rec.current_review_start, rec.fk_project_alt,rec.fk_project,rec.fk_prj_alt);
end loop;
end;
/

******************************* ETAT_REVIEW_END *************************************

declare
last_status varchar2(50) := 'ETAT Review';
cursor c is

  select pk_project_alt, fk_project, pk_prj_alt, time_stamp, fk_status from
  t_project_alt
  union
  select fk_project_alt as pk_project_alt, fk_project, pk_prj_alt, time_stamp, fk_status from
  t_project_alt_history
  order by pk_project_alt, time_stamp;
```

```
begin
for rec in c
loop
   if (rec.fk_status = 'ETAT Review Complete' and last_status = 'ETAT Review') then
      insert into t_milestones values (0,2,rec.time_stamp,rec.pk_project_alt,
         rec.fk_project,rec.pk_prj_alt);
   end if;
   last_status := rec.fk_status;
end loop;
end;
/

**************** SUMMARY FINALIZED (Mike's SQL, added fk_project, pk_prj_alt) *************************

insert into t_milestones (FK_MILESTONE_TYPE, TIME_STAMP, FK_PROJECT_ALT, fk_project, alt_id)
select 3, sum_rpt.finalize_date, alt.pk_project_alt, alt.fk_project, alt.pk_prj_alt
from t_summary_report sum_rpt,
(select fk_project, fk_prj_alt, summary_report_id
from t_effects_summary
group by fk_project, fk_prj_alt, summary_report_id) eff_sum,
t_project_alt alt
where sum_rpt.pk_summary_report = eff_sum.summary_report_id
and eff_sum.fk_prj_alt = alt.pk_prj_alt
and eff_sum.fk_project = alt.fk_project
and sum_rpt.finalize_date is not null

************************ QA/QC on MILESTONES ************************************

declare
   last_type number := 1;
   last_alt number := 0;
   cursor c is select pk_milestone, fk_project_alt, fk_milestone_type from t_milestones
      order by fk_project_alt, time_stamp desc;
begin
for rec in c
loop
   if (last_alt = rec.fk_project_alt and last_type <= rec.fk_milestone_type) then
      update t_milestones set qaqc = 'Y' where pk_milestone = rec.pk_milestone;
   end if;
   last_type := rec.fk_milestone_type;
   last_alt := rec.fk_project_alt;
end loop;
end;

********************* Populate T_PROJECT_MILESTONES *******************************

declare
   cursor c is select pk_milestone, fk_project, time_stamp from t_milestones
      where fk_project_alt not in (
      select fk_project_alt from t_milestones where qaqc = 'Y');
begin
for rec in c
loop
   insert into t_project_milestones select rec.pk_milestone,
      PK_PROJECT,
      PRJNAME,
      PLAN_ID,
      FK_ETDM_STAGE,
      nvl(USERNAME,'lc_mccain'),
      TIME_STAMP,
      FROM_FACILITY,
      TO_FACILITY,
      FEDERAL_AID,
      STATE_FINANCIAL,
      PRJ_TIMESTAMP,
      PRJ_LOCKED,
      FEATURE_TYPE,
      FK_ORG,
      TEMP_ID,
      FK_PROJECT_CLOB,
      FK_ORG_USER,
      0
      from v_chrono_projects
      where (pk_project,time_stamp) in (
          select rec.fk_project,max(time_stamp)
          from v_chrono_projects
          where pk_project = rec.fk_project
          and time_stamp <= rec.time_stamp);
end loop;
end;
/

********************** Populate T_PROJECT_ALT_HISTORY.FK_ETDM_STATUS *****************

declare
   cursor c is select pk_etdm_status, name from t_etdm_statuses;
begin
for rec in c
loop
   update t_project_alt_history set fk_etdm_status = rec.pk_etdm_status
   where fk_status = rec.name;
end loop;
end;

update t_project_alt_history set fk_etdm_status = 5 where fk_status = 'ETDM Q/A';

********************** Populate T_PROJECT_ALT_HISTORY.FK_ORG_USER *****************

declare
   cursor c is select pk_project_alt, fk_org_user, username from t_project_alt;
begin
for rec in c
loop
   update t_project_alt_history set fk_org_user = rec.fk_org_user
   where fk_project_alt = rec.pk_project_alt
   and username = rec.username;
end loop;
end;
declare
   cursor c is select distinct pk_project, p.fk_org, user_id, pk_org_user
   from t_project_alt_history ah, t_projects p, t_users, t_org_users ou
   where ah.fk_org_user is null
   and fk_project = pk_project
   and p.fk_org = ou.fk_org
   and ah.username = user_id
   and ou.fk_user = pk_user;
begin
for rec in c
loop
   update t_project_alt_history
   set fk_org_user = rec.pk_org_user
   where fk_project = rec.pk_project
   and username = rec.user_id
   and fk_org_user is null;
end loop;
end;

insert into t_org_users (fk_org,fk_user)
select distinct pk_org,pk_user from t_orgs, t_projects, t_project_alt_history ah,  t_users
where fk_project = pk_project
and fk_org = pk_org
and ah.username = user_id
and ah.fk_org_user is null

declare
   cursor c is select distinct pk_project, p.fk_org, user_id, pk_org_user
   from t_project_alt_history ah, t_projects p, t_users, t_org_users ou
   where ah.fk_org_user is null
   and fk_project = pk_project
   and p.fk_org = ou.fk_org
   and ah.username = user_id
   and ou.fk_user = pk_user;
begin
for rec in c
loop
   update t_project_alt_history
   set fk_org_user = rec.pk_org_user
   where fk_project = rec.pk_project
   and username = rec.user_id
   and fk_org_user is null;
end loop;
end;

********************** Populate T_PROJECT_ALT_MILESTONES *****************************

-- there were still issues with null fk_org_users.  so disable fk_org_user constraint,
-- inserted records as shown below.  will fix nulls then re-enable constaint.

declare
   cursor c is select pk_milestone, fk_project_alt, time_stamp
      from t_milestones
      where fk_project_alt not in (
      select fk_project_alt from t_milestones where qaqc = 'Y');
begin
for rec in c
loop
   insert into t_project_alt_milestones select rec.pk_milestone,
      PK_PRJ_ALT,
      FK_PROJECT,
      COST,
      MODE_ROADWAY,
      MODE_TRANSIT,
      MODE_BIKE,
      MODE_PEDESTRIAN,
      LENGTH,
      XMIN,
      XMAX,
      YMIN,
      YMAX,
      PRJALT_FROM_FACILITY,
      PRJALT_TO_FACILITY,
      LOCAL_ID,
      TIME_STAMP,
      MODE_TOBEDETERMINED,
      MODE_RAIL,
      FK_STATUS,
      EDIT_REVIEW_CYCLE,
      CURRENT_REVIEW_START,
      REVIEW_DURATION,
      REVIEW_LOCKED,
      FEATURE_TYPE,
      USERNAME,
      FK_ALT_TYPE,
      FIHS,
      RDWYID,
      BMP,
      EMP,
      NEW_ANALYSIS_RUN,
      ORIG_STATUS,
      PK_PROJECT_ALT,
      TEMP_ID,
      FK_ETDM_STATUS,
      FK_ORG_USER,
      0
      from v_chrono_alternatives
      where (pk_project_alt,time_stamp) in (
         select rec.fk_project_alt,max(time_stamp)
         from v_chrono_alternatives
         where pk_project_alt = rec.fk_project_alt
         and time_stamp <= rec.time_stamp);
end loop;
end;
/

-- There were 95 null fk_org_user value in milestone tables.  The following was done to assign them a
-- superuser fk_org_user value for the appropriate org.
-- pk_user for 'superuser' = 417.

  1  insert into t_org_users (fk_org,fk_user)
  2  select distinct pk_org, 417
  3   from t_orgs o, t_projects p, t_project_alt_milestones m
  4  where fk_org = pk_org
  5  and m.fk_org_user is null
  6  and fk_project = pk_project
  7  minus
  8* select fk_org,fk_user from t_org_users where fk_user = 417
SQL> /

declare
cursor c is select distinct pk_project, fk_org
   from t_projects where pk_project in ( select fk_project
   from t_project_alt_milestones where fk_org_user is null);
begin
for rec in c
loop
   update t_project_alt_milestones set fk_org_user = (
      select distinct pk_org_user
      from t_org_users
      where fk_user = 417
      and fk_org = rec.fk_org)
   where fk_org_user is null
   and fk_project = rec.fk_project;
end loop;
end;
```

```
-- flagged the 95 records in t_project_alt_milestones where
-- fk_org_user was populated with a superuser fk_org_user value.
-- set flag_org_user = 'Y'.

-- re-enabled not null constraint on fk_org_user in the milestone table.

********************** Populate T_PROJECT_CLOB_MILESTONES ********************************

declare
  cursor c is select pk_milestone, fk_project, time_stamp from t_milestones
    where fk_project_alt not in (
    select fk_project_alt from t_milestones where qaqc = 'Y');
begin
for rec in c
loop
  insert into t_project_clob_milestones select rec.pk_milestone,
    FK_PROJECT,
    PK_PRJCLBHISTORY_DATE,
    PURPOSE_NEED,
    PROJ_PUB_CMT_SMRY,
    DESCRIPTION_SMRY,
    DESIRED_FEATURES_CLC,
    PK_PROJECT_CLOB,
    TEMP_ID,
    0
    from v_chrono_prj_clobs
    where (fk_project,PK_PRJCLBHISTORY_DATE) in (
      select rec.fk_project,max(PK_PRJCLBHISTORY_DATE)
      from v_chrono_prj_clobs
      where fk_project = rec.fk_project
      and PK_PRJCLBHISTORY_DATE <= rec.time_stamp);
end loop;
end;
/

*************************** Populate T_PROJECT_SEGMENTS.FK_PROJECT_ALT *********************

declare
cursor c is select distinct fk_project, fK_PRJ_alt from t_project_segments
where fk_project_alt is null;
begin
for rec in c
loop
update t_project_segments set fk_project_alt = ( select pk_project_alt from
t_project_alt where fk_project = rec.fk_project and pk_prj_alt = rec.fk_prj_alt )
where  fk_project = rec.fk_project and fk_prj_alt = rec.fk_prj_alt;
end loop;
end;
/

declare
cursor c is select distinct fk_project, fK_PRJ_alt from t_project_seg_history
where fk_project_alt is null;
begin
for rec in c
loop
update t_project_seg_history set fk_project_alt = ( select pk_project_alt from
t_project_alt where fk_project = rec.fk_project and pk_prj_alt = rec.fk_prj_alt )
where  fk_project = rec.fk_project and fk_prj_alt = rec.fk_prj_alt;
end loop;
end;
```

```
/

********************* Populate T_PROJECT_SEGMENT_MILESTONES *******************************

declare
  cursor c is select pk_milestone, fk_project_alt, time_stamp from t_milestones
    where fk_project_alt not in (
    select fk_project_alt from t_milestones where qaqc = 'Y');
begin
for rec in c
loop
  insert into t_project_segment_milestones select rec.pk_milestone,
    PK_SEGMENT,
    FK_PROJECT,
    FK_PRJ_ALT,
    DATE_STAMP,
    SEGMENT_ID,
    CURR_YEAR,
    CURR_AADT,
    CURR_LANES,
    CURR_CONFIG,
    INTRM_YEAR,
    INTRM_AADT,
    INTRM_LANES,
    INTRM_CONFIG,
    PLAN_YEAR,
    COST_AADT,
    COST_LANES,
    COST_CONFIG,
    NEED_AADT,
    NEED_LANES,
    NEED_CONFIG,
    STREET_NAME,
    FROM_STREET,
    TO_STREET,
    LENGTH,
    FK_JURISDICTION,
    URB_SERVICE,
    TCEA,
    FK_ROADWAY_FCLASS,
    BASEMAP_SEGMENT_ID,
    RDWYID,
    BEGPT,
    ENDPT,
    nvl(USERNAME,'lc_mccain'),
    SEGMENT_ORDER,
    LOCAL_ID,
    SEG_LOCAL_ID,
    SOURCE,
    EDIT_REVIEW_CYCLE,
    FK_PROJECT_ALT,
    TEMP_ID,
    0
    from v_chrono_segments
    where (fk_project_alt,date_stamp) in (
      select rec.fk_project_alt,max(date_stamp)
      from v_chrono_segments
      where fk_project_alt = rec.fk_project_alt
      and date_stamp <= rec.time_stamp);
end loop;
end;
```

```
/

********************** Populate T_SEGMENT_FUNDING_MILESTONES *********************************

cannot populate.
no time_stamp field.

********************** Populate T_SIGNATURES_MILESTONES ********************************

declare
  cursor c is select pk_milestone, fk_project, time_stamp from t_milestones
    where fk_project_alt not in (
    select fk_project_alt from t_milestones where qaqc = 'Y');
begin
for rec in c
loop
  insert into t_signature_milestones select rec.pk_milestone,
    FK_PROJECT,
    FK_USER_ID,
    REVIEW_STATUS,
    DETERM_DATE,
    EXPLANATION,
    TIME_STAMP,
    0
    from v_chrono_signatures
    where (fk_project,time_stamp) in (
       select rec.fk_project,max(time_stamp)
       from v_chrono_signatures
       where fk_project = rec.fk_project
       and time_stamp <= rec.time_stamp);
end loop;
end;
/

********************** Populate T_PROJECT_COMMITMENT_HISTORY.FK_ORG_USER *****************

declare
  cursor c is select fk_project, fk_org_user, fk_userid from t_project_commitment;
begin
for rec in c
loop
  update t_project_commitment_history set fk_org_user = rec.fk_org_user
  where fk_project = rec.fk_project
  and fk_userid = rec.fk_userid;
end loop;
end;

********************** Populate T_PROJECT_COMMIT_MILESTONES *********************************

declare
  cursor c is select pk_milestone, fk_project, time_stamp from t_milestones
    where fk_project_alt not in (
    select fk_project_alt from t_milestones where qaqc = 'Y');
begin
for rec in c
loop
  insert into t_project_commit_milestones select rec.pk_milestone,
    PK_PROJECT_COMMITMENT,
    FK_PROJECT,
    COMMIT_DATE,
    COMMIT_TEXT,
    COMMIT_ORG,
    FK_USERID,
    COMMIT_TEXT_CLOB,
    TIME_STAMP,
    FK_ORG_USER,
    TEMP_ID,
    0
    from v_chrono_commitments
    where (fk_project,time_stamp) in (
       select rec.fk_project,max(time_stamp)
       from v_chrono_commitments
       where fk_project = rec.fk_project
       and time_stamp <= rec.time_stamp);
end loop;
end;
/

********************** Populate T_PROJECT_RESPONSE_MILESTONES ********************************

-- 0 records added.

declare
  cursor c is select pk_milestone, fk_project, time_stamp from t_milestones
    where fk_project_alt not in (
    select fk_project_alt from t_milestones where qaqc = 'Y');
begin
for rec in c
loop
  insert into t_project_response_milestones select rec.pk_milestone,
    PK_RESPONSE,
    RESPONSE_DATE,
    RESPONSE_TEXT,
    FK_REVIEW,
    RESPONSE_ORG,
    LAST_UPDATED,
    FK_PROJECT,
    0,
    FK_ETDM_STAGE,
    PLAN_YEAR
    from v_chrono_responses
    where (fk_project,response_date) in (
       select rec.fk_project,max(response_date)
       from v_chrono_responses
       where fk_project = rec.fk_project
       and response_date <= rec.time_stamp);
end loop;
end;
/
********************** Populate T_PROJECT_REVIEW_MILESTONES ********************************

declare
  cursor c is select pk_milestone, fk_project, alt_id, time_stamp from t_milestones
    where fk_project_alt not in (
    select fk_project_alt from t_milestones where qaqc = 'Y');
begin
for rec in c
loop
  insert into t_project_review_milestones select rec.pk_milestone,
    FK_ISSUE,
    FK_REVIEW_ORG,
    FK_PROJECT,
    FK_ETDM_STAGE,
    AGENCY_INVOLVEMENT,
    REVIEW_TEXT,
    FK_USERID,
    REVIEW_DATE,
    FK_REVIEW_STATUS,
    PK_REVIEW,
    FK_EFFECT,
    FK_PRJ_ALT,
    FK_ORG_USER,
    FK_EFFECT_ID,
    FK_PROJECT_ALT,
    0
    from v_chrono_reviews
    where (fk_project,fk_prj_alt,review_date) in (
       select rec.fk_project,rec.alt_id,max(review_date)
       from v_chrono_reviews
       where fk_project = rec.fk_project
       and review_date <= rec.time_stamp);
end loop;
end;
/

********************** Populate T_ENVIRO_DETERM_MILESTONES ********************************

could not add records.
no timestamp.

********************** Populate T_PN_REVIEW_NEW_MILESTONES ********************************

declare
  cursor c is select pk_milestone, fk_project, time_stamp from t_milestones
    where fk_project_alt not in (
    select fk_project_alt from t_milestones where qaqc = 'Y');
begin
for rec in c
loop
  insert into t_pn_review_new_milestones select rec.pk_milestone,
    PK_PN_REVIEW,
    FK_PROJECT,
    COMMENTS,
    APPROVED_TEXT,
    PN_REVIEW_DATE,
    FK_ORG_USER,
    0
    from v_chrono_pn_review
    where (fk_project,PN_REVIEW_DATE) in (
       select rec.fk_project,max(PN_REVIEW_DATE)
       from v_chrono_responses
       where fk_project = rec.fk_project
       and PN_REVIEW_DATE <= rec.time_stamp);
end loop;
end;
/

********************** Populate T_EFFECTS_SUMMARY_MILESTONES ********************************

declare
  cursor c is select pk_milestone, fk_project, alt_id, time_stamp from t_milestones
    where fk_project_alt not in (
    select fk_project_alt from t_milestones where qaqc = 'Y');
```

```
begin
for rec in c
loop
   insert into t_effects_summary_milestones select
      rec.pk_milestone,
      FK_ISSUE,
      FK_PROJECT,
      FK_ETDM_STAGE,
      FK_USERID,
      FK_SUMMARY_EFFECT,
      SUMMARY_TEXT,
      TECH_STUDIES,
      STATUS_FINAL,
      FK_PRJ_ALT,
      FK_CALC_EFFECT,
      SUMMARY_REPORT_ID,
      PK_EFFECTS_SUMMARY,
      TIME_STAMP,
      0
      from v_chrono_effects
      where (fk_project,fk_prj_alt,time_stamp) in (
          select rec.fk_project,rec.alt_id,max(time_stamp)
          from v_chrono_effects
          where fk_project = rec.fk_project
          and time_stamp <= rec.time_stamp);
end loop;
end;
/

********************* Populate Z_ANALYSIS_REPORT_MILESTONES *******************************

declare
   cursor c is select pk_milestone, fk_project, alt_id, time_stamp from t_milestones
      where fk_project_alt not in (
      select fk_project_alt from t_milestones where qaqc = 'Y');
begin
for rec in c

loop
   insert into Z_ANALYSIS_REPORT_milestones select
      rec.pk_milestone,
      FK_PROJECT,
      FK_PRJ_ALT,
      FK_ANALYSIS_TYPE,
      FK_BUFFER_DISTANCE,
      DK_REPORT_DATE,
      ROUTINE_START_DATE,
      RECORD_TYPE,
      SORT_ID,
      FIELD1,
      FIELD2,
      FIELD3,
      FIELD4,
      FIELD5,
      FIELD6,
      FIELD7,
      FIELD8,
      FIELD9,
      FIELD10,
      FIELD11,
      FIELD12,
    FIELD13,
    FIELD14,
    TOTAL_NUM_FEATURES,
    TOTAL_FEAT_ACRES,
    0
    from Z_ANALYSIS_REPORT_NEW
    where (fk_project,fk_prj_alt,ROUTINE_START_dATE) in (
       select rec.fk_project,rec.alt_id,max(ROUTINE_START_dATE)
       from Z_ANALYSIS_REPORT_NEW
       where fk_project = rec.fk_project
       and ROUTINE_START_dATE <= rec.time_stamp);
end loop;
end;
/
```

A.2 EST Database Changes as of September 30, 2007

Flat files that were being stored on the Web server were imported into the database. The following tables were affected:

- T_DISPUTE_LOG
- T_PROJECT_DOCS & PROJECT_DOCS_MILESTONES
- T_ETAT_LIBRARY
- T_PLAN_REG_CONSISTENCY

A.2.1 T_DISPUTE_LOG

1. Copied dispute log flat files from Web server to temp directory on database server.
2. Used Oracle's SQLLDR to import the dispute_log flat files into the database to a table called T_BLOB.

