ITIL System Documentation

Last Updated: June 22, 2016

www.agiloft.com/documentation/itil-kb-documentation.pdf

Table of Contents

PROJECT STATEMENT

Introduction

The ITIL knowledgebase provides all the functionality to manage a complex IT organization, including Service Request Management, Incident Management, Problem Management, Change Management, Configuration Management, Purchase Management, Project Management, and more.

Our goal has been to provide ready-to-go, out-of-the-box ITIL-compliant structures and process flows for managing IT services, while enabling complete extensibility to meet the needs of any particular organization.

ITIL is a set of best practices intended to improve IT service while reducing failures and costs. The actual implementation of any ITIL process is open to a wide range of interpretations. While the basic structure of Service, Incident, Problem, Change, and Configuration Management is likely to be part of any ITIL implementation, the services that fall under each category, the relationship of the Service Catalog to these various processes, the ways requests are structured and managed and many other details will vary widely based on the needs and preferences of a particular organization, as well as the functionality of the particular software program used to implement ITIL.

Companies may implement ITIL along a spectrum moving from simple to complex -- from a fairly streamlined, direct approach, to a more “controlled” approach, with more process steps and approvals required for relatively simple tasks. Based on our years of experience implementing IT service solutions that people actually use, we have designed our out-of-the-box implementation to keep things as simple as possible within the ITIL framework.

For instance, we have chosen to put requests for standard changes that require no special approvals into the Service Request table rather than the Change request table, keeping the workflows for each request type more distinct and simpler. Services such as New Employee Setup and Password Resets are therefore handled within Service Requests.

We have also designed the system so that if a person has a problem with his printer, it does not **necessarily** require the creation of an incident, a problem, and a change request, just to get a new ink cartridge installed, something that at least one competitor has defined as the “correct” ITIL process. However, if you want to follow this more extended process, we have made it as easy as possible – from any record a button can be clicked to create the related records and map field values from the current record. There are no duplicate text entries when creating problems, change requests and incidents.

Since there are so many different ways of implementing ITIL, this document is a detailed guide to how it has been implemented in the out-of-the-box Agiloft ITIL KB. We have pre-built the complex relationships and functions that many companies may want, while trying not to force too much complexity on those who may prefer a nimbler and more efficient implementation. This is a rather difficult balancing act, and while we have done our best to get it right for the largest number of customers, the real power of the system is in how easy it is to change it to adapt to your company’s specific preferences and needs.

We will offer some guidance throughout this document to where to go to make changes to the critical relationships and behavior to suit your needs.

Groups and Teams

A first step in defining your processes and customizing the system is to consider the different sets of users who will be using the application and what kinds of access they will need.

Users in Agiloft belong simultaneously to both groups and teams. A user can belong to multiple groups (receiving the superset of those groups' permissions) and to a primary team with additional teams. A user's access to the system – the tables and tabs he sees, the records he sees, the fields he sees, the records he can create and edit, and the menu actions he can perform - depends on his group memberships. While you can create as many groups as you need, it is preferable from a maintenance perspective to keep the number of groups small.

A user's primary team determines what look and feel scheme he sees – so you can have customers on different teams actually seeing a differently branded interface with different logos and colors. Staff Teams are generally used to define functional groups to whom tickets will be assigned and emails sent.

In brief, groups determine the content of what members see. End user teams determine look and feel while staff teams define working units.

This section describes the different sets of users and the default breakdown of users into groups and teams.

Terminology

A note about terminology: We use the term **end user** to mean users who access the system through the end user interface, a simplified interface that allows them to create records of any kind, view any records made available to them, edit records defined as their own, and view any FAQs made available to them. These users cannot edit records defined as belonging to other people and they use the unlimited end user license.

We use the expressions “End User” or “Customer” interchangeably in this document to refer to company employees whose main role in the system is to make requests on their own behalf or for someone else (typically their supervisor or supervisee).

We use the term **staff** to indicate the people who are working on other people's issues – they may be solvers, technical support staff, IT staff, approvers, developers, sales reps, managers, or any other types of users who access the system through the staff interface.

“Technician” may also be used to refer to members of the IT organization or other teams that will be responsible for handling, creating, or responding to requests submitted by customers or other technicians.

Both end users and staff users may be employees of your company. Staff users require their own named license or may share a concurrent staff license.

Groups

This table lists the default groups that have been set up for the ITIL KB and describes the general permissions of each. To see more information about a group’s permissions, you can print out or save to a file the full details of that group's permission by going to Setup/Access/Manage Groups, selecting the group, and mousing over the printer icon to choose the printout you want.

Groups	Type	General description of their access permissions
Admin	Staff	These users can edit the system including rules and workflow, and can view, change and delete all records
Approver	Staff	This group holds people who can approve Contracts or Change Requests. Approvers will primarily interact with their Approval records, but they can also view change requests for which they are an approver. They can also view and edit Contracts for which they are an approver, and can view tables necessary to approve Contracts such as Steps, Approvals, Approval Templates, and Companies.
Base ServiceDesk	Staff	This group has the base permissions that should apply to all more privileged groups dealing with the Service desk tables. Users in those groups should also be made a member of the Base Servicedesk group. This group has full create/edit access to records in the Service Request, Incident, Problems, and Tasks tables, and create/edit own access to Change Requests and time entries. It has full view access to Configuration Items, Services, Companies, and Employees and can edit its own employee record, but has no other create or edit access in those tables. It can create/edit end users (external customers). It cannot delete records.
Document Creator	End User	Can create documents and edit their own – customer of document table.
Document Manager	Staff	People who can approve and publish documents. Cannot modify status of document records manually.
Document Reviewer	Staff	Can edit approvals for which they are the approver.
Service Manager	Staff	For staff responsible for maintaining the Service Portfolio (Service table) and the Task Workflows/Templates table. Only Service Manager can create new Services.
Change Manager	staff	This group is responsible for management of Change Request records and has full privileges on the Change Request table. Members can create, edit, and delete records in this table and will typically be Change Managers or Change Owners. They can also create task and approval workflows for Change Requests, and can edit Change Request related services.
Configuration Manager	staff	This group has full access to the Configuration Item records and is responsible for creating, editing, and deleting those records. People responsible for working on and configuring CIs, managing CI resources, and so on would typically be in this group. They might also be added to the Service Manager group if they are responsible for setting up change request workflows or services related to CI's.

Procurement Group	Staff	This group is responsible for managing the Purchase Request and Item tables.
Internal customer	End user	Internal Customer in employee table, can request Service Requests, Purchase Requests, and report Incidents, as well as see their own Configuration Items and edit some of their profile information and view other employee contact information. They may also access Knowledge FAQs.
Contract Creator	End User	For internal employees who can create contracts.
Contract Manager	Staff	This group has full access to the Contract table, Approvals table, Steps and Approval Workflow tables, and Companies table. They also have some access to End Users and Employees. They are responsible for creating, editing, and approving contracts for customers or the company.
Contract Owner	Staff	This group has a subset of the permissions Contract Managers do. They are responsible for Contracts assigned to them, and have full permissions there, but can only view Contracts that they did not create or were not assigned to. They have all the other permissions necessary to allow them to use Contracts effectively.
Customer Manager	Staff	Customer Manager – relevant if providing external customer support. Customer managers can view all support cases for their own company.
Customer	End user	Unused unless providing external customer support. Then this group is used for end user customers, who can submit and view their own support cases.
Sales	Staff	This group is responsible for recording information regarding sales efforts to specific companies as well as Purchase Orders made. They can also create and update Support Case records for the companies they represent. They have full access to: Company, Contract, Lead, Opportunity, and PO tables. Partial access to Campaign, Product, Product Quoted, Project, Quote, Support Case, Tasks, Teams, Time Entry, People: End Users.
Marketing	Staff	This group is responsible for coordinating and recording information about marketing campaigns and providing quotes to prospective customers. They have full access to: Campaign, Company, Lead, Opportunity, and Product tables. They have some access to Product Quoted, Quote, Tasks, Teams, Time Entry, People: End User, and People: Employee tables.
Guest	End user	This group may be used in hyperlinks to allow creation of new requests of any kind (such as leads, users, Incidents) without seeing the rest of the user interface
Anonymous	Staff	This group is used to enable unregistered users to click on an email hyperlink sent in an outbound email in order to edit that record. It is

		used in conjunction with the Anonymous user. If all your users reside in the system, you will not need this group.

Teams

Teams serve a different purpose for customers/end users and for staff users. For end users, unless you need to provide multiple branded interfaces to different sets of users, or expect your end users to access the system in multiple languages, it is simplest to put them all in one team with the word 'customer' in the title, such as Customer Team or Internal Customer Team.

If you need to provide a custom look and feel and customer branding to different sets of users, then you will need a separate customer team to go with each look and feel scheme. Teams can also be associated with a different default language, so it would make sense to have language-based customer teams if you were planning to run the program in a multi-lingual environment.

For staff users, teams are used to identify the functional units to whom records may be assigned. You will want a staff team for each assignment group (sometimes called a Queue). Teams can be hierarchical, so you may set up a hierarchy such that you can send an email to a mid-level team and the members of its subteams will receive the email. Users have one primary team, which defines the look and feel they see and their default table views. They can be members of as many additional teams as needed, so they are cc'd on emails and included in the assignment list for items assignable to those teams.

Teams Table is Special

The Teams table is a system table with some special hard-coded fields, such as the Working Hours and language fields, the team name, label, description, Team Leader, and some screen refresh options.

It also has some custom fields that can be modified, and you can add your own custom fields to the teams table to manage any other information associated with your teams. If you edit the teams table setup, you will only be able to modify the custom fields and their layout options.

Creating new teams and accessing all of their attributes is done through the Setup/Access/Manage Teams screen. Editing just the custom fields and adding users may also be done through the Teams tab on the left toolbar.

Managing Team Membership

Team membership is controlled by two fields in the user record that are linked to the Teams Table: Primary Team (single choice), and Teams (multi-choice). We recommend adding a user's Primary Team to the multi-choice Teams field along with any additional teams. This way a user's entire team membership can be found in a single field, which makes searching and filtering easier.

When you initially import users into the system, you can import their primary team and teams' values, with multiple comma separated values with no spaces. Once users are in the system, you may update team membership by editing specific users or by editing the team through the Teams tab, and looking up and importing users for the embedded Primary Team Members or Team Members tables shown for the Team. Looking up and selecting a user for either of these tables changes the linked field in the user's own user record to point to this team.

Default Teams

The teams shown below have been set up as the default teams for the application. You can delete any teams you do not need and you can rename any team to match your own company's naming conventions. You can of course also create any additional teams you need. This is best done by going to Setup / Access / Manage Teams and either editing a team or creating a new one there.

Teams	Description
1st Level Support Team	First Level Support Team
2nd Level Support Team	Second Level Support Team
Admin Team	Used for system notifications about rule, email and other errors
Backup and Storage Team	CI Backup and Storage Team
Change Approver Team	Change Approver Team
Change Management Team	Change Team
Compliance Team	Compliance Team used in contract approvals
Configuration Management Team	CI Team
Contract Management Team	Contract Management Team
Contract Owner Team	For staff contract owners/buyers
Custom Applications Team	CI Software Team
Customer Team	External Customer Team
Database Team	CI Database Team
Desktop Applications Team	CI Desktop and Support Team
Document Creator Team	Document Creator Team for internal end users. If using external users, then change the parent team.
Document Management Team	Team for Document Management
Document Reviewers Team	People who have been identified to review documents.

Facilities Team	Facilities support team
Finance Team	Finance team used in contract approvals
HR Team	Human Resources Team. Acts on new employee Service Request Tasks.
Internal Customer Team	Internal Customer Team
Knowledge Team	Knowledge Team for publishing FAQs
Legal Team	Legal Team approves contracts
Marketing Team	Marketing Team
Network Operations Team	CI Network Team
Office Mgmt Team	Office mgmt support team
Professional Services Team	will be assigned to projects
Project Manager Team	Project Manager Team
Purchasing Team	Purchasing Team
Risk Team	Risk team approves contracts
Sales Team	Sales Team
Security Team	CI Security and Support and Change Team
Server Team	CI Server Team
Service Management Team	This is the primary team for Service Managers
System Administration Team	CI Sys Admin Team
Vendor Management Team	Vendor Management Team
Vendor Team	Container for external Vendors - used in Contract Management

DEFAULT CONFIGURATION OF TABLES AND PROCESSES

Overview

The default configuration includes several global tables that are used by more than one process, several process tables, and several background tables that provide data used in other tables.

We will start with an overview of the process table structure for the main ITIL support operation tables, then look at the setup for the most critical "global" tables: those controlling tasks and approvals, since both tasks and approvals are used in the support operation tables, as well as for other processes in the system.

Then we will look at some of the other processes included in the default out-of-the-box system, such as contract management and CRM automation.

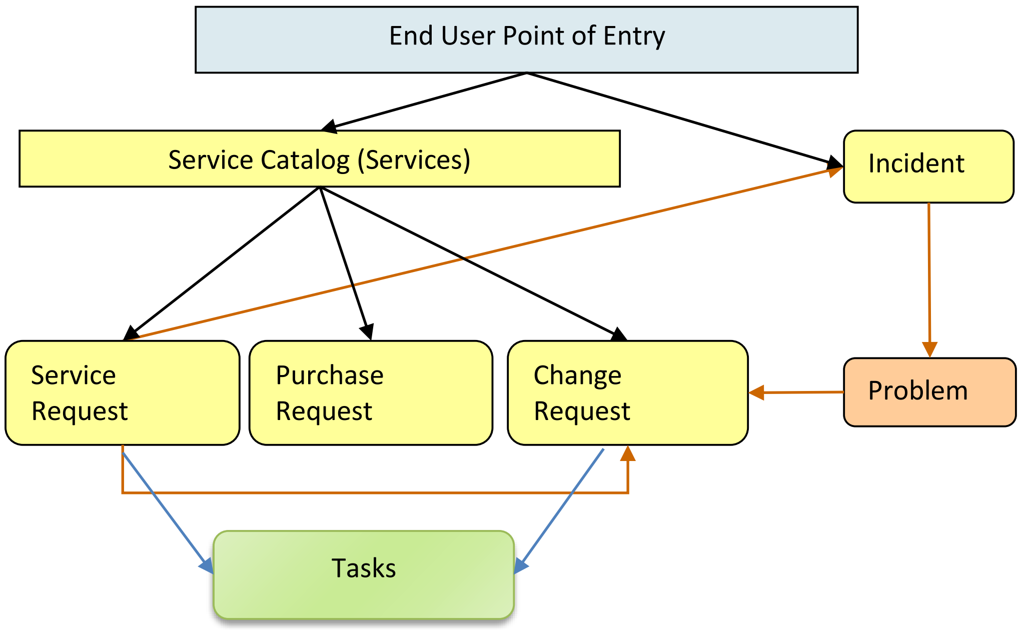
Request Tables Overview

Deciding how to segment the service catalog and what table structure to use for an ITIL implementation can be quite a challenge. In defining the default table structure, we had a few basic goals:

1) To make it easy for end users (the customers) to know where to go to request what they need and to follow up on their requests
2) To make the backend processing and workflows as simple and clear cut as possible for the technical users and approvers who must process the requests
3) To make the system easy for administrators to understand and maintain so that changes may be easily made to one kind of request and how it is processed without affecting everything else

The default setup aims to find a balance between all three of these goals.

We have chosen to segment customer requests into a few distinct request types and tables illustrated in the diagram:



The service catalog offers services to the end user that may be created in one of three tables: Service Requests, Purchase Requests, or Change Requests. Whether or not a particular group of users can request particular services can depend on their group permissions – for instance, our default internal end users cannot directly create a Change Request, while technical users can. An end user may also submit an incident to report a problem or interruption to a business or technical service. If merited, that incident may lead to the creation of a Problem and a Change Request by technical staff.

Service Requests and Change Requests may include a set of tasks that need to be completed before the request can be completed.

The system may be configured so that end users simply create records directly in any of these tables and choose the appropriate service from the service catalog once in the record, or it may be set up to offer users a set of hotlinks to create requests for specific services without them needing to know what table will be used to store and process the requests until after they have created them.

The separation of requests into these tables makes maintenance and processing simpler than if they were all in a single table, since the fields, form layouts, access permissions, workflows and rules can be specific to the needs of the particular request type.

The orange arrows in the diagram above show the predefined conversion paths a request may easily take, if a technician believes it is appropriate: a service request can, with a click of a button, generate an Incident or a Change Request, or can be linked to existing incidents or changes. Likewise, an Incident can generate or link to an existing Problem, while a Problem can generate or link to a Change Request.

A Problem may be related to multiple Incidents and a Change Request may be related to multiple Problems, Service Requests, and (through Problems) Incidents.

Naturally there are relationships within each of these tables to Users, Teams, Configuration Items, and other essential components of the system.

Service Requests vs. Change Requests

For some services, it may be unclear whether they should be offered through a service request or a change request. For instance, account provisioning, employee offboarding or onboarding, or simple application changes could theoretically go under either category.

The system is constructed on the principle that **service requests do not require approvals**. They include only “pre-approved” services.

Change requests may have approval workflows and complex approval processes.

So if you do not require approvals for employee offboarding or simple application changes, they may be best handled as service requests, while if you require approvals for employee onboarding, it may make sense to put that service in the change request category.

Of course, it is also possible to build out approvals for service requests if needed, but we like to use the approval question to help distinguish where a particular service should go.

SERVICE CATALOG – SERVICES TABLE

The Services table is the source for the ITIL Service Catalog. It holds a record for each service that may be offered to end users through the service catalog. All services offered by the organization to end users should be managed here and this table’s workflow is used to manage the approval process for a new service.

Each service is offered through a different request type, such as a Service Request, Purchase Request, or Change Request. Each service is therefore associated with one of these three tables, and this association is defined in the Services field “Show in Table”.

There are two ways to request a particular service: one is to create a new request of the correct type, such as a Service Request, and then to pick the service category and then the service from the drop-down list of services. The second method is to use an internal hotlink such as “Request a Password Reset” which will take the user to a new record in the appropriate table with the Service automatically set to password reset.

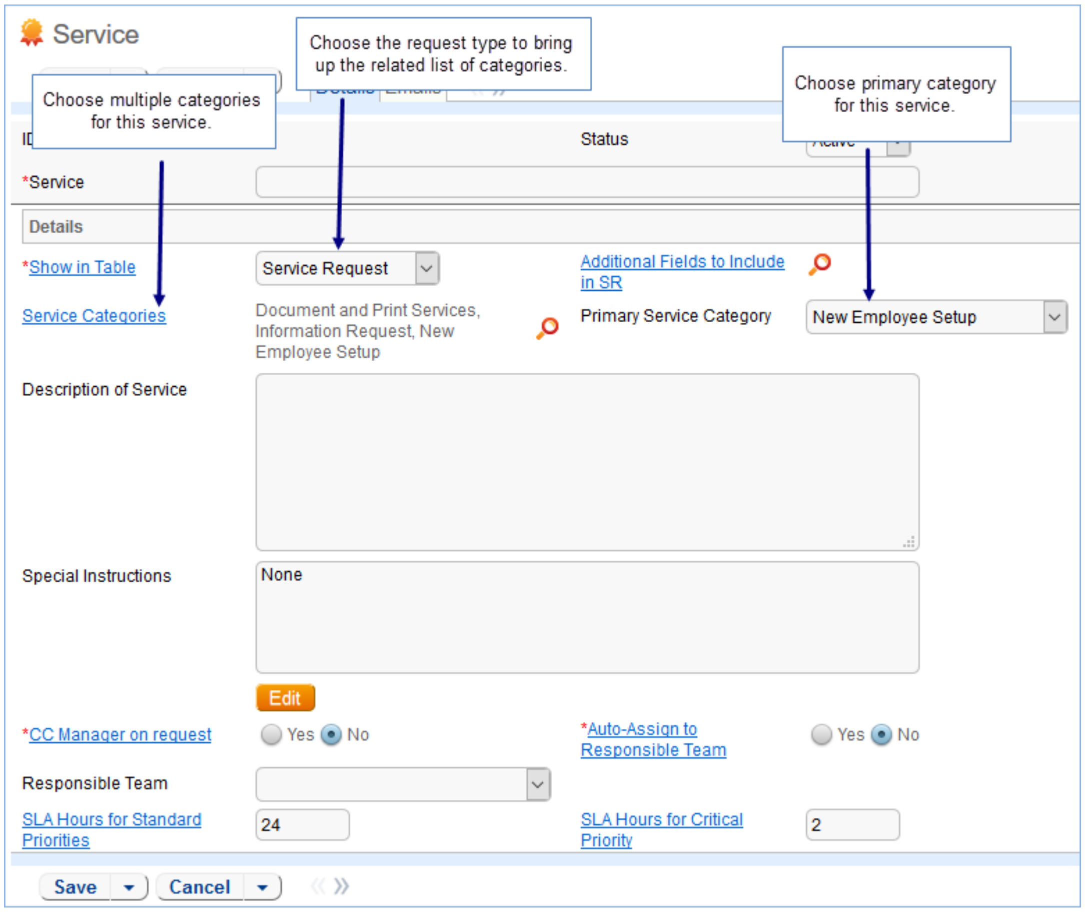
We have created some sample services. Naturally, your organization will want to create its own services, delete some of our sample services, or change the associations we have made for those services.

For instance, the onboarding of new employees is currently managed through the Service Request table under the Service Category New Employee Setup. Some organizations may prefer to manage new employee setup through the Change Request table instead. Making this kind of change is a matter of modifying the service record and changing the field called “Show in Table” to point to the desired table.

Since the Service Catalog is the backbone for many ITIL processes, the Services table has several special fields that are pulled into the corresponding request to enable automation, escalation, and special field visibilities and dependencies based on the type of service.

Note that in an effort to allow simplicity, we have not added a table to hold different SLAs, but have simply added some simple SLA fields to the services table. For organizations with a variety of complex SLAs based on multi-field criteria, it will make sense to create a separate table to manage all SLAs and then to pull in the appropriate SLA record for a particular service.

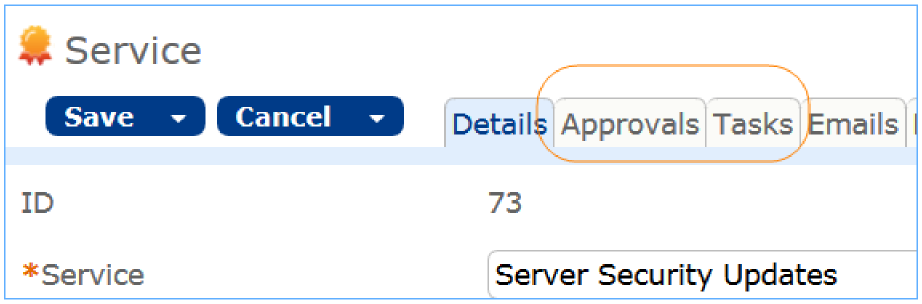
Below is a picture of the first tab of the Services form.


Service
Choose multiple categories for this service.
Choose the request type to bring up the related list of categories.
Choose primary category for this service.
Save
Cancel
Edit

ID		Status	Active
*Service			
Details			
*Show in Table	Service Request	Additional Fields to Include in SR	
Service Categories	Document and Print Services, Information Request, New Employee Setup	Primary Service Category	New Employee Setup
Description of Service			
Special Instructions	None		
*CC Manager on request	Yes [✓ No]	*Auto-Assign to Responsible Team	Yes [✓ No]
Responsible Team			
SLA Hours for Standard Priorities	24	SLA Hours for Critical Priority	2

Approvals and Tasks Related to Services

Some services may have associated approval workflows or standard tasks or task workflows. These options are available from the additional tabs in the Service record:



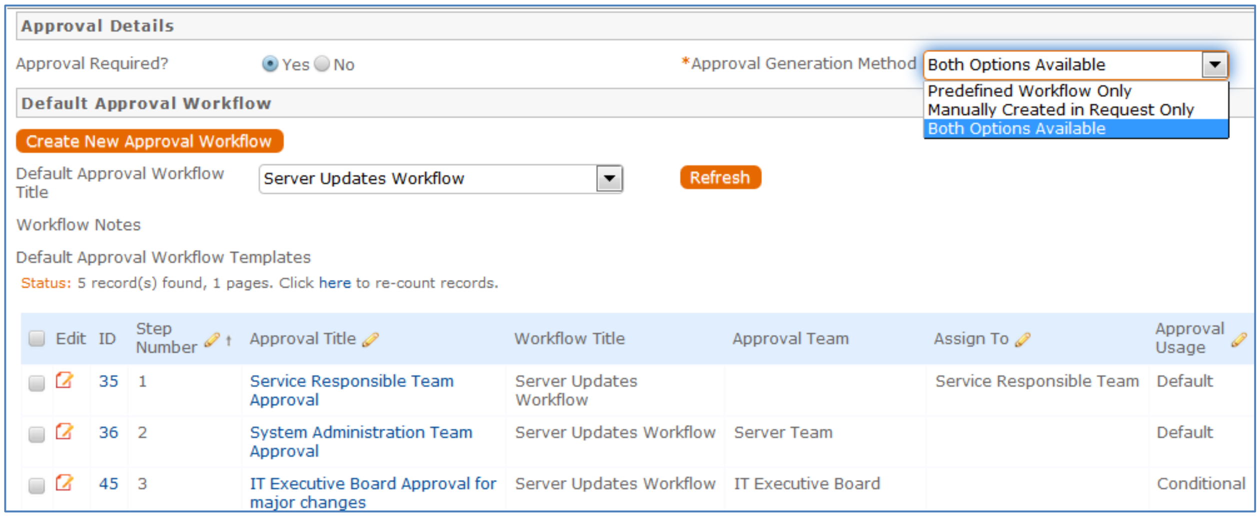
ID	73
*Service	Server Security Updates

In the current version of the ITIL template, approvals are available out of the box only for Change Requests.

Tasks are available for both Service Requests and Change Requests.

Approvals for Change Requests

There are three options for how to handle approvals for a change request, shown in the Approval Generation Method drop-down below: using a predefined approval workflow, manually created ad hoc approvals, or both:



Approval Details

Approval Required? [✓ Yes] No

*Approval Generation Method: Both Options Available

Predefined Workflow Only
Manually Created in Request Only
Both Options Available

Default Approval Workflow

Create New Approval Workflow

Default Approval Workflow Title: Server Updates Workflow

Workflow Notes

Default Approval Workflow Templates

Status: 5 record(s) found, 1 pages. Click here to re-count records.

Edit	ID	Step Number	Approval Title	Workflow Title	Approval Team	Assign To	Approval Usage
	35	1	Service Responsible Team Approval	Server Updates Workflow		Service Responsible Team	Default
	36	2	System Administration Team Approval	Server Updates Workflow	Server Team		Default
	45	3	IT Executive Board Approval for major changes	Server Updates Workflow	IT Executive Board		Conditional

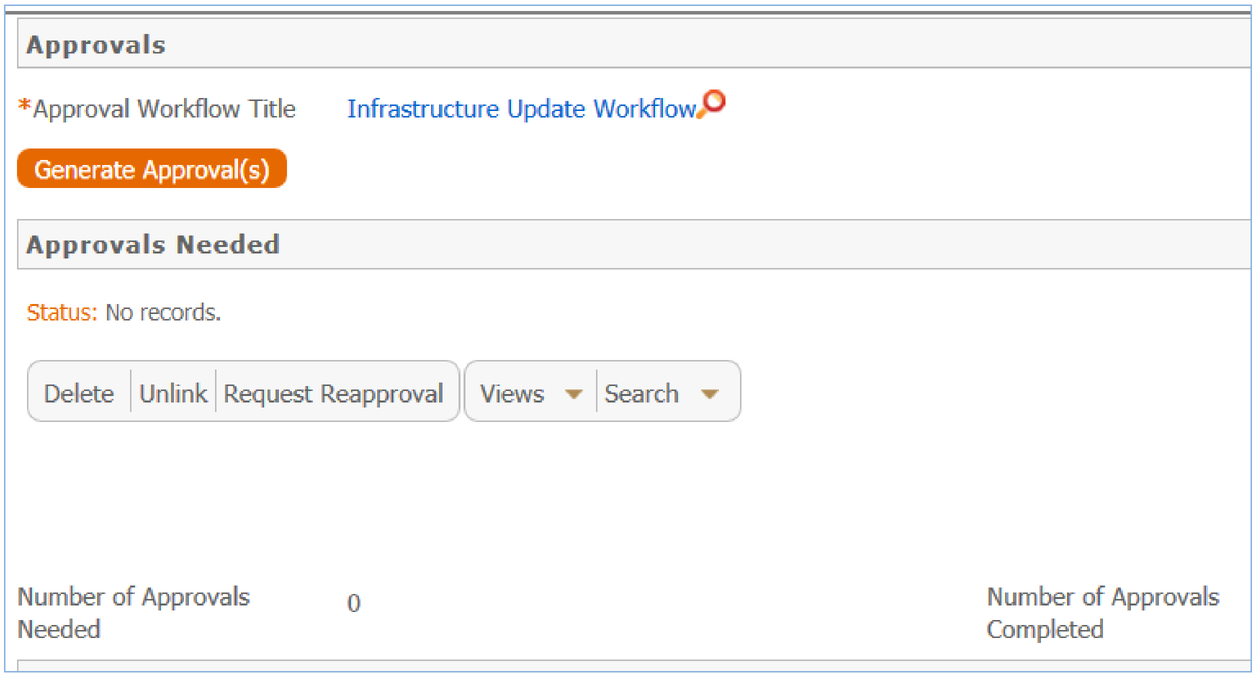
For a **Predefined Workflow Only**, the user will select from existing approval workflows related to change requests. If none yet exist, he can use the button to create a new approval workflow on the fly.

If he selects **Manually Created in Request Only** then no further information is needed in the service record, and the user will be able to add the approvals manually within the individual change request.

Approval View in the Change Request

The selections made in the Service will result in different options appearing within the Change Request.

In a change request based on a predefined workflow only, the user will see the workflow listed and a button to generate the approval records. If the user has permission to edit the Approval Workflow Title field, he would also be able to change to a different workflow as needed:


Approvals

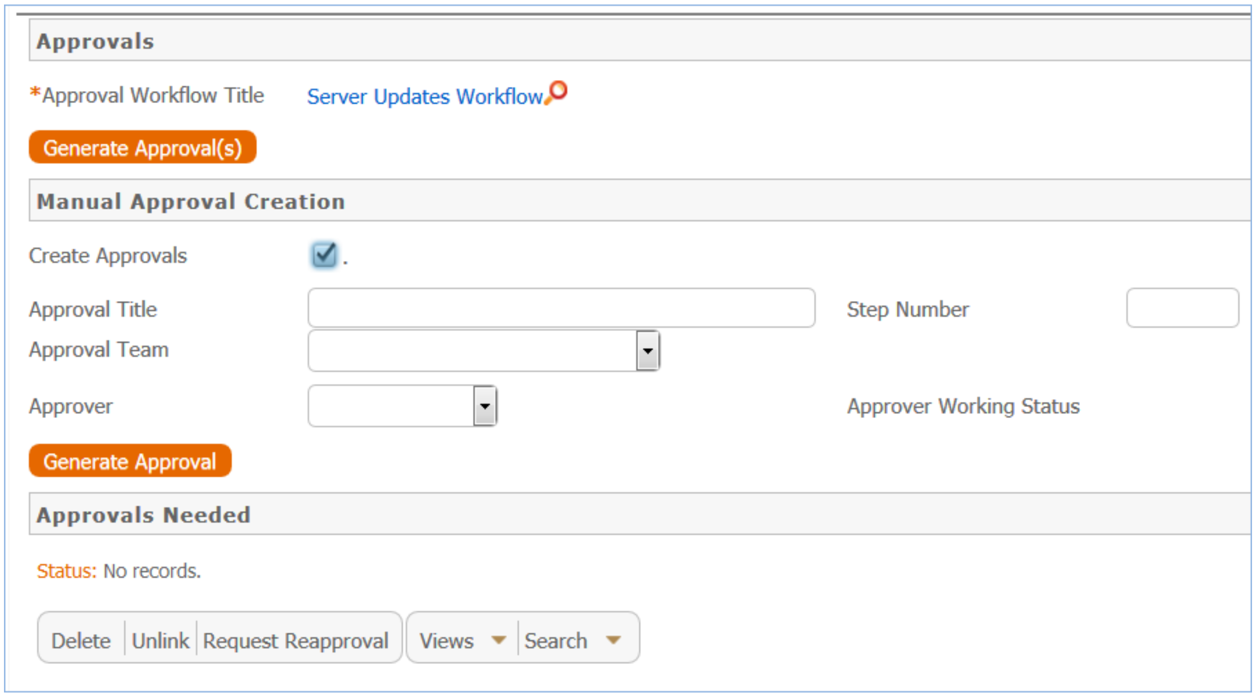
*Approval Workflow Title Infrastructure Update Workflow

Approvals Needed

Status: No records.

Number of Approvals Needed	0	Number of Approvals Completed	

If both predefined and manual approval creation are permitted, the user will see an additional checkbox (Create Approvals) that can be expanded to add one or more ad hoc approvals:


Approvals

*Approval Workflow Title Server Updates Workflow

Manual Approval Creation

Create Approvals ✓

Approval Title		Step Number	
Approval Team			
Approver		Approver Working Status	

Approvals Needed

Status: No records.

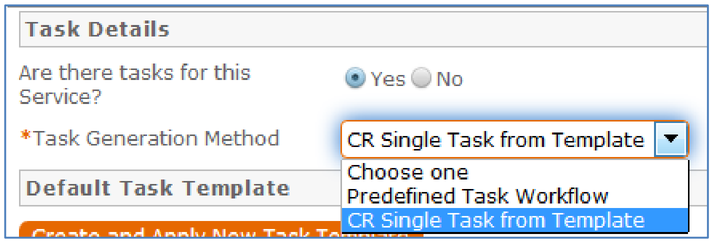
If only the manual approval creation is defined, then the user will just see the option to Create Approvals and will add any approvals needed.

Note that if the Approval Required field in the service has a Yes value, at least one approval will need to be created and approved to move the change request forward to completion.

Tasks for Change Requests

The Tasks tab of the service allows the selection of a method for task generation. The available methods depend on the request type and they are different for service requests and change requests.

Change Requests have two options for the task generation method: Predefined Task Workflow and CR Single Task from Template:



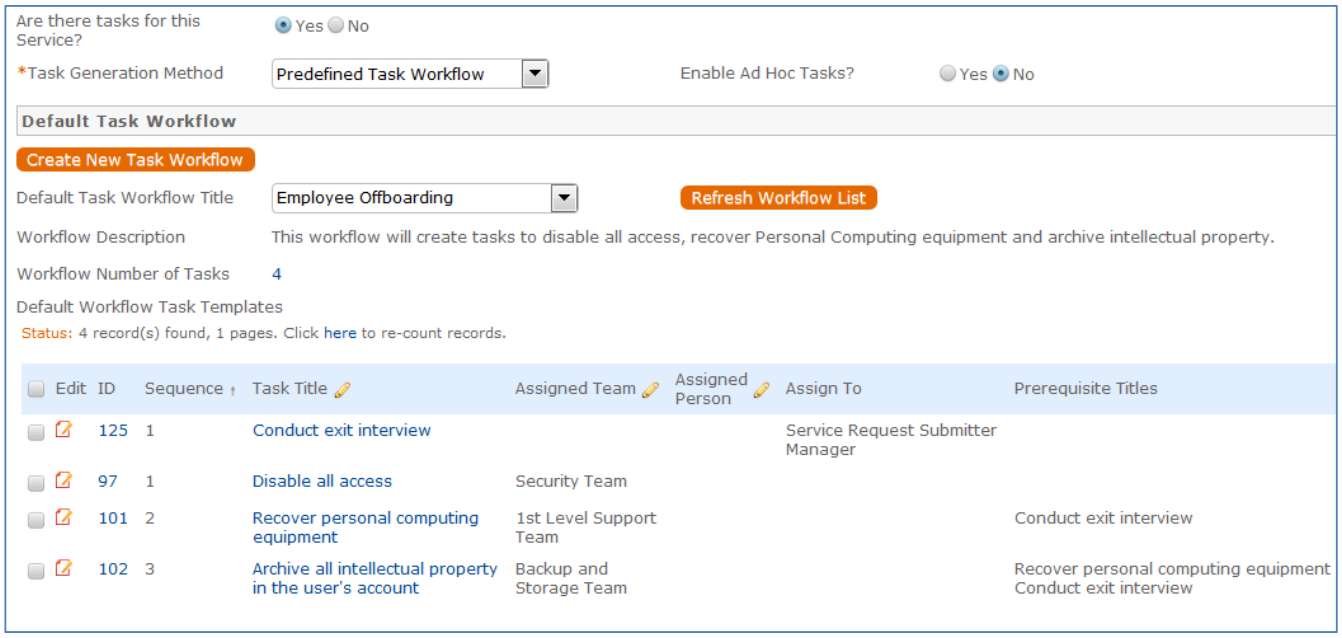
Task Details

Are there tasks for this Service?	[✓ Yes] No
*Task Generation Method	CR Single Task from Template

Change requests are typically associated with one or more configuration items. Task handling for change requests is therefore designed to create a task for each configuration item that has been associated with the request. If the user chooses 100 configuration items for a change request and generates tasks, whichever tasks are defined for the change request are created 100 times, one for each configuration item.

The tasks are generated in this way by using a task template and pushing its details down through the configuration item records, so this necessitates restricting some of the other task generation options.

Predefined Task Workflow - allows the selection and/or creation of a Task Workflow with predefined task templates arranged in a sequential or parallel order (see more details about configuring task workflows in the Task Workflows section):



Are there tasks for this Service?	[✓ Yes] No
*Task Generation Method	Predefined Task Workflow
Enable Ad Hoc Tasks?	Yes [✓ No]

Default Task Workflow

Default Task Workflow Title	Employee Offboarding
Workflow Description	This workflow will create tasks to disable all access, recover Personal Computing equipment and archive intellectual property.
Workflow Number of Tasks	4

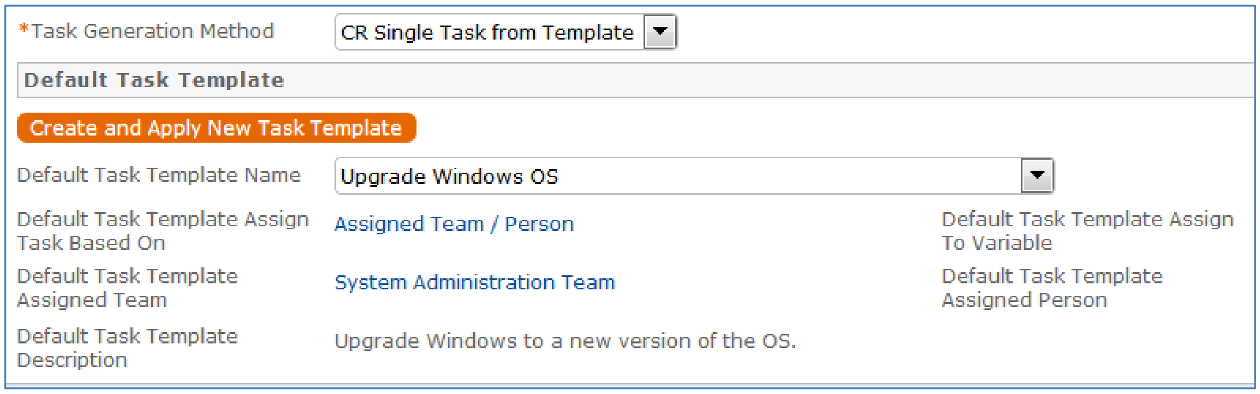
Default Workflow Task Templates

Status: 4 record(s) found, 1 pages. Click here to re-count records.

Edit	ID	Sequence	Task Title	Assigned Team	Assigned Person	Assign To	Prerequisite Titles
	125	1	Conduct exit interview			Service Request Submitter Manager	
	97	1	Disable all access	Security Team			
	101	2	Recover personal computing equipment	1st Level Support Team			Conduct exit interview
	102	3	Archive all intellectual property in the user's account	Backup and Storage Team			Recover personal computing equipment Conduct exit interview

Note that if your workflow has 3 tasks, they will each be generated for each configuration item.

The **CR Single Task from Template** option is more common for change requests involving numerous configuration items. Here you define a single task template to be used with the change request:



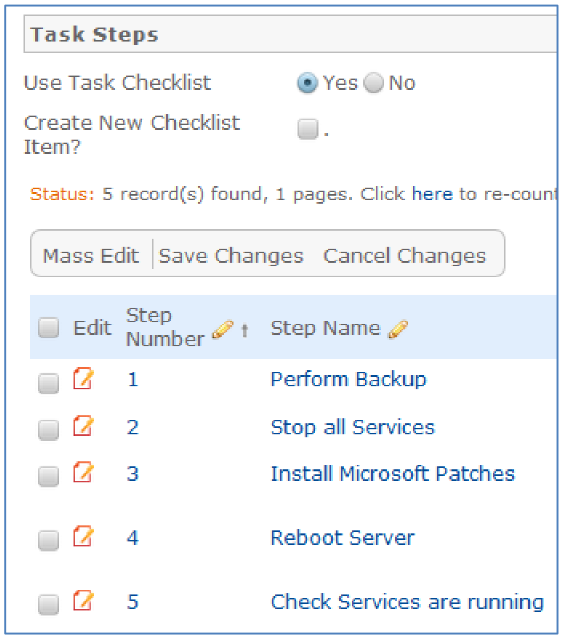
*Task Generation Method	CR Single Task from Template		

Default Task Template

Create and Apply New Task Template

Default Task Template Name	Upgrade Windows OS		
Default Task Template Assign Task Based On	Assigned Team / Person	Default Task Template Assign To Variable	
Default Task Template Assigned Team	System Administration Team	Default Task Template Assigned Person	
Default Task Template Description	Upgrade Windows to a new version of the OS.		

Note that a single task may have several predefined steps, presented as a checklist, to assist technicians in performing the task. See the sections on Task Templates and Task Steps below for more details on how to set this up. Below is an example of a task template that has been defined with specific steps:



Task Steps

Use Task Checklist [✓ Yes] No

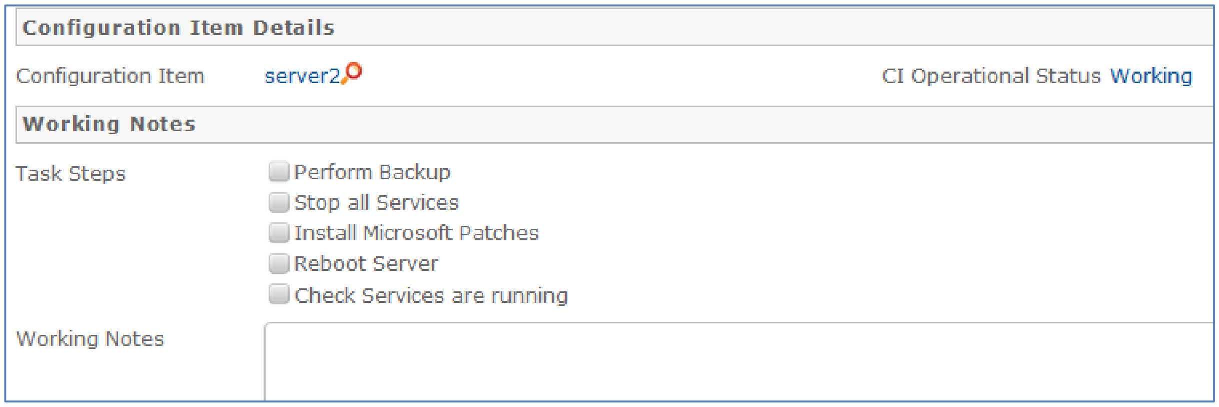
Create New Checklist Item? .

Status: 5 record(s) found, 1 pages. Click here to re-count

Mass Edit | Save Changes | Cancel Changes

	Edit	Step Number	Step Name
		1	Perform Backup
		2	Stop all Services
		3	Install Microsoft Patches
		4	Reboot Server
		5	Check Services are running

Here is how the setup above will appear in the task that is assigned and updated by the user doing the work:



Configuration Item Details

Configuration Item	server2	CI Operational Status	Working

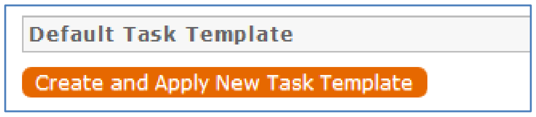
Working Notes

Task Steps
- ☐ Perform Backup
- ☐ Stop all Services
- ☐ Install Microsoft Patches
- ☐ Reboot Server
- ☐ Check Services are running

Working Notes

Using task steps within a single task makes good sense for a change request.

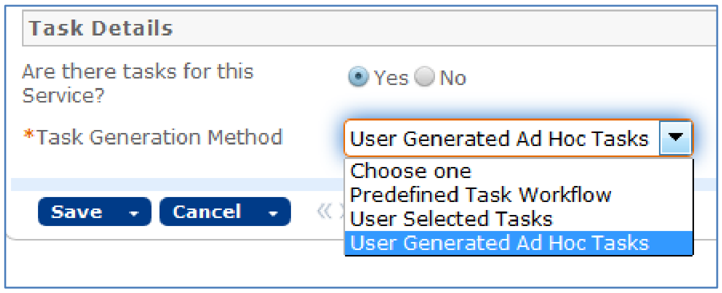
While setting up the service, if the task template you want to use does not yet exist, the Create and Apply New Task Template button can be used to bring up the task template screen to define a new template, and once saved, that template will be selected for the service you launched it from:



In the actual change request, it is possible when using this task method to select more than this one task template to generate. Once having generated the default task, the user may select from other pre-defined tasks and generate them as well.

Tasks for Service Requests

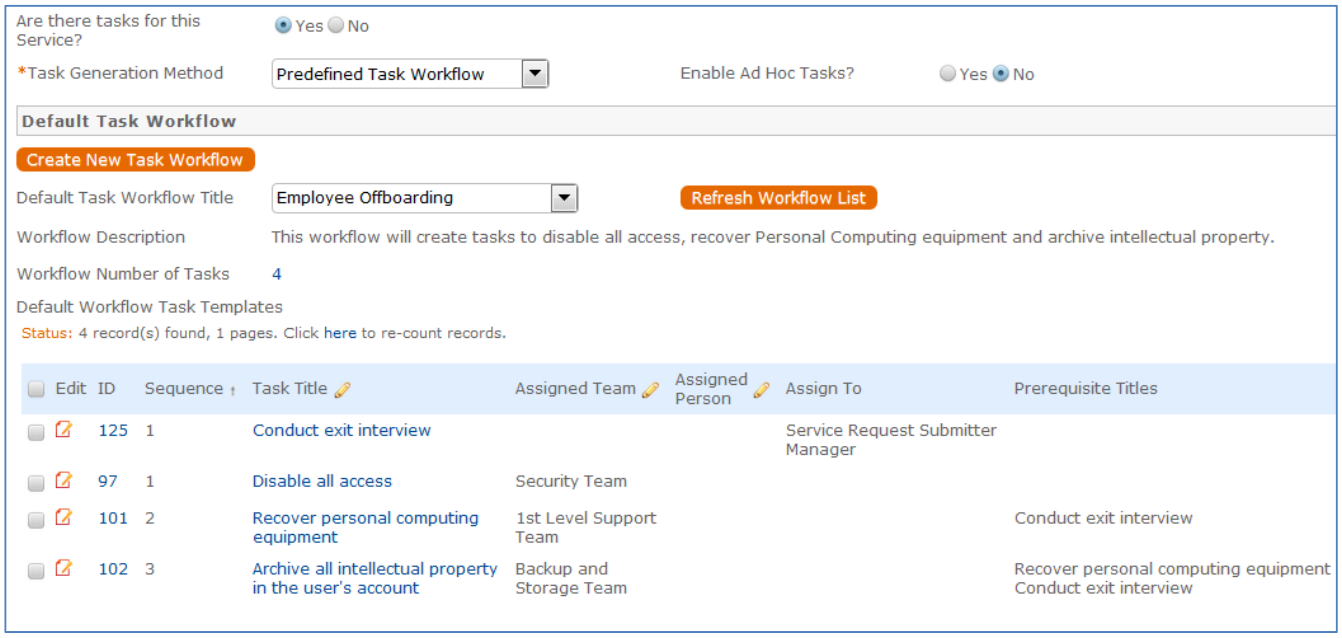
For service requests, there are some different task generation options. There are three possible methods:



Task Details

Are there tasks for this Service?	[✓ Yes] No
*Task Generation Method	User Generated Ad Hoc Tasks

Predefined Task Workflow – this is the same as the usage in change requests, except that the tasks are not tied to CI's and are just created once per request.



Are there tasks for this Service?	[✓ Yes] No		
*Task Generation Method	Predefined Task Workflow	Enable Ad Hoc Tasks?	Yes [✓ No]

Default Task Workflow

Default Task Workflow Title	Employee Offboarding
Workflow Description	This workflow will create tasks to disable all access, recover Personal Computing equipment and archive intellectual property.
Workflow Number of Tasks	4

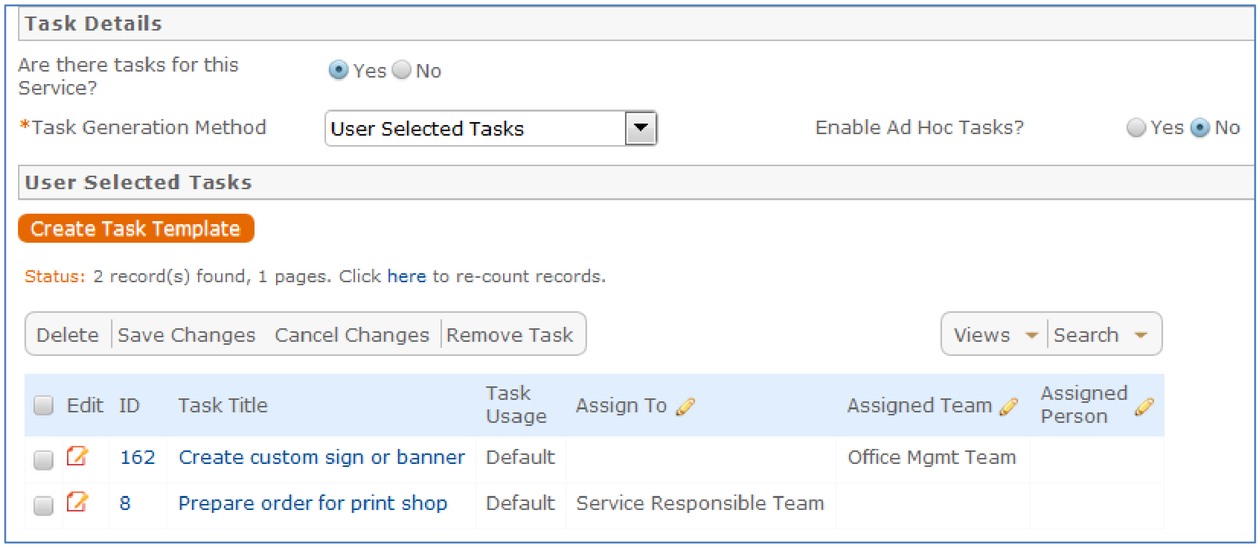
Default Workflow Task Templates

Status: 4 record(s) found, 1 pages. Click here to re-count records.

Edit	ID	Sequence	Task Title	Assigned Team	Assigned Person	Assign To	Prerequisite Titles
	125	1	Conduct exit interview			Service Request Submitter Manager	
	97	1	Disable all access	Security Team			
	101	2	Recover personal computing equipment	1st Level Support Team			Conduct exit interview
	102	3	Archive all intellectual property in the user's account	Backup and Storage Team			Recover personal computing equipment Conduct exit interview

With this choice in a service request's service, another field - Enable Ad Hoc Tasks – appears, and if it has a Yes value, then in addition to the predefined tasks, the user may create additional tasks for a particular request.

User Selected Tasks –allows the service manager to create a set of possible tasks that will be displayed in a service request as checkbox items, so the user may select which tasks are relevant for this particular service request and generate just those tasks:


Task Details

Are there tasks for this Service? [✓ Yes] No

*Task Generation Method: User Selected Tasks

Enable Ad Hoc Tasks? Yes [✓ No]

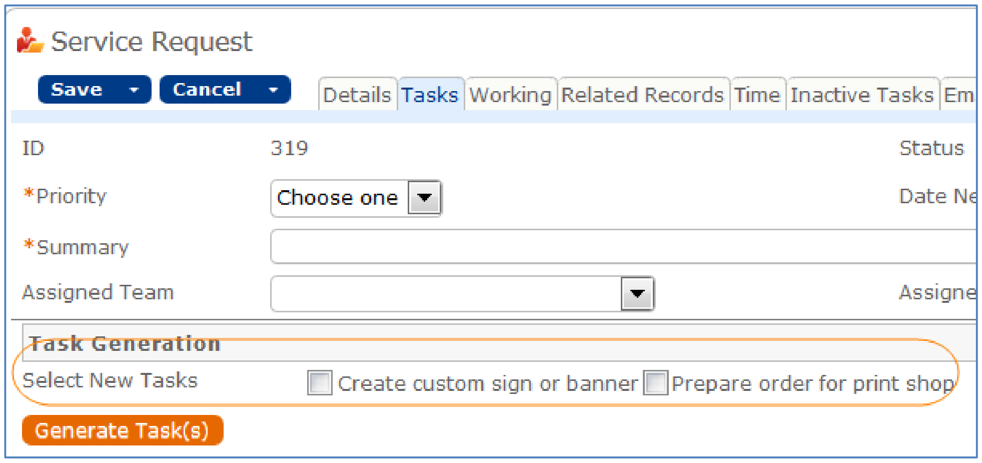
User Selected Tasks

Status: 2 record(s) found, 1 pages. Click here to re-count records.

Edit	ID	Task Title	Task Usage	Assign To	Assigned Team	Assigned Person
	162	Create custom sign or banner	Default		Office Mgmt Team	
	8	Prepare order for print shop	Default	Service Responsible Team		

The Create Task Template button is used to create the potential tasks as templates. Ad hoc tasks may also be enabled for this type by selecting Yes for the Enable Ad Hoc Tasks? field shown above.

The result of the service shown above in a given service request:


Service Request

ID	319
*Priority	Choose one
*Summary	
Assigned Team	

Status

Date Ne

Assigne

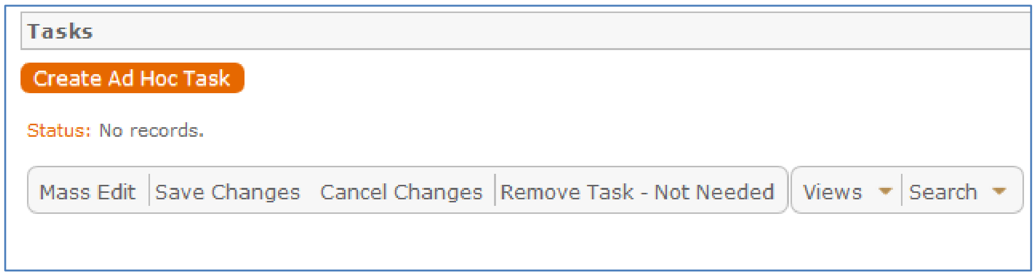
Task Generation

Select New Tasks ☐ Create custom sign or banner ☐ Prepare order for print shop

The user checks the boxes for the tasks he wants, and clicks the Generate Task(s) button to generate the individual tasks.

These tasks, since they are all optional, are defined as parallel tasks without specific prerequisites.

User Generated Ad Hoc Tasks – this selection allows the user to create manual tasks in the service request without reference to any preexisting templates. In this case, the user will see a button in the service request to create a new task:



Ad hoc tasks can be made sequential during creation by selecting one or more prerequisite tasks as the individual tasks are created.

Service Catalog Management

Because the Services table has some special fields, several of them have onscreen explanations. These explanations are accessed in the input form by clicking on the underlined field labels.

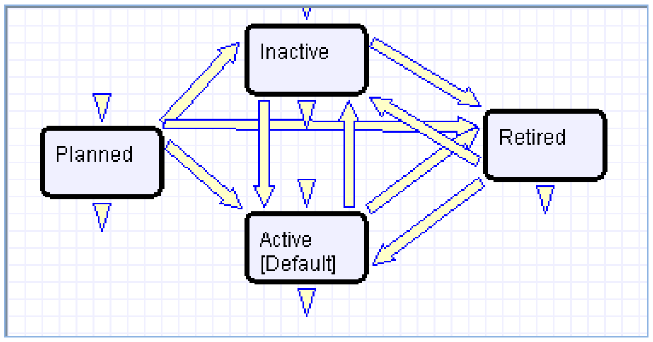
Once the Service Catalog is initially set up by creating the appropriate services in the Service table, new service items will be created by staff members only when a new service is planned or initiated. Existing services may naturally be modified as needed. Initial permissions are set so only users in the admin, Service Manager, and Change Manager groups can create/modify/delete services.

Service records may be created in a state of Planned or Active. Once all active services have been set up during the initial system configuration, it may be desirable to edit the workflow to deselect the Creatable box for the Active State, so that future services must start in a status of Planned and go through an approval process. Temporarily inactive services should be placed in the Inactive state. Services no longer in use can be placed in the Retired state for recordkeeping.

Ownership

Services are owned by the person who creates the service.

Workflow



Saved Searches

Some of the default saved searches provided are detailed below.

Saved Search Name	Search Description

Service Pipeline	Status is Planned
Service Catalog	Status is Active
Inactive Services	Status is Inactive
Retired Services	Status is Retired
Planned Services	Status is Planned

APPROVAL MANAGEMENT

There are three tables that control the automated and ad hoc creation of approvals related to various processes within the system. Currently approvals are used within Change Requests, Contracts, and Documents. They can be added to other processes as needed.

The approval processes are managed by the Approval Workflows, Approval Templates, and Approvals tables.

Approval Workflows Table

The Approval Workflows table predefine individual workflows for approval processes. A workflow consists of one or more Approval Templates, each of which defines a team or person to be the approver, the order in which the approval will be needed, and some other characteristics of the approval.

By default, approval workflows can be used for contracts and change requests.

The approval process can include a combination of parallel and sequential approval steps.

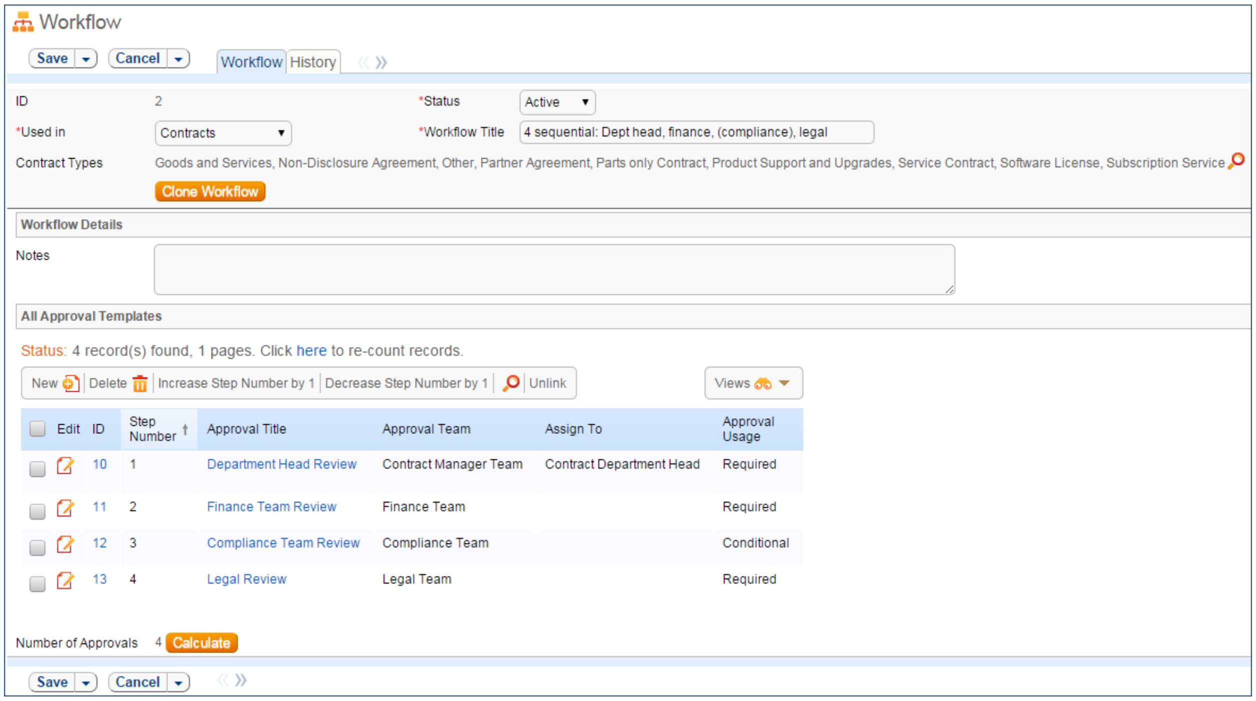
Use Case

Approval workflow records may be created by members of the Admin, Contract Manager, Change Manager, and Service Manager groups.

To create a new approval workflow, click New from the Approval Workflow table action bar or from within the Contract Type or Service record for a change request.

The Related To field must be set to the correct value for the correct linked fields to appear, based on the process for which the approval workflow will be used.

The next step is to add approval templates to the workflow by clicking New in the action bar of the Approval Templates related table. For more information on how to design the templates to get a mixture of sequential and parallel approvals, see the SEQUENCE OF APPROVALS section.


Workflow

Save | Cancel | Workflow | History

ID	2	*Status	Active
*Used in	Contracts	*Workflow Title	4 sequential: Dept head, finance, (compliance), legal
Contract Types	Goods and Services, Non-Disclosure Agreement, Other, Partner Agreement, Parts only Contract, Product Support and Upgrades, Service Contract, Software License, Subscription Service		

Clone Workflow

Workflow Details

Notes

All Approval Templates

Status: 4 record(s) found, 1 pages. Click here to re-count records.

New | Delete | Increase Step Number by 1 | Decrease Step Number by 1 | Unlink | Views

Edit	ID	Step Number	Approval Title	Approval Team	Assign To	Approval Usage
	10	1	Department Head Review	Contract Manager Team	Contract Department Head	Required
	11	2	Finance Team Review	Finance Team		Required
	12	3	Compliance Team Review	Compliance Team		Conditional
	13	4	Legal Review	Legal Team		Required

Number of Approvals 4 Calculate

Save | Cancel

Required fields are marked by a red asterisk. These include the Status, Workflow Title, and which table the workflow is Used In. If the choice of table in the Used In field is Contracts, additional options allow the user to specify which Contract Types can use this workflow.

When the workflow is selected for use in an approval process, the system will generate required Approval records based on the specified approval templates. Additional ad hoc approvals may be created directly from within the contract or change request if that method is supported, based on the service or contract type setup.

Managing and Reusing Approval Workflows

Approval workflows can be modified even if there are outstanding contracts using them. Since the approvals are generated up front as soon as the contract moves to Pending Approval, later modifications to the workflow and its templates will only have an impact if there are conditional approvals which are rechecked later in the process.

A workflow can be cloned by using an action button. Clicking Clone Workflow will copy the workflow, and make a copy of each of its approval templates. The cloned Approval Templates will be automatically linked to the newly cloned Workflow record.

Approval Workflow Statuses

The approval workflow Status is either Active or Inactive.

An Active workflow appears in the Workflow Title drop-down menu from a Contract or Service record. Workflows with a Status of Inactive are no longer available for use in the approval process; inactive workflows do not appear in the drop-down menu as a possible approval process selection. Admin group members can change a workflow from Active to Inactive if needed.

Ownership

Workflow records are owned by the user who creates them. Specifically, a record is owned by the user whose Login matches the Creator Login field.

Approval Templates Table

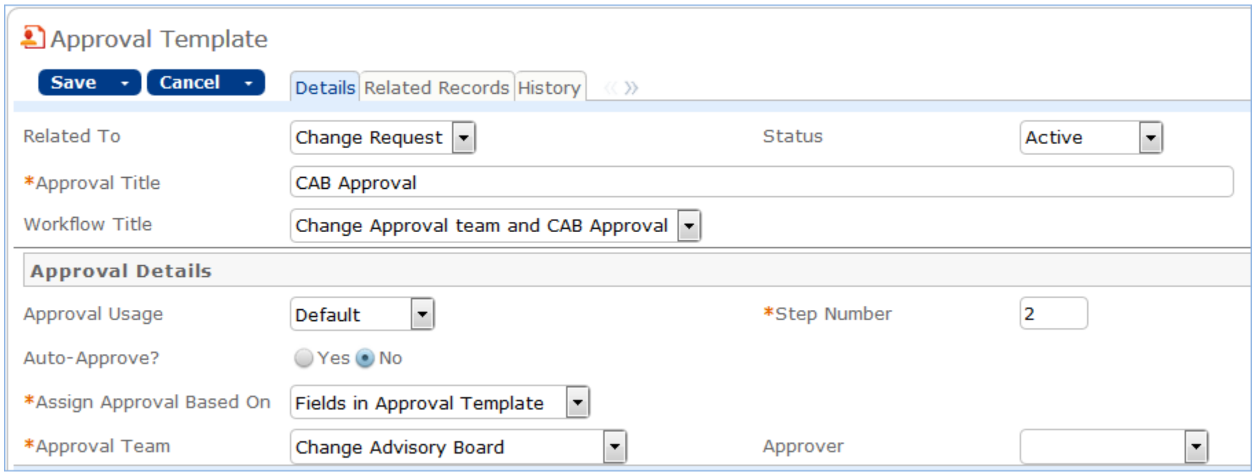
The Approval Templates table holds a record for each approval to be created as part of a standard Approval Workflow. It predefines the approval team or person and also controls whether the approval is conditional on some meta data criteria being met or is always generated.

Each approval template can only be used by a single approval workflow, i.e., there is a one-to-many relationship between approval workflows and approval templates.

Use Case

Approval Templates can be created by users in the Admin, Contract Manager, Change Manager, and Service Manager groups. New approval template records are normally added from within an approval workflow record. Creating an approval template from the Approval Workflow table populates the link to the workflow. If an approval template is created directly from the Approval Template table, a linked Workflow Title should be manually selected.

Each template contains information about which Workflow uses the approval template, whether the approval is Required or Conditional, and how the approval is assigned. The required fields are Approval Title, Assign Approval Based On, and Step Number.



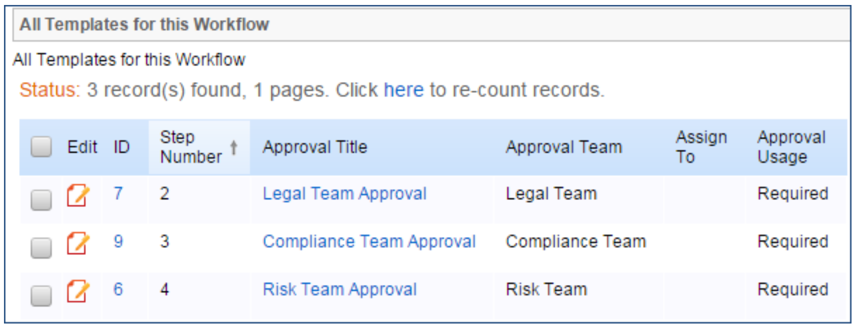
Related To	Change Request	Status	Active
*Approval Title	CAB Approval		
Workflow Title	Change Approval team and CAB Approval		

Approval Details

Approval Usage	Default	*Step Number	2
Auto-Approve?	Yes [✓ No]		
*Assign Approval Based On	Fields in Approval Template		
*Approval Team	Change Advisory Board	Approver	

Figure 1. Sample approval template record form.

The Related Records tab displays all other approval templates used in the same workflow.


All Templates for this Workflow

All Templates for this Workflow

Status: 3 record(s) found, 1 pages. Click here to re-count records.

	Edit	ID	Step Number	Approval Title	Approval Team	Assign To	Approval Usage
		7	2	Legal Team Approval	Legal Team		Required
		9	3	Compliance Team Approval	Compliance Team		Required
		6	4	Risk Team Approval	Risk Team		Required

Figure 2. The Related Records tab shows all other templates in the same workflow.

The Step Number and title for the approval are used when converting the template to an Approval record.

Required vs. Conditional Approvals

In the Approval Usage field, the user sets whether the approval is Required or Conditional. Both required and conditional approvals are automatically generated when the Create Approvals button is clicked in the contract record or the Generate Approvals from Workflow button is clicked in the change request record.

When Conditional is selected, the Condition field will appear. The user can then input a formula that is evaluated when approvals are generated. This formula can contain any number of conditions that can be linked together by operators such as "or", "and", "contains", etc. Conditional approvals are only generated if the condition is met (typically the Condition is a search criterion based on some field value(s) in the Contract).

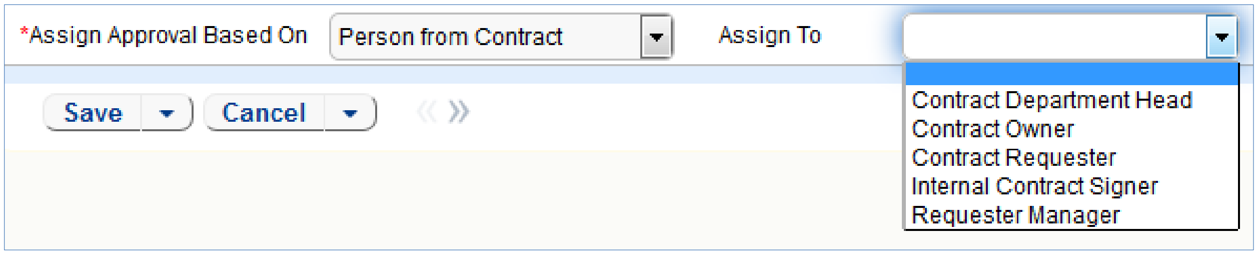
Assigning Approvals

Approvals can be assigned to teams or users based on fields in the Approval Template or other variable fields from the parent record such as Contract Owner, Requester Manager, Contract Department Head, or Change Manager, Change Requester, etc.

If "Fields in Approval Template" is selected in the Assign Approval Based On field, the Approval Team and Approver fields are visible. The Approver field is filtered to members of the selected Approval Team.

If, for an approval template for contracts, "Person from Contract" is selected, the Assign To field appears. The drop-down selection in this field shows a list of user fields from the Contract record, such as Contract Owner, Contract Requester, and Requester Manager.

***Admin Note**: The selections in the Assign To field are linked from the Replacement Variables table. Refer to the* REPLACEMENT VARIABLES TABLE *section for more information.*


*Assign Approval Based On: Person from Contract

Assign To:

Figure 3. Approvals can be automatically assigned based on fields in the contract or change request record.

Similarly, when "Team from Contract" is selected, a team can be chosen from the Assign To dropdown. For approval templates related to change requests, the options for Assign Approval Based On are "Person from Change Request" and "Team from Change Request."

Sequence of Approvals

The sequence of approvals depends on the Step Numbers. Approvals are generated and ordered based on the Step Number order, and are triggered to Pending Approval from the lowest to the highest Step Number. To set up parallel Approvals, give the same Step Number to each Approval Template in a parallel step. All concurrent Approvals (that is, Approvals with the same step number) must be approved in order to trigger the next step.

The automation that controls the triggering and ordering of approvals is managed from the Approval record, not the Approval Template. Fields in the approval record are used to determine if there are concurrent approvals and to define the Lowest Step Number (which may not be 1, if the first step is a conditional approval marked Not Needed).

Automatic Approvals

You can use an approval template simply as a notification rather than an approval. This is done by setting the Auto Approve? field to Yes. When the template is converted into an Approval record, a rule running on the Approval table automatically sets the approval's Status to Approved. If Notify for Auto-Approval is set to Yes, this rule also sends a custom notification message in place of the assignment notification.

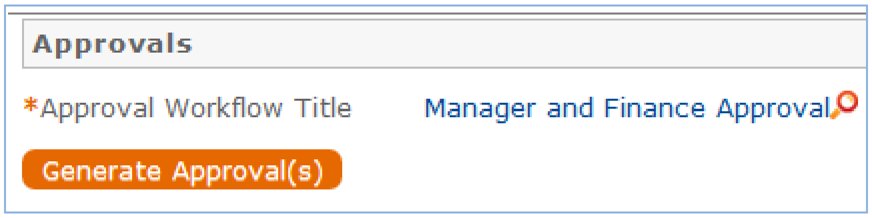
Approvals Table

The Approvals table holds all of the approvals that are sent to users and approved or rejected by them. Each record in the table is an individual approval or rejection linked to a parent Change Request, Contract, or Document record.

In addition to being generated automatically based on a workflow and its approval templates, ad hoc approvals may be created by users with the appropriate privileges, either in addition to the predefined approvals, or instead of them.

Use Case for Change Requests

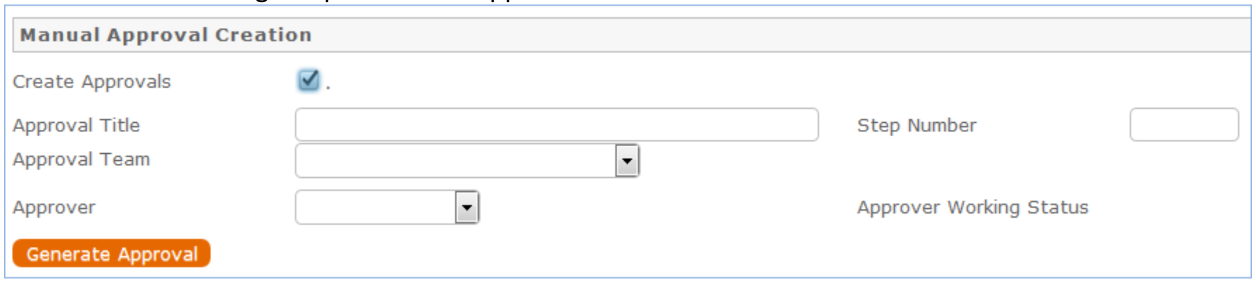
The default status for a new approval record is Queued. There are two main ways in which an approval is created for a change request. The first is by the change manager clicking a button on the Approvals tab:



This will generate all of the approvals for that workflow.

If an approval template's Approval Usage field has a value of Conditional, then it is not created unless the condition is met.

The other method is for a change manager to create an ad hoc approval using the Create Approvals checkbox in the change request on the Approvals tab:


Manual Approval Creation

Create Approvals	✓ .		
Approval Title		Step Number	
Approval Team			
Approver		Approver Working Status	

Note that this checkbox is only visible if the service was defined to permit ad hoc approvals and only until the approvals are launched.

The approvers are not notified until the Change Manager launches the approval process, by clicking the Launch Approvals button.

At that point, the lowest numbered approval record will be updated to Pending Approval and the Approver or the Approval Team (if the Approver field is blank) will be notified that the approval is due. If there are any concurrent approvals for a particular step, the approval record will be updated with those concurrent approvals.

If an approval step is marked as Auto-Approved, the Approval will be updated to Approved and the Approver or the Approval Team will be notified of the auto-approval, unless the Notify for Auto-Approval field has a No value. This is a method of notifying someone about a change without requiring them to respond.

The approver can approve the change by email through a hotlink or click a Require Change or Reject hotlink to edit the Approval record directly and enter some comments and click the appropriate button. If editing the approval record directly, comments will be required if the approval record is rejected or marked as requiring changes.

Any individual attempting to click one of the three buttons to approve, reject or require changes will be subject to a validation rule requiring that they be a member of the Approval Team.

Whenever the approval notes field is updated, a rule will copy the update into the linked Change Request's All Approval Notes append only field and the field in the approval will be blanked out.

If the approver marks the approval as Requires Change, the linked change request will not be changed, but the Change Management team and Change Manager (if any) will be notified and they will decide whether any changes made require restarting the whole approval process or just carrying on from the rejected approval.   If they want to start over, they will click a "Relaunch Approvals" button and that will first set all linked approvals to Queued and then launch the first step again. Otherwise they can set the "requires change" approval back to Pending approval for the process to carry on from there again.

If the approval is marked Approved and there are no concurrent approvals, the next approval record in the sequence will be updated to Pending Approval and the Approver or Approval Team will be notified. If there are concurrent approvals, all concurrent approvals must be completed prior to the next approval step being updated to Pending Approval. The approver and the date approved will be updated to reflect whoever clicked the Approve button.

If the approver marks the approval as Permanently Rejected, the linked change request will be updated to a status of Rejected and the Requester and Change Management Team notified. The user will be required to resubmit the change request if still required, with appropriate changes. All other approvals in the Change Request in a status of Queued or Pending Approval will be updated to Not Needed. If an approval is updated from a status of Pending Approval to Not Needed as a result of the above action, the Approver or Approval Team will be notified that their approval is no longer necessary.

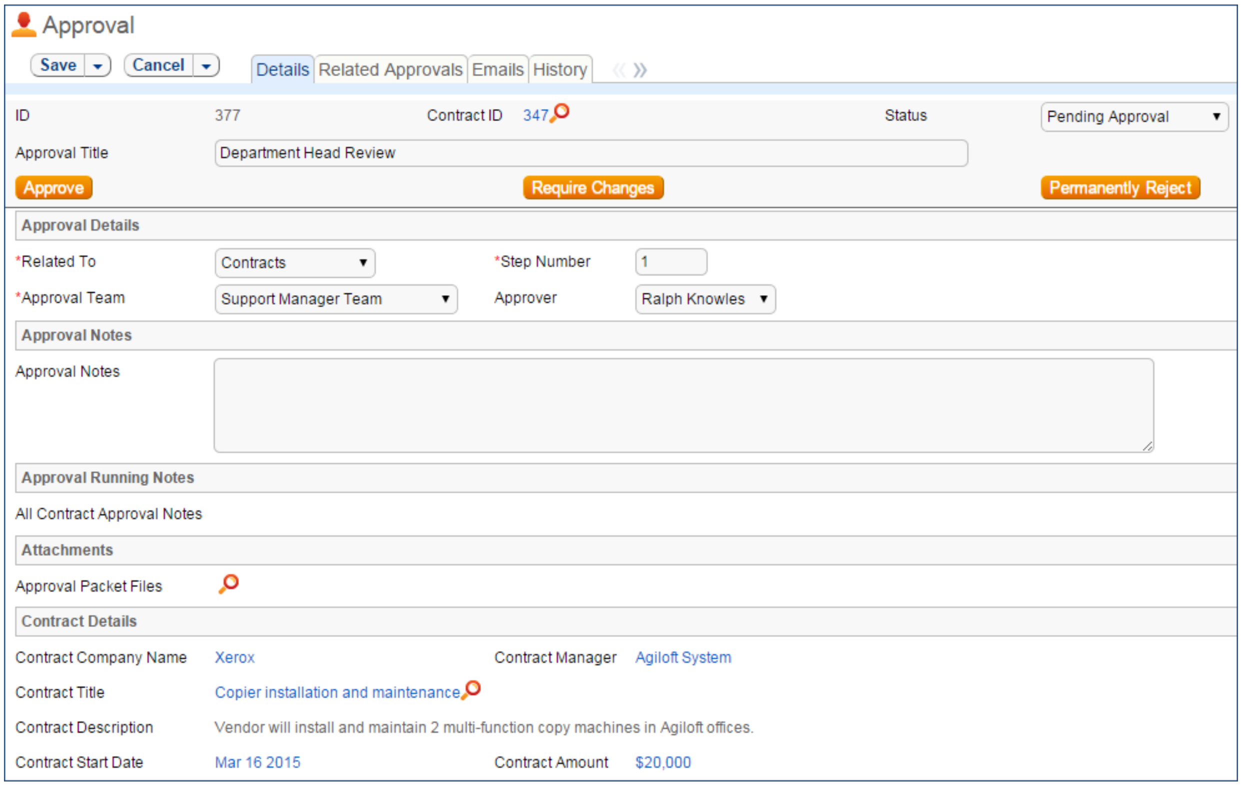
Use Case for Contracts

Creation of Approvals

The process begins when the Create Approvals button is clicked in the contract record. Contract approval records are created from Approval Templates using a conversion action or manually on an ad hoc basis. For information on starting the approval process for a contract, refer to the HANDLING APPROVALS section.

Approval Records

Each approval record stores the parent Contract ID, Approval Team, and Approver, the user who submitted the approval. The Date Approved/Rejected field captures and displays the timestamp of the approval, shown on the History tab.



ID	377	Contract ID	347	Status	Pending Approval
Approval Title	Department Head Review				

Approval Details

*Related To	Contracts	*Step Number	1
*Approval Team	Support Manager Team	Approver	Ralph Knowles

Approval Notes

Approval Notes

Approval Running Notes

All Contract Approval Notes

Attachments

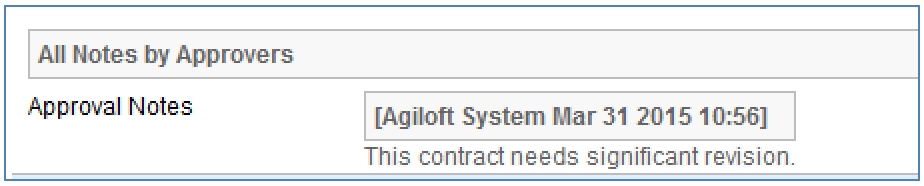
Approval Packet Files

Contract Details

Contract Company Name	Xerox	Contract Manager	Agiloft System
Contract Title	Copier installation and maintenance		
Contract Description	Vendor will install and maintain 2 multi-function copy machines in Agiloft offices.		
Contract Start Date	Mar 16 2015	Contract Amount	$20,000

Figure 4. Sample Approval record.

Notes about the approval or rejection are entered into the Approval Notes field.
When the record is saved, the notes are appended to the All Contract Approval Notes field and are visible from *any* approval linked to that Contract. Additionally, the notes are appended to the Approval Notes field in the parent contract record, located on the Approvals tab.



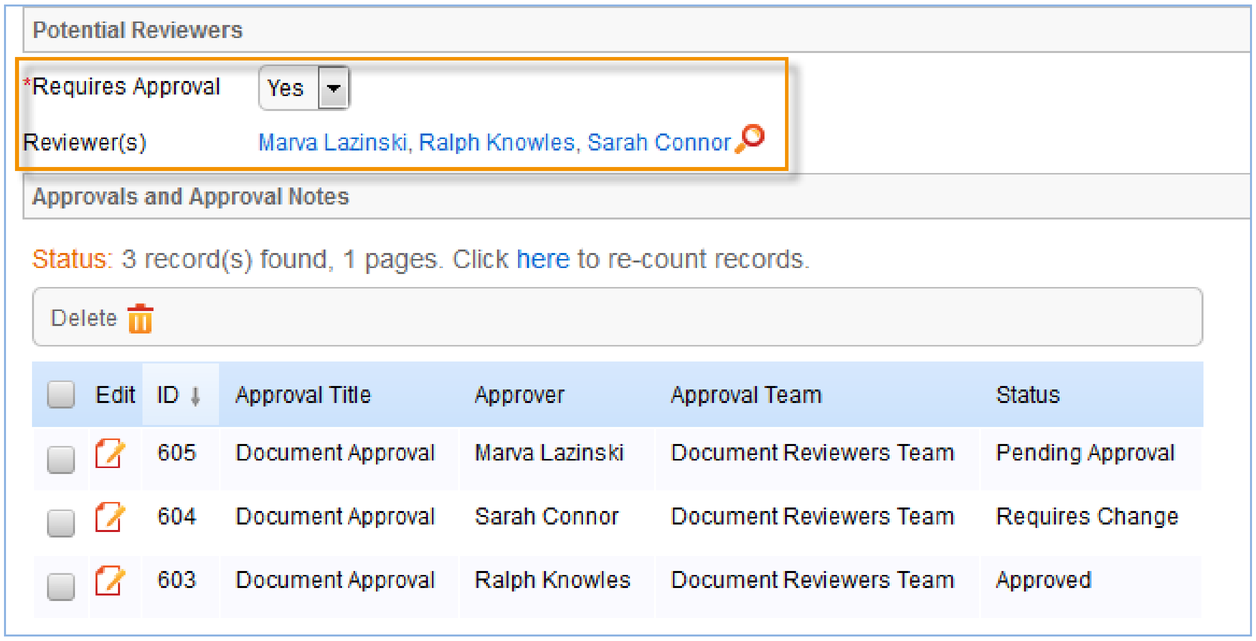
All Notes by Approvers

Approval Notes	[Agiloft System Mar 31 2015 10:56] This contract needs significant revision.

Only Approvers and members of the Approval Team can update approval records. Users updating the Status to Permanently Rejected or Requires Change must enter Approval Notes.

Use Case for Documents

To begin the approval process for documents, select Yes for Requires Approval on the Progress tab of a Document record. Select one or more Reviewer(s) who will be assigned to review and approve the document. Click Submit for Approval to generate the approval records.



Potential Reviewers

*Requires Approval	Yes
Reviewer(s)	Marva Lazinski, Ralph Knowles, Sarah Connor

Approvals and Approval Notes

Status: 3 record(s) found, 1 pages. Click here to re-count records.

Edit	ID	Approval Title	Approver	Approval Team	Status
	605	Document Approval	Marva Lazinski	Document Reviewers Team	Pending Approval
	604	Document Approval	Sarah Connor	Document Reviewers Team	Requires Change
	603	Document Approval	Ralph Knowles	Document Reviewers Team	Approved

Figure 5. Choose the document reviewers before generating approvals.

All Document approvals are created with an Approval Status of Pending Approval. Document approvals are parallel, i.e. all document-related approval records are created and a Step Number of "1" through conversion. Notes added to document approvals are appended to the Approval Notes field in the parent document.

For more information on approvals for documents, see the Documents Table section on **Error! Reference source not found.**

Ownership

Approval records and Approval Template records are owned by the user who creates them. Specifically, a record is owned by the user whose Login matches the Creator Login field.

TASK MANAGEMENT

There is a set of tables used to manage the automated creation of tasks that may appear in service requests, change requests, projects, and other places within the system. These tables include Tasks, Task Templates, Task Steps, and Task Workflows, which are each described below.

Task Workflows Table

Overview

Task Workflows are used to create a set of task templates that may be organized to trigger all at once or in a specified sequence.

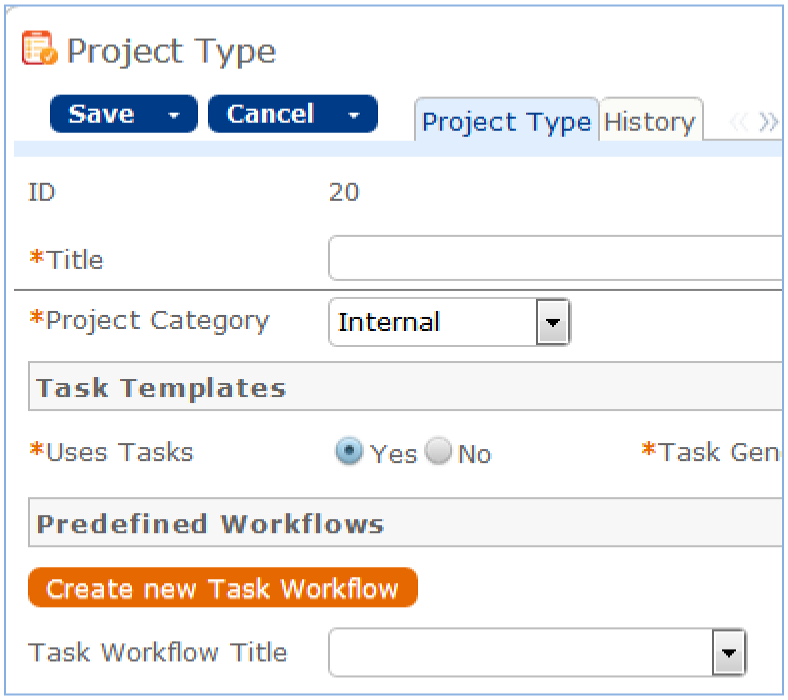
They can be used in services such as Change Requests or Services Requests, and they are also used in Project Types. They essentially predefine the tasks that should be done for a particular request type. They are used in conjunction with the Task Generation Method of Predefined Task Workflow.

Use Case

Task workflows can be created by members of the Project Manager, Change Manager, Service Manager, and Admin groups. All except admin users may only edit the workflows related to their primary tables, so change managers can edit workflows related to change requests, while project managers can edit workflows related to project types.

Creation of Records

A task workflow can be created directly in the Task Workflows table or it can be launched from within a service for a change request or service request or from a project type record, using an action button:



ID	20
*Title	
*Project Category	Internal

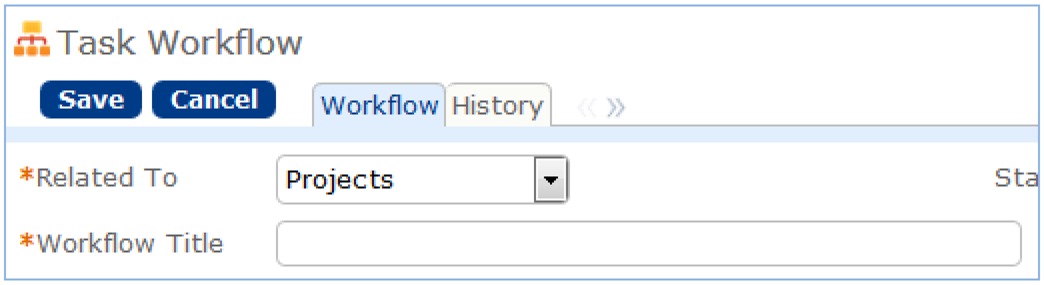
Task Templates

*Uses Tasks (Yes [✓] No) *Task Gen

Predefined Workflows

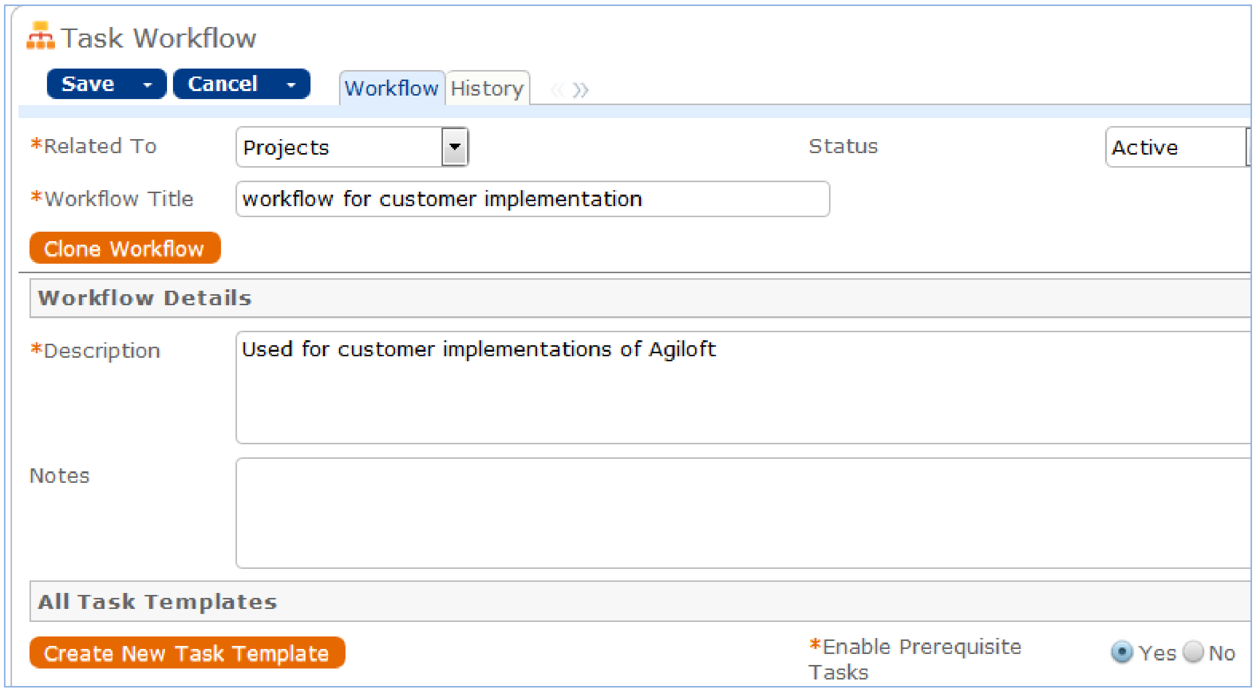
Task Workflow Title

The advantage of doing it from a service record or project type record is that the Related to field will be populated automatically with the correct table:



*Related To	Projects
*Workflow Title	

Either way, the process of setting up a new task workflow involves naming it with a unique name, ensuring it is related to the correct table, adding a description, and then creating the task templates:



*Related To	Projects	Status	Active
*Workflow Title	workflow for customer implementation		

Workflow Details

*Description	Used for customer implementations of Agiloft
Notes	

All Task Templates

*Enable Prerequisite Tasks [✓ Yes] No

Clicking the Create New Task Template button saves the workflow and then brings up the new task template screen (see above in the section on Task Templates for more information on how to create and fill out the task template form).

Once the first template is created, if the Enable Prerequisite Tasks is set to Yes in the workflow, then additional templates may select the earlier ones as prerequisites.

Prerequisite Task Handling

Within a single task workflow, there may be multiple threads of dependent tasks with different prerequisites, and these threads may proceed independently.

It is possible to launch two separate threads and bring them back into alignment by making their last tasks prerequisites of the next task, for instance.

Task templates can be conditional, and if their condition is not met, then they will be marked as not needed when they would otherwise be assigned. When they are marked as not needed, their dependent tasks will be assigned.

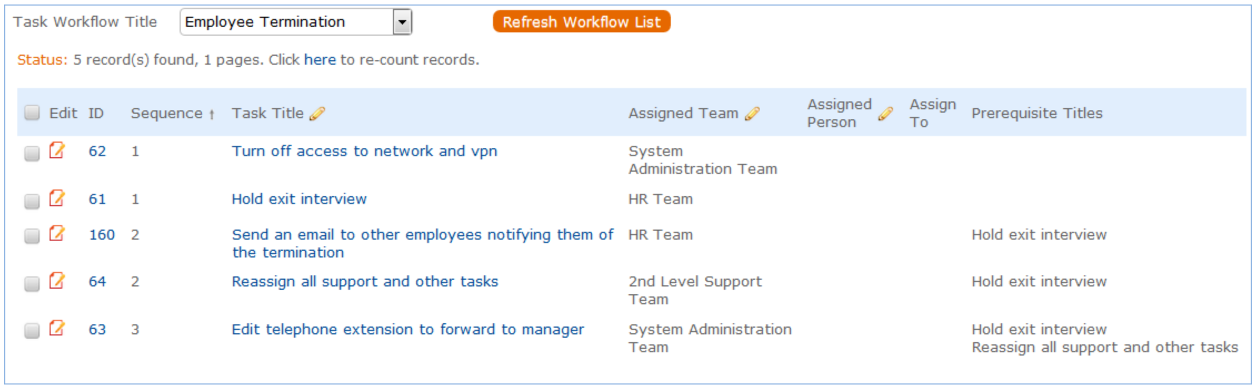
This is an important point: tasks that are dependent on conditional tasks are not prevented if the condition is not met. If they **should** be prevented, then they should also be made conditional based on the same condition as their prerequisite task.

Note that when tasks are launched for a project or a request using a Launch Tasks button, the Status of tasks will be changed to Assigned based on the following logic:

- The task status was Queued or Conditional
- The task has no prerequisites and is either not conditional, or its condition is met

- The task has some prerequisites but they are set to a status of Not Needed or Completed – i.e. they are conditional and are not needed, or they have been "removed" (marked as not needed) by a technician before the tasks are "launched"

Below is an example of a task workflow for employee termination:



Status: 5 record(s) found, 1 pages. Click here to re-count records.

Edit	ID	Sequence	Task Title	Assigned Team	Assigned Person	Assign To	Prerequisite Titles
	62	1	Turn off access to network and vpn	System Administration Team			
	61	1	Hold exit interview	HR Team			
	160	2	Send an email to other employees notifying them of the termination	HR Team			Hold exit interview
	64	2	Reassign all support and other tasks	2nd Level Support Team			Hold exit interview
	63	3	Edit telephone extension to forward to manager	System Administration Team			Hold exit interview Reassign all support and other tasks

Here there are two tasks that will be assigned right away (ID62 and ID61 - those with no prerequisites). The tasks with a sequence value of 2 will be assigned as soon as the Hold Exit interview is completed, so they might get assigned before ID62 is completed. The final task will be assigned as soon as the exit interview is completed and the support and tasks are reassigned. So it is conceivable that ID62, ID160 and ID63 are all in progress at the same time and any may be completed first.

The ability to choose exactly which tasks must be finished before a task is assigned gives a very flexible model for complex task workflows.

The sequence number gives a general impression of the order of tasks, but it is not definitive and does not control any of the processing. It allows the tasks to be sorted more or less in order of assignment.

Cloning a Task Workflow

Workflows can also be cloned using the Clone Workflow button. The effect of cloning a workflow is to create a new workflow with the workflow title changed to prefix it with information that is was cloned, i.e. Cloned on 05/27/2016 02:58 - General IT Project. The task templates are also cloned and linked to the new workflow.

Note that task templates may only be linked to one workflow and are created independently for each workflow, since they may have a sequence and order within that workflow that would be different within a different workflow.

Task Templates Table

Overview

This table holds records that serve as templates for automatically generated tasks. Each Template record specifies a Task related to one of the other tables. Task Templates have a Related to field, just as tasks do, and they may be linked to a specific service, task workflow, or project type.

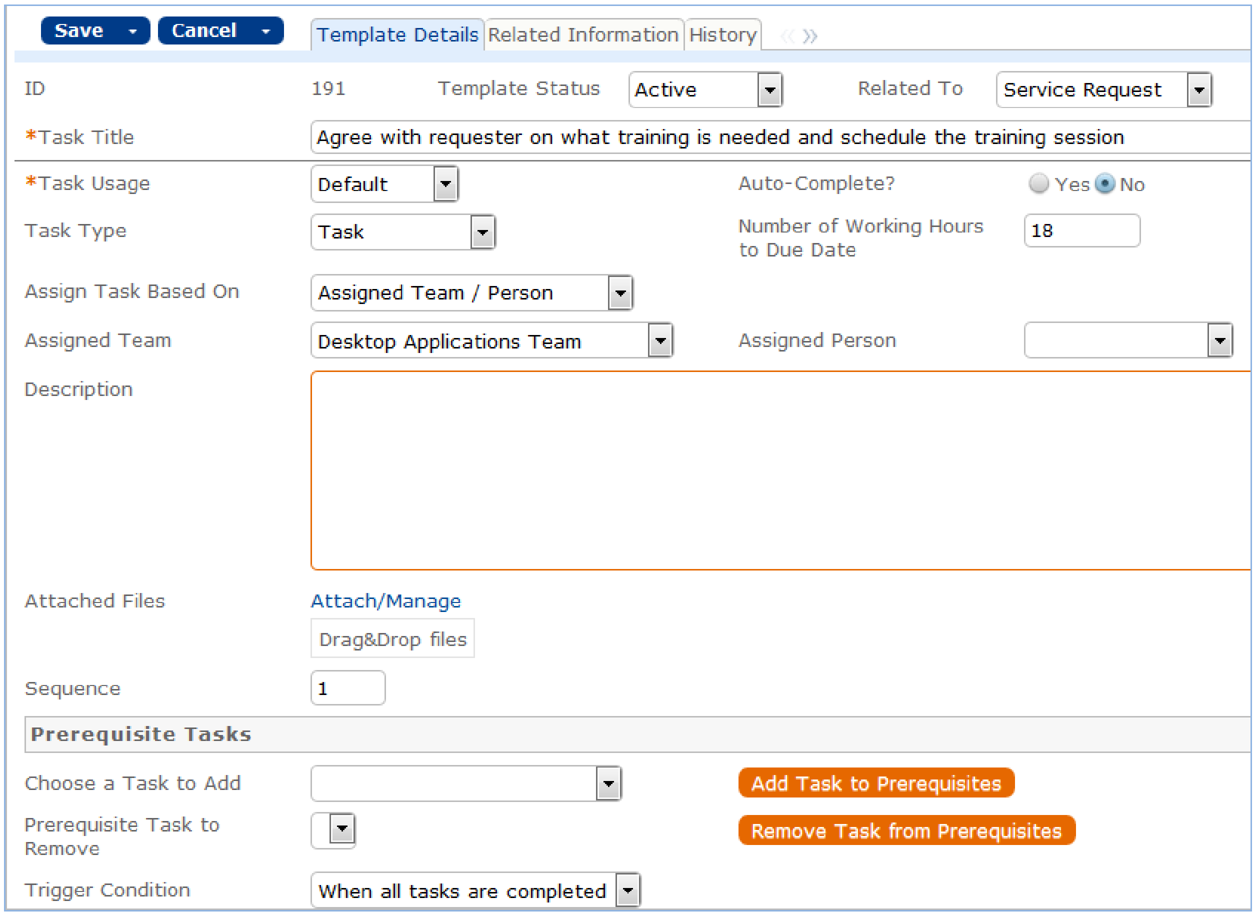
They define the method of assignment, the expected number of working hours to complete, and any prerequisite tasks.

Task Templates may be combined within a Task Workflow, or they may exist independently or a workflow. They can be combined to form a set of "User Selected Tasks" linked to a service or project type. Or a single task template may be linked to a service for a change request.

Overview of the Task Template Layout

Template Details Tab

The user entered fields are all on the Template Details tab:



ID	191	Template Status	Active	Related To	Service Request
*Task Title	Agree with requester on what training is needed and schedule the training session				
*Task Usage	Default	Auto-Complete?	Yes [✓ No]		
Task Type	Task	Number of Working Hours to Due Date	18		
Assign Task Based On	Assigned Team / Person				
Assigned Team	Desktop Applications Team	Assigned Person			
Description					
Attached Files	Attach/Manage				
Sequence	1				

Prerequisite Tasks

Choose a Task to Add	
Prerequisite Task to Remove	
Trigger Condition	When all tasks are completed

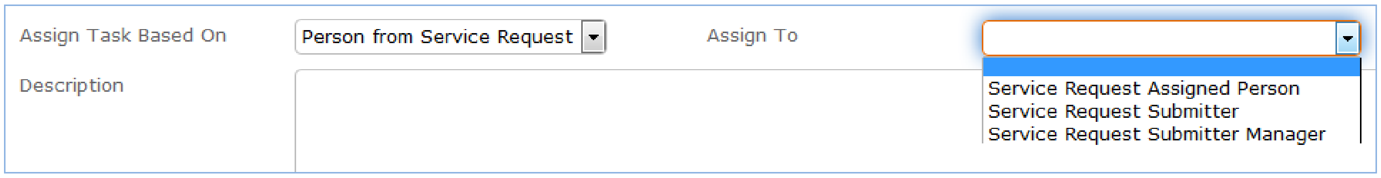
The title may not contain any commas.

Task templates may be related to Service Requests, Change Requests, and Projects.

The Task Usage may be Default or Conditional. If conditional, then a saved search condition based on meta data in the record where the task is generated should be defined. At the point where the task would be assigned, the condition is checked to see if it is met or not and the task's status is changed accordingly.

The Number of Working Hours to Due date is used to set the task's Date Due once it is set to a status of Assigned, by adding the defined number of working hours (for the assigned team) to the current time.

The Assign Task Based On allows you to define a task template whose resulting task will be assigned to a person defined in the main record. The default of Assigned Team / Person lets you “hard code” the assigned team or person in the task template. A different value allows you to choose a variable that has been preconfigured in the Replacement Variables table. For instance, for a task related to a service request you will see these options:



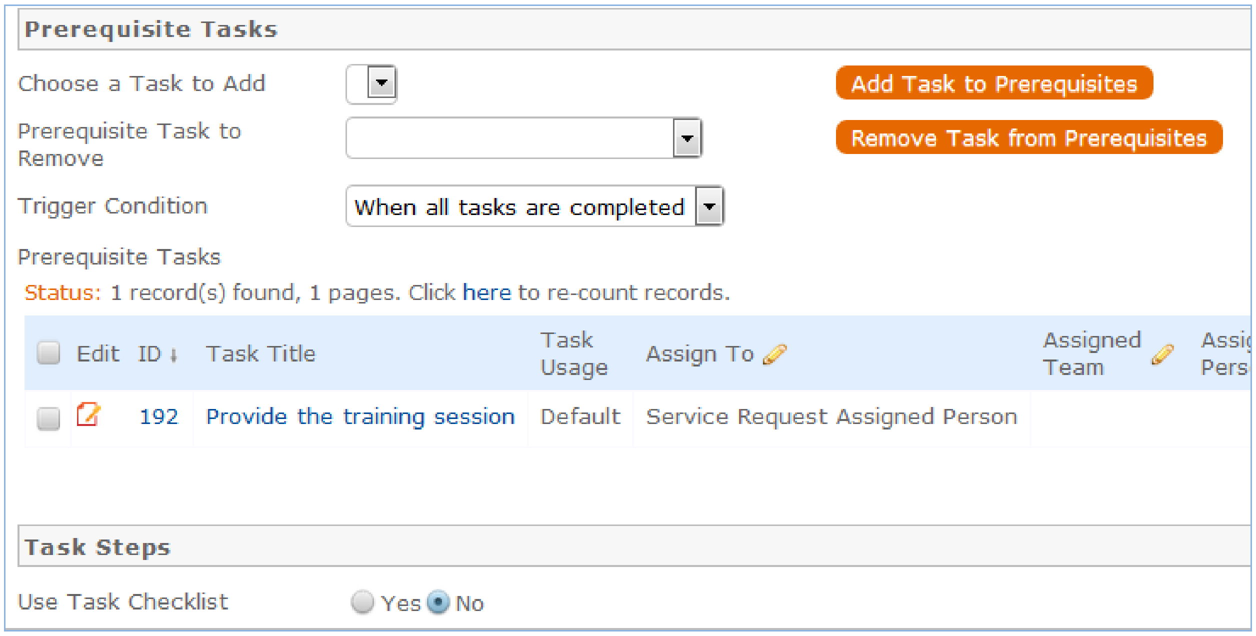
Assign Task Based On	Person from Service Request	Assign To	
Description			Service Request Assigned Person Service Request Submitter Service Request Submitter Manager

The choices in the Assign to field are set in the Replacement Variables table and can be edited there, or new fields added.

Note that if you set a task to be assigned in this way, but there is no value in the selected field -- for instance, there is no submitter manager defined in the service request-- then the task will not be properly assigned. As a backup, such tasks are assigned to the 1st level Support Team.

The Choose a Task to Add field is used to select a task for the same workflow to be a prerequisite task to this one. The Prerequisite Task to Remove is used to remove a task that was previously defined as a prerequisite. These fields are used in conjunction with the action buttons to their right.

Note that prerequisite task setup is only visible and enabled if the task template is linked to a Task Workflow and if the linked field Workflow Enable Task Prerequisites=Yes. Otherwise these fields will not be visible.



Prerequisite Tasks

Choose a Task to Add		Add Task to Prerequisites
Prerequisite Task to Remove		Remove Task from Prerequisites
Trigger Condition	When all tasks are completed	

Prerequisite Tasks

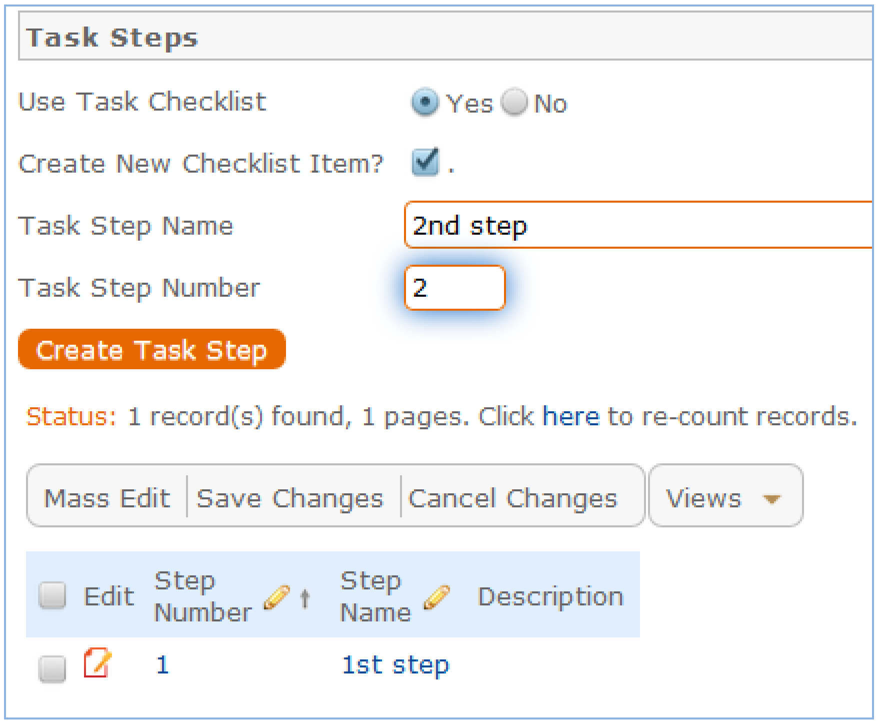
Status: 1 record(s) found, 1 pages. Click here to re-count records.

	Edit	ID ↓	Task Title	Task Usage	Assign To	Assigned Team	Assi Pers
☐		192	Provide the training session	Default	Service Request Assigned Person		

Task Steps

Use Task Checklist ○ Yes [✓ No]

At the bottom of the tab is the option to set up a task checklist. If Yes is selected, then you can add individual steps and define the order the checkboxes should use:


Task Steps

Use Task Checklist	[✓ Yes] No
Create New Checklist Item?	✓ .
Task Step Name	2nd step
Task Step Number	2

Status: 1 record(s) found, 1 pages. Click here to re-count records.

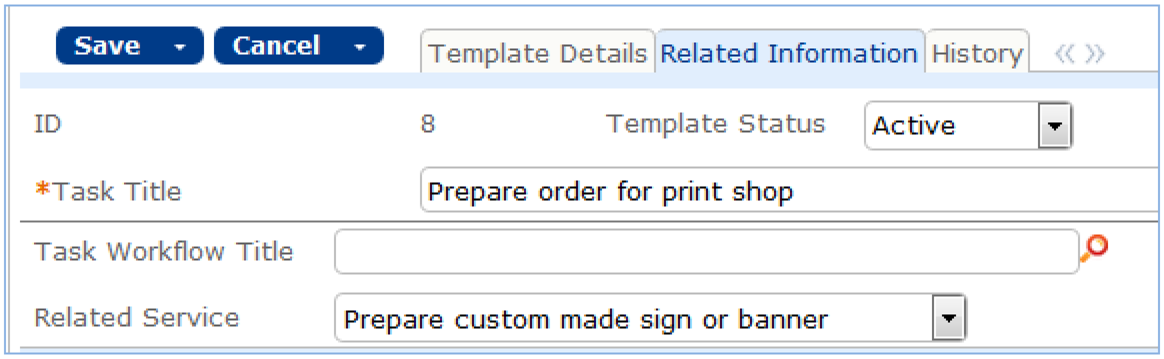
	Edit	Step Number	Step Name	Description
		1	1st step	

To add steps, first check the Create New Checklist Item? Box, then fill out a name and the step number, then click the Create Task Step button.

This is a handy way to provide a list of troubleshooting steps or a list of things to be done within a single task.

Related Information Tab

This tab may show the record that contains this task template. If the template is part of a task workflow, the workflow will be listed. If it is linked to a particular service as a "User Selected Task", then the service name will be linked in:



ID	8	Template Status	Active
*Task Title	Prepare order for print shop		
Task Workflow Title			
Related Service	Prepare custom made sign or banner		

Use Case

Task Templates are related to either the Project, Change Request, or Service Request tables using the "Related to" field at the top of the form.

Whenever a Task Template is used to generate a Task, a conversion is done from the Task Template table to the Task table, mapping important information in the Task Template to the Task record that is generated. This Task record is linked to the Service Request, Project, or other table that spawned it, as well as being linked back to the source task template.

Templates can be made inactive when they should no longer be used, and that will prevent them from being generated as tasks. When the Status is changed to Inactive, an edit rule called Edit: Other edit actions - unlink inactive task template, handle cloned templates will blank out any link to a task workflow, project type, or service will be eliminated so it will no longer appear as available in those services or new projects.

Another rule prevents task titles from having commas in them, and if a comma is stripped out, provides a popup message alerting the creator that it has been stripped.

Automation and Workflow

The automation on the Task Template table is almost all designed to handle the conversion into tasks when triggered from elsewhere, rather than to manage a process within the Task Template itself. It is described in the appendix.

In general, there are "flag" fields in the task template that get set to Yes to trigger conversion for each of the different main record types, and linked fields to the source records so that the new tasks that are triggered can be linked to the correct service request, project, or change request.

In addition, we keep track of all service requests and projects for which a task template has been generated so that if a user tries to generate the same task for the same parent record, the system prevents it. Since change request tasks must be generated multiple times, one for each CI, we cannot block this in the same way.

Cloning Task Templates Along with a Task Workflow

Task workflows may be cloned and then modified, and when a task workflow is cloned, its task templates are also cloned and linked to the new workflow.

Cloning reproduces all the elements of the original task template except any task steps. If task steps were involved in the task template, they will need to be recreated in the cloned template.

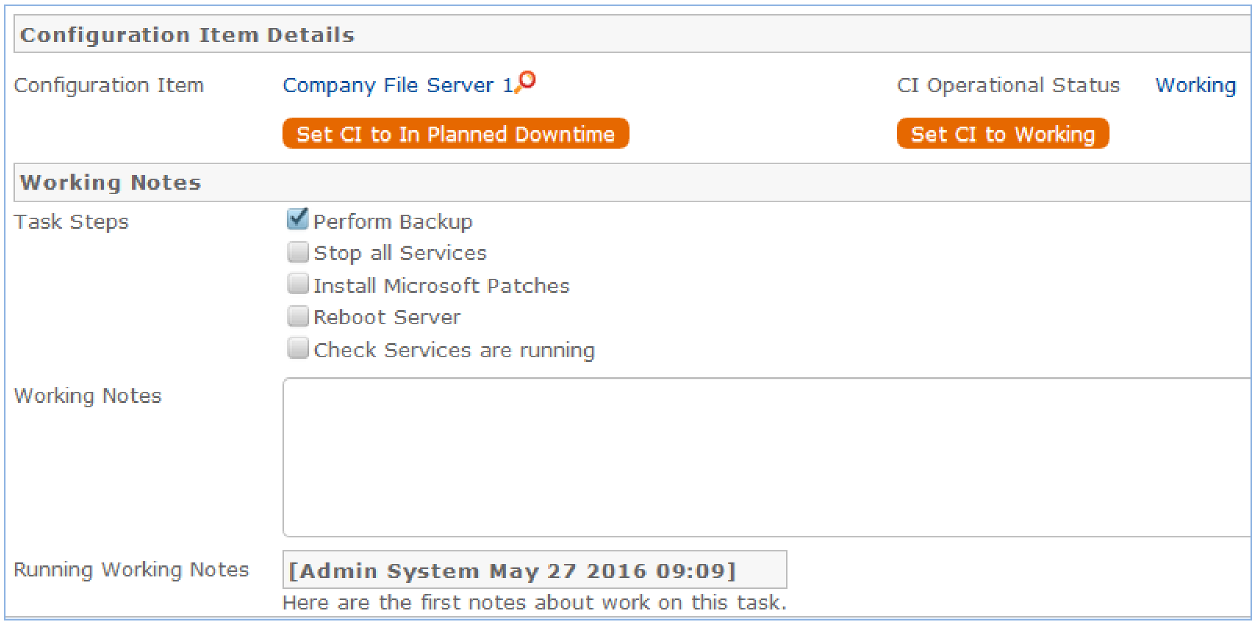
Task Steps Table

Overview

Task steps are used to define a multi-step process within a single task rather than creating several tasks. This makes sense especially when the same person is performing all the steps.

Individual records are linked to a specific task template and can only be used in that template. They are very simple records with just a title and a sort order.

They appear within task records as a set of checkboxes that are ordered based on the sort order:


Configuration Item Details

Configuration Item	Company File Server 1	CI Operational Status	Working

Working Notes

Task Steps
- [x] Perform Backup
- [ ] Stop all Services
- [ ] Install Microsoft Patches
- [ ] Reboot Server
- [ ] Check Services are running

Working Notes

Running Working Notes
[Admin System May 27 2016 09:09]
Here are the first notes about work on this task.

Use Case

Task steps are always created from within a task template, as shown above in the task template section. They should not be created directly, since without being linked to a task template, they will never appear anywhere.

There is no automation on this table.

Tasks Table

Overview

Like Services, the Tasks table is a global table that holds records that are often linked to and appear within the records in other tables.

The Tasks table holds individual tasks. While it may be used for tasks of many different kinds, the system is currently set up to allow tasks to be related to Service Requests, Change Requests, and Projects, as well as to Configuration Items, or to be completely independent of other tables. It is possible to modify the setup to relate tasks to records in any table and to show embedded tasks in any other table.

Fields related to task generation and an embedded Tasks table (labeled Tasks) are shown in service requests and change requests for services that include tasks.

Tasks and the fields for generating them are also shown within Projects and Configuration Items. Projects allow the same methods of task generation as service requests, while Configuration Items allow single tasks to be created and linked to the CI, and they also show all tasks that have been created from other records (change requests or service requests) that relate to that CI.

Within the task table, the field Related to indicates what type of record the task is associated with.

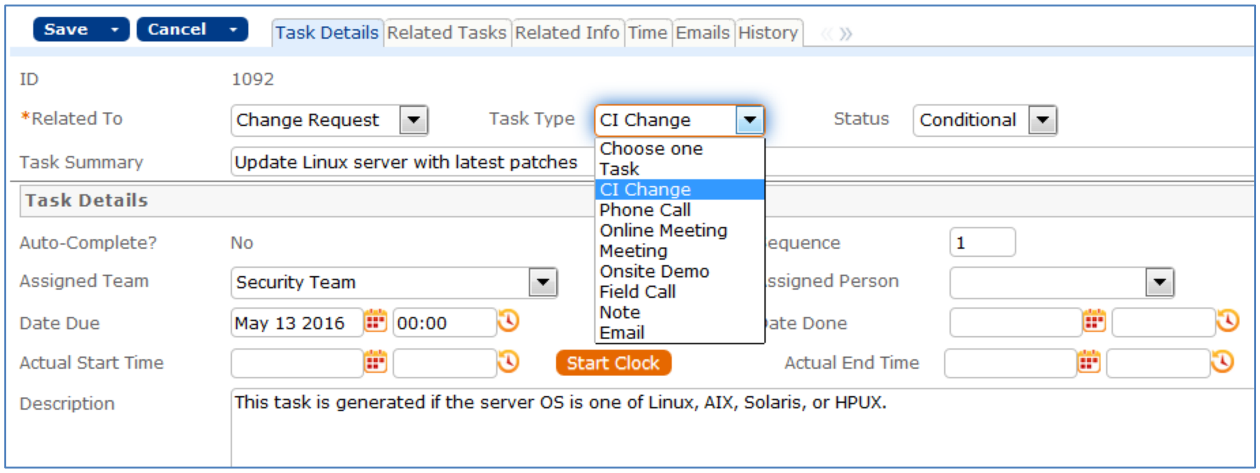
If you need to use tasks within any additional tables, you can modify this field and add that table to the choice list. Then create a linked set of fields from that other table, and make these fields visibility dependent on the value in the Related to field just as we have done for the linked Service Request and

Change Request fields. Then create a related table in the other table pointing to the Tasks table, and you will be able to generate tasks from there. This advanced functionality is best done after attending a training class or with our professional consulting team's assistance.

Overview of the Task Layout

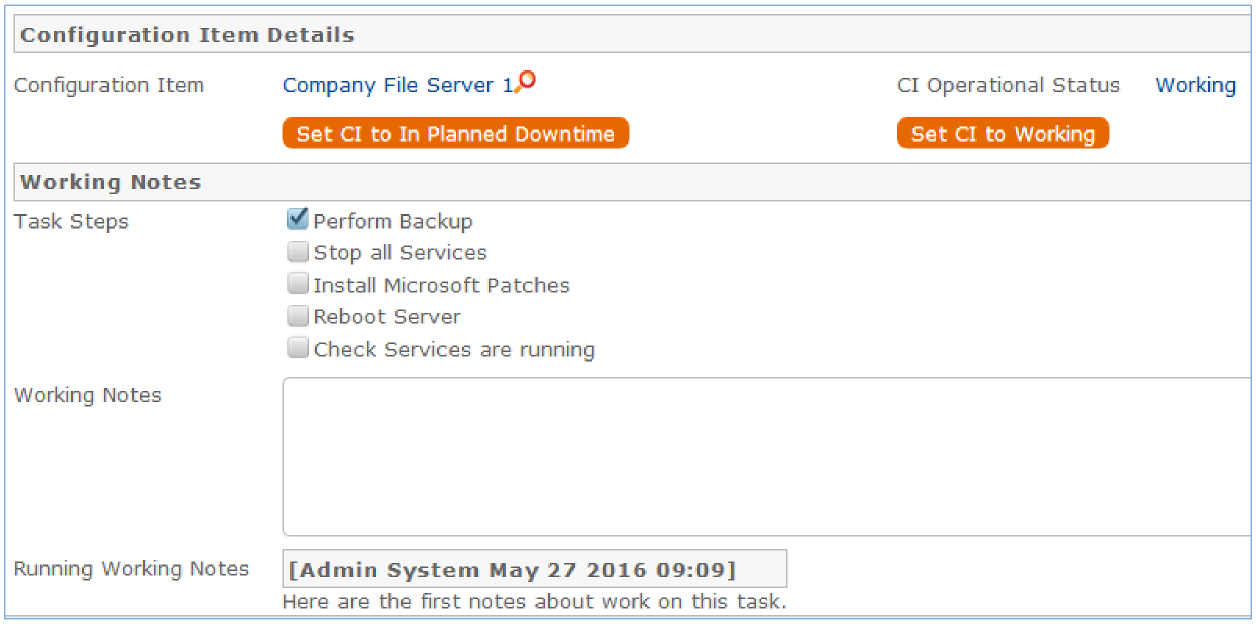
Task Details Tab

The main task screen has fields for assignment, the Task Type, its Status, its Date Due, and so on.



ID	1092
*Related To	Change Request
Task Type	CI Change
Status	Conditional
Task Summary	Update Linux server with latest patches
Task Details	
Auto-Complete?	No
equence	1
Assigned Team	Security Team
ssigned Person	
Date Due	May 13 2016 00:00
ate Done	
Actual Start Time	
Actual End Time	
Description	This task is generated if the server OS is one of Linux, AIX, Solaris, or HPUX.

It also includes an area for managing the related configuration item, if any, and a place to add working notes and to display all the running working notes that have been added:



Configuration Item Details

Configuration Item	Company File Server 1
CI Operational Status	Working

Working Notes

Task Steps
- ✓ Perform Backup
- ☐ Stop all Services
- ☐ Install Microsoft Patches
- ☐ Reboot Server
- ☐ Check Services are running

Working Notes

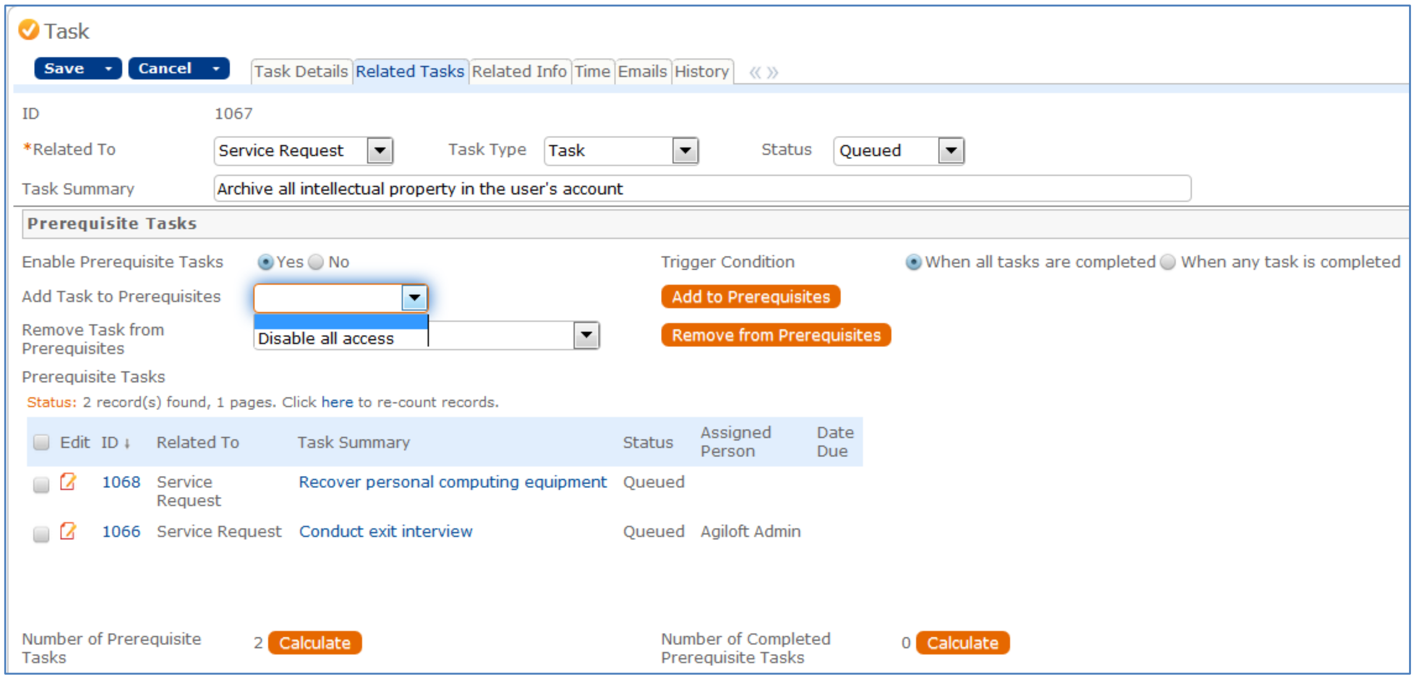
Running Working Notes
[Admin System May 27 2016 09:09]
Here are the first notes about work on this task.

When working on a CI-related task, the technician can click one of the buttons shown above to update the Operational Status of the CI to reflect that he has taken it offline or brought it back online.

Some tasks may have Task Steps defined. These steps are set up in the task templates. Task steps will appear as checkboxes above the Working Notes field. There are no default rules enforcing that all checkboxes are checked before completing a task, but such a rule could be easily added.

Related Tasks Tab

The Related Tasks tab shows the prerequisite tasks, if any, and allows the current task to be related to prerequisite tasks within in the same parent record. It also shows any dependent tasks (those for which this task is a prerequisite):



ID 1067
*Related To: Service Request
Task Type: Task
Status: Queued
Task Summary: Archive all intellectual property in the user's account

Prerequisite Tasks

Enable Prerequisite Tasks: [✓ Yes] No
Trigger Condition: [✓ When all tasks are completed] When any task is completed
Add Task to Prerequisites
Remove Task from Prerequisites: Disable all access
Prerequisite Tasks
Status: 2 record(s) found, 1 pages. Click here to re-count records.

Edit	ID ↓	Related To	Task Summary	Status	Assigned Person	Date Due
	1068	Service Request	Recover personal computing equipment	Queued		
	1066	Service Request	Conduct exit interview	Queued	Agiloft Admin	

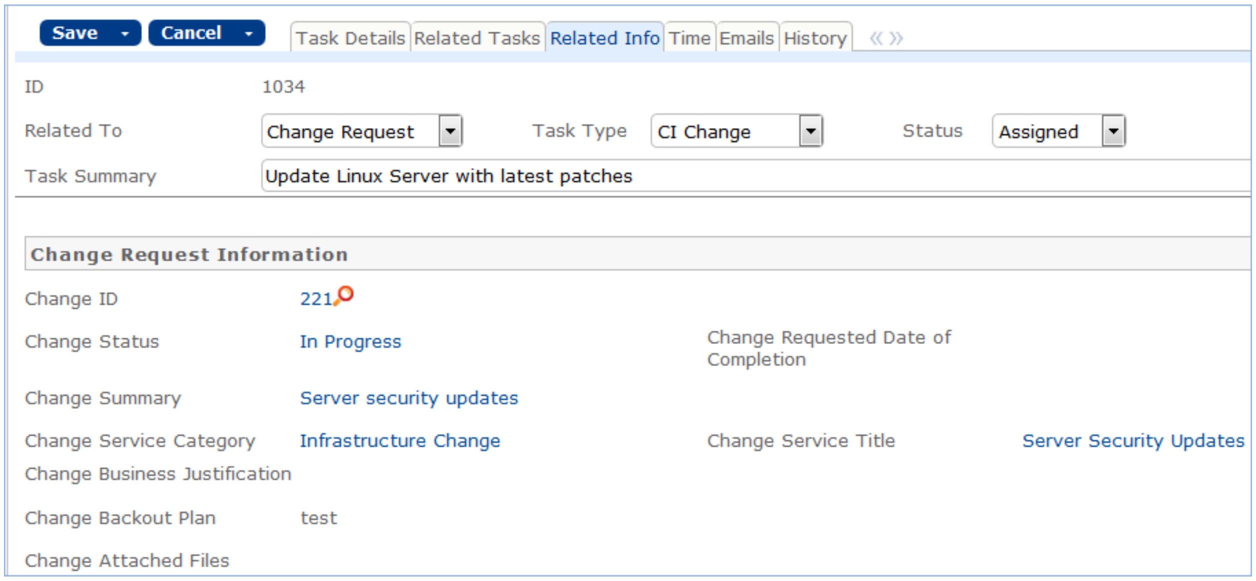
Number of Prerequisite Tasks: 2
Number of Completed Prerequisite Tasks: 0

In the above example, we see a task that has two prerequisites already defined. The Trigger Condition defines whether this task will be set to Assigned only when both prerequisites are done, or as soon as any of them are done. Additional tasks associated with the same Service Request can be added to the prerequisites by selecting the task and clicking the Add to Prerequisites button. Either of these two tasks could be removed by selecting it in the Remove Task from Prerequisites field and clicking the Remove from Prerequisites button.

The default Status for a task is Queued when it is created. Typically a rule will set the Task to Assigned when the tasks for a record are launched (see below for more details).

Related Info Tab

This tab shows details about the record linked to the task, which will typically be either a project, service request, or change request. It provides hyperlinks to get to the source request to see more information.



ID	1034				
Related To	Change Request	Task Type	CI Change	Status	Assigned
Task Summary	Update Linux Server with latest patches				

Change Request Information

Change ID	221		
Change Status	In Progress	Change Requested Date of Completion	
Change Summary	Server security updates		
Change Service Category	Infrastructure Change	Change Service Title	Server Security Updates
Change Business Justification			
Change Backout Plan	test		
Change Attached Files			

Other Tabs

The Time tab allows time to be entered and shows all time entered for the task. Note that any time entered for the task will also roll up into the request to which the task is linked.

The Emails and History tabs hold the standard fields.

Use Case

Task Creation from a Template

Tasks are most often created when a user clicks a button to Generate Tasks from the record in which the tasks will be done, i.e. from within a Project, Service Request, or Change Request. Such tasks are generated from Task Templates that have been created previously and defined to be used for the particular project or service type.

When generated from a template, they will be auto-assigned to the appropriate team or person based on the task template record.

Note that currently, users are prevented from creating tasks whose task title has a comma, since this breaks some of the automation for prerequisite tasks. If a task is created with a comma, the comma is stripped out. Commas are also prevented in task templates for the same reason.

Ad Hoc/Manual Task Creation

Tasks can also be created manually outside of any other record or by clicking a button to create an ad hoc task from within one of the request records. In this case, the user will choose the Assigned Team and/or Assigned Person and may also set a Date Due, select prerequisite tasks, and so on.

Creation Automation

When a task is created manually, if it is related to a Project whose status is Completed or Cancelled, the user is prevented from saving the task.

If the Date Due is in the past when the task is created, the user is warned but allowed to correct or save the task. These actions are done by the rule: Create: All create validations.

Next a rule called Create: All Creation Actions runs, and it performs several actions based on the record the task is related to. If the task template usage is Conditional, then the Status of the task is set to Conditional. Otherwise the default status is Queued.

If the task was assigned to an individual from the related record (i.e. the Change Manager or Project Manager), then the Assigned Team is set to that person's Primary Team. If no one was assigned at all due to some failure of the template, then the Assigned Team is set to the 1st Level Support Team.

If the task is created in a Status of Assigned, then if the Assigned Person is not the creator and there is an assigned person, that person is emailed, otherwise the assigned team is emailed.

If the task is for a change request and was generated as a single task, its sequence value is set to 1.

If the task source is a task template that had prerequisite templates, then the corresponding tasks are set to be prerequisites. The prerequisites are then sorted and the one with the highest sequence value is set in another linked field called Highest Sequence.

A rule called Create/Edit: Update Sequence based on Highest Sequence than runs to set the new task's sequence value to the highest sequence plus 1.

Note that the Sequence field is purely informational. No automation is triggered based on the sequence, but it is there to provide a general idea of the order in which tasks will be triggered and completed.

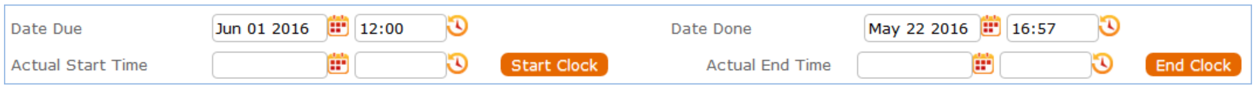
Processing a Task

The person assigned to a task can add working notes to it, refer to the linked Service Request, Project, or Change Request from it, and ultimately complete the task. There are several default statuses: Queued, Conditional, Assigned, Completed, Not Needed, Failed, and Waiting for Others.

When the person has completed the task, he may change the Status to Completed and save. If there is no value in the Date Done field, the system will put the current date/time into the Date Done field. He may alternatively choose to enter a time in that field directly. The system allows the user to override the value in that field.

While working on the task, the user may enter any time spent on the task in the Time Spent and Time Description fields, then click "Add Time" to convert them to Time Entries.

When working on a CI, the Start Clock and End Clock buttons can be used to set the Actual Start Time and Actual End Time as needed. The user may also manually put values in these fields.



Date Due	Jun 01 2016	12:00		Date Done	May 22 2016	16:57
Actual Start Time			Start Clock	Actual End Time		

When a task record is saved the rule called Edit: All Edit Actions (API enabled) is run.

If the Status has changed to Completed, Failed, or Not Needed, the system updates the Dependent tasks by refreshing their Number of prerequisite tasks counts. If this was the last prerequisite, then that will trigger the next rule to assign the dependent tasks.

The rule also sets the Assigned Person, if it is blank, to the person who completed the task.

If the task is for a project, the project manager is notified of the task completion.

If the working notes field was updated, the text is copied into the Running Notes field and blanked out. And if the task is related to a change request, the notes are also updated into the Change Request's running working notes.

If the status has just changed to Assigned (by launching the tasks from the main record or by the prerequisite tasks having been completed), the Date Due is set based on specified criteria. If the task was an auto-completing task, then it is marked as Completed and the assignee notified. Otherwise, the assigned person or assigned team is notified that the task is now assigned.

When a prerequisite task is completed, its dependent tasks are updated, and if all prerequisite tasks are now completed, then the rule called Edit: Assign tasks when number of completed prerequisites meets criteria (API) runs. This rule checks if the task was conditional, and if so, checks the condition to see if it is met. If the task is not conditional or its condition is met, then it sets the Status to Assigned. Otherwise, it sets the status to Not Needed.

When all tasks for a particular record are completed or marked as failed or not needed, the person / team assigned to the main request is notified that all tasks are done.

Measuring Time for a Task

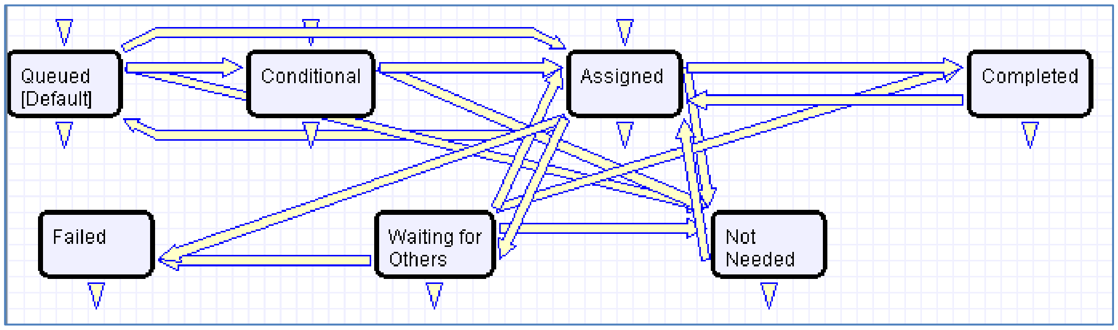
In addition to time that is manually entered, the system tracks two kind of elapsed time. The Working Hours to Complete field is set to the difference between the Date Created and Date Done, excluding the non-working hours of the assigned team and also excluding the time during which the Status was Queued, Conditional, or Not Needed.

The Actual Working Hours field measures the time between the Actual Start Time and Actual End Time, excluding the non-working hours of the assigned team.

These fields can be used in reports to see the average amounts of time tasks of specific types are taking. They can also be compared against the Template Number of Hours to Due Date value, which sets the expected working hours that should be needed for the task.

Automation and Workflow

There is a simple workflow for tasks that currently executes no actions:



Time Based rules

There are two time based rules that are set up but not running.

The first is TB: (DISABLED) Notify of upcoming task. It will notify the assigned person or team when the due date is one day away, once it is turned on. Currently it is disabled. There is a radio button at the bottom of the General tab in the table settings if this rule is desired. The schedule may need to be changed from every 10 years to something more useful.

The second is TB: (DISABLED) Set alert color to red if overdue. This sets the alert color to red when the date due has passed, so that views that use row coloring can show to users that the task is overdue.

SERVICE OPERATION TABLES

This section describes the setup for the main process tables involved in running the support operation: Service Requests, Incidents, Problems, Change Requests, Configuration Items, and Purchase Requests.

Service Requests Table

Overview

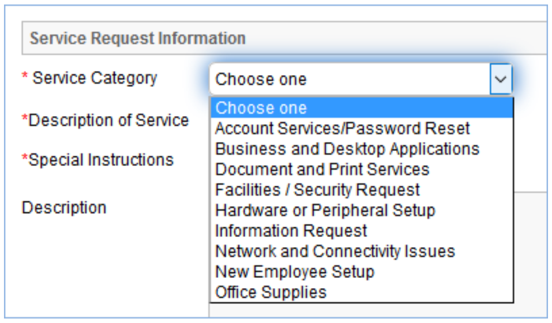
The Service Requests table holds all customer service requests. A service request (SR) is created when an internal customer needs a new service, a standard (pre-approved) change to an existing service, a password reset, or some other kind of assistance or information.

When the user needs to report an interruption in service or a problem with an existing service, an Incident should be created instead. We have chosen to separate service requests from incidents because these two kinds of request have rather different workflows and resolution criteria. An incident may often require the generation of a Problem and a Change Request, while Service Requests will generally be self-sufficient and relatively quick to resolve.

The easy way to distinguish Service Requests from Incidents is to recognize that Service Requests are the kinds of things you will choose from a service catalog, such as application assistance, password resets, new employee setup, documentation or copying services, and so on.

If a user creates a Service Request when he should have created an Incident, the technician may click the Save Changes and Copy to Incident button on the Related Records tab of the SR form to easily convert that Service Request into an Incident and have all the fields mapped appropriately.

The Service Request table uses the same choice list of service categories as the Services table.


Service Request Information

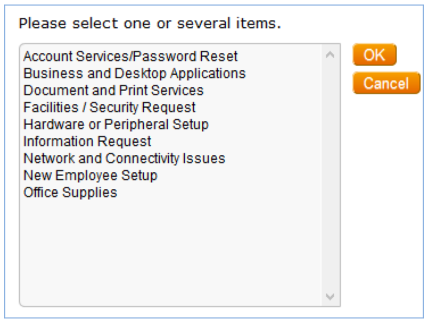
* Service Category: Choose one

*Description of Service

*Special Instructions

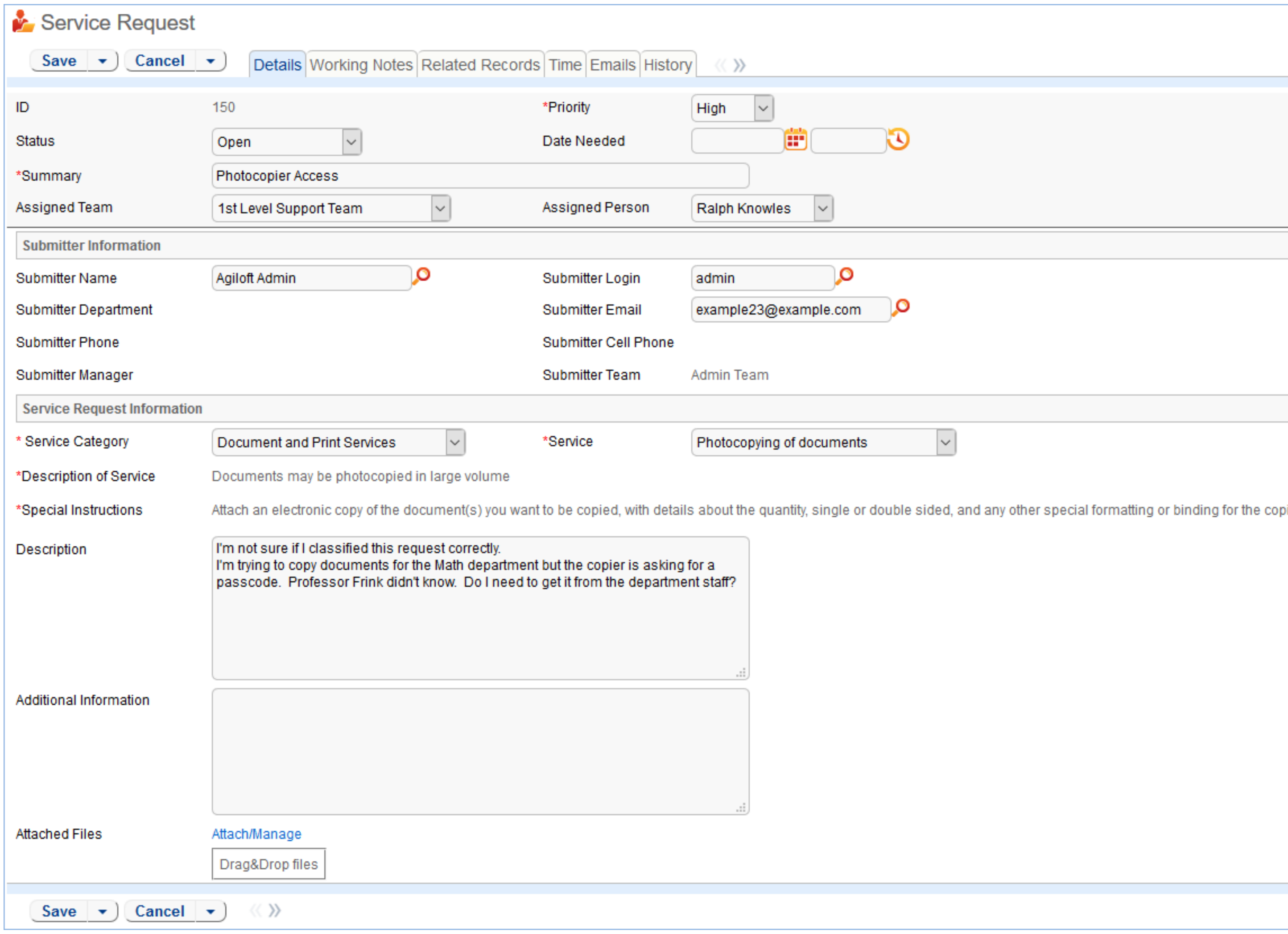
Description

Choose one
Account Services/Password Reset
Business and Desktop Applications
Document and Print Services
Facilities / Security Request
Hardware or Peripheral Setup
Information Request
Network and Connectivity Issues
New Employee Setup
Office Supplies


Please select one or several items.

Account Services/Password Reset
Business and Desktop Applications
Document and Print Services
Facilities / Security Request
Hardware or Peripheral Setup
Information Request
Network and Connectivity Issues
New Employee Setup
Office Supplies
OK
Cancel

A user creating a service request chooses the category of service from a drop-down, and then sees all the active services that belong to that category in a second drop-down list. The default service categories for service requests can be seen in the services table (or by creating a SR record and opening the drop-down list).



ID	150	*Priority	High
Status	Open	Date Needed	
*Summary	Photocopier Access		
Assigned Team	1st Level Support Team	Assigned Person	Ralph Knowles

Submitter Information

Submitter Name	Agiloft Admin	Submitter Login	admin
Submitter Department		Submitter Email	example23@example.com
Submitter Phone		Submitter Cell Phone	
Submitter Manager		Submitter Team	Admin Team

Service Request Information

* Service Category	Document and Print Services	*Service	Photocopying of documents
*Description of Service	Documents may be photocopied in large volume		
*Special Instructions	Attach an electronic copy of the document(s) you want to be copied, with details about the quantity, single or double sided, and any other special formatting or binding for the copi		
Description	I'm not sure if I classified this request correctly. I'm trying to copy documents for the Math department but the copier is asking for a passcode. Professor Frink didn't know. Do I need to get it from the department staff?		
Additional Information			
Attached Files	Attach/Manage Drag&Drop files		

Use case

Service Requests (SR's) may be created by telephone support staff or technicians on behalf of customers or by customers directly through the web interface or by inbound email, should an inbound email account be set up.

End User Record Submission

When a customer submits a Service Request, the contact information fields automatically populate based on the details in his/her User record. In the default setup, a request cannot be submitted on behalf of a user who has no user record in the system. This may easily be changed by modifying the linked fields from the User table to enable "non-source" values to be used.

The user is required to select a service after selecting a service category. When the service is selected, the user will see a description of the service, any special instructions for that service, and any additional dependent fields defined to be visible for that service.

He may then provide further information and click Save to save the form, after which he will receive an email acknowledging receipt of the request.

Automatic Assignment of New Requests

When a new request is saved, it is automatically assigned to a team based on the values in three fields pulled from other tables (Rule: All new Service Request actions, Action: Assign Service Request). The related service has a Responsible Team associated with it and a field called Give Service Team Priority over CI Team that has Yes/No values. If a Configuration Item was identified for the request, then there is also a CI Responsible Team field that will be populated in the SR.

The current assignment logic is: if the Service Request involves a specific configuration item (CI), then the request is assigned to the responsible team for that CI, unless the Give Service Team Priority over CI Team field is set to Yes. In this case, or If no CI is defined, then it is assigned to the Responsible Team for the Service. If there is no Responsible Team for the Service, then it is assigned to the 1st Level Support Team.

Naturally, this logic may easily be changed to suit your organization. You can simply set a default value such as the 1st Level Support Team or you may choose some other logic to use based on different fields. The current logic is implemented in the SR rule named “All New Service Request Actions” and may be modified there.

Technician Record Submission

If a technician creates the SR, he may choose an assigned team and assigned person from the drop-down fields, and his changes will not be overwritten – the rule only assigns if the Assigned Team field is empty when the record is saved.

New SRs are created in a Status of Open by default. During creation, a technician may change the Status to In Progress (if they wish to indicate they are already starting work on it) or Closed (if they were able to resolve the issue immediately and want to capture the request for reporting purposes).

Automatic Emails Sent upon Submission

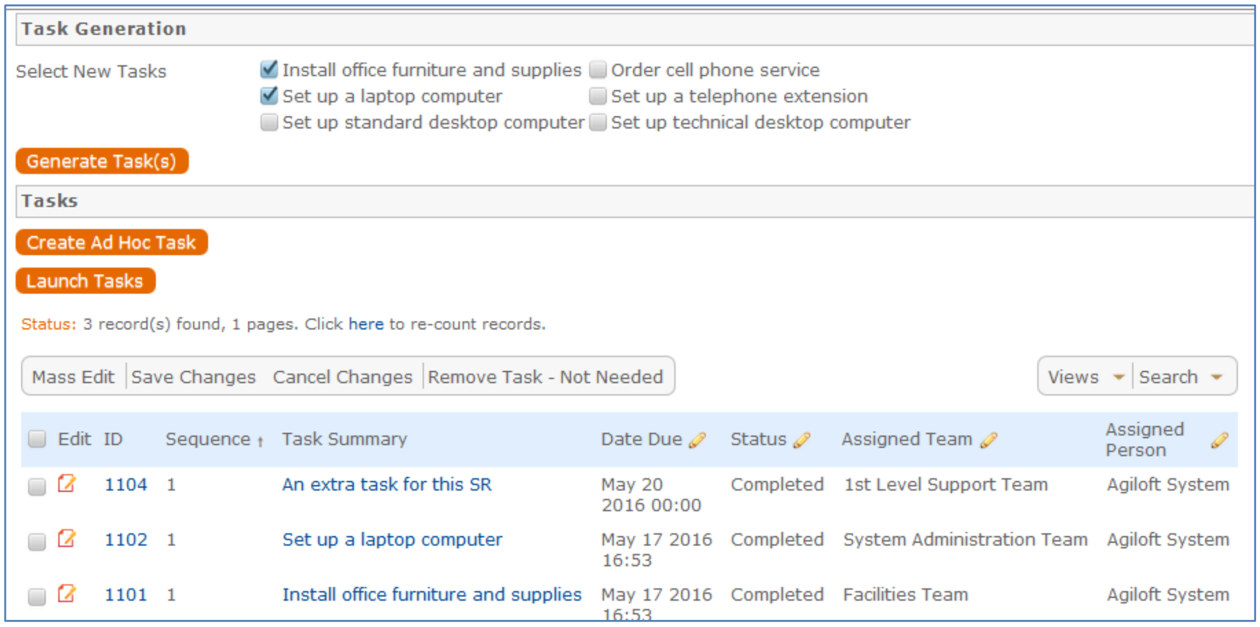
If a Support Staff technician creates a record in a status of Closed an email is sent to the customer telling them how to reopen the Service Request. This email is sent by a workflow action and is displayed as a checkbox that can be turned off by technician users. (This option is set in the workflow options, and can be modified).

If the status is not Closed when saved, the rule named “All New Service Request Actions” will send the customer an acknowledgement email and will also send an email to either the Assigned Person, if there is one, or to the Assigned Team.

Using Tasks for a Service Request

Some services are defined to have a set of tasks that should be completed. The task method is defined in the service record as described above in Tasks for Service Requests.

If tasks are associated with the service, they are manually generated by the technician responsible for the service request when ready, on the Tasks tab:


Task Generation

Select New Tasks

- [x] Install office furniture and supplies
- [x] Set up a laptop computer
- [ ] Set up standard desktop computer
- [ ] Order cell phone service
- [ ] Set up a telephone extension
- [ ] Set up technical desktop computer

Status: 3 record(s) found, 1 pages. Click here to re-count records.

Edit	ID	Sequence	Task Summary	Date Due	Status	Assigned Team	Assigned Person
	1104	1	An extra task for this SR	May 20 2016 00:00	Completed	1st Level Support Team	Agiloft System
	1102	1	Set up a laptop computer	May 17 2016 16:53	Completed	System Administration Team	Agiloft System
	1101	1	Install office furniture and supplies	May 17 2016 16:53	Completed	Facilities Team	Agiloft System

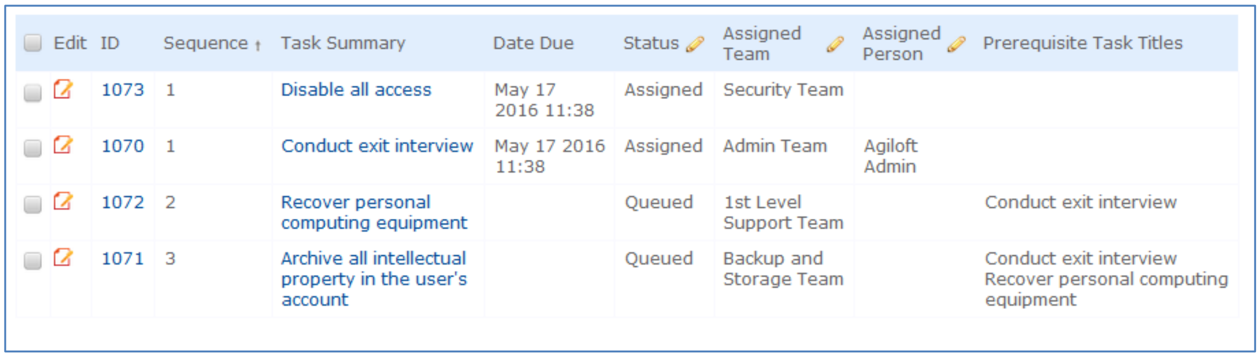
In the screenshot above, the user has generated two of the user-selected tasks by using the Generate Tasks button, and has also created an ad hoc task using the Create Ad Hoc Task button.

For the user-selected tasks, the Generate Tasks button remains visible in case the user decides to check additional boxes after generating the first tasks. Validation prevents the same tasks from being created a second time for the same service request.

Tasks are created with a status of Queued or Conditional (if the task is marked as a conditional task, that is, the task is only needed if some particular field condition is met, which is defined in the task template). A rule running on creation handles assignment.

Once the tasks have been generated, the Launch Tasks button becomes visible. When the Launch Tasks button is clicked, all tasks that have no prerequisite tasks, or whose prerequisites have been marked as Not Needed, are set to a Status of Assigned, and the assigned team or person is notified with an email. The Date Due is also set based on several criteria.

Below is an example of tasks for a service request that were generated from a task workflow in which two of the tasks have prerequisite tasks:

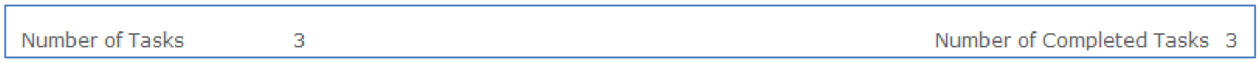


Edit	ID	Sequence ↑	Task Summary	Date Due	Status	Assigned Team	Assigned Person	Prerequisite Task Titles
	1073	1	Disable all access	May 17 2016 11:38	Assigned	Security Team		
	1070	1	Conduct exit interview	May 17 2016 11:38	Assigned	Admin Team	Agiloft Admin	
	1072	2	Recover personal computing equipment		Queued	1st Level Support Team		Conduct exit interview
	1071	3	Archive all intellectual property in the user's account		Queued	Backup and Storage Team		Conduct exit interview Recover personal computing equipment

The Sequence field is used to indicate the general sequence of tasks. If all the tasks have a Sequence value of 1, this indicates that they are likely all parallel tasks triggered at the same time. If a task has one or more prerequisite tasks, its sequence value is automatically set to the highest sequence value of its prerequisite tasks plus 1. So if a task has prerequisites whose Sequence values are 1 and 4, it will be set to 5. This numbering is not functional, it is just descriptive of the general order in which the tasks are expected to be done.

If there are tasks that have prerequisite tasks, those tasks remain in the Queued or Conditional status until all of their prerequisites are completed, at which point, any conditions are checked (for the conditional tasks) and the status is set to Assigned or Not Needed (if the condition for a conditional task is not met). In the example above, the last two tasks are Queued waiting for their preceding tasks to be completed. The top two tasks were assigned at the same time, since they had no prerequisites.

When tasks are used, two fields track the number of tasks and the number of completed tasks (those with a terminal status).



Number of Tasks	3	Number of Completed Tasks	3

When these two fields match, meaning that the final task has been completed, a rule notifies the assigned person or assigned team (if no assigned person) that the final task is done (Rule: Tasks Just Completed; Action: Notify Assigned team or person of completed tasks).

Processing of Records

If a technician working on a Service Request needs more information from the customer in order to take further action, she can set the status to Pending Customer. A workflow action will automatically send an email to the customer requesting further information and including the content of the Additional Notes field, an append-only field that is used to communicate with the customer. The email includes a hyperlink for the customer to click to edit the Service Request directly.

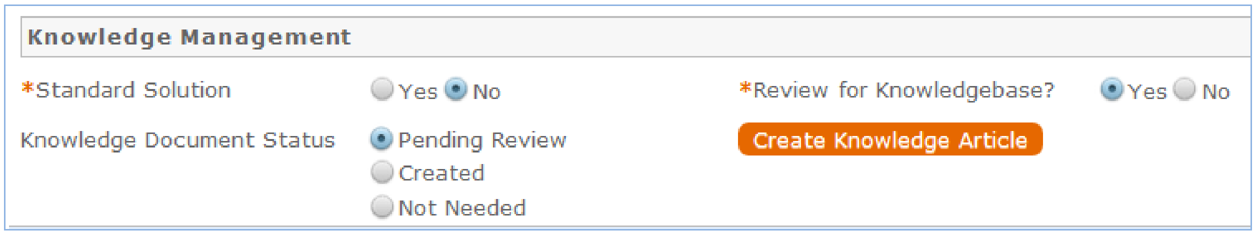
If the technician needs to reassign the Service Request to a different team or person, he or she simply changes the Assigned Team and/or Assigned Person field and the system will email the new assignee notifying them of the reassignment (Rule: Assigned team or person changed; Action: I: Notify team or person of new assignment).

The Staff Only Notes field is an append-only field that holds technician working notes that are not visible to the customer.

When the technician has completed work on the Service Request, they set the Status field to Closed and put the solution notes into the Solution field. This triggers a workflow email to the customer that includes the content of the Solution field and tells the customer that the request has been completed. By default, no escalation rules are set up for the Service Request table.

Reviewing for Knowledgebase and Standard Solution Inclusion

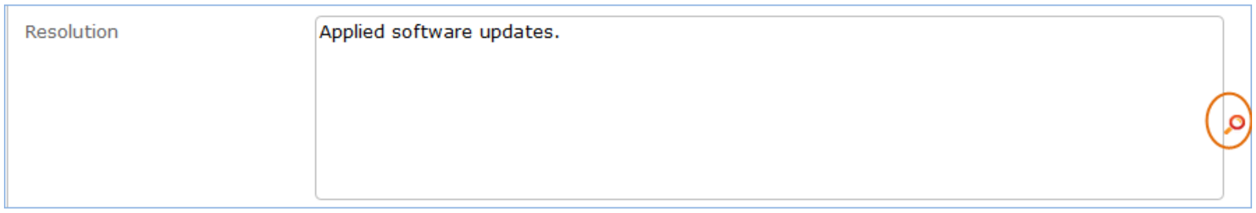
When the Status is changed to Closed, additional required fields become visible at the bottom of the Working tab:



Knowledge Management

*Standard Solution	Yes [✓ No]	*Review for Knowledgebase?	[✓ Yes] No
Knowledge Document Status	[✓ Pending Review] Created Not Needed	Create Knowledge Article	

The person closing the request is required to update the Standard Solution field and the Review for Knowledgebase fields. The Standard Solution defines a subset of requests that have a Resolution field value that may be reused in another request. Requests with a Yes in that field become available to the lookup next to the Resolution field:

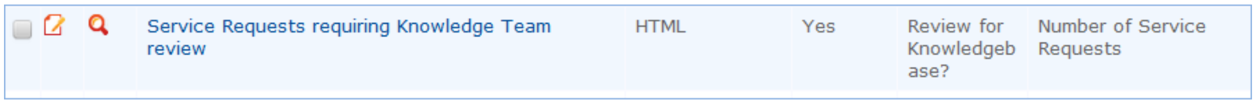


Resolution	Applied software updates.

That lookup searches for standard solutions and provides a button to import their Resolution field value into the request that is currently being edited. So it is helpful to mark useful responses in this way so they can be reused.

The Review for Knowledgebase? Field is used to create another subset of requests that should be considered for conversion into a Document FAQ. The conversion to a Knowledge Article may be done directly by the service technician using the Create Knowledge Article button shown above, or that button may be permission-controlled to be visible only to someone in a higher level group.

There is a built-in report finding all service requests with a Yes value in the Review for Knowledgebase field and a Pending Review value in the Knowledge Document Status field. This report can be scheduled and mailed to a team responsible for creating new Knowledge Articles in the Documents table.



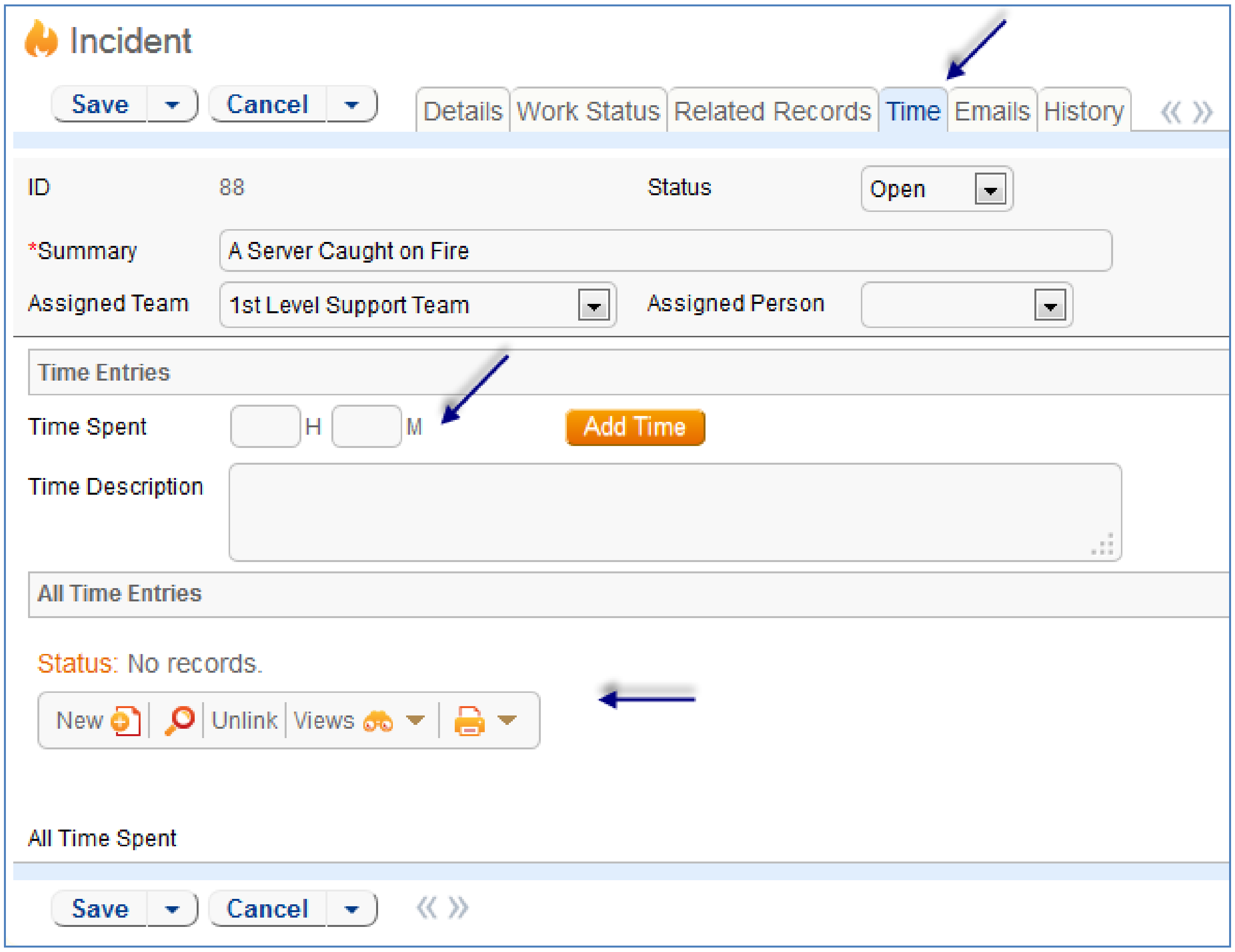
Service Requests requiring Knowledge Team review	HTML	Yes	Review for Knowledgebase?	Number of Service Requests

When the Create Knowledge Article button is pressed, it brings up a new document record and maps fields from the request into fields in the new record. Additional fields can be filled out as well. When the document record is saved, a rule running there comes back and updates the service request to set the Knowledge Document Status to a value of Created. That hides the create button so it will not be converted again.

If upon reviewing the request, the reviewer determines it should not be converted, she can change that field to Not Needed so that the request will stop appearing on the reports of items to review.

Reporting Time Spent

Technicians may easily report the time they spend on handling service requests. There are two fields: Time Spent and Time Description, and an action button: Add Time, on the Working Notes tab of the layout. Entering values there will automatically create a new time entry record when the Add Time button is clicked. The time entry will show the work done by the technician and on the current date.


Incident

Save | Cancel

Details | Work Status | Related Records | Time | Emails | History

ID: 88
Status: Open
*Summary: A Server Caught on Fire
Assigned Team: 1st Level Support Team
Assigned Person:

Time Entries
Time Spent: H M
Add Time
Time Description:

All Time Entries
Status: No records.
New | Unlink | Views

All Time Spent

Save | Cancel

All the time entries for a request can be seen on the Time tab as well. If a technician needs to report time that was spent on a different day or by a different user, he may click the New button on the Time spent table to submit a time entry directly and change the date or “Done by” field. All time entered is totaled in the All Time Spent field, which can be used in reporting or billing.

The Add Time action button converts the time fields into a Time Entry record and then blanks out the current values so they will be empty.

Note that this same time entry methodology is used in the Incident, Problem, Change Request, and Task tables. So time spent on each type of request will be held in a single table (Time Entries) from which reports may easily be run for all kinds of time.

Customer Updates

If the customer updates the Service Request at any point, an email notifies the assigned person or assigned team (if no assigned person) of the update.

When the customer replies to the email or edits a Service Request in a status of Pending Customer, the status changes to Updated by Customer (Rule: all customer update actions, Action: User Update Actions) and an email notifies the assigned person or the assigned team (if assigned person is empty) that the customer has replied. The customer is able to update the Additional Notes field directly and email replies are mapped to that same field.

The closing email to customers includes a hotlink back to the record if they wish to reopen it and instructs them to explain why they are not satisfied with the solution. Clicking the hotlink will automatically change the "I Would Like To Reopen My Service Request" field to Yes, which in turn sets the Status of the Service Request to Reopened and notifies the assigned person (Rule: All customer update actions, Action: User Update Actions).

Workflow

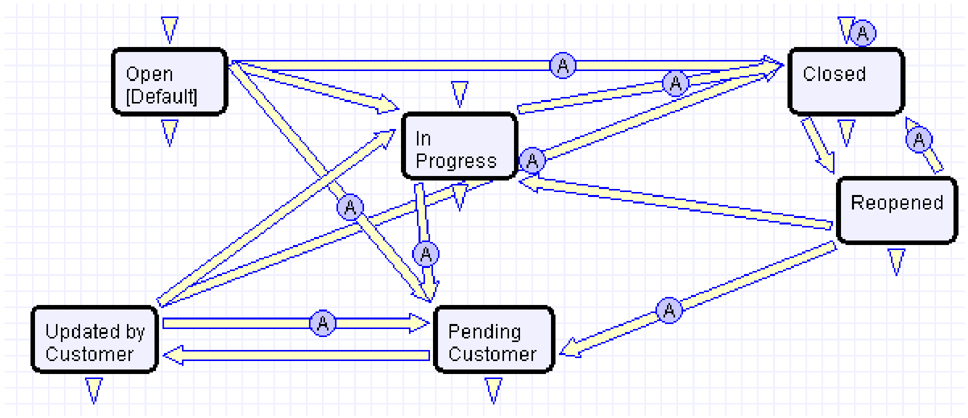


Figure 6. Service Request table default Workflow

Ownership

Records in this table are "owned" by the individual customer. This means each record is associated with a particular login and no other "end user" employee will be able to edit that record. All technician users are able to edit any service request by default.

Reporting and Statistics

Several fields are included that are used for statistical reporting. The following fields may be of interest:

Number of Assignees – is auto-incremented each time the assigned person changes (Rule: Assigned team or person changed, Action: Notify team or person of new assignment)

Number of Teams Assigned – is incremented each time a different team is put in the Assigned Team field (Rule: Assigned team or person changed, Action: Notify team or person of new assignment)

Number of Reopens – is incremented each time a closed request is reopened by a customer

Total Hours to Close – elapsed time between Date Created and Date Closed (default value)

Working Hours to Close – elapsed time between Date Created and Date Closed minus the non-working hours of the team in the Assigned Team field and the time during which the Status was Pending Customer (default value)

Solved Within SLA – yes/no field with default value of Yes, is set to No when the total hours to close is greater than the relevant SLA Hours to complete field (pulled in with the service and based on priority) (Rule: Status Change Actions, Action: All status change actions)

There are default reports measuring Total Time Spent by Service Category, Average time to close by service category and by person who closed, as well as averages for number of people assigned, number of teams assigned, number of reopens, and so on. With the field structures already there, it is easy to add reports to slice and dice the information the way you need it.

Incidents Table

Overview

Incidents are used to report service problems and outages. They are interruptions to customer service. The objective is to return to the normal service as defined in an SLA as quickly as possible with minimal business impact. An example Incident for an end user might be: “I cannot access my email”.

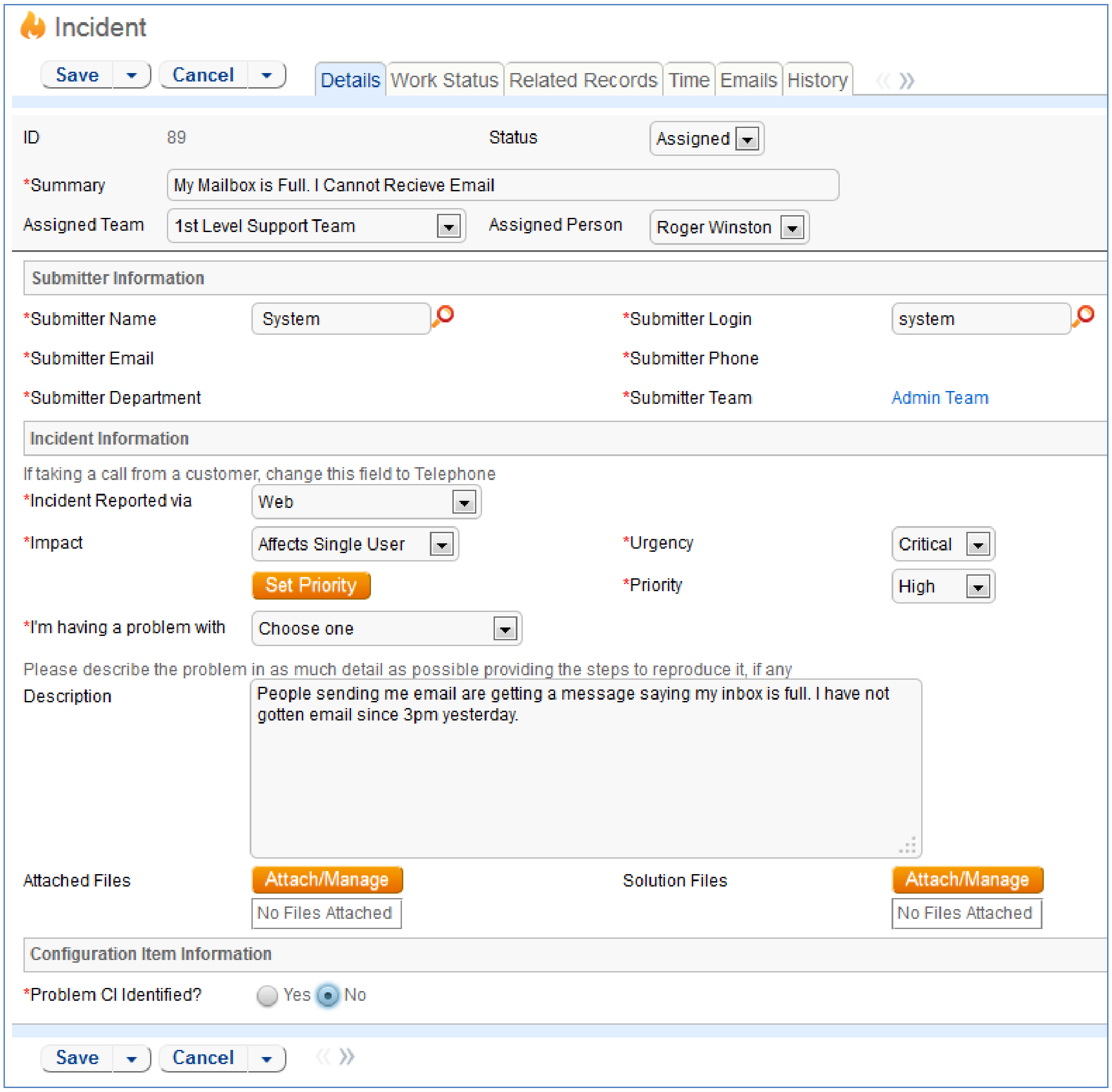
What is the distinction between Service Request and Incident? Incidents are interruptions or degradations of an existing service that need to be rectified. They are problems that need to be solved, rather than a request for new service or a change to a functioning service, such as a new employee setup or new software application request.

Incidents may result from underlying problems caused by misconfiguration or hardware failures, and an underlying problem may result in the submission of several incidents by end users. When this is the case, the underlying problem may be reported and managed in a separate Problem record.

When an incident is caused by an underlying problem, the technician working on the incident can link the incident to an existing Problem record or convert it to create a new Problem record. They may also click the Save and Create Change Request if they wish. The Change Request table will keep track of Incidents that are converted onto it.

Problems may require significant changes in order to be permanently resolved. In this case, the technician working on the Problem will link it to a new or existing Change Request. While a Problem may stay open until the Change is implemented, often a workaround is found to resolve the interruption in service for the customer.

The system is set up to make it easy to generate and relate Incidents, Problems, and Changes, and to quickly and automatically propagate workarounds from problems into all related Incidents.


Incident

Save | Cancel

Details | Work Status | Related Records | Time | Emails | History

ID	89	Status	Assigned
*Summary	My Mailbox is Full. I Cannot Recieve Email		
Assigned Team	1st Level Support Team	Assigned Person	Roger Winston

Submitter Information

*Submitter Name	System	*Submitter Login	system
*Submitter Email		*Submitter Phone	
*Submitter Department		*Submitter Team	Admin Team

Incident Information

If taking a call from a customer, change this field to Telephone

*Incident Reported via	Web		
*Impact	Affects Single User	*Urgency	Critical
	Set Priority	*Priority	High
*I'm having a problem with	Choose one		

Please describe the problem in as much detail as possible providing the steps to reproduce it, if any

Description: People sending me email are getting a message saying my inbox is full. I have not gotten email since 3pm yesterday.

Attached Files	Attach/Manage No Files Attached	Solution Files	Attach/Manage No Files Attached

Configuration Item Information

*Problem CI Identified? Yes [✓ No]

Save | Cancel

Use Case

Incidents may be created by internal customers through the web portal or via email, or by a technician taking a telephone call from a customer. They may also be created by network monitoring systems configured to send problem reports to the system through one of the standard APIs. A field in the Incident form specifies the reporting source (email, web, phone, etc.). A technician may also convert a Service Request to a new Incident via an action button in the SR form if a customer submitted a Service Request when he should really have submitted an Incident. See Details of Incident Rules in the Appendix for more details.

End User Record Submission

When a customer submits an Incident, the contact fields automatically populate based on the details in his/her User record. In the default setup, an Incident cannot be submitted on behalf of a user who has no user record in the system. This may easily be changed by modifying the linked fields from the User table to enable “non-source” values to be used.

The user selects a category for the Incident from the drop down list "I'm having a problem with...", such as 'Email' or 'Access Problem', then describes his/her problem in more detail via the Summary and Description fields.

The user is also given two fields to describe the priority of the Incident: Impact and Urgency. Impact specifies how widespread the Incident is: A user whose desktop computer is malfunctioning would choose 'Affects Single User', whereas a critical server failure might be classified as 'Affects Company'. Urgency is a subjective measure of the criticality and time sensitivity of the Incident, and ranges from Low to Critical. Both Impact and Urgency are used to automatically determine the official Priority of the Incident, a field which is hidden from the end user and only editable by staff.   See Incident Rules for the current logic used to set the Priority. (Priority is only set by a rule if the staff user has not already manually set it).

After providing the summary, description, category, impact, and urgency, the user clicks Finish and receives an automatic email notification when the system creates the record.

Technician Record Submission

A technician creating an Incident sees additional information relating to the status of the Incident, including assignee and status fields. Unlike end-users, staff can see an additional section on the details tab for Configuration Item Information that shows fields relating to a specific Configuration Item. By default, Incidents are created with the "Problem CI Identified?" field set to No, but if a staff person determines the CI that is the root cause of the Incident, he or she may set this field to Yes and select a Configuration Item via a linked field set.

By default, Incidents without a CI identified are assigned to the 1st Level Support Team. Technicians can also set the assigned team and assigned person manually via drop-down lists in the Work Status tab. If the Problem CI has been identified, the staffer can set the field "Assign to CI Responsible Team" to Yes and when the Incident is saved, a rule will detect the change and assign the team responsible for the CI selected to the Incident. A technician may change the Assigned Team field later, as the rule will only set the Assigned Team at the time the CI field changes to Yes.

New Incidents are created in a status of Open by default. During creation, a technician may change the status to Assigned (if he wishes to indicate work has already started on it) or Closed (if he was able to resolve the issue over the phone and wants to capture the request for reporting purposes).

Automatic Emails Sent upon Submission

If a Support Staff technician creates a record in a status of Closed an email is sent to the customer telling him/her how to reopen the Incident. This email is sent by a workflow action and is displayed as a checkbox that can be turned off by technician users. (This option is set in the workflow options).

If the status is not Closed when saved, the rule named “Incident - All Creation Actions” will send the customer an acknowledgement email and will also send an email to either the Assigned Person, if there is one, or to the Assigned Team if not.

Processing of Records

When a technician works on an Incident, if she needs more information from the customer in order to take further action, she can set the status to Pending Customer. A workflow action will automatically send an email to the submitter requesting further information and including the content of the Additional Information field, an append-only field that is used to communicate with the submitter. The email includes a hyperlink for the customer to click to edit the Incident directly.

If the technician needs to reassign the Incident to a different team or person, he or she simply changes the Assigned Team and/or Assigned Person field and the system will email the new assignee notifying them of the reassignment (Rule: Incident Edit Actions, Action: I: Notification Actions).

The Staff Only Notes field is an append-only field that holds technician working notes that are not visible to the customer.

When the technician has completed work on the Incident, he/she sets the Status field to Closed and puts the solution notes into the Resolution field. This triggers a workflow email to the customer that includes the content of the Resolution fields and tells the customer that the request has been completed.

Relating Incidents to Problems

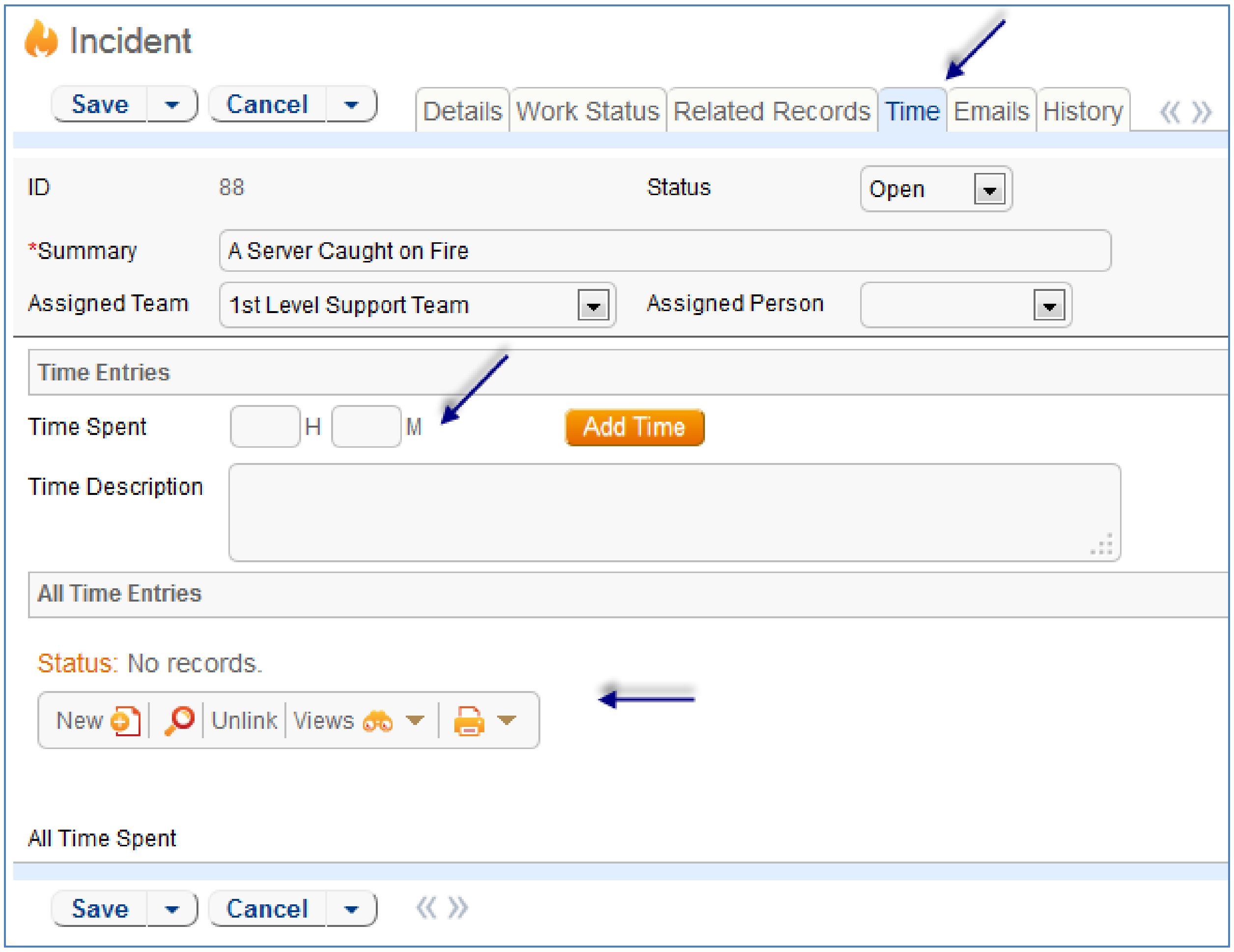
If an incident has an underlying problem that needs to be addressed, the technician may quickly convert the Incident into a related Problem record. Or she may use the Linked Problem lookup to search for existing Problems that may be the cause of this incident, and if one is found, link the Incident to that Problem.

When an Incident is linked to a new or existing Problem, the Incident may be automatically updated from the Problem record when a workaround or solution to the Problem is identified.

Within the Problem record, a button may be pressed to copy the workaround for the problem into all related Incidents, set the Status of the Incident to Workaround Provided, and email the customer and assigned rep. Another button may be pressed to populate the Problem’s Solution into the related Incidents, set their Status to Closed, and email the customers.

Reporting Time Spent

Technician users may easily report the time they spend on handling Incidents. There are two fields and an action button for this purpose (Time Spent, Time Description, and Add Time) on the Working Notes tab of the layout. Entering values there will automatically create a new time entry record when the Add Time button is clicked. The time entry will show the work done by the technician and the current date.


Incident

Save | Cancel

Details | Work Status | Related Records | Time | Emails | History

ID	88	Status	Open
*Summary	A Server Caught on Fire		
Assigned Team	1st Level Support Team	Assigned Person	

Time Entries

Time Spent H M Add Time

Time Description

All Time Entries

Status: No records.

New | Unlink | Views

All Time Spent

Save | Cancel

All the time entries for a request can be seen on the Time tab as well. If a technician needs to report time that was spent on a different day or by a different user, he may click the New button on the Time spent table to submit a time entry directly and change the date or "Done by" field. All time entered is totaled in the All Time Spent field, which can be used in reporting or billing.

Note that this same time entry methodology is used in the Service Request, Problem, Change Request, and Task tables.

Customer Updates

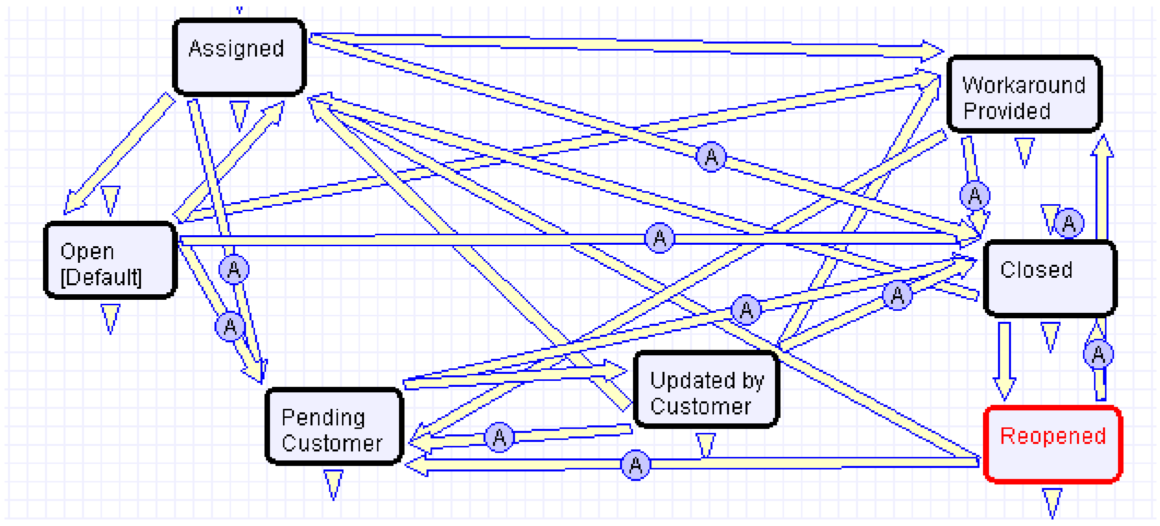
If the customer updates the Incident at any point, an email notifies the assigned person or assigned team (if no assigned person) of the update.

When the customer replies to the email or edits an Incident in a status of Pending Customer, the status changes to Updated by Customer (Rule: Incident Customer Update Actions, Action: Submitter Update Actions) and an email notifies the assigned person or the assigned team (if assigned person is empty) that the customer has replied. The customer is able to update the Additional Information field directly and any email reply from the customer is mapped to that same field.

The closing email to customers includes a hotlink back to the record if they wish to reopen it and instructs them to explain why they are not satisfied with the solution. Clicking the hotlink will automatically change the "I Would Like To Reopen My Incident" field to Yes, which in turn sets the

Status of the Incident to Reopened and notifies the assigned person (Rule: Incident Customer Update Actions, Action: Submitter Update Actions).

Workflow



Incident table default Workflow

Ownership

Records in the Incident table are "owned" by the individual submitter. This means each record is associated with a particular employee login and no other "end user" employee will be able to edit that record. All technician users are able to edit any Incident by default.

Reporting and Statistics

Several fields are included that are used for statistical reporting. The following fields may be of interest:

Number of Assignees – is auto-incremented each time the assigned person changes (Rule: Incident Edit Actions, Action: Notification Actions)

Number of Teams Assigned – is incremented each time a different team is put in the Assigned Team field (Rule: Incident Edit Actions, Action: Notification Actions)

Number of Reopens – is incremented each time a closed request is reopened by a customer

Total Hours to Close – elapsed time between Date Created and Date Closed

Working Hours to Close – elapsed time between Date Created and Date Closed minus the non-working hours of the team in the Assigned Team field and the time during which the Status was Pending Customer

There are default reports measuring Total Time Spent by Type of Problem, Average time to close by Type of Problem and by person who closed, as well as averages for number of people assigned, number of teams assigned, number of reopens, and so on. With the field structures already there, it is easy to add reports to slice and dice the information the way you need it.

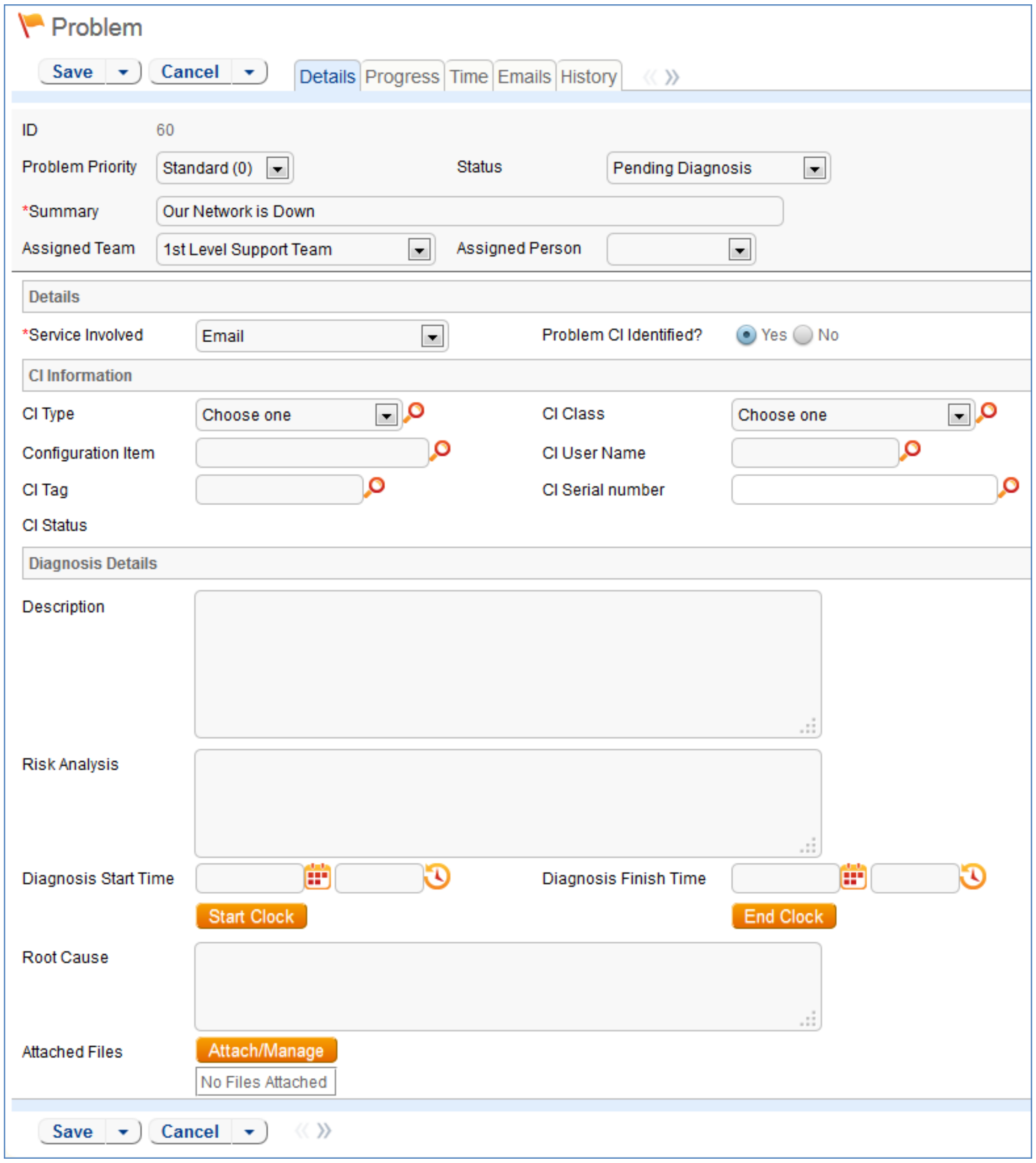
Problems Table

Overview

Problems represent the root cause of one or more Incidents or possible Incidents. Resolving a problem means resolving/preventing related Incidents.

Incident Management is concerned with restoring service as quickly as possible, whereas Problem Management is concerned with determining and eliminating root causes (and hence eliminating repeat problems).

The primary objectives of Problem Management are to prevent Incidents from happening and to minimize the impact of Incidents that cannot be prevented.



ID 60

Problem Priority: Standard (0)

Status: Pending Diagnosis

*Summary: Our Network is Down

Assigned Team: 1st Level Support Team

Assigned Person:

Details

*Service Involved: Email

Problem CI Identified? [✓ Yes] No

CI Information

CI Type: Choose one

CI Class: Choose one

Configuration Item:

CI User Name:

CI Tag:

CI Serial number:

CI Status

Diagnosis Details

Description

Risk Analysis

Diagnosis Start Time

Diagnosis Finish Time

Root Cause

Attached Files

No Files Attached

Use case

Problem Creation

When an incident is determined to be based on an underlying Problem, first-response support technicians create a Problem record from the Incident, automatically creating a link between the Incident record and the Problem. The new problem record imports relevant information from the Incident, such as the linked Configuration Item. Problems are also creatable independent of existing Incidents, such as in cases where a problem is discovered internally but no Incidents have been reported.

Resolution of problems may require Changes to the system. Staff addressing the problem may determine that a shared CI needs to be replaced or modified, and may therefore file a Change Request.

When the problem is resolved, a technician updates the record with relevant information and closes the record, in turn prompting automation to begin closing procedures for related Incidents. If the problem cannot be resolved it may be classified as a Known Error and a permanent work-around supplied. This will also update the related Incidents.

Processing of Records

Once created, Problem records can be linked to Incidents from either the Problem record or from an Incident record.

Priority, a measure of the Problem's urgency and relative importance, is set by default to the Priority of the spawning Incident, but can be changed at the time of creation. Problem Priority, a measure of impact and risk, particularly with regards to IT service operations, may be determined separately as part of Problem Classification procedures. Together Priority and Problem Priority help IT managers make factual decisions for scheduling and planning and help determine the category of related Change Requests. By default, new Problems have a Problem Priority of Standard (0), indicating no special review or authorization is needed.

Diagnosis

The Problem follows its own workflow separate from the Incident. Ops team members open the problem record in the default state of "Pending Diagnosis" to indicate that a diagnosis has yet to be determined, or "Diagnosed" if no further steps are required. A record may sit in a state of Pending Diagnosis for some time before staff actually begin to perform the diagnosis, so Start Clock and Stop Clock buttons let users indicate how much time was spent diagnosing a particular problem.

As part of the diagnosis, technicians select the service involved based on existing Incident reports or based on the most likely service to be affected by the Problem. If a particular Configuration Item is identified as the source of the problem's root cause, staff can quickly link the Problem to the CI record.

Once a diagnosis is supplied, staff will move the record into a state of "Diagnosed" and fill out the Root Cause description. They may suggest a temporary fix, or "workaround", for the problem and related incidents (e.g. "use Printer B3 for now instead") if a permanent solution is not readily available. While determining a workaround and/or permanent resolution, technicians can use Start and Stop Clock buttons to track the time spent determining workarounds and solutions for this Problem.

If at any time during the root cause diagnosis or determination of the proper solution staff need more information from a separate process, such as Incident details from first level support, the Problem record may be placed in a state of "Pending More Information".

If a Problem is deemed too risky or of lower priority than more imminent issues, it may be put in a status of "Deferred" to reflect no ongoing diagnosis or pending changes.

Solution

In most cases where a Problem's root cause deals with a Configuration Item, a Change Request will be submitted to make the appropriate fixes to the Configuration Item. Change Requests are creatable directly from the Problem record, instantly linking them. While a problem is waiting for a Change Request it can be put in the status of "Pending Change".

If the Change Request linked to a Problem is closed, the system will send an email notification to the problem assignee so that the individual can take additional steps to close the Problem record if it was pending change for resolution.

A problem whose root cause is known but for which there is no permanent resolution is considered a Known Error. Known Errors should have Workarounds to allow Incident Management to restore service as quickly as possible. The "Update Incident with Workaround" button allows staff working the Problem to quickly disseminate workaround information to linked Incidents with the click of a button. Clicking the button will post the workaround in all related incidents and change their status to Workaround Provided, which will also trigger an email to both the end user and the assigned staff person of the Incident(s).

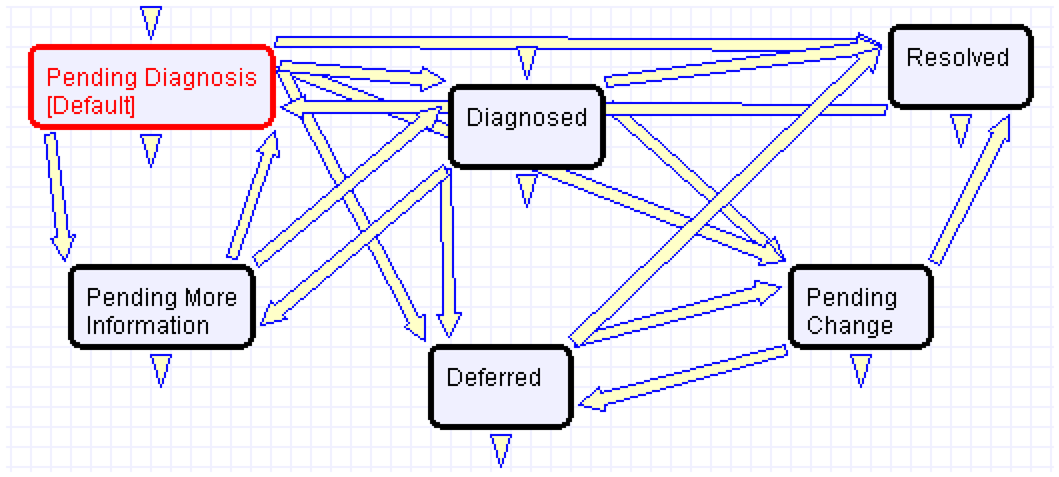
A similar button is used to transfer the problem Solution to related Incidents. Clicking the “Update Incidents with Solution” button in the problem will populate the Solution field of the Problem into the Incident Solution field and set the status of the Incident to Closed, emailing the customer.

Once a permanent resolution is determined and implemented, staff users enter the description in the Resolution field and set the status to Resolved. If the resolution contains information that is useful outside of this problem's particular scope, the “Add to Knowledgebase?” field can be set to Yes to make the Resolution field available via FAQs.

Ownership

Problems are "owned" by the staff member who creates the Problem record. Since only internal staff will see Problem records, groups may share responsibilities between Incident Management and Problem Management and multiple individuals may share ownership over time.

Workflow



Change Requests (RFC) Table

Overview

The Change Requests table is used for Change Management. A Change request is created when a change is needed to a configuration item or to any other business object that may require a set of approvals before such a change can be completed.

Once a change request is created, it can be assigned to the appropriate teams or individuals for one or many approvals, and may then be moved along in the process, from approval to scheduling, to action by staff members actually making the change, to completion, testing, and release.

Of course each organization has its own definition of types of change that may need to be managed in this way, but we have prepopulated the service catalog with a list of common change types. You can easily add your own change categories and types to the services table.

Change Requests differ from Service Requests in that they are not visible by default to end users and typically follow a stricter set of procedures and approvals.

Use Case

Change Requests (CRs) may be directly created by technician users or may be created by conversion from a Service Request or a Problem, in which case they are linked back to the generating record.

End User Record Submission

By default, Change Requests are not visible to end-users and cannot be created by them. This is defined by group permissions, and if you would like to allow end users to submit change requests, you can simply change the group permissions of the relevant groups to enable this.

Technician Record Submission

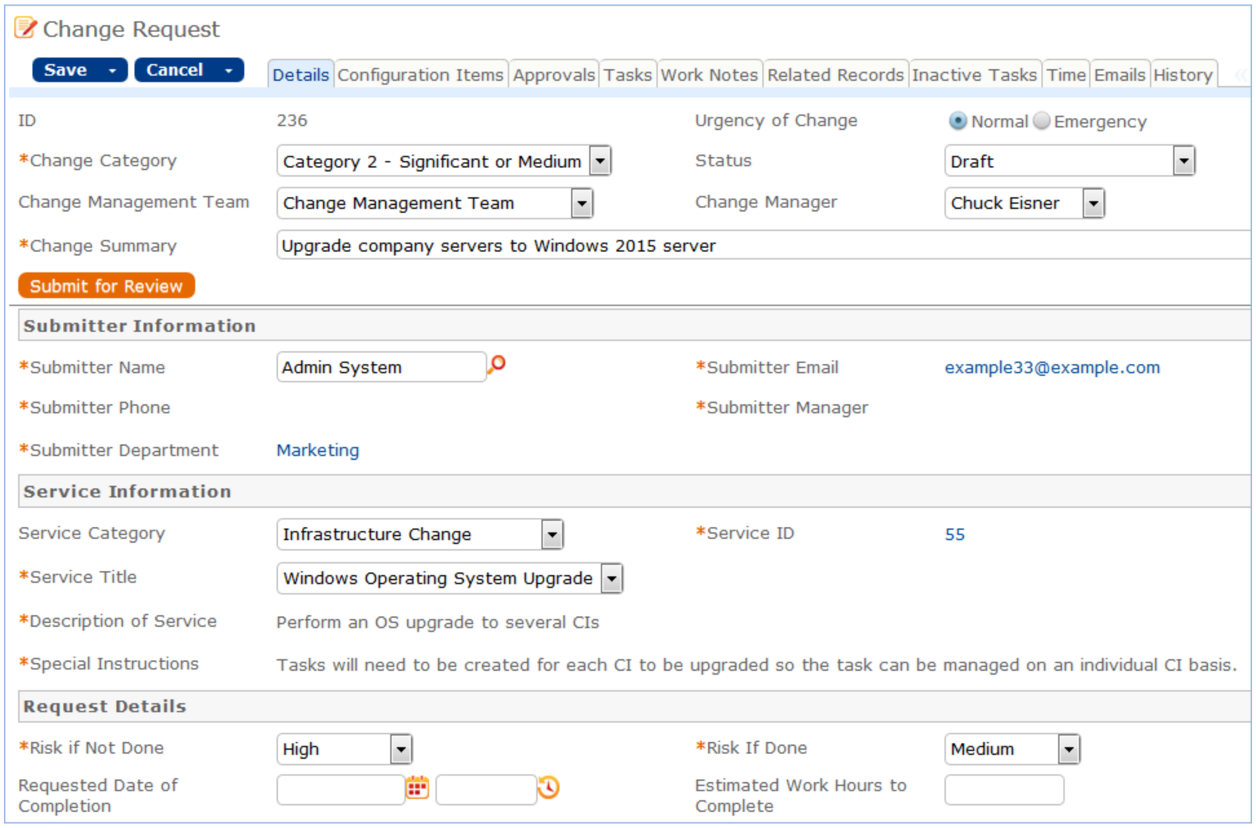
When a technician submits a Change Request, the Submitter contact information fields automatically populate based on the details in his/her User record. In the default setup, a request cannot be submitted on behalf of a user who has no user record in the system. However, the creator can select a different person if needed.

The user is required to select a service after selecting a service category. When the service is selected, the user will see a description of the service, and any special instructions for that service.

When a technician creates the CR he may choose the Change Management Team and the Change Manager from the drop-down fields. If he does not choose the team, when the record is saved, the request will be assigned to the Service Responsible team, if any, or to the Change Management Team, if there is no responsible team defined for the service.

New CRs are created in the Status of Open by default. During creation, a technician may click the Submit for Review button to change the status to Pending Change Manager.

The Change Request table pulls its services from linked entries in the Services table. A user creating a change request chooses the category of service from a drop-down, and then sees all the active services that belong to that category in a second drop-down list called Service Title. The default service categories for change requests can be seen in the Configuration Items table (or by creating a CR record and opening the drop-down list). When the service is selected, several additional fields are pulled in with the service, including fields defining whether approvals are needed and the approval generation type.



ID	236	Urgency of Change	[● Normal] Emergency
*Change Category	Category 2 - Significant or Medium	Status	Draft
Change Management Team	Change Management Team	Change Manager	Chuck Eisner
*Change Summary	Upgrade company servers to Windows 2015 server		

Submitter Information

*Submitter Name	Admin System	*Submitter Email	example33@example.com
*Submitter Phone		*Submitter Manager	
*Submitter Department	Marketing		

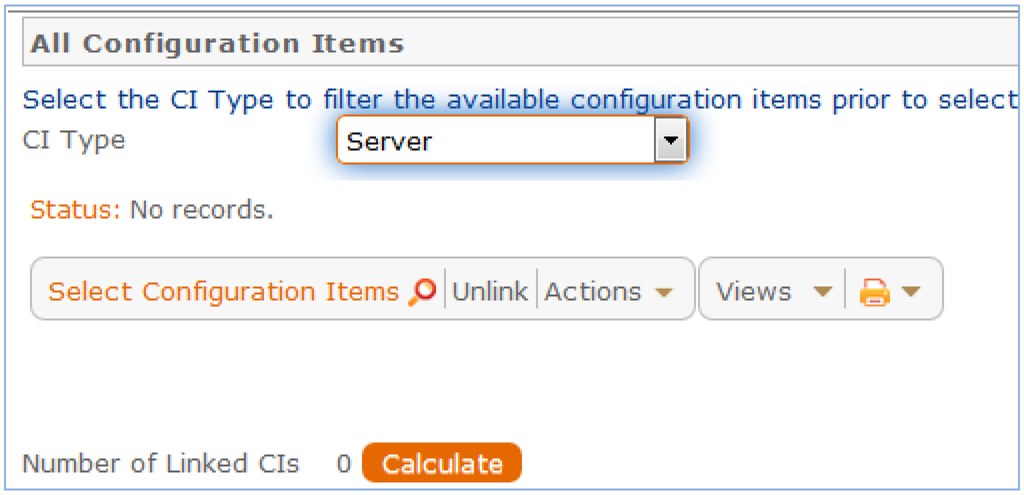
Service Information

Service Category	Infrastructure Change	*Service ID	55
*Service Title	Windows Operating System Upgrade		
*Description of Service	Perform an OS upgrade to several CIs		
*Special Instructions	Tasks will need to be created for each CI to be upgraded so the task can be managed on an individual CI basis.		

Request Details

*Risk if Not Done	High	*Risk If Done	Medium
Requested Date of Completion		Estimated Work Hours to Complete	

Once the main fields are filled in, the user navigates to the Configuration Items tab. This allows the user to look up several configuration items. The CI's can be filtered by choosing a CI Type first and then clicking the Select Configuration Items lookup.



All Configuration Items

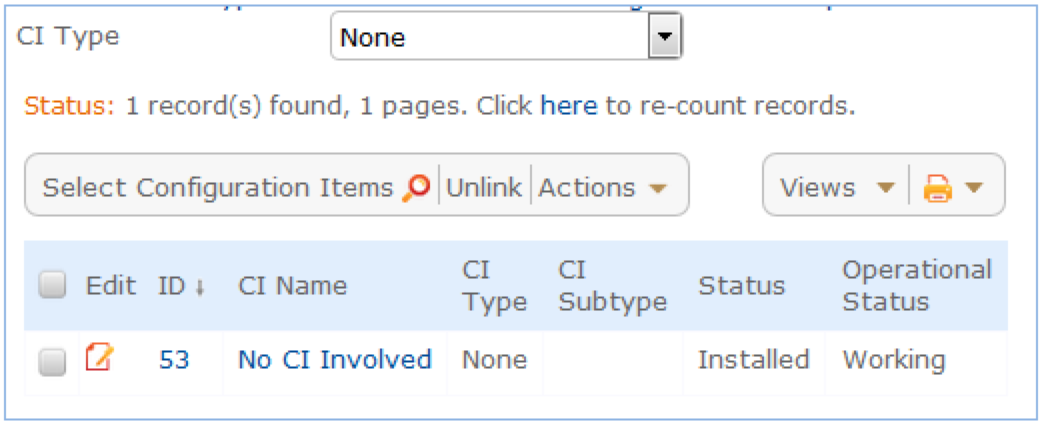
Select the CI Type to filter the available configuration items prior to select

CI Type: Server

Status: No records.

Number of Linked CIs 0

Note that if no configuration items are involved in the change request, you are still required to select one, because of the way tasks are managed in change requests. We have created a "No CI Involved" configuration item record with the CI Type of None that can be used in this situation. This will allow all task generation and other functionality to work as expected.

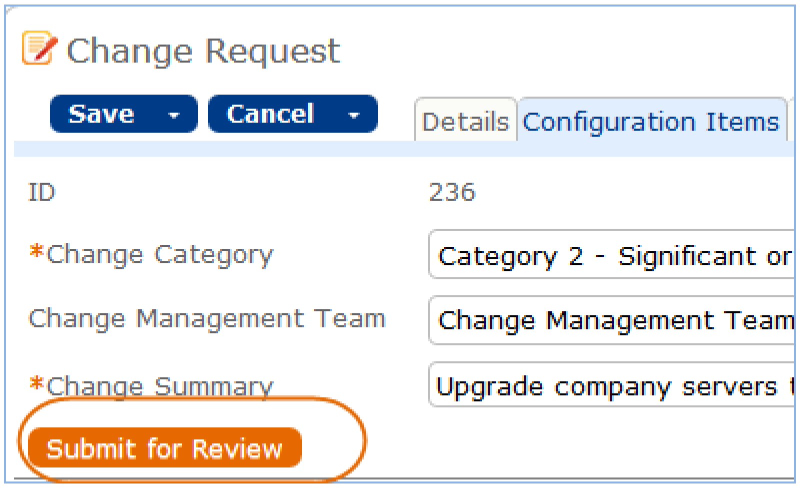


CI Type: None

Status: 1 record(s) found, 1 pages. Click here to re-count records.

Edit	ID	CI Name	CI Type	CI Subtype	Status	Operational Status
	53	No CI Involved	None		Installed	Working

Once the configuration items have been selected, to further process the change request, the user needs to click the Submit for Review button:



ID 236

*Change Category: Category 2 - Significant or

Change Management Team: Change Management Team

*Change Summary: Upgrade company servers t

This saves the change request (validating the required fields), sets some background fields that control visibility of the approval and task buttons, and sets the status to Pending Change Manager.

Automatic Emails Sent upon Submission

If the status is not Closed when saved, the rule named "CR- All Creation Actions" will send the submitter an acknowledgement email and will also send an email to either the Change Manager, if there is one, or to the Change Manager Team.

Changing Assigned Team or Change Manager

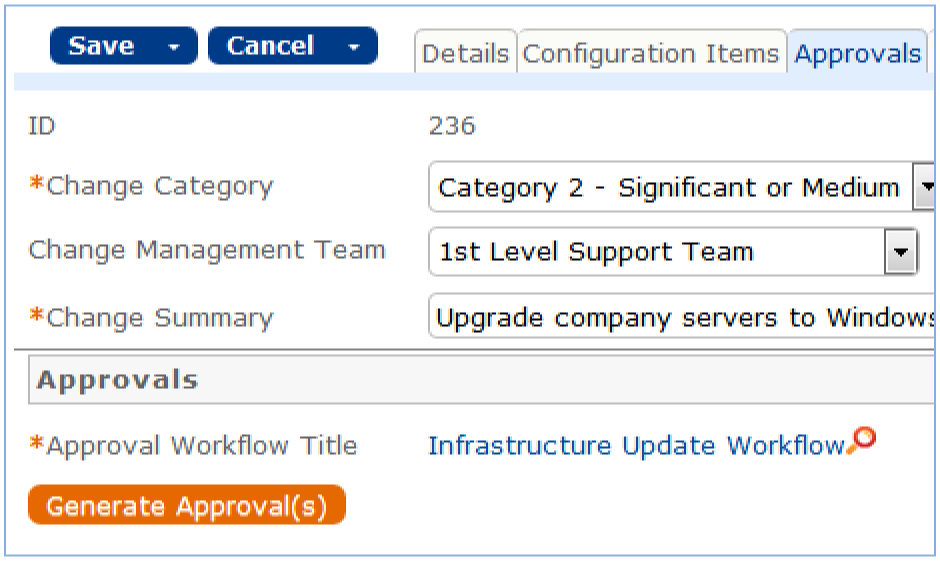
If a technician needs to reassign the Change Request to a different Change Management Team or Change Manager, he or she simply changes the field(s) and the system will email the new assignee notifying them of the reassignment (Rule: CR - Edit Actions (API Enabled), Action: Assignee Change Notifications).

Managing the Approval Process

Change requests can use different approval methods, discussed in the Approvals for Change Requests section above. Based on the approval method defined in the service record, approvals will typically be created and launched within the change request using buttons.

If the change service does include approvals, then at least one approval record must be created and approved in order to proceed to generating and launching any tasks and moving the status forward.

Once the request has been sent for review, if it was for a service that requires approvals, the change manager can review it and generate any needed approvals:

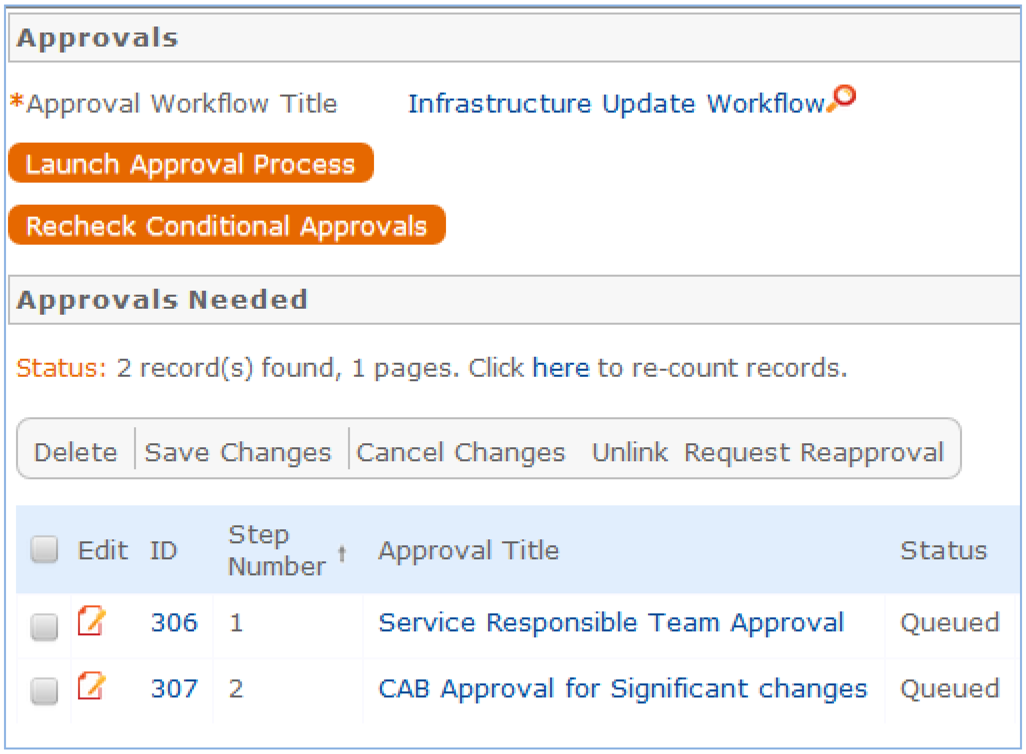


ID	236
*Change Category	Category 2 - Significant or Medium
Change Management Team	1st Level Support Team
*Change Summary	Upgrade company servers to Windows

Approvals

*Approval Workflow Title	Infrastructure Update Workflow

Clicking the Generate Approval(s) button will generate the approvals based on the predefined approval workflow. If the change request service allows ad hoc approvals, options will be available to generate additional approvals as needed. Approvals are generated in a Status of Queued by default. Once the approvals are generated, new buttons appear:


Approvals

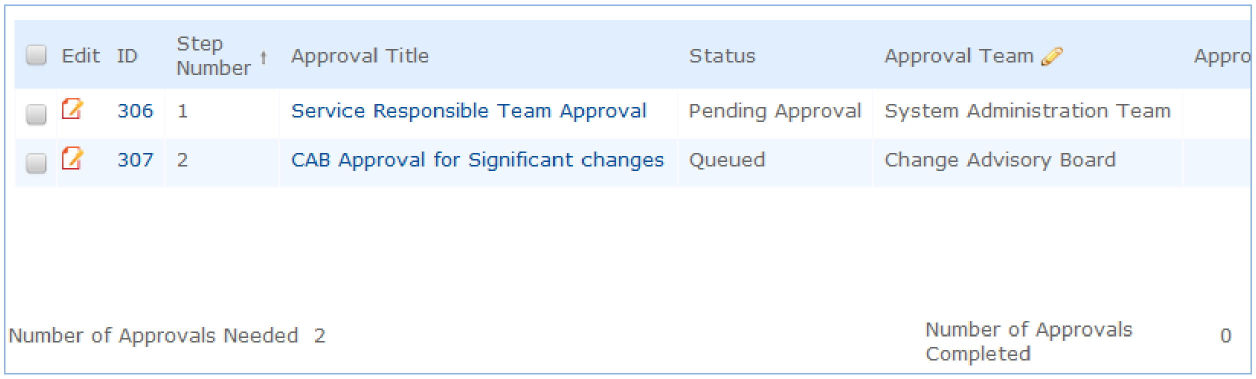
*Approval Workflow Title	Infrastructure Update Workflow

Approvals Needed

Status: 2 record(s) found, 1 pages. Click here to re-count records.

	Edit	ID	Step Number ↑	Approval Title	Status
☐		306	1	Service Responsible Team Approval	Queued
☐		307	2	CAB Approval for Significant changes	Queued

When the change manager is ready for the approval process to start, she clicks the Launch Approval Process button, which will set all approvals with the lowest step number to Pending Approval. It will also update the Status of the Change Request to Pending Approval:



	Edit	ID	Step Number	Approval Title	Status	Approval Team	Appro
		306	1	Service Responsible Team Approval	Pending Approval	System Administration Team	
		307	2	CAB Approval for Significant changes	Queued	Change Advisory Board	

Number of Approvals Needed 2

Number of Approvals Completed 0

When an approval's status is set to Pending Approval, the approval team and/or Approver are notified by email.

The approval process is controlled by the step numbers. Approvals with the same step number value are launched at the same time, when the approvals with the next lower step number are completed.

This is a different model than in the task workflows, in which the sequence field is not functional. With approvals, the step number controls the process and there are no separate sequences or threads in the approval process.

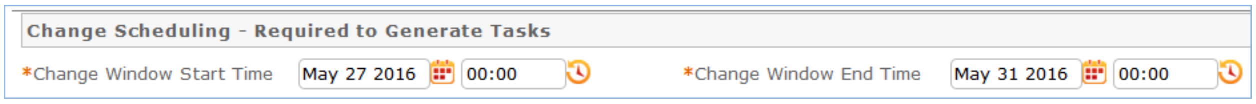
Another way in which approval generation is different from the handling of tasks is that conditional approvals are not created at all unless their condition is met at the time of generating the approval. Since these conditions generally depend on the meta data and fields in the change request, it is important to make any critical fields required to hold a value before getting to the point of generating approvals. The Recheck Conditional Approvals button does allow the technician to recheck the condition if after generating the first approvals, the meta data changes. But it is best to ensure that any needed meta data is filled out first.

As the first approvals are completed, those with the next step number are set to Pending Approval and their assignees notified. Any approval notes put into the approvals are copied up to the change request in the running All Approval Notes field.

Two fields below the approvals table track the Number of Approvals Needed and the Number of Approvals Completed, and once those two numbers are equal and great than or equal to 1, the Status is changed to Approved and the Requester and the Change Management team are notified.

Creating Tasks for the CR

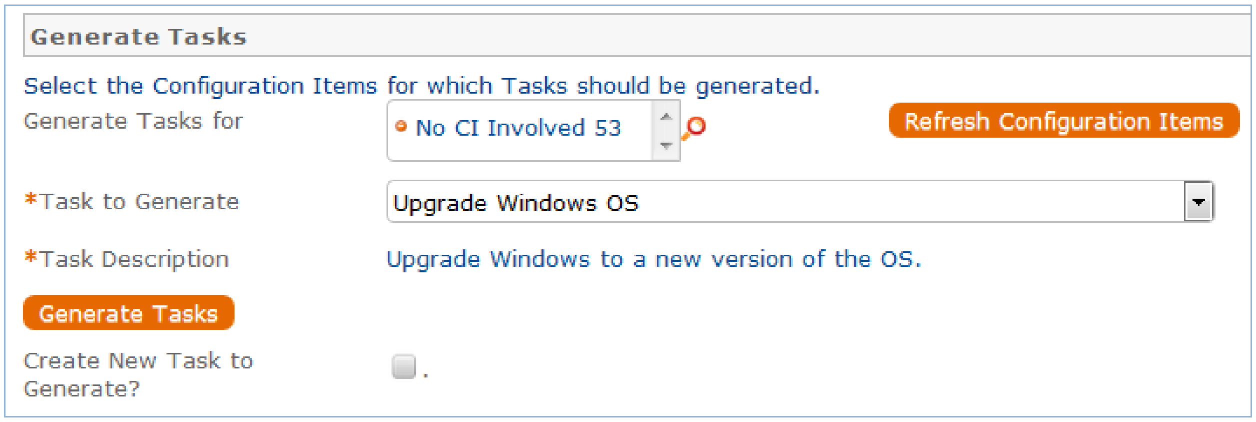
Tasks can be created for the CR before the approvals are done. The only requirement is that the user must first enter the Change Window Start Time and Change Window End Time, as the End Time will be used as the Date Due for the tasks:



Change Scheduling - Required to Generate Tasks

*Change Window Start Time	May 27 2016	00:00	*Change Window End Time	May 31 2016	00:00

Depending on the task generation method, one or more tasks will be created for each selected configuration item. If the task generation method is CR Single Task from Template, then the predefined template will be shown:



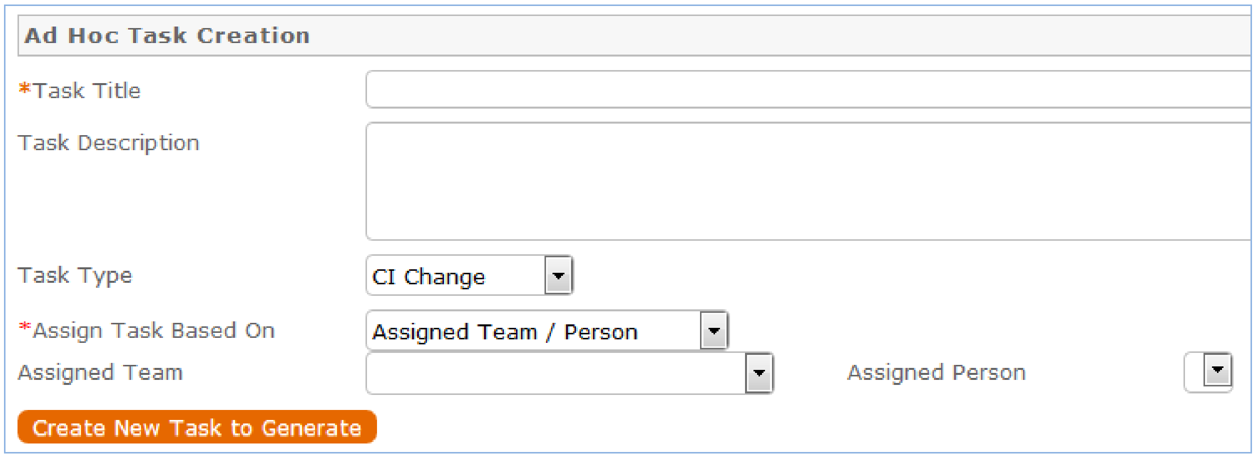
Generate Tasks

Select the Configuration Items for which Tasks should be generated.

Generate Tasks for	No CI Involved 53
*Task to Generate	Upgrade Windows OS
*Task Description	Upgrade Windows to a new version of the OS.
Create New Task to Generate?	☐

Here the default task template is the one called Upgrade Windows OS. Clicking the Generate Tasks button will generate one task for each of the selected CI's (just one in this case).

The user may then select another task to generate when that is done, if desired. He may also check the box for Create New Task to Generate? to create another task template that can be generated for all the selected CI's. Checking that box will bring up some fields to fill in for the ad hoc task:



Ad Hoc Task Creation

*Task Title			
Task Description			
Task Type	CI Change		
*Assign Task Based On	Assigned Team / Person		
Assigned Team		Assigned Person	

Clicking the button to Create New Task to Generate will create a new Task Template, save the change request, and reopen it to edit, with the new task selected in the Task to Generate field. From here, you can click the Generate Tasks button again to add the Task to this CR for each CI that is selected.

Launching Tasks

Tasks may not be launched until any approval process is completed. This is to ensure that no work is started without approval.

Once the approval has been received, clicking the Launch Tasks button will set the first tasks (those with no prerequisite or with any prerequisites completed or not needed) to a status of Assigned. It will also set the Status of the Change Request to In Progress, if it is not already in that status.

See the Task Templates and Task sections above for more details on how tasks are launched and processed.

Completing the Change Request

When all tasks are completed, the Change Manager is notified. Based on whether the tasks were all completed or whether some of them failed, the change manager will then set the Status of the Change Request to Completed or Completed with Failures. Validation prevents the user from choosing Completed as the status if there are tasks whose status is Failed.

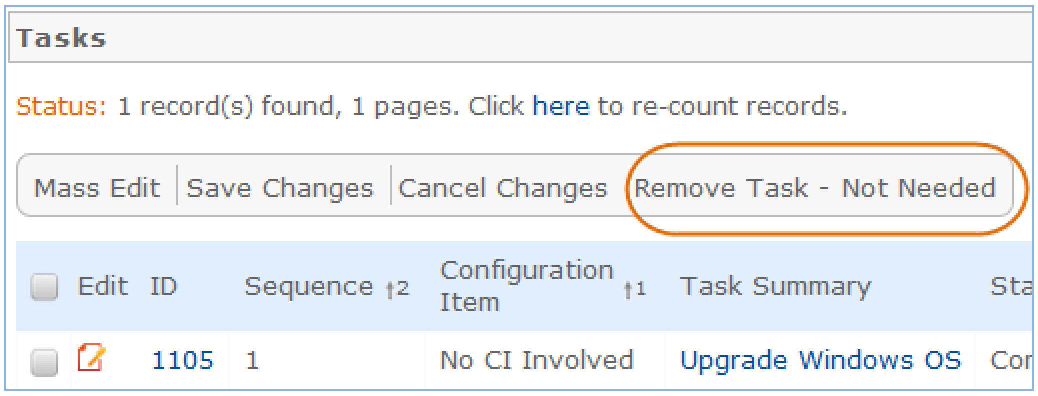
Changing the status to Completed or Completed with Failures will send an email to the Requester informing him that the change is completed.

Working Notes

The Work Notes field is updated from the running working notes for all of the related Tasks.

Related Records and Removing Tasks

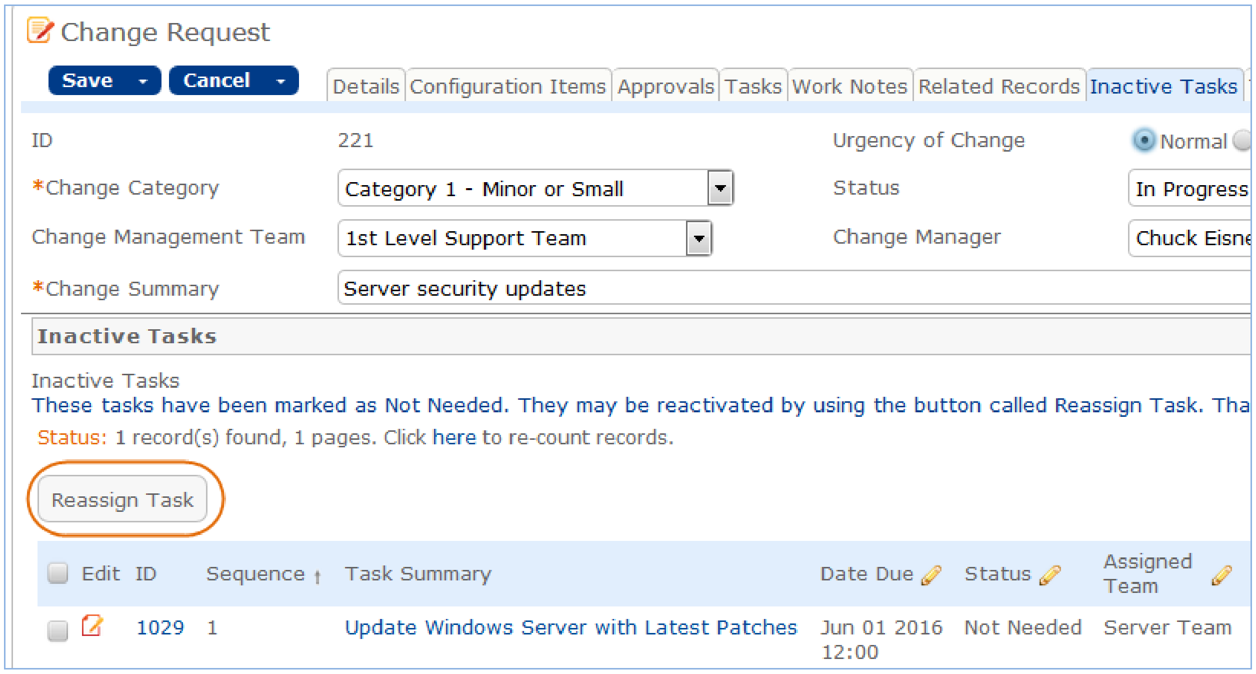
During the working process, some tasks may be removed as not needed by the change manager. The action bar above the Tasks table in the change request includes a button to remove one or more tasks:



Status: 1 record(s) found, 1 pages. Click here to re-count records.

	Edit	ID	Sequence ↑2	Configuration Item ↑1	Task Summary	Sta
		1105	1	No CI Involved	Upgrade Windows OS	Cor

This changes the status of the selected tasks to Not Needed and they move into the Inactive Tasks table shown on the Inactive Tasks tab:


Change Request

ID	221	Urgency of Change	Normal
*Change Category	Category 1 - Minor or Small	Status	In Progress
Change Management Team	1st Level Support Team	Change Manager	Chuck Eisne
*Change Summary	Server security updates		

Inactive Tasks

Inactive Tasks
These tasks have been marked as Not Needed. They may be reactivated by using the button called Reassign Task. Tha
Status: 1 record(s) found, 1 pages. Click here to re-count records.

Edit	ID	Sequence	Task Summary	Date Due	Status	Assigned Team
	1029	1	Update Windows Server with Latest Patches	Jun 01 2016 12:00	Not Needed	Server Team

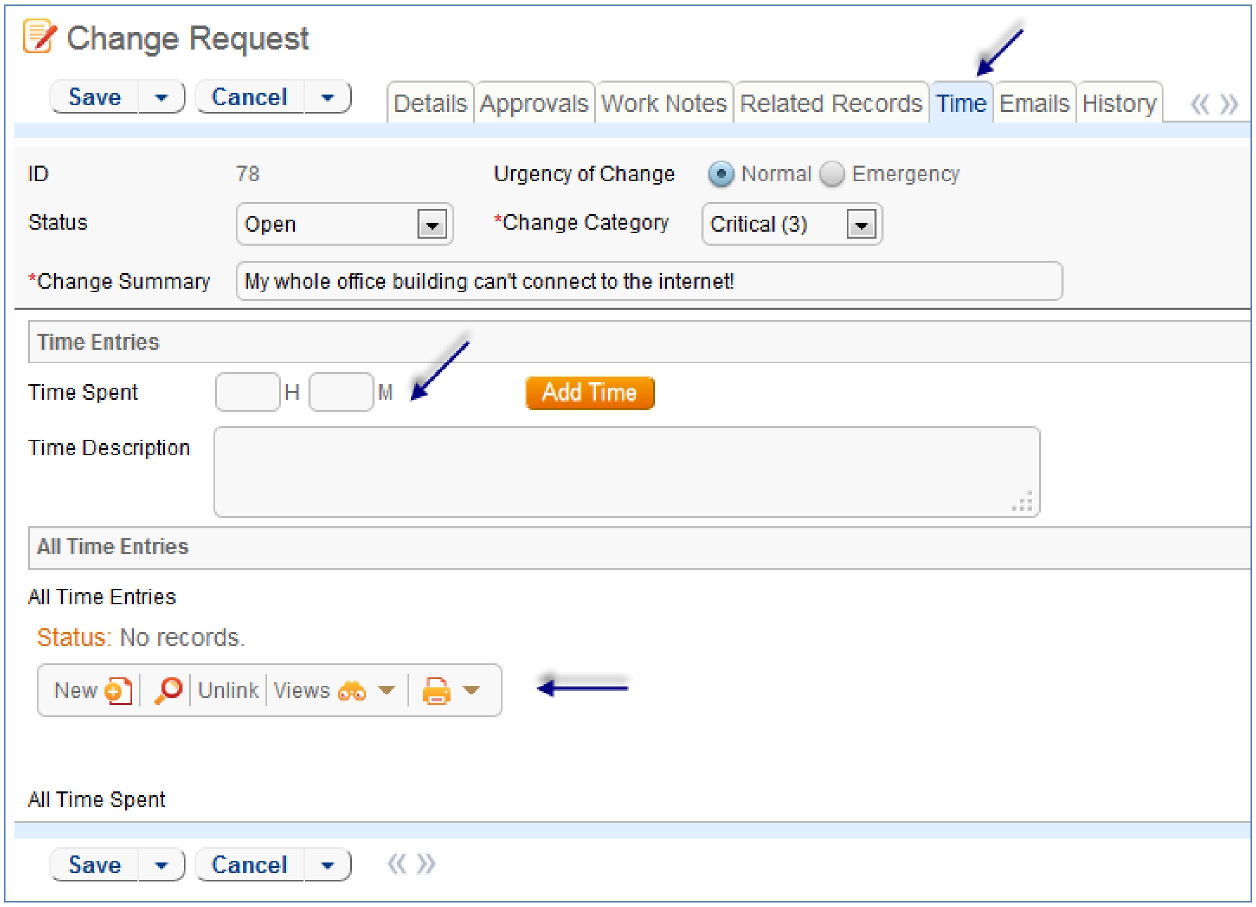
From this area, if it turns out that the task actually is needed, the change manager can select one or more tasks and click the Reassign Task to set their status to Assigned and move them back into the active Tasks table.

Parent-Child Change Request Structure

Previously we used parent-child change requests to manage multiple changes to CI's. That possibility is still available but the fields for relating parent and child change requests have been removed from the layout, since that was a much more cumbersome process. If there is a need to relate change requests in a parent child structure, this logic can be reutilized.

Reporting Time Spent

Technician users may easily report the time they spend on handling change requests. There are two fields and an action button: Time Spent, Time Description, and Add Time on the Work Notes tab of the layout. Entering values there will automatically create a new time entry record when the Add Time button is clicked. The time entry will show the work done by the technician and on the current date.



ID	78	Urgency of Change	[● Normal] Emergency
Status	Open	*Change Category	Critical (3)
*Change Summary	My whole office building can't connect to the internet!		

Time Entries

Time Spent	H M
Time Description	

All Time Entries

All Time Entries

Status: No records.

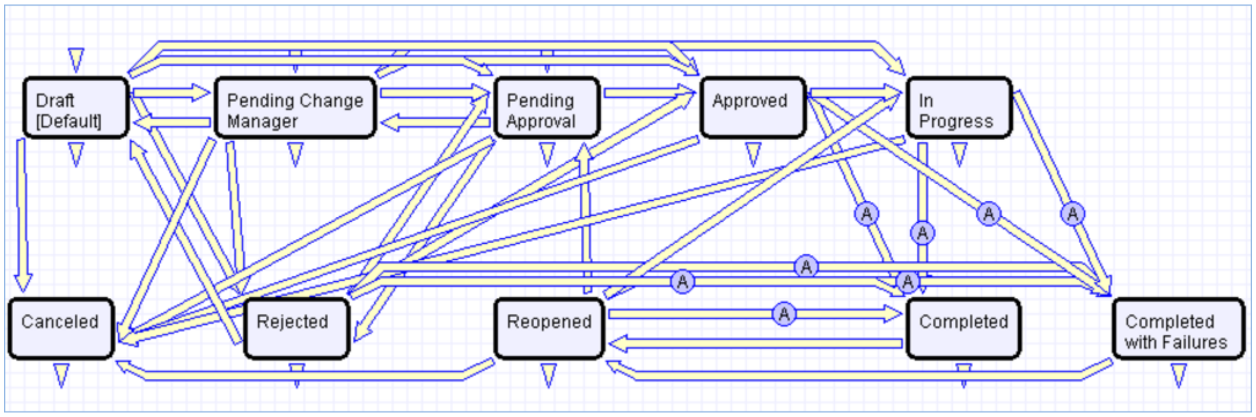
All Time Spent

All the time entries for a request can be seen on the Time tab as well. If a technician needs to report time that was spent on a different day or by a different user, he may click the New button on the Time spent table to submit a time entry directly and change the date or “Done by” field. All time entered is totaled in the All Time Spent field, which can be used in reporting or billing.

Note that this same time entry methodology is used in the Incident, Problem, Service Request, and Task tables.

Workflow

The workflow is used to send the email when the change request is completed. Otherwise, it controls the state transitions, as shown in the screenshot below:



Ownership

Records in the Change Request table are “owned” by the person whose login matches the Creator Login field.

Reporting and Statistics

Forward Schedule of Change

The Forward Schedule of Change represents approved changes and the proposed implementation dates. More generally, it is a list of Change Requests set to occur in the future. Functionally it might include CRs with restricted criteria, such as approved changes only, but by default the Forward Schedule of Change simply shows all Change Requests that are not in a status of Completed, Completed with Failures, or Canceled.

The Forward Schedule of Change is a saved search available from the vertical navigation menu under the Change Request table name. Change Requests are listed in order of Requested Date of Completion. This search is also used in a structured HTML report of the same name. The report groups the CRs by requested date of completion (by week) and shows summary data, including the sum of the Estimated Time to Complete for all records. The report can be configured to be sent in a weekly email to the Change Management team.

Accomplished Change

Accomplished Change is the complement to the Forward Schedule of Change and by default shows all completed Change Requests, sorted by Date Closed.

Configuration Items Table

The Configuration Items table holds records containing information about your company's Configuration Items (assets). It may include CI’s that are in inventory and not in service, as well as all assets that are in service.

Use Case

Configuration Item Creation

Staff may add a Configuration Item manually to the system by creating a new record, or integrated Discovery systems may create CI records automatically with specified attributes.

Configuration Items have type-specific attribute fields (Manufacturer, Model, Serial Number, etc.) and relationships to other CIs, Service Requests, Problems, and Change Requests.

Related Service Requests, Problems, and Change Requests are displayed as lists of records within a CI, so the history of a given CI is readily available.

Configuration Items may also be related to other CI's in an upstream and downstream embedded table.

Record Creation

Configuration Items may be created independently or when selecting Configuration Items in Change Request and Contract records. Configuration Item Name and Configuration Item Type are required fields.

Ownership

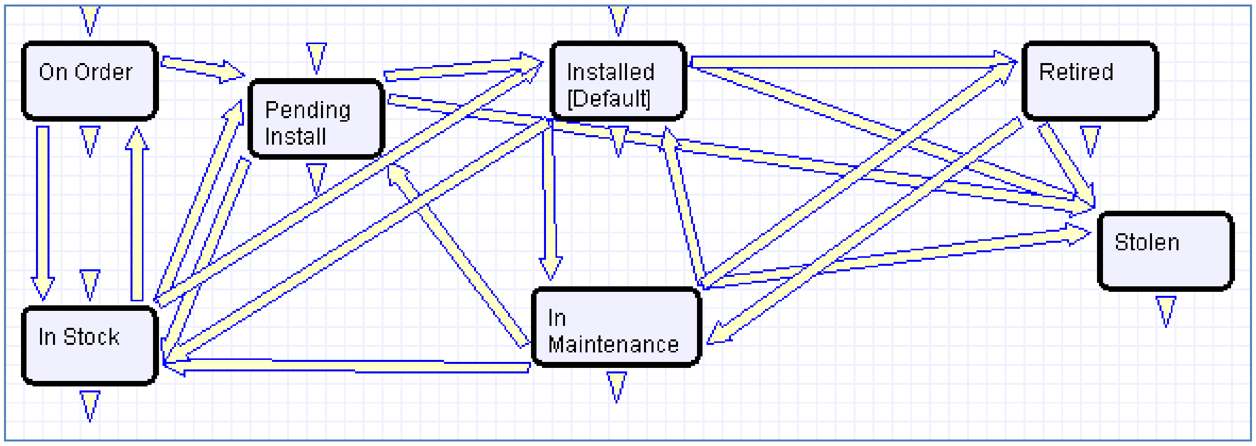
Configuration Item ownership is defined as the user whose Login matches the User Login field in the Configuration Item record.

Processing of Records

Configuration Items follow a lifecycle workflow from ordering through retirement. There are no business rules associated with the Configuration Item table.

Configuration Items are created in the state of Installed by default (see Workflow diagram below). Configuration Items that are requested but not immediately available might be created in a state of On Order, moved to In Stock when the Configuration Item is received, Pending Install while Operations is tasked with installing the Configuration Item, and finally Installed. Installed Configuration Items can change to In Maintenance for repairs and Retired or Stolen when the Configuration Item is no longer in use.

Workflow



Models Table

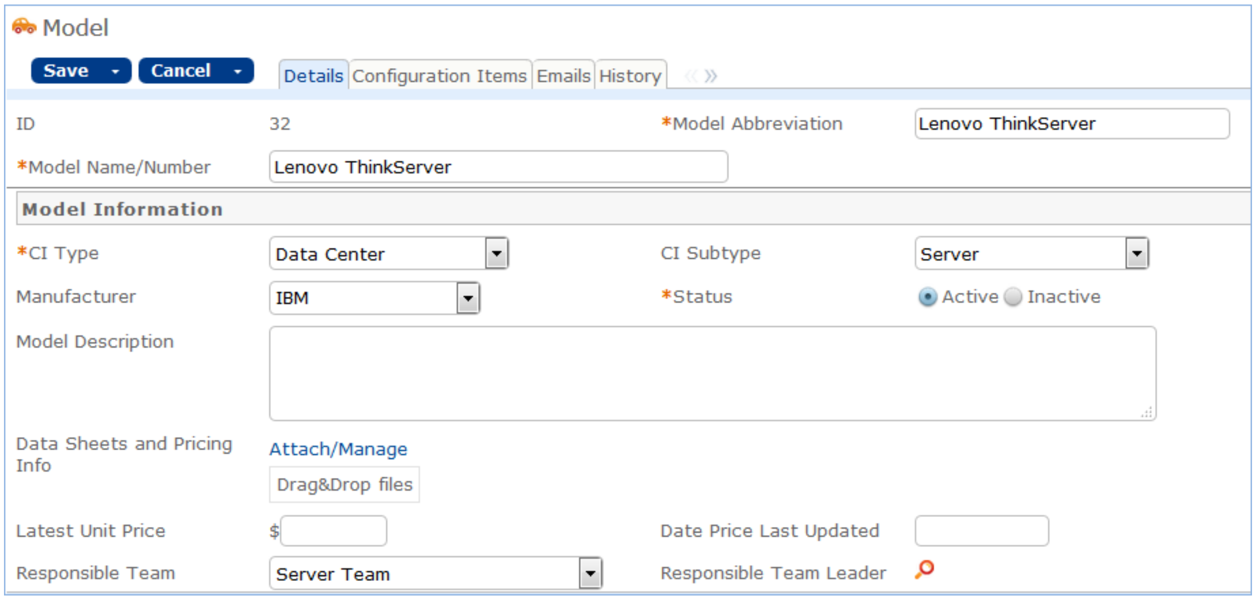
Overview

This table controls the basic configuration item information that may be used within an individual configuration item. It holds records that define details of the CI Type, CI Subtype, Vendor, Manufacturer, and specific model numbers, and so on. Configuration Items then link in the model on which that item is based to prepopulate all of that information for a single CI.

Use Case

Models can be created and edited by members of the admin, admin import, and configuration manager groups. It is expected that the Configuration Managers will be responsible for maintaining model information.

The model form contains the following fields:



ID	32	*Model Abbreviation	Lenovo ThinkServer
*Model Name/Number	Lenovo ThinkServer		

Model Information

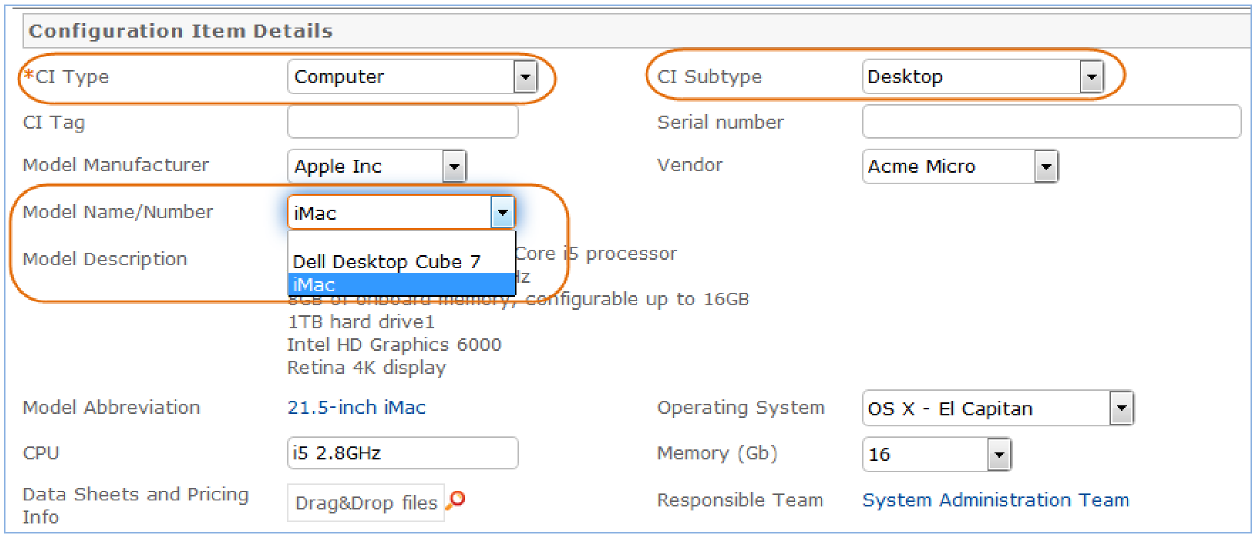
*CI Type	Data Center	CI Subtype	Server
Manufacturer	IBM	*Status	● Active ○ Inactive
Model Description			
Data Sheets and Pricing Info	Attach/Manage Drag&Drop files		
Latest Unit Price	$	Date Price Last Updated	
Responsible Team	Server Team	Responsible Team Leader	

The list of CI Types and CI Subtypes are related in a hierarchical dependency. See the Configuration Item Type and Subtype Dependencies in the appendix for details of that mapping.

The Latest Unit Price gives an idea of the latest pricing for the model. Rules could be set up to push the latest price paid in a purchase into this record to keep it updated as model purchases are made.

Configuration Items Use of Models

Models are pulled into CI's, as shown below. When setting up a new configuration item, the user selects the CI Type, and then the CI Subtype, and then chooses a matching Model Name/Number from the available models. The models are filtered based on CI Type and CI Subtype, and once selected, their details are pulled into the CI record:



Configuration Item Details

*CI Type	Computer	CI Subtype	Desktop
CI Tag		Serial number	
Model Manufacturer	Apple Inc	Vendor	Acme Micro
Model Name/Number	iMac Dell Desktop Cube 7 iMac		
Model Description	Core i5 processor lz configurable up to 16GB 1TB hard drive1 Intel HD Graphics 6000 Retina 4K display		
Model Abbreviation	21.5-inch iMac	Operating System	OS X - El Capitan
CPU	i5 2.8GHz	Memory (Gb)	16
Data Sheets and Pricing Info	Drag&Drop files	Responsible Team	System Administration Team

Automation

There is no automation on the Models table. It is purely a background table.

Purchase Requests Table

Overview

Purchase Requests are used to enable an end user to request new equipment, software, or furniture. A single purchase request can include multiple requested items, which are selected from a list of available items.

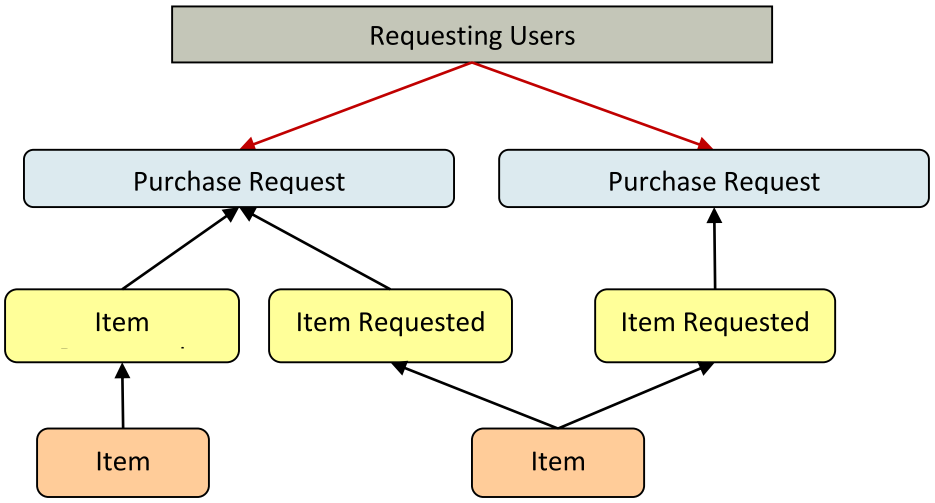
Example Use case

An employee end-user (*lsmith*) needs a new laptop. She logs in and creates a new Purchase Request from the End-User interface. She'll specify who it's for (the record's creator is the requestor by default), the type of purchase by Service Category, the business justification for the request, etc. Lastly she'll select the item(s) she's requesting via a Related Table to the Items Requested table, which in turn draws from the Items table. She may also select from Configuration Items in stock that matches the type of purchase she chose.

Jim Sullivan (*jsullivan*), *lsmith's* manager, receives an email that a new request needs his approval. He can approve or reject directly from hotlinks in the email or edit the record manually from the staff interface. Jim, as a supervisor, will change the Supervisor Approved field to either Yes or No. A rule running on edits will then change the status appropriately (to Supervisor Approved or Supervisor Rejected). If he approves, a notification will be sent to the Procurement Team.

Members of the Procurement Team (e.g. *awilliams*) receive an email stating that a new Purchase Request is awaiting their approval. Members of the Procurement Group will see some additional fields in the Purchase Request record, including a Procurement Group Approved field similar in function to the Supervisor Approved field. They will also see another Yes/No choice field called "All Items Received?" When the recipient/requestor has been satisfied, the Procurement Group changes the status to Fulfilled.

Relationship Diagram



Record Creation

Purchase Requests are creatable by internal customers from the End-User Interface, or by members of the Purchasing team on behalf of other users.

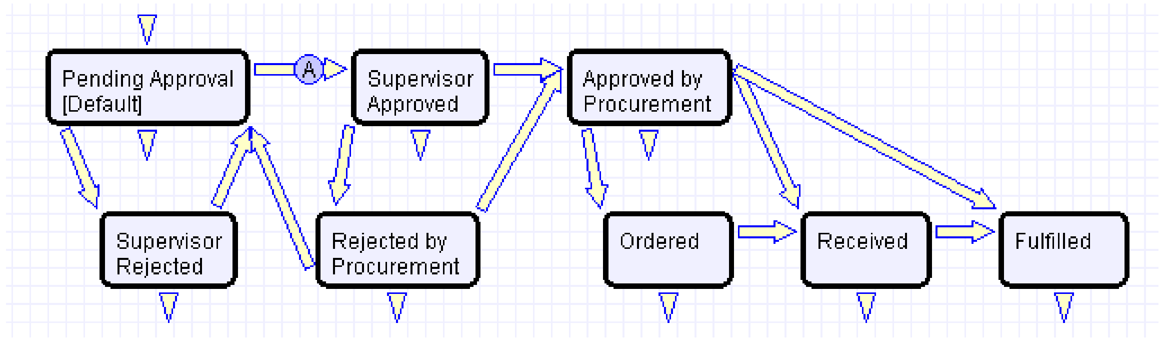
Upon creation, a rule sends emails to the Requestor, the Requestor's Supervisor, and the Purchasing Team notifying them of the new Purchase Request.

Processing of Records

Once created, Purchase Requests follow a two stage approval process. The creation emails sent to the Requestor's Manager and the Purchasing Team lets them know that their approval is required and provides a link in the email to edit the record. A supervisor can edit the "Supervisor Approved?" field to yes or no, and the Purchasing Team edits the "Procurement Group Approved" field to yes or no. Rules validate these changes and move the Purchase Request to the appropriate status, such as "Supervisor Approved" when the supervisor changes the field to Yes.

Once fully approved, Purchase Requests are Ordered, Received, and Fulfilled. Ordered and Received are used if the requested items are not immediately available, and changing the "All Items Received?" field to Yes will move the request into the status of Received. Purchase Requests can move to a status of Fulfilled any time after approval.

Workflow



Purchase Requests have a two stage approval process, first by the Supervisor and secondly by Procurement staff. Once approved, Purchase Requests can move to Ordered/Received states or directly to Fulfilled.

Ownership

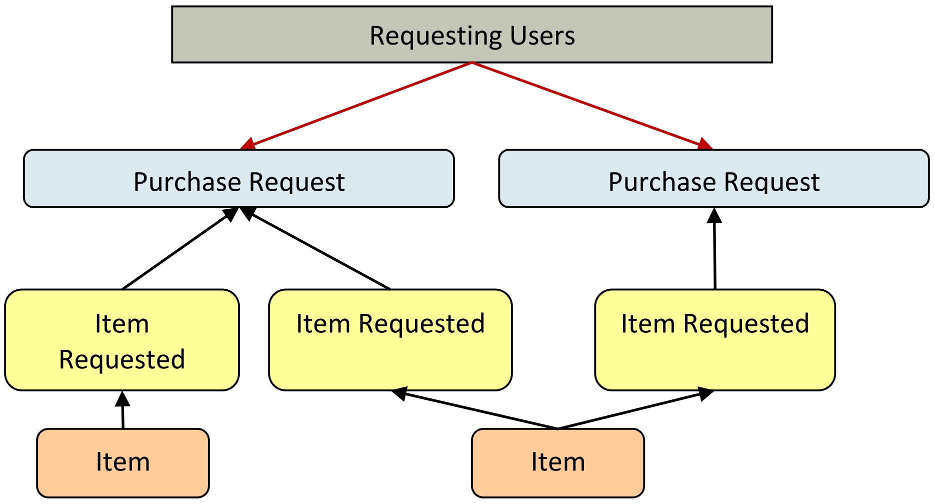
Purchase Requests are owned by the user whose Login matches the Creator Login field in the Purchase Request.

Items Requested Table

Use case

The Items Requested table acts as an intermediary table for the Purchase Request and Item tables. When users request items as part of a Purchase Request, they are creating new Item Requested records. These records represent the Items selected and their quantity for a particular Purchase Request.

Relationship Diagram



Workflow

These records have no workflow and instead follow the Purchase Request record they are linked to.

Record Creation

From a Purchase Request record (see Purchase Request Use Case), a user creates new Item Requested records for each Item desired in the embedded table view. After selecting the Type of Purchase in the Purchase Request record, clicking "New" from the Purchase Request linked Items related table presents the user with the new Item Requested form, filtered to items that fall under the selected Purchase Request Type of Purchase.

The Item Requested form has three major sections: Item Requested details, linked Purchase Request Information, and an area to select a matching Configuration Item. The requestor selects an item from the Item Name drop-down box, which is filtered to show only items in the Type of Purchase category chosen in the Purchase Request record.

The Purchase Request Information fields are automatically populated with data from the Purchase Request this Item Requested was created from and represents the link to the Purchase Request.

The Configuration Item section allows the user or fulfilling staffer to select an existing Configuration Item from the CI table. The CIs available for selection are filtered to only in-stock Configuration Items that match the Type of Item for this Item Requested. The CI fields link the Item Requested to a physical, tracked asset (CI) rather than just the template description (Item), enabling greater control and accurate recordkeeping.

Clicking the "Assign to Requestor" button will automatically update the selected Configuration Item by linking that CI's "User" fields to the same Contact linked here as the Requestor. That is, the button will assign the CI selected to the person requesting the item.

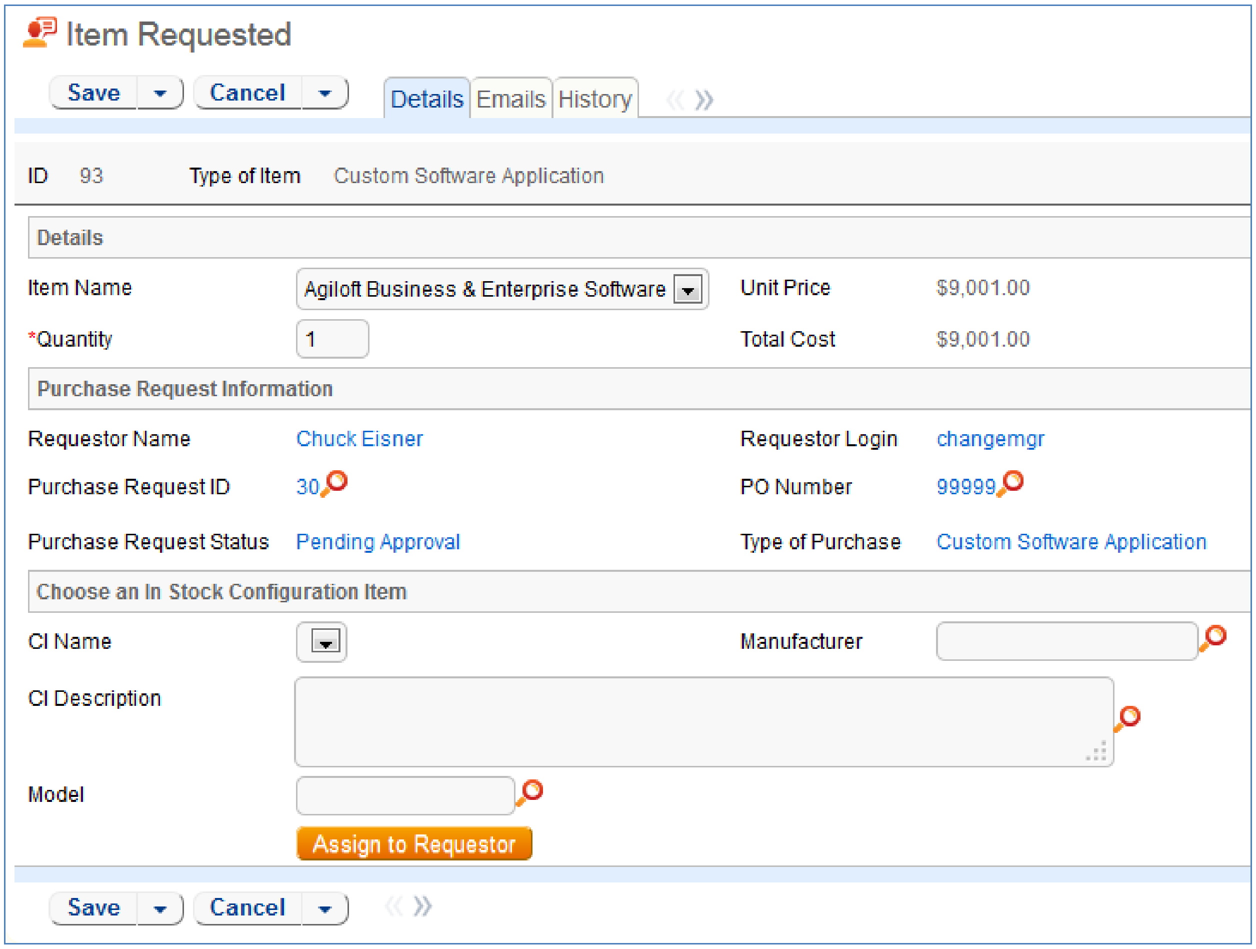
Processing of Records

Once created, Item Requested records remain unchanged and stay linked to the Purchase Request. Since the Item Requested records are maintained separately from the Item records they were spawned from, they retain data that might change over time, such as Unit Price. Thus, even when Items change, Purchase Requests will always show Item data as it was at the time of the Purchase Request and reflect an accurate total.

Ownership

Item Requested records are "owned" by the Contact record whose Login matches the Creator Login field in the Item Requested. Since Items Requested are not maintained outside of a Purchase Request, the relevant ownership stems from the Purchase Request record.

Default Layout



ID 93 Type of Item Custom Software Application

Details

Item Name	Agiloft Business & Enterprise Software	Unit Price	$9,001.00
*Quantity	1	Total Cost	$9,001.00

Purchase Request Information

Requestor Name	Chuck Eisner	Requestor Login	changemgr
Purchase Request ID	30	PO Number	99999
Purchase Request Status	Pending Approval	Type of Purchase	Custom Software Application

Choose an In Stock Configuration Item

CI Name		Manufacturer	
CI Description			
Model			

Items Table

Use case

The Item table stores information for each type of item that can be requested as part of a Purchase Request. The Item Requested table allows the user to select from the Item records in this table when creating a Purchase Request.

Record Creation

Item records are created by internal procurement staff users to represent items that may be selected by end-users for Purchase Requests.

For example, say the Dell Inspiron 2000 Desktop PC is the default model for IT staffers and all employees are set up with one when hired. Procurement staff in charge of maintaining Purchase Requests would create a single record in this Item table to represent the Dell Inspiron 2000 model with all relevant details, such as Make/Model, Unit Price, and descriptive text/choice fields. When users request a Dell Inspiron 2000, an Item Requested record is created from the Item record as a template and then linked to the Purchase Request.

Processing of Records

Once created, Item records require little modification. By default, new Items are created as Active and not requiring Supervisor Approval. If a particular Item type is unavailable, no longer in use, or not ready for distribution, Procurement staff may set the Item to Inactive. Expensive or controlled Items can be toggled to require Supervisor Approval, which can act as a flag to apply approval processes when the Item is requested.

Workflow

Items are not associated with any status fields, but instead contain an Active/Inactive choice field to differentiate selectable Items from non-selectable Items. By default, inactive items are not available for Purchase Requests.

The Item table does not have a Workflow State diagram.

Ownership

Items are owned by the user with the login in the Creator Login field. However, since Items are templates for Requested Items, ownership is less relevant to the individual and records will likely only be maintained by a Procurement Group or similar staff team.

BACKGROUND TABLES

There are several other tables essential to the system that holds data that is less active and workflow-oriented than the request tables. These tables are briefly described here.

Companies Table

Purpose: This table holds information about companies that interact with your organization. It may include customers, vendors, prospects, manufacturer, and so on.

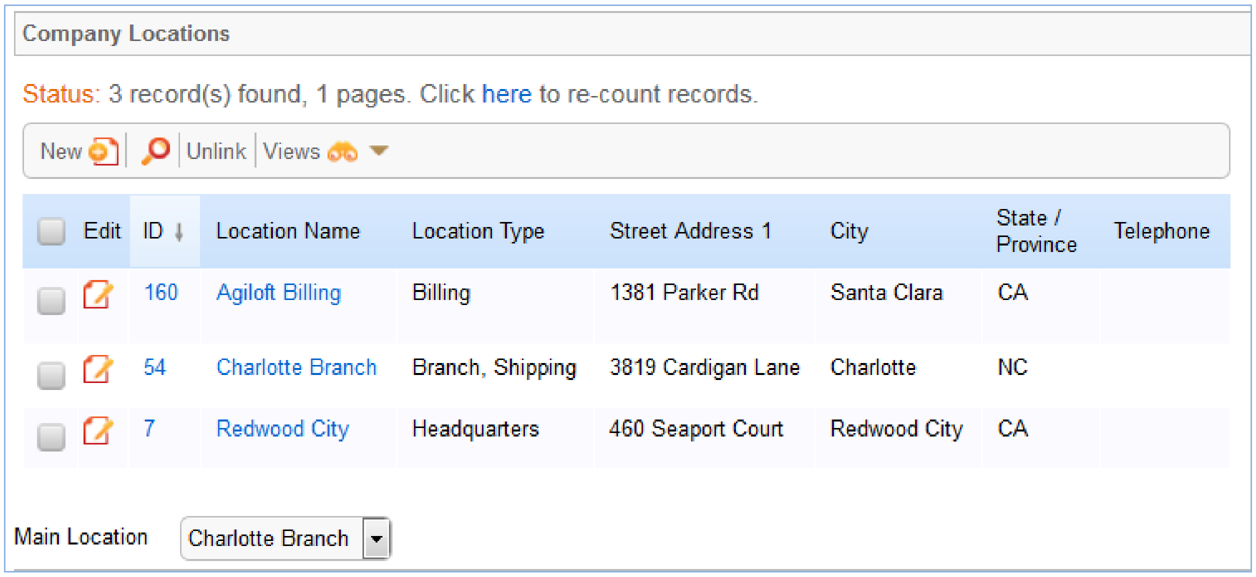
Use Case

Companies may be created by conversion from the Leads table. Members of the Admin, Professional Services, Support Staff and Sales groups may also create new Company records directly. For contract management users, companies can be created with an **action button** when filling out a new Contract record.

The Company table contains mostly static data, and thus does not have any associated workflow actions.

Information about a company and a company's relationship to your organization is stored in a Company record. This information can be everything from the company's billing address to information about their support contracts with your company. More complex data relationships are displayed via linked fields, which pull any records from the Contacts and Opportunities tables that reference the Company record. This allows you to see any business deals or transactions in progress and keep track of any Company employees that your organization contacts.

Company address information is stored in the Locations background table and displayed on the Locations and Contacts tab. A parent company may have several locations, e.g., a billing office, branch locations, and headquarters.



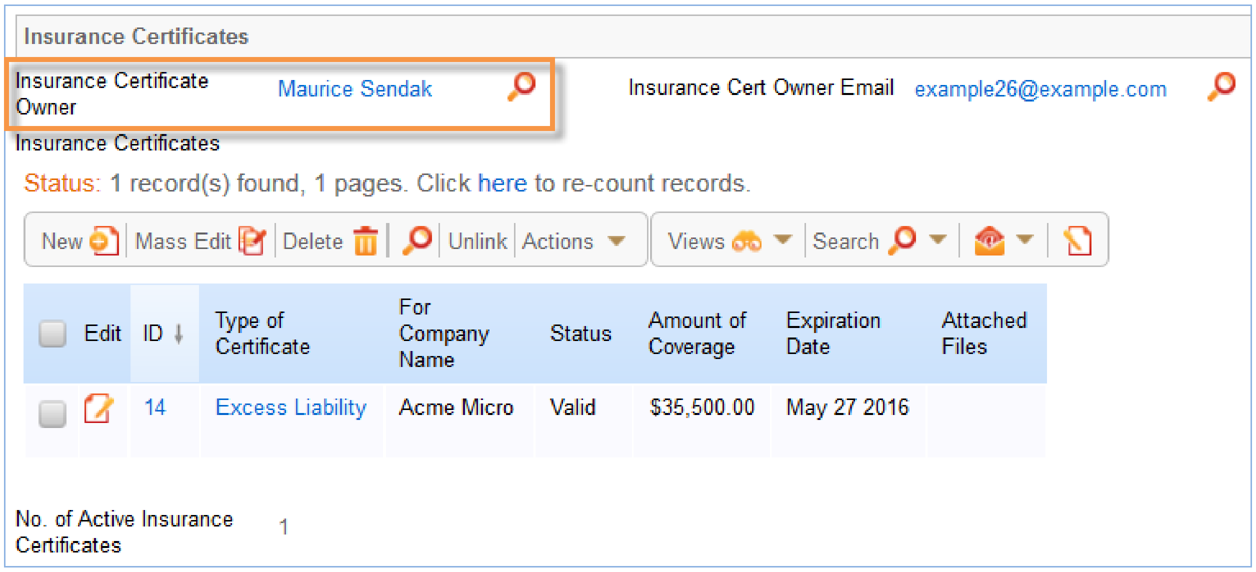
Status: 3 record(s) found, 1 pages. Click here to re-count records.

Edit	ID	Location Name	Location Type	Street Address 1	City	State / Province	Telephone
	160	Agiloft Billing	Billing	1381 Parker Rd	Santa Clara	CA	
	54	Charlotte Branch	Branch, Shipping	3819 Cardigan Lane	Charlotte	NC	
	7	Redwood City	Headquarters	460 Seaport Court	Redwood City	CA	

Main Location: Charlotte Branch

As a background table, many other tables link to the information stored in the Companies table. The Related Records tab shows related tables for Support Cases, Contracts, Insurance Certificates, and Assets.

New Insurance Certificates are typically added from the Company record. An Insurance Certificate Owner is defined just above the Insurance Certificates related table. The Insurance Certificate Owner is notified fourteen days before one or more certificates is due to expire, provided that there is at least one active or pending contract. In the certificate record, the Main Contact is the primary contact at the vendor company.



Insurance Certificates

Insurance Certificate Owner	Maurice Sendak	Insurance Cert Owner Email	example26@example.com

Insurance Certificates

Status: 1 record(s) found, 1 pages. Click here to re-count records.

	Edit	ID	Type of Certificate	For Company Name	Status	Amount of Coverage	Expiration Date	Attached Files
		14	Excess Liability	Acme Micro	Valid	$35,500.00	May 27 2016	

No. of Active Insurance Certificates 1

Insurance certificates added to a company record are automatically linked to all contracts associated with the vendor company. See the INSURANCE CERTIFICATES TABLE for more information.

Ownership

Records in this table are "owned" by the individual assigned sales rep, so each record is associated with a particular user login. Admins and members of the Contract Owner, Contract Manager, Marketing, Project Manager, and Sales groups can view and edit Companies. Most groups can view their own Company, and most internal users can view others' Companies.

People Table, Employees and End Users Sub-Tables

Purpose: The User table tree is used to store information about individuals who use your system.

Use Case

The People table contains data about individual users of your system and individuals with whom your organization has contact or business relationships. There are two basic subtypes of Users: End Users and Employees. Most external Contacts will go in the End User table, as the Employee table tracks information that does not apply to people outside your company, such as working hours and hire date. People records normally have a link to a record in the Company table to define which Company is related to the Person. By default, People records have an embedded list of records from another table, called a related table, listing all of their Support or Helpdesk Case records (tickets).

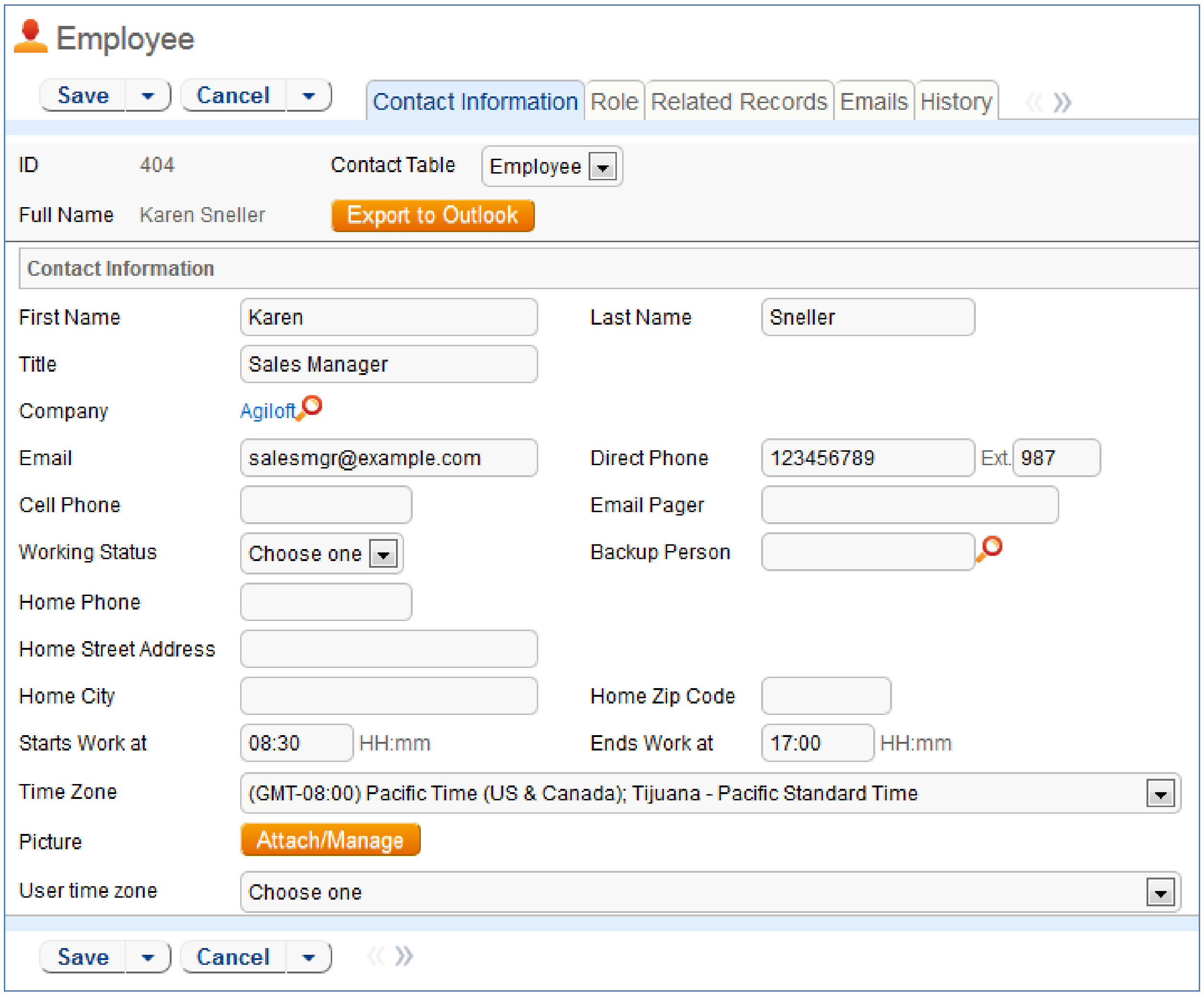
People are considered static in that they do not change status. By default, no workflow is specified for the People table.

People may be created manually by Guests, Support Staff, and members of the Professional Services, Sales staff and Admin groups.

Guest record creation is enabled by default so users can create their own logins using the limited-access "register" account. Records created by the Guest login "register" are added to the "Selfregistered users" group automatically by a rule.

People may also be created as the result of a conversion from a Lead record, or may be created as part of an import from an external database such as an LDAP directory.

Each Person record is assigned either a Team or an individual from the Employees subtable as the "Owner".


Employee

Save | Cancel | Contact Information | Role | Related Records | Emails | History

ID	404	Contact Table	Employee
Full Name	Karen Sneller	Export to Outlook	

Contact Information

First Name	Karen	Last Name	Sneller
Title	Sales Manager		
Company	Agiloft		
Email	salesmgr@example.com	Direct Phone	123456789 Ext. 987
Cell Phone		Email Pager	
Working Status	Choose one	Backup Person	
Home Phone			
Home Street Address			
Home City		Home Zip Code	
Starts Work at	08:30 HH:mm	Ends Work at	17:00 HH:mm
Time Zone	(GMT-08:00) Pacific Time (US & Canada); Tijuana - Pacific Standard Time		
Picture	Attach/Manage		
User time zone	Choose one		

Save | Cancel

The Employee sub-table has a rule attached to it that runs a conversion of an Employee record to the Approvals table when triggered by a field changed on the Employee record. This facilitates the Documents approval process.

Calendars Table

Purpose: This table contains event records displayed in the Calendar block in the left pane, or using a normal table view.

Special Use Case

The Calendar table is a special table used in the Staff Interface Calendar pane feature. Records in this table display as events in the Calendars for the individuals and groups that they reference. Calendar records are mostly static and do not have any default associated workflow, rules, email setup, saved searches or charts.

The Calendar pane is only available to Staff users, and cannot be made available to End Users.

By default, only members of the Admin, Professional Services and Support Staff groups may create Calendar records, and Professional Services and Support Staff may only view their own. No other groups have access to view Calendar entries by default, so other staff groups using the Calendar pane must be individually activated or given view and edit permissions.

Records in this table are "owned" by their creators, and are linked to the creator's login.

EUI Templates Table

Purpose: This table contains HTML files used in creating a customized End User portal. It has sample files that can be used, and you may create your own html files and upload the html content to records in this table to serve up the pages.

Special Use Case

The EUI Template table is a special table used for storing HTML files used in changing the presentation of the Agiloft end user interface. Since these files undergo little or no change during their lifetimes, no workflows, rules, or actions are associated with them, and no charts, reports or saved searches exist for this table.

EUI and Documents Table

When the Documents Table is enabled, the EUI can be configured to allow access by end users to the table. In order to enable this, the "main.php" and "menu.htm" files need to be edited so that all references to "global.menu.file" and "global.home.file" are "globoal.menu-doc.file" and "global.home-doc.file", respectively. These references can be found in '#ew_include("$ewText.get('')")' calls in those html pages.

ADDITIONAL FUNCTIONS

In addition to the tables and functions described above, several additional functional tables are included with the ITIL KB that may be activated for use depending on the functions you requested when signing up for a free KB or downloading and installing. They cover the general functions of Contract Management, Project Management, and CRM (customer support, sales, and marketing). These include the following main tables in addition to their supporting tables: Contracts, Documents, Projects, PO's, Surveys and Survey Answers, Leads, Opportunities, Quotes, Products, and Campaigns.

Contract Management

The Contract Management system is designed to be ready out of the box with a variety of approval processes built-in. It is easy to eliminate or modify any functions to better fit a desired business process.

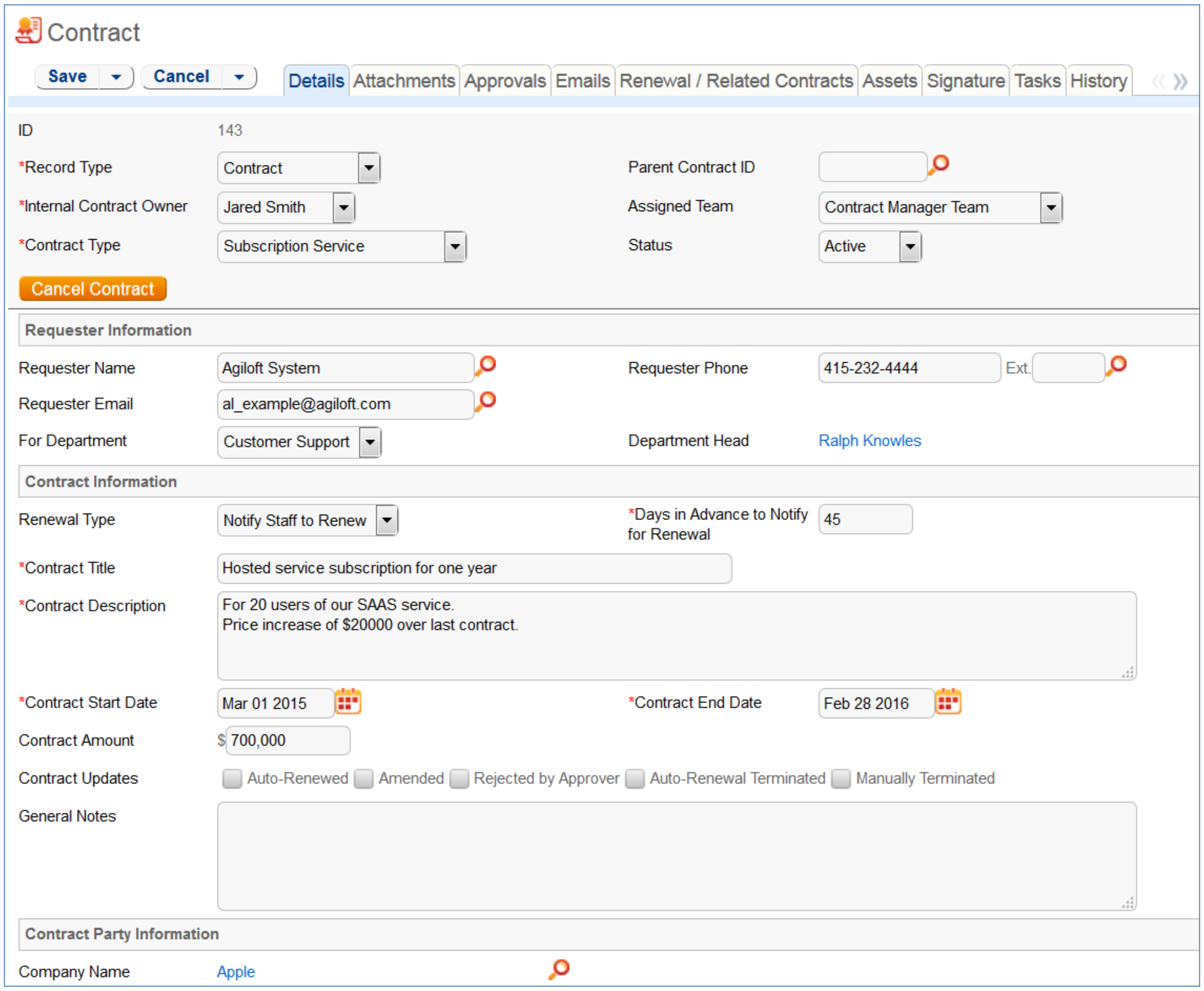
Some time-based rules are disabled by default and need to be enabled for the table to work correctly. To do this, go to the Rules tab of the Setup Contracts page, edit a rule with “(Disabled)” in the title, and click “Yes” in the “Rule is enabled” section at the bottom of the page.

This section describes how the default contract management system is configured.

The Contract Management system is comprised of a few main tables with background tables playing a supporting role. The main tables are described below.

Contracts Table

The Contracts table holds all contract records. It also controls all associated automation and notifications related to contracts. A representative record is shown below.



ID	143		
*Record Type	Contract	Parent Contract ID	
*Internal Contract Owner	Jared Smith	Assigned Team	Contract Manager Team
*Contract Type	Subscription Service	Status	Active

Requester Information

Requester Name	Agiloft System	Requester Phone	415-232-4444 Ext.
Requester Email	al_example@agiloft.com		
For Department	Customer Support	Department Head	Ralph Knowles

Contract Information

Renewal Type	Notify Staff to Renew	*Days in Advance to Notify for Renewal	45
*Contract Title	Hosted service subscription for one year		
*Contract Description	For 20 users of our SAAS service. Price increase of $20000 over last contract.		
*Contract Start Date	Mar 01 2015	*Contract End Date	Feb 28 2016
Contract Amount	$ 700,000		
Contract Updates	☐ Auto-Renewed ☐ Amended ☐ Rejected by Approver ☐ Auto-Renewal Terminated ☐ Manually Terminated		
General Notes			

Contract Party Information

Company Name	Apple

Figure 7. Sample contract record.

Required fields, marked with a red asterisk, are: Record Type, Internal Contract Owner, Contract Type, Days in Advance to Notify for Renewal, Contract Title, Contract Description, Contract Start Date, and Contract End Date.

Staff Use Case

This section covers the use case for staff members inside Agiloft.

Each record in the table holds information about an individual contract including details about the contracting party, approval information, attached contract file and supporting documents, and renewal details.

Contract Creation

Contract records may be created by members of the Admin, Contract Creator, Contract Manager, and Contract Owner groups.

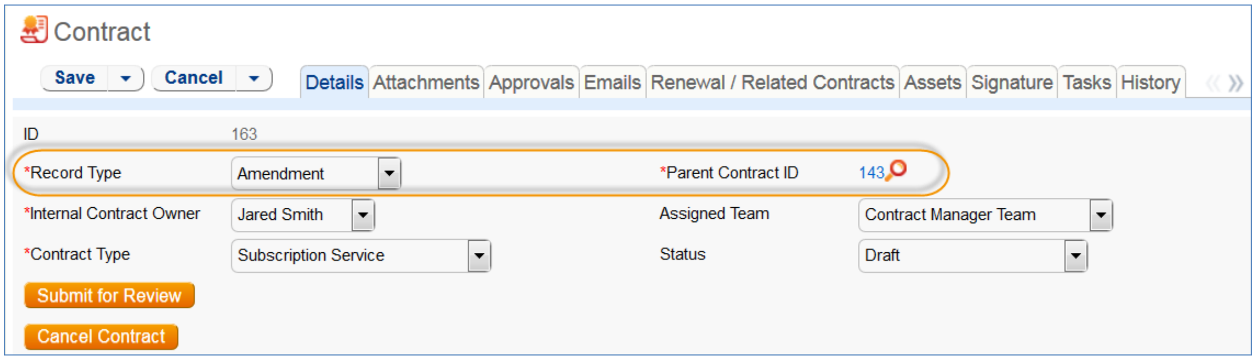
Contracts may be created in one of two ways:

- Click New in the Contract table action bar
- Use the Create Related Contract button on the Renewal / Related Contracts tab to create a renewal, subcontract, or amendment from the current contract. This button is available only if the contract is in a Status of Signed, Active, Expired, or Canceled. Creating a new contract with the Create Related Contract button will automatically link it to the current contract by populating the Parent Contract ID field in the newly created record. For a more detailed explanation of creation by this method, see the HANDLING RELATED CONTRACTS AND RENEWALS section.

Contracts fall into one of four categories:

- Contracts
- Master Agreements
- Subcontracts
- Amendments

The Record Type field in the common area indicates the contract’s category. “Contract” is the default Record Type for newly created contracts. It can be used to indicate either a stand-alone contract or a contract that exists under a master agreement. In the latter case, the Parent Contract ID can be filled in upon creation of the contract. Subcontracts and Amendments will be linked to a parent contract automatically, if they are created by clicking the Create Related Contract button in the parent contract record.

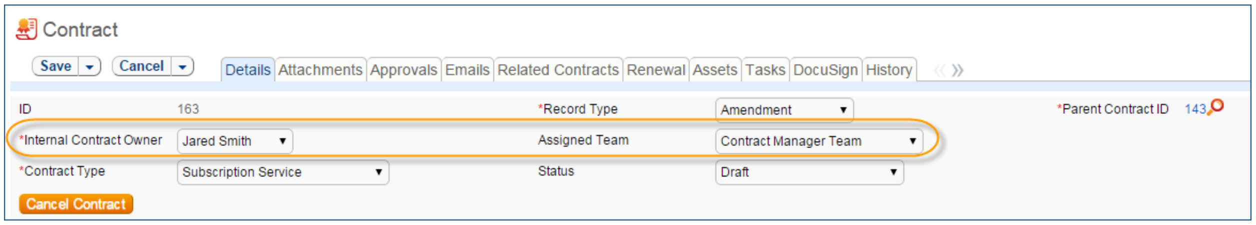


Contract

ID	163		
*Record Type	Amendment	*Parent Contract ID	143
*Internal Contract Owner	Jared Smith	Assigned Team	Contract Manager Team
*Contract Type	Subscription Service	Status	Draft

Figure 8. The type of contract is shown in the Record Type field.

Below the Record Type are fields storing the Assigned Team and the Internal Contract Owner. The default Assigned Team is the Contract Manager Team. The Internal Contract Owner is the person responsible for overseeing the contract and ensuring timely renewals and approvals. The list of available owners is filtered to users who are in the Contract Owner Team or the Contract Manager Team. The default Internal Contract Owner is the user who creates the contract record, provided that user is in the Contract Manager or Contract Owner Team. Users in the Contract Manager or Admin group can manually change the Assigned Team and Internal Contract Owner if needed.



Contract

ID	163	*Record Type	Amendment	*Parent Contract ID	143
*Internal Contract Owner	Jared Smith	Assigned Team	Contract Manager Team		
*Contract Type	Subscription Service	Status	Draft		

Figure 9. The Internal Contract Owner is responsible for overseeing the contract.

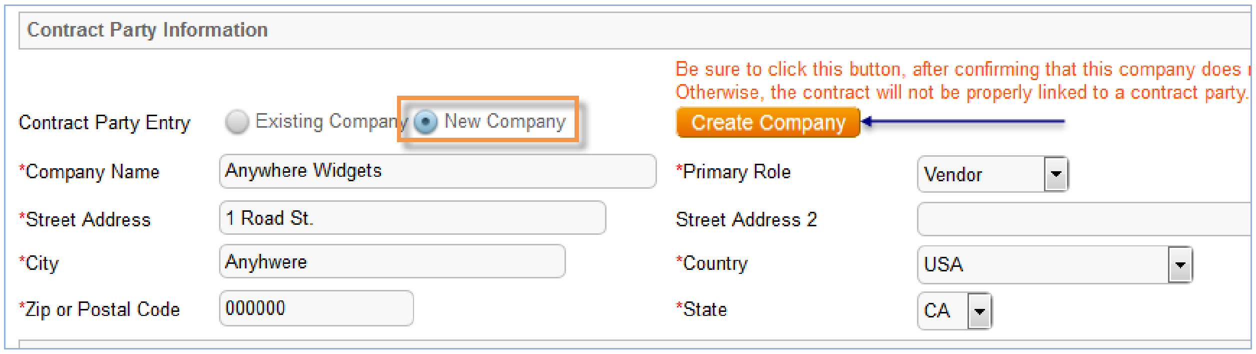
Once the appropriate fields are filled in, the contract may be saved in a Status of Draft.

The contract's Status field is changed automatically by the system at appropriate points in the workflow, generally when an action button is pressed or when some condition is met; however, users in the Admin group can manually override the Status if necessary.

Information about the contract requester, external company involved, and locations related to the contract may be added by clicking the lookup icon next to those fields. If a desired Requester, Company, or Location does not exist, a record must first be created in order to link it to the Contract.

Creating companies and contacts during contract creation

In the Contract Party Information section, users can find and link to an existing company, or create new ones as needed. To find an existing company record, use the lookup icon to search for the company. If the contract party isn't found, staff users can add a new company by selecting New Company. Enter the name and address information, then click Create Company.



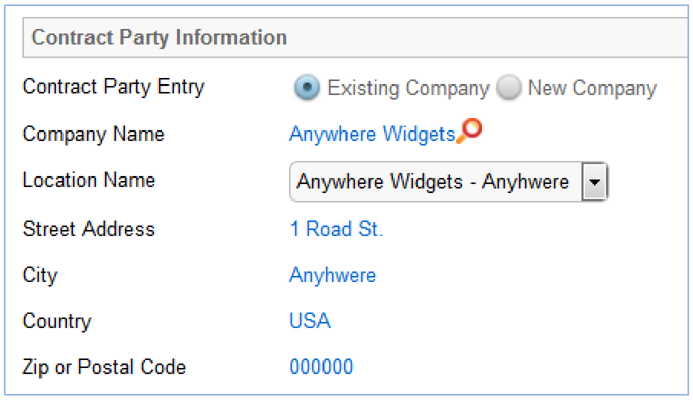
Contract Party Information

Be sure to click this button, after confirming that this company does
Otherwise, the contract will not be properly linked to a contract party.

Contract Party Entry	Existing Company [✓ New Company]	Create Company	
*Company Name	Anywhere Widgets	*Primary Role	Vendor
*Street Address	1 Road St.	Street Address 2	
*City	Anyhwere	*Country	USA
*Zip or Postal Code	000000	*State	CA

Figure 10. Create and link to new company records when filling out the new contract form.

Both company and location records are created in the background, and linked to the new contract when the record refreshes.

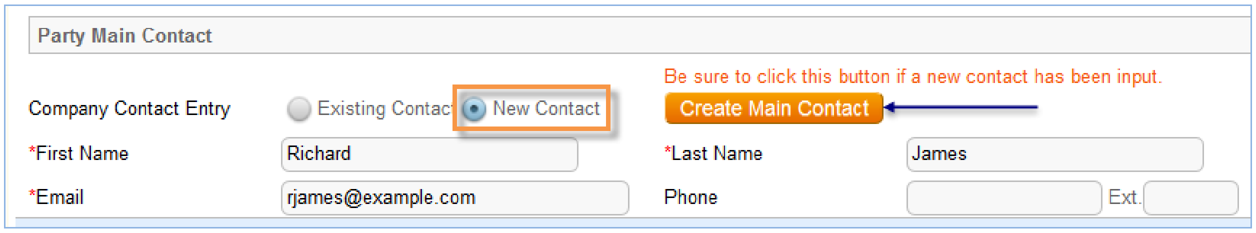


Contract Party Information

Contract Party Entry	[✓ Existing Company] New Company
Company Name	Anywhere Widgets
Location Name	Anywhere Widgets - Anyhwere
Street Address	1 Road St.
City	Anyhwere
Country	USA
Zip or Postal Code	000000

Figure 11. Company records created in the contract form are automatically linked after record creation.

Similarly, you can create new entries in the people table if the main contact for the contract party is not an existing contact. Select New Contact, enter the information, and click Create Main Contact. Multiple contacts can be created in this way, and then the primary contact can be selected from the Main Contact dropdown.



Party Main Contact

Be sure to click this button if a new contact has been input.

Company Contact Entry	Existing Contact [✓ New Contact]	Create Main Contact	
*First Name	Richard	*Last Name	James
*Email	rjames@example.com	Phone	Ext.

Figure 12. Create a new contact if the contract party is not yet in the system.

When Contract Requesters submit a new contract through the end user interface, they can enter new company and contact information, but they don't see the buttons to create new records. When a contract is submitted for review, Contract Managers can edit the contract party information, or click the button to confirm the new company or contact and finish adding records.

This setup is intended to prevent end users from creating duplicate companies with slightly different spellings and to ensure that contract managers have ownership and control of this data. Permissions can be changed easily to allow end users to use the buttons to create companies and contacts directly.

Note that it is important for the contract managers to evaluate the new additions and either replace with existing data or click the buttons to create the appropriate background data. Otherwise, the linked data can't be used in print templates and emails.

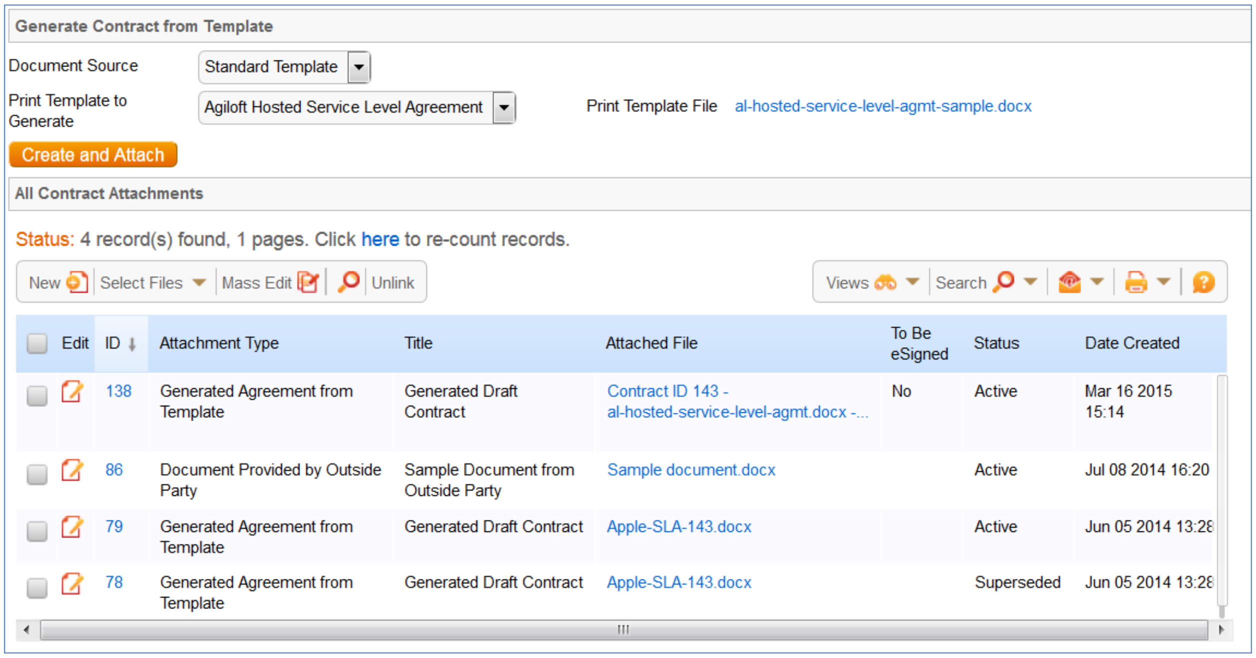
Creating contract attachments from default print templates

The Attachments tab shows all attached files related to the contract and provides the user with options to create and edit related attachments. For certain contract types, when the Document Source is set to Standard Template or Modified Template, the Print Template to Generate field appears. The Document Source can also be set to 3rd Party or Internal – Other to indicate a document provided by the external contract party, or which has been internally generated but not from a template, respectively. The Print Template to Generate has a default value set based on the contract type. The Create and Attach action button is used to auto-generate a contract document from the MS Word print template specified in the Print Template to Generate field and stored in the Print Template File field. Clicking Create and Attach generates the document and creates an attachment record to hold the attached file. It also increments the Version Number field.

If the Contract Party Information section is not properly filled out and linked to records, a validation action warns the user to finish creating new records before generating the attachment.

Creating contracts and attachments from inbound emails

Attachments can be automatically generated from inbound emails containing attached files. Files from inbound emails are mapped to the Transitional Contract Files field in the contract, assigned a predetermined Attachment Title and Attachment Type value, and then converted into a new attachment record. The file is mapped into the Attached File field in the attachment record. If multiple files are attached to an inbound email, each one is converted into a separate attachment record for that contract.


Generate Contract from Template

Document Source	Standard Template
Print Template to Generate	Agiloft Hosted Service Level Agreement
Print Template File	al-hosted-service-level-agmt-sample.docx

Status: 4 record(s) found, 1 pages. Click here to re-count records.

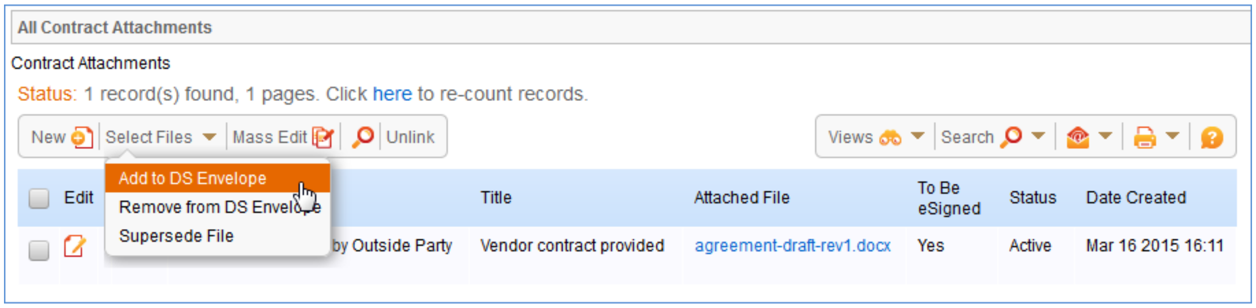
Edit	ID	Attachment Type	Title	Attached File	To Be eSigned	Status	Date Created
	138	Generated Agreement from Template	Generated Draft Contract	Contract ID 143 - al-hosted-service-level-agmt.docx -...	No	Active	Mar 16 2015 15:14
	86	Document Provided by Outside Party	Sample Document from Outside Party	Sample document.docx		Active	Jul 08 2014 16:20
	79	Generated Agreement from Template	Generated Draft Contract	Apple-SLA-143.docx		Active	Jun 05 2014 13:28
	78	Generated Agreement from Template	Generated Draft Contract	Apple-SLA-143.docx		Superseded	Jun 05 2014 13:28

Figure 13. Attachments are managed and created on the Attachments tab.

Working with attachments

The All Contract Attachments related table is also used to manually create and attach files to the contract, such as performance bonds, signed agreements, or other documents from the contract parties. Create attachments from within the contract record by clicking New on the related table's action bar.

The Select Files > Add/Remove from DS Envelope actions on the action bar can be used to add or remove selected attachments to and from the DocuSign envelope when using DocuSign integration. When these actions are selected, the To Be eSigned field is set to Yes or No, and the DS Files to Sign field on the DocuSign tab is updated with the attachments. The Select Files > Supersede File action sets the Status of the selected file to Superseded.


All Contract Attachments

Contract Attachments

Status: 1 record(s) found, 1 pages. Click here to re-count records.

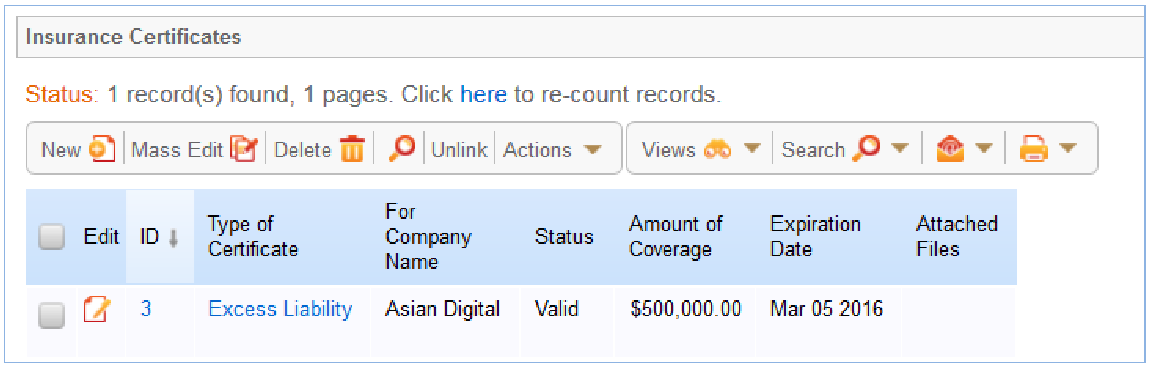
Edit		Title	Attached File	To Be eSigned	Status	Date Created
	by Outside Party	Vendor contract provided	agreement-draft-rev1.docx	Yes	Active	Mar 16 2015 16:11

Figure 14. Attachments can be added or removed from the DocuSign envelope before sending.

For more information on Attachments, refer to the ERROR! REFERENCE SOURCE NOT FOUND. section.

Tracking insurance certificates

The Attachments tab also holds a *related table* that can be used to track insurance certificates separately from other types of attachments. Certificates can be found directly in the Insurance Certificates table, and in *related tables* in both the Company and Contracts tables.


Insurance Certificates

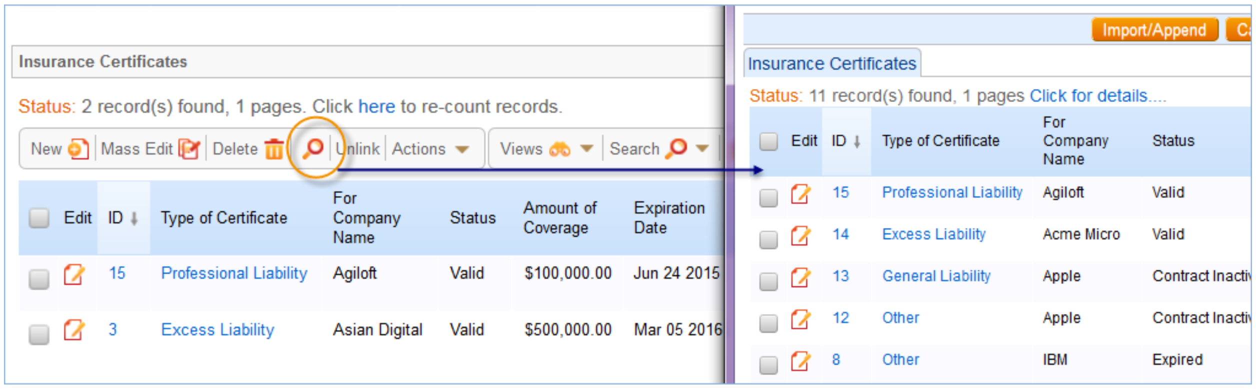
Status: 1 record(s) found, 1 pages. Click here to re-count records.

Edit	ID	Type of Certificate	For Company Name	Status	Amount of Coverage	Expiration Date	Attached Files
	3	Excess Liability	Asian Digital	Valid	$500,000.00	Mar 05 2016	

Figure 15. Create and manage insurance certificates on the Attachments tab.

When you add an insurance certificate to a company record, it is automatically linked to all pending or active contracts with that vendor by a daily time-based rule (disabled by default).

To create a certificate from a contract record, click New on the related table **action bar**; the certificate will be automatically linked to the contract. You will need to look up the company vendor in the For Company Name field when creating certificates from a contract.



Insurance Certificates

Status: 2 record(s) found, 1 pages. Click here to re-count records.

	Edit	ID	Type of Certificate	For Company Name	Status	Amount of Coverage	Expiration Date
		15	Professional Liability	Agiloft	Valid	$100,000.00	Jun 24 2015
		3	Excess Liability	Asian Digital	Valid	$500,000.00	Mar 05 2016

Insurance Certificates

Status: 11 record(s) found, 1 pages Click for details....

	Edit	ID	Type of Certificate	For Company Name	Status
		15	Professional Liability	Agiloft	Valid
		14	Excess Liability	Acme Micro	Valid
		13	General Liability	Apple	Contract Inacti
		12	Other	Apple	Contract Inacti
		8	Other	IBM	Expired

Figure 16. Use the lookup icon to find and link to existing insurance certificates.

The Insurance Certificate Owner (defined in the company record) is notified fourteen days before an insurance certificate's Expiration Date. When a certificate expires without being renewed, the Contract Managers of any related active contracts are notified on the date of the expiration.

For more information about insurance certificates and related automation, see the INSURANCE CERTIFICATES TABLE section.

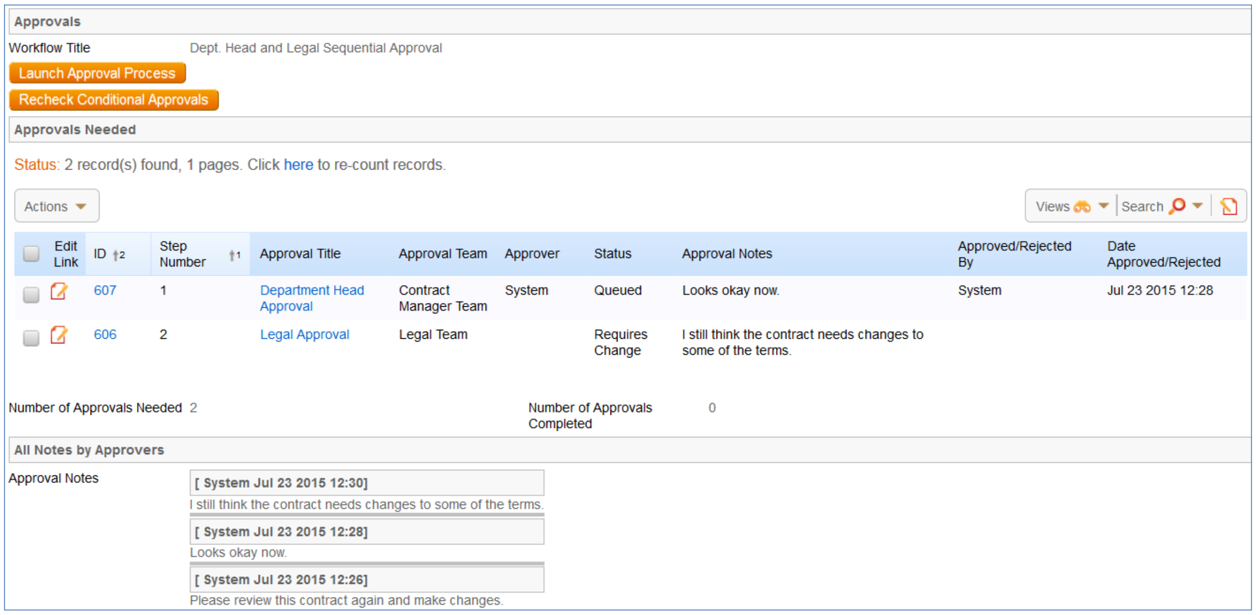
Another method for managing insurance certificates

You may prefer to manage insurance certificates as contract attachments, instead of using the Insurance Certificates table. For example, if Contract Managers are responsible for renewing and maintaining insurance certificates, it may be best to use the Attachments table to hold the certificates, along with all other types of attached documents for contracts. Managing insurance certificates as contract attachments means the Contract Manager will receive email notifications instead of the Internal Certificate Owner.

If this method makes the most sense for your particular business, you can remove the default Insurance Certificates field from the contract layout. Then, go to the Attachment Type table and edit the record called Insurance Certificate. By default, this record's Status is Inactive; change the Status to Active to make it visible in the Attachments table.

Adding notes and sending emails

Notes relevant to the general contract process are entered into the General Notes field on the Details tab, and notes pertaining to the approval process are entered into the Approval Notes field on the Approvals tab.


Approvals

Workflow Title: Dept. Head and Legal Sequential Approval

Launch Approval Process

Recheck Conditional Approvals

Approvals Needed

Status: 2 record(s) found, 1 pages. Click here to re-count records.

Edit Link	ID	Step Number	Approval Title	Approval Team	Approver	Status	Approval Notes	Approved/Rejected By	Date Approved/Rejected
	607	1	Department Head Approval	Contract Manager Team	System	Queued	Looks okay now.	System	Jul 23 2015 12:28
	606	2	Legal Approval	Legal Team		Requires Change	I still think the contract needs changes to some of the terms.		

Number of Approvals Needed: 2

Number of Approvals Completed: 0

All Notes by Approvers

Approval Notes

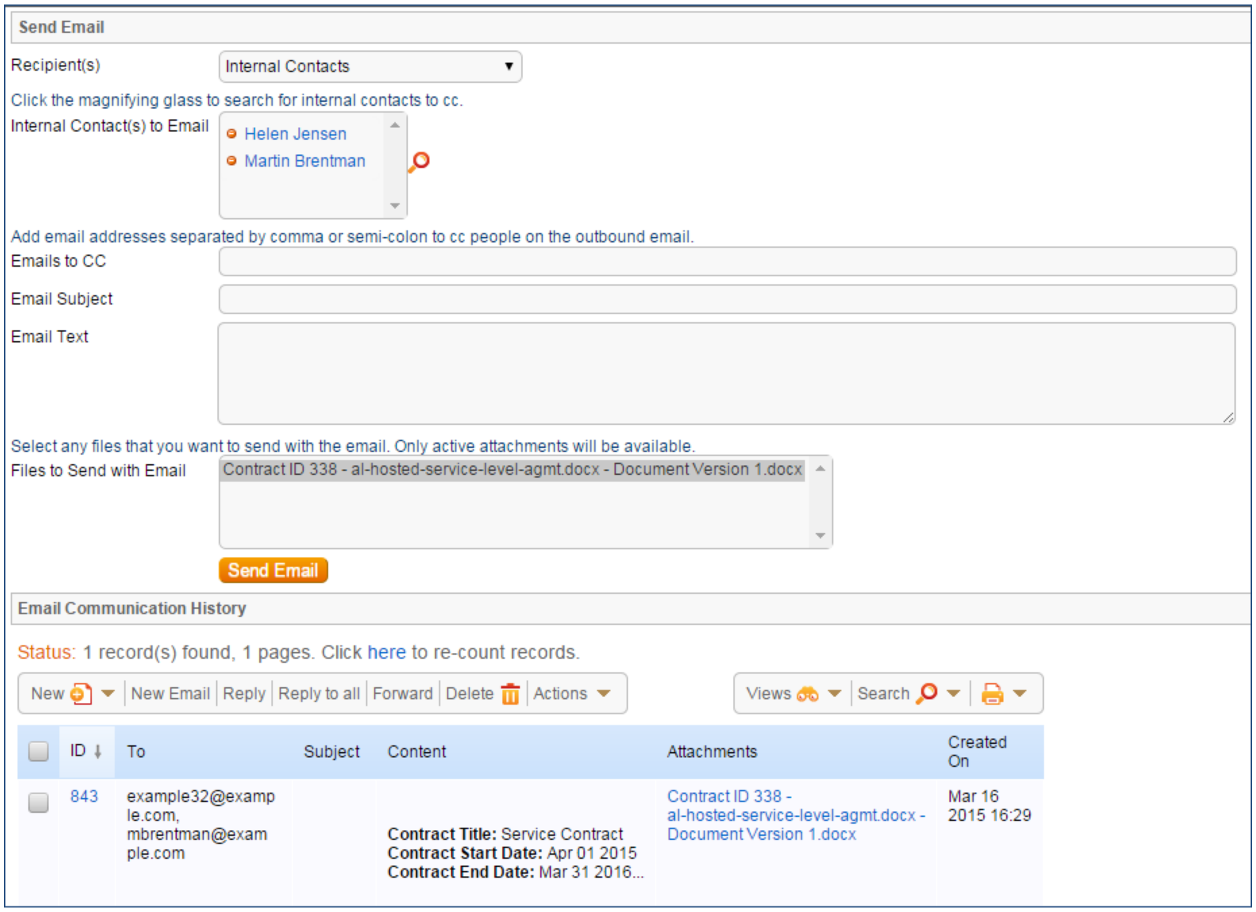
[System Jul 23 2015 12:30]
I still think the contract needs changes to some of the terms.

[System Jul 23 2015 12:28]
Looks okay now.

[System Jul 23 2015 12:26]
Please review this contract again and make changes.

Figure 17. The Approval Notes field shows notes from all approvers for the contract.

Emails can be sent to Internal Contacts and External Party Contacts from the Emails tab. First choose the type of Recipient(s) (Internal Contacts, External Party Contacts, or External Party and Internal Contacts), then click the lookup icon to select which contacts to email. Fill out the Email Subject and Email Text fields as needed, select any files to send with the email, then click Send Email.


Send Email

Recipient(s): Internal Contacts

Click the magnifying glass to search for internal contacts to cc.

Internal Contact(s) to Email: Helen Jensen, Martin Brentman

Add email addresses separated by comma or semi-colon to cc people on the outbound email.

Emails to CC

Email Subject

Email Text

Select any files that you want to send with the email. Only active attachments will be available.

Files to Send with Email: Contract ID 338 - al-hosted-service-level-agmt.docx - Document Version 1.docx

Email Communication History

Status: 1 record(s) found, 1 pages. Click here to re-count records.

ID	To	Subject	Content	Attachments	Created On
843	example32@example.com, mbrentman@example.com		**Contract Title:** Service Contract **Contract Start Date:** Apr 01 2015 **Contract End Date:** Mar 31 2016...	Contract ID 338 - al-hosted-service-level-agmt.docx - Document Version 1.docx	Mar 16 2015 16:29

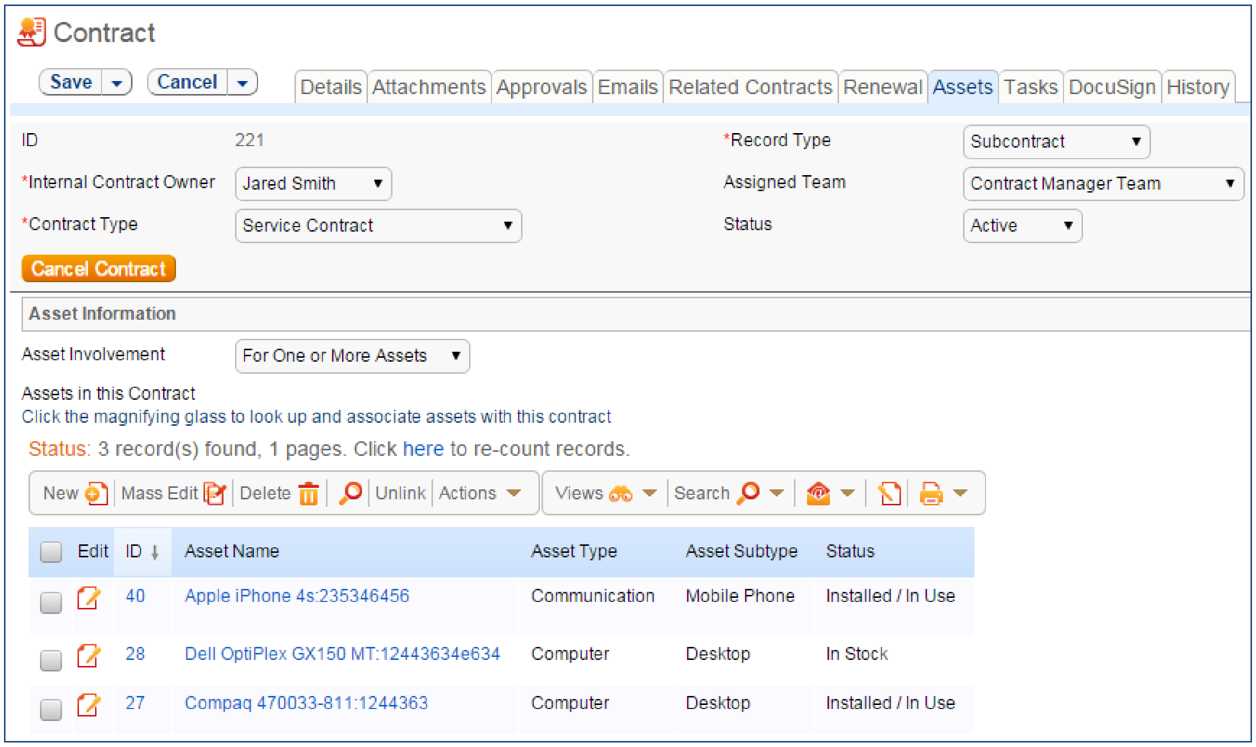
Figure 18. The Emails tab provides a way to send email to contract parties and include attached files. It also shows all previous related emails.

Renewals and previous contracts

Information about the renewal process is stored on the Renewal / Related Contracts tab. Fields for capturing the Renewal Notification Date and Days in Advance to Notify for Renewal are provided as a default. If relevant, information about the renewal contract or previous contract are automatically updated by the system. Users typically do not manually link contracts.

Related assets

Assets can be linked to individual contracts from the Assets tab. Assets must be added separately to the Assets table before they are available to attach to contracts.



Contract

Save | Cancel

Details | Attachments | Approvals | Emails | Related Contracts | Renewal | Assets | Tasks | DocuSign | History

ID	221	*Record Type	Subcontract
*Internal Contract Owner	Jared Smith	Assigned Team	Contract Manager Team
*Contract Type	Service Contract	Status	Active

Cancel Contract

Asset Information

Asset Involvement: For One or More Assets

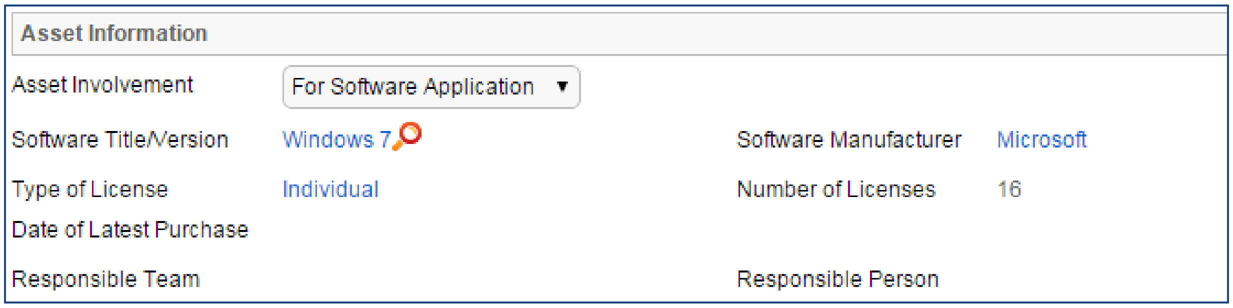
Assets in this Contract
Click the magnifying glass to look up and associate assets with this contract

Status: 3 record(s) found, 1 pages. Click here to re-count records.

Edit	ID	Asset Name	Asset Type	Asset Subtype	Status
	40	Apple iPhone 4s:235346456	Communication	Mobile Phone	Installed / In Use
	28	Dell OptiPlex GX150 MT:12443634e634	Computer	Desktop	In Stock
	27	Compaq 470033-811:1244363	Computer	Desktop	Installed / In Use

Figure 19. Assets related to the contract are managed on the Assets tab.

Depending on the selection in the Asset Involvement field, different fields appear for the user to input linked assets.



Asset Information

Asset Involvement	For Software Application		
Software Title/Version	Windows 7	Software Manufacturer	Microsoft
Type of License	Individual	Number of Licenses	16
Date of Latest Purchase			
Responsible Team		Responsible Person	

Figure 20. For software assets, fields such as Software Title/Version and Software Manufacturer appear.

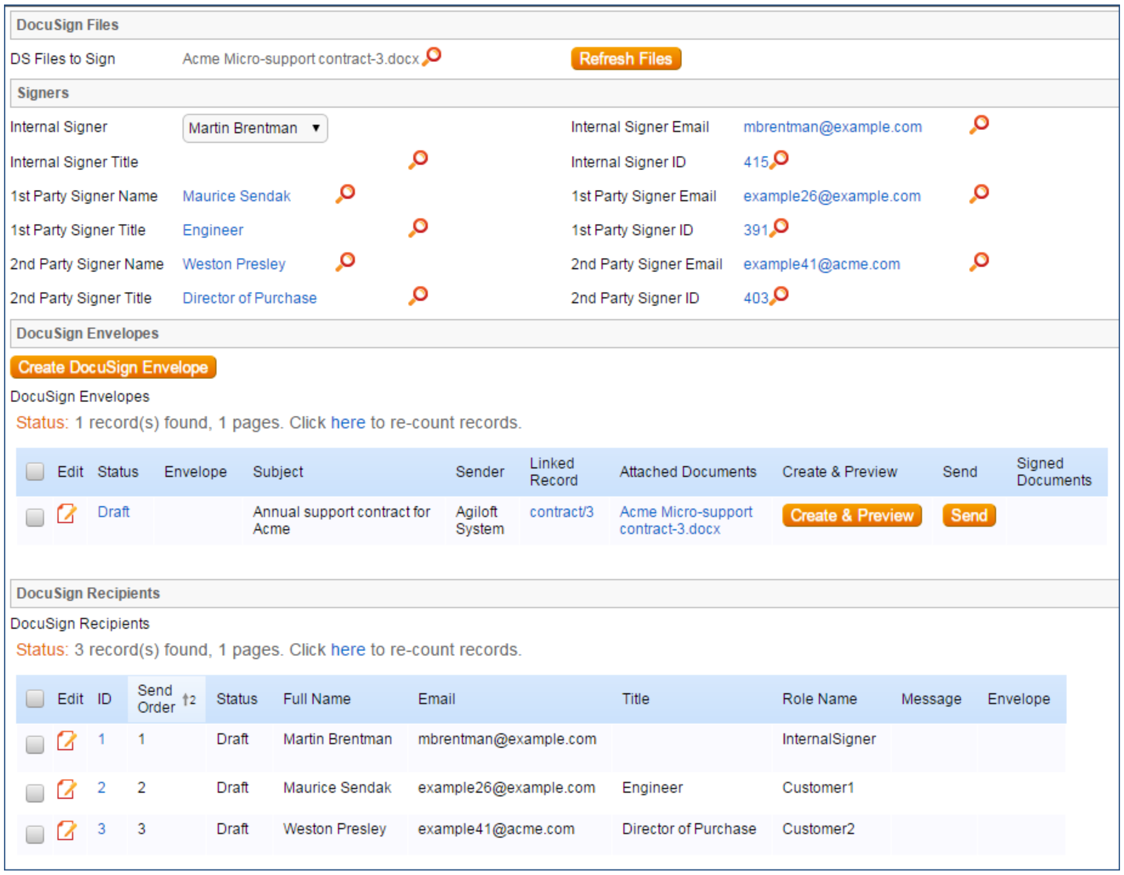
For hardware assets, select “For One or More Assets”. For software assets, choose “For Software Application”. For contracts with no linked assets, use “No Assets” from the Asset Involvement drop-down.

Signing contracts with DocuSign

A Signature tab appears to the right of the Assets tab. This tab contains fields for contract document Signers, and contains all fields related to DocuSign. The DS Files to Sign field holds attached files from the Attachments table that have a To Be eSigned value of Yes. The Refresh Files action button refreshes this field in case changes to the attachments were made in the same session under the contract’s Attachments tab.

Fields under the Signers heading are used to populate the signature page of the contract’s print templates and to add DocuSign tags if the contract uses DocuSign.

Once the contract is ready to sign, use the Create DocuSign Envelope action button to create a DocuSign Envelope record and attach the files held in the DS Files to Sign field. It also creates a DocuSign Recipient record for each signer. These records are then shown in the related tables for DocuSign Envelopes and DocuSign Recipients on the Signature tab. Only users in the Admin and DocuSign Users groups can access the DocuSign fields on the Signature tab. For more information on DocuSign, refer to the ERROR! REFERENCE SOURCE NOT FOUND. section and the detailed DocuSign User Manual at http://www.agiloft.com/documentation/docusign-users-manual.pdf.


DocuSign Files

DS Files to Sign	Acme Micro-support contract-3.docx	Refresh Files	

Signers

Internal Signer	Martin Brentman	Internal Signer Email	mbrentman@example.com
Internal Signer Title		Internal Signer ID	415
1st Party Signer Name	Maurice Sendak	1st Party Signer Email	example26@example.com
1st Party Signer Title	Engineer	1st Party Signer ID	391
2nd Party Signer Name	Weston Presley	2nd Party Signer Email	example41@acme.com
2nd Party Signer Title	Director of Purchase	2nd Party Signer ID	403

DocuSign Envelopes

Create DocuSign Envelope

DocuSign Envelopes

Status: 1 record(s) found, 1 pages. Click here to re-count records.

Edit	Status	Envelope	Subject	Sender	Linked Record	Attached Documents	Create & Preview	Send	Signed Documents
	Draft		Annual support contract for Acme	Agiloft System	contract/3	Acme Micro-support contract-3.docx	Create & Preview	Send	

DocuSign Recipients

DocuSign Recipients

Status: 3 record(s) found, 1 pages. Click here to re-count records.

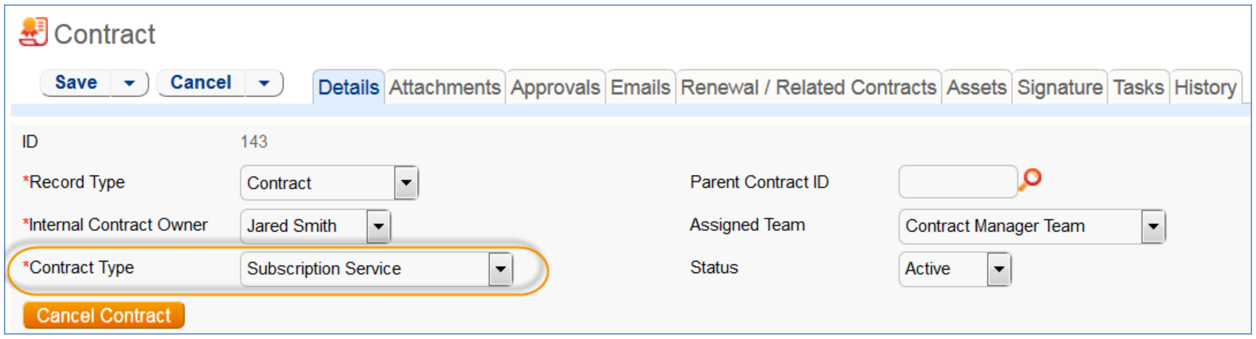
Edit	ID	Send Order ↑2	Status	Full Name	Email	Title	Role Name	Message	Envelope
	1	1	Draft	Martin Brentman	mbrentman@example.com		InternalSigner		
	2	2	Draft	Maurice Sendak	example26@example.com	Engineer	Customer1		
	3	3	Draft	Weston Presley	example41@acme.com	Director of Purchase	Customer2		

Figure 21. Contract signers and DocuSign envelopes are managed on the Signature tab.

Handling Approvals

The order and nature of approvals depends on the Workflow Title selected on the Approvals tab. Selecting a Contract Type in the common area filters the available Workflow Title choices. For information on setting up the individual Workflows, refer to the Approval Workflows Table.

The order and nature of approvals depends on the Workflow Title selected on the Approvals tab. Selecting a Contract Type in the common area filters the available Workflow Title choices. For information on setting up the individual Workflows, refer to the APPROVALS TABLE section.

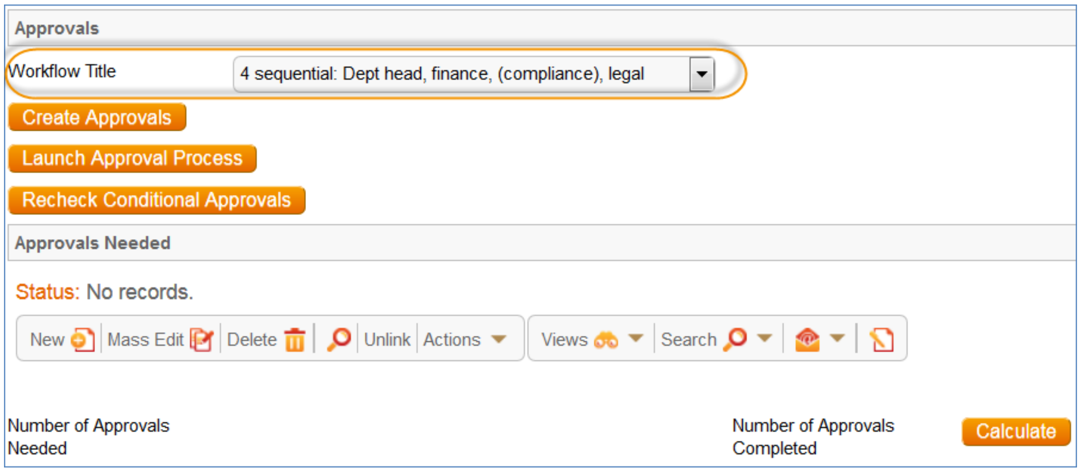

Contract

Save
Cancel
Details | Attachments | Approvals | Emails | Renewal / Related Contracts | Assets | Signature | Tasks | History

ID	143		
*Record Type	Contract	Parent Contract ID	
*Internal Contract Owner	Jared Smith	Assigned Team	Contract Manager Team
*Contract Type	Subscription Service	Status	Active

Figure 22. The Contract Type filters the available workflows.

To submit a contract for approval, first select the correct Workflow Title from the drop-down provided. Click Create Approvals to generate a set of Approval records in a Status of Queued. Once the records have been created, click Launch Approval Process to update the Status of the contract to Pending Approval and update the Status of the approval(s) with the lowest step number to Pending Approval. There may be more than one approval in the lowest step number for parallel approvals.


Approvals

Workflow Title: 4 sequential: Dept head, finance, (compliance), legal

Create Approvals
Launch Approval Process
Recheck Conditional Approvals

Approvals Needed

Status: No records.

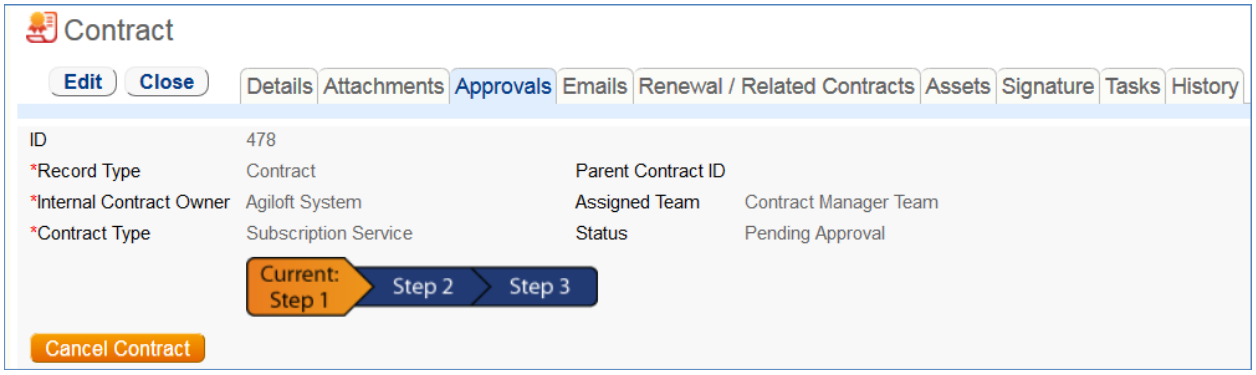
New | Mass Edit | Delete | Unlink | Actions | Views | Search

Number of Approvals Needed

Number of Approvals Completed

Figure 23. Select the Workflow Title to choose the series of approvals needed.

The system automatically notifies the first approver in the sequence. A progress bar also appears in the common area to provide a quick visual reference of the approval process.


Contract

Edit
Close
Details | Attachments | Approvals | Emails | Renewal / Related Contracts | Assets | Signature | Tasks | History

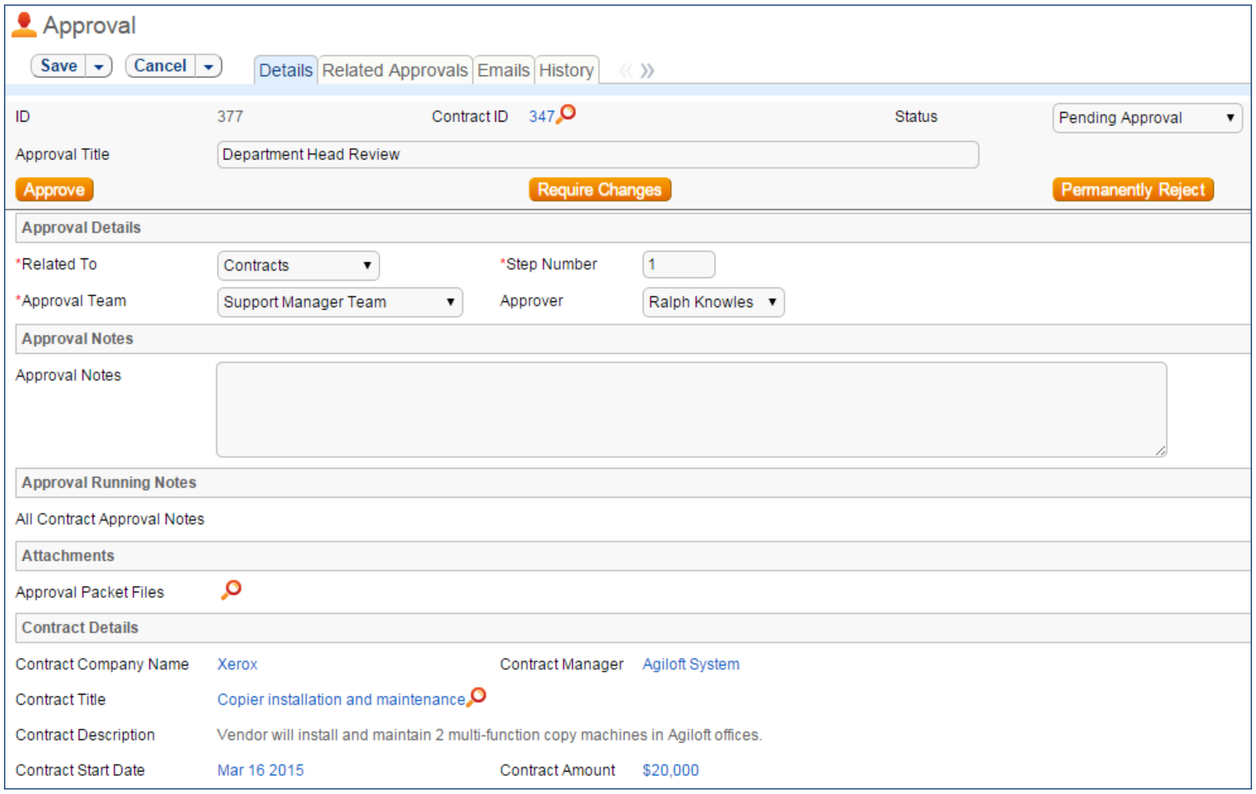
ID	478		
*Record Type	Contract	Parent Contract ID	
*Internal Contract Owner	Agiloft System	Assigned Team	Contract Manager Team
*Contract Type	Subscription Service	Status	Pending Approval

Current: Step 1
Step 2
Step 3
Cancel Contract

Figure 24. The common area shows a progress bar of the approval process.

To advance the contract workflow, an approver will use one of three action buttons to change the Status of the approval record:

- Approve to send the contract to the next approver in the sequence.
- Require Changes to send the contract to the previous approver for changes.
- Permanently Reject if the contract requires significant changes.



ID	377	Contract ID	347	Status	Pending Approval
Approval Title	Department Head Review				

Approval Details

*Related To	Contracts	*Step Number	1
*Approval Team	Support Manager Team	Approver	Ralph Knowles

Approval Notes

Approval Notes

Approval Running Notes

All Contract Approval Notes

Attachments

Approval Packet Files

Contract Details

Contract Company Name	Xerox	Contract Manager	Agiloft System
Contract Title	Copier installation and maintenance		
Contract Description	Vendor will install and maintain 2 multi-function copy machines in Agiloft offices.		
Contract Start Date	Mar 16 2015	Contract Amount	$20,000

Figure 25. Approvers use one of three action buttons to change the status of their approval record.

Both Require Changes and Permanently Reject require the user to enter notes in the Approval Notes field. These notes are appended to the Approval Notes field in the contract record and are also viewable from any other approval record linked to that contract.

As each approval is approved, the system notifies the next approver in the sequence that a contract is pending their approval. A list of all approval records is displayed under the Approvals Needed subsection and automatically updated as approval records are modified.

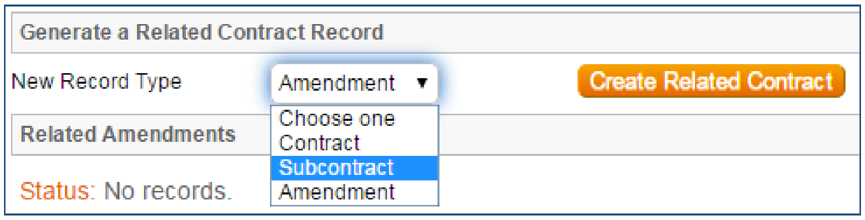
If a user who is not on the current approval team attempts to approve a contract, the system recognizes the error and prevents the user from completing the approval action.

When all the required approvals are received by the system, the contract Status is automatically changed to Approved.

Handling Related Contracts and Renewals

Related contracts are handled in the Renewal / Related Contracts tab of a contract record.

To create a related contract, select the New Record Type of the contract. The choices in the drop down are filtered by the Record Type of the originating contract. These fields are visible only when the contract's Status is Active, Canceled, Expired, or Signed. Clicking Create Related Contract maps relevant information from the current contract into the new one.


Generate a Related Contract Record

New Record Type: Amendment

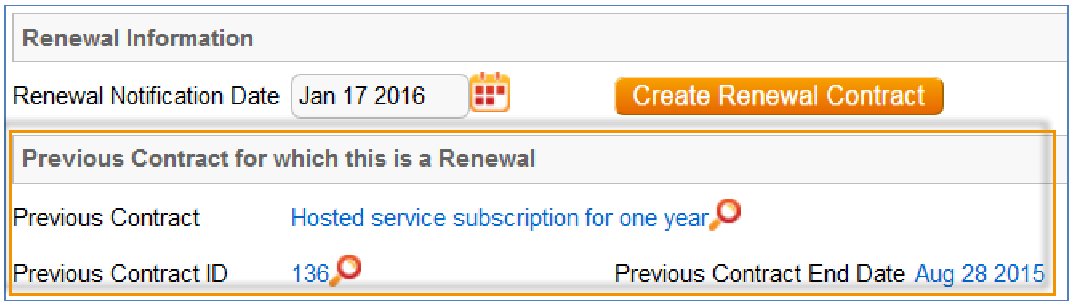
Related Amendments

Status: No records.

Figure 26. To create a related contract, select the New Record Type and click Create Related Contract.

If the new contract is a renewal, click Create Renewal Contract on the Renewal / Related Contracts tab of the parent contract. This button is only visible when the contract's Status is Active, Canceled, Expired, or Signed. A new contract generated in this way can be edited before saving.

If applicable, the system automatically links renewal contracts to any preceding contracts, creating a chain for auditability. Any assets linked to the preceding contract will be linked to the renewal contract as well. This information and related information on the contract process for renewals are stored in the Renewal / Related Contracts tab. Users typically do not enter information into the Renewal Contract and Previous Contract Information fields manually.


Renewal Information

Renewal Notification Date: Jan 17 2016

Previous Contract for which this is a Renewal

Previous Contract: Hosted service subscription for one year

Previous Contract ID: 136

Previous Contract End Date: Aug 28 2015

Figure 27. Renewed contracts show details about the previous contract on the Renewal / Related Contracts tab.

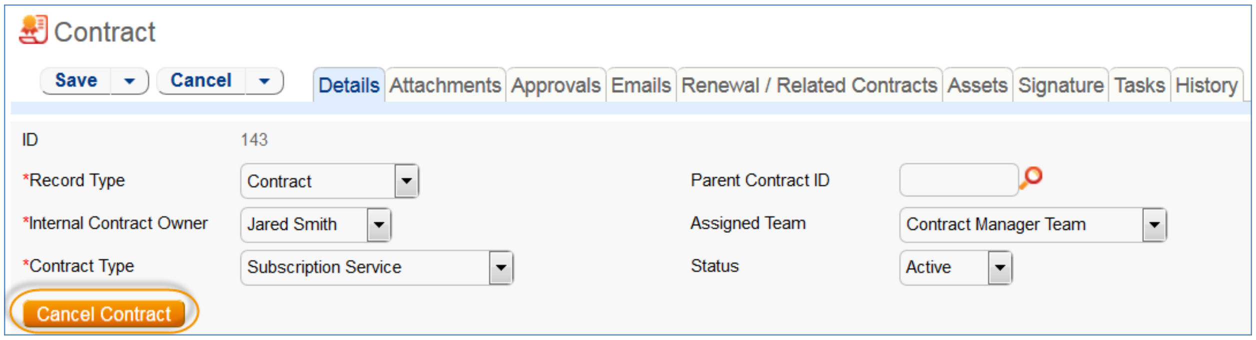
Contract Processing

This section covers the remaining Status changes not mentioned in the explanations above.

Once a contract changes to a Status of Signed, the system automatically updates the contract to a Status of Active when the Contract Start Date occurs.

Similarly, the Status is changed when the Contract End Date occurs. If the contract does not have an associated renewal contract, the Status is automatically set to Expired; if the contract does have a renewal, the Status is set to Renewed. If the contract is auto-renewing, the Contract End Date is increased by the Renewal Term in Months and the contract remains Active.

To cancel a contract, click the Cancel Contract button in the common area. Additionally, users of the Admin group can manually change the contract status to Canceled.


Contract

Save | Cancel

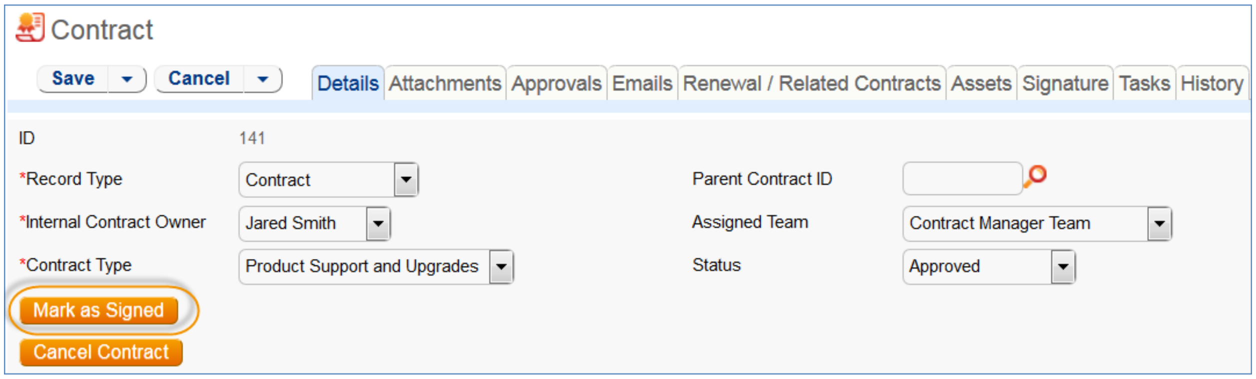
Details | Attachments | Approvals | Emails | Renewal / Related Contracts | Assets | Signature | Tasks | History

ID	143		
*Record Type	Contract	Parent Contract ID	
*Internal Contract Owner	Jared Smith	Assigned Team	Contract Manager Team
*Contract Type	Subscription Service	Status	Active

Cancel Contract

Figure 28. Cancel Contract appears in the contract common area.

If the contract has a Status of Approved, the Mark as Signed button is visible in the common area. This button updates the contract Status to Signed.


Contract

Save | Cancel

Details | Attachments | Approvals | Emails | Renewal / Related Contracts | Assets | Signature | Tasks | History

ID	141		
*Record Type	Contract	Parent Contract ID	
*Internal Contract Owner	Jared Smith	Assigned Team	Contract Manager Team
*Contract Type	Product Support and Upgrades	Status	Approved

Mark as Signed

Cancel Contract

Figure 29. The Mark as Signed button is a quick way to change the status to Signed.

Turning Off Approvals

Contract Management tables have Approval handling set up by default. The associated processes may be turned off in order to use Agiloft as a contract repository. To turn off Approvals, do the following:

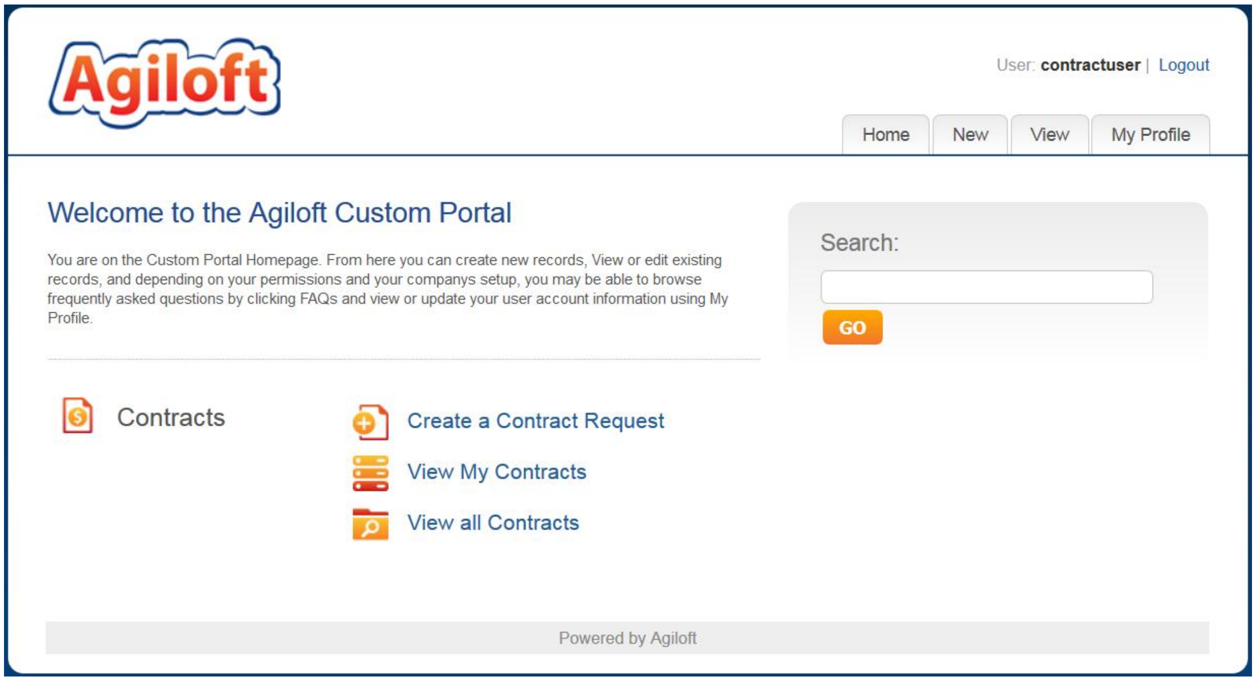
1. Edit field permissions to allow the Contract Manager Group to edit the Contract Status field in their own contracts and in others' contracts.
2. Remove status-changing buttons from the layout: Submit for Approval and Mark as Signed.
3. Remove the Approval tab and related fields from the layout.

For help on configuring table layouts, please refer to the Administrator Reference Manual or online help.

End User Use Case

This section covers the use case for end users in a Contract Management context.

Members of the Contract Creator group are internal employees accessing the system via the end user interface. Contract creators, also called contract requesters, are users who submit requests for contracts but do not work on other users' records. Below is a representative home page for an end user in the Contract Creator group.


User: **contractuser** | Logout

Home | New | View | My Profile

Welcome to the Agiloft Custom Portal

You are on the Custom Portal Homepage. From here you can create new records, View or edit existing records, and depending on your permissions and your companys setup, you may be able to browse frequently asked questions by clicking FAQs and view or update your user account information using My Profile.

Search:

GO

Contracts

Create a Contract Request

View My Contracts

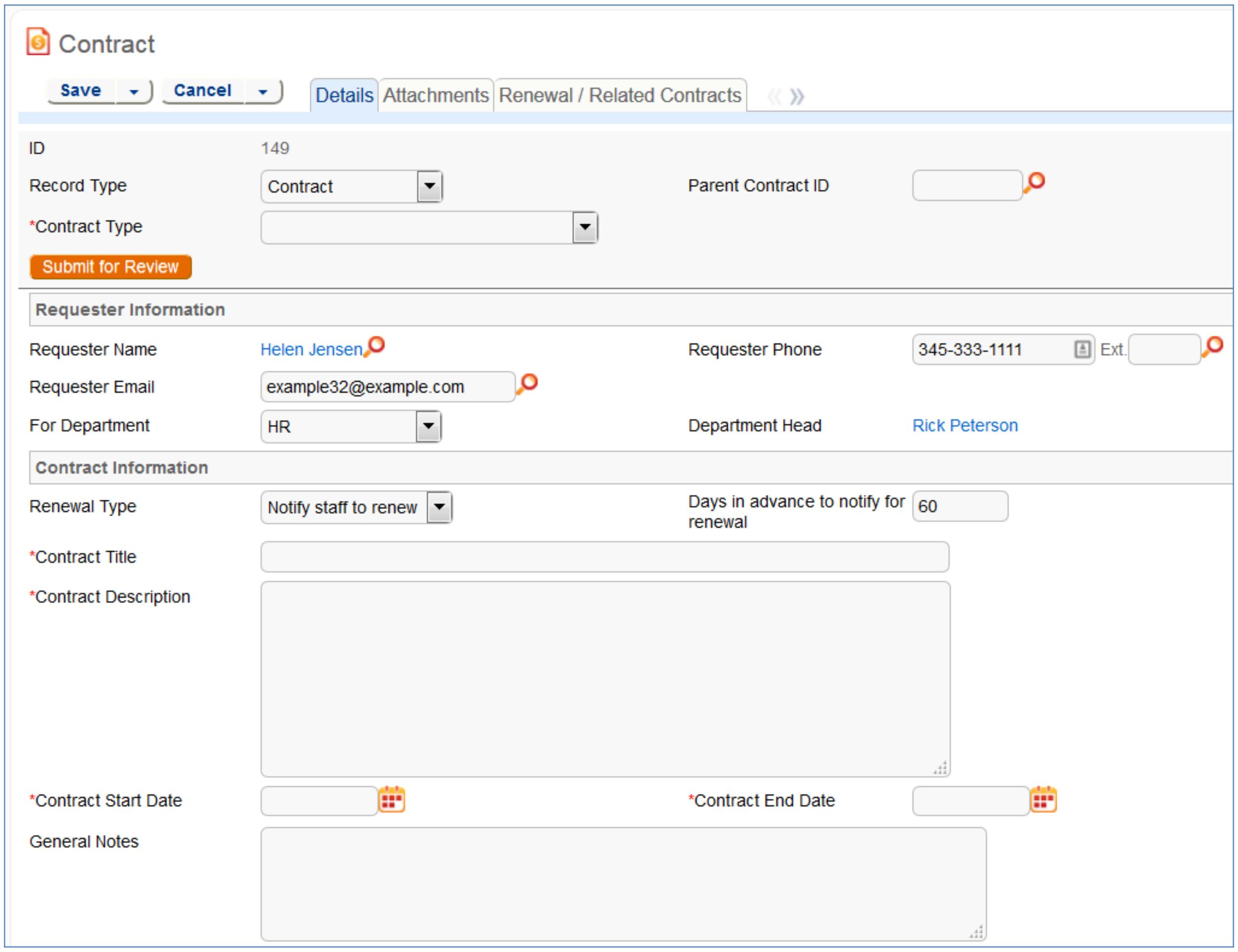
View all Contracts

Powered by Agiloft

Figure 30. Home page for Contract Creators in the default end user interface.

Contract Creation

Users in the Contract Creator group may create contracts by clicking the Submit a Contract link or tab on the home page. A simplified contract form is presented to the end user. Many of the fields are hidden from the layout or restricted by field-level permissions.



Contract

Save | Cancel | Details | Attachments | Renewal / Related Contracts

ID	149		
Record Type	Contract	Parent Contract ID	
*Contract Type			
Submit for Review			

Requester Information

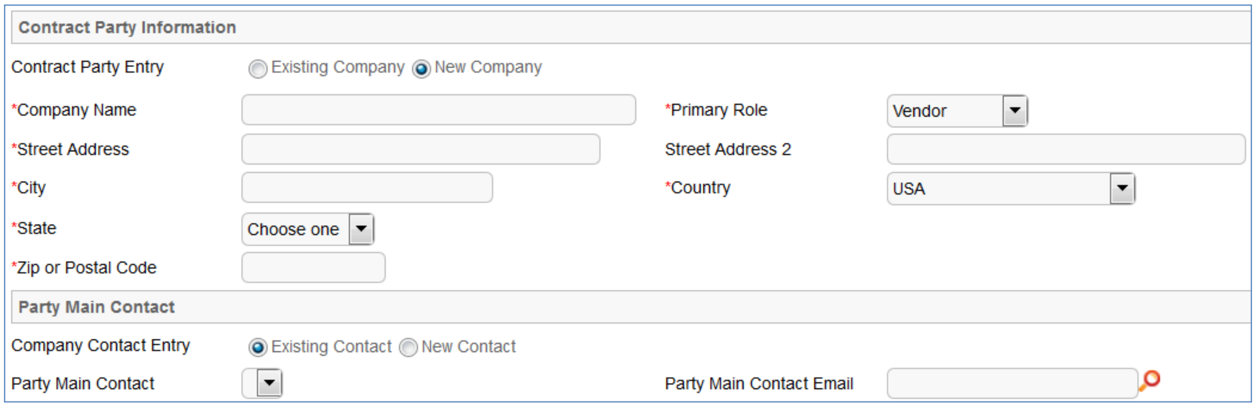
Requester Name	Helen Jensen	Requester Phone	345-333-1111 Ext.
Requester Email	example32@example.com		
For Department	HR	Department Head	Rick Peterson

Contract Information

Renewal Type	Notify staff to renew	Days in advance to notify for renewal	60
*Contract Title			
*Contract Description			
*Contract Start Date		*Contract End Date	
General Notes			

Figure 31. Contract Creators see a simplified contract form in the EUI.

In the Contract Party Information section, users can find and link to an existing company and contact, or create new ones as needed. Select New Company and then enter the name and address information. After the contract is submitted for review, Contract Managers can confirm the new company to finish adding a record. Similarly, end users can suggest a new company contact by filling out the fields in the Party Main Contact section of the form.



Contract Party Information

Contract Party Entry	Existing Company [✓ New Company]		
*Company Name		*Primary Role	Vendor
*Street Address		Street Address 2	
*City		*Country	USA
*State	Choose one		
*Zip or Postal Code			

Party Main Contact

Company Contact Entry	[✓ Existing Contact] New Contact		
Party Main Contact		Party Main Contact Email	

Once the required information is filled in, the contract can be saved for later revisions. The contract requester can also press the Submit for Review button to request approval from a Contract Manager. Contract requesters can be contacted to update the submitted contract, but they are typically no longer involved in the approval process from this point forward.

After the contract requester submits the contract for review, a contract manager decides whether to continue with the approval process or reject the contract request.

Working with Contracts

At any time, the contract requester can view contracts they previously submitted by clicking the My Contracts tab. Contract Creator group members can edit select fields in contracts they own. Certain fields such as Contract Amount, Contract Start Date, and Contract End Date (among others) are not editable by the contract requester if the contract's Status is Pending Approval, Approved, Signed, Active, Renewed, or Expired. This is to prevent changes to currently active or in process contracts.

The contract requester can view all contracts they have permission to see by clicking on the All Contracts home page link or tab.

Ownership

Records in this table are owned by the Contract Requester. Specifically, a record is owned by the user whose ID matches the number in the Requester ID field. By default, the Contract Requester is the user who created the contract record.

Automation

The Contracts table has the following rules set up. Rules which are run based on a schedule (rather than those which are event-triggered) are identified by the prefix "TB", for time-based.

Create: All New Contract Actions

Purpose: This rule handles all update and linking actions upon creating a new record.

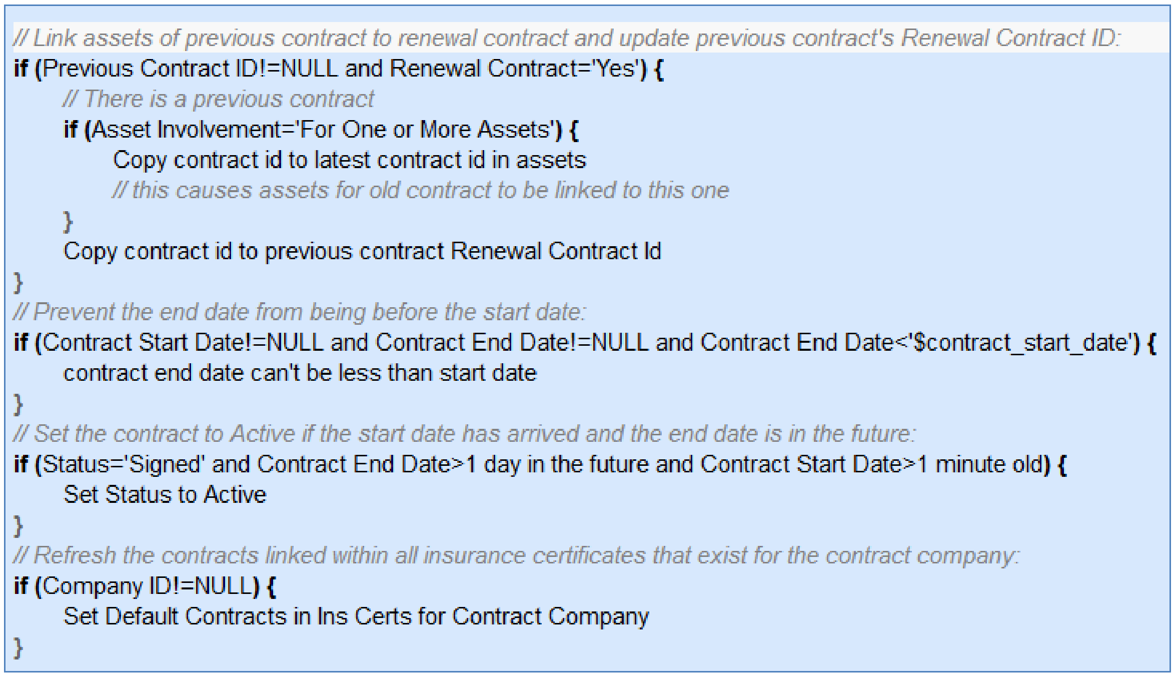
Runs: When created by web or API.

Search Filter: None.

Priority: 1

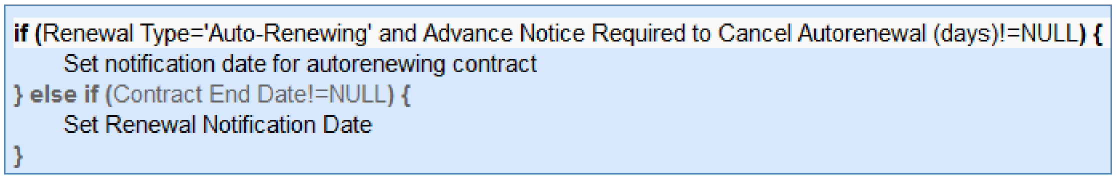
Actions:

I: All New Contract Actions



```
// Link assets of previous contract to renewal contract and update previous contract's Renewal Contract ID:
if (Previous Contract ID!=NULL and Renewal Contract='Yes') {
    // There is a previous contract
    if (Asset Involvement='For One or More Assets') {
        Copy contract id to latest contract id in assets
        // this causes assets for old contract to be linked to this one
    }
    Copy contract id to previous contract Renewal Contract Id
}
// Prevent the end date from being before the start date:
if (Contract Start Date!=NULL and Contract End Date!=NULL and Contract End Date<'$contract_start_date') {
    contract end date can't be less than start date
}
// Set the contract to Active if the start date has arrived and the end date is in the future:
if (Status='Signed' and Contract End Date>1 day in the future and Contract Start Date>1 minute old) {
    Set Status to Active
}
// Refresh the contracts linked within all insurance certificates that exist for the contract company:
if (Company ID!=NULL) {
    Set Default Contracts in Ins Certs for Contract Company
}
```

I: Set Renewal Notification Date based on Renewal Type



```
if (Renewal Type='Auto-Renewing' and Advance Notice Required to Cancel Autorenewal (days)!=NULL) {
    Set notification date for autorenewing contract
} else if (Contract End Date!=NULL) {
    Set Renewal Notification Date
}
```

Create/Edit: If Transitional Contract Files field has value, convert to Attachment

Purpose: This rule handles new attachments uploaded to the Transitional Contract Files field by converting them into Attachment records.

Runs: When edited by email, web or API.

Search Filter: R: Transitional Contract Files field changed

Priority: 15

Actions:

C: Convert to Attachment

Summary: This converts the values in the Transitional Contract Files, Transitional Attachment Title, and Transitional Attachment Type fields into a new Attachment record and links it back to the Contract via the Contract ID.

U: Blank out Transitional Contract Files field

Updated Fields: Transitional Contract Files, Transitional Attachment Title, and Transitional Attachment Type.

Updated Values: Null.

Summary: This resets the Transitional Contract Files, Transitional Attachment Title, and the Transitional Attachment Type fields to empty after the conversion takes place.

Edit: Updates by Party

Purpose: This rule handles notifications for contracts that are updated by an outside party without direct access to the system, as well as the Attachment Type and Attachment Title for inbound files.

Runs: When edited by email or web.

Search Filter: R: Updates by Vendor or Customer Team

Priority: 1

Actions:

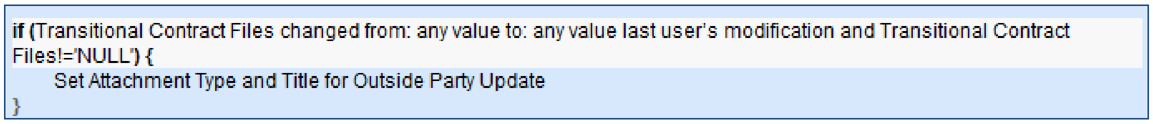
E: Email Owner and Contract Manager Team of Outside Party Update

Sent To: Contract Manager Team, Internal Contract Owner.

Template Name: Outside party has updated contract.

Summary: This email notifies the relevant users that an outside party has updated the contract.

I: Handle Inbound Files



```
if (Transitional Contract Files changed from: any value to: any value last user's modification and Transitional Contract
Files!='NULL') {
      Set Attachment Type and Title for Outside Party Update
}
```

Edit: All edit actions by web or api**Purpose:** This rule handles the processes resulting from changes made by staff or rules.

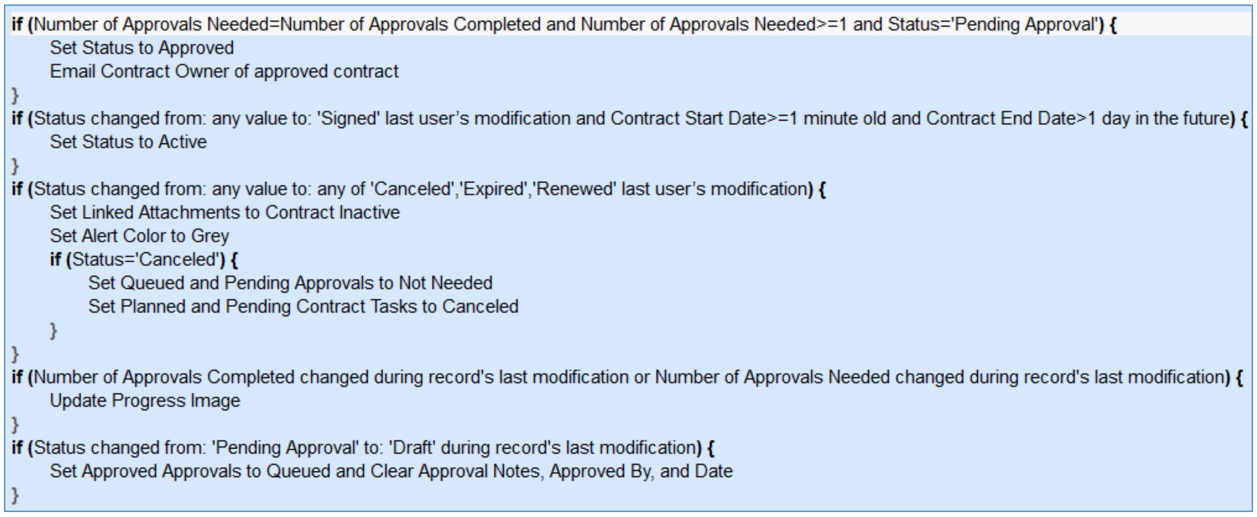
Runs: When edited by web or API.

Search Filter: None.

Priority: 2

Actions:

I: All Edit Actions with API



```
if (Number of Approvals Needed=Number of Approvals Completed and Number of Approvals Needed>=1 and Status='Pending Approval') {
    Set Status to Approved
    Email Contract Owner of approved contract
}
if (Status changed from: any value to: 'Signed' last user's modification and Contract Start Date>=1 minute old and Contract End Date>1 day in the future) {
    Set Status to Active
}
if (Status changed from: any value to: any of 'Canceled','Expired','Renewed' last user's modification) {
    Set Linked Attachments to Contract Inactive
    Set Alert Color to Grey
    if (Status='Canceled') {
        Set Queued and Pending Approvals to Not Needed
        Set Planned and Pending Contract Tasks to Canceled
    }
}
if (Number of Approvals Completed changed during record's last modification or Number of Approvals Needed changed during record's last modification) {
    Update Progress Image
}
if (Status changed from: 'Pending Approval' to: 'Draft' during record's last modification) {
    Set Approved Approvals to Queued and Clear Approval Notes, Approved By, and Date
}
```

Edit: Update Renewal Notification Date if underlying fields change (web, API)

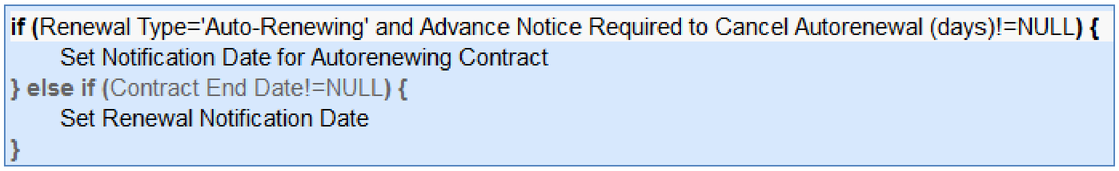
Purpose: This rule updates the Renewal Notification Date if the Contract End Date, Days in Advance to Notify for Renewal, or the Advance Notice Required to Cancel Autorenewal (days) change.

Runs: When edited by web.

Search Filter: R: End Date, Days in Advance, Advanced Notice to Cancel, or Renewal Type changed last user mod

Priority: 3

I: Set Renewal Notification Date based on Renewal Type



```
if (Renewal Type='Auto-Renewing' and Advance Notice Required to Cancel Autorenewal (days)!=NULL) {
    Set Notification Date for Autorenewing Contract
} else if (Contract End Date!=NULL) {
    Set Renewal Notification Date
}
```

Edit: Refresh Approvals when Number of Approvals Needed Changes (web, API)

Purpose: This handles the recalculation of next steps and concurrent approvals when additional approvals are added or removed. This rule will update the Lowest Step, Next Step Number, Concurrent Approvals, and Next Approvals fields in all linked approval records. The linked record action uses a chain through the Latest Contract ID to reach the linked approvals and then runs several *update fields* actions, each of which is based on a saved search.

Runs: When edited by web or API.

Search filter: R: Number of Approvals Needed Changed last modification

Priority: 10

L: Run Contract Creation Actions in Approval

Summary: The actions below update several fields in the Approvals table.

Chain: Contract ← Approval through fields: All Contract Approval Notes, Contract Assigned Team, Contract Company Name, Contract Amount, etc.

U: Set Next Step

Updated Fields: Next Step Number

Updated Values: Use the MT: Same contract, higher Step Number, not Not Needed, sorted ascending by Step Number search.

U: Set Concurrent Approvals

Updated Fields: Concurrent Approvals

Updated Values: Use the MT: Approval for same contract with same step number search.

U: Set Lowest Step Number

Updated Fields: Lowest Step Number

Updated Values: Use the LF: Contract ID matches, sorted by Step Number search.

U: Update Next Approvals in Approval Records

Updated Fields: Next Approvals

Updated Values: Use the MT: Same contract - Step Number matches Next Step Number and Status is not Not Needed search.

TB: Notify of upcoming expirations (disabled)

Purpose: This rule controls the email notifications for upcoming expirations of contracts

Runs: Every 2 days at 3 AM.

Search Filter: R: Renewal date is tomorrow and Renewal Type is not one-time

Priority: 5

Action:

E: Email Contract Owner about Renewal Date

Sent To: User in the Internal Contract Owner field

Template Name: Contract renewal notification date has arrived.

Summary: Reminds the Contract Manager of the impending renewal date of a contract.

TB: Daily Check for Start Date (disabled)

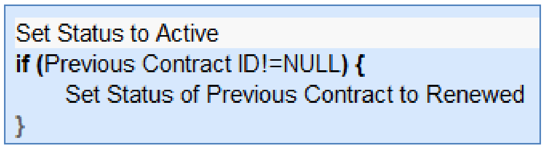
Purpose: This rule checks for contracts whose start date has arrived and changes the Status accordingly.

Runs: Every 2 days at 4 AM.

Search Filter: R: Status is signed and Start Date is today or earlier

Priority: 6

I: Actions When Start Date Arrives



```
Set Status to Active
if (Previous Contract ID!=NULL) {
    Set Status of Previous Contract to Renewed
}
```

TB: Daily Check for Expiration Date (disabled)

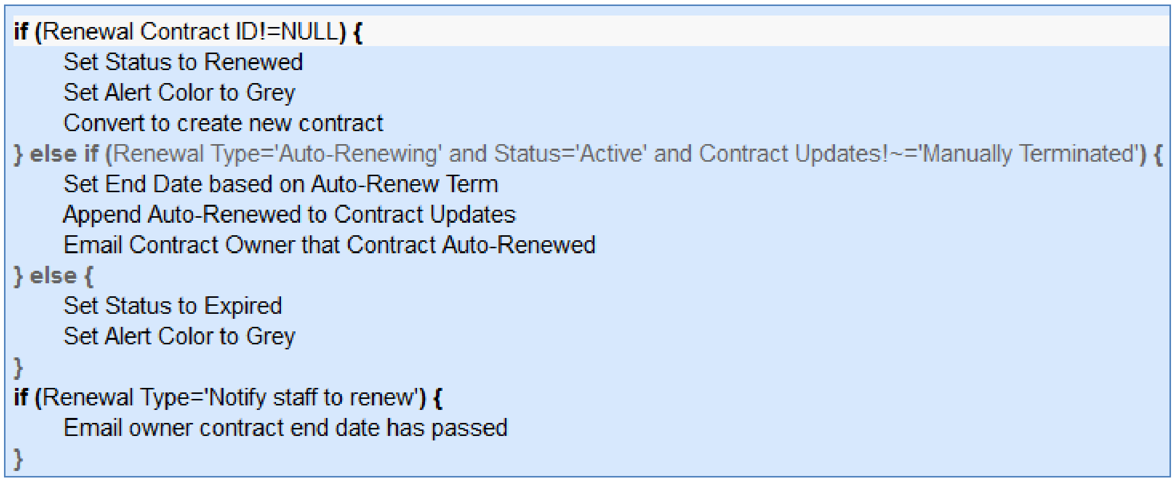
Purpose: This rule checks for contracts whose End Date has arrived and takes the appropriate actions.

Runs: Every 2 days at 2 AM

Search Filter: R: Active contract expired today

Priority: 5

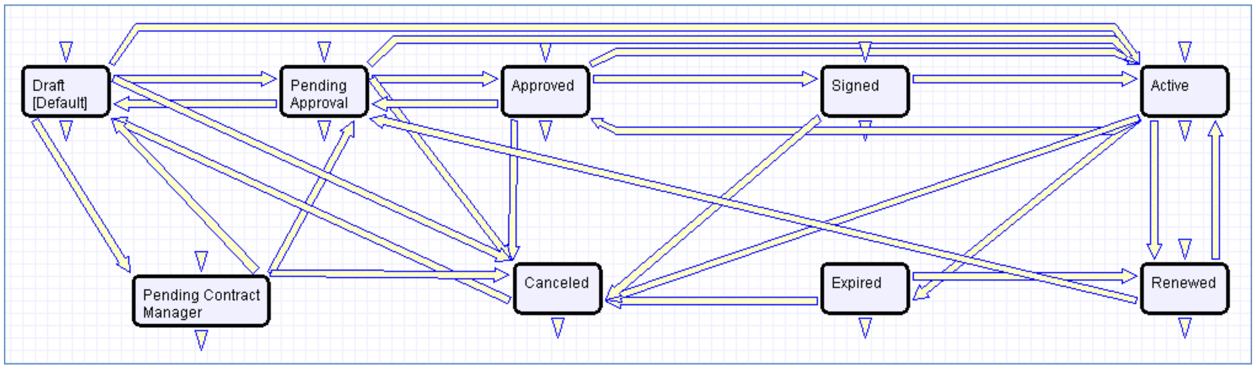
I: Handle Expired Contract



```
if (Renewal Contract ID!=NULL) {
    Set Status to Renewed
    Set Alert Color to Grey
    Convert to create new contract
} else if (Renewal Type='Auto-Renewing' and Status='Active' and Contract Updates!~='Manually Terminated') {
    Set End Date based on Auto-Renew Term
    Append Auto-Renewed to Contract Updates
    Email Contract Owner that Contract Auto-Renewed
} else {
    Set Status to Expired
    Set Alert Color to Grey
}
if (Renewal Type='Notify staff to renew') {
    Email owner contract end date has passed
}
```

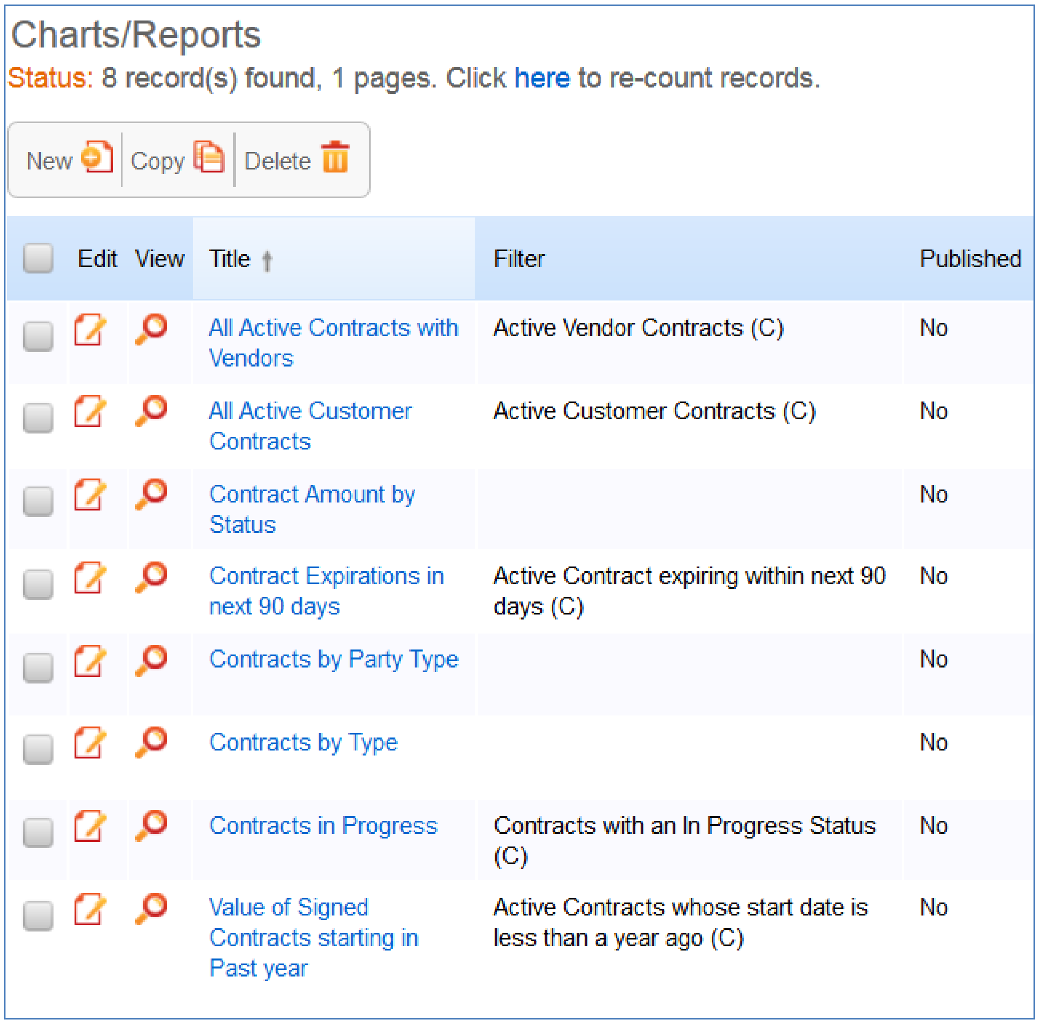
Workflow

The Contracts table has the following default workflow:



Reports

The Contracts table contains the following default Charts/Reports:


Charts/Reports

Status: 8 record(s) found, 1 pages. Click here to re-count records.

Edit	View	Title	Filter	Published
		All Active Contracts with Vendors	Active Vendor Contracts (C)	No
		All Active Customer Contracts	Active Customer Contracts (C)	No
		Contract Amount by Status		No
		Contract Expirations in next 90 days	Active Contract expiring within next 90 days (C)	No
		Contracts by Party Type		No
		Contracts by Type		No
		Contracts in Progress	Contracts with an In Progress Status (C)	No
		Value of Signed Contracts starting in Past year	Active Contracts whose start date is less than a year ago (C)	No

Insurance Certificates Table

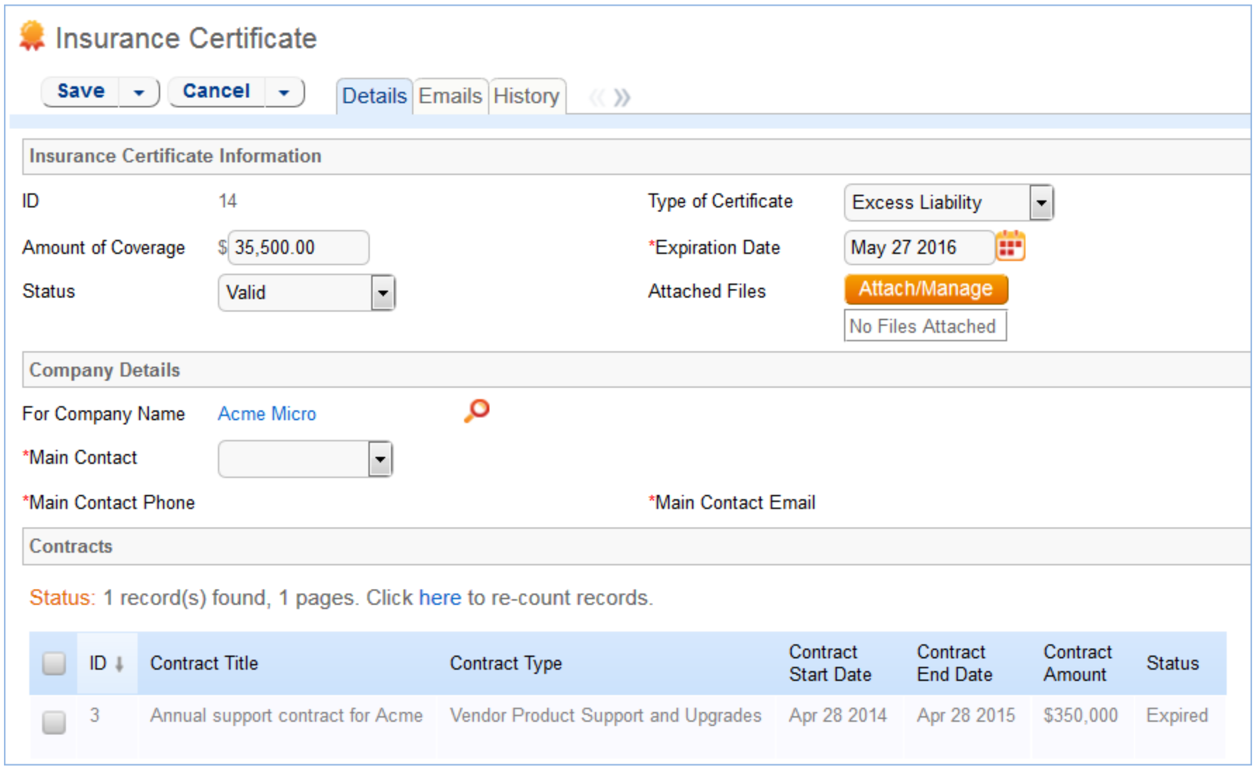
This table holds insurance certificates linked as attachments to Companies and Contracts. Each record in the table represents one insurance certificate linked to a unique vendor company, but may be linked to one or more contracts.

Use Case

Insurance certificates can be created in one of three ways:

- From a Company record, using the related table on the Related Records tab (preferred)
- From a particular Contract record, using the related table on the Attachments tab
- From the Insurance Certificate table **action bar**

New Insurance Certificates are created with a default Status of Valid. They may also have a Status of Expired or Contract Inactive. The Expiration Date and Main Contact are required fields. The Main Contact is the person at the vendor company from whom a replacement certificate can be requested. You can customize the Type of Certificate choice list as needed; the default options include Auto, Worker’s Comp, Excess Liability, General Liability, etc.



Insurance Certificate Information

ID	14	Type of Certificate	Excess Liability
Amount of Coverage	$ 35,500.00	*Expiration Date	May 27 2016
Status	Valid	Attached Files	No Files Attached

Company Details

For Company Name	Acme Micro		
*Main Contact			
*Main Contact Phone		*Main Contact Email	

Contracts

Status: 1 record(s) found, 1 pages. Click here to re-count records.

ID	Contract Title	Contract Type	Contract Start Date	Contract End Date	Contract Amount	Status
3	Annual support contract for Acme	Vendor Product Support and Upgrades	Apr 28 2014	Apr 28 2015	$350,000	Expired

Figure 32. The default tab of an Insurance Certificate.

Insurance certificates are primarily linked to vendor companies in the Company table. Once you add an insurance certificate to a vendor record, it can be linked to any contracts with that vendor.

Insurance certificate records contain information about related contracts for reference, held in the Contracts field. A time based rule, disabled by default, is designed to update the Contracts linked field with all contracts for that vendor that are currently pending or active. As a result, insurance certificates are automatically displayed within those contracts.

An Insurance Certificate Owner is defined in the company record just above the Insurance Certificates *related table*. The Insurance Certificate Owner is notified fourteen days before one or more certificates is due to expire.

When the Expiration Date arrives, the certificate's Status is updated in one of two ways:

- If any of the associated contracts have a Status of Active, the system updates the certificate's Status to Expired and an email notification is sent to all the Contract Owners about the expired insurance certificate.
- If none of the related contracts are Active, the system updates the certificate's Status to Contract Inactive.

Ownership

Insurance Certificate records are owned by their creator. Specifically, an Insurance Certificate record is owned by the user whose Login matches the Creator Login field.

Automation

The Insurance Certificates table has the following rules set up. A rule running on the Company table sends notifications about upcoming expiring certificates.

TB: Update Contracts (daily) (disabled)

Purpose: This rule updates the linked field of Contracts to find all contracts related to the insurance certificate.

Runs: Daily at 1:00 AM

Search Filter: R: Status is Valid

Priority: 0

Action:

U: Set Default Contracts

Updated Field: Contracts

Updated Value: Use the Insurance Certificate to Contract: Status is greater than or equal to Active and Company matches search, and choose all records***.***

TB: Handle Insurance Certificate Expiration (disabled)

Purpose: This rule checks for expired insurance certificates and then updates the certificate's Status. If the related contract is still valid, the Contract Owner is notified of the expired certificate.

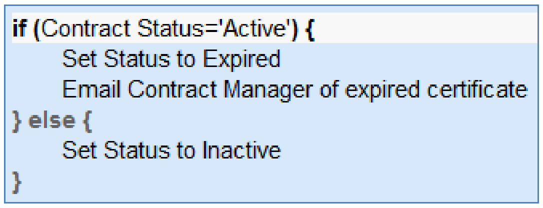
Runs: Every two days at 1:30 AM

Search Filter: Valid Certificates expiring TODAY

Priority: 1

Action:

I: Handles Certs Expiring TODAY



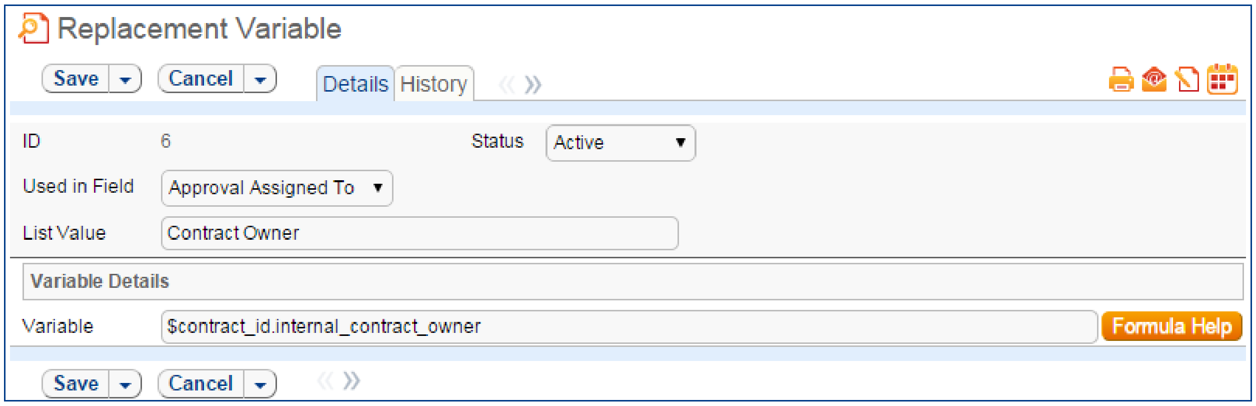
```
if (Contract Status='Active') {
    Set Status to Expired
    Email Contract Manager of expired certificate
} else {
    Set Status to Inactive
}
```

Ownership

Records in this table are owned by the creator of the certificate. Specifically, a record is owned by the user whose Login matches the login in the Creator Login field.

Replacement Variables Table

The Replacement Variables table is a background table that can be used to support any of the process tables, such as Approvals, Contracts, Projects, or Tasks. Replacement Variable records store a **variable chain** used to pull in values from a linked field relationship. Generally, these variables are copied into template records (e.g., in the Approval Templates table) and then used to dynamically assign records or set other field values when a process table record is generated from a template.



ID	6	Status	Active
Used in Field	Approval Assigned To		
List Value	Contract Owner		

Variable Details

Variable	$contract_id.internal_contract_owner

Figure 33. Sample Replacement Variable record.

Use Case

Replacement Variables are used by Approval Templates to generate Approval records for contracts, change requests, documents, or service requests.

To create a Replacement Variable that will be used in Approvals, set Used In Field to the value "Approval Assigned To". The value in the List Value field will shows up in the Assign To drop-down in an Approval Template. Last, create a **variable chain** from the Approvals table to the appropriate field variable. For example, `$contact_id.requester_name` finds the name of the Requester in the linked Contract record, while `$contract_id.requester_id.manager_name` finds the Requester's manager.

Caution: *The Variable should be a variable chain that starts from the Approval record—not from the Approval Template!*

Admin Note: *The values "Task Assigned To" and "Other" do not link to any automation in the default setup, but they can be deployed and used with minor modifications to Task Templates or other template table.*

To use a Replacement Variable in an Approval Template, staff users can create an Approval Template and set Assign Approval Based On to the value "Person from Contract". The user can then select a value for the Assign To dropdown, which contains a link to all Replacement Variable records for which Used in Field is "Approval Assigned To". When an Approval is generated from that Approval Template (see the

Documents Table

Purpose: This table can be used to manage the creation and publication of documents of various types, from marketing collateral to employee procedure manuals. A light-weight parallel publication approval process is included.

Examples of documents that may be covered: FAQs, official memos, published company policies, user manuals, newsletters, press releases, and so on. The table may be used to manage documents that are accessed only through Agiloft (though the records in this table) or documents that are published at the company website, intranet or printed and distributed. Access to the documents is controlled through permissions based on a choice field within the record.

Use Case

End User Record Submission

An end user belonging to the Document Creator team can create a document through the **EUI**. Action buttons will be provided to the end user to move the document through the workflow.

When a user submits a document for review, the contact information fields are automatically populated based on the details in his/her record - including the user's department. A direct link to the Department and the Department Manager will be auto-populated based on the submitter's department.

The record will be created in a default status of Draft. After supplying the required information and uploading a document, the user will click the Submit for Review button to begin the review process. The status of the record will be updated to Pending Review. If the user is not prepared to submit the record immediately, then can save the record and make further updates.

Technician Record Submission

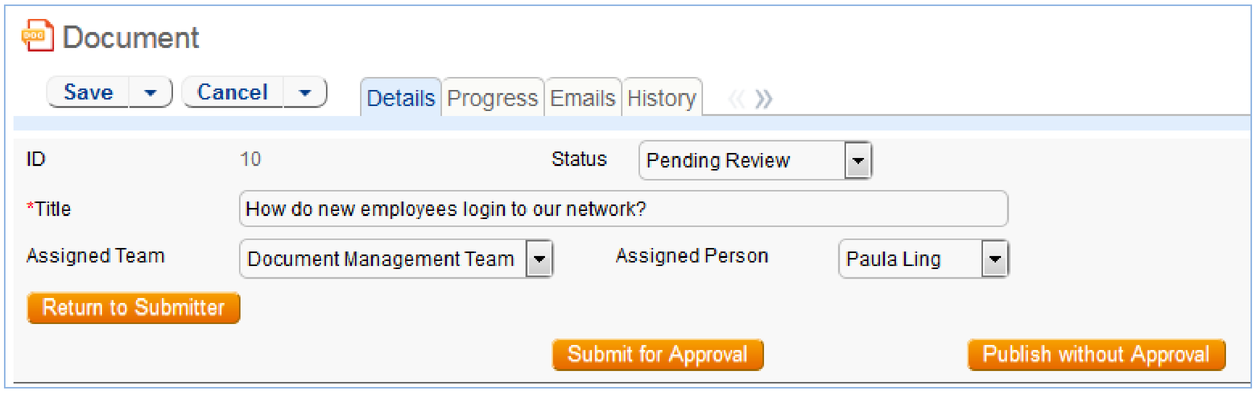
Staff users in the Document Management, Admin and Document Creator groups can submit documents. Support Staff members may also convert to create a new FAQ Document from an Incident or Service Request. Only Admins and Document Managers can update the status of a Document record manually. All other groups will use Action Buttons to move the document through the workflow.

The record will be created in a default status of Draft. After supplying the required information and uploading a document, the user will click the Submit for Review button to begin the review process. If the user is not prepared to submit the record immediately, they can save the record and make further updates.

Processing of Records

When a document record is submitted, the Document Management Team is assigned by default and notified. A Document Manager reviews the document for content and formatting, and to determine if the document requires additional review.

If there are any issues with the initial document, the Document Manager clicks Return to Submitter to send the document back to Draft status. A **rule** then notifies the Submitter that the document requires revision. The user makes appropriate updates to the document and record, then clicks Submit for Approval again.



ID	10	Status	Pending Review
*Title	How do new employees login to our network?		
Assigned Team	Document Management Team	Assigned Person	Paula Ling

Figure 34. The common area holds action buttons to control the document workflow.

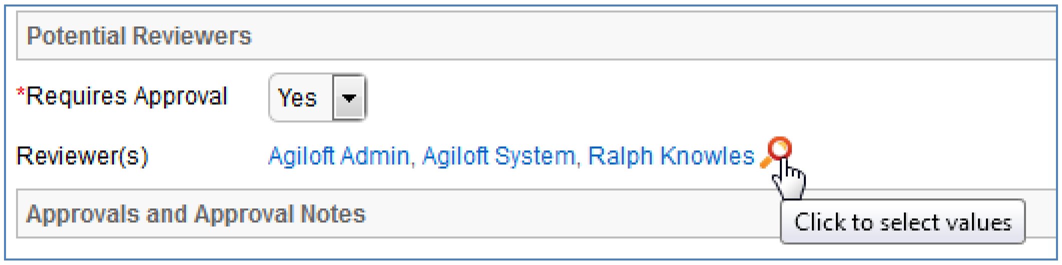
Handling Approvals

The Requires Approval field on the Progress tab determines whether approvals are needed before publication.

If the document does not require approval by document Reviewer(s), the Document Manager clicks the Publish without Approval action button. The Submitter is notified that the document has been published. If the Requires Approval field is set to Yes, a validation rule notifies the Document Manager that approval is required if the Document Manager attempts to publish without approvals.

Selecting Approvers and Creating Approvals

If the document requires approvals by Reviewer(s), the Document Manager will select the appropriate reviewers by adding reviewer names under the Potential Reviewers heading on the Progress tab. This field is a *link to a single field* (Full Name) with multiple values enabled in the Employee table, displayed as a multiple value box with a popup selection list and filtered to people on the Document Reviewers Team.


Potential Reviewers

*Requires Approval: Yes

Reviewer(s): Agiloft Admin, Agiloft System, Ralph Knowles

Approvals and Approval Notes

Figure 35. Use the lookup icon to select Reviewers.

After selecting the appropriate Reviewer(s), the Document Manager clicks Submit for Approval, or manually changes the Status of the record to Pending Approval. A validation rule checks to see if the Document Manager has actually added reviewers to the form.

When the record is saved, an approval record is created for each reviewer by a conversion process from the reviewer's Employee record. A *linked record action* updates the Last Document ID field in the Employee record.

When the system detects a change in the Last Document ID field it performs a *data conversion action,* using the employee record to create an Approval record. The approval record is linked to the document record through the mapping of Last Document ID field to the linked Document set in the approval record. Additional reviewers can be added during the approval process; automation rules and actions will prevent the creation of multiple approval records for existing approvers.

The default Approval Status for new approval records is Pending Approval. When an approval record is created, the reviewer is automatically notified that they have a document to review and approve. In the Approval record, the **linked field** set from the document record includes a hyperlink to the document using the view only source field display option for the Document(s) field under the Attachment heading. The reviewer can click the link to open the document, allowing the reviewer to mark up the document which can then be uploaded to the approval record (reviewers are not able to upload the document to the source record). All updates to the document will be done by the Submitter, the Document Manager or an admin.

Rejecting Documents

The reviewer may either approve or reject the document and provide Approval Notes. If the approval record is rejected, the reviewer must provide Approval Notes explaining why the document is rejected. A validation rule reminds the reviewer that they must provide rejection notes if a reviewer attempts to reject the approval without providing notes.

When any of the approvals are rejected, a rule in the Document table detects this via a *calculation on multiple linked records* field that keeps track of the number of rejections. When this field is updated by rejections, the rule sets the document's Status to Draft, emails the Submitter that the document requires an update, and sets a trigger field named Document Rejected, Requires Update in each of the linked approvals to Yes.

Setting the Document Rejected field to Yes in the approvals then triggers the rule below that sets the Approval Status field in all of the linked approvals to Requires Reapproval, unless that Approval already has a value of Rejected.

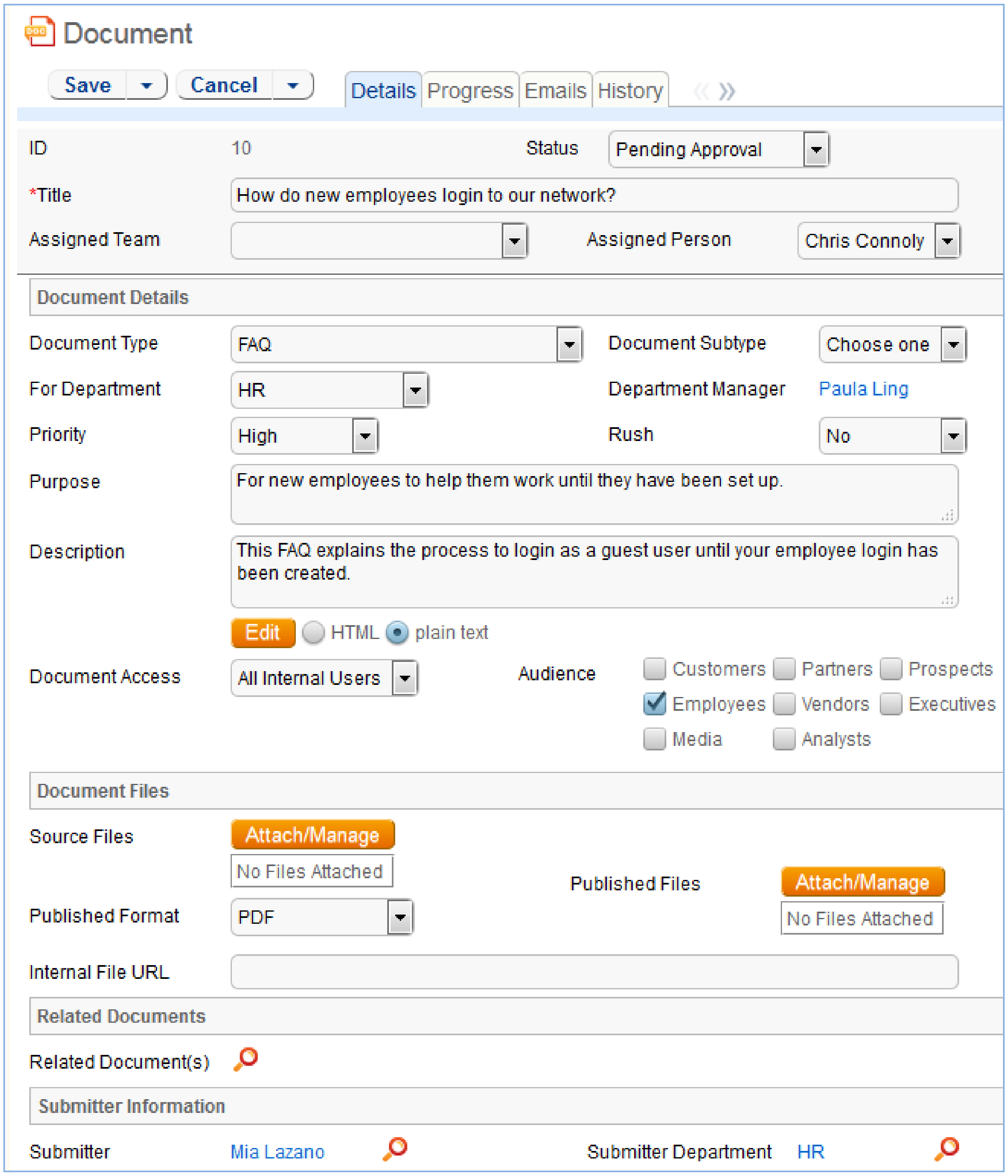
Since any rejected Approvals set the Status of the document back to Draft, the process essentially starts over. The Submitter can review the comments made by the reviewers and see any red-lined documents that have been attached to the approval records. Then the Submitter edits the document and presses the Submit for Review button again to change the Status of the Document to Pending Review and notify the Document Manager. The Document Manager can then keep the same Reviewer(s) or change them and Submit for Approval again. If a document is rejected by an approver and then accepted again (and the Status therefore changed from Pending Approval to Draft and back to Pending Approval), the status of all of the linked approvals is restored to Pending Approval, and all Reviewer(s) are notified by email.

Publishing Documents

Once all approvals are completed and the value of the field Total Number Still Awaiting Approval is 0, a rule sets the Status of the document to Ready for Publication. Both the Submitter and Document Manager are emailed that the document is ready for publication.

The Document Manager can then attach the final approved document to the Published Files field and update the Status to Published. A validation rule checks again to see if there are still any pending approvals.

Once the Document is published, an email notification is sent to the Submitter.



ID	10	Status	Pending Approval
*Title	How do new employees login to our network?		
Assigned Team		Assigned Person	Chris Connoly

Document Details

Document Type	FAQ	Document Subtype	Choose one
For Department	HR	Department Manager	Paula Ling
Priority	High	Rush	No
Purpose	For new employees to help them work until they have been set up.		
Description	This FAQ explains the process to login as a guest user until your employee login has been created.		
	Edit HTML [✓ plain text]		
Document Access	All Internal Users	Audience	Customers, Partners, Prospects, [✓ Employees], Vendors, Executives, Media, Analysts

Document Files

Source Files	Attach/Manage No Files Attached	Published Files	Attach/Manage No Files Attached
Published Format	PDF		
Internal File URL			

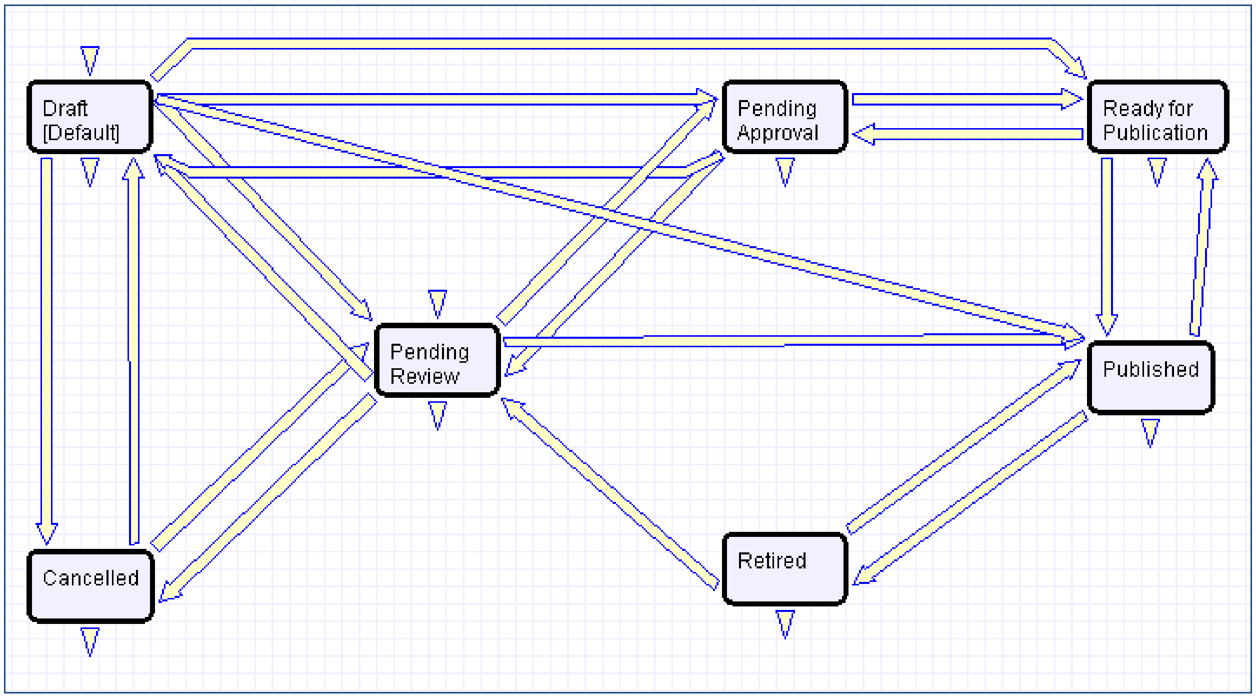
Related Documents

Related Document(s)

Submitter Information

Submitter	Mia Lazano	Submitter Department	HR

Workflow



Project Management Tables

Projects Table

Purpose: This table holds records for project management activities. It is currently optimized for companies providing consulting services to their clients, and allows them to manage their billable and unbillable hours, work authorizations, and overall project status. It also accommodates internal project management instead.

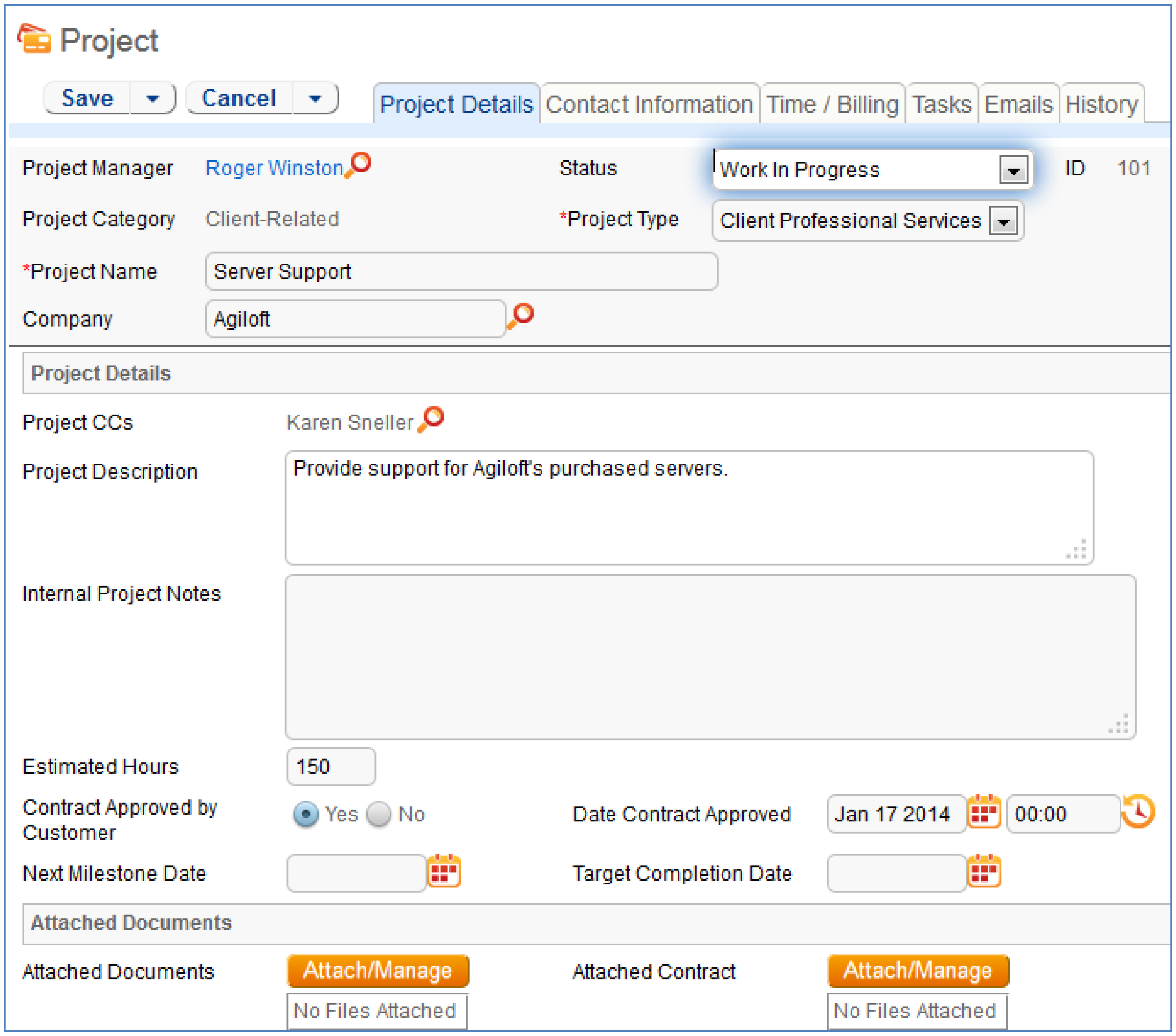
Use Case

Members of the Admin or Project Manager groups may manually create Project records. Projects are creatable only in the “Planned”, “Assigned”, "Work in Progress" and "Awaiting Customer Feedback" states.

Only members of the Product Manager and admin Groups may edit others’ Project records, but Support Staff (Base ServiceDesk group) may view their own Project records. The user who created the Project is automatically set as the Project Manager, and will get email notifications pertaining to the Project’s status, such as when all tasks are completed or the hours spent on the project have exceeded what was authorized. Project CCs can be specified who will also receive notice when the Project is completed. The information on the “Contact Information” tab will be filled out with the information of the Project Manager’s Manager, provided by the Employee record of the Project Manager.

Project records are divided into two broad categories: Internal and Client-Related. These categories are further divided by type. The tasks that are automatically generated by a project depend on the type selected. For each Project type there may be task workflows or user selected task templates defined.

These will result in certain tasks being automatically generated. The exact selection of tasks that will be generated can be specified on the “Tasks” tab of the Project record. Ad-hoc tasks can also be created using an action button on that tab. Additional fields are visible when a Project’s Category is “Client-Related”. These fields hold information about customer contracts, contact details, and authorized hours.



Project Manager	Roger Winston	Status	Work In Progress	ID	101
Project Category	Client-Related	*Project Type	Client Professional Services		
*Project Name	Server Support				
Company	Agiloft				

Project Details

Project CCs	Karen Sneller		
Project Description	Provide support for Agiloft's purchased servers.		
Internal Project Notes			
Estimated Hours	150		
Contract Approved by Customer	[✓ Yes] No	Date Contract Approved	Jan 17 2014 00:00
Next Milestone Date		Target Completion Date	

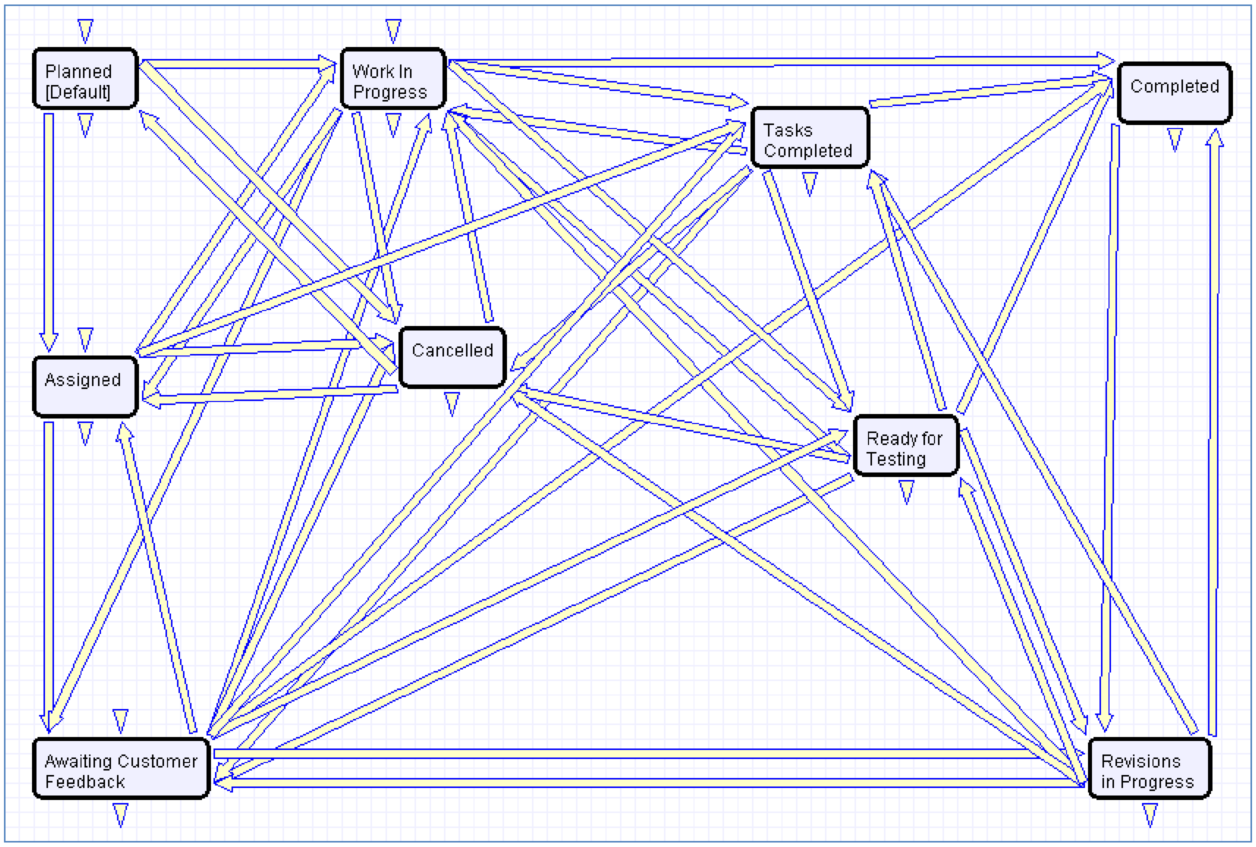
Attached Documents

Attached Documents	No Files Attached	Attached Contract	No Files Attached

Main tab of Project record.

Time spent on the project is tracked on the “Time / Billing” tab. It includes a small form to enter time spent and a related table of time entries for this project. Time entries can be searched for in the related table and linked to the project manually.

Workflow



Default workflow for the Projects table

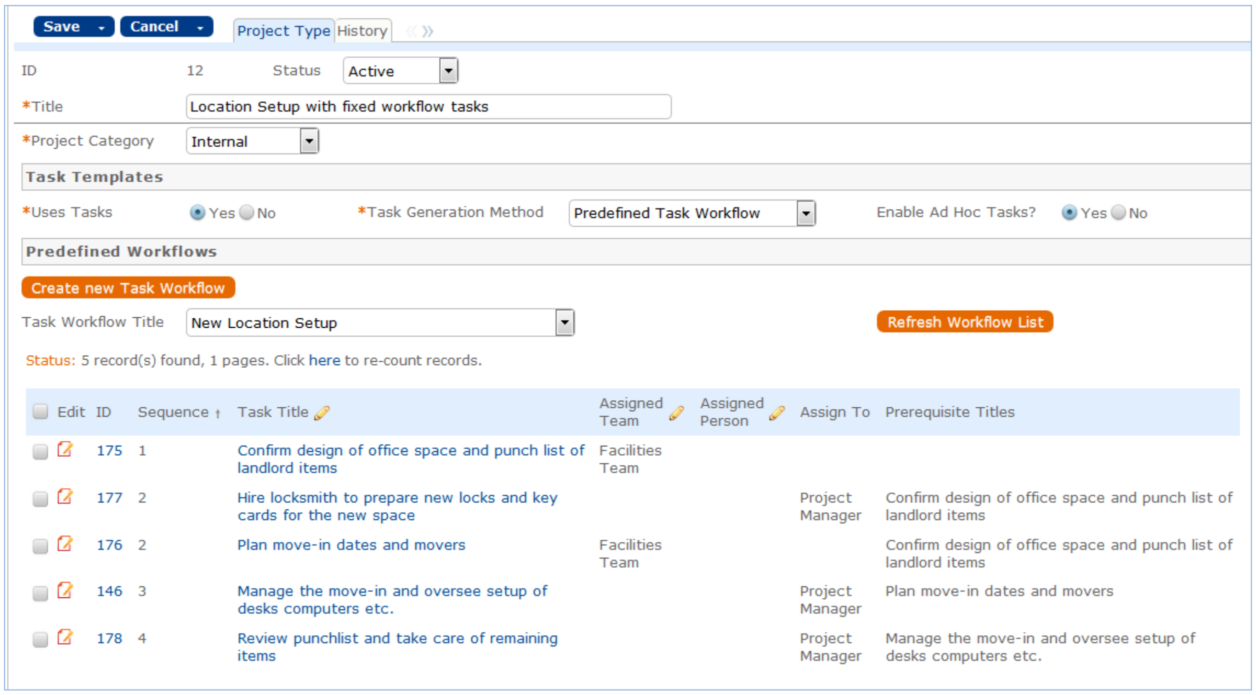
Ownership

Records in this table are “owned” by the Employee designated as the Project Manager.

Project Types Table

Purpose: This is a background table that holds the Project Types referenced by the Projects table. It allows the creation of new Project Types by Project Managers and other users without Admin group privileges.

Project Types define the task generation method, if any, and if task workflows will be used, they are defined in the project type record:



ID	12	Status	Active
*Title	Location Setup with fixed workflow tasks		
*Project Category	Internal		

Task Templates

*Uses Tasks [✓ Yes] No *Task Generation Method: Predefined Task Workflow Enable Ad Hoc Tasks? [✓ Yes] No

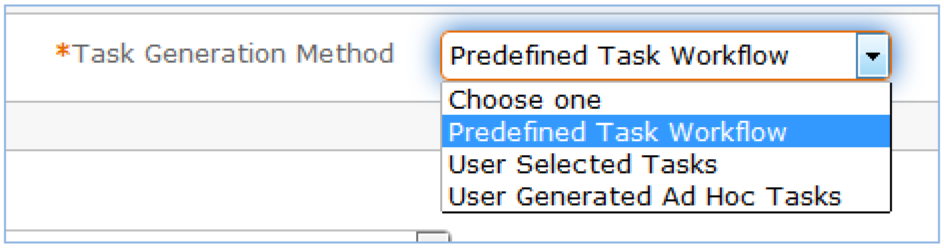
Predefined Workflows

Task Workflow Title: New Location Setup

Status: 5 record(s) found, 1 pages. Click here to re-count records.

Edit	ID	Sequence	Task Title	Assigned Team	Assigned Person	Assign To	Prerequisite Titles
	175	1	Confirm design of office space and punch list of landlord items	Facilities Team			
	177	2	Hire locksmith to prepare new locks and key cards for the new space			Project Manager	Confirm design of office space and punch list of landlord items
	176	2	Plan move-in dates and movers	Facilities Team			Confirm design of office space and punch list of landlord items
	146	3	Manage the move-in and oversee setup of desks computers etc.			Project Manager	Plan move-in dates and movers
	178	4	Review punchlist and take care of remaining items			Project Manager	Manage the move-in and oversee setup of desks computers etc.

In addition to Predefined Task Workflows, project types may use Users Selected Tasks or User Generated Ad Hoc Tasks as the task generation methods.



The differences between these methods is discussed in more detail in the Tasks for Service Requests section above, and the same options available for Service Requests apply to Projects.

The methodology for setting up task workflows and the way in which the prerequisite tasks are handled is discussed above in the Task Workflows Table section and in the Task Templates section.

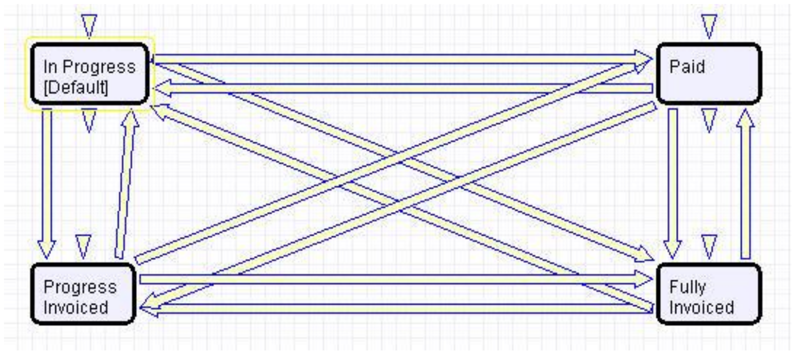
Purchase Orders (PO) Table

Purpose: The Purchase Order table tracks authorized billable hours for a project. It is shown as a related table within the Project table. It could easily be linked to support cases or Quotes or other tables within the system.

Use Case

PO records are creatable manually via the web form, from within Project records in the related table, or via mass import. Only members of the Professional Services, Sales and admin Groups may create or import records. PO records are creatable in any workflow state.

Only members of the Professional Services, Sales and admin Groups may edit records, but Support Staff may view all PO records.



Default workflow for the Purchase Orders table

Ownership

Records in this table are "owned" by an Employee, generally the one who created the record. Each record is associated with a particular Employee login.

Support Cases Table

Purpose: This table is used to manage external customer support requests. It is set up to be accessed by the Customer group and the Base Technical Group.

Use Case

End User Record Submission

Customers may create support cases using the tab in the end user interface or by sending an email (once an inbound email address is set up).

When a customer submits a support case, the contact information fields automatically populate based on the details in his/her user record. If the user record doesn't contain a value in the Customer Name or Email fields, the customer will be required to enter a value in those fields manually.

The Type of Issue is set by default to Question. If the customer changes it to Installation Issue or Bug, they will be required to fill out the Steps to Reproduce field. The case is assigned by default to the Support Team and the default status is Open.

Technician Record Submission

Staff members may also submit support cases on behalf of a customer, associating the customer with the case. If an internal staff user creates a case, he may assign it directly to an individual or a team other than the Support Team and may set its starting status to Open, Assigned, or Closed.

When the record is created, emails are sent to the customer acknowledging receipt of the support case and to the assigned team (or person) telling them the case has been assigned to them.

If a Support Staff technician creates a record in a status of Closed an email is sent to the customer telling them how to reopen their case.

Workflow actions send these emails automatically, but staff users can override them if given permission to override workflow actions.

Processing of Records

When a technician works on a case, if he needs more information from the customer in order to take further action, he can set the status to Sent to Customer. This will automatically send an email to the customer requesting further information and include the content of the Additional Notes field, an append-only field that is used to communicate with the customer. The email includes a hyperlink for the customer to click to login to edit the case directly.

When the customer edits the case or replies to the email, the status changes to Updated by Customer and an email notifies the assigned person that the customer has replied. The customer is able to update the Additional Notes field directly and any text from an email reply to a system email maps to that same field.

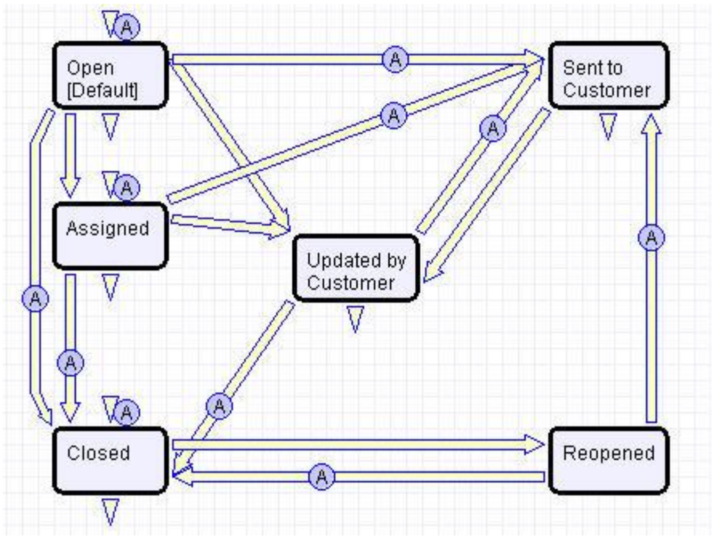
If the customer updates the case at any point, an email notifies the assigned person of the update.

If the technician needs to reassign the case to someone else, he or she simply changes the Assigned Person field to that person's name and the system will email the new assignee notifying them of the reassignment.

The Staff Only Notes field holds working notes that should not be visible to the customer.

When the technician has completed work on the case, he/she sets the Status field to Closed and puts the solution notes into the Solution field. This triggers an email to the customer that includes the content of the Solution field and tells the customer that the work is done. This closing email gives the customer a hotlink back to the record if they wish to reopen it and instructs them to explain why they are not satisfied with the solution. Clicking the hotlink will automatically change the "I Would Like To Reopen My Ticket" field to Yes, which in turn sets the Status of the ticket to Reopened and notifies the assigned person.

By default, no escalation rules are set up for the Support Case table.



Support Case table default Workflow

Ownership

Records in this table are “owned” by the individual customer. This means each record is associated with a particular customer login and no other customers will be able to edit that record, though members of the Customer Manager group can view all records submitted by other people at their company.

Sales/CRM Tables: Leads Table

Purpose: This table is used as the initial point of entry for sales leads. Leads may self-register at your website or may be imported from a lead generation program. The initial qualification is done in this table, which contains all the fields desirable for managing sales.

Once a lead is qualified, it may either be fully worked in this table, or converted into a contact, an opportunity, and a company/account record, and the sales process may be managed in those records.

Use Case

Record Creation

Leads can be created directly using the web form, or, once an inbound email address is set up, via email. New leads may only start out as Qualified or Unqualified in the workflow. The Leads table is set up by default to allow those in the Guest group to create records, allowing "click to register" lead generation hotlinks, and the embedding of the lead creation form in a web page.

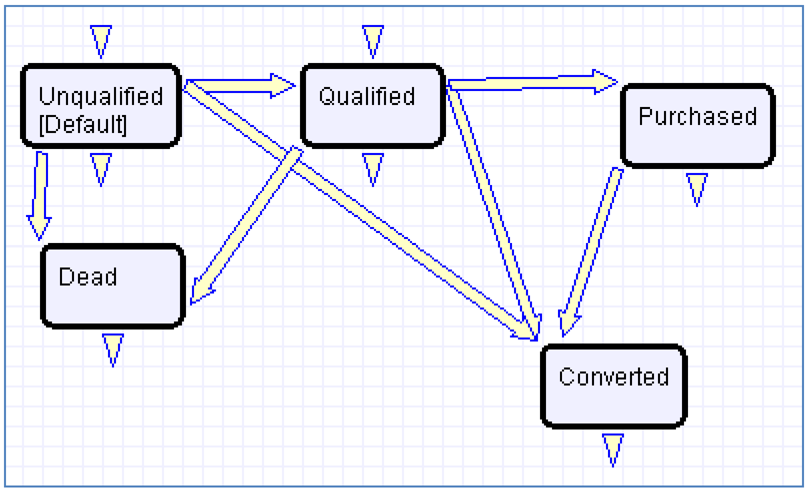
Naturally leads may also be imported from a spreadsheet from a lead generation program.

Processing of Records

Admins and members of the Sales group can create, view, and edit Leads. No other groups have access to the table by default.

When a lead's status changes to "Converted", Agiloft converts the information in the record into three new records in the three other tables: Company, Opportunity, and Contact. This order of creation is important because both the Opportunity and Contact records will contain links to the original Company via the new record. If the Company record is not created first, the Contact and Opportunity records will be unlinked and orphaned, disabling reporting features.

Data fields containing information relevant to the company, such as address and billing address, company website, industry, annual revenue and number of employees, map to the Company record. Sales-specific data fields, such as key requirements, earliest and latest possible close date, and sales actions taken, map to the Opportunity record. All data from the lead referencing a specific person at the company map into a new Contact containing the individual's desk and cell phone numbers, email address, email preferences, work hours, and so on.



Default workflow for Leads table

Ownership

Records in this table are “owned” by the individual assigned sales rep, so each record is associated with a particular user login. Only members of the Sales and admin groups can view or edit Leads.

Sales/CRM Tables: Opportunities Table, Deals Sub-table

Purpose: This table tracks sales opportunity information and contract information when an opportunity becomes a sale. An opportunity, when closed, is simply changed into a Deal, and all information is maintained, while new contract fields appear.

Use Case

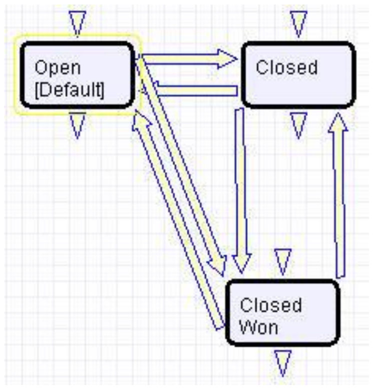
Most Opportunity records are created from Leads via a Conversion rule. Deals are typically created from Opportunities when they change to "Closed Won". Members of the Admin or Sales groups may create records manually.

An Opportunity record should be considered a possible sale, while a Deal is an actual finalized and confirmed agreement.

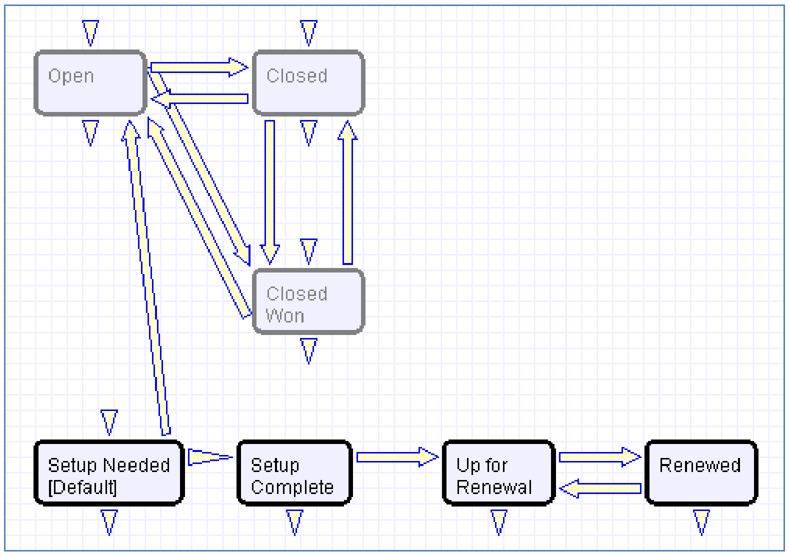
Admins and members of the Sales group can create, view and edit Opportunities. Members of the Professional Services and Support Staff groups may view all Opportunities. No other groups have access to the table by default.

The Opportunities table tracks information on possible sales for your organization. The Deals section handles Opportunities that have been successfully closed and turned into contracts and contains specialized fields for each contract's unique information, such as SLA requirements, renewal dates, and so on.

When an Opportunity's status changes to "Closed Won", a rule converts it to a new Deal record.



Default workflow for the Opportunities Table



Default workflow for the Deals sub-table

Ownership

Records in this table are "owned" by the individual assigned sales rep, so each record is associated with a particular user login from the Contacts table.

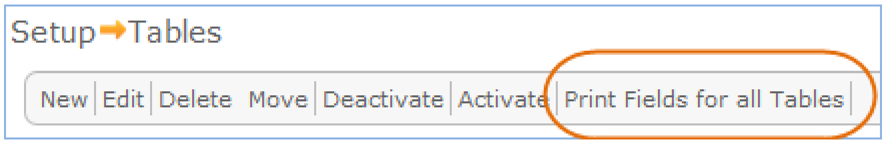
APPENDIX A – DETAILED RULES SETUP

In this Appendix we provide detailed technical information on the rules and actions that are preconfigured and other details of the out-of-the-box implementation. The rules and actions that run in the background to validate that users follow the correct processes and to automate process steps and notifications are documented with screenshots and details.

This automation is described in more general terms in the Use Case section of the main document for each table.

The tables are listed alphabetically here.

It is now possible to create an excel file of all the tables and fields of the system by going to Setup > Tables > Print Fields for all Tables:



This will always be the up to date documentation of the table setup. We have removed earlier Word tables of fields. The excel spreadsheet is also available for download from our website in the resources area.

Approvals

Approval Rules and Actions

The Approvals table has the following rules set up.

Create: All Contract Approval Create Actions (web, API)

Purpose: This handles the creation and assignment actions for all newly created contract approval records.

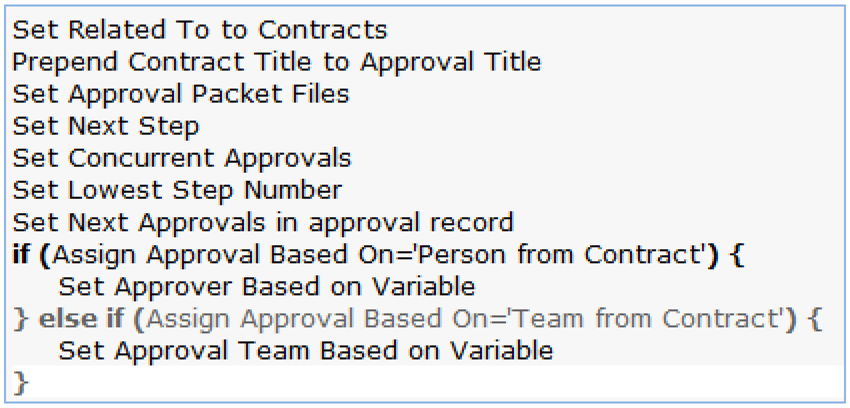
Runs: When created by web or API.

Search Filter: R: Contract ID is not empty

Priority: 1

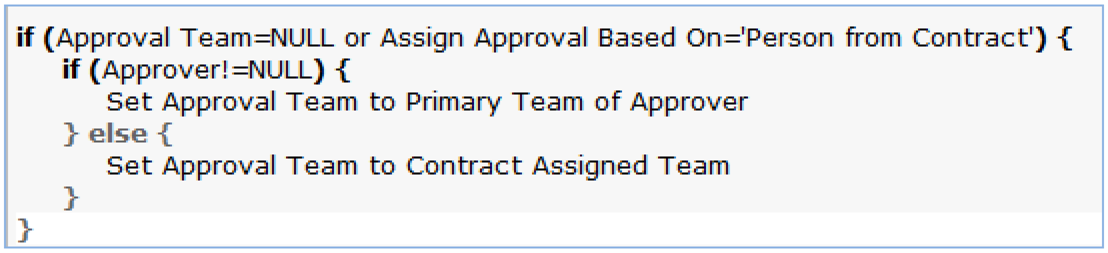
Actions:

I: Contract Approval Creation Actions



```
Set Related To to Contracts
Prepend Contract Title to Approval Title
Set Approval Packet Files
Set Next Step
Set Concurrent Approvals
Set Lowest Step Number
Set Next Approvals in approval record
if (Assign Approval Based On='Person from Contract') {
    Set Approver Based on Variable
} else if (Assign Approval Based On='Team from Contract') {
    Set Approval Team Based on Variable
}
```

I: Set Approval Team to Contract Assigned Team if empty



```
if (Approval Team=NULL or Assign Approval Based On='Person from Contract') {
    if (Approver!=NULL) {
        Set Approval Team to Primary Team of Approver
    } else {
        Set Approval Team to Contract Assigned Team
    }
}
```

Create: All Document Approval Creation Actions (web, API)

Purpose: This handles the creation, assignment, and notification actions for all newly created document approval records.

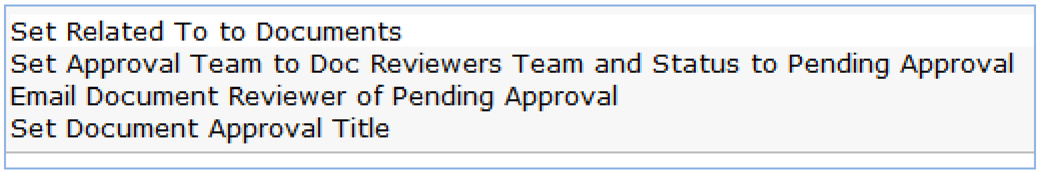
Runs: When created by web or API

Search filter: R: Document ID is not empty

Priority: 1

Actions:

I: Create Actions for Document Approvals



```
Set Related To to Documents
Set Approval Team to Doc Reviewers Team and Status to Pending Approval
Email Document Reviewer of Pending Approval
Set Document Approval Title
```

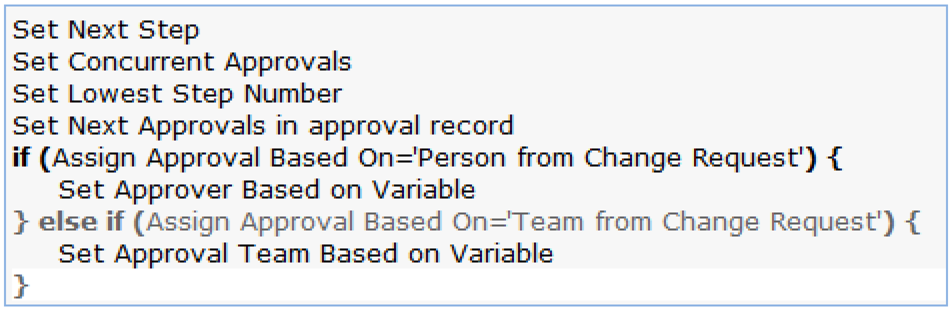
Create: All Change Approval Creation Actions (web, API) – New

Purpose: This handles the setting of concurrent approvals, next step, lowest step number and all next approvals as well as the assignment actions

Runs: When created by web or API

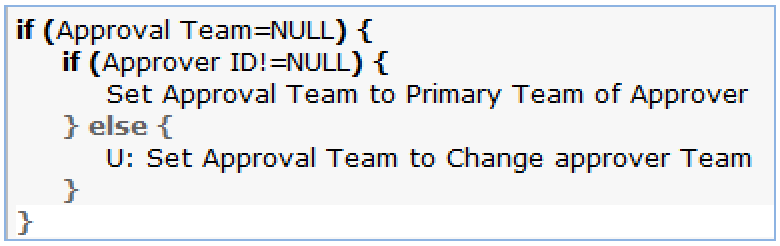
Search filter: R: Change Request ID is Not Null

I: Change Request Creation Actions



```
Set Next Step
Set Concurrent Approvals
Set Lowest Step Number
Set Next Approvals in approval record
if (Assign Approval Based On='Person from Change Request') {
    Set Approver Based on Variable
} else if (Assign Approval Based On='Team from Change Request') {
    Set Approval Team Based on Variable
}
```

I: Assign Actions



```
if (Approval Team=NULL) {
    if (Approver ID!=NULL) {
        Set Approval Team to Primary Team of Approver
    } else {
        U: Set Approval Team to Change approver Team
    }
}
```

Create: Notify if Needed for Contract Approval (web, API)

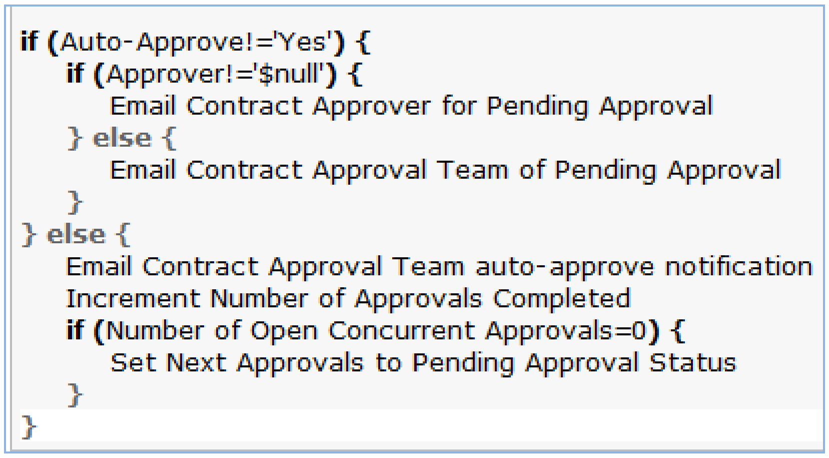
Purpose: If the approval is created in a status of Pending Approval, notify the Approver or Approval Team or auto-approve if appropriate. This would apply to an ad hoc approval when it is created.

Runs: When created by web or API

Priority: 3

Search Filter: Status is Pending Approval and Contract ID is not empty (i.e., it is for a contract)

I: Handle Approvals created in Pending Approval Status



```
if (Auto-Approve!='Yes') {
    if (Approver!='$null') {
        Email Contract Approver for Pending Approval
    } else {
        Email Contract Approval Team of Pending Approval
    }
} else {
    Email Contract Approval Team auto-approve notification
    Increment Number of Approvals Completed
    if (Number of Open Concurrent Approvals=0) {
        Set Next Approvals to Pending Approval Status
    }
}
```

Edit: All Edit Actions for Document Approval Updates (web and API)

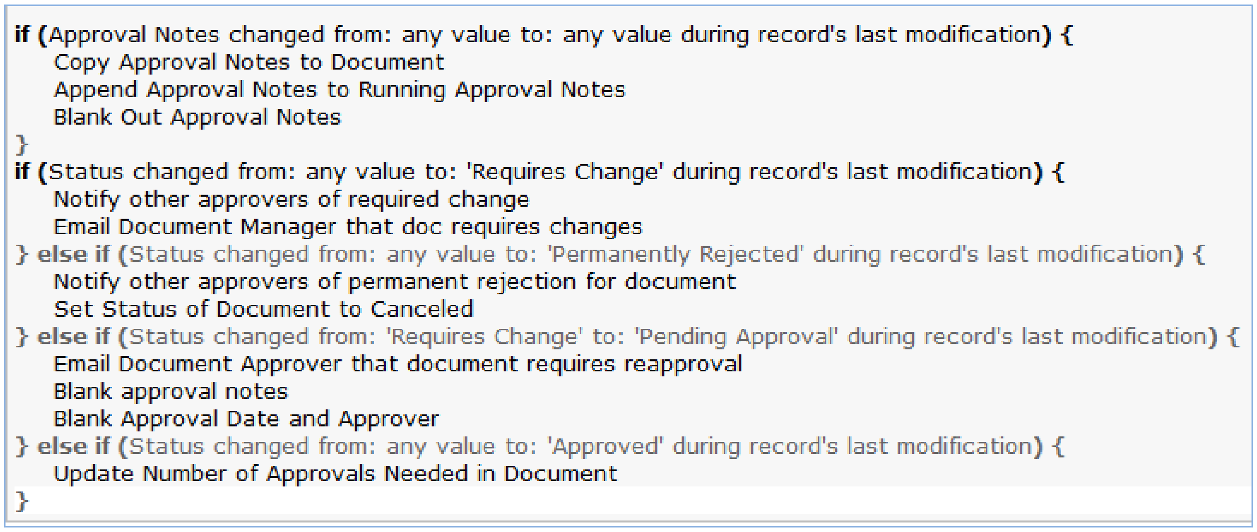
Purpose: This handles all document-related approvals and rejections, as well as approval notes updates.

Runs: When edited by web or API

Search filter: Related to Documents

Priority: 4

I: All Document Approval Edit Actions



```
if (Approval Notes changed from: any value to: any value during record's last modification) {
    Copy Approval Notes to Document
    Append Approval Notes to Running Approval Notes
    Blank Out Approval Notes
}
if (Status changed from: any value to: 'Requires Change' during record's last modification) {
    Notify other approvers of required change
    Email Document Manager that doc requires changes
} else if (Status changed from: any value to: 'Permanently Rejected' during record's last modification) {
    Notify other approvers of permanent rejection for document
    Set Status of Document to Canceled
} else if (Status changed from: 'Requires Change' to: 'Pending Approval' during record's last modification) {
    Email Document Approver that document requires reapproval
    Blank approval notes
    Blank Approval Date and Approver
} else if (Status changed from: any value to: 'Approved' during record's last modification) {
    Update Number of Approvals Needed in Document
}
```

Edit: All Contract Approval Edit Actions

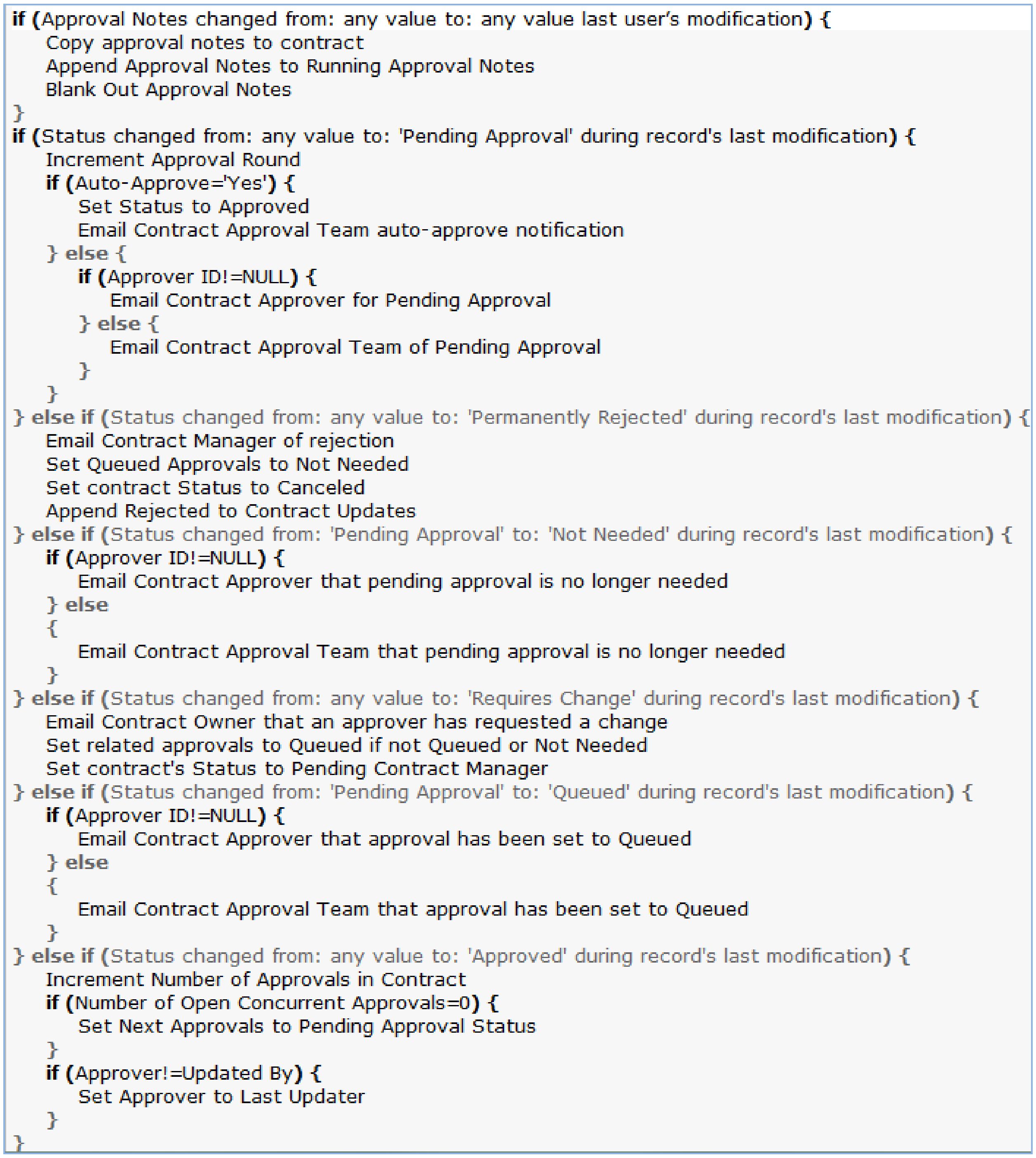
Purpose: Handles updates to approvals for a contract. Launches the next approval in the workflow when an approval is completed.

Runs: When edited by web or API

Priority: 5

Saved Search: R: Related to is Contracts

I: All Contract Approval Edit Actions



```
if (Approval Notes changed from: any value to: any value last user's modification) {
    Copy approval notes to contract
    Append Approval Notes to Running Approval Notes
    Blank Out Approval Notes
}
if (Status changed from: any value to: 'Pending Approval' during record's last modification) {
    Increment Approval Round
    if (Auto-Approve='Yes') {
        Set Status to Approved
        Email Contract Approval Team auto-approve notification
    } else {
        if (Approver ID!=NULL) {
            Email Contract Approver for Pending Approval
        } else {
            Email Contract Approval Team of Pending Approval
        }
    }
} else if (Status changed from: any value to: 'Permanently Rejected' during record's last modification) {
    Email Contract Manager of rejection
    Set Queued Approvals to Not Needed
    Set contract Status to Canceled
    Append Rejected to Contract Updates
} else if (Status changed from: 'Pending Approval' to: 'Not Needed' during record's last modification) {
    if (Approver ID!=NULL) {
        Email Contract Approver that pending approval is no longer needed
    } else
    {
        Email Contract Approval Team that pending approval is no longer needed
    }
} else if (Status changed from: any value to: 'Requires Change' during record's last modification) {
    Email Contract Owner that an approver has requested a change
    Set related approvals to Queued if not Queued or Not Needed
    Set contract's Status to Pending Contract Manager
} else if (Status changed from: 'Pending Approval' to: 'Queued' during record's last modification) {
    if (Approver ID!=NULL) {
        Email Contract Approver that approval has been set to Queued
    } else
    {
        Email Contract Approval Team that approval has been set to Queued
    }
} else if (Status changed from: any value to: 'Approved' during record's last modification) {
    Increment Number of Approvals in Contract
    if (Number of Open Concurrent Approvals=0) {
        Set Next Approvals to Pending Approval Status
    }
    if (Approver!=Updated By) {
        Set Approver to Last Updater
    }
}
```

Edit: Approval by Email (web)

Purpose: Checks for updates by email that change the Approved by Email field from No to Yes, and changes the Status to Approved accordingly

Runs: On edit by web

Priority: 6

Saved Search: R: Approved by Email changed to Yes and Status is Pending Approval

U: Set Approved by Email to No

U: Set Approval Status to Approved

Edit: All Change Approval Edit Actions (web, API) – New

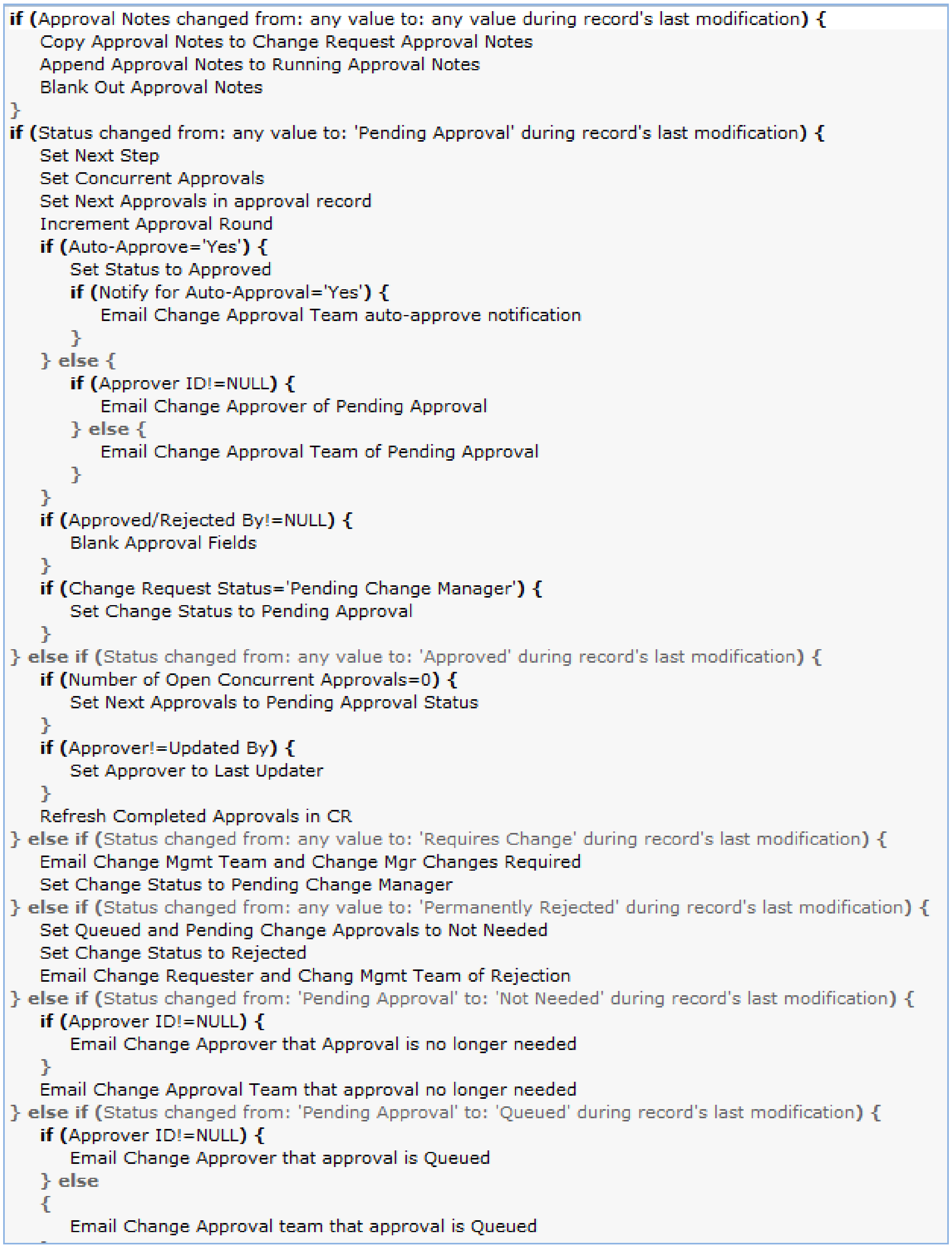
Purpose: This handles the updates of CR for approval notes and workflow for approvals, rejections or changes.

Runs: When edited by web or API

Priority: 5

Saved Search: R: Related to Change Request

I: All Change Approval Edit Actions



```
if (Approval Notes changed from: any value to: any value during record's last modification) {
    Copy Approval Notes to Change Request Approval Notes
    Append Approval Notes to Running Approval Notes
    Blank Out Approval Notes
}
if (Status changed from: any value to: 'Pending Approval' during record's last modification) {
    Set Next Step
    Set Concurrent Approvals
    Set Next Approvals in approval record
    Increment Approval Round
    if (Auto-Approve='Yes') {
        Set Status to Approved
        if (Notify for Auto-Approval='Yes') {
            Email Change Approval Team auto-approve notification
        }
    } else {
        if (Approver ID!=NULL) {
            Email Change Approver of Pending Approval
        } else {
            Email Change Approval Team of Pending Approval
        }
    }
    if (Approved/Rejected By!=NULL) {
        Blank Approval Fields
    }
    if (Change Request Status='Pending Change Manager') {
        Set Change Status to Pending Approval
    }
} else if (Status changed from: any value to: 'Approved' during record's last modification) {
    if (Number of Open Concurrent Approvals=0) {
        Set Next Approvals to Pending Approval Status
    }
    if (Approver!=Updated By) {
        Set Approver to Last Updater
    }
    Refresh Completed Approvals in CR
} else if (Status changed from: any value to: 'Requires Change' during record's last modification) {
    Email Change Mgmt Team and Change Mgr Changes Required
    Set Change Status to Pending Change Manager
} else if (Status changed from: any value to: 'Permanently Rejected' during record's last modification) {
    Set Queued and Pending Change Approvals to Not Needed
    Set Change Status to Rejected
    Email Change Requester and Chang Mgmt Team of Rejection
} else if (Status changed from: 'Pending Approval' to: 'Not Needed' during record's last modification) {
    if (Approver ID!=NULL) {
        Email Change Approver that Approval is no longer needed
    }
    Email Change Approval Team that approval no longer needed
} else if (Status changed from: 'Pending Approval' to: 'Queued' during record's last modification) {
    if (Approver ID!=NULL) {
        Email Change Approver that approval is Queued
    } else
    {
        Email Change Approval team that approval is Queued
```

Approval Templates Table

Rules and Actions

The Approval Templates table has the following rules set up:

Create/Edit: Blank out fields depending on Assign Approval Based On (web, API)

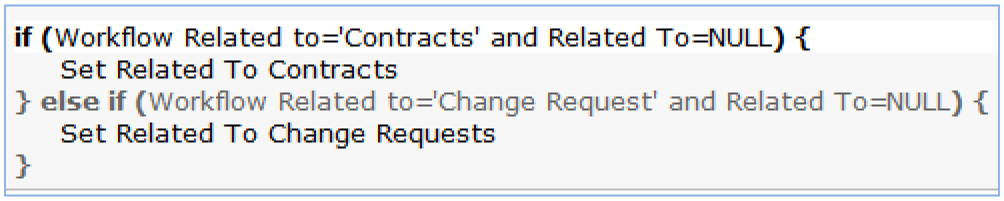
Purpose: Clear out the non-applicable fields depending on the value of Assign Approval Based On. This takes care of a situation in which the person starts out by defining the template one way, then changes to another way without clearing out the value. It also sets the Related to field for the Approval template

Runs: When created or edited by web or API

Search Filter: None

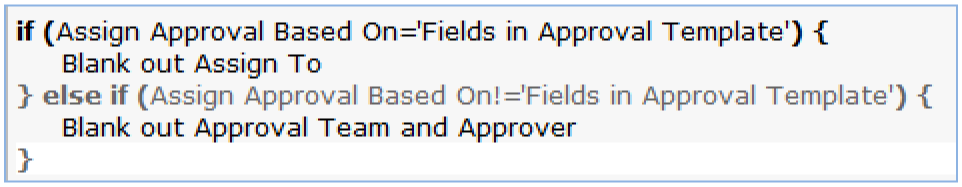
Priority: 1

I: Set Approval Template Relationship



```
if (Workflow Related to='Contracts' and Related To=NULL) {
    Set Related To Contracts
} else if (Workflow Related to='Change Request' and Related To=NULL) {
    Set Related To Change Requests
}
```

I: Blank out fields based on Assign Approval Based On



```
if (Assign Approval Based On='Fields in Approval Template') {
    Blank out Assign To
} else if (Assign Approval Based On!='Fields in Approval Template') {
    Blank out Approval Team and Approver
}
```

Edit: All Contract Approval Edit Actions (web, api)

Purpose: This rule handles the creation of an approval record from an approval template when the use in the contract has clicked a button to Generate Approvals.

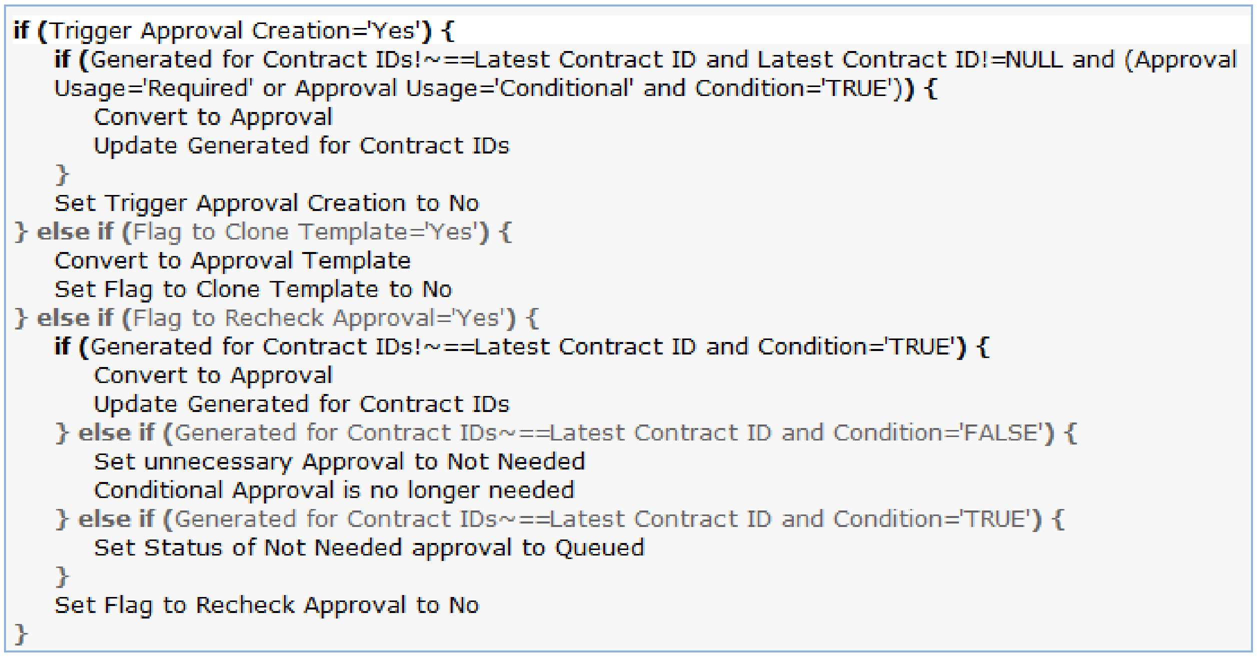
Runs: When created or edited by web or API.

Search Filter: CF: Workflow Related to Contracts

Priority: 2

Actions:

I: Handle Flags to Generate or Check Approvals



```
if (Trigger Approval Creation='Yes') {
    if (Generated for Contract IDs!~==Latest Contract ID and Latest Contract ID!=NULL and (Approval
    Usage='Required' or Approval Usage='Conditional' and Condition='TRUE')) {
        Convert to Approval
        Update Generated for Contract IDs
    }
    Set Trigger Approval Creation to No
} else if (Flag to Clone Template='Yes') {
    Convert to Approval Template
    Set Flag to Clone Template to No
} else if (Flag to Recheck Approval='Yes') {
    if (Generated for Contract IDs!~==Latest Contract ID and Condition='TRUE') {
        Convert to Approval
        Update Generated for Contract IDs
    } else if (Generated for Contract IDs~==Latest Contract ID and Condition='FALSE') {
        Set unnecessary Approval to Not Needed
        Conditional Approval is no longer needed
    } else if (Generated for Contract IDs~==Latest Contract ID and Condition='TRUE') {
        Set Status of Not Needed approval to Queued
    }
    Set Flag to Recheck Approval to No
}
```

Edit: All Change Approval Edit Actions (web, API)

Purpose: This triggers the appropriate approval records for a change request when the Generate Approvals or Check Conditional Approvals button is pressed from the change request. The rule only generates a conditional approval if the condition is true.

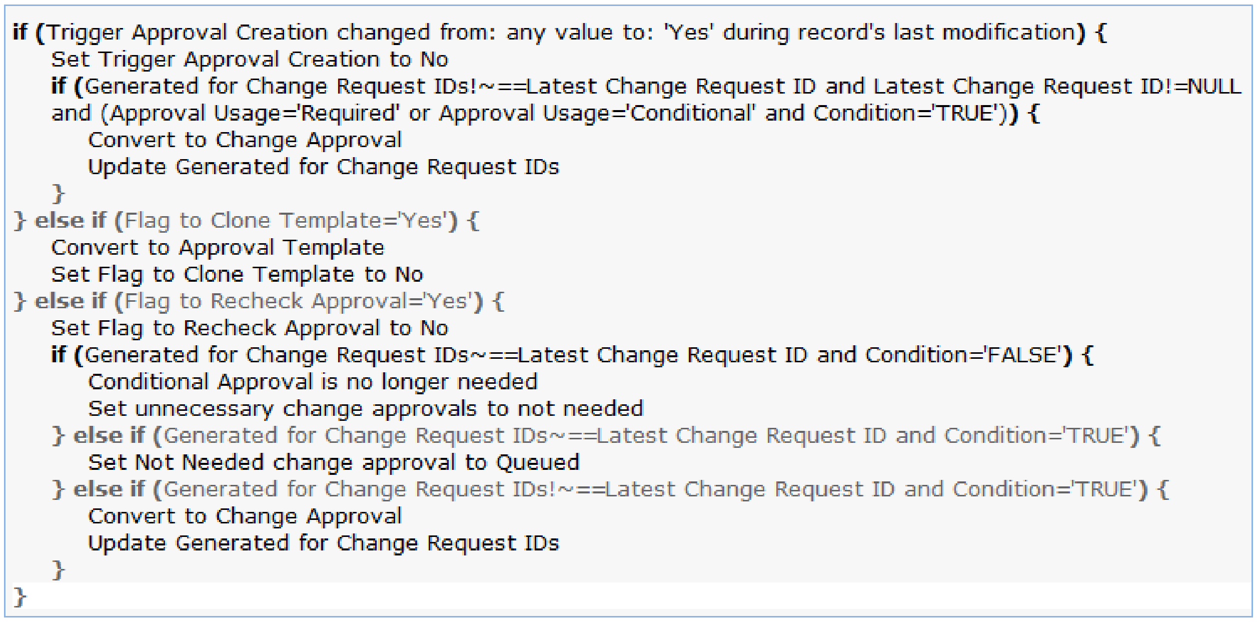
Runs: on Edit by Web or API

Search Filter: CF: Workflow Related to Change Requests

Priority: 2

Actions:

I: Handle Flags to Generate or Check Change Approvals



```
if (Trigger Approval Creation changed from: any value to: 'Yes' during record's last modification) {
    Set Trigger Approval Creation to No
    if (Generated for Change Request IDs!~==Latest Change Request ID and Latest Change Request ID!=NULL
    and (Approval Usage='Required' or Approval Usage='Conditional' and Condition='TRUE')) {
        Convert to Change Approval
        Update Generated for Change Request IDs
    }
} else if (Flag to Clone Template='Yes') {
    Convert to Approval Template
    Set Flag to Clone Template to No
} else if (Flag to Recheck Approval='Yes') {
    Set Flag to Recheck Approval to No
    if (Generated for Change Request IDs~==Latest Change Request ID and Condition='FALSE') {
        Conditional Approval is no longer needed
        Set unnecessary change approvals to not needed
    } else if (Generated for Change Request IDs~==Latest Change Request ID and Condition='TRUE') {
        Set Not Needed change approval to Queued
    } else if (Generated for Change Request IDs!~==Latest Change Request ID and Condition='TRUE') {
        Convert to Change Approval
        Update Generated for Change Request IDs
    }
}
```

Approval Workflows Table

Rules and Actions

Create: All Create Actions (web, API)

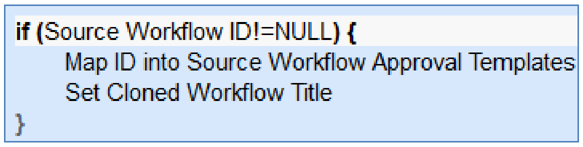
Purpose: This rule handles workflows created by cloning.

Runs: When created by web or API

Search filter: None

Priority: 1

I: All Create Actions



```
if (Source Workflow ID!=NULL) {
    Map ID into Source Workflow Approval Templates
    Set Cloned Workflow Title
}
```

Change Requests Table

Rules and Actions

Rule: Create: All Creation Actions

Purpose: Updates the originating record if the change request was created from an SR, Incident, or Problem. Sets the Assigned Team to the responsible team for the service, if there is one. Sends an acknowledgement email to the submitter and to the assigned person or team.

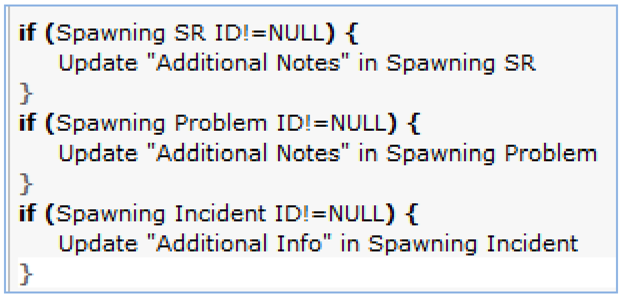
Runs: On creation by Web or API

Search Filter: None

Priority: 2

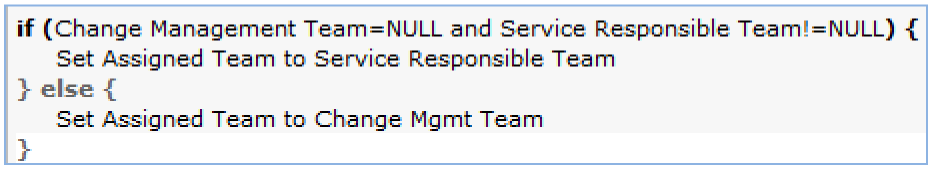
Actions:

I: Validations and Conditional Actions



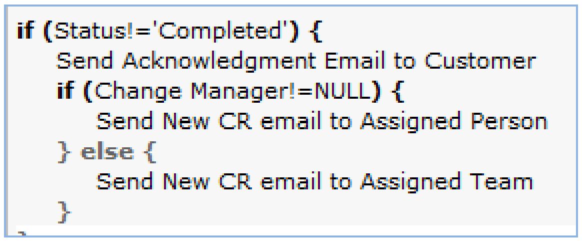
```
if (Spawning SR ID!=NULL) {
    Update "Additional Notes" in Spawning SR
}
if (Spawning Problem ID!=NULL) {
    Update "Additional Notes" in Spawning Problem
}
if (Spawning Incident ID!=NULL) {
    Update "Additional Info" in Spawning Incident
}
```

I: Assign Change Request



```
if (Change Management Team=NULL and Service Responsible Team!=NULL) {
    Set Assigned Team to Service Responsible Team
} else {
    Set Assigned Team to Change Mgmt Team
}
```

I: Send Emails to Submitter and Assignee



```
if (Status!='Completed') {
    Send Acknowledgment Email to Customer
    if (Change Manager!=NULL) {
        Send New CR email to Assigned Person
    } else {
        Send New CR email to Assigned Team
    }
}
```

Rule: Edit: All Edit Validations (web only)

Purpose: Prevents marking a change request as completed if there are unfinished tasks. Prevents setting the status to In Progress before the change request has been approved. If the status is changed to Completed but there were some tasks with failures, changes the status to Completed with Failures.

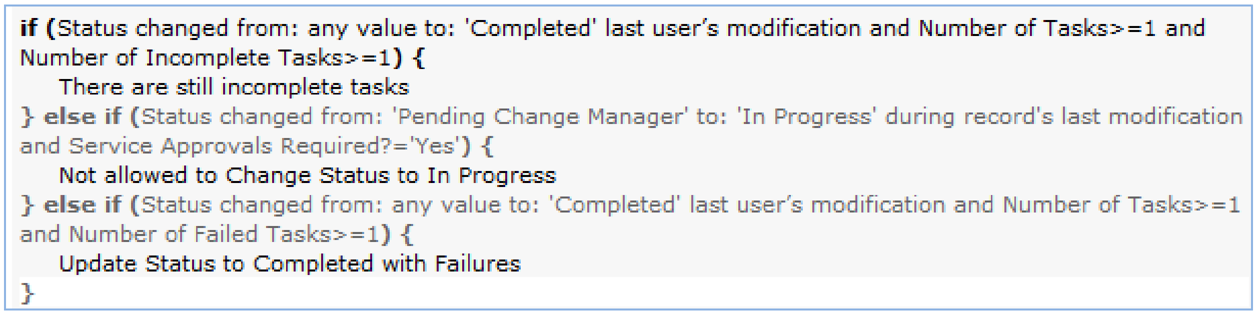
Runs: On edit by web only

Search Filter: None

Priority: 0

Actions:

I: All Edit Validations



```
if (Status changed from: any value to: 'Completed' last user's modification and Number of Tasks>=1 and
Number of Incomplete Tasks>=1) {
    There are still incomplete tasks
} else if (Status changed from: 'Pending Change Manager' to: 'In Progress' during record's last modification
and Service Approvals Required?='Yes') {
    Not allowed to Change Status to In Progress
} else if (Status changed from: any value to: 'Completed' last user's modification and Number of Tasks>=1
and Number of Failed Tasks>=1) {
    Update Status to Completed with Failures
}
```

Rule: Edit: All Edit actions (API enabled)

Purpose: Runs several actions. Watches for completion of approvals and when all approvals have been completed, changes the status to Pending Approval. Maps the All Cis into the Generate Tasks for CI's field. Updates the source records with information when the change is completed. Notifies the change manager when all tasks are completed.

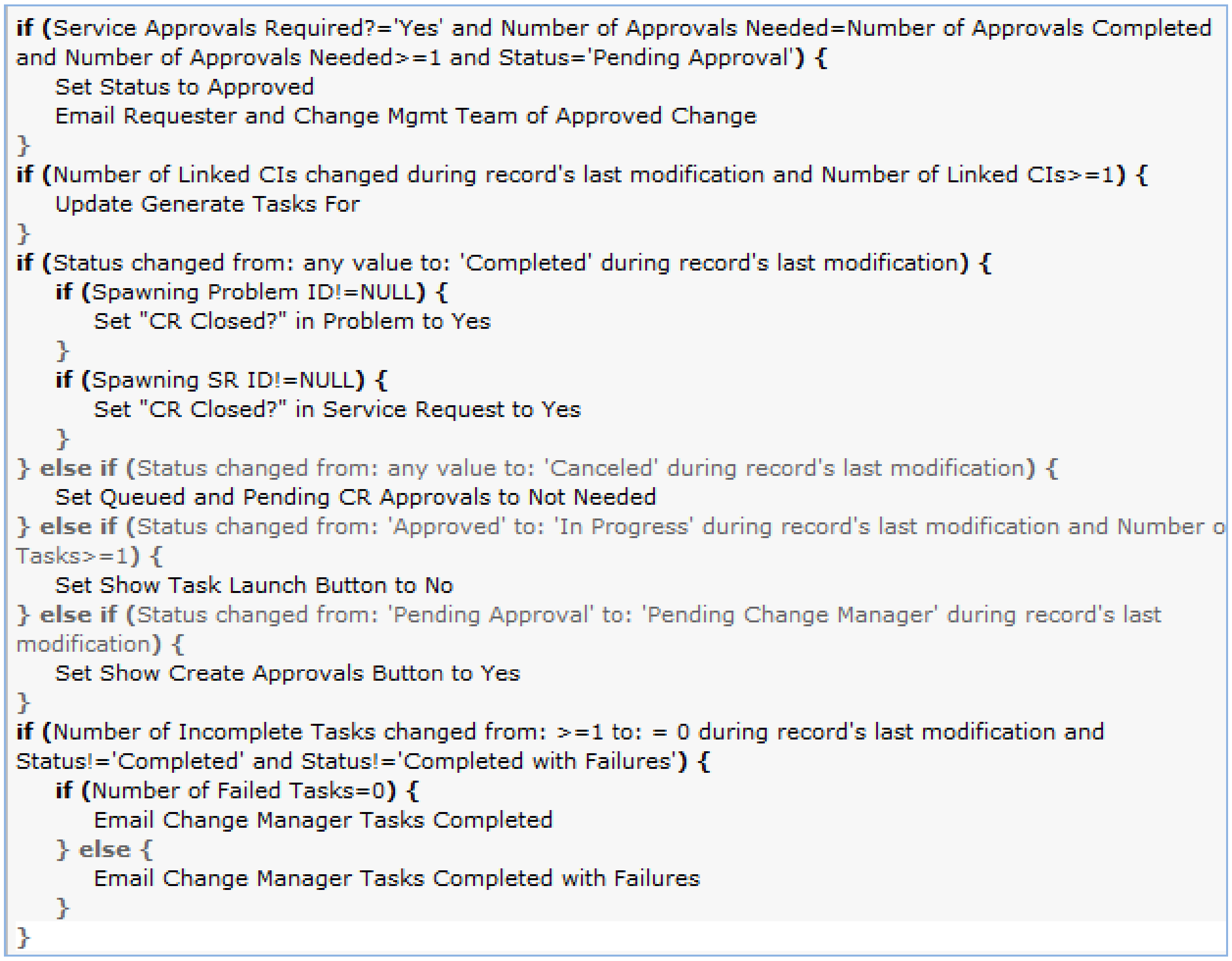
Runs: On edit by web or API

Search Filter: None

Priority: 3

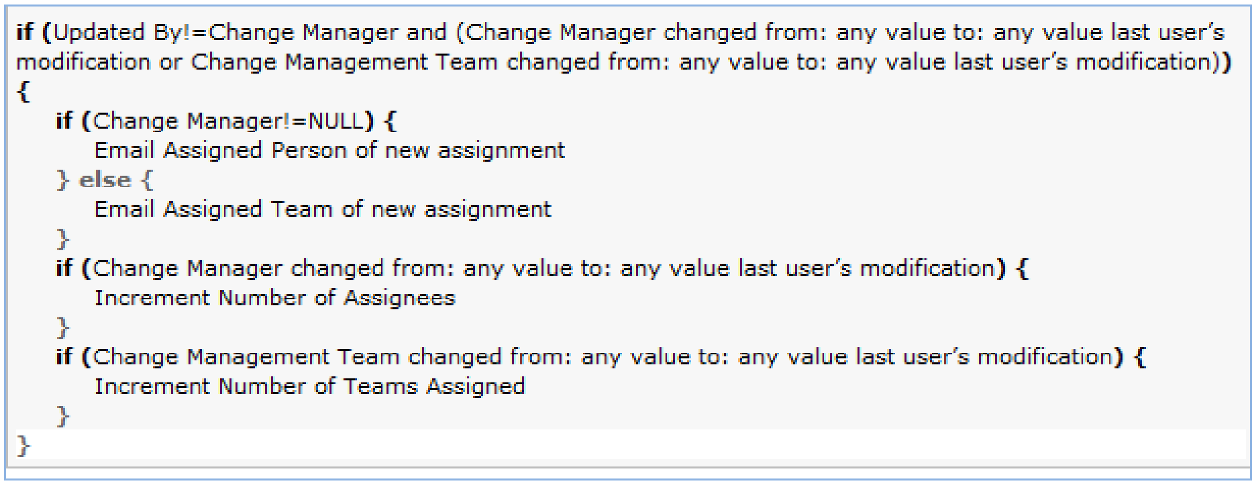
Actions:

I: All Edit Actions with API



```
if (Service Approvals Required?='Yes' and Number of Approvals Needed=Number of Approvals Completed
and Number of Approvals Needed>=1 and Status='Pending Approval') {
    Set Status to Approved
    Email Requester and Change Mgmt Team of Approved Change
}
if (Number of Linked CIs changed during record's last modification and Number of Linked CIs>=1) {
    Update Generate Tasks For
}
if (Status changed from: any value to: 'Completed' during record's last modification) {
    if (Spawning Problem ID!=NULL) {
        Set "CR Closed?" in Problem to Yes
    }
    if (Spawning SR ID!=NULL) {
        Set "CR Closed?" in Service Request to Yes
    }
} else if (Status changed from: any value to: 'Canceled' during record's last modification) {
    Set Queued and Pending CR Approvals to Not Needed
} else if (Status changed from: 'Approved' to: 'In Progress' during record's last modification and Number o
Tasks>=1) {
    Set Show Task Launch Button to No
} else if (Status changed from: 'Pending Approval' to: 'Pending Change Manager' during record's last
modification) {
    Set Show Create Approvals Button to Yes
}
if (Number of Incomplete Tasks changed from: >=1 to: = 0 during record's last modification and
Status!='Completed' and Status!='Completed with Failures') {
    if (Number of Failed Tasks=0) {
        Email Change Manager Tasks Completed
    } else {
        Email Change Manager Tasks Completed with Failures
    }
}
```

I: Assignee Change Notifications

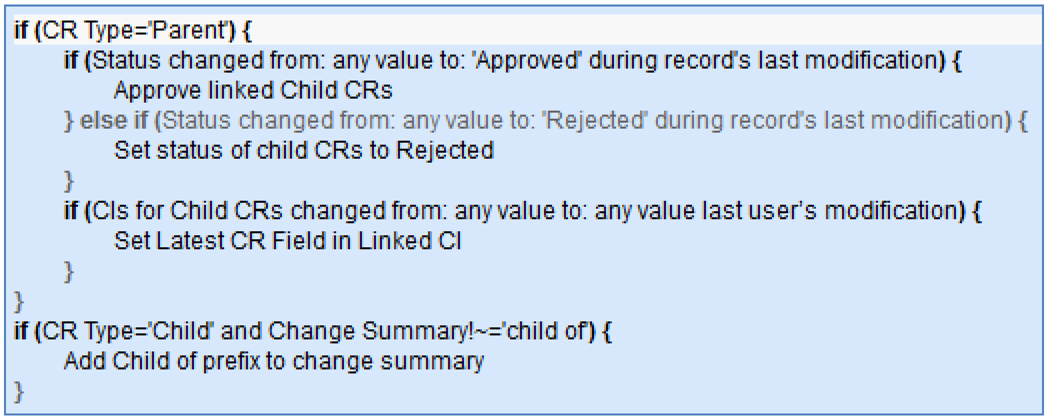


```
if (Updated By!=Change Manager and (Change Manager changed from: any value to: any value last user's
modification or Change Management Team changed from: any value to: any value last user's modification))
{
    if (Change Manager!=NULL) {
        Email Assigned Person of new assignment
    } else {
        Email Assigned Team of new assignment
    }
    if (Change Manager changed from: any value to: any value last user's modification) {
        Increment Number of Assignees
    }
    if (Change Management Team changed from: any value to: any value last user's modification) {
        Increment Number of Teams Assigned
    }
}
```

Rule: Parent/Child CR Handling

Runs on create and edit, including edits by other rules on creation

Saved Search: CR Type != Ind.



```
if (CR Type='Parent') {
    if (Status changed from: any value to: 'Approved' during record's last modification) {
        Approve linked Child CRs
    } else if (Status changed from: any value to: 'Rejected' during record's last modification) {
        Set status of child CRs to Rejected
    }
    if (CIs for Child CRs changed from: any value to: any value last user's modification) {
        Set Latest CR Field in Linked CI
    }
}
if (CR Type='Child' and Change Summary!~='child of') {
    Add Child of prefix to change summary
}
```

Action Buttons

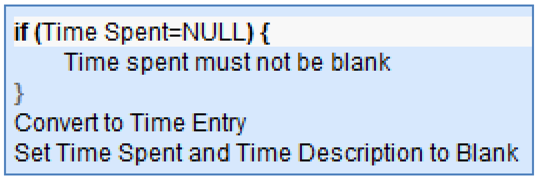
Add Time

Purpose: Provide users with a way to add Time Entries to a Change Request from the Change Request record itself.

On Click: Save Record, then execute action.

Actions:

I: Add Time



```
if (Time Spent=NULL) {
    Time spent must not be blank
}
Convert to Time Entry
Set Time Spent and Time Description to Blank
```

Notes: Triggers action to validate that Time Spent is not blank, then converts Time Spent and Time Description fields to a Time Entry record linked to the current Change Request record, then blanks out Time Spent and Time Description

Create New Task Template

See field itself for details

Generate Approval(s) from Workflow

Purpose: Triggers the creation of the approvals based on the workflow defined

On Click: Execute action.

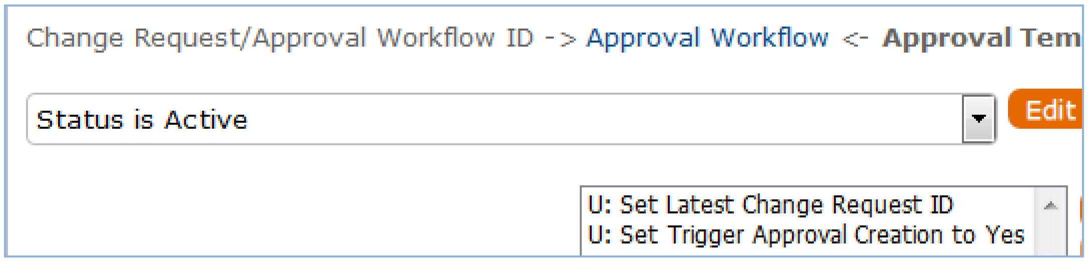
Visibility: visible when status is Pending Change Manager

Action:

I: Generate Approval Actions

```
if (Approval Workflow ID=NULL) {
    No workflow defined
} else {
    Trigger Required and Conditional Approval Creation
}
```

This goes down to the active approval templates and updates them to trigger the creation of the appropriate approvals:


Change Request/Approval Workflow ID -> Approval Workflow <- Approval Tem

Status is Active

U: Set Latest Change Request ID
U: Set Trigger Approval Creation to Yes

Generate Tasks

This button generates tasks from a workflow. It creates one task per selected CI per template. It goes down through the Configuration Items to generate the tasks. See field itself for more details.

Launch Approval Process

This launches the first approval(s), those with the lowest step number. See field itself for details.

Launch Tasks

This sets the tasks with no prerequisites to Assigned, unless they are conditional and the condition is not met, in which case it marks them as Not Needed. See the field itself and the use case for details.

Recheck Conditional Approvals

This updates the approval templates on which the conditional approvