

Appendix: Agents

REPOSITORY

This section provides a list of all the agents provided in the system databases with an emphasis on the scheduled agents that provide essential background services.

Admin\ImplementPendingNameChangesNow

Operation: Manual, Action Menu, Requires PGADMIN role.

Function: Executes the *Implement ProjectAndParticipantName Change* Agent immediately.

Usage: When you must implement some change immediately and cannot wait for the scheduled running of the agent.

Admin\ReviseProjectID

Operation: Manual, Action Menu, Requires PGADMIN role.

Function: This will allow you to manually enter a number that will be recorded as the uniqueprojectid number in a project profile.

Usage: In very rare cases it may be necessary to change the uniqueprojectID number set for a project. This situation might arise if someone accidentally deletes a project profile. The only way to get a replacement, assuming no backups are available, is to create a new project profile and then use this function to change its number to that of the missing profile. The correct value of the uniqueprojectId will be found any every assignment document of the original project.

It is your responsibility to ensure that the number entered is indeed unique.

Admin\RunProcessTrackingNow

Operation: Manual, Action Menu, Requires PGADMIN role.

Function: Executes the *ProjectGo/Process Tracking* agent immediately.

Usage: Use to remind all project leaders of their process problems and to immediately update the process information shown in the project profile and rendered in the process column of the project dashboard.

Admin/UpdateDashboardsNow

Operation: Manual, Action Menu, Requires PGADMIN role.

Function: Executes the *Maintain Dashboards* agent immediately.

Usage: Use when dashboard data must be completely current. Helpful when doing testing or demos, should not be needed in a normal production operation. Typically requires about one minute per 2000 documents in the database.

Note, the process column of the project dashboard is maintained by the ProjectGo\ProcessTracking agent, not the Maintain Dashboards agent.

Admin\UpdateObservatoryNow

Operation: Manual, Action Menu, Requires PGADMIN role.

Function: Executes the *XMLRepositoryLink* agent immediately.

Usage: Use when Observatory data must be completely current. Helpful when doing testing or demos, should not be needed in a normal production operation. Typically requires 10 seconds per project.

Implement ProjectAndParticipant Name Change

Operation: Scheduled Daily.

Function: Implements all name changes that have been requested using the Identification Actions of Project and/or Participant profiles

When a project name is changed, issues, risks, scope changes, project documents, status reports, interproject dependencies, document dependencies, all project assignments and the project profile documents are updated

When a participant name changes is processed, assignments, timesheets, and status reports are updated and a permanent record document is created which will be used by the Project Gateway Tools to assist in synchronization of existing projects. These permanent records will be found in the view entitled "Participant Name Changes" on the Reports navigator panel.

If no name changes have been requested since the previous execution the agent will complete in about one minute per 500 projects + participants. The time for implementing changes will depend upon the number of changes and the size of the projects.

Maintain Dashboards

Operation: Scheduled Daily or more often. Can be executed manually using the Admin\UpdateDashboardsNow action.

Function: Extensive. This agent must be running in order for primary system features to function.

This agent updates all of the statistics found in project profile documents which are also displayed in the project dashboard view and exported to Project Observatory by the XMLRepositoryLink agent.

In the process of doing this, it examines all assignment, scope, issue, risk, action item, and project profile documents in the database. Project profiles are updated only if there is a change to be recorded. Project assignments are updated only if a linked document has been deleted.

Scope, Issue and Risk forms are updated when linked assignments change.

This agent also updates the estimated completion date and effort at completion for all deliverables that are linked to projects, key events, or sub deliverables. This provides the used for the deliverables dashboard view.

The agent send emails to project managers and deliverable owners when indicators transition to "red state" as a result of data changes if appropriate controls are preset.

This agent finds signoff documents that have been approved and makes the necessary updates to the project profile to record the signoff.

The agent sends a reminder if a scope change has been approved but no comments have been placed in the resolution field after a period of time.

The agent sends a reminder five days before documentation assignments are due.

This agent finds assignment documents which use signoff document identifiers in the documents required field. Once found, such assignments are synchronized with the status of the signoff such that the assignment will be marked done when the signoff is approved and forced into a non-finished status so long as the signoff is not approved. If the signoff identifier code is not valid, then an email is set (repetitively) to the project manager requesting correction of the invalid code.