```
SQL> spool disputelog_prod.txt
SQL> select t_blob_seq.nextval||',3,'''
            ||replace(log_item_url,'/etdmBatch/dispute/')
            ||''','''||lower(substr(path_url,-3,3))||''','||fk_org_user||','''
            ||log_item_date||''''
       from t_dispute_log where log_item_url is not null;
SQL> spool off
SQL> create table temp_blob as select * from t_blob where pk_blob = 0;  /* pk_blob = 0 for 0
rows */

## example of lines output to dispute_prod.txt
oracle@etdmsde>>tail -5 disputelog_prod.txt
1752,7,'Conclusion.pdf','pdf',500,'22-NOV-04 00:00:00'
1753,7,'USFS Dispute Res Mtg Minutes_2-22-07_.pdf','pdf',682,'25-JAN-07 00:00:00'
1754,7,'061205 Minutes Package.pdf','pdf',317,'05-DEC-06 00:00:00'
1755,7,'SR 7 ETDM Meeting Minutes.pdf','pdf',317,'09-NOV-06 00:00:00'
1756,7,'Crosstown Parkway Third E-W Bridge Update 2-20-07.doc','doc',317,'20-FEB-07
00:00:00'

## SQLLDR control file
oracle@etdmsde>>cat disputelog_prod.ctl
LOAD DATA
INFILE 'disputelog_prod.txt'
   INTO TABLE temp_blob
   FIELDS TERMINATED BY ','
   OPTIONALLY ENCLOSED BY "'"
   (pk_blob,
    fk_document_type,
    file_name,
    file_ext,
    file_data LOBFILE(file_name) TERMINATED BY EOF,
    fk_org_user,
    time_stamp)
```

```
oracle@etdmsde>>sqlldr control=disputelog_prod.ctl

SQL> insert into t_blob select * from temp_blob;
```

3. Populated the field FK_BLOB in the T_DISPUTE_LOG table to reference/link to the dispute log documents that were imported to T_BLOB.

```
SQL> /* Values for LOG_ITEM_URL, LOG_ITEM_DATE combination are unique */
SQL> select count(*) from t_dispute_log
    where LOG_ITEM_URL is not null
    group by LOG_ITEM_URL, LOG_ITEM_DATE
    having count(*) > 1;

no rows selected

SQL> alter table t_dispute_log disable all triggers;
SQL> select pk_dispute_log from t_dispute_log for update;
SQL> begin
    for i in (select file_name,time_stamp,pk_blob from t_blob where fk_document_type = 7)
    loop
    update t_dispute_log set fk_blob = i.pk_blob
    where replace(log_item_url,'/etdmBatch/dispute/') = i.file_name
    and log_item_date = i.time_stamp;
    end loop;
    end;
    /
SQL> alter table t_project_docs enable all triggers;
```

A.2.2 T_PROJECT_DOCS & PROJECT_DOCS_MILESTONES

1. Copied project document flat files from web server to temp directory on database server.
2. Used Oralce's SQLLDR to import the project document flat files into the database to a table called T_BLOB.

```
SQL> spool projectdocs_prod.txt
SQL> select t_blob_seq.nextval||',3,"'
            ||replace(path_url,'/etdmBatch/projectdocs/')
            ||'","'||lower(substr(path_url,-3,3))||'",',42/*superuser */,"'||doc_date||'"'
    from t_project_docs where fk_blob is null;
SQL> spool off
SQL> drop table temp_blob;
SQL> create table temp_blob as select * from t_blob where pk_blob = 0;  /* pk_blob = 0 for 0
rows */

## example of lines output to projectdocs_prod.txt
oracle@etdmsde>>tail -10 projectdocs_prod.txt
887,3,"Indirect and Cumulative Effects Action Plan.doc","doc",42,"21-SEP-07 10:49:49"
888,3,"Coastal and Marine Action Plan.doc","doc",42,"21-SEP-07 10:50:17"
889,3,"DEP SR 292 Sorrento Rd.pdf","pdf",42,"24-SEP-07 17:01:54"
890,3,"DHHS SR 173.pdf","pdf",42,"24-SEP-07 17:40:47"
891,3,"WFRPC SR 173.pdf","pdf",42,"24-SEP-07 17:44:26"

## SQLLDR control file
oracle@etdmsde>>cat projectdocs_prod.ctl
```

```
LOAD DATA
INFILE 'projectdocs_prod.txt'
  INTO TABLE temp_blob
  FIELDS TERMINATED BY ','
  OPTIONALLY ENCLOSED BY '"'
  (pk_blob,
   fk_document_type,
   file_name,
   file_ext,
   file_data LOBFILE(file_name) TERMINATED BY EOF,
   fk_org_user,
   time_stamp)

oracle@etdmsde>>sqlldr control=projectdocs_prod.ctl
```

3. Populated the field FK_BLOB in the T_PROJECT_DOCS & T_PROJECT_DOCS_MILESTONES tables to reference/link to the dispute log documents that were imported to T_BLOB.

```
SQL> /* Values for PATH_URL, DOC_DATE combination are unique */
SQL> select PATH_URL,DOC_DATE
  2  from t_project_docs
  3  group by PATH_URL,DOC_DATE
  4  having count(*) > 1;

no rows selected

SQL> alter table t_project_docs disable all triggers;
SQL> select pk_project_doc from t_project_docs for update;
SQL> begin
    for i in (select file_name,time_stamp,pk_blob from t_blob where fk_document_type = 3)
    loop
    update t_project_docs set fk_blob = i.pk_blob
    where replace(path_url,'/etdmBatch/projectdocs/') = i.file_name
    and doc_date = i.time_stamp;
    end loop;
    end;
    /
SQL> alter table t_project_docs enable all triggers;
SQL> alter table t_project_docs modify (fk_blob not null);

SQL> alter table T_PROJECT_DOCS_MILESTONES disable all triggers;
SQL> begin
    for i in (select fk_blob,path_url,pk_project_doc,doc_date from t_project_docs)
    loop
    update t_project_docs_milestones set fk_blob = i.fk_blob
    where path_url = i.path_url
    and doc_date = i.doc_date
    and fk_project_doc = i.pk_project_doc
    and fk_blob is null;
    end loop;
    end;
    /
SQL> alter table T_PROJECT_DOCS_MILESTONES enable all triggers;
SQL> alter table t_project_docs_milestones modify (fk_blob not null);
```

A.2.3 T_ETAT_LIBRARY

The following JSP page was run to transform the existent library records into BLOB object and store them into the Database.

```
<%--
  User: Marcelo.Bosio
  Date: 09/9/2007
  Page to transform library records from physical files to BLOB objects
  and store them into the Database
  File: procOldLibrary.jsp
--%>

<%@ page import="org.apache.velocity.VelocityContext,
                                  org.etdm.model.*,
                                  java.util.Date,
                                  org.etdm.util.StringEST,
                                  java.util.*,
                                  org.hibernate.Session,
                                  org.etdm.appmgr.App,
                                  java.io.*,
                                  org.etdm.application.edms.*"%>

<%
  VelocityContext ctx = App.velocityContext();
          Session hbSession = App.hibernate();
  String fileName="";
  Document newDoc = new Document();
          String urlUploadFile = "/home/upload/etdmBatch/library";
  List<OldLibrary> lib = hbSession
        .createQuery("from OldLibrary lib where lib.document = null")
          .list();
  org.etdm.application.edms.Queue newQueue =
        new org.etdm.application.edms.Queue();
  String fileExt =  "";
  for (OldLibrary l: lib){
          //check if the file exists
          newDoc = new Document(); //Creating a new Blob object
          newQueue = new org.etdm.application.edms.Queue();
        //Creating a new Queue object
          File file = new File(urlUploadFile + "/" + l.getDocumentURL());
          if (file.exists()){
                  InputStream is = new FileInputStream(file);
                  newDoc.setDescription(new StringEST(l.getDocumentName()));
                  newDoc.setDocumentType(DocumentType.ETDM_LIBRARY);
                  fileName = l.getDocumentURL().toString();
                  fileExt =  l.getDocumentURL().toString()
.substring(l.getDocumentURL().length()-3);
                  newDoc.setFileExtension(fileExt);
                  newDoc.setFileName(fileName);
                  newDoc.setBinaryDocument(org
        .hibernate.Hibernate.createBlob(is));
                  Set<OrgUser> identities = l.getUser().getOrgUsers();
                  //If the user has more than one identity,
        //the Super user identity
                //is used. Then, those records needs to be cleaned up
                if (identities.size() == 1) {
                        newDoc.setOrgUser(identities.iterator().next());
                        l.setOrgUser(identities.iterator().next());
                        newQueue.setOrgUser(identities.iterator().next());
                }else{
newDoc.setOrgUser((OrgUser) hbSession.load(OrgUser.class, 42));
                        l.setOrgUser((OrgUser) hbSession.load(
        OrgUser.class, 42));
                        newQueue.setOrgUser((OrgUser) hbSession.load(
        OrgUser.class, 42));
                }
                if (l.getDescription()!= null)
        newQueue.setDoc_dt(new Date(l.getDescription()));
                newQueue.setDocName(new StringEST(l.getDocumentName()));
                newQueue.setXmpt_pub_rec_cd("N");
          newQueue.setDocDesc(new StringEST(l.getDocumentName()));
                Category catType = (Category) hbSession
.createQuery("from Category where description = :paraDesc")
                        .setParameter("paraDesc", l.getCategory().toString())
                        .uniqueResult();
                if (catType != null) newQueue.setFk_Category(catType);
                newQueue.setObsolete("N");
                newQueue.setObsolete_Sync("N");
                //Save objects
                App.beginTransaction();
                        hbSession.saveOrUpdate(newDoc);
                        newQueue.setFk_Blob(newDoc);
                        newQueue.setToBeImported();
                        hbSession.saveOrUpdate(newQueue);
                        l.setDocument(newDoc);
                        hbSession.saveOrUpdate(l);
                App.commit();
          }
  }
%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
   <title>EDMS ETAT Library - Data Transformation</title>
  <meta http-equiv="pragma" content="no-cache">
  <meta http-equiv="cache-control" content="no-cache">
  <meta http-equiv="expires" content="0">
  <meta http-equiv="keywords" content=" ">
  <meta http-equiv="description" content=" ETAT Library - Data Transformation ">
  </head>
  <body>
   EDMS ETAT Library - Data Transformation - Done.<br>
  </body>
</html>
```

A.2.4 T_PLAN_REG_CONSISTENCY

A new table (T_PLAN_REG_CONSISTENCY) was created to replace the now deprecated T_PRJ_CONSISTENCY. The code used to transform the data from the original format to fit the new table is given below under:

```
SQL> alter trigger t_plan_reg_consistency_pk disable;
SQL> alter trigger t_plan_reg_consistency_timestp disable;
SQL> alter trigger t_plan_reg_consistency_hstry disable;

SQL> insert into t_plan_reg_consistency (
    pk_plan_reg_consistency,FK_PROJECT,FK_PLAN_REG_NAME,IS_CONSISTENT)
    select pk_prj_consistency,FK_PROJECT,FK_PLAN_REG_NAME,CONSISTENT
    from t_prj_consistency
    order by pk_prj_consistency;

SQL> select max(PK_PLAN_REG_CONSISTENCY) from t_plan_reg_consistency;
SQL> create sequence t_plan_reg_consistency_seq start with _______;

SQL> alter trigger t_plan_reg_consistency_timestp enable;
SQL> alter trigger t_plan_reg_consistency_hstry enable;
SQL> alter trigger t_plan_reg_consistency_pk enable;

SQL> ALTER TABLE "T_PLAN_REG_CONSISTENCY"
    ADD (CONSTRAINT "T_PLAN_REG_CONSISTENCY_CHK2"
    CHECK(FK_ORG_USER_CRUDBY IS NOT NULL) NOVALIDATE,
    CONSTRAINT "T_PLAN_REG_CONSISTENCY_CHK3" CHECK(TIME_STAMP IS
    NOT NULL) NOVALIDATE);

SQL> create sequence t_plan_reg_con_milestones_seq;

SQL> insert into t_plan_reg_con_milestones
(FK_MILESTONE,FK_PROJECT_MILESTONE,FK_PLAN_REG_CONSISTENCY,
    FK_PROJECT,FK_PLAN_REG_NAME,IS_CONSISTENT)
    select fk_milestone, pk_project_milestone, pK_PLAN_REG_CONSISTENCY,
    con.FK_PROJECT,FK_PLAN_REG_NAME,IS_CONSISTENT
    from t_project_milestones pms, t_plan_reg_consistency con
    where pms.pk_project = con.fk_project;

SQL> rename t_prj_consistency to t_prj_consistency_deprecated;
```

A.3 EST Database Changes October 1, 2007 – December 31, 2008

The following database changes were made during the period of October 1, 2007 through December 31, 2008:

- Populating new table, AT_MILESTONE_GROUP with pre-existing milestone data
- Data cleaning for DOE Org users
- Move FINAL INVOICE PDFs from INVOICE table to T_BLOB table

A.3.1 Populating New Table, AT_MILESTONE_GROUP, with Pre-existing Milestone Data

Date Implemented: October 3, 2007

Background: We have been capturing milestone data at the alternative level since we began milestoning data. We then needed some way of grouping the alternative-level event data by project. The table at_milestone_group was created to maintain associations between alt-level events and project level events. For example, when a given project (e.g., project 1000) goes into ETAT review, data are snapshotted for each alternative in the project (say, Alternatives 1, 2, 3). Ideally, there would be one Identifier for the the event at the project level (project 1000, ETAT Review Start) to which all other data would be tied. As it is, a distinct project event record is stored for each of the 3 alternatives.

The table AT_MILETONE_GROUP stores information about the event for each alternative, and stores the maximum milestone ID for the project event. This max milestone ID is then used as the topmost Identifier for the project event.

```
-- TABLE Definition:  AT_MILESTONE_GROUP
-- Column Name                          Null?    Type
-- ---------------------------------------- -------- -----------
--  FK_MILESTONE_EVENTMAX                    NOT NULL NUMBER(10)
--  FK_MILESTONE                             NOT NULL NUMBER(10)
--  FK_PROJECT                               NOT NULL NUMBER(10)
--  FK_PROJECT_ALT                           NOT NULL NUMBER(10)
--
```

To begin using the table, we first had to populate it with existing milestone data and identify the max milestone ID (FK_MILESTONE_EVENTMAX) per group. The PL/SQL code below was used to populate the table initially. And thereafter, the table has been written to via a database procedure upon each new milestone event.

```
declare
   lastprj number := 0;
   lastms number :=0 ;
   laststage number :=0;
   lasterc number :=0;
   lasttype number :=0;
   lastts date;
   countreview number :=0;
   countpub number :=0;
   altcovered char(1);
```

```
begin
for i in (select * from t_milestones
          where fk_milestone_type in (1,2,3,4)
          order by fk_project,pk_milestone)
loop

if (i.fk_milestone_type in (3,4))
then

select count(distinct fk_project_alt)
  into countreview
  from at_milestone_group
 where fk_milestone_eventmin in (select max(fk_milestone_eventmin) from at_milestone_group
       where fk_etdm_stage = i.fk_etdm_stage and
             edit_review_cycle = i.edit_review_cycle and
             fk_project = i.fk_project and
             fk_milestone_type = 2);

select count(distinct fk_project_alt)
  into countpub
  from at_milestone_group
 where fk_milestone_eventmin in (select max(fk_milestone_eventmin) from at_milestone_group
       where fk_etdm_stage = i.fk_etdm_stage and
             edit_review_cycle = i.edit_review_cycle and
             fk_milestone_type = i.fk_milestone_type and
             fk_project = i.fk_project and
             fk_milestone_type = i.fk_milestone_type and
             fk_project_alt <> i.fk_project_alt);

select decode(count(*),0,'N','Y')
  into altcovered
  from at_milestone_group
 where fk_milestone_eventmin in (select max(fk_milestone_eventmin) from at_milestone_group
       where fk_etdm_stage = i.fk_etdm_stage and
             edit_review_cycle = i.edit_review_cycle and
             fk_milestone_type = i.fk_milestone_type and
             fk_project = i.fk_project and
             fk_milestone_type = i.fk_milestone_type and
             fk_project_alt in i.fk_project_alt);

end if;

if (i.fk_etdm_stage = laststage and
    i.edit_review_cycle = lasterc and
    i.fk_milestone_type = lasttype and
    i.fk_project = lastprj and
    ((i.fk_milestone_type =3 and (countpub < countreview or altcovered = 'N'))
     or (i.fk_milestone_type = 4 and (mod(countpub,countreview)<>0 or altcovered='N'))
     or (i.fk_milestone_type in (1,2) and (i.time_stamp-lastts) between 0 and 45 /*days*/)))
then
   insert into at_milestone_group values(
      null, --fk_milestone_eventmax
      i.pk_milestone,
      i.fk_project,
      i.fk_project_alt,
      i.time_stamp,
      (i.time_stamp-lastts)*24*60,
      i.fk_milestone_type,
      i.fk_etdm_stage,
      i.edit_review_cycle,
      i.qaqc,
      lastms,
      case when i.fk_milestone_type in (3,4) then countreview else null end,
      case when i.fk_milestone_type in (3,4) then countpub else null end,
      case when i.fk_milestone_type in (3,4) then altcovered else null end
   );
else
   insert into at_milestone_group values(
      null,
      i.pk_milestone,
      i.fk_project,
      i.fk_project_alt,
      i.time_stamp,
      0, --OFFSETMIN
      i.fk_milestone_type,
      i.fk_etdm_stage,
      i.edit_review_cycle,
      i.qaqc,
      i.pk_milestone,
      case when i.fk_milestone_type in (3,4) then countreview else null end,
      case when i.fk_milestone_type in (3,4) then countpub else null end,
      case when i.fk_milestone_type in (3,4) then altcovered else null end
   );

   if (i.qaqc <> 'Y' or i.qaqc is null) then
      lastprj := i.fk_project;
      lastms := i.pk_milestone;
      laststage := i.fk_etdm_stage;
      lasterc := i.edit_review_cycle;
      lasttype := i.fk_milestone_type;
      lastts := i.time_stamp;
   end if;
end if;
end loop;
end;
/
```

A.3.2 Data Cleaning for DOE Org Users

Date Implemented: October 24, 2007.

```
/* copy records-to-be-updated to temporary tables (on prod only) */
CREATE table z_effects_summary_origorgusers
   as select * from t_effects_summary
 where (pk_effects_summary,fk_project_alt,fk_issue)
   in (select pk_effects_summary,fk_project_alt,fk_issue
       from etdmv3dev.bug_464_t_effects_summary);
```

```
CREATE table z_effects_summary_ms_orig_ou
    as select * from t_effects_summary_milestones
 where (pk_effects_summary_milestone,fk_project_alt,fk_issue)
    in (select pk_effects_summary_milestone,fk_project_alt,fk_issue
          from etdmv3dev.bug_464_t_effects_summary_ms);

/* disable history, notification, and timestamp triggers.
 * history trigger must be disabled since timestamp is part of history table pk.
 */
ALTER TRIGGER T_EFFECTS_SUMMARY_HISTORY DISABLE;
ALTER TRIGGER T_EFFECTS_SUMMARY_NOTIFSTATUS DISABLE;
ALTER TRIGGER T_EFFECTS_SUMMARY_TIMESTAMP DISABLE;

/* select all records for update to restrict editing until after
 * triggers have been re-enabled.
 */
SELECT PK_EFFECTS_SUMMARY from t_effects_summary for update;

/* Update matching records in effects summary table from cleaned data.
 *
 */
BEGIN
  FOR cur IN (select fk_org_user_to_be, user_id_to_be, a.pk_effects_summary
    from etdmv3dev.bug_464_t_effects_summary a, t_effects_summary  b
    where a.pk_effects_summary = b.pk_effects_summary
    and a.fk_project_alt = b.fk_project_alt
    and a.fk_issue = b.fk_issue
    and a.fk_project  = b.fk_project)
  LOOP
    update t_effects_summary set
    fk_org_user = cur.fk_org_user_to_be,
    fk_userid = cur.user_id_to_be
    where pk_effects_summary = cur.pk_eff_summ;
  END LOOP;
END;
/

/* re-enable history, notification, and timestamp triggers
 *
 */
ALTER TRIGGER T_EFFECTS_SUMMARY_HISTORY ENABLE;
ALTER TRIGGER T_EFFECTS_SUMMARY_NOTIFSTATUS ENABLE;
ALTER TRIGGER T_EFFECTS_SUMMARY_TIMESTAMP ENABLE;

/* disable milestone read-only trigger
 *
 */
ALTER TRIGGER TEMP_DISABLE_UPDATES_VIA_GUI DISABLE;

/* select all records for update to restrict editing until after
 * triggers have been re-enabled.
 */
SELECT PK_EFFECTS_SUMMARY_milestone from t_effects_summary_milestones for update;
```

```
/* Update matching records in effects summary milestone table from cleaned data.
 *
 */
BEGIN
  FOR cur IN (select fk_org_user_to_be, user_id_to_be, a.pk_effects_summary_milestone from
etdmv3dev.bug_464_t_effects_summary_ms a, t_effects_summary_milestones  b
    where a.pk_effects_summary_milestone = b.pk_effects_summary_milestone
    and a.fk_project_alt = b.fk_project_alt
    and a.fk_issue = b.fk_issue
    and a.fk_project  = b.fk_project)
  LOOP
    update t_effects_summary_milestones set
    fk_org_user = cur.fk_org_user_to_be,
    fk_userid = cur.user_id_to_be
    where pk_effects_summary_milestone = cur.pk_eff_summ_ms;
  END LOOP;
END;
/

/* re-enable milestone read-only trigger
 *
 */
  ALTER TRIGGER TEMP_DISABLE_UPDATES_VIA_GUI ENABLE;
```

A.3.3 Move FINAL INVOICE PDFs from INVOICE Table to T_BLOB Table

Below is the SQL code used to add support for storing invoices in the blob table.

Date Implemented: April 10, 2008

```
alter table t_invoice add (
    fk_blob number(10),
    constraint t_invoice_fk_blob foreign key (fk_blob) references t_blob(pk_blob));

alter table t_invoice rename column final_invoice_pdf to final_invoice_pdf_deprecated;

alter table t_invoice_history add (
    fk_blob number(10));

alter table t_invoice_history rename column final_invoice_pdf to final_invoice_pdf_deprecated;

-- MODIFY History trigger to support the new field, FK_BLOB.

CREATE OR REPLACE TRIGGER "T_INVOICE_HSTRY" AFTER
UPDATE
OR DELETE ON "T_INVOICE" FOR EACH ROW WHEN (user like '%DEV%'
or
(new.time_stamp <> old.time_stamp)) declare

   errorMsg varchar(512);
   eRecipient varchar2(200) := 'ccartee@edats.com,rphillips@edats.com';
   eBcc varchar2(2) := 'lc';
```

```
begin

   insert into t_invoice_history values (
      :old.PK_INVOICE,
      :old.FK_AGREEMENT,
      :old.SUBMISSION_EVENT,
      :old.PERIOD_END,
      :old.SALARY_OVERHEAD,
      :old.STATUS,
      :old.SCREENING_PROBLEMS_SUGGESTIONS,
      :old.OTHER_SCREENING_COMMENTS,
      :old.ADMIN_TASKS,
      :old.COORDINATION,
      :old.PERFORMANCE_MEASURES,
      :old.TASK_PROBLEMS,
      :old.ANTI_HOURS,
      :old.ANTI_PEOPLE,
      :old.final_invoice_pdf_deprecated,
      :old.TIME_STAMP,
      :old.invoice_no,
      :old.ANTI_DESCRIPTION,
      :old.fk_org_user_commentby,
      :old.comment_text,
      :old.comment_date,
      :old.salary_overhead_amount,
      :old.final_invoice_pdf_date,
      :old.notes,
      :old.reporting_period_end,
      :old.fk_blob
   );

EXCEPTION
WHEN OTHERS THEN

   errorMsg := sqlerrm;
   if (user = 'ETDMV3') then eRecipient := 'errors@fla-etat.org'; eBcc := null; end if;
   common.javamail('mail.geoplan.ufl.edu','errors@fla-etat.org',eRecipient,'',eBcc,'',
            user||'.t_invoice_hstry trigger failed',
            'PK_INVOICE : '||:new.pk_invoice||CHR(13)||
            'TIME_STAMP : '||to_char(:new.time_stamp,'dd-mon-yy hh24:mi:ss')||CHR(13)||CHR(13)||
            errorMsg);
   raise;

end;

-----------------------------------
-- CREATED:  09 Mar 2006, LCM
-- UPDATED:  14 Jul 2006, LCM.  Added invoice_no.
-- UPDATED:  15 Feb 2007, LCM.  Added salary_overhead_amount.
-- UPDATED:  02 Mar 2007, LCM.  Added final_invoice_pdf_date.
-- UPDATED:  14 May 2007, LCM.  Added notes field.
-- UPDATED:  08 Jan 2008, LCM.  Added reporting_period_end field.
-- UPDATED:  15 Feb 2008, LCM.  Added fk_blob.
```

```
----------------------------------;
/

-- recompile all affected database objects

!scripts/ADMIN/recompile2.sh

alter TRIGGER T_INVOICE_COPYTO_EDMS disable;

-- Copy invoice PDFS to the blob table.

insert into T_BLOB
SELECT
        T_BLOB_SEQ.nextval as blobId,
        5 as documentType,
        'FinalInvoice_'||inv.PK_INVOICE||'.pdf' as fileName,
        'pdf' as fileExt,
        'Final Invoice submission for '||org.org_name as fileDescription,
         inv.FINAL_INVOICE_PDF_deprecated as fileData,
         ou.PK_ORG_USER as orgUser,
         inv.FINAL_INVOICE_PDF_date as timeStamp,
         'Invoice #'||inv.invoice_no||' for '||org.org_name as documentTitle
FROM
        T_INVOICE inv
        join T_AGREEMENTS agr on inv.FK_AGREEMENT = agr.PK_AGREEMENT
        join T_ORGS org on agr.FK_ORG = org.PK_ORG
        join T_org_Users ou on org.PK_ORG = ou.FK_ORG
        join Z_ORG_USER_AUTHORITY oua on ou.PK_ORG_USER = oua.FK_ORG_USER
WHERE 1=1
        and inv.FINAL_INVOICE_PDF_deprecated is not null
        and oua.FK_AUTH_ROLE = 27
        and ou.enabled='Y';

        update T_INVOICE inv
        set inv.FK_BLOB=  (select PK_BLOB from T_BLOB where T_BLOB.File_Name =
'FinalInvoice_'||inv.PK_INVOICE||'.pdf' )
        where inv.FINAL_INVOICE_PDF_deprecated is not null;

-- modify trigger, T_INVOICE_COPYTO_EDMS, to support new fk_blob field.

CREATE OR REPLACE TRIGGER "T_INVOICE_COPYTO_EDMS"
    AFTER
INSERT
OR UPDATE OF "FK_BLOB" ON "T_INVOICE" FOR EACH ROW WHEN (new.fk_blob is not null
and
new.fk_blob <> nvl(old.fk_blob,0)

) declare

   errorMsg varchar2(512);

begin

   insert into t_edms_queue (
      fk_category,
      docname,
      docdesc,
      doc_dt,
      conto,
      invc_id,
      finproj,
      xmpt_pub_rec_cd,
      agr_id,
      fk_invoice,
      fk_org_user,
      tobeimported,
      fk_blob
   )
   select
      129,
      org_name,
      'Final Invoice for '||org_name||' '||upper(agreement_type)||':'||master_no||'
INVOICE_NO:'||:new.invoice_no||' for PERIOD_ENDING:'||to_char(event_date,'dd-MON-yy'),
      :new.time_stamp,
      decode(fk_agreement_type,2,master_no,null),
      :new.invoice_no,
      finance_number,
      'Y',
      decode(fk_agreement_type,1,master_no,null),
      :new.pk_invoice,
      nvl(:new.fk_org_user_commentby,42),
      'Y',
      :new.fk_blob
   from
      t_agreements,
      t_orgs,
      lu_agreement_types,
      t_funding_events
  where
      pk_agreement = :new.fk_agreement
      and pk_org = fk_org
      and fk_agreement_type = pk_agreement_type
      and :new.period_end = pk_funding_event (+)
      and (:new.pk_invoice,:new.time_stamp) not in (select fk_invoice,doc_dt from t_edms_queue
where fk_invoice is not null and doc_dt is not null)
      ;

EXCEPTION
WHEN OTHERS THEN

   errorMsg := sqlerrm;

   javamail2.send('mail.geoplan.ufl.edu','lc','lc','','',user || '.t_invoice_copyto_edms trigger failed',
      'pk_invoice : '||:new.pk_invoice||CHR(13)||CHR(13)||
      errorMsg);

end;
```

```
-----------------------------------------------
-- CREATED:  11 Oct 2007.  LCM, GeoPlan Center.
-- UPDATED:  15 Feb 2008.  Updated to work with new FK_BLOB field as requested.  For
--              more info see Bug 810.  LCM, GeoPlan Center
-----------------------------------------------;
/

alter table t_edms_queue drop constraint t_edms_q_chk_noblob_invc_an;
alter table t_edms_queue add constraint t_edms_q_chk_noblob_an check (fk_category <> 128 or
fk_blob is null);

select pk_invoice,fk_blob from t_invoice where fk_blob is not null or final_invoice_pdf_deprecated
is not null
minus
select fk_invoice,fk_blob from t_edms_queue;

alter table t_edms_queue disable all triggers;
select pk_edms_import from t_edms_queue for update

begin
for i in (select pk_invoice,fk_blob from t_invoice where fk_blob is not null
         and (pk_invoice,1) in (select fk_invoice,count(*) from t_edms_queue group by fk_invoice))
loop
update t_edms_queue set fk_blob = i.fk_blob where fk_invoice=i.pk_invoice and fk_blob is null;
end loop;
end;
/

select pk_invoice,fk_blob from t_invoice where fk_blob is not null or final_invoice_pdf_deprecated
is not null
minus
select fk_invoice,fk_blob from t_edms_queue;


alter table t_edms_queue enable all triggers;

------------------------

alter table t_edms_queue disable all triggers;

insert into T_BLOB
SELECT
       T_BLOB_SEQ.nextval as blobId,
       5 as documentType,
       'FinalInvoice_'||inv.FK_INVOICE||'.pdf' as fileName,
       'pdf' as fileExt,
       'Final Invoice submission for '||org.org_name as fileDescription,
        inv.FINAL_INVOICE_PDF_deprecated as fileData,
        ou.PK_ORG_USER as orgUser,
        inv.FINAL_INVOICE_PDF_date as timeStamp,
        'Invoice #'||inv.invoice_no||' for '||org.org_name as documentTitle
FROM
       T_INVOICE_HISTORY inv
       join T_AGREEMENTS agr on invp.FK_AGREEMENT = agr.PK_AGREEMENT
```

```
        join T_ORGS org on agr.FK_ORG = org.PK_ORG
        join T_org_Users ou on org.PK_ORG = ou.FK_ORG
        join Z_ORG_USER_AUTHORITY oua on ou.PK_ORG_USER = oua.FK_ORG_USER
WHERE 1=1
        and pk_invoice=fk_invoice
        --and (inv.fk_invoice,final_invoice_pdf_date,1) in (select fk_invoice,doc_dt,count(*) from
t_edms_queue where fk_blob is null group by fk_invoice,doc_dt)
        --and
('FinalInvoice_'||inv.FK_INVOICE||'.pdf',ou.PK_ORG_USER,inv.FINAL_INVOICE_PDF_date) not
in (select file_name,fk_org_user,time_stamp from t_blob)
        and (inv.fk_invoice,inv.time_stamp) in
            (select i.fk_invoice,min(i.time_stamp)
               from t_invoice_history i,
                    t_edms_queue q
              where i.fk_invoice=q.fk_invoice
                and q.fk_blob is null
                and q.doc_dt = i.final_invoice_pdf_date
                and i.final_invoice_pdf_deprecated is not null
                and trunc(i.time_stamp) = q.doc_dt
              group by  i.fk_invoice,final_invoice_pdf_date)
        and inv.FINAL_INVOICE_PDF_deprecated is not null
        and oua.FK_AUTH_ROLE = 27
        and ou.enabled='Y';

-----------------------

        update T_INVOICE_HISTORY inv
        set inv.FK_BLOB=  (select PK_BLOB from T_BLOB where T_BLOB.File_Name =
'FinalInvoice_'||inv.fK_INVOICE||'.pdf' and time_stamp = inv.final_invoice_pdf_date )
        where inv.FINAL_INVOICE_PDF_deprecated is not null
        and (fk_invoice,time_stamp)
         in (select i.fk_invoice,min(i.time_stamp)
               from t_invoice_history i,
                    t_edms_queue q
              where i.fk_invoice=q.fk_invoice
                and q.fk_blob is null
                and q.doc_dt = i.final_invoice_pdf_date
                and i.final_invoice_pdf_deprecated is not null
                and trunc(i.time_stamp) = q.doc_dt
              group by  i.fk_invoice,final_invoice_pdf_date);

-----------------------------

select pk_invoice,fk_blob from t_invoice where fk_blob is not null or final_invoice_pdf_deprecated
is not null
minus
select fk_invoice,fk_blob from t_edms_queue;

alter table t_edms_queue disable all triggers;
select pk_edms_import from t_edms_queue for update

set pagesize 0;
set trimspool on;
spool copyto_edms.sql
```

```
select 'update t_edms_queue set fk_blob='||fk_blob||' where fk_invoice='||fk_invoice
        ||' and doc_dt=to_date('''||final_invoice_pdf_date||''') and fk_blob is null;'
       from t_invoice_history where fk_blob is not null;
spool off
!vi copyto_edms.sql
-- %s/''/'/g
@copyto_edms.sql

-----------------------------------

set pagesize 0;
set trimspool on;
spool copyto_edms.sql
select 'update t_edms_queue set fk_blob='||fk_blob||' where fk_invoice='||pk_invoice
        ||' and doc_dt=to_date('''||final_invoice_pdf_date||''') and fk_blob is null;'
       from t_invoice where pk_invoice in (select fk_invoice from t_edms_queue where fk_blob is
null);
spool off
!vi copyto_edms.sql
-- %s/''/'/g
@copyto_edms.sql
set pagesize 100

select pk_invoice,fk_blob from t_invoice where fk_blob is not null or final_invoice_pdf_deprecated
is not null
minus
select fk_invoice,fk_blob from t_edms_queue;

select count(*) from t_edms_queue where fk_invoice is not null and fk_blob is null;

alter table t_edms_queue enable all triggers;
```

A.4 EST Database Changes January 1, 2009 – December 31, 2009

The following database changes were made during the period of January 1, 2009 through December 31, 2009.

A.4.1 Transforming data from S_SEGMENTS_REVIEWED and S_POLYGONS_REVIEWED to S_SEGMENTS_MILESTONES and S_POLYGONS_MILESTONES

Since the inception of ETDM, we were capturing and storing snapshots of project segments at the end of each project "edit review cycle," or before projects entered their second, third, or higher review . When support for project polygons was added, we began snapshotting polygon data in the same way. The data were stored in s_segments_reviewed and s_polygons_reviewed, respectively. To be consistent with the way in which we have been snapshotting and storing other project review-related data, milestone layers were created for project segments, polygons and points (newly supported). The new milestone spatial layers, s_segments_milestones, s_polygons_milestones, and s_points_milestones, are being populated at the beginning of each review undergone by a project alternative instead of at the end of what we've referred to as an edit review cycle.

The steps, ArcSDE commands and SQL statements, used to transform data from s_segments_reviewed and s_polygons_reviewed to s_segments_milestones and s_polygons_milestones are provided below. Note: there were no points data to transform.

1. Created oracle views, v_milestones, v_segments_to_milestone, and v_polygons_to_milestone, to faciltiate data transformation by joining together data from t_milestones, t_project_milestones, t_project_alt_milestones, and t_project_segment_milestones or t_project_polygon_milestones. See SQL statements below.

