Agents

REPOSITORY

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

Admin\Delete Workplan Now

Operation: Manual, Action Menu, Required PGADMIN role.

Function: This action will remove all assignments of selected projects. It will remove the .cpf file attachment from the project profile. It will reset the project format to "Project Gateway".

Usage: Select using the checkmarks, one or more project profiles in the project profile or project dashboard view then run this command.

The purpose of this command is to allow administrators to remove workplans when the wrong plan has been uploaded, or during system testing. The correct way to remove projects is to archive them when they have been finished or canceled.

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\PostCostsNow

Operation: Manual, Action Menu, Requires PGADMIN role.

Function: Executes the ***PGCostingLink*** agent immediately.

Usage: When you need to get timesheet data into the costing center immediately and are unwilling to wait for the scheduled running of the agent.

Admin\PostTimesheetsNow

Operation: Manual, Action Menu, Requires PGADMIN role.

Function: Executes the ***PGPostem*** agent immediately.

Usage: When you need to post timesheet data immediately and are unwilling to wait for the scheduled running of the agent. This is most useful in training classes.

Admin\ProcessMetricsNow

Operation: Manual, Action Menu, Requires PGADMIN role.

Function: Executes the ***PGMetrics1, PGMetrics2 and PGMetrics3*** agents in order immediately.

Usage: When you need to advance the metrics process immediately and are unwilling to wait for the scheduled running of the agents. This is most useful in training classes.

Agent 1 creates the questionnaires for the metrics manager. Agent 2 processes those questionnaires and creates the questionnaires for the project managers and participants. Agent 3 processes those questionnaires and puts data into the Metrics Center application.

Admin\Recost Timesheets

Operation: Manual, Action Menu, Requires PGADMIN role.

Function: Operates on preselected timesheets. Clears the "processed" flag that is set by the Process Costs process so that the timesheet can be reexported to the costing center application.

This is useful when rebuilding a costing center. Note that a normal timesheet correction clears will be handled without the need for this action.

Note, you must also run the Admin\Process Costs or wait for the scheduled action to get the data transferred to the costing center

Usage: Select the timesheet to be recosted by setting checkmarks or using select all in the timesheets view.

Admin\Reset Reference Now

Operation: Manual, Action Menu, Requires PGADMIN role.

Function: Operates on preselected project profiles. Set checkmark next to project name(s) before using. Typically used for manually created projects, but can be used to update any project. Use with care.

For each selected project, this agent will set or reset the "reference" cost, effort, start and finish dates (fields pg_refwork, pg_refcost, pg_refstart and pg_reffinish) in every assignment document of that project and in the project profile document. The reference values are set to the current actual values if available, or to the scheduled values if no actual value is available. There is one special case in this process. If the assignment has an actual start, but does not have an actual finish, and if the planned finish is before the date of the actual start, then the reference finish is set to actualstart + one day. When done, the project reference finish will be the latest reference finish of any assignment and the project reference work will be the sum of the assignment reference work fields.

Admin\Reset Metrics

Operation: Manual, Action Menu, Requires PGADMIN role.

Function: Clears the "Metrics Processed" flag set in assignments and project profiles by the metrics agents. When this flag is cleared, these items will be recognized as requiring metrics processing by the Metrics 1 agent.

Operation: Select the project profiles or assignments, then run the action.

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\UndoObsolesence

Operation: Manual, Action Menu, Requires PGADMIN role.

Function: Will remove the flag that forces an assignment to be ignored as an obsolete assignment.

Usage: Very rare. Can be used to correct some effects of synchronizing with the wrong project plan.

Admin\UpdateChecklistsNow

Operation: Manual, Action Menu, Requires PGADMIN role.

Function: Executes the ***ProjectGo\Checklist Tracking*** agent immediately.

Usage: The Checklist Tracking agent processes notification and verification actions on completed checklist items. This allow you to do these actions immediately without waiting for the scheduled agent.

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.

Admin\UpdateOrganization

Operation: Manual, Action Menu.

Function: Does the same function as ***Maintain Organizations.***

Usage: Use when changes have been made in UserID for project or participants, Program or Organization setting for project or participants that must be reflected in the database today.

Admin\UpdateOrganizationTimesheet

Operation: Manual, Action Menu.

Updates Existing Timesheet documents to contain the current organization codes for the associated participants.

