Projects Extensions for Billing, Revenue Accruals and More: An NSF Case Study (v2)

Josh Scheumann
Zanett
Robert Kalvaitis
NSF International

Use Oracle Projects Extensions to handle complex Billing & Revenue Accrual scenarios with Billing Assignments, control and validate Project setups with Status Verification, and bill on variable dates with custom Billing Cycles. These "hooks" allowed NSF to implement Project Billing in a 20,000+ projects per year environment and are just three examples of solutions available in the Projects suite.

An Overview of NSF International

NSF International, The Public Health and Safety Company™, a not-for-profit, non-governmental organization, is the world leader in standards development, product certification, education, and risk-management for public health and safety. For 60 years, NSF has been committed to public health, safety, and protection of the environment. While focusing on food, water, indoor air, and the environment, NSF develops national standards, provides learning opportunities through its Center for Public Health Education, and provides third-party conformity assessment services while representing the interests of all stakeholders. The primary stakeholder groups include industry, the regulatory community, and the public at large.

NSF is widely recognized for its scientific and technical expertise in the health and environmental sciences. Its professional staff includes engineers, chemists, toxicologists, and environmental health professionals with broad experience both in public and private organizations.

NSF has earned the Collaborating Center designations by the World Health Organization (WHO) for Food and Water Safety and Indoor Environment. Serving manufacturers operating in 80 countries, NSF was founded in 1944 and is headquartered in Ann Arbor, MI USA. The NSF Mark is recognized for its value in international trade around the world and is respected by regulatory agencies at the local, state, and federal levels.

Oracle Projects – Setup & Use Details

NSF began its Oracle Applications implementation in January 2006. Partnering with INRANGE Consulting to lead it through its implementation, NSF devoted key process owners from all aspects of its business to be stakeholders in the setup, implementation, and testing of the Oracle solution. Oracle Projects is the underlying foundation of the operational aspects of NSF. Obviously, implementing Oracle Financials has given the Accounting department enhanced efficiencies and standards, but the Projects suite is where all of the operational transactions start and finish (up until the hand-off to Accounting is ready).

Project Billing and Costing are the main components of the suite that went live in December 2006. Project Management and Resource Management aspects may be added in 2007/2008, but the Billing components were the first critical components to get live. NSF handles over 20,000 projects per year with varying timing for both revenue recognition and invoicing (billing). Nearly half of the projects are billed in advance with revenue recognition either occurring at time of work performed or spread throughout the year (based on the type of service provided). Other projects are billed either on a routine basis, when specific tasks are completed, or at the end of the project (project completion).

The implementation of Oracle has allowed NSF to greatly automate much of the project management, revenue, and billing processes. Due to the nature of NSF's business, there are a number of proprietary systems that NSF uses to maintain its testing laboratories, product certification, and audit tracking systems. These systems have all utilized the Oracle Project APIs to feed project completion details into Oracle. Most of these are fed via the PRC: Transactions Interface process. Because of these feeds and the automation that NSF has been able to facilitate, a number of project extensions were also need to help validate data as well as trigger the AutoAccounting and billing functions (e.g., raising automatic Events).