Execution: Typical run time is about one minute per 2000 documents, but can vary widely depending upon the number of updates and document mix.

Maintain AccessControl

Operation: Schedule Agent, Run Daily. Best if run after Maintain TeamOrganizations and Maintain Team Participants.

Function: Updates all documents which use "Teams" for access control to incorporate changes in the membership of those teams.

This is a key function which grants or withholds visibility and/or edit rights to individuals to specific documents based upon their team membership. If it is not running on a regular basis, then changes made in teams will not be reflected in existing documents until they are manually edited by someone and resaved.

Note that the two other agents can modify the membership of existing teams, so they should run earlier than this agent. If this is not the case, then an organization change or participant user id change might take two agent cycles to be fully implemented in existing documents rather than one.

MaintainNews

Operation: Scheduled Agent, Run Daily.

Function: This agent reviews all News documents. If a document's "first publication" time has arrived, it is updated to make it visible in the news view. If the expiration date has arrived, it will be marked for subsequent deletion by the PGDocumentPurgeAgent. Certain document with one day a week display status will be enabled or disabled as appropriate.

This agent must run daily in the evening to ensure the proper implementation of the news display rules. Typical execution time will be less than one minute.

MaintainNumbering

Operation: Scheduled Agent, Run Daily.

Function: Certain documents (issues, risks, scope changes) are automatically numbered when they are created. However, in a replicated or web environment, it is possible to create two documents with the same number. This agent detects such events and renumbers one of the documents to the next available number in sequence.

Note that this has nothing to do with "replication conflicts", but rather with resolving the situation where two separate documents get initially assigned the same sequence number.

Typical execution time will be less than one minute.

MaintainTeamOrganizations

Operation. Scheduled Agent, Run Daily. Best if run prior to "Maintain Access Control".

Function: The agent finds all Type I Team Profiles which have an organization name specified. Then the agent updates the member list of each team profile with the collection of current userid's of the members of the specified organization and all its sub organizations.

This means that, if a particular document is set to be visible only to the members of a team, and if that team is defined to include the members of a particular repository organization, then, if a new participant is added to the organization, this agent will modify the team list to add the userid of that individual. Subsequently, the Maintain Access Control Agent will update the readers list of the document to the revised list of team members. This will allow that document it to appear in the appropriate document view when opened by the new organization member.

MaintainTeamParticipants

Operation. Scheduled Agent, Run Daily. Best if run prior to "Maintain Access Control".

Function: The agent finds all Type II Team Profiles and updates the member list of each team profile with the collection of current userid's of the participants selected in the specified team.

See MaintainTeamOrganizations above.

Open Journal

Operation: Can be used manually from the agents menu and is executed automatically by the "Journal" action of the Project Documents navigator panel when used with a Notes Client.

Function: Locates the "Journal" document associated with the current username and opens it for display. Creates such a document if one does not exist.

PGDocumentPurgeAgent

Operation: Scheduled Agent, Run Daily or Weekly.

Function: Deletes normal documents such as issues, etc. that have been marked for deletion after a waiting period. This waiting period is preset to 30 days, but can be changed by a declaration found within the agent.

Deletes all response documents associated with the deleted document even though the response documents are not specifically marked for deletion.

Also deletes all xxDeletedCalendarForms and xxDeletedTodoForms and xxReserveForm documents which are preset to force quick deletion. In no case will a document be deleted until it is at least 1 hour old.

Documents which appear in the view "All Deleted Documents" will be processed by this agent.

This agent also removes ReviewComments and HistoricalReviewComments documents from the repository when the parent document has been deleted or archived.

Project Go!\Document Tracking

Operation: Scheduled Agent, Run Weekly. Key system agent.

Function: This agent sends an email to Project managers to summarize status of project documents with respect to document tracking requirements and provide a complete list of open issues.

Note that this agent will not send notifications if the project is not currently in an approved status.

ProjectGo!\Action Tracking

Operation: Scheduled Agent, Run Daily. Key system agent.