Usage: Use when recent Organization changes must be implemented in timesheets for reporting.

Admin\Virtual Timesheet Undo

Operation: Manual, Action Menu.

Function: Updates Assignment document that were modified by the Virtual timesheet agent to undo the effects created by that agent. Does not have any effect upon regular timesheets or assignments that were updated by regular timesheet posting. See the ***Virtual Timesheet*** Agent description.

Usage: Use when you don't like the effect of the aforementioned agent.

Capitalize\Set Cap=NO

Capitalize\Set Cap=YES

Operation: Manual, Action Menu.

Appropriate assignment documents must be preselected (checked) before executing these agents. This is best done with the Outline w\Hours view displayed (Project Assignments navigator panel selection.

Function: Every assignment contains a field which can be set to indicate that this work is capitalizable or not for accounting purposes. These agents simply set or clear this field for a collection of assignments, which is faster than opening the individual documents and changing the value. The only obvious effect of the capitalization field value is that actual hours recorded on assignments with the cap flag set are shown in the Cap Hours field of the Project Profile.

Graphics\CreateSelectedCharts

Operation: Manual, Action Menu.

Appropriate graphics chart documents must be preselected (checked) before executing these agents. This is best done with the Chart Definitions view displayed (Graphics Charts navigator panel selection.

Function: Each of the checked Chart definitions will be processed to prepare a viewable chart which is stored as a Adobe Acrobat PDF file attachment on the definitions document.

Graphics\Background

Operation: Automatic, normally on a Daily Schedule.

Prepare Charts in PDF format from Graphics Definitions.

Additional Info: Uses "temp files" to create file attachments. Must run as an unrestricted agent.

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, 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.

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.

MaintainOrganization

Operation: Scheduled Agent, Run Daily.

Function: Update Assignment documents to incorporate changes in Project and Participant Profiles for Program, Organization, and UserID.

This is a key system agent whose purpose is to keep existing assignment documents and certain forms of graphics chart definitions consistent with the current information found in the logically associated participant profile and project profile.

The fields that are maintained within the assignment documents are the following: projectname, programname (pg_prgname), organization name (pg_groupname), and the list of editors (pgx_editaccesslist). This list is updated to include the userid of the project manager from the project profile and the userid of the participant from the participant profile.

If a participant or project profile is deleted without deleting the associated assignments, the agent will not be able to find the referenced profile document. In this case, it will change the program and/or organization name for that assignment to "NONE". This will be very obvious in the database. This should never happen in a properly used system.

If it does happen, the first step is to determine how the problem happened, and then to recreate the missing participant profile or to retrieve the project profile from the most recent backup tape.

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.

Missing Timesheets

Operation: Scheduled Agent, Run once per week after timesheet are due. Typically set to run on Sat, Sun, Mon or Tues.

Function: This agent updates the "missing timesheets" folder and sends email reminders for people who have not submitted timesheet for most recently completed week.

This agent ignores participants who are marked as "No Timesheet Allowed" or "Via Portal" in their participant profile.

Emails are sent only if names are provided in the "Notify about Missing Timesheets" field on the participant profile.

Note that there is also an action on the repository administration form to create a list of participants with missing timesheets.

Missing Timesheets - Reminder

Operation: Scheduled Agent, Run once per week before timesheet are due. Typically set to run on Thurs or Friday. Optimum time is probably Friday at noon.

Sends email reminders to people who have ***not yet*** created timesheets for the current week. This is a "proactive" reminder to do your timesheet before it is due as opposed to the Missing Timesheets agent, which complains after it is due.

The agent does not declare a person as having a missing timesheet if there are no reportable assignments for that person and no overhead categories defined.

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.

PGCostingLink

Operation: Scheduled Agent, Run Daily. Can also be executed using the Admin\ProcessCostsNow agent menu command by people with the PGADMIN role.

Function: This agent creates costing center "billing records" from recently posted or corrected timesheets or approved charge slips. A billing record is created for each unique project for which a participant reports time on that timesheet. One timesheet may result in several billing records. When a timesheet has been corrected, new billing records will be created. The costing center application will recognize these corrected billing records and recalculate costs appropriately.

The date at which costing was done is stored in the timesheets and is displayed in one of the timesheet views.

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.

PGMetrics1

Operation: Scheduled Agent, Run Daily or Weekly. Can be executed by the Admin\ProcessMetricsNow function when used by a person with the [PGADMIN] role.