```
create view V_MILESTONES
as
select pk_milestone,
       ms.fk_project,
       pm.prjname,
       pm.fk_org,
       ms.fk_project||'-'||ms.alt_id as alt_id,
       ms.fk_project_alt,
       am.fk_alt_type,
       am.fk_etdm_status,
       ms.edit_review_cycle,
       ms.fk_etdm_stage,
       nvl(am.current_reiew_start,ms.time_stamp) as current_review_start,
       ms.fk_milestone_type
  from t_milestones ms, t_project_milestones pm, t_project_alt_milestones am
 where fk_milestone_type=1
       and ms.pk_milestone=pm.fk_milestone
       and ms.pk_milestone=am.fk_milestone
       and pm.test_project_ind='N'
/

create view V_SEGMENTS_TO_MILESTONE
as
select pk_milestone as fk_milestone,
       ms.fk_project,
       ms.prjname,
       ms.fk_org,
       ms.alt_id,
       ms.fk_project_alt,
       ms.fk_alt_type,
       ms.fk_etdm_status,
       ms.edit_review_cycle,
       ms.fk_etdm_stage,
       ms.current_review_start,
       ms.fk_milestone_type,
       fms.segment_id,
       fms.pk_segment as fk_segment,
       fms.street_name,
       fms.from_street,
       fms.to_street,
       fms.length
from
       t_project_segment_milestones fms,
            v_milestones ms
        where
            fms.fk_milestone=ms.pk_milestone
            and ms.fk_milestone_type=1
            and (pk_segment,fk_milestone) not in (select fk_segment,fk_milestone from
         s_segments_milestones)
/
        create view V_POLYGONS_TO_MILESTONE
        as
        select pk_milestone as fk_milestone,
            ms.fk_project,
            ms.prjname,
            ms.fk_org,
            ms.alt_id,
            ms.fk_project_alt,
            ms.fk_alt_type,
            ms.fk_etdm_status,
            ms.edit_review_cycle,
            ms.fk_etdm_stage,
            ms.current_review_start,
            ms.fk_milestone_type,
            fms.poly_id,
            fms.fk_poly as fk_polygon,
            fms.street_name,
            fms.from_street,
            fms.to_street
        from
            t_project_polygon_milestones fms,
            v_milestones ms
        where
            fms.fk_milestone=ms.pk_milestone
            and ms.fk_milestone_type=1
            and (fk_poly,fk_milestone) not in (select fk_polygon,fk_milestone from s_
        polygons_milestones)
```

2. Created spatial views, sv_segments_to_milestone and sv_polygons_to_milestone, to facilitate data transformation by joining together data from s_segments_reviewed or s_polygons reviewed and one of the new created oracle views, v_segments_to_milestone or v_polygons_to_milestone. See ArcSDE commands below.

   ```
    sdetable -o create_view -T sv_segments_to_milestone \
    -t "s_segments_reviewed,v_segments_to_milestone" \
    -c "s_segments_reviewed.shape, v_segments_to_milestone.*" \
   -w "s_segments_reviewed.fk_segment = v_segments_to_milestone.fk_segment" -u -p


    sdetable -o create_view -T sv_polygons_to_milestone \
    -t "s_polygons_reviewed,v_polygons_to_milestone" \
    -c "s_polygons_reviewed.shape, v_polygons_to_milestone.*" \
   -w "s_polygons_reviewed.fk_polygon = v_polygons_to_milestone.fk_polygon" -u -p
   ```

3. Created new spatial layers, s_segments_milestones and s_polygons_milestones, from data represented in the spatial views, sv_segments_to_milestone and sv_polygons_to_milestone.

   ```
    sdeexport -o create -l sv_segments_to_milestone,shape -f - -u -p | sdeimport -o create -l
   s_segments_milestones,shape -f - -u -p

     sdeexport -o create -l sv_polygons_to_milestone,shape -f - -u -p | sdeimport -o create -l
   s_polygons_milestones,shape -f - -u -p
   ```

4. Recreated spatial views to pull data from s_segments and s_polygons for alternatives that had been reviewed or were in review but had not yet had their spatial features copied to s_segments_reviewed or s_polygons_reviewed, and were considered to be milestone data.

   ```
     sdetable -o create_view -T sv_segments_to_milestone \
     -t "s_segments,v_segments_to_milestone" \
     -c "s_segments.shape, v_segments_to_milestone.*" \
    -w "s_segments.fk_segment = v_segments_to_milestone.fk_segment" -u -p

     sdetable -o create_view -T sv_polygons_to_milestone \
     -t "s_polygons,v_polygons_to_milestone" \
     -c "s_polygons.shape, v_polygons_to_milestone.*" \
    -w "s_polygons.fk_polygon = v_polygons_to_milestone.fk_polygon" -u -p
   ```

5. Appended s_segments_milestones and s_polygons_milestones with new data from spatial views.

   ```
     sdeexport -o create -l sv_segments_to_milestone,shape -f - -u -p | sdeimport -o append -l
   s_segments_milestones,shape -f - -u -p

     sdeexport -o create -l sv_polygons_to_milestone,shape -f - -u -p | sdeimport -o append -l
   s_polygons_milestones,shape -f - -u –p
   ```

A.5 EST Database Changes January 1, 2010 – August 31, 2011

The following database changes were made during the period of January 1, 2010 through August 31, 2011.

A.5.1 Data Cleaning for IWHRS and POTHAB_RCH Analysis Results

Data cleaning was needed to remove duplicate values from the GIS analysis report for datasets IWHRS and POTHAB_RCH, and summing acres and pct_acreage where needed. The only affected records were those for analysis types, 186 and 4969, for analyses performed after April 1, 2010. Data changes were captured in temporary tables for QAQC/verification before production records were modified. For more information on the QA/QC process, see: https://codebase.fla-etat.org/bugzilla/show_bug.cgi?id=1950#c1.

Upon verification:

1. Records were deleted from t_analysis_report where they matched records in TEMP_ANALYSISRPT_186_4969_OLD, and pushed to history

2. Records were copied from TEMP_ANALYSISRPT_186_4969_NEW to t_analysis_report

And for milestoned data:

3. Records were deleted from t_analysis_report_milestones where they matched records in TEMP_ANALYSIS_MS_186_4969_OLD, and pushed to history

4. Records were copied from TEMP_ANALYSIS_MS_186_4969_NEW to t_analysis_report_milestones

Below are the SQL statements that were used to create the temporary tables mentioned above, and to delete and replace duplicate records 'cleaned' records.

```
---------------------------------------------------------
create or replace view v_analysis_grid_desc_dups
as
select distinct
       a.pk_analysis_type fk_analysis_type
       --,a.fk_layer_descript,
       --l.layername,
       --g.descript,
       --count(distinct g.value) count_per_descript
  from t_grid_lookup g,
       t_analysis_types a,
       t_layer_descript l
 where g.layer = l.layername
   and a.fk_layer_descript = l.pk_layer_descript
 group by a.pk_analysis_type,
       a.fk_layer_descript,
       l.layername,
       g.descript
having count(distinct g.value) > 1
/
---------------------------------------------------------
create view v_analysis_rpt_grid_summ
as
select FK_PROJECT,
       FK_PRJ_ALT,
       FK_PROJECT_ALT,
       FK_ANALYSIS_TYPE,
       BUFFER_DISTANCE,
       MIN(SORT_ID) OLD_SORT_ID,
       routine_start_date,
       max(time_stamp) as time_stamp,
       record_type,
       FIELD1,
       to_char(sum(field2)) field2,
       to_char(sum(field3)) field3,
       TOTAL_NUM_FEATURES,
       TOTAL_FEAT_ACRES
  from t_analysis_report
 where (fk_project_alt,fk_analysis_type,buffer_distance,routine_start_date)
    in (select fk_project_alt,fk_analysis_type,buffer_distance,routine_start_date
       from t_analysis_report
       where fk_analysis_type in (select fk_analysis_type from v_analysis_grid_desc_dups)
       and record_type='V'
       group by fk_project_alt,fk_analysis_type,buffer_distance,routine_start_date,field1
       having count(*) > 1)
   and record_type = 'V'
 group by
       FK_PROJECT,
       FK_PRJ_ALT,
       FK_PROJECT_ALT,
       FK_ANALYSIS_TYPE,
       BUFFER_DISTANCE,
       routine_start_date,
       record_type,
       FIELD1,
       TOTAL_NUM_FEATURES,
       TOTAL_FEAT_ACRES
/
-------------------------------------------------------
create table temp as
select * from v_analysis_rpt_grid_summ
union
select FK_PROJECT,
       FK_PRJ_ALT,
       FK_PROJECT_ALT,
       FK_ANALYSIS_TYPE,
       BUFFER_DISTANCE,
       SORT_ID,
       routine_start_date,
       time_stamp,
       record_type,
       FIELD1,
       field2,
       field3,
       TOTAL_NUM_FEATURES,
       TOTAL_FEAT_ACRES
from t_analysis_report
where record_type <> 'V'
  and (fk_project_alt,fk_analysis_type,buffer_distance,routine_start_date)
   in (select fk_project_alt,fk_analysis_type,buffer_distance,routine_start_date from
v_analysis_rpt_grid_summ);
-------------------------------------------------------
create table temp_analysisrpt_186_4969_new as select
       rownum id,
       FK_PROJECT,
       FK_PRJ_ALT,
       FK_PROJECT_ALT,
       FK_ANALYSIS_TYPE,
       BUFFER_DISTANCE,
       OLD_SORT_ID,
       routine_start_date,
       time_stamp,
       record_type,
       FIELD1,
       field2,
       field3,
       TOTAL_NUM_FEATURES,
      TOTAL_FEAT_ACRES
      from temp
      order by fk_project_alt,fk_analysis_type,OLD_sort_id;

alter table temp_analysisrpt_186_4969_new add (sort_id number);

begin for i in (
  select fk_project_alt,fk_analysis_type,
  min(old_sort_id) min_sort_id,
  min(id) min_id
  from temp_analysisrpt_186_4969_new where record_type = 'H'
  group by fk_project_alt,fk_analysis_type
) loop
update temp_analysisrpt_186_4969_new set sort_id = id - i.min_id + i.min_sort_id
where fk_project_alt = i.fk_project_alt
and fk_analysis_type = i.fk_analysis_type
and sort_id is null;
end loop;
end;
/
---------------------------------------------------------
create table temp_analysisrpt_186_4969_old as select * from t_analysis_report
where (fk_project_alt,fk_analysis_type,buffer_distance,routine_start_date)
  in (select fk_project_alt,fk_analysis_type,buffer_distance,routine_start_date from
temp_analysisrpt_186_4969_new);

---------------------------------------------------------
/*** Replace duplicate records in t_analysis_report ****/
delete from t_analysis_report
where (fk_project_alt,fk_analysis_type,buffer_distance,routine_start_date)
  in (select fk_project_alt,fk_analysis_type,buffer_distance,routine_start_date from
temp_analysisrpt_186_4969_new);

insert into t_analysis_report (
      FK_PROJECT,
      FK_PRJ_ALT,
      FK_PROJECT_ALT,
      FK_ANALYSIS_TYPE,
      BUFFER_DISTANCE,
      SORT_ID,
      routine_start_date,
      time_stamp,
      record_type,
      FIELD1,
      field2,
      field3,
      TOTAL_NUM_FEATURES,
      TOTAL_FEAT_ACRES
) select
      FK_PROJECT,
      FK_PRJ_ALT,
      FK_PROJECT_ALT,
      FK_ANALYSIS_TYPE,
      BUFFER_DISTANCE,
      SORT_ID,
      routine_start_date,
      time_stamp,
      record_type,
      FIELD1,
      field2,
      field3,
      TOTAL_NUM_FEATURES,
      TOTAL_FEAT_ACRES
from temp_analysisrpt_186_4969_new
order by id;

-----------------------------------------------------------
create view v_analysis_rpt_ms_grid_summ
as
select FK_MILESTONE,
      FK_PROJECT,
      FK_PRJ_ALT,
      FK_PROJECT_ALT,
      FK_ANALYSIS_TYPE,
      BUFFER_DISTANCE,
      MIN(SORT_ID) OLD_SORT_ID,
      routine_start_date,
      max(time_stamp) as time_stamp,
      record_type,
      FIELD1,
      to_char(sum(field2)) field2,
      to_char(sum(field3)) field3,
      TOTAL_NUM_FEATURES,
      TOTAL_FEAT_ACRES
  from t_analysis_report_milestones
 where (fk_milestone,fk_project_alt,fk_analysis_type,buffer_distance,routine_start_date)
   in (select fk_milestone,fk_project_alt,fk_analysis_type,buffer_distance,routine_start_date
      from t_analysis_report_milestones
      where fk_analysis_type in (select fk_analysis_type from v_analysis_grid_desc_dups)
      and record_type='V'
      group by
fk_milestone,fk_project_alt,fk_analysis_type,buffer_distance,routine_start_date,field1
      having count(*) > 1)
  and record_type = 'V'
 group by
      FK_MILESTONE,
      FK_PROJECT,
      FK_PRJ_ALT,
      FK_PROJECT_ALT,
      FK_ANALYSIS_TYPE,
      BUFFER_DISTANCE,
      routine_start_date,
      record_type,
      FIELD1,
      TOTAL_NUM_FEATURES,
      TOTAL_FEAT_ACRES
/
-----------------------------------------------------------
```

```
drop table temp;

create table temp as
select * from v_analysis_rpt_ms_grid_summ
union
select FK_MILESTONE,
      FK_PROJECT,
      FK_PRJ_ALT,
      FK_PROJECT_ALT,
      FK_ANALYSIS_TYPE,
      BUFFER_DISTANCE,
      SORT_ID,
      routine_start_date,
      time_stamp,
      record_type,
      FIELD1,
      field2,
      field3,
      TOTAL_NUM_FEATURES,
      TOTAL_FEAT_ACRES
from t_analysis_report_milestones
where record_type <> 'V'
  and (fk_milestone,fk_project_alt,fk_analysis_type,buffer_distance,routine_start_date)
   in (select fk_milestone,fk_project_alt,fk_analysis_type,buffer_distance,routine_start_date from
v_analysis_rpt_ms_grid_summ);

-----------------------------------------------------------
create table temp_analysis_ms_186_4969_new as select
      rownum id,
      FK_MILESTONE,
      FK_PROJECT,
      FK_PRJ_ALT,
      FK_PROJECT_ALT,
      FK_ANALYSIS_TYPE,
      BUFFER_DISTANCE,
      OLD_SORT_ID,
      routine_start_date,
      time_stamp,
      record_type,
      FIELD1,
      field2,
      field3,
      TOTAL_NUM_FEATURES,
      TOTAL_FEAT_ACRES
      from temp
      order by fk_milestone,fk_project_alt,fk_analysis_type,OLD_sort_id;

alter table temp_analysis_ms_186_4969_new add (sort_id number);

begin for i in (
   select fk_milestone,fk_project_alt,fk_analysis_type,
   min(old_sort_id) min_sort_id,
   min(id) min_id
   from temp_analysis_ms_186_4969_new where record_type = 'H'
  group by fk_milestone,fk_project_alt,fk_analysis_type
) loop
update temp_analysis_ms_186_4969_new set sort_id = id - i.min_id + i.min_sort_id
where fk_milestone = i.fk_milestone
and fk_project_alt = i.fk_project_alt
and fk_analysis_type = i.fk_analysis_type
and sort_id is null;
end loop;
end;
/
--------------------------------------------------------
create table temp_analysis_ms_186_4969_old as select * from t_analysis_report_milestones
where (fk_milestone,fk_project_alt,fk_analysis_type,buffer_distance,routine_start_date)
  in (select fk_milestone,fk_project_alt,fk_analysis_type,buffer_distance,routine_start_date from
temp_analysis_ms_186_4969_new);

-------------------------------------------------------------------
/*** Replace duplicate records in t_analysis_report_milestones ****/

delete from t_analysis_report_milestones
where (fk_milestone, fk_project_alt,fk_analysis_type,buffer_distance,routine_start_date)
  in (select fk_milestone, fk_project_alt,fk_analysis_type,buffer_distance,routine_start_date from
temp_analysis_ms_186_4969_new);

insert into t_analysis_report_milestones (
      fk_milestone,
      FK_PROJECT,
      FK_PRJ_ALT,
      FK_PROJECT_ALT,
      FK_ANALYSIS_TYPE,
      BUFFER_DISTANCE,
      SORT_ID,
      routine_start_date,
      time_stamp,
      record_type,
      FIELD1,
      field2,
      field3,
      TOTAL_NUM_FEATURES,
      TOTAL_FEAT_ACRES
) select
      fk_milestone,
      FK_PROJECT,
      FK_PRJ_ALT,
      FK_PROJECT_ALT,
      FK_ANALYSIS_TYPE,
      BUFFER_DISTANCE,
      SORT_ID,
      routine_start_date,
      time_stamp,
      record_type,
      FIELD1,
      field2,
      field3,
        TOTAL_NUM_FEATURES,
        TOTAL_FEAT_ACRES
from temp_analysis_ms_186_4969_new
order by id;

/**** UPON verification and bug closure, DELETED all temp objects and views created.  ***/
```

A.5.2 Move Analysis Report Tables to a New "Reports" Schema

The report tables (for alternative and feature level) analysis results, and their associated triggers and indexes were moved to a new "reportsv3" schema on 2011-02-05 to reduce the space needed for nightly dumps/backups. Analysis report **milestone data were retained in the current schema (etdmv3) and have continued to be captured with the nightly dumps. Tables in the new schema were put on a different backup schedule -- one in which they are being backed up incrementally each night, and then backed up with a full schema dump each month.

The steps for moving the tables to the new schema are listed below:

1. Created a dump file of the tables to be moved using Oracle's import command.

```
OS> exp etdmv3 tables=\(T_ANALYSIS_REPORT, T_ANALYSIS_REPORT_HISTORY, \
   T_ANALYSIS_REPORT_FEATURE, T_ANALYSIS_REPORT_FEAT_HIST, \
   T_ANALYSIS_FEAT_LIST_UNQ, T_ANALYSIS_ALT_LIST_UNQ, \
   T_ANALYSIS_RPT_LIST_DATA_MS, T_ANALYSIS_RPT_SUMM_PL_RS_2_MS, \
   T_ANALYSIS_RPT_SUMM_PT_LN_D_MS, T_ANALYSIS_ALT_MS_LIST_UNQ, \
   T_ANALYSIS_FT_LIST_DATA_MS, T_ANALYSIS_FT_SUMM_PL_RS_2_MS, \
   T_ANALYSIS_FT_SUMM_PT_LN_D_MS, T_ANALYSIS_FEAT_MS_LIST_UNQ\) \
   rows=y indexes=y constraints=y grants=y triggers=y statistics=none file=etdmv3_reports.dmp
```

2. Imported tables to the new schema using Oracle's import command.