The agent scans all open action items in the database to find any that required reminders to be sent at the end of the reminder period and to find any that are overdue to send reminders to the assignee and creator.

This agent also removes any canceled action items from the database by soft deleting them.

ProjectGo!\Process Tracking

Operation: Scheduled Agent, Run Daily. A key system agent.

Function: This agent scans all project profiles. It causes processes to be adopted as required according to the process management setting of the associated program profile document. For all projects that have

processes, it executes an evaluation of the process condition. If the condition has changed, it updates the project profile document with the new status and sends an email to the project leader (and optionally to the program manager) reporting the condition.

The evaluation process compiles a complete list of required and conditional signoffs process required project documents. It determines whether or not any of these are overdue or due in the near future.

If a project is finished or canceled, and if the current date is later than the effective date of the status, then the agent will ignore the project.

For projects with process problems whose conditions have not changed, messages will be resent every 14 days. This can be adjusted by a parameter in the declaration of the agent.

XML Project Observatory Link

Operation: Scheduled Agent, Run Daily or when requested by the Admin\UpdateObservatoryNow action that may be performed by a person with the [PGADMIN] role. Best if run after *Maintain Dashboards*.

Function: Gathers status information about the projects in this repository and transmits it to a Project Observatory database. Transmission can be done directly (if both databases are on the same server) or by automated email. The data is produced in an XML format (PROJECTABSTRACT) using a Data Type Definition embedded within each message. Details on this DTD can be found on the Marin Research Web site.

If a costing center database is available, the project target and actual cost will be retrieved from the costing center.

There are several configuration parameters for this agent which are set in the Repository Administration form. If no Observatory is installed, the agent will exit after just a couple seconds.

(BriefingAssistant)

Operation: Run on demand via URL. This URLs are created in the report section of profile documents and in Project Observatory Briefing pages.

Function: This agent will locate a Gantt chart if one is available for a specified project, program or organization and will cause the server to return the .PDF file attachment to the browser.

An agent is used to open the chart because the filename of the .pdf file attachment changes each time it is updated. As a result, you cannot "bookmark" or compute a static link that will open the chart file.

This agent also retrieves project task lists for Observatory users.

(FindDBN)

Operation: A web agent that runs as a query save agent to retrieve a particular document using the document number. If no document of

that number exists, the agent returns the Reports - Numbered Documents view to the user.

(Holidays.c2k)

Operation: Run from a navigator link.

Function: Scans all participant profiles in the repository and returns an XML file using the .c2k format which is understood by the Project Gateway XML Macros for Microsoft Project.

The .C2K file contains a list of all participants and a list of holiday days for each. This list is constructed from the Schedule Info entries entered in the repository manually and from those entered by the Observatory when Holiday Synchronization is enabled.

(PGTSWebReject)

Operation: This agent is run when a web timesheet is rejected by the owner or approver.

Function: It resets the internal status of the TimeSheetWeb document so that it can be edited by the owner.

(RSSFeed.xml)

Operation: Invoked by URL

Function: Returns selected repository data in RSS newsreader protocol. The set of data to be returned is determined by the URL parameters.

(WebActionItems)

Operation: Used when any of a number of web transactions are completed including the creation of action items, reassignment, changing of due dates, closure, cancellation.

Also used to create and process the Action Item Update form and the Action Item reports.

(WebAddParticipantAgent)

Operation: Used when the WebAdditionalAssignment form created by the "New Coworker" Assignment action button is submitted.

Function: Starts from the information on the WebAdditionalAssignment form to find the originating assignment. Uses the (UniqueTaskResourceView) to find all the existing coworkers. Creates the new assignment documents by copying the original and then updating participant dependent fields such as resourcename, taskhelp, pg_groupname and the pgx_editaccesslist. When the new documents have been created it then updates all of the assignments with doclinks to each other and redisplay the originating assignment.

(WebAgentStatusReport)

Operation: Used when the AgentStatusReport action is clicked by a web user.

Function: Iterates over all database agents to get name and basic information, then generates an HTML page containing the data in tabular form.

(WebAIParentSelect)

Operation: Used when the Note action is clicked by a web user.