Function: This agent locates assignments, phases, and projects which have become finished since the previous time this agent was run. It then creates questionnaires intended for use by the designated Metrics Manager. It sends an email announcement to the email address provided in the Administration from for the Metrics Manager.

Each questionnaires provides space for up to 20 completed events.

PGMetrics2

Operation: Scheduled Agent, Run Daily or Weekly. Can be executed by the Admin\ProcessMetricsNow function when used by a person with the [PGADMIN] role.

Function: This agent locates questionnaires that were created by PGMetrics1 and have been completed by the Metrics Manager.

As it processes each completed questionnaire, the agent creates new project experience questionnaires for each selected project manager or participant that was designated by the Metrics Manager to provide information about the recently completed assignment, phase, or projects.

Processed questionnaires are deleted.

It sends an email announcement to the designated project manager or participant for each questionnaire that is created.

For each event, the agent accesses the Metrics Database to find the question or list of questions specified for that event. These questions are then assembled onto the new project experience questionnaire. Multiple events for the same person are consolidated when possible to minimize the number of questionnaires created.

PGMetrics3

Operation: Scheduled Agent, Run Daily or Weekly. Can be executed by the Admin\ProcessMetricsNow function when used by a person with the [PGADMIN] role.

Function: This agent locates questionnaires that were created by PGMetrics2 and have been completed by the designed person.

For each item on each completed questionnaire, the agent generated a new metrics data record and places it into the Metrics Database. This data record contains the response to the question and a collection of repository information such as actual work hours.

Processed questionnaires are deleted.

PGPostem

Operation: Scheduled Agent, Run Daily or Weekly. Can be executed by the Admin\PostTimesheetsNow function when used by a person with the

[PGADMIN] role. Should be prior to Maintain Dashboards, and PGCostingLink.

This is a key system agent which must be running for proper system operation.

Function: Posts all timesheets when they have been approved. The posting process takes the actual and remaining work for each reported task and updates the corresponding assignment document.

When posting is completed, a field (dtposted) is set to the current date. This date is displayed in one of the timesheet views.

A second function of this agent is to examine all "unlocked" timesheets and relock them if the unlocking was more than 6 hours old. This ensures that all corrected timesheet will become relocked, even if the user fails to use the relock function. When the timesheet has been relocked, it will be reprocessed by the PGCostingLink agent.

Posting must run before a person can create a subsequent timesheet

Project Go!\Assignment Tracking

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

Function: This agent does three important functions. It maintains the status code of unfinished assignment, it maintains the "predecessor" status, and it sends email reminders about upcoming and overdue assignments.

Function 1: This agent changes the status code of an assignment which should be marked as LATE but which is not so marked. This is necessary so that FUTURE assignments eventually become LATE when they are not otherwise modified. In Project Gateway 5.1 and later versions this status code is used to select work for the Future and Overdue views.

Function 2: The agent determines if the assignment has predecessors and, if it does, it evaluates the status of those predecessors. If this status has changed, it updates the "predecessors completed" message in the assignment document. This functions for both imported and user defined predecessor relationships.

Function 3: This agent sends the email reminders for upcoming, late start, or late finished tasks or key events according to the rules specified in the Participant Agent Profile and Management Agent Profile sections of the Participant and Project Profile forms.

Email will only be send when the following are true: a) there is a valid email address provided in the profile document and b) the assignment is associated with a project for which the policy allows project go notifications c) the associated project's approval setting is "approved" status (as opposed to proposed or canceled etc. status) and d) the conditions specified in one or more of the rules are met. e) the assignment is not a buffer assignment.

Email is only sent once for a particular combination of assignment and rule. If two or more rules are triggered at the same point in time, one email will be sent about that assignment to that participant or project manager. The assignment document is modified to contain the date on which each rule was triggered. This date is used by the agent to prevent repetitive reminders.

Project Go!\Assignment Transfer Agent

Operation: Scheduled Agent, Run Daily or Weekly.

Function: Transfers upcoming assignments to Notes Mailboxes where they appear in the ToDo list and can be synchronized to PalmPilots or IBM WorkPads. Does nothing unless the appropriate notification rule is selected on the participant profile.

Project Go!\Dashboard Transfer Agent

Operation: Scheduled Agent, Run Daily.

Function: Transfers a short summary of project dashboard status reports to your Notes Mailbox where they appear in the ToDo list and can be synchronized to PalmPilots or IBM WorkPads. Does nothing unless the appropriate notification rule is selected in the project or program profile.

Do not use this function unless you are able to deploy the Mailbox agent because it will clutter your Notes inbox with redundant messages.

Project Go!\Document Tracking

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

Function: This agent sends and 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.

Project Go!\InterProject and Calendar Agent

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

Function: This agent reviews all InterProject Dependency documents and Calendar Dependency documents. It will update these documents when the status of the relationship between the events changes. It will send notifications to managers if dependency relationships become invalid due to changes in the schedule of either the controlling or dependent project if this feature is selected on the form.

This agent also implements the "slaved" events feature of interproject dependencies so that changes in one project can be reflected by proxy events maintained in another project.

The agent will update dependency documents for changes in the project names of the projects.

Project Go! Mailbox Agent

Operation: This agent should only be run in a mailbox database.

This agent should be copied to the Notes Mailbox database of participants who use the assignment or dashboard transfer features.   It processes incoming messages from the repository and removes duplicates and outdated information.

IT IS DISTRIBUTED IN THE REPOSITORY DESIGN, BUT NOT INTENDED TO BE USED HERE. However, if it is executed in the repository, it will not do anything and will not create any problems because the scheduling is preset to never run, even if enabled.

ProjectGo\Checklist Tracking

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

Function: This agent scans all assignments in the database to find any that have items marked done that required notification or verification and sends the appropriate notifications to the project profile defined checklist verifier. It also scans all verification request documents to find those that have been approved or rejected, updates the assignments and notifies assignees when the request for verification was rejected.

Note that, if a verification is not received after a 14 day wait period, the system automatically marks the item as verified.

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 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.

ProjectWeb Publishing Agent

Operation: This agent should be set to run Daily or more frequently. However, unless you actually use ProjectWeb Publisher, then you should disable the agent as it has no other purpose. In order to function properly this agent requires that the Server API code be installed on the server.

Function: Process new projects and project revisions which have been uploaded from clients using the ProjectWeb publisher

Remove outdated Pdx files on the server

Operation: Scheduled Agent, Run Weekly.

Function: This removes any temporary work files (.PD0 etc.) left over by the graphics agents.

Timesheet Agent Auto-initiate

Operation: Scheduled Agent, Run Weekly, typically Thursday night.

Function: This agent creates timesheets for participants whose timesheet initiation field is set to "by Agent Notes", "by Agent Web" , "by Agent or User Notes" or "by Agent or User Web". A column showing these settings is found in the Participant Profile view.

Timesheets are preloaded with the "most appropriate" tasks taking into account the settings found in the Administration form, the project approval status, and the individual assignment level timesheet controls.

A timesheet will not be created except for the current period (that is the period which is current when the agent runs.

A timesheet will not be created if another timesheet already exists for the same period.

When a timesheet is created, an email will be sent to the participant which announces and provides either a doclink (for Notes Timesheets) or a URL (for Web Timesheets) to the timesheet document.

Virtual Timesheet

Operation: Scheduled Agent, Run Weekly, typically Monday or Tuesday.

Function: This agent will update assignments that were NOT worked on with zero hours records for the period of the most recent timesheet. When applied to tasks that have previous actual work, this records the "did not work" actual information which would be otherwise subject to interpretation by the project scheduling system.

This "zero fill" is important when you use a project planning system such as Microsoft Project 2000 or ABT Project Workbench that keeps track of work hours by period.

When applied to unstarted tasks, this agent has a subtle effect. The subtle effect is to adjust the allocation of the unstarted work as it appears in the workload forecast views, but not adjust the planned dates that exist in the assignment. In this way, the overdue work is pushed into the near future from a workload project perspective, but that assignment itself remains scheduled in the past as an overdue item.

This agent only affects assignments that belong to participants who submit timesheets. And it only modifies assignments with details for the dates of the most recent timesheet period.

The effect of the agent can be completely removed by the Admin\VirtualTimesheetUndo action which may be performed by a person with the [PGADMIN] role.

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.

(AskPGQuery)

Operation: Run on demand via URL and by the submission of forms that are dynamically created by this agent. There are many launch points in the Repository database, the first is the link connected to the word QUERY on the Repository Center Web Home page.

Function: This agent implements dozens of query functions to get project, program, participant and organization status, workload and task lists. It can also be used for very limited data entry.

(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.

Graphics\Create Selected Charts

Operation: This is a manual agent that is executed from the Notes Actions menu. Before selecting the action, you must preselect one or more chart definitions using one of the two Graphics views.

Function: This agent runs the three following agents to produce Gantt charts.

(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 holidays 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.

(ResQuery)

Operation: Invoked by URL.

Function: This is a specialized variation of the AskPGQuery agent which is used by the Resource Loading Report, Create a New Assignment form, and Resource Preview process to generate a table showing resource workload by period.

(RetrieveProjectFile)

Operation: Executed whenever a "Project Info Request" form is submitted by a web client. Can take 30 seconds or more to execute depending upon project size.

Function: This agent gathers information from the attached form and then calls the ServerAPI code. When the API process is finished the agent redirects the user to a page which opens a document containing the requested project data as a .CPF file attachment.

This is part of the ProjectWeb publishing function. Requires that the Server API code be installed on the server.

(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.

(TSRQueryOpenAgent)

Operation: Executed as part of the Web Timesheet Form. Used whenever a TimeSheetWeb document is opened. If this agent is not signed, then an error 500 will result when the user attempts to create or open a timesheet.

Function: Among other functions, this agent uses the list of selected candidates (from the checklist) to construct the timesheet data entry grid.

(TSRQuerySaveAgent)

Operation: Executed as part of the Web Timesheet Form. Used whenever a TimeSheetWeb document is submitted after editing.

Function: Among other functions, this agent creates the list of candidates for the user to select from. It does this by starting with the list provided in the HiddenWebTSCandidates view, and then eliminating any that do not meet the selection rules and project policy settings.

This agent will, when the web timesheet is submitted as complete, invoke the checksheet function to update assignment checklists and implement the project policy functions for timesheet notification.

(TSRWebTSRenewAgent)

Operation: Executed as part of the pgTSWebRenew Form which is displayed whenever the AddTasks action is selected on a Web Timesheet.

Function: Resets internal data to allow the redisplay of the assignment selection checklist.

(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 redisplays 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 and HTML page containing the data in tabular form.

(WebAssignmentChangeAgent)

Operation: Used as a query save agent on the pgReassignAsgProjectForma and pgChangeTasknameWeb forms to implement changes in taskname and project association for assignments for web users.

Function: Gets assignment identification from transaction form, finds all assignments for the task and updates each with the new settings.

(WebCorrectTimesheetAgent)

Operation: Used when the "Correct" or "Unlock" action on a Web Timesheet is clicked by a web user. The "Correct" action only appears after the Unlock action has been performed. The Unlock action appears only for users with the role of [PGADMIN].

Function: This agent will display a series of html forms. These forms contain submit buttons and hyperlinks that reexecute the agent.

When the timesheet is unlocked, the participant is notified.

The agent generated forms allow the user to change hours reported on the timesheet, add new tasks and overhead categories, and relock the timesheet. When each change is made, the related assignment documents are immediately updated.

When the timesheet is relocked, the original approver, if any, is notified, for review, but not approval or rejection. Use of this function is described in the chapter entitled "Timesheet Administration ".

(WebCreateAsgDepAgent)

Operation: Run when the form associated with the "Dep" action on the Assignment is submitted by a Web user.

Function: This agent validates the selection of predecessor and then implements the construction or predecessor and successor doclinks in all of the coworker assignments of both the predecessor and successor assignments. It creates/updates the pg_UserDefinedPredecessor fields used by the ProjectGo Assignment Tracking Agent and by the Project Gateway API to construct task dependencies in exported projects.

A separate, but logically equivalent implementation is provided in the Notes implementation of the Dep action in the design of the Assignment form. The same result is produced by either process.

(WebCreateAssignmentAgent)

Operation: This agent executes whenever "A New Assignment" form is submitted by a web user.

Function: The agent divides the specified work and creates versions of the assignment for each participant. Each is preset with the appropriate access control. Doclinks are constructed so that each assignment is linked to all of the others. A methodology link is created when requested on the new assignment form. Sets up the assignment history record.

If the option is selected, each assignee will be sent an email telling them of the new task.

When complete, one of the new assignments is redisplayed to the user.

(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.