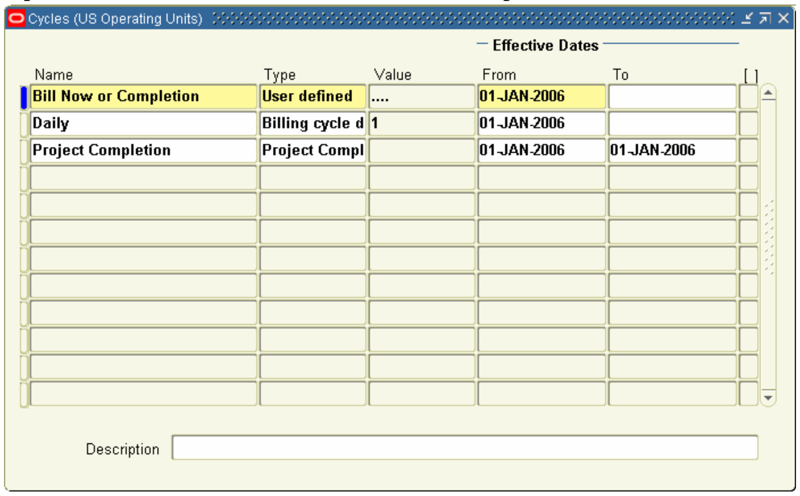
Project Extensions – Billing Cycles

NSF is utilizing both Work/Work and Work/Event type projects (the first being the revenue recognition method / the second being the billing method). The Work/Work scenarios are in need of a Billing Cycle extension so that project managers can simply change the status of the Project to trigger billing. The extension is fired with the Generate Draft Invoice process runs, which looks at the type of Billing Cycle attached to the project. The Billing Cycle extension will fire whenever it sees a user-defined billing cycle.

```
CREATE OR REPLACE PACKAGE BODY PA_Client_Extn_Bill_Cycle AS

Get_Next_Billing_Date (      X_Project_ID               IN  Number,
                             X_Project_Start_Date       IN  Date,
                             X_Billing_Cycle_ID         IN  Number,
                             X_Bill_Thru_Date           IN  Date,
                             X_Last_Bill_Thru_Date      IN  Date ) RETURN Date
```

To modify the "hook" that Oracle has provided, utilize the seeded PAXIBCXB.pls file that contains the PA_Client_Extn_Bill_Cycle package and Get_Next_Billing_Date function. These names cannot be changed nor the parameters of the function. Note that Oracle always recommends copying the file and renaming it. The function returns either returns NULL or returns a date. If NULL (no date) is returned, the invoicing process does not run and no invoice is generated. If a date is returned then that date is used as the next billing date and compared to the Bill Through Date parameter specified when the Generate Draft Invoice process is executed. A screenshot of the Cycles window is shown to display a User defined billing cycle, which must be created in order to have this "extension" evoked during processing.



Name	Type	Value	From	To
Bill Now or Completion	User defined	....	01-JAN-2006	
Daily	Billing cycle d	1	01-JAN-2006	
Project Completion	Project Compl		01-JAN-2006	01-JAN-2006

Project Extensions – Project Billing Assignments (Revenue/Invoice Generation)

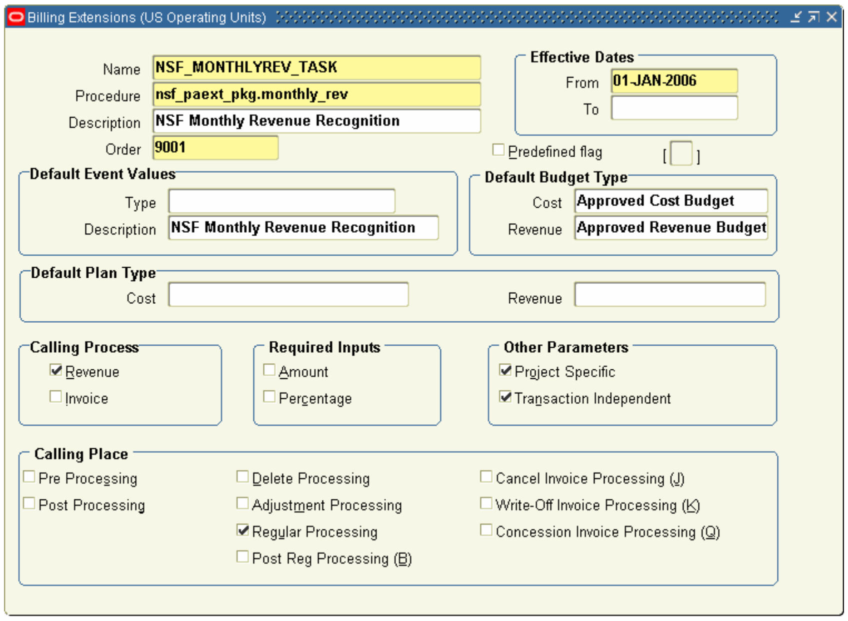
The Revenue and Invoice generation for each project is the critical piece to the Project Billing process. Due to the various timing scenarios as discussed earlier, NSF has a number of extensions for each. These extensions are driving the key functionality for NSF and the areas of greatest automation efficiency. Below is a display of these extensions.

Billing Extensions	Revenue Extensions	Revenue Extensions
NSF_PREBILL_TASK	NSF_PREREV_TASK	NSF_MONTHLY_REV_TASK
NSF_SPECIAL_BILL_TASK		NSF_SPECIAL_REV_TASK
NSF_POSTBILL_TASK	NSF_POSTREV_TASK	

The Oracle API file containing the user-exit for these extensions is the PAXITMPB.pls file (package body). It contains sample billing extension coding that can be used as a basis on which to build. The parameters of the procedures need to be kept as is (do not change them). A corresponding PAXITMPS.pls will also need modified (package specification).

```
CREATE OR REPLACE PACKAGE BODY nsf_paext_pkg AS

PROCEDURE event_blrv (    X_project_id                IN    NUMBER,
                          X_top_task_id               IN    NUMBER   DEFAULT NULL,
                          X_calling_process           IN    VARCHAR2 DEFAULT NULL,
                          X_calling_place             IN    VARCHAR2 DEFAULT NULL,
                          X_amount                    IN    NUMBER   DEFAULT NULL,
                          X_percentage                IN    NUMBER   DEFAULT NULL,
                          X_rev_or_bill_date          IN    DATE     DEFAULT NULL,
                          X_bill_extn_assignment_id   IN    NUMBER   DEFAULT NULL,
                          X_bill_extension_id         IN    NUMBER   DEFAULT NULL,
                          X_request_id                IN    NUMBER   DEFAULT NULL)
```



Name	NSF_MONTHLYREV_TASK
Procedure	nsf_paext_pkg.monthly_rev
Description	NSF Monthly Revenue Recognition
Order	9001

Effective Dates
From 01-JAN-2006
To

☐ Predefined flag [ ]

Default Event Values
Type
Description NSF Monthly Revenue Recognition

Default Budget Type
Cost Approved Cost Budget
Revenue Approved Revenue Budget

Default Plan Type
Cost
Revenue

Calling Process
☑ Revenue
☐ Invoice

Required Inputs
☐ Amount
☐ Percentage

Other Parameters
☑ Project Specific
☑ Transaction Independent

Calling Place
☐ Pre Processing
☐ Post Processing
☐ Delete Processing
☐ Adjustment Processing
☑ Regular Processing
☐ Post Reg Processing (B)
☐ Cancel Invoice Processing (J)
☐ Write-Off Invoice Processing (K)
☐ Concession Invoice Processing (Q)

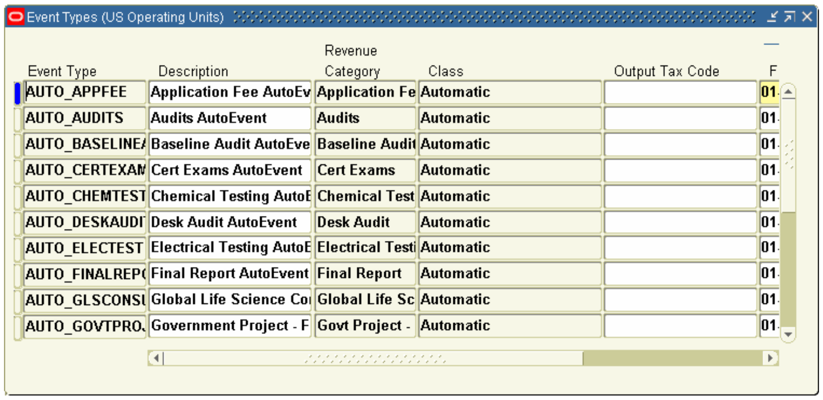
NSF chose to go the route of creating 1 main custom procedure that will execute during all Regular processing of the Generate Draft Revenue and Invoicing processes. Alternatively, much more module code could be generated with each extension calling a different procedure. NSF did not go this route do the large similarities in processing between each extension; however, a phase two plan does exist to evaluate the refinement of the current logic for a much more modular approach (without hard coding of extension names, etc). The custom procedure is linked to each billing extension that must set up in the system (via the Application's screens).

A few items should also be mentioned on the Billing Extensions setup screen itself. Note first that all populated fields(in the screenshot of the header portion as well as the *Default Event*, *Default Budget*, and *Other Parameters* regions are required if you are processing revenue and billing. The *Procedure* field is where the package and procedure name should be entered. This is your main link. The *Calling Process* checkbox will vary if the process you are calling is for Revenue, Invoicing, or both. The options in the Calling Place can also vary. For NSF, only *Regular Processing* is being utilized at this time.