Function: Used to setup the parent association for the new action item or to force a project selection prior to displaying the pgSendNote form.

This also creates a return parameter so that the system can return to the originating document after creating the new action item.

(WebAssetInitiatorAgent)

Operation: Used when "A New Asset" link is executed.

Function: Determines the type of the asset owner and then presets parameters into the obsNewLibraryForm so that the asset is associated with the owning document.

(WebAssetVersionAgent)

Operation: Initiated by the Create New Version action on Asset documents.

Function: Copies the existing document and makes the copy a response to the original document. Thus the document becomes ready for editing as a new version of itself.

(WebCreateDocDependencyAgent)

Operation: This agent executes when the pgWebProjectLinkSelect form is submitted. This form is created when the "Link" action is clicked on an Issue, Risk or Scope Change Form.

Function: Creates one or more DependencyLink documents representing the dependency of this Issue, Risk, or Scope Change on a project. All the information in these documents is preset by the agent. These dependency link documents are displayed in views of issues, risks, and scope changes and in reports, but are not counted in the project dashboard statistics.

(WebCreateProjectAgent)

Operation: This agent executes when the A New Project form is submitted.

Function: This agent creates one or more Gantt Chart definition documents for the new project using the preinstalled Graphic templates. This provides the same function for manually created projects as is provided when a project is imported. The new Gantt charts will not normally be available for viewing until the next day.

(WebEditNewVersionAgent)

Operation: This agent executes when the "EditNewVersion" action is clicked on a project document by a web user.

Function: The agent creates a new document which is a complete copy of the current document, and makes it a response document (\$Ref) to the current document which is then opened for editing by the user. A field called "VersionNotice" is created within the copied document which shows the date of the versioning.

Clears the version notice and removes all approval status in the new version so that it must be reapproved.

(WebGanttChartAgent)

Operation: This agent is executed by a URL that specifies its actions. It returns a PDF file or, if no data was available, a simple response page.

Function: This agent creates Gantt charts in PDF format for collections of repository documents. It implements the Issues, Risk, Documents, and Action Item Gantt charts.

The agent does very little. It simply decodes the parameters for the chart from the URL and then invokes the pgGanttDataSupport library to assemble the chart.

(WebIssueSaveAgent)

Operation: This agent is executed whenever an issue or risk form is submitted by a web user.

Function: This agent assigns the next available document reference number to the document if it is a new issue or if it does not already have a document reference number (pgDocRefNumber field) assigned.

(WebLibraryActionAgent)

Operation: Called as an agent from the Delete and Associate actions on the Asset form. Called as a querysave agent on the various AssetAssociate... forms.

Function: Implements the softdelete and Associate actions on Asset documents.

(WebMyApprovals)

Operation: Run by URL which specifies user and optional project.

Function: Generates the HTML MyApprovals report which lists documents and timesheets that await the approval of the user.

The report data is prepared by the pgApprovalSupport library functions and then is formatted for display by this agent.

(WebMyIssuesB.csv)

Operation: These agents are initiated by a URL with various parameters which specify the particular report subject and format.

Function: Creates participant, project, and program project status reports, Interproject Dependencies, Discussions, and Deliverables reports and related thread reports.

(WebMyIssues.csv)

Operation: These agents are initiated by a URL with various parameters which specify the particular action. These URL are precomputed on all profile documents and in forms initiated by links labeled ProjectReports and ProgramReports. The URLs can be bookmarked and recorded within the profile documents.

Function: These agents provide HTML report generation for Issues, Risks, Scope Changes, Project Documents, Signoff Reports, and response threads.

URLS: The URL parameters specify the report type, the columns, sorting, grouping, and the subject matter.

(WebMyProjects)

Operation: This agent is initiated by a URL trigger by the My Projects selection from repository center.

Function: This agent scans all projects in the project dashboard view and finds those whose user id field contains the current username specified in the URL &User= parameter. It outputs an HTML table listing the project name, the icons for Time, Effort, Progress, and Issues shown on the project dashboard, the current approval, process stage, and process icon and hyperlinks to a project synopsis report using the (askPgQuery) agent and to the display of the project calendar using the (WebToDoAgent). The icons numbers are actually retrieved from the project dashboard view.

(WebMyReviews)

Operation: Executed by URL containing the current user name. Initiated by the My Reviews action from repository center.

Function: Creates an HTML report listing the documents for which the user has initiated or been invited to participant in a review cycle.

(WebNewInfoDisplay)

Operation: This agent is executed by a URL generated from the "Whats New" action on the "My.." menu from repository center.

Function: The output of the agent is a HTML table that shows documents in the repository related to the current user. The table is organized by rows for each document type. Documents are listed left to right in order of recency and color coded.

Because the generation of this tables data would be very time consuming, the agent employs a caching mechanism. Summary information for all recently created documents is retained in the agent data object. This data set is refreshed on request or on the next invocation after an expiration time. Normal requests simply reload the cached agent data into memory, filter for the user, and format the results for display.

A maximum of 1000 documents and a maximum age of 30 days are used to limit the caching.

(WebNewLibraryDocumentAgent)

Operation: This agent is executed as a querysave agent on the New Asset form.

Function: Sets up the LDSubformName field and resets the form name to obsLibraryForm and redirects the browser to open the Asset document for display.

(WebNewPjtDoc)

Operation: This runs as a query open agent when a project document is opened, but it functions primarily when a new document is created.

Function: The agent constructs a list of document types used in the project by scanning the assignments of the project in the "DocumentGo" view. It also looks up the project profile to find the processes adopted by the project and then adds any document types that are used as required documents in the process definitions of the adopted processes.

The result is the set of document types available for selection by the user.

(WebNewPjtAgent)

Operation: This agent executes whenever a project document is opened, but it does nothing unless a new document is being composed.

Function: If a new document is being composed, the agent creates a list of the document tracking codes used by assignments associated with the owning project. This list is preloaded into the form so that it is displayed to the user when the form opens.

(WebNoteAgent)

Operation: This agent is run when the form associated with the Note action is submitted by a web user. Much of the function is implemented using a script library *pgJournalSupport* which is shared with the Notes client implementation of the identical functionality.

Function: The agent interprets the "destination" checkmarks on the Note form and finds the appropriate email address for the user, the profile managers or the assignee. It then composes emails to these people.

If My Journal is selected, the agent locates or creates the journal document for the users name and appends a new section to the bottom of the existing text containing the (optional) doc and url link, and document title along with the note body message text.

(WebPDocApprovalAgent)

Operation: This is called as a query save agent in a project document, the document approval form (pgPDWebApvlForm), and the administrative override forms.

Function: This very complex agent executes the document approval process by generating the list of required approvers, sending notifications to approvers, processing approvals, abstentions, and rejections. It also implements the administrative override actions.

When used on a signoff request, the approver list is generated from signoff definition document instead of the project profile.

A very similar set of routines is found in the (WebScopeApprovalAgent) and in the global routines of the pgxProjectDocument subform and the Scope Change form.

(WebPipelinesAction)

(WebPipelinesSelect)

(WebPipelines)

These three agents implement the Activity Pipelines facility from the server side.

The "action" agent is used only when the action panel "go" function is used. It takes a list of document identifiers and an action code and makes changes to the selected documents.

The "select" agent is used to create and process the Activity Pipelines Selector panel, which exists only as the output of this agent.

The "WebPipelines" agent is used to create the pipelines presentations in all formats. It is called directly from the repository menus as well as from all profiles.

(WebPermissionAgent)

Operation: This is called by URL from Signoff Process Definition Forms, Signoff Definition forms, and Document Type definition forms by a simulated Delete button.

Function: This agent determined if the document is currently in use, either because it has been adopted by a project, or used in a process definition. If so, the user is politely told that it cannot be deleted. If not in used, the user is redirected to the normal delete action page.

(WebPGWhatIfAgent)

Operation: This agent is executed by URL and by query open and query save actions on the pgSelectGanttWhatIfActionWeb form and pgSelectGanttWhatIfOnlyWeb forms.

Function: 1. The agent generates the Project Buffers report. 2. The agent generates the GanttWhatIf chart. 3. The agent implements the reschedule actions (create proposal, implement proposal, undo, etc.) from the project profile.

Most of the processing is done by the pgGanttDataSupport library.

(WebPOChangeNameAgent)

Operation: Runs as a querysave agent from the pgPOChangeNameFormWeb form that is triggered by the Identification action on Organization and Program profiles.

Function: This agent implements the immediate aspects of the name change including the revision of all affected programs or organizations and attached participants and projects and chart definitions. The

propagation of these changes to other documents are handled by the Maintain Organization and Maintain Dashboard agents.

Similar logic is found in the QueryClose methods of the Program and Organization profile forms.

(WebPOValidationDeleteAgent)

Operation: Runs from URL.

Function: Determines if a program, participant, or organization can be deleted and if so, redirects the user to a delete action form. These groups cannot be deleted if they are in use on other profiles or have children or are assigned to tasks.

(WebProjectApprovalChangeAgent)

Operation: Runs as querysave agent from the pgWebPjtApvl form.

Function: Determines if the approval state of the project can be changed to the new state selected by the user on the project approval change - Step 1 form.

This uses a function in pgJournalSupport to evaluate the process requirements of the adopted processes with the signoffs obtained. Based upon the project approval interlock of the adopted signoff process definition(s), the user may not be able to transition from proposed to approved or to finished. When not permitted, a page will be returned explaining the reason why.

If the transition is permitted, then the next form in the approval change process will appear.

(WebProjectLogAgent)

Operation: Run by URL which contains the uniqueprojectid.

Function: Generates a formatted project log page with embedded Javascript filtering controls. The generation of the document list for this report is done using the pgLogSupport script library. That in turn uses the pgProjectLog view to find the ordered set of documents. This list is merged with key event and project checklist events and returned to the agent for rendering.

(WebPromoteDocAgent)

Operation: This is run when the form created by the "Promote" action on a project document is submitted by a web user.

Function: The agent removes the \$Ref field value and the VersionNotice field value. This allows the document to be seen as a first level document again rather than a response document.

Note: Once the document is promoted, it can be associated with a different project using the Associate action.

(WebReassignAllAgent)

Operation: Runs as a query save agent on the ReassignAll action from the administration form.

Function: This will change the participant assigned on issues, risks and scope changes and action items throughout the repository. Requires that the user have [PGADMIN] rights.

(WebRecentDocuments)

Operation: This agent is executed by a URL which is precalculated on project and participant profiles.

Function: The agent will produce an HTML listing of documents created within a time period. This is done by using a special view (pgRecentlyCreated) of documents sorted by descending creation date.

Parameters specified on the URL set the "lookback" period and whether Assignments, Documents (Issues, Risks, Scope Changes, Project Documents), or Timesheets will be selected and the name of the project or participant.

(WebReviewAgent)

Operation: Executed as a querysave agent from a number of forms used in the review process.

Function: Implements all aspects of the group review initiation and comment management process including the creation of the review comment and review comments documents.

Most of the non-UI processing is done in the pgJournalSupport library.

(WebROCopyAgent)

Operation: Executed by URL from the create template form and project report link.

Function: Used to make a copy of an existing project template and to create a new project template from an existing project.

(WebSchInfoAgent)

Operation: This agent is executed when the form associated with the "Schedule" action is opened and also when it is submitted by a web user. The Schedule action appears on project, program, organization and participant profile documents.

Function: On open, the agent retrieves the existing schedule information data from the profile, removes any elements which were defined to end prior to today, and populates the interactive document with the remainder. When the document is displayed, the user sees the existing schedule items that are still active.

On submit, the agent retrieves the edited schedule list from the form and records it into the profile document.

Note: Schedule information is recorded in a profile field called pgScheduleInfo which is a text list containing one or more items. This

field is displayed only in the Schedule Info form, but is also used by the WebTodoAgent.

(WebScopeChangeAgent)

Operation: This agent is executed with a scope change form is submitted and when a scope approval form (pgSCWebApvlForm) is submitted and when a scope Admin action form (pgSCWebAdminActForm) is submitted.

Function: This agent implements the numbering of new scope change documents, the update of 4 plain text fields which are available for reporting, and the entire scope change approval process for web users.

The approval process involves the interpretation of the project policy to generate the list of approvers, the dispatch of emails, the processing of approval forms to record approval, rejection or abstention and the processing of administrative override actions.

The agent process is replicated in the Notes environment by code built into the scope change form.

(WebSetLogoAgent)

Operation: This agent is executed when the pgSetLogoForm that is composed by the "SetLogo" action on the GatewayFieldMapForm (Administration Form) is submitted.

Function: The agent removes any existing logo document and constructs a link to the new logo file attachment (or external file). This link is stored in the Administration form and is used by the \$\$NavigatorTemplate for (RepositoryCenterWeb) to place the logo graphic on the repository home page.

(WebShowFieldsAgent)

Operation: This agent is executed from a URL that specifies a documentuniqueid as &DocID=. A link to this agent, labeled "SUPPORT" is generated on the "NOTE" action form (pgSendNote) when the user has the PGADMIN role and is not Anonymous.

Function: To generate a human readable printout of all of the fields in the referenced document. This output can then be sent to Marin Research for analysis.

Notes client users can use the built in "document properties" dialog to see the values of document fields. This provides a similar function for web users.

(WebSignoffAdoptAgent)

Operation: Runs from URLs from self created forms initiated from the adopt and remove actions of the project and program profile process management panels.

Function: Implements the adoption and removal of processes. A field "pg_SignoffsRequested" contains the list of adopted processes.

(WebSignoffRequestAgent)

Operation: Runs from URL, query open and query save on a variety of forms in the signoff process.

Function: Creates the list of signoff candidates by evaluating the processes. Creates the project document from the selected candidate. Generates a variety of reports including SignoffProcessUsage, Signoff Goals, ProcessMetrics, DocumentTypeUsage, Document Instances.

Much of the logic is contained in the pgSignoffSupport and pgJournalSupport libraries.

(WebSoftDeleteDocAgent)

Operation: This agent is executed whenever a Delete Document (pgDeleteDocForm) is submitted. NOTE: this agent should be set with the agent property "RunAsWebUser".

Function: If deletion is selected on the action form, the agent modifies the referenced document to set the form name to xxDeletedItem, and create the new fields xxDeletedOn, xxDeletedBy, and xxDeletedForm. These values are used by the PgDocumentPurgeAgent to determine when to remove the document and by the Restore function on the xxDeletedDocForm to resurrect the document back to active status in the event of an accidental deletion. The agent also removes the \$Ref field if there was one to allow the document to appear in the AllDeletedDocuments view.

(WebSpecialReportAgent)

Operation: Executed by URL.

Implements the Remove All Favorites function form the Repository Center Favorites page.

(WebSubSetup)

Operation: Executed as a query save agent on the pgCalSubscribeP and J forms (participant and project).

Function: Determine the uniquedocumentid of the selected profile document and passes this as a parameter to the pgCalSubscribe form which displays the subscription URL.

(WebTeamSelectAgent)

Operation: This agent is executed when a pgTeamSelectWeb form is opened and when it is submitted. That form is opened in response to the SelectParticipants action on the Type II Team Profile form (Team Profile B)

Function: When the form is opened, the agent prepares a "page" of participants (arranges as 4 columns of 25) for display as a checkbox array. The url determines which "page" is displayed.

On close, the agent reads the participant selections, adds them as team members and update the teamlist. The pgTeamSelectWeb form redirects the user to the next page until the participant list is exhausted.

There is a parallel Notes client implementation of the same selection process built into the Team Profile B form.

(WebTodoAgent)

Operation: This agent is executed by a URL generated by the Todo Action item found in all profiles and also created by the Custom Todo Report links and on SelectProjectReport and SelectProgramReport actions.

Function: This agent takes the URL specifications and generates the project, program, participant, organization or team calendar with the options specified in the URL. This output can be formatted in Month Calendar, Week Calendar or Table formats.

The actual selection of data is done by the script library *pgTodoSupport* which is also used by the Notes implementation.

(WebUpdateHistoryAgent)

Operation: Runs as a queriesave agent on the Ref action form from assignment documents.

Function: Calls the pgAgentSupport library to update the AsgHistory record of the assignment with the new reference data.

(WXWebMyIssuesB.csv)

Operation: Runs from "Split Screen" url on reports.

Function: Generates a dynamic frameset and launches the WebMyIssuesB.csv agent into the top frame.

(WXWebMyIssues.csv)

Operation: Runs from "Split Screen" url on reports.

Function: Generates a dynamic frameset and launches the WebMyIssues.csv agent into the top frame.

REQUEST CENTER

AdminProcessNow

Operation: Manual, Action Menu, Requires the PGADMIN role.

Function: Runs the WorkRequestProcessing agent immediately.

The primary purposes for this action are training or testing.

WorkRequestProcessing

Operation: Scheduled Agent, Run Hourly or Daily. Can also be run manually. This is a key system agent which must operate for the request center application to function correctly.

Function: This agent does many functions, but its primary purpose is to intermediate between the Request Center and the Repository. When a request is marked as "scheduled" this agent generates the repository assignments or project profile. When a request is "committed" this agent looks up the status of the repository work and updates the work request document with the current information. When the repository work is finished or canceled, the agent makes appropriate status changes in the work request and sends a final email to the requester.

For deferred work requests, this agent will put them back into active status when the restart date arrives.

XML Request Observatory Link

Operation: Scheduled Agent, Run Daily.

Function: Transmits a periodic report to observatory in which the content of the requested work is modeled as a single proposed project. Each request appears as a project milestone, each requester as a resource.

This allows the Observatory to see the workload implied by the pending requests.

(WebAgentStatusReport)

Operation: Used when the AgentStatusReport action is clicked by a web user from the costing admin profile document.

Function: Iterates over all database agents to get name and basic information, then generates an HTML page containing the data in tabular form.

(WebRoutingAgent)

Operation: Used when routing forms are submitted by web users using the pgRouteDraft and pgRouteDraftWeb forms

Function: Used to send email notifications when request documents are routed by web users.

(WebWRSetLogoAgent)

Operation: This agent is executed when the pgWRSetLogoForm that is composed by the "SetLogo" action on the Request Admin Profile Form is submitted.

Function: The agent removes any existing logo document and constructs a link to the new logo file attachment (or external file). This link is stored in the Admin form and is used by the \$\$NavigatorTemplate for RequestCenter to place the logo graphic on the Request Center home page.

REPOSITORY ARCHIVE

Archive

Operation: Scheduled Agent, Run Daily or Weekly.

Function: This agent performs all archiving tasks when archive project or archive deliverable request forms are pending. It may run for long periods of time if many projects are waiting for archiving or if the repository is very large.

AutoArchiveRequestAgent

Operation: Scheduled Agent, Run Daily.

Function: This agent scans the projects in the repository and identifies those which are ready to be archived according to the rules established in the archive admin profile document. When such a project is identified, the agent creates an Archive Project Request form for that project which will be subsequently processed by the Archive agent.

A project becomes a candidate for auto archiving only if the "Allow Archiving" checkbox is checked AND then project approval status is set to "FINISHED" or "CANCELED" AND the Effective date of Completion has expired by least the Automatic Archiving Delay set in the profile form (which is 14 days by default). Hence it is very unlikely that a project would be selected for archiving accidentally.

This agent relieves the user from the necessity to manually select projects for archiving, but does not prevent the user from still doing so.

(WebAgentStatusReport)

Operation: Used when the AgentStatusReport action is clicked by a web user from the Archive Admin Profile document.

Function: Iterates over all database agents to get name and basic information, then generates an HTML page containing the data in tabular form.

(WebARSetLogoAgent)

Operation: This agent is executed when the pgARSetLogoForm that is composed by the "SetLogo" action on the Archive Admin Profile Form is submitted.

Function: The agent removes any existing logo document and constructs a link to the new logo file attachment (or external file). This link is stored in the Archive Admin form and is used by the `$$NavigatorTemplate for (RepositoryCenterWeb)` to place the logo graphic on the Archive Center home page.

Other Archive Agents

All other agents in the archive are the same as those in the repository. Note that these are all "on demand" web agents used for reporting or form operations. There are no other scheduled agents except the two listed above