(WebEarnedValueAgent)

Operation: Run by URL containing uniqueprojectid.

Function: Generates the Earned value details report in HTML format.

(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.

(WebExportTSFile.csv)

Operation: This agent is executed by a URL which is constructed by the $$Return formula of the WebTimesheetExportForm. It is not a QuerySaveAgent. The URL contains the specifications for the export process.

Function: When the agent executes it will either return an HTML table, or will set the content type of the output to (application/csv) and return a file to the browser. The agent name must be .csv in order to be properly interpreted by the browser as a file type.

The purpose of the agent is to extract detailed timesheet data from a collection of timesheets for a specific date range. This can be created in html table or comma separated value format with optional column headers. The result will be a series of records corresponding to individual tasks or overhead categories in the selected timesheets.

The agent has optimizations to greatly accelerate a resource specific export. The agent will also search and export archive data using the connection information set in the repository administration form.

There is a maximum number of records that can be exported in a single operation.

(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, Deliverables, Milestones, Selected Milestones, Project Assignments, Participant Assignments, and Assignments 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.

(WebMoveDeliverableAgent)

Operation: This agent is executed in conjunction with the pgMoveDeliverableWeb form which is displayed when the Move action is selected by a web user on a Deliverable document.

The agent executes once when the form is opened and once again when the form is submitted. An on form variable, aac, is used to determine whether the open or close process is to be executed.

Function: On open, it prepares a list of possible parents for the deliverable. This list excludes all the current children of the deliverable. The list is computed before the form is displayed to the user.

On submit, it interprets the selected action and modifies the deliverable to be a child of its new parent by modifying the $Ref field of the deliverable document and the first part of the deliverable name. If the action specifies that the deliverable is to become a first level deliverable, the $Ref property is removed and the deliverable name is modified to remove the previous parental prefix.

(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.

(WebMyAsgns)

Operation: Run by URL

Function: If called with a username, it determines if the user name that is specified corresponds to a unique participant name. If not, returns selection dialog. Once a participant name has been provided it determines if there are any assignments in the repository for that participant. If so, it uses the participant profile to get the organization name, and the redirects the browser to the Assignments by Person, All by Date view with the organization and participant name set as the start key.

If the participant has no assignments, it returns simple response page. This precheck is required to prevent a page not found error when no assignments exist.

(WebMyChecklists)

Operation: Run by URL from repository center, participant, project, program and organization profiles and by the submission of web checksheet forms. This agent will be invoked for checksheet subscription references and supports both narrow (cellphone) and normal screen formats.

Function: This agent produces and processes the Checksheet transaction. It also produces the IncompleteChecklistsReport.

Note that the web timesheet checksheet page is generated by the TSRQuerySave agent, but is processed by this agent. In this case the agent gets the $$Return text from the timesheet document, modifies the heading, and emits it to the user as the final response page.

When creating a Checksheet transaction form, the agent constructs the list of candidate assignments using the (HiddenTSWebCandidates) view and filtering for project approval by checking the project profile using the pgProjects view. It generates the report heading and then uses functions in the pgCkListSupport library to create the report body. The agent is reinvoked when the Update action is used. The returned form information if again processed by the pgCkListSupport library routine to update the assignments.

When creating an IncompleteAssignments report, the agent simply invokes functions in the pgCkListSupport library to emit the report.

(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.

(WebMyTimesheet)

Operation: This agent is executed by a URL generated in several places including the middle of the Web Repository Home Page (labeled as My Timesheet) and action items on the Participant profile and Timesheet Navigator page.

Function: The agent first determines the desired participant if necessary using the pgDirectory view. It then returns a participant selection page or opens the existing timesheet or creates a new timesheet, or displays an error page.

(WebMyWork)

Operation: This agent is executed by a URL. It is initiated by the My Work action from repository center.

Function: The agent first determines the desired participant if necessary using the pgDirectory view. It then returns a participant selection page or redirects the user to a URL which executes the (AskPGQuery) agent which produces a detailed list of current tasks for the participant.

(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.

(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 a 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.

(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.

(WebProjectTemplateAgent)

Operation: Run from URL and as a submission agent on the template editing forms. There are some 35 different entry sequences used to invoke this agent.

Function: This agent, in conjunction with the pgTemplateSupport, pgLoadingSupport, pgJournalSupport, pgGanttDataSupport, pgGanttSupport, pgAgentSupport libraries and the (ResUsage) agent implements the Schedule Template subsystem for creation, editing, display, generation and regeneration of project schedules.

The agent a set of multivalued fields in the template document as the place to store all information about the template assignments, documents, groups, and resources.

The visual display is created by computing a series of fields to hold html that renders into the Gantt style display. The buttons surrounding the image are rendered by the form, but invoke this agent for all actions.

Note that this agent implement the generation function by dynamically generating and processing the schedule generation options form.

The Notes client implementation of the generation function is implemented in the pgTemplateGNotes form and replicates part of the code of this agent.

Because of all of the dependencies it is essential that, if any of the libraries are changed that all of them and this agent be recompiled.

(WebProjectTemplateRptAgent)

Operation: Run by URL from actions on the template form.

Function: This agent produces the Instances Report and the much more complex Task Instances Report.

The Instances shows which projects have instances of the template. The Task Instance Report shows the planned and actual experience for each task generated from the template in all the projects in which it exists.

(WebPromoteDocAgent)

Operation: Runs as querysave agent from the pgPromoteDocWeb form.

Function: Converts a document version into a primary document by removing $Ref flags and version information.

(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 thoughout the repository. Requires that the user have [PGADMIN] rights.

(WebReassignNotify)

Operation: Runs as a query save agent from the pgReassignForm2 which is used by web users to commit a task reassignment.

Function: Implements the option email notification of the new participant.

(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.

(WebResEvalAgent)

Operation: Run by an embedded button on the Create New Assignment form.

Function: Creates a resource preview web page that shows the workload plan for each of the resources that have been selected on the Create New Assignment form. In addition to the existing projects, each has a row showing the new work that would be created if the assignment were created. The resource loading page is built dynamically and incorporates iframes. The URL's set into these iFrames execute the (ResUsage) agent to generate the table. The workload for the new assignment is passed in the URL to the iFrame.

(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.

Function: Implements the Unassigned Participants Report and Missing Timesheets reports from the repository administration form actions.

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

(WebSubSetup)

Operation: Executed as a query save agent on the pgCalSubscripeP 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 querysave 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.

COSTING CENTER

Admin\ProcessCostsNow

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

Function: Runs the ***ProcessCost*** agent immediately.

Admin\UpdateDashboardNow

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

Function: Runs the ***UpdateDashboard*** agent immediately.

Preload Projects and Participants

Operation: Scheduled Agent, Run Daily or Weekly.

Function: This agent automatically creates cost rate profiles for all repository participants and costing category profiles for all repository projects if they do not already exist.

The purpose in doing this is to allow the administrator to setup the costing information before any timesheets are actually produced. This increases the "up front" work, but greatly reduces the continuing administrative work.

By allowing the agent to run periodically, each new repository participant and project will be created in the costing system automatically.

All these new profiles will be "waiting for manual review" by the administrator.

Process Costs

Operation: Scheduled Agent, Run Daily. May also be run by the Admin\ProcessCostNow action if you have the role of [PGADMIN] in the costing database.

Function: This agent processes new billing records created by the PGCostingLink agent running in the Repository Center and creates Cost Records, new Costing Category Profiles, new Participant Cost profiles as necessary. The agent also process all pending records so that, when the administrator has marked a profile as ready for use, all billing records which were waiting for that profile to be completed will be processed.

Special Admin\ResetAllCostCalculations

Operation: Manual, Action Menu, Requires PGADMIN role. Requires second confirmation before operation. Would normally be used in a sequence - SpecialAdmin\ResetAll, Admin\ProcessCosts, Admin\UpdateDashboard.

Function: Removes all calculated costs from the costing database and resets the status of all processed billing records to "unprocessed". Once this action is done, all previous computed cost data is permanently lost.

Since all the billing records are now unprocessed, the next execution of the process costs agent will completely recalculate the results using the current cost rates and account specifications. And the subsequent execution of Update Dashboards will retotal all of these newly calculated costs onto the appropriate projects, participants, and account codes.

This is a special purpose tool, it has no "regular" production use. However if incorrect billing rates or account assignments have been made in the past, this allows you to regenerate all the results using the correct parameters.

UpdateDashboard

Operation: Scheduled Agent, Run Daily or Weekly. Best if run after ProcessCosts.

Function: This agent tabulates processed costs to create totals on costing category profiles, participant cost rate profiles, and account code documents. It also compares the totals to the specified targets. When targets are exceeded email messages may be sent.

Note: The total actual cost for projects is exported to the Observatory by the XML Repository Link Agent as the project actual cost.

(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 and HTML page containing the data in tabular form.

(WebCCSetLogoAgent)

Operation: This agent is executed when the pgCCSetLogoForm that is composed by the "SetLogo" action on the Costing 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 CostingCenter to place the logo graphic on the Costing Center home page.

METRICS CENTER

Admin\ProcessNow

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

Function: Runs the ***pgMtrProcessNewData*** agent followed by the ***pgMaintainProperties*** agent immediately.

The primary purposes for this action are training or testing. When used in conjunction with the *Repository Admin\ProcessMetricsNow* action, it allows you to run the entire metrics process in just a few minutes rather than waiting for scheduled agents to execute.

pgMaintainProperties

Operation: Scheduled Agent, Run Daily. May also be run by the Admin\ProcessNow action if you have the role of [PGADMIN] in the metrics database.

Function: This agent updates assignment and phase level knowledge records with information obtained from phase or project level questionnaires which come in at a later time.

pgMtrProcessNewData

Operation: Scheduled Agent, Run Daily. May also be run by the Admin\ProcessCostNow action if you have the role of [PGADMIN] in the metrics database.

This is the key agent of the metrics system which takes metrics data records created by PGMetrics3 running in Repository Center and constructs Metrics Knowledge Records. It also implements the QA process and processes data manually reviewed and approved.

pgQA

Operation: This is a web agent is used on the forms created by the Estimate and Query functions when run by a web client. Should be set to ScheduleNever.

Function: This implements the entire query process using the specifications provide on the submitting form. The result is an HTML report page returned to the browser.

(WebAgentStatusReport)

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

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

(WebMTRSetLogoAgent)

Operation: This agent is executed when the pgMTRSetLogoForm that is composed by the "SetLogo" action on the Metrics 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 MetricsCenter to place the logo graphic on the Metrics Center home page.

REQUEST CENTER

Admin\ProcessNow

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 and 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 and 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

ENTERPRISE TIMEPORTAL

Admin\ProcessTimesheetsNow

Operation: Manual, Action Menu, Requires PGADMIN role.

Function: Runs the ProcessTimesheets agent immediately.

Admin\Update Participants

Operation: Manual, Action Menu, Requires PGADMIN role.

Function: Runs the Maintain Participants agent immediately.

Maintain Participants

Operation: ScheduledAgent, Run Daily or may be executed by the Admin\Update Participants action.

Function: Scans all participants in all registered repositories and updates the participant registration records in this database according to the rules specified in each repository registration form.

Missing Timesheets

This is the same as the repository agent of the same name. See above.

Missing Timesheets - Reminder

This is the same as the repository agent of the same name. See above.

PGDocumentPurgeAgent

This is the same as the repository agent of the same name. See above.

ProcessTimesheets

Operation: ScheduledAgent, Run Daily or may be executed by the Admin\ProcessTimesheetsNow action.

Function: This agent finds approved master timesheets and creates "subset" timesheets in each registered repository for which work has been reported by the user. The agent maintains a list of such "subset" timesheets which it has created and updates the master timesheet when all of the subset timesheets have been posted.

(PGTSWebReject)

This is the same as the repository agent of the same name. See above.

(TSRQueryOpenAgent)

This is the same as the repository agent of the same name. See above.

(TSRQuerySaveAgent)

This is the same as the repository agent of the same name except that the task lists from all the registered repositories are consolidated into a single checklist for the user.

(TSRWebTSRenewAgent)

This is the same as the repository agent of the same name. See above.

(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 and HTML page containing the data in tabular form.

(WebCorrectTimesheetAgent)

This is the same as the repository agent of the same name with the exception that only overhead categories may be corrected, not tasks. Task correction must be done on the posted timesheet in the appropriate repository.

(WebExportTSFile.csv)

This is the same as the repository agent of the same name. See above.

(WebMyTimesheet)

This is the same as the repository agent of the same name. See above.

(WebSetLogoAgent)

Operation: This agent is executed when the pgSetLogoForm that is composed by the "SetLogo" action on the TimePortal 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 TimePortalCenter to place the logo graphic on the TimePortal home page.