Oracle also provides you the ability to raise Automatic Events from within the extension logic. NSF has utilized this API to raise events for Invoicing on Work/Event projects. This mainly comes into play for pre-bill scenarios and gives further flexibility in project setup. The end goal of these extensions is to make billing and revenue recognition as automated as possible with very little steps for the project manager in order to "trigger" either event to occur.

```
pa_billing_PUB.Insert_Event ( X_rev_amt        =>    l_amount,
                              X_bill_amt       =>    l_amount,
                              X_project_id     =>    X_project_id,
                              X_event_type     =>    l_event_type,
                              X_top_task_id    =>    X_top_task_id,
X_organization_id       =>      l_project_org,
X_completion_date       =>      l_master_date,
X_event_description     =>      l_event_desc,
X_status                =>      l_status,
X_error_message         =>      l_error_message);
```

In order to raise an event, the event type reference in the pa_billing_PUB - Insert_Event API must first exist in the billing Event Types setup screen. They must also be setup with a Class of "Automatic" and be associated with a Revenue Category. The category for NSF also drives the revenue account generation based on the AutoAccounting setups. An event description can also be passed in the API call. If one is not specified, then it will automatically pull it from the Description setup of the Event Type (see the screenshot).


Event Types (US Operating Units)

Event Type	Description	Revenue Category	Class	Output Tax Code	F
AUTO_APPFEE	Application Fee AutoEv	Application Fe	Automatic		01
AUTO_AUDITS	Audits AutoEvent	Audits	Automatic		01
AUTO_BASELINEA	Baseline Audit AutoEve	Baseline Audit	Automatic		01
AUTO_CERTEXAM	Cert Exams AutoEvent	Cert Exams	Automatic		01
AUTO_CHEMTEST	Chemical Testing AutoE	Chemical Test	Automatic		01
AUTO_DESKAUDI	Desk Audit AutoEvent	Desk Audit	Automatic		01
AUTO_ELECTEST	Electrical Testing AutoE	Electrical Testi	Automatic		01
AUTO_FINALREP(	Final Report AutoEvent	Final Report	Automatic		01
AUTO_GLSCONSI	Global Life Science Co	Global Life Sc	Automatic		01
AUTO_GOVTPRO.	Government Project - F	Govt Project -	Automatic		01

Project Extensions – Project Verification

Having seven or more Billing Assignment extensions can make setting up a project a little confusing to a project manager. The biggest question that comes into play is when can I use (or when do I need to use) certain extensions and when can I not. Obviously, Project Templates are your best friend in governing your setups. However, there are always scenarios that come up in which a template may not fit quite right and you need to add additional tasks to a project copied from a template. In order to police these scenarios, NSF utilized the Project Verification extension to "validate" the type of project and the billing assignment extensions attached to it (or not attached to it).

The PAXPCECB.pls file contains the pa_client_extn_proj_status package and Verify_Project_Status_Change procedure that needs to be modified in order to add custom logic during a change in project status. Note that the names of this package and procedure also cannot be changed nor the parameters of the function.

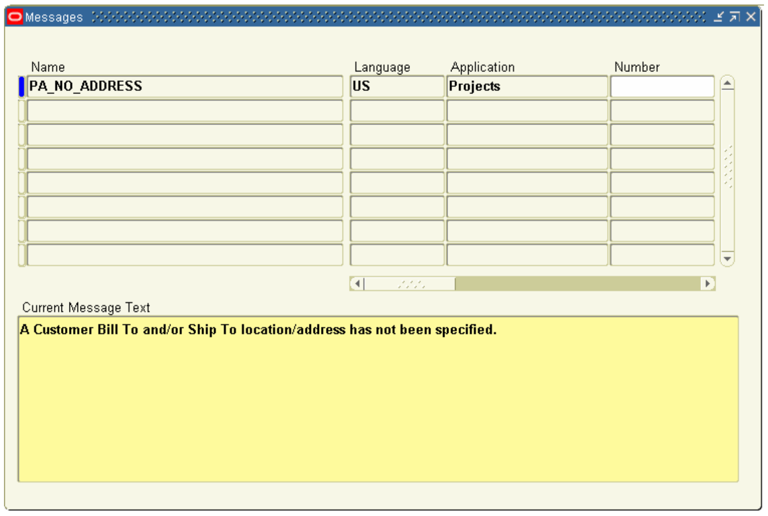
```
CREATE OR REPLACE PACKAGE BODY pa_client_extn_proj_status AS

PROCEDURE Verify_Project_Status_Change ( X_calling_module         IN    VARCHAR2,
                                         X_project_id             IN    NUMBER,
                                         X_old_proj_status_code   IN    VARCHAR2,
                                         X_new_proj_status_code   IN    VARCHAR2,
                                         X_project_type           IN    VARCHAR2,
                                         X_project_start_date     IN    DATE,
                                         X_project_end_date       IN    DATE,
                                         X_public_sector_flag     IN    VARCHAR2,
                                         X_attribute_category     IN    VARCHAR2,
                                         X_attribute1             IN    VARCHAR2,
                                         X_attribute10            IN    VARCHAR2,
                                         X_pm_product_code        IN    VARCHAR2,
                                         X_err_code               OUT   NUMBER,
                                         X_warnings_only_flag     OUT   VARCHAR2 ) IS
```

NSF needed to validate the extensions attached to each task instead of at the project level. Therefore, a cursor was needed to loop through all tasks on a project and throw out an error after the whole project has been validated. The logic can also be coded to only fire on either a change from a certain status or to a certain status (or both). NSF also added additional validation to the procedure so that it is also checking the extension assigned based on whether the project is Work/Work or Work/Event, the Project Type or Classification of the project, and the naming of the project Task.

There is a very nice function that can be called to display user friendly messages to the project manager so that they know exactly what is in error.

```
PA_UTILS.Add_Message
(p_app_short_name => 'PA',
 p_msg_name => 'PA_NO_ADDRESS',
 p_token1 => NULL,
 p_value1 => NULL);
```



Name	Language	Application	Number
PA_NO_ADDRESS	US	Projects	

Current Message Text: A Customer Bill To and/or Ship To location/address has not been specified.

The Messages displayed to the user can be entered through the standard Messages form in the Application to make these messages configurable or changeable without having to modify the code. A message variable is simply specified in the function that is then tied into the setup in the Messages form. This not only adds additional flexibility but helps to tightly tie in your customizations into the core applications and gives even your custom hooks the same look and feel of base Oracle logic.

Issues Along the Way

Most of the issues encountered through the extension development process was not necessarily in the development itself but more in the testing / processing phase. Several features were uncovered that were not 100% clear in the supplied documentation as to the detailed specificity of the functionality.

The largest of these issues was with the Billing Cycle extension. NSF has several scenarios where the invoicing should not be triggered until the project status is changed to a certain value. What was found was that the invoicing was generating as soon as the first revenue transaction existed for the project (i.e., work was performed). It was discovered through several Service Requests via Oracle Support and some conversations by them explaining the specific functionality and documentation that the Billy Cycle extension only comes into play once an invoice exists in a Released status (Approved and Released). After that initial invoice has been released, then the billing cycle extension works as he had originally anticipated. NSF has not yet devised a solution to this scenario and is currently using a procedural workaround for resolution.

Another issue NSF is having is with the differences between the two invoice creation processes: *Generate Draft Invoices for a Single Project* and the *Generate Draft Invoices by a Range of Projects*. The "single" process affectively deletes all "Unapproved" invoices and regenerates them with all work from the original (deleted) invoice and any new work that has been processed since then. This is how NSF needs the functionality to occur since they have a large number of projects in process throughout the year and do not approve all invoices on a daily basis. However, the single process is not feasible to run except in one-off scenarios. Therefore, the "range" process needs to operate in the same manner only in a mass project invoicing fashion. Again, through several Service Requests, it was found that the range processing is slightly different from the single. It does not delete the existing invoices. If any are found for a project, it skips generating any new invoices for that project even if un-invoiced and invoicable revenue exists. This was confusing as the parameters for each process are essentially identical. Oracle's recommended procedural workaround was to first run the Delete Unreleased Invoices process but to also set the parameters so that it deletes Unapproved invoices as well. This worked from a deletion standpoint, but the new invoices were not generated when the generate process was run again. This issue is still in SR's hands (in development)…..**at time of delivery this has been resolved and is working.**

The only other issues are really more of utilizing functionality in two additional extensions that did not make the phase one list. The first of these is to create invoices with Receivables (AR) Transaction Types that vary based on certain criteria instead of having just one Transaction Type of "Project Invoice" (default) from Projects (PA) to AR. For example, NSF would like to create separate Transaction Types per Project Type. This will allow AR to have greater flexibility in controlling invoice processing by transaction type (e.g., print invoices by transaction type, generate AR statements and finance charges by transaction type, and print dunning letters by transaction type) as well as enhance reporting on the core accounting side (outside of Projects).

The last issue on the phase two list is to utilize extensions and workflows to control the creation and approval of credit memos within Projects. Credit memos are in an Approved status automatically upon creation. This default process needs to be changed to allow workflow to dictate approval (and rejection) of credit memos so that creation has greater control and eliminates further paper trail.

Conclusion

NSF implemented the Oracle E-Business Suite as its ERP foundation in order to gain numerous efficiencies across its organization. Projects are the key operational aspect of its business that really drives all transactional processing. The accounting department and other back-office functions have gained substantial functionality as well, but the Projects suite is the source of all activity. Oracle Time and Labor (OTL) timecards as well as Internet Expenses both require projects for entry as well. What this means is that NSF drives its entire business off of administrative (internal), contract, and capital projects.

The use of project extensions is invaluable in order to get everything out of the Oracle suite that NSF needs it to do. Tying in each of NSF's third-party systems to Oracle is extremely beneficial. However, even that would not have added near the value it has if they were not able to automate the revenue and billing routines that need to be triggered based on the work performed by those systems. Because of the Billing Cycle, Billing Assignment, and Project Verification extensions, NSF was able to create a system that puts Project Managers in control of just that – managing projects. The PM for a project focuses on getting the project set up correctly and ensuring all tasks are done on time and correctly and doesn't have to worry about the timing of revenue or when a project is invoice eligible. They still have to approve the invoices once they are generated so that they can hand them off to AR for release and printing, but with the extensions in place, now those unapproved invoices are sitting in the project manager's queue waiting to be approved.

Addendum – Listing of Oracle provided Extensions within Projects

Below is a quick listing of the extensions available within the Projects suite.
Note: Each extension has a S and B (suffixed) file – one for the package specification and one for the package body.

Client Extension	**File(s)**	**Use**	**Description**
Allocation	PAPALCCS (and B)	Costing	Expand allocation functionality
AR Transaction Type	PAXITRXS (and B)	Billing	Determine (set) the AR trx type
Archive Custom Tables	PAXAPVXS (and B)	Foundation	Archive custom tables during standard archiving process
Archive Project Validation	PAXAPVXS (and B)	Foundation	Add business rules during validation
Asset Allocation Basis	PACCXAAS (and B)	Costing	Custom allocation for unassigned and common costs across projects
Asset Assignment	PAPGALCS (and B)	Costing	Executes if Generate Asset Lines process is unable to assign an asset
Asset Lines Processing	PACCXACS (and B)	Costing	Create assets and asset assignments automatically based on criteria
Assignment Approval Changes	PARAAPCS (and B)	Resource Mgmt	Enforces conditions to determine if approval is required
Assignment Approval Notification	PARAWFCS (and B)	Resource Mgmt	Change list of default contacts used by the assignment approval flow
AutoApproval	PAXPTEES (and B)	Costing	Autoapproving for Expense reports and Timecards
Automatic Invoice Approve/Release	PAXPIACS (and B)	Billing	Make auto approve and release part of the Draft Invoice process
Billing Cycle	PAXIBCXS (and B)	Billing	Derive the next billing date
Billing Extensions	PAXITMPS (and B)	Billing	Implement specific business rules to create auto revenue/billing events
Budget Calculation	PAXBCECS (and B)	Project Mgmt	Control the processing of budgets and forecasts
Budget Verification	PAXBCECS (and B)	Project Mgmt	Use rules to validate a forecast or budget when submitted or baselined
Budget Workflow	PAWFBCES (and B)	Project Mgmt	Customize workflow for changing the status of a budget or forecast
Burden Costing	PAXCCEBS (and B)	Costing	Override a burden schedule ID
Candidate Notification Workflow	PARCWFCS (and B)	Resource Mgmt	*Customize the candidate workflow processes
Capital Event Processing	PACCXBCS (and B)	Costing	Create assets and asset assignments automatically based on criteria (PRC:Create Periodic Capital Event)
Capitalized Interest	PACINTXS (and B)	Costing	Customize the capitalized interest calculation process
CIP Account Override	PACCXCOS (and B)	Costing	Override the CIP account and/or CIP clearing accounts
CIP Grouping	PAXGCES (and B)	Costing	Define how expenditure lines are grouped to form asset lines
Commitment Changes	PACECMTS (and B)	Foundation	Check commitments during PRC: Update Project Summary Accounts
Control Item Document Numbering	PACINRXS (and B)	Project Mgmt	Custom logic to derive numbering of issues and change documents
Cost Accrual Billing	PAXICOSS (and B)	Billing	Apply specific business rules to cost accrual procedures
Cost Accrual Identification	PAICPCAS (and B)	Billing	Identify cross charge projects that use cost accrual (during Rev gen)

Client Extension	File(s)	Use	Description
Cross-Charge Processing Method Override	PACCIXTS (and B)	Costing	Implement specific business rules during cross charge processing
Custom Performance Measure	PJISCO1S (and B)	Project Mgmt	Create own custom measures on which to report project performance
Depreciation Account Override	PAXCXDES (and B)	Costing	Programmatically derive the depreciation expense account
Descriptive Flexfield Mapping	PAPDFFCS (and B)	Foundation	Control mapping of flexfields from AP to PA and PA to AP.
Estimate to Complete Generation Method	PAFPFGCS (and B)	Project Mgmt	*Control the calculation of ETC quantities and amounts in forecasts
Funding Revaluation Factor	PAXBFRCS (and B)	Billing	Apply funding revaluation factor to the funding backlog amount or to implement escalation indices
Issue and Change Workflow	PACIWFCS (and B)	Project Mgmt	Customize workflow for submitting and approving change documents
Labor Billing	PAXICTMS (and B)	Billing	Derive labor billing amounts for individual labor transactions
Labor Costing	PAXCCECS (and B)	Costing	Derive raw cost amounts for individual labor transactions
Labor Transaction	PAXCCETS (and B)	Costing	Create additional transactions for individual labor items charges
Non-Labor Billing	PAXINCTS (and B)	Billing	* Derive billing amounts for individual non-labor transactions
Overtime Calculation	PAXDLCOS (and B)	Costing	Utilize custom overtime calculation policies
Output Tax	PAXPOTXS (and B)	Billing	Custom method to assign default output tax codes for invoice lines
Post-Import	PAXTTRXS (and B)	Foundation	Run after Transaction Import
Pre-Import	PAXTTRXS (and B)	Foundation	Run before Transaction Import
Project and Task Date	PAPMGCES (and B)	Foundation	Update and control the numerous start & completion dates on projects
Project Performance Status	PAPESCLS (and B)	Project Mgmt	*Customize logic involved in retrieving overall performance status
Project Security	PAPSECXS (and B)	Foundation	Override default project-based security
Project Status Inquiry	PAXVPS2S (and B)	Project Mgmt	Derive an alternate column value in the PSI columns or override totals
Project Status Report Workflow	PAPRWFCS (and B)	Project Mgmt	Customize workflow for submitting, approving, & publishing status rpts
Project Verification	PAXPCECS (and B)	Foundation	Apply business rules during status change
Project Workflow	PAWFPCES (and B)	Foundation	Control workflow for project status changes
Provider and Receiver Organizations Override	PACCIXTS (and B)		Enforce cross-charge rules at higher level than level at which resources and projects are assigned
Receivables Installation Override	PAPARICS (and B)	Billing	Allows use of a 3rd party AR system yet allow importing of customer data from Oracle AR
Retention Billing	PAXBRTCS (and B)	Billing	Define specific rules to bill withheld amounts
Transaction Control	PAXTTCXS (and B)	Costing	Define rules to control expenditure entry policies (before commits)
Transfer Price Determination	PAPTPRCS (and B)		Enforce different transfer price rules

Client Extension	**File(s)**	**Use**	**Description**
Transfer Price Override	PAPTPRCS (and B)		Enforce different transfer price rules to determine the override amount
Transfer Price Currency Conversion Override	PAPMCECS (and B)		Use to override the default attributes that calculate the transfer price (transaction to functional currency)
Verify Organization Change	PAXORCES (and B)	Foundation	Apply business rules when an organization change is made
Workplan Workflow	PAXSTWCS (and B)	Project Mgmt	Customize workflow for submitting, approving, & publishing workplans