```
OS> imp system fromuser=etdmv3 touser=reportsv3 file=etdmv3_reports.dmp ignore=y
```

3. Renamed tables in old schema with the prefix 'X' prior to dropping.

```
SQL> connect etdmv3
SQL> rename T_ANALYSIS_REPORT to T_ANALYSIS_REPORTX;
SQL> rename T_ANALYSIS_REPORT_HISTORY to T_ANALYSIS_REPORT_HISTORYX;
SQL> rename T_ANALYSIS_REPORT_FEATURE to T_ANALYSIS_REPORT_FEATUREX;
SQL> rename T_ANALYSIS_REPORT_FEAT_HIST to T_ANALYSIS_REPORT_FEAT_HISTX;
SQL> rename T_ANALYSIS_FEAT_LIST_UNQ to T_ANALYSIS_FEAT_LIST_UNQX;
SQL> rename T_ANALYSIS_ALT_LIST_UNQ to T_ANALYSIS_ALT_LIST_UNQX;
SQL> rename T_ANALYSIS_RPT_LIST_DATA_MS to T_ANALYSIS_RPT_LIST_DATA_MSX;
SQL> rename T_ANALYSIS_RPT_SUMM_PL_RS_2_MS to
T_ANALYSIS_RPT_SUMM_PL_RS_2_MSX;
```

```
SQL> rename T_ANALYSIS_RPT_SUMM_PT_LN_D_MS to
T_ANALYSIS_RPT_SUMM_PT_LN_D_MSX;

SQL> rename T_ANALYSIS_ALT_MS_LIST_UNQ to T_ANALYSIS_ALT_MS_LIST_UNQX;

SQL> rename T_ANALYSIS_FT_LIST_DATA_MS to T_ANALYSIS_FT_LIST_DATA_MSX;

SQL> rename T_ANALYSIS_FT_SUMM_PL_RS_2_MS to T_ANALYSIS_FT_SUMM_PL_RS_2_MSX;

SQL> rename T_ANALYSIS_FT_SUMM_PT_LN_D_MS to
T_ANALYSIS_FT_SUMM_PT_LN_D_MSX;

SQL> rename T_ANALYSIS_FEAT_MS_LIST_UNQ to T_ANALYSIS_FEAT_MS_LIST_UNQX;
```

4. Created synonyms under the old schema for the tables under the new schema.

```
SQL> connect etdmv3

SQL> create synonym T_ANALYSIS_REPORT for reportsv3.T_ANALYSIS_REPORT;

SQL> create synonym T_ANALYSIS_REPORT_HISTORY for
reportsv3.T_ANALYSIS_REPORT_HISTORY;

SQL> create synonym T_ANALYSIS_REPORT_FEATURE for
reportsv3.T_ANALYSIS_REPORT_FEATURE;

SQL> create synonym T_ANALYSIS_REPORT_FEAT_HIST for
reportsv3.T_ANALYSIS_REPORT_FEAT_HIST;

SQL> create synonym T_ANALYSIS_FEAT_LIST_UNQ for reportsv3.T_ANALYSIS_FEAT_LIST_UNQ;

SQL> create synonym T_ANALYSIS_ALT_LIST_UNQ for reportsv3.T_ANALYSIS_ALT_LIST_UNQ;

SQL> create synonym T_ANALYSIS_RPT_LIST_DATA_MS for
reportsv3.T_ANALYSIS_RPT_LIST_DATA_MS;

SQL> create synonym T_ANALYSIS_RPT_SUMM_PL_RS_2_MS for
reportsv3.T_ANALYSIS_RPT_SUMM_PL_RS_2_MS;

SQL> create synonym T_ANALYSIS_RPT_SUMM_PT_LN_D_MS for
reportsv3.T_ANALYSIS_RPT_SUMM_PT_LN_D_MS;

SQL> create synonym T_ANALYSIS_ALT_MS_LIST_UNQ for
reportsv3.T_ANALYSIS_ALT_MS_LIST_UNQ;

SQL> create synonym T_ANALYSIS_FT_LIST_DATA_MS for
reportsv3.T_ANALYSIS_FT_LIST_DATA_MS;

SQL> create synonym T_ANALYSIS_FT_SUMM_PL_RS_2_MS for
reportsv3.T_ANALYSIS_FT_SUMM_PL_RS_2_MS;

SQL> create synonym T_ANALYSIS_FT_SUMM_PT_LN_D_MS for
reportsv3.T_ANALYSIS_FT_SUMM_PT_LN_D_MS;

SQL> create synonym T_ANALYSIS_FEAT_MS_LIST_UNQ for
reportsv3.T_ANALYSIS_FEAT_MS_LIST_UNQ;
```

5. Recompiled all affected views, triggers, procedures.

6. Got "OKAY" to drop tables from the old schema.

```
SQL> connect etdmv3
SQL> drop table etdmv3.T_ANALYSIS_REPORTX;
SQL> drop table etdmv3.T_ANALYSIS_REPORT_HISTORYX;
SQL> drop table etdmv3.T_ANALYSIS_REPORT_FEATUREX;
SQL> drop table etdmv3.T_ANALYSIS_REPORT_FEAT_HISTX;
SQL> drop table etdmv3.T_ANALYSIS_FEAT_LIST_UNQX;
SQL> drop table etdmv3.T_ANALYSIS_ALT_LIST_UNQX;
SQL> drop table etdmv3.T_ANALYSIS_RPT_LIST_DATA_MSX;
SQL> drop table etdmv3.T_ANALYSIS_RPT_SUMM_PL_RS_2_MSX;
SQL> drop table etdmv3.T_ANALYSIS_RPT_SUMM_PT_LN_D_MSX;
SQL> drop table etdmv3.T_ANALYSIS_ALT_MS_LIST_UNQX;
SQL> drop table etdmv3.T_ANALYSIS_FT_LIST_DATA_MSX;
SQL> drop table etdmv3.T_ANALYSIS_FT_SUMM_PL_RS_2_MSX;
SQL> drop table etdmv3.T_ANALYSIS_FT_SUMM_PT_LN_D_MSX;
SQL> drop table etdmv3.T_ANALYSIS_FEAT_MS_LIST_UNQX;
```

A.5.3 Populate t_notification_log with Missing Records

Following a major edit to the milestone.notify_etat package procedure (Apr 2011), it was discovered that ETAT Review notifications had not been logged properly for several projects (7963, 13143, 13168, 13228, 13288). While copies of the notifications were saved to t_email_history, they were not recorded in t_log_notification. Since the quarterly ETAT performance reports are dependent on the data stored in t_notification_log, we needed to populate the table with the missing records. The data were derived from t_email_hisotry and t_org_users and inserted to t_notification_log using the SQL statements below (1). Additionally, it was discovered that some existing log records had not flagged exempt agencies. A separate process, also shown below (2), was used to correct those records. All data were QA/QC'ed by Stephanie Clemons and/or Mike Konikoff before final updates were made to the production database.

1. Populating t_notification_log with records for projects 963, 13143, 13168, 13228, 13288:

```
----------------------------------------------------

/******* v_recip_for_draft_hist_notices *******/
create or replace view v_recip_for_draft_hist_notices as
SELECT 1 as fk_notification_type,
  v.fk_org_user,
  ou.fk_org,
  v.fk_project
FROM v_etat_email_prj v,
     t_org_users ou
   where v.fk_org_user = ou.pk_org_user
```

```
-----
UNION
-----
SELECT an.fk_notification_type,
  an.fk_org_user,
  an.fk_org,
  an.fk_project
FROM v_an_ou_recipients an
-----
UNION
-----
SELECT e.fk_notification_type,
  an.fk_org_user,
  ou.fk_org,
  e.fk_project
FROM v_an_ou_add an,
  t_org_users ou,
  t_email_draft_history e
WHERE an.fk_email_draft = e.fk_email_draft
and e.time_stamp > '26-Apr-2011'
AND an.fk_org_user      = pk_org_user
/
/******* sent_notices *******/
create table temp_sent_notices
as
select distinct pk_email_history,
       fk_notification_type,
       timestamp,
       replace(regexp_replace(regexp_replace(substr(subject,1,11),'Advance.*'),'Notice.*'),'(ATTN: FTA',61) fk_org,
       to_number(substr(regexp_replace(regexp_replace(subject,'Project Sponsor.*$'),'^.*Project','Project',1,1,'i'),10,5)) fk_project,
       9999 fk_org_user_sender
  from t_email_history
 where pk_email_history
    in (select fk_email_history from t_notification_log)
  and fk_notification_type in (select fk_notification_type from t_email_draft_history)
  and timestamp > '26-Apr-2011'
 order by timestamp
/
update temp_sent_notices set fk_org=121 where fk_org like '%FDOT%';
update temp_sent_notices set fk_org=42 where fk_org like '%FHWA%';
begin for i in (select distinct pk_email_history,fk_project,n.fk_org_user_sender
  from temp_sent_notices s, t_notification_log n
  where pk_email_history = fk_email_history
) loop update temp_sent_notices set fk_org_user_sender = i.fk_org_user_sender where pk_email_history = i.pk_email_history; end loop; end;
/
/******* missing_log_records *******/
create table temp_missing_logs
as
select pk_email_history,
      fk_notification_type,
      timestamp,
      regexp_replace(regexp_replace(substr(subject,1,12),'Advance.*'),'Notice.*') subject,
      replace(regexp_replace(regexp_replace(substr(subject,1,12),'Advance.*'),'Notice.*'),'(ATTN: FTA)',61) fk_org,
      to_number(substr(regexp_replace(regexp_replace(subject,'Project Sponsor.*$'),'^.*Project','Project',1,1,'i'),10,5)) fk_project
  from t_email_history
 where pk_email_history
not in (select fk_email_history from t_notification_log)
  and fk_notification_type in (select fk_notification_type from t_email_draft_history)
  and timestamp > '01-Apr-2011'
 order by timestamp
/
/******* insert_p13143_e17199 *******/
insert into t_notification_log (
  FK_PROJECT_ALT,
  FK_ORG_USER_SENDER,
  FK_ORG_USER_RECEIVER,
  FK_EMAIL_HISTORY,
  FK_NOTIFICATION_TYPE,
  TIME_STAMP,
  EDIT_REVIEW_CYCLE,
  IS_EXEMPT
)
select distinct
  ms.fk_project_alt,
  sent.fk_org_user_sender,
  recip.fk_org_user,
  tolog.pk_email_history,
  tolog.fk_notification_type,
  tolog.timestamp,
  ms.erc,
  'N'
from
  (select fk_project, fk_project_alt, max(edit_review_cycle) erc
  from t_milestones where fk_milestone_type = 1
  and fk_project in (select fk_project from temp_sent_notices)
  group by fk_project, fk_project_alt) ms,
  temp_sent_notices sent,
  temp_missing_logs tolog,
  v_recip_for_draft_hist_notices recip
where
  ms.fk_project = tolog.fk_project
  and tolog.fk_project = sent.fk_project
  and recip.fk_notification_type = tolog.fk_notification_type
  and recip.fk_project = tolog.fk_project
  and (recip.fk_project, recip.fk_org, recip.fk_notification_type)
     not in (select fk_project, fk_org, fk_notification_type from temp_sent_notices)
  ----------------------------------
  and tolog.fk_project = 13148
  and tolog.fk_notification_type = 1 --23
  ------------------------------------
  --and recip.fk_org not in (121)
  --and recip.fk_org not in (61,42) and tolog.fk_org is null
  --and recip.fk_org in (61) and tolog.fk_org = 61
/
/******* insert_p13168_e17337 *******/
insert into t_notification_log (
  FK_PROJECT_ALT,
  FK_ORG_USER_SENDER,
  FK_ORG_USER_RECEIVER,
  FK_EMAIL_HISTORY,
  FK_NOTIFICATION_TYPE,
  TIME_STAMP,
  EDIT_REVIEW_CYCLE,
  IS_EXEMPT
)
select distinct
  ms.fk_project_alt,
  sent.fk_org_user_sender,
  sent.fk_org_user_sender,
  tolog.pk_email_history,
  tolog.fk_notification_type,
  tolog.timestamp,
  ms.erc,
  'Y'
from
  (select fk_project, fk_project_alt, max(edit_review_cycle) erc
  from t_milestones where fk_milestone_type = 1
  and fk_project in (select fk_project from temp_sent_notices)
  group by fk_project, fk_project_alt) ms,
  temp_sent_notices sent,
  temp_missing_logs tolog
where
  ms.fk_project = tolog.fk_project
  and tolog.fk_project = sent.fk_project
  ----------------------------------
  and tolog.fk_project = 13168
  and tolog.fk_notification_type = 25
```

```
/
/******* insert_p13168_e17338 *******/
insert into t_notification_log (
  FK_PROJECT_ALT,
  FK_ORG_USER_SENDER,
  FK_ORG_USER_RECEIVER,
  FK_EMAIL_HISTORY,
  FK_NOTIFICATION_TYPE,
  TIME_STAMP,
  EDIT_REVIEW_CYCLE,
  IS_EXEMPT
)
select distinct
  ms.fk_project_alt,
  sent.fk_org_user_sender,
  recip.fk_org_user,
  tolog.pk_email_history,
  tolog.fk_notification_type,
  tolog.timestamp,
  ms.erc,
  'N'
from
  (select fk_project, fk_project_alt, max(edit_review_cycle) erc
  from t_milestones where fk_milestone_type = 1
  and fk_project in (select fk_project from temp_sent_notices)
  group by fk_project, fk_project_alt) ms,
  temp_sent_notices sent,
  temp_missing_logs tolog,
  v_recip_for_draft_hist_notices recip
where
  ms.fk_project = tolog.fk_project
  and tolog.fk_project = sent.fk_project
  and recip.fk_notification_type = tolog.fk_notification_type
  and recip.fk_project = tolog.fk_project
  and (recip.fk_project, recip.fk_org, recip.fk_notification_type)
```

```
    not in (select fk_project, fk_org, fk_notification_type from temp_sent_notices)
  -----------------------------------
  and tolog.fk_project = 13168
  and tolog.fk_notification_type = 23
  -------------------------------------
  --and recip.fk_org not in (121)
  and recip.fk_org not in (61) and tolog.fk_org is null
  --and recip.fk_org in (61) and tolog.fk_org = 61
/
/******* insert_p13168_e17340 *******/
insert into t_notification_log (
  FK_PROJECT_ALT,
  FK_ORG_USER_SENDER,
  FK_ORG_USER_RECEIVER,
  FK_EMAIL_HISTORY,
  FK_NOTIFICATION_TYPE,
  TIME_STAMP,
  EDIT_REVIEW_CYCLE,
  IS_EXEMPT
)
select distinct
  ms.fk_project_alt,
  sent.fk_org_user_sender,
  recip.fk_org_user,
  tolog.pk_email_history,
  tolog.fk_notification_type,
  tolog.timestamp,
  ms.erc,
  'N'
from
  (select fk_project, fk_project_alt, max(edit_review_cycle) erc
  from t_milestones where fk_milestone_type = 1
  and fk_project in (select fk_project from temp_sent_notices)
  group by fk_project, fk_project_alt) ms,
  temp_sent_notices sent,
  temp_missing_logs tolog,
  v_recip_for_draft_hist_notices recip
where
  ms.fk_project = tolog.fk_project
  and tolog.fk_project = sent.fk_project
  and recip.fk_notification_type = tolog.fk_notification_type
  and recip.fk_project = tolog.fk_project
  and (recip.fk_project, recip.fk_org, recip.fk_notification_type)
     not in (select fk_project, fk_org, fk_notification_type from temp_sent_notices)
  ----------------------------------
  and tolog.fk_project = 13168
  and tolog.fk_notification_type = 23
  ------------------------------------
  --and recip.fk_org not in (121)
  --and recip.fk_org not in (61) and tolog.fk_org is null
  and recip.fk_org in (61) and tolog.fk_org = 61
/
/******* insert_p13228_e17178 *******/
insert into t_notification_log (
  FK_PROJECT_ALT,
  FK_ORG_USER_SENDER,
  FK_ORG_USER_RECEIVER,
  FK_EMAIL_HISTORY,
  FK_NOTIFICATION_TYPE,
  TIME_STAMP,
  EDIT_REVIEW_CYCLE,
  IS_EXEMPT
)
select distinct
  ms.fk_project_alt,
  sent.fk_org_user_sender,
  recip.fk_org_user,
  tolog.pk_email_history,
  tolog.fk_notification_type,
  tolog.timestamp,
  ms.erc,
  'N'
from
  (select fk_project, fk_project_alt, max(edit_review_cycle) erc
  from t_milestones where fk_milestone_type = 1
  and fk_project in (select fk_project from temp_sent_notices)
  group by fk_project, fk_project_alt) ms,
  temp_sent_notices sent,
  temp_missing_logs tolog,
  v_recip_for_draft_hist_notices recip
where
  ms.fk_project = tolog.fk_project
  and tolog.fk_project = sent.fk_project
  and recip.fk_notification_type = tolog.fk_notification_type
  and recip.fk_project = tolog.fk_project
  and (recip.fk_project, recip.fk_org, recip.fk_notification_type)
     not in (select fk_project, fk_org, fk_notification_type from temp_sent_notices)
  ----------------------------------
  and tolog.fk_project = 13228
  and tolog.fk_notification_type = 23
  ------------------------------------
  --and recip.fk_org not in (61,42) and tolog.fk_org is null
  --and recip.fk_org in (61) and tolog.fk_org = 61
/
/******* insert_p13228_e17180 *******/
insert into t_notification_log values(null,7766,2642,2642,17180,25,'10-MAY-11 17:49:23',null,1,'Y')
/
/******* insert_p13288_e17302 *******/
insert into t_notification_log (
  FK_PROJECT_ALT,
  FK_ORG_USER_SENDER,
  FK_ORG_USER_RECEIVER,
  FK_EMAIL_HISTORY,
  FK_NOTIFICATION_TYPE,
  TIME_STAMP,
  EDIT_REVIEW_CYCLE,
  IS_EXEMPT
)
select distinct
  ms.fk_project_alt,
  sent.fk_org_user_sender,
  recip.fk_org_user,
  tolog.pk_email_history,
  tolog.fk_notification_type,
  tolog.timestamp,
  ms.erc,
  'N'
from
  (select fk_project, fk_project_alt, max(edit_review_cycle) erc
  from t_milestones where fk_milestone_type = 1
  and fk_project in (select fk_project from temp_sent_notices)
  group by fk_project, fk_project_alt) ms,
  temp_sent_notices sent,
  temp_missing_logs tolog,
  v_recip_for_draft_hist_notices recip
where
  ms.fk_project = tolog.fk_project
  and tolog.fk_project = sent.fk_project
  and recip.fk_notification_type = tolog.fk_notification_type
  and recip.fk_project = tolog.fk_project
  and (recip.fk_project, recip.fk_org, recip.fk_notification_type)
     not in (select fk_project, fk_org, fk_notification_type from temp_sent_notices)
  ----------------------------------

  and tolog.fk_project = 13288
  and tolog.fk_notification_type = 22 --23
  ------------------------------------
  --and recip.fk_org not in (121)
  --and recip.fk_org not in (61,42) and tolog.fk_org is null
  --and recip.fk_org in (61) and tolog.fk_org = 61
/
```

```
/******* insert_p13288_e17303 *******/
insert into t_notification_log (
  FK_PROJECT_ALT,
  FK_ORG_USER_SENDER,
  FK_ORG_USER_RECEIVER,
  FK_EMAIL_HISTORY,
  FK_NOTIFICATION_TYPE,
  TIME_STAMP,
  EDIT_REVIEW_CYCLE,
  IS_EXEMPT
)
select distinct
  ms.fk_project_alt,
  sent.fk_org_user_sender,
  recip.fk_org_user,
  tolog.pk_email_history,
  tolog.fk_notification_type,
  tolog.timestamp,
  ms.erc,
  'N'
from
  (select fk_project, fk_project_alt, max(edit_review_cycle) erc
  from t_milestones where fk_milestone_type = 1
  and fk_project in (select fk_project from temp_sent_notices)
  group by fk_project, fk_project_alt) ms,
  temp_sent_notices sent,
  temp_missing_logs tolog,
  v_recip_for_draft_hist_notices recip
where
  ms.fk_project = tolog.fk_project
  and tolog.fk_project = sent.fk_project
  and recip.fk_notification_type = tolog.fk_notification_type
  and recip.fk_project = tolog.fk_project
  and (recip.fk_project, recip.fk_org, recip.fk_notification_type)
     not in (select fk_project, fk_org, fk_notification_type from temp_sent_notices)
  ---------------------------------
  and tolog.fk_project = 13288
  and tolog.fk_notification_type = 24
  -----------------------------------
  --and recip.fk_org not in (121)
  and recip.fk_org not in (61) and tolog.fk_org is null
  --and recip.fk_org in (61) and tolog.fk_org = 61
/
/******* insert_p13288_e17305 *******/
insert into t_notification_log (
  FK_PROJECT_ALT,
  FK_ORG_USER_SENDER,
  FK_ORG_USER_RECEIVER,
  FK_EMAIL_HISTORY,
  FK_NOTIFICATION_TYPE,
  TIME_STAMP,
  EDIT_REVIEW_CYCLE,
  IS_EXEMPT
)
select distinct
  ms.fk_project_alt,
  sent.fk_org_user_sender,
  recip.fk_org_user,
  tolog.pk_email_history,
  tolog.fk_notification_type,
  tolog.timestamp,
  ms.erc,
  'N'
from
  (select fk_project, fk_project_alt, max(edit_review_cycle) erc
  from t_milestones where fk_milestone_type = 1
  and fk_project in (select fk_project from temp_sent_notices)
  group by fk_project, fk_project_alt) ms,
  temp_sent_notices sent,
  temp_missing_logs tolog,
      v_recip_for_draft_hist_notices recip
    where
      ms.fk_project = tolog.fk_project
      and tolog.fk_project = sent.fk_project
      and recip.fk_notification_type = tolog.fk_notification_type
      and recip.fk_project = tolog.fk_project
      and (recip.fk_project, recip.fk_org, recip.fk_notification_type)
        not in (select fk_project, fk_org, fk_notification_type from temp_sent_notices)
      ----------------------------------
      and tolog.fk_project = 13288
      and tolog.fk_notification_type = 24
      ------------------------------------
      --and recip.fk_org not in (121)
      --and recip.fk_org not in (61,42) and tolog.fk_org is null
      and recip.fk_org in (61) and tolog.fk_org = 61
    /
    /******* insert_p7963_e17140 *******/
    insert into t_notification_log values(null,1923,429,429,17140,25,'26-APR-11 11:03:43',null,2,'Y')
    /
/******* insert_p7963_e17141 *******/
insert into t_notification_log (
  FK_PROJECT_ALT,
  FK_ORG_USER_SENDER,
  FK_ORG_USER_RECEIVER,
  FK_EMAIL_HISTORY,
  FK_NOTIFICATION_TYPE,
  TIME_STAMP,
  EDIT_REVIEW_CYCLE,
  IS_EXEMPT
)
select distinct
  ms.fk_project_alt,
  sent.fk_org_user_sender,
  recip.fk_org_user,
  tolog.pk_email_history,
  tolog.fk_notification_type,
  tolog.timestamp,
  ms.erc,
  'N'
from
  (select fk_project, fk_project_alt, max(edit_review_cycle) erc
  from t_milestones where fk_milestone_type = 1
  and fk_project in (select fk_project from temp_sent_notices)
  group by fk_project, fk_project_alt) ms,
  temp_sent_notices sent,
  temp_missing_logs tolog,
  v_recip_for_draft_hist_notices recip
where
  ms.fk_project = tolog.fk_project
  and tolog.fk_project = sent.fk_project
  and recip.fk_notification_type = tolog.fk_notification_type
  and recip.fk_project = tolog.fk_project
  ----------------------------------
  and tolog.fk_project = 7963
  and tolog.fk_notification_type = 23
  ------------------------------------
  and recip.fk_org not in (61,42) and tolog.fk_org is null
  --and recip.fk_org in (61) and tolog.fk_org = 61
/
/******* insert_p7963_e17143 *******/
insert into t_notification_log (
  FK_PROJECT_ALT,
  FK_ORG_USER_SENDER,
  FK_ORG_USER_RECEIVER,
  FK_EMAIL_HISTORY,
  FK_NOTIFICATION_TYPE,
  TIME_STAMP,
  EDIT_REVIEW_CYCLE,
  IS_EXEMPT
)
```

```
select distinct
  ms.fk_project_alt,
  sent.fk_org_user_sender,
  recip.fk_org_user,
  tolog.pk_email_history,
  tolog.fk_notification_type,
  tolog.timestamp,
  ms.erc,
  'N'
from
  (select fk_project, fk_project_alt, max(edit_review_cycle) erc
  from t_milestones where fk_milestone_type = 1
  and fk_project in (select fk_project from temp_sent_notices)
  group by fk_project, fk_project_alt) ms,
  temp_sent_notices sent,
  temp_missing_logs tolog,
  v_recip_for_draft_hist_notices recip
where
  ms.fk_project = tolog.fk_project
  and tolog.fk_project = sent.fk_project
  and recip.fk_notification_type = tolog.fk_notification_type
  and recip.fk_project = tolog.fk_project
  ----------------------------------
  and tolog.fk_project = 7963
  and tolog.fk_notification_type = 23
  ------------------------------------
  --and recip.fk_org not in (61,42) and tolog.fk_org is null
  and recip.fk_org in (61) and tolog.fk_org = 61
/
-------------------------------------------------------------------
```

2. Setting is_exempt flag in t_notification_log where needed:

```
create table temp_update_is_exempt2
as
select distinct fk_project, log.fk_project_alt, fk_email_history,
fk_org_user_receiver, log.is_exempt, decode(log.is_exempt,'Y','N','N','Y')
change_to
from
(
select fk_project,fk_project_alt,trunc(nl.time_stamp) time_stamp, is_exempt
from t_notification_log nl, t_project_alt a
where fk_project_alt = pk_project_alt
and fk_org_user_receiver in (select pk_org_user from t_org_users where
fk_org=61)
and (fk_project,trunc(nl.time_stamp)) in (select fk_project,trunc(time_stamp)
from
t_milestones where fk_milestone_type = 1)
and trunc(nl.time_stamp) between '01-Jul-2010' and '12-Apr-2011'
minus
select ms.fk_project,ms.fk_project_alt,trunc(ms.time_stamp)
time_stamp,decode(max(mode_transit),'Y','N','Y') is_exempt
from t_milestones ms, t_project_alt_milestones ams
where pk_milestone = fk_milestone
group by ms.fk_project,ms.fk_project_alt,ms.time_stamp
order by time_stamp,fk_project
) fix,
t_notification_log log
where fix.is_exempt=log.is_exempt
and log.is_exempt='N'
and trunc(log.time_stamp) = fix.time_stamp
and fix.fk_project_alt = log.fk_project_alt
and log.fk_org_user_receiver in (select pk_org_user from t_org_users where
fk_org=61)
;
begin
for i in (select * from temp_update_is_exempt2)
loop
  update t_notification_log set is_exempt = i.change_to
   where fk_project_alt = i.fk_project_alt
     and fk_email_history = i.fk_email_history
```

```
      and fk_org_user_receiver = i.fk_org_user_receiver
      and is_exempt = 'N';
end loop;
end;
```

A.5.4 Add FK_ORG_USER to T_PROJECT_CONTACT and Populate

Code used to populate new fk_org_user field in t_project_contact is shown below.

```
--------- 0000.sql
alter table t_project_contact add (
fk_org_user number(10),
constraint t_project_contact_fkou foreign key (fk_org_user) references t_org_users(pk_org_user)
);
--------- 0001.sql
create or replace view temp_project_contact_ou
as
  select distinct max(pk_org_user) fk_org_user,email,fk_project as pk_project,
         development_team,first_name,last_name,
         least(enabled_for_auth,enabled_for_region,enabled_for_org) as enabled,
         read_only
    from v_users            u,
         t_users            tu,
         t_projects         prj,
         at_regions_projects reg
   where   u.fk_auth_role = 17 /*Project Manager*/
           --and read_only='N'
           and u.user_id = tu.user_id
           --and enabled_for_org='Y'
           --and enabled_for_region='Y'
           --and enabled_for_auth='Y'
           --and development_team='N'
     and u.fk_org = prj.fk_org
     and u.fk_region = reg.fk_region
     and reg.fk_project = prj.pk_project
   group by email,fk_project,development_team,
         least(enabled_for_auth,enabled_for_region,enabled_for_org),
      first_name,last_name,read_only
/
--------- 0002.sql
alter table t_project_contact disable all triggers;
begin
for i in (select  * from temp_project_contact_ou where development_team='N' and read_only='N' and
enabled='Y')
loop
  update t_project_contact set fk_org_user = i.fk_org_user
  where fk_org_user is null
  and fk_project = i.pk_project
  --and lower(email) = lower(i.email);
  and ((name like '%'||i.first_name||'%'||i.last_name||'%' or name like
'%'||i.last_name||'%'||i.first_name||'%')
      and
      length(i.first_name||i.last_name) > 2
     );
end loop;
end;
/
begin
for i in (select  * from temp_project_contact_ou order by read_only, enabled desc, development_team)
loop
  update t_project_contact set fk_org_user = i.fk_org_user
  where fk_org_user is null
  and fk_project = i.pk_project
  and (
     --lower(email) = lower(i.email)
     --or
       (
        (name like '%'||i.first_name||'%'||i.last_name||'%' or name like
'%'||i.last_name||'%'||i.first_name||'%')
        and
        length(i.first_name||i.last_name) > 2
       )
     )
```

```
   ;
end loop;
end;
/
alter table t_project_contact enable all triggers;
--------- 0003.sql
alter table t_project_contact add (constraint t_project_contact_unq_prjou unique
(fk_project,fk_org_user));
alter table t_project_contact modify (fk_org_user not null);
```

etdm
Efficient Transportation Decision Making

Design Stage

Draft User Manual

The updated User Handbook for the Environmental Screening Tool (EST) is provided in **Appendix E** of this ISDM submittal.

Draft Test Plan

Table of Contents

This section describes the approach used for acceptance testing enhancements to the Environmental Screening Tool. Testing will occur in several steps to resolve any coding errors and to ensure that the new application meets user requirements.

Chapter 1 Acceptance Criteria

Testing will result in a system that meets or exceeds the following acceptance criteria:

- Prior to moving to production, the code will be 98-100% free of critical errors. All identified code errors that effect user performance will have been resolved. Minor errors such as formatting may be postponed upon the discretion of the client's technical project manager. Any errors which are subsequently identified by users will be corrected immediately.
- Individual programs include the following minimum documentation:
  - Comment header with a title, history (creation and modification dates), author, and short description
  - In-line comments at each logical block briefly describing functionality that is occurring within that block
- Ideally, documentation will also include variable definitions and identification of other programs which call and are called by the program.
- For programs that are being converted from existing production programs, the new program will include all logical consistency, security, and error checking that occur in the existing program. Converted programs will at least support existing functionality and incorporate enhancements specified in the GUI design. Optimally, known outstanding enhancements, will also be incorporated during conversion.
- Programs meet approved GUI design specifications.
- Minimally, program functionality meets the defined user requirements. Ideally, functionality delights the users and makes their tasks easier.

Chapter 2 Procedures

Each program will be tested individually and in common work processes to ensure the system meets the acceptance criteria specified above. Testing procedures are:

1. Prior to loading programs to the Environmental Screening Tool Development server, programmers will test and correct their own work.
2. On the Development server, the programmer will review and test the code. This includes testing for code errors as well as a review of the code for in-line documentation and efficient programming techniques. Whenever possible, converted programs will be reviewed and tested by the author of the original code, if different from the current author. The program author will correct any errors and omissions and re-load to the Development server. After errors are correct, the code is migrated to the Stage platform.
3. On the Stage platform, a designated peer reviewer will review and test the program functionality to ensure it meets user requirements and GUI specifications. This reviewer will be a member of the development, different than the program author. Errors will be reported in the bug tracking software used by the development team. The program author will correct any errors and omissions and re-load to the Development and Stage servers. Two types of testing will occur:

1. Testing specific program modifications – using a test account with appropriate role assignments, the reviewer will test programs which have been specifically modified
2. Testing program integration – using multiple test accounts, the reviewer will go through the project life cycle testing critical functions used at various project milestones to ensure that the new components of the application do not effect other sections

4. Pre-selected users will perform their tasks on the application to ensure the programs are working properly. They will report their findings to the Environmental Screening Tool Help Desk via email. Program authors will resolve any errors and omissions prior to moving the application to Production.
5. Upon approval of the client’s technical project manager, the application will move to production.

Chapter 3 Error Reporting

The designated users who test the application will report errors and omissions to the ETDM helpdesk via email. The following information will be requested for each error:

- Project # (if applicable)
- Page Title or File Name
- Description of Steps Leading to Error
- Error Type (Crash, hang, privileges, etc.)
- Error message

Testers will submit results to the Help Desk email address. Tasks will be assigned by the application development project manager to the appropriate programmer for resolution.

Updated Project Schedule

Environmental Screening Tool Implementation Schedule

Time Line	Action
January 2000 – December 2001	Participate in Agency meetings to determine technology requirements as process is defined
September 2000	Interagency Technology Work Group develops strategy for technology
January 2001	Demonstrate Initial Prototype of GIS application to Agency Working Group
July 2001	Implementation Plan and general requirements document complete
Fall 2001	● Conduct Interagency GIS Workshops to refine data requirements ● Demonstrate refined prototype at FDOT Environmental Management conference
January 2002 – May 2002	Continue meeting with focus groups, task work groups and steering committee to refine prototype as general operating procedures are developed for the ETDM Process
June 2002	Conduct Mock Environmental Technical Advisory Team (ETAT) meeting to test EST and ETDM Process
July 2002 – February 2003	Revise prototype based on additional feedback on process as specific operating procedures are developed
March 2003	● Begin Staged Implementation of ETDM Process ● EST will be released, but additional requirements and enhancements are anticipated as the ETDM Process is used for the first time
June 2003 – December 2003	Respond to priority enhancements identified during training
January 2004 – December 2004	Respond to enhancements identified during 1st year of ETDM implementation
October 2004 - October 2005	Convert existing site to new integrated interface
October - November 2005	Testing Stage ● Converted site moved to STAGE server ● Beta Testing with Target Users ● User handbook is completed ● Test team provide input to dev team ● Development Team corrects errors
November 2005	Implementation Stage ● User Training Conducted
December 2005	Converted site moved to Production server
January 2006	Begin Maintenance Stage ● Respond to priority enhancements identified from Task Work Groups and Steering Committee ● Respond to outstanding user enhancement requests
Program corrections and minor enhancements from this point forward are tracked in Bugzilla, the EST task management application. Milestones of major enhancements are listed below.	
February 2006	New Public Site requirements complete
March 2006	● Public Site – Design Stage

Time Line	Action
	- On-line Invoicing – Requirements and Design Complete - Advance Notification/Federal Consistency – Requirements Complete
April 2006	- Public Site – Development Stage Began - Integrated Map Viewer – Design - On-line Invoicing – Development Stage Began - Advance Notification/Federal Consistency – Development
August 2006	Summary of Public Comment Report – Requirements/Design
September 2006	- Summary of Public Comment Report – Development - Integrated Map Viewer Development Began - Performance Management System – Requirements
October 2006	- Public Site – Testing - Summary of Public Comment Report – Internal Testing - Performance Management System – Design - On-line Invoicing – Testing
October 31, 2006	Public Site Deployed to Production
November 2006	- ETDM Coordinator and CLC Training on Public Site - Performance Management System – Development Began
December 2006	- Cumulative Effects Prototype Requirements complete - On-line Invoicing – Production for historic data migration
January 2007	Cumulative Effects Prototype Development Began
February 2007	EDMS/EST Integration – Requirements began
April 2007	Advance Notification/Federal Consistency – Testing
May 2007	- Advance Notification – Production - Summary of SCE Comments Report – Requirements/Design/Development - On-Line Invoicing – Initial Agency Training - EDMS/EST Integration – Design/Development
June 2007	- Summary of Public Comment Report – Production - Summary of SCE Comments Report – Testing - Performance Management System – Testing Began
July 2007	Cumulative Effects Prototype Testing
August 2007	- SCE Map Tools – Design/Development - EDMS/EST Integration – Testing of document transfer complete
September 2007	- Integrated Map Viewer – Testing - EDMS/EST Integration – Production - SCE Ad Hoc Report – Development

Time Line	Action
October 2007	- SCE Map Tools – Testing (part of new Map Viewer) - SCE Ad Hoc Report – Testing
November 2007	- Summary of SCE Comments Report – Production - SCE Ad Hoc Report – Production
January 2008	- Performance Management System – Production - Invoicing System Enhancements – Design/Development
February 2008	- CCI Enhancements – Design/Development - Security Enhancements – Testing
March 2008	Security Enhancements – Production
May 2008	- AN/Federal Consistency – Testing - CCI Enhancements – Testing - Invoicing System Enhancements – Testing - Quality Assurance Reports – Design/Development
June 2008	- AN/Federal Consistency – Production - Invoicing Enhancements – Production
August 2008	- CCI Enhancements – Production - Integrated Map Viewer – Production - Quality Assurance Reports – Testing
September 2008	Quality Assurance Reports – Production
October 2008	- Project Tracker v 1 – Requirements/Design - Invoicing – Offline Activity Log –Training - Project Schedule Enhancements – Requirements/Design
November 2008	- Project Tracker v 1 – Design/Development - Project Schedule Enhancements – Development
December 2008	- Project Tracker v 1 – Testing - Project Tracker v 2 – Requirements/Design
January 2009	- Project Tracker v 2 – Development - Project Schedule Enhancements – Testing
February 2009	Project Schedule Enhancements – Production
March 2009	Document Review – Requirements
April 2009	Document Review – Design/Development
May 2009	Project Tracker – Training/Production
June 2009	- Project Schedule – Training /Production - Document Review – Testing

Time Line	Action
July 2009	SCE Participation Report – Requirements
August 2009	SCE Participation Report – Design
September 2009	● 2010 ETDM Surveys – Planning/Requirements ● Performance Management Report Enhancements – Planning
October 2009	● 2010 ETDM Surveys – Development ● Performance Management Report Enhancements – Requirements ● SCE Participation Report – Development
November 2009	Document Review – Production
December 2009	● 2010 ETDM Surveys – Testing ● SCE Participation Report – Testing
January 2010	● 2010 ETDM Surveys – Production ● Performance Management Report Enhancements – Design ● AN Package Simplification Enhancements – Planning ● Map Viewer/Editor Simplification Enhancements – Planning
February 2010	Performance Management Report Enhancements – Development/Testing
March 2010	● Performance Management Report Enhancements – Production ● AN Package Simplification Enhancements – Requirements ● Map Viewer/Editor Simplification Enhancements – Requirements ● SCE Participation Report – Production
April 2010	● AN Package Simplification Enhancements – Design and Development ● Site Search – Production
May 2010	Create AN Package Simplification – Testing and Production
June 2010	AN Transmittal List Simplification – Design and Development
July 2010	AN Transmittal List Simplification – Development and Testing
August 2010	● EST Menu Simplification – Production ● AN Transmittal List Simplification – Production ● Calendar Simplification Enhancements – Requirements
September 2010	● Map Viewer/Editor Simplification – Design ● Calendar Simplification Enhancements – Design ● GIS Analysis Results Report (Feature Level Analysis Results) – Requirements
October 2010	● GIS Analysis Results Report (Feature Level Analysis Results) – Development ● Calendar Simplification Enhancements – Development
November 2010	Map Viewer/Editor Simplification – Development
December 2010	Local Agency Program Enhancements – Production

Time Line	Action
January 2011	Calendar Simplification Enhancements – Testing
February 2011	Map Viewer Simplification – Development
March 2011	Calendar Simplification – Testing
April 2011	● Calendar Simplification – Production ● GIS Analysis Results Report (Feature Level Analysis Results) - Production ● Map Viewer Simplification – Testing
May 2011	Map Viewer Simplification – Production
June 2011	Map Editor Simplification – Development
July 2011	Map Editor Simplification – Testing
August 2011	● Map Editor Simplification – Production ● Project Tracker Simplification – Requirements

Project Review Issue List

Information Systems Development Methodology
Project Review Issue List

Project: Environmental Screening Tool
Meeting Date: ____________________
Moderator: Peter McGilvray
Recorder: ____________________
Document Name: Updated Design Document
Version: 2007_1130
Directory Location: ____________________

Review Type:
☑ Inspection ❐ Re-inspection ❐ Peer Review ❐ Re-Review
❐ Requirements Document ☑ Design Document ❐ Code ❐ Other

Preparation Time: Date ____________ Time (hours) ____________ Date ____________ Time (hours) ____________
____________ ____________ ____________ ____________

Location	Issue Description	Issue		
		Type	Class	Severity

Issue Type: (Use issue types as shown for category of item being inspected)
Documentation *(Requirements Doc., Design Doc., Code)* - CS=Consistency, CT=Content, DN=Definition, HF=Human Factors, OR=Organization, RD=Readability, SN=Syntax, ST=Standards, OT=Other

Architecture or code (*Design, Code)* - DA=Data, DC=Documentation, FN=Functionality, HF=Human Factors, IF=Interface, IO=Input/Output, LO=Logic, MN=Maintainability, PF=Performance, SN=Syntax, ST=Standards, OT=other

Test Plans - DA=Data, DC=Documentation, FN=Functionality, HF=Human Factors, IF=Interface, IO=Input/Output, LO=Logic, MN=Maintainability, PF=Performance, SN=Syntax, ST=Standards, TC=Test Case, TE=Test Environment, TP=Test Plan, OT=other
Issue Class: M=Missing, W=Wrong, E=Extra, A=Ambiguous, I=Inconsistent
Issue Severity: J=Major, N=Minor

Project Review Issue List Description

Project - Name of project
Moderator - Name of moderator
Project Review Item - Item being reviewed
Meeting Date - Date of review
Recorder - Recorder's name
Version - Version number of the review items, if applicable
Preparation Log - Used to track time spent on review's preparation task (nearest 1/2 hour)
Location - Location of the issue (paragraph, section, line, module etc.)
Issue Description - Brief description of the issue
Issue Type –

Type	Description
(CS) – consistency	Inconsistent specification either within the document or with other planning documents
(CT) – content	Inadequate, incorrect, or unnecessary information
(DA) – data	Issues in data specification; improper declaration, initialization, or description of data; incorrect data usage, conversion of data types, or array boundaries
(DC) – documentation	Inadequate or incorrect component descriptions
(DN) – definition	Missing, wrong, or extra definition of terminology
(FN) – functionality	Issues in the specification of the functions of a component
(HF) - human factors	Poor or lacking regard to human factors; unnecessary operator involvement
(IF) - interface	Issues in the communication between software components
(IO) - input/output	Issues in communication with or specification of external data or devices
(LO) - logic	Issues in procedures or in sequence, selection, iteration of operations; incorrect algorithms or mathematical computation
(MN) - maintainability	An expectation that the work product is difficult to maintain, excluding issues in documentation
(OR) - organization	Awkward or noncohesive presentation of information
(PF) - performance	An expectation of not meeting the required execution efficiency
(RD) - readability	Difficult to understand; inappropriate language, syntax, word use or notation
(SN) - syntax	Issues in grammar, punctuation, spelling, and specification language usage
(ST) - standards	A deviation from procedural or representational standards
(TC) - test case	Incomplete or inaccurate specifications of a test condition, or a deviation from the test plan
(TE) - test environment	Issues in the definition or specification of the test hardware or software environment, level of security, or proprietary components
(TP) - test plan	Issues in the definition or specification of test scope, strategy (including test completeness and issue tolerance levels), personnel, tasks, items, or features
(OT) - other	An undefined or ambiguous issue condition

Issue Class - (M)issing, (W)rong, (E)xtra, (A)mbiguous, (I)nconsistent
Issue Severity - ma(J)or: Issues that would result in failure of the item or an observable departure from specifications.
mi(N)or: Issues that would affect only the nonfunctional aspects of the item.

Information Systems Development Methodology
Project Review Issue List

Project: Environmental Screening Tool Meeting Date: ____________

Moderator: Peter McGilvray Recorder: ____________

Document Name: Updated Design Document Version: 2009_0731

Directory Location: ____________

Review Type: ☑ Inspection □ Re-inspection □ Peer Review □ Re-Review
□ Requirements Document ☑ Design Document □ Code □ Other

Preparation Time: Date ________ Time (hours) ________ Date ________ Time (hours) ________
________ ________ ________ ________

Location	Issue Description	Issue		
		Type	Class	Severity

Issue Type: (Use issue types as shown for category of item being inspected)
Documentation *(Requirements Doc., Design Doc., Code)* - CS=Consistency, CT=Content, DN=Definition, HF=Human Factors, OR=Organization, RD=Readability, SN=Syntax, ST=Standards, OT=Other

Architecture or code (*Design, Code)* - DA=Data, DC=Documentation, FN=Functionality, HF=Human Factors, IF=Interface, IO=Input/Output, LO=Logic, MN=Maintainability, PF=Performance, SN=Syntax, ST=Standards, OT=other

Test Plans - DA=Data, DC=Documentation, FN=Functionality, HF=Human Factors, IF=Interface, IO=Input/Output, LO=Logic, MN=Maintainability, PF=Performance, SN=Syntax, ST=Standards, TC=Test Case, TE=Test Environment, TP=Test Plan, OT=other
Issue Class: M=Missing, W=Wrong, E=Extra, A=Ambiguous, I=Inconsistent
Issue Severity: J=Major, N=Minor

Project Review Issue List Description

Project - Name of project
Moderator - Name of moderator
Project Review Item - Item being reviewed
Meeting Date - Date of review
Recorder - Recorder's name
Version - Version number of the review items, if applicable
Preparation Log - Used to track time spent on review's preparation task (nearest 1/2 hour)
Location - Location of the issue (paragraph, section, line, module etc.)
Issue Description - Brief description of the issue
Issue Type –

Type	Description
(CS) – consistency	Inconsistent specification either within the document or with other planning documents
(CT) – content	Inadequate, incorrect, or unnecessary information
(DA) – data	Issues in data specification; improper declaration, initialization, or description of data; incorrect data usage, conversion of data types, or array boundaries
(DC) – documentation	Inadequate or incorrect component descriptions
(DN) – definition	Missing, wrong, or extra definition of terminology
(FN) – functionality	Issues in the specification of the functions of a component
(HF) - human factors	Poor or lacking regard to human factors; unnecessary operator involvement
(IF) - interface	Issues in the communication between software components
(IO) - input/output	Issues in communication with or specification of external data or devices
(LO) - logic	Issues in procedures or in sequence, selection, iteration of operations; incorrect algorithms or mathematical computation
(MN) - maintainability	An expectation that the work product is difficult to maintain, excluding issues in documentation
(OR) - organization	Awkward or noncohesive presentation of information
(PF) - performance	An expectation of not meeting the required execution efficiency
(RD) - readability	Difficult to understand; inappropriate language, syntax, word use or notation
(SN) - syntax	Issues in grammar, punctuation, spelling, and specification language usage
(ST) - standards	A deviation from procedural or representational standards
(TC) - test case	Incomplete or inaccurate specifications of a test condition, or a deviation from the test plan
(TE) - test environment	Issues in the definition or specification of the test hardware or software environment, level of security, or proprietary components
(TP) - test plan	Issues in the definition or specification of test scope, strategy (including test completeness and issue tolerance levels), personnel, tasks, items, or features
(OT) - other	An undefined or ambiguous issue condition

Issue Class - (M)issing, (W)rong, (E)xtra, (A)mbiguous, (I)nconsistent
Issue Severity - ma(J)or: Issues that would result in failure of the item or an observable departure from specifications.
mi(N)or: Issues that would affect only the nonfunctional aspects of the item.

Information Systems Development Methodology
Project Review Issue List

Project: Environmental Screening Tool
Meeting Date: ____________
Moderator: Peter McGilvray
Recorder: ____________
Document Name: Updated Design Document (Attachment A)
Version: 2010_0731
Directory Location: ____________

Review Type: ☑ Inspection □ Re-inspection □ Peer Review □ Re-Review
□ Requirements Document ☑ Design Document □ Code □ Other

Preparation Time: Date ____________ Time (hours) ____________ Date ____________ Time (hours) ____________

Location	Issue Description	Issue		
		Type	Class	Severity

Issue Type: (Use issue types as shown for category of item being inspected)
Documentation *(Requirements Doc., Design Doc., Code)* - CS=Consistency, CT=Content, DN=Definition, HF=Human Factors, OR=Organization, RD=Readability, SN=Syntax, ST=Standards, OT=Other

Architecture or code (*Design, Code)* - DA=Data, DC=Documentation, FN=Functionality, HF=Human Factors, IF=Interface, IO=Input/Output, LO=Logic, MN=Maintainability, PF=Performance, SN=Syntax, ST=Standards, OT=other

Test Plans - DA=Data, DC=Documentation, FN=Functionality, HF=Human Factors, IF=Interface, IO=Input/Output, LO=Logic, MN=Maintainability, PF=Performance, SN=Syntax, ST=Standards, TC=Test Case, TE=Test Environment, TP=Test Plan, OT=other
Issue Class: M=Missing, W=Wrong, E=Extra, A=Ambiguous, I=Inconsistent
Issue Severity: J=Major, N=Minor

Project Review Issue List Description

Project - Name of project
Moderator - Name of moderator
Project Review Item - Item being reviewed
Meeting Date - Date of review
Recorder - Recorder's name
Version - Version number of the review items, if applicable
Preparation Log - Used to track time spent on review's preparation task (nearest 1/2 hour)
Location - Location of the issue (paragraph, section, line, module etc.)
Issue Description - Brief description of the issue
Issue Type –

Type	Description
(CS) – consistency	Inconsistent specification either within the document or with other planning documents
(CT) – content	Inadequate, incorrect, or unnecessary information
(DA) – data	Issues in data specification; improper declaration, initialization, or description of data; incorrect data usage, conversion of data types, or array boundaries
(DC) – documentation	Inadequate or incorrect component descriptions
(DN) – definition	Missing, wrong, or extra definition of terminology
(FN) – functionality	Issues in the specification of the functions of a component
(HF) - human factors	Poor or lacking regard to human factors; unnecessary operator involvement
(IF) - interface	Issues in the communication between software components
(IO) - input/output	Issues in communication with or specification of external data or devices
(LO) - logic	Issues in procedures or in sequence, selection, iteration of operations; incorrect algorithms or mathematical computation
(MN) - maintainability	An expectation that the work product is difficult to maintain, excluding issues in documentation
(OR) - organization	Awkward or noncohesive presentation of information
(PF) - performance	An expectation of not meeting the required execution efficiency
(RD) - readability	Difficult to understand; inappropriate language, syntax, word use or notation
(SN) - syntax	Issues in grammar, punctuation, spelling, and specification language usage
(ST) - standards	A deviation from procedural or representational standards
(TC) - test case	Incomplete or inaccurate specifications of a test condition, or a deviation from the test plan
(TE) - test environment	Issues in the definition or specification of the test hardware or software environment, level of security, or proprietary components
(TP) - test plan	Issues in the definition or specification of test scope, strategy (including test completeness and issue tolerance levels), personnel, tasks, items, or features
(OT) - other	An undefined or ambiguous issue condition

Issue Class - (M)issing, (W)rong, (E)xtra, (A)mbiguous, (I)nconsistent
Issue Severity - ma(J)or: Issues that would result in failure of the item or an observable departure from specifications.
mi(N)or: Issues that would affect only the nonfunctional aspects of the item.

Information Systems Development Methodology
Project Review Issue List

Project: Environmental Screening Tool
Meeting Date:
Moderator: Peter McGilvray
Recorder:
Document Name: Updated Design Document (Attachment A)
Version: 2012_0731
Directory Location:

Review Type: ☑ Inspection □ Re-inspection □ Peer Review □ Re-Review
□ Requirements Document ☑ Design Document □ Code □ Other

Preparation Time: Date Time (hours) Date Time (hours)

Location	Issue Description	Issue		
		Type	Class	Severity

Issue Type: (Use issue types as shown for category of item being inspected)
Documentation *(Requirements Doc., Design Doc., Code)* - CS=Consistency, CT=Content, DN=Definition, HF=Human Factors, OR=Organization, RD=Readability, SN=Syntax, ST=Standards, OT=Other

Architecture or Code (*Design, Code)* - DA=Data, DC=Documentation, FN=Functionality, HF=Human Factors, IF=Interface, IO=Input/Output, LO=Logic, MN=Maintainability, PF=Performance, SN=Syntax, ST=Standards, OT=other

Test Plans - DA=Data, DC=Documentation, FN=Functionality, HF=Human Factors, IF=Interface, IO=Input/Output, LO=Logic, MN=Maintainability, PF=Performance, SN=Syntax, ST=Standards, TC=Test Case, TE=Test Environment, TP=Test Plan, OT=other
Issue Class: M=Missing, W=Wrong, E=Extra, A=Ambiguous, I=Inconsistent
Issue Severity: J=Major, N=Minor

Project Review Issue List Description

Project - Name of project
Moderator - Name of moderator
Project Review Item - Item being reviewed
Meeting Date - Date of review
Recorder - Recorder's name
Version - Version number of the review items, if applicable
Preparation Log - Used to track time spent on review's preparation task (nearest 1/2 hour)
Location - Location of the issue (paragraph, section, line, module, etc.)
Issue Description - Brief description of the issue
Issue Type –

Type	Description
(CS) – consistency	Inconsistent specification either within the document or with other planning documents
(CT) – content	Inadequate, incorrect, or unnecessary information
(DA) – data	Issues in data specification; improper declaration, initialization, or description of data; incorrect data usage, conversion of data types, or array boundaries
(DC) – documentation	Inadequate or incorrect component descriptions
(DN) – definition	Missing, wrong, or extra definition of terminology
(FN) – functionality	Issues in the specification of the functions of a component
(HF) - human factors	Poor or lacking regard to human factors; unnecessary operator involvement
(IF) - interface	Issues in the communication between software components
(IO) - input/output	Issues in communication with or specification of external data or devices
(LO) - logic	Issues in procedures or in sequence, selection, iteration of operations; incorrect algorithms or mathematical computation
(MN) - maintainability	An expectation that the work product is difficult to maintain, excluding issues in documentation
(OR) - organization	Awkward or non-cohesive presentation of information
(PF) - performance	An expectation of not meeting the required execution efficiency
(RD) - readability	Difficult to understand; inappropriate language, syntax, word use or notation
(SN) - syntax	Issues in grammar, punctuation, spelling, and specification language usage
(ST) - standards	A deviation from procedural or representational standards
(TC) - test case	Incomplete or inaccurate specifications of a test condition, or a deviation from the test plan
(TE) - test environment	Issues in the definition or specification of the test hardware or software environment, level of security, or proprietary components
(TP) - test plan	Issues in the definition or specification of test scope, strategy (including test completeness and issue tolerance levels), personnel, tasks, items, or features
(OT) - other	An undefined or ambiguous issue condition

Issue Class - (M)issing, (W)rong, (E)xtra, (A)mbiguous, (I)nconsistent
Issue Severity - ma(J)or: Issues that would result in failure of the item or an observable departure from specifications.
mi(N)or: Issues that would affect only the nonfunctional aspects of the item.

etdm
Efficient Transportation Decision Making

Design Stage

Project Review Management Report

Information Systems Development Methodology
Project Review Management Report

Project:	Environmental Screening Tool	Meeting Date:	
Moderator:	Peter McGilvray	Recorder:	
Document Name:	Updated Design Document	Version:	2007-1130
Directory Location:			

Meeting Type:	☑ Inspection	□ Re-inspection	□ Peer Review	□ Re-Review
Review Type:	□ Requirements Document	☑ Design Document	□ Code	□ Other
Disposition:	□ Accept	□ Conditional	□ Re-inspect	

Duration of Review Meeting(s):	________	(hours)	Number of Reviews:	________	(all)
Size of Materials:	________	(lines/pages)	Total Preparation Time:	________	(hours)
Total Minor Issues:	________		Total Major Issues:	________	
Number of Review Meetings:	________				
Rework Completed by:	________	(date)	Estimated Rework Effort:	________	(hours)
Re-inspection Scheduled for:	________	(date)	Actual Rework Effort:	________	(hours)

Inspectors/Peers:

________________________________ ________________________________

________________________________ ________________________________

________________________________ ________________________________

Additional Moderator Time (For Conditional Disposition): __________ (hours)

Moderator Signature: ________________________________

Completion Date: ________________________________

Additional Comments:

____

____

____

Project Review Management Report Description

Project – Name of project
Meeting Date – Date of review
Moderator – Name of moderator
Recorder – Name of recorder
Document Name – Titled by Author
Version - Version number of review product, if applicable
Directory Location – Location of the Project's information, located in S:\Technology Projects directory
Meeting Type – Inspection – first occurrence of inspecting this product/documentation. Re-inspection – inspecting this product/documentation a duplicate time. Peer Review – first occurrence of a peer review of this product/documentation. Re-review – a peer review of this product/documentation a duplicate time.
Review Type – the document that will be reviewed (includes the Requirements document; Design document; Code, etc.)
Disposition - Accept - review of documentation/code is acceptable, project can proceed; Conditional - minor issues, only moderator needs to review changes; Re-inspect/review: Review team should re-review the project after changes are made.
Duration of meeting(s) - The total time used for the review meeting(s). Meeting breaks are included.
Number of participants - Total number of reviewers including the moderator and author.
Size of materials - The amount of the materials reviewed (Please note that this is normally reported in pages, except for code)
Total preparation time - The sum of all individual preparation times, including moderator’s prep time, for all meeting sessions, per review.
Total minor issues - The sum of all minor issues.
Total major issues - The sum of all major issues.
Number of review meetings – One (1) if review completed in one meeting, more than one for multi-session meetings.
Rework completed by - The author’s commitment to a completion date for rework. This date is generally not entered into the Project database.
Estimated rework effort - The author’s estimate of the amount of work required to resolve the issues. This estimate is generally not entered into the Project database.
Re-inspection/review scheduled for - Used only when the Disposition is re-inspect. This date is generally not entered into the Project database.
Actual rework effort - The amount of effort the author has expended to resolve the issues. This field is completed after reexamination by the moderator or after a re-inspection or re-review meeting. If the Disposition is “conditional”, then leave this field blank and the estimated rework effort will be used. This information is entered into the Project database.
Inspectors/Peers - Lists all the inspectors/peers excluding the author
Moderator Review Time (For Conditional Disposition): Record the time required (hours) to verify that the issues found during the review have been corrected. (Applies only to reviews where the Disposition was “Conditional”.)
Moderator signature - The signature of the moderator is given at the completion of the meeting unless the Disposition is “Conditional”, whereby it will be given after the rework is examined.
Completion date - Date of Disposition or upon completion of rework examination in the case of a “Conditional” Disposition.
Additional comments - May be provided to note any conditions, suggestions, etc., which the inspectors/peers wish to record, such as recommended changes to standards. These comments are not stored in the Product database.

NOTE: The author should not be specifically identified. In this way, issue data is separated from management review of author performance data.

Information Systems Development Methodology
Project Review Management Report

Project:	Environmental Screening Tool	Meeting Date:	
Moderator:	Peter McGilvray	Recorder:	
Document Name:	Updated Design Document	Version:	2009_0731
Directory Location:			

Meeting Type:	☑ Inspection	□ Re-inspection	□ Peer Review	□ Re-Review
Review Type:	□ Requirements Document	☑ Design Document	□ Code	□ Other
Disposition:	□ Accept	□ Conditional	□ Re-inspect	

Duration of Review Meeting(s): ________ (hours) Number of Reviews: ________ (all)

Size of Materials: ________ (lines/pages) Total Preparation Time: ________ (hours)

Total Minor Issues: ________ Total Major Issues: ________

Number of Review Meetings: ________

Rework Completed by: ________ (date) Estimated Rework Effort: ________ (hours)

Re-inspection Scheduled for: ________ (date) Actual Rework Effort: ________ (hours)

Inspectors/Peers:

________________________ ________________________

________________________ ________________________

________________________ ________________________

Additional Moderator Time (For Conditional Disposition): __________ (hours)

Moderator Signature: ________________________________

Completion Date: ____________________________

Additional Comments:

____

____

____

Project Review Management Report Description

Project – Name of project
Meeting Date – Date of review
Moderator – Name of moderator
Recorder – Name of recorder
Document Name – Titled by Author
Version - Version number of review product, if applicable
Directory Location – Location of the Project's information, located in S:\Technology Projects directory
Meeting Type – Inspection – first occurrence of inspecting this product/documentation. Re-inspection – inspecting this product/documentation a duplicate time. Peer Review – first occurrence of a peer review of this product/documentation. Re-review – a peer review of this product/documentation a duplicate time.
Review Type – the document that will be reviewed (includes the Requirements document; Design document; Code, etc.)
Disposition - Accept - review of documentation/code is acceptable, project can proceed; Conditional - minor issues, only moderator needs to review changes; Re-inspect/review: Review team should re-review the project after changes are made.
Duration of meeting(s) - The total time used for the review meeting(s). Meeting breaks are included.
Number of participants - Total number of reviewers including the moderator and author.
Size of materials - The amount of the materials reviewed (Please note that this is normally reported in pages, except for code)
Total preparation time - The sum of all individual preparation times, including moderator's prep time, for all meeting sessions, per review.
Total minor issues - The sum of all minor issues.
Total major issues - The sum of all major issues.
Number of review meetings – One (1) if review completed in one meeting, more than one for multi-session meetings.
Rework completed by - The author's commitment to a completion date for rework. This date is generally not entered into the Project database.
Estimated rework effort - The author's estimate of the amount of work required to resolve the issues. This estimate is generally not entered into the Project database.
Re-inspection/review scheduled for - Used only when the Disposition is re-inspect. This date is generally not entered into the Project database.
Actual rework effort - The amount of effort the author has expended to resolve the issues. This field is completed after reexamination by the moderator or after a re-inspection or re-review meeting. If the Disposition is "conditional", then leave this field blank and the estimated rework effort will be used. This information is entered into the Project database.
Inspectors/Peers - Lists all the inspectors/peers excluding the author
Moderator Review Time (For Conditional Disposition): Record the time required (hours) to verify that the issues found during the review have been corrected. (Applies only to reviews where the Disposition was "Conditional".)
Moderator signature - The signature of the moderator is given at the completion of the meeting unless the Disposition is "Conditional", whereby it will be given after the rework is examined.
Completion date - Date of Disposition or upon completion of rework examination in the case of a "Conditional" Disposition.
Additional comments - May be provided to note any conditions, suggestions, etc., which the inspectors/peers wish to record, such as recommended changes to standards. These comments are not stored in the Product database.

NOTE: The author should not be specifically identified. In this way, issue data is separated from management review of author performance data.

Information Systems Development Methodology
Project Review Management Report

Project: Environmental Screening Tool
Moderator: Peter McGilvray
Document Name: Updated Design Document (Attachment A)
Meeting Date: ________
Recorder: ________
Version: 2010_0731
Directory Location: ________

Meeting Type:	☑ Inspection	□ Re-inspection	□ Peer Review	□ Re-Review
Review Type:	□ Requirements Document	☑ Design Document	□ Code	□ Other
Disposition:	□ Accept	□ Conditional	□ Re-inspect	

Duration of Review Meeting(s): ________ (hours)
Number of Reviews: ________ (all)

Size of Materials: ________ (lines/pages)
Total Preparation Time: ________ (hours)

Total Minor Issues: ________
Total Major Issues: ________

Number of Review Meetings: ________

Rework Completed by: ________ (date)
Estimated Rework Effort: ________ (hours)

Re-inspection Scheduled for: ________ (date)
Actual Rework Effort: ________ (hours)

Inspectors/Peers:

________ ________
________ ________
________ ________

Additional Moderator Time (For Conditional Disposition): ________ (hours)

Moderator Signature: ________

Completion Date: ________

Additional Comments:

____
____
____

Project Review Management Report Description

Project – Name of project
Meeting Date – Date of review
Moderator – Name of moderator
Recorder – Name of recorder
Document Name – Titled by Author
Version - Version number of review product, if applicable
Directory Location – Location of the Project's information, located in S:\Technology Projects directory
Meeting Type – Inspection – first occurrence of inspecting this product/documentation. Re-inspection – inspecting this product/documentation a duplicate time. Peer Review – first occurrence of a peer review of this product/documentation. Re-review – a peer review of this product/documentation a duplicate time.
Review Type – the document that will be reviewed (includes the Requirements document; Design document; Code, etc.)
Disposition - Accept - review of documentation/code is acceptable, project can proceed; Conditional - minor issues, only moderator needs to review changes; Re-inspect/review: Review team should re-review the project after changes are made.
Duration of meeting(s) - The total time used for the review meeting(s). Meeting breaks are included.
Number of participants - Total number of reviewers including the moderator and author.
Size of materials - The amount of the materials reviewed (Please note that this is normally reported in pages, except for code)
Total preparation time - The sum of all individual preparation times, including moderator's prep time, for all meeting sessions, per review.
Total minor issues - The sum of all minor issues.
Total major issues - The sum of all major issues.
Number of review meetings – One (1) if review completed in one meeting, more than one for multi-session meetings.
Rework completed by - The author's commitment to a completion date for rework. This date is generally not entered into the Project database.
Estimated rework effort - The author's estimate of the amount of work required to resolve the issues. This estimate is generally not entered into the Project database.
Re-inspection/review scheduled for - Used only when the Disposition is re-inspect. This date is generally not entered into the Project database.
Actual rework effort - The amount of effort the author has expended to resolve the issues. This field is completed after reexamination by the moderator or after a re-inspection or re-review meeting. If the Disposition is "conditional," then leave this field blank and the estimated rework effort will be used. This information is entered into the Project database.
Inspectors/Peers - Lists all the inspectors/peers excluding the author
Moderator Review Time (For Conditional Disposition): Record the time required (hours) to verify that the issues found during the review have been corrected. (Applies only to reviews where the Disposition was "Conditional.")
Moderator signature - The signature of the moderator is given at the completion of the meeting unless the Disposition is "Conditional," whereby it will be given after the rework is examined.
Completion date - Date of Disposition or upon completion of rework examination in the case of a "Conditional" Disposition.
Additional comments - May be provided to note any conditions, suggestions, etc., which the inspectors/peers wish to record, such as recommended changes to standards. These comments are not stored in the Product database.

NOTE: The author should not be specifically identified. In this way, issue data is separated from management review of author performance data.

Information Systems Development Methodology
Project Review Management Report

Project:	Environmental Screening Tool	Meeting Date:	
Moderator:	Peter McGilvray	Recorder:	
Document Name:	Updated Design Document (Attachment A)	Version:	2012_0731
Directory Location:			

Meeting Type:	☑ Inspection	□ Re-inspection	□ Peer Review	□ Re-Review
Review Type:	□ Requirements Document	☑ Design Document	□ Code	□ Other
Disposition:	□ Accept	□ Conditional	□ Re-inspect	

Duration of Review Meeting(s): ________ (hours)

Number of Reviews: ________ (all)

Size of Materials: ________ (lines/pages)

Total Preparation Time: ________ (hours)

Total Minor Issues: ________

Total Major Issues: ________

Number of Review Meetings: ________

Rework Completed by: ________ (date)

Estimated Rework Effort: ________ (hours)

Re-inspection Scheduled for: ________ (date)

Actual Rework Effort: ________ (hours)

Inspectors/Peers:

Additional Moderator Time (For Conditional Disposition): __________ (hours)

Moderator Signature: ____

Completion Date: ____

Additional Comments:

Project Review Management Report Description

Project – Name of project
Meeting Date – Date of review
Moderator – Name of moderator
Recorder – Name of recorder
Document Name – Titled by Author
Version – Version number of review product, if applicable
Directory Location – Location of the Project's information, located in S:\Technology Projects directory
Meeting Type – Inspection – first occurrence of inspecting this product/documentation. Re-inspection – inspecting this product/documentation a duplicate time. Peer Review – first occurrence of a peer review of this product/documentation. Re-review – a peer review of this product/documentation a duplicate time.
Review Type – the document that will be reviewed (includes the Requirements document; Design document; Code, etc.)
Disposition – Accept - review of documentation/code is acceptable, project can proceed; Conditional - minor issues, only moderator needs to review changes; Re-inspect/review: Review team should re-review the project after changes are made.
Duration of meeting(s) – The total time used for the review meeting(s). Meeting breaks are included.
Number of participants – Total number of reviewers including the moderator and author.
Size of materials – The amount of the materials reviewed (Please note that this is normally reported in pages, except for code)
Total preparation time – The sum of all individual preparation times, including moderator's prep time, for all meeting sessions, per review.
Total minor issues – The sum of all minor issues.
Total major issues – The sum of all major issues.
Number of review meetings – One (1) if review completed in one meeting, more than one for multi-session meetings.
Rework completed by – The author's commitment to a completion date for rework. This date is generally not entered into the Project database.
Estimated rework effort – The author's estimate of the amount of work required to resolve the issues. This estimate is generally not entered into the Project database.
Re-inspection/review scheduled for – Used only when the Disposition is re-inspect. This date is generally not entered into the Project database.
Actual rework effort – The amount of effort the author has expended to resolve the issues. This field is completed after reexamination by the moderator or after a re-inspection or re-review meeting. If the Disposition is "conditional," then leave this field blank and the estimated rework effort will be used. This information is entered into the Project database.
Inspectors/Peers – Lists all the inspectors/peers excluding the author
Moderator Review Time (For Conditional Disposition) – Record the time required (hours) to verify that the issues found during the review have been corrected. (Applies only to reviews where the Disposition was "Conditional.")
Moderator signature – The signature of the moderator is given at the completion of the meeting unless the Disposition is "Conditional," whereby it will be given after the rework is examined.
Completion date – Date of Disposition or upon completion of rework examination in the case of a "Conditional" Disposition.
Additional comments – May be provided to note any conditions, suggestions, etc., which the inspectors/peers wish to record, such as recommended changes to standards. These comments are not stored in the Product database.

NOTE: The author should not be specifically identified. In this way, issue data is separated from management review of author performance data.

Project Revision Log

Project Revision Log

Date

July 31, 2012

Project Name Version

Environmental Screening Tool (EST)

Document Name

Updated Design Document

Revision #	Date	Brief Description of Change	Project Manager Initials
1	4/30/2008	Updated Section 1.2 Graphical User Interface Design; Section 2.6 Data Migration and Transformation; and Added Attachment A - Data Migration and Transformation Procedures	
2	7/31/2009	Updated Attachment A - Data Migration and Transformation Procedures	
3	7/31/2010	Updated Attachment A - Data Migration and Transformation Procedures to include transformations performed 1/1/2009 through 12/31/2009	
4	7/31/2012	Updated Attachment A - Data Migration and Transformation Procedures to include transformations performed 1/1/2010 through 8/31/2011	

Stage-end Walkthrough Form

Stage-end Walkthrough Form

Stage:	**Design**
Project Manager:	**Peter McGilvray**
Project Name:	**Environmental Screening Tool**
Date:	**November 30, 2007**

List each deliverable that was completed during this stage:

Draft Test Plan (no changes)
Updated Design Document
Updated User Handbook
Updated Project Schedule
Stage-end Walkthrough Form

Open Issues	**Resolved Prior to Next Stage? (Y/N)**
None	

Comments: Payment of invoices signified acceptance of the deliverables. Authorization and subsequent Task Work Orders signifies notice to proceed.

Project Team Members

Name	**Signature**
Peter McGilvray	

Outcome: (circle one)

[✓ Move to Next Stage]

Resolve Issues Prior to Moving to Next Stage

Stage-end Walkthrough Form

Stage:	**Design**
Project Manager:	**Peter McGilvray**
Project Name:	**Environmental Screening Tool**
Date:	**August 29, 2008**

List each deliverable that was completed during this stage:

Updated Project Schedule
Stage-end Walkthrough Form

Open Issues	**Resolved Prior to Next Stage? (Y/N)**
None	

Comments: Payment of invoices signified acceptance of the deliverables. Authorization and subsequent Task Work Orders signifies notice to proceed.

Project Team Members

Name	**Signature**
Peter McGilvray	

Outcome: (circle one)

[✓ Move to Next Stage]

Resolve Issues Prior to Moving to Next Stage

Stage-end Walkthrough Form

Stage:	**Design**
Project Manager:	**Peter McGilvray**
Project Name:	**Environmental Screening Tool**
Date:	**July 31, 2009**

List each deliverable that was completed during this stage:

Updated Project Schedule
Stage-end Walkthrough Form

Open Issues	**Resolved Prior to Next Stage? (Y/N)**
None	

Comments: Payment of invoices signified acceptance of the deliverables. Authorization and subsequent Task Work Orders signifies notice to proceed.

Project Team Members

Name	**Signature**
Peter McGilvray	

Outcome: (circle one)

[✓ Move to Next Stage]

Resolve Issues Prior to Moving to Next Stage

Stage-end Walkthrough Form

Stage:	**Design**
Project Manager:	**Peter McGilvray**
Project Name:	**Environmental Screening Tool**
Date:	**July 31, 2010**

List each deliverable that was completed during this stage:

Updated Design Document (Attachment A)
Updated User Handbook
Updated Project Schedule
Stage-end Walkthrough Form

Open Issues	**Resolved Prior to Next Stage? (Y/N)**
None	

Comments: Payment of invoices signified acceptance of the deliverables. Authorization and subsequent Task Work Orders signifies notice to proceed.

Project Team Members

Name	**Signature**
Peter McGilvray	

Outcome: (circle one)

[✓ Move to Next Stage]

Resolve Issues Prior to Moving to Next Stage

Stage-end Walkthrough Form

Stage:	**Design**
Project Manager:	**Peter McGilvray**
Project Name:	**Environmental Screening Tool**
Date:	**July 31, 2012**

List each deliverable that was completed during this stage:

Updated Design Document (Attachment A)
Updated User Handbook
Updated Project Schedule
Stage-end Walkthrough Form

Open Issues	**Resolved Prior to Next Stage? (Y/N)**
None	

Comments: Payment of invoices signified acceptance of the deliverables. Authorization and subsequent Task Work Orders signifies notice to proceed.

Project Team Members

Name	**Signature**
Peter McGilvray	

Outcome: (circle one)

[✓ Move to Next Stage]

Resolve Issues Prior to Moving to Next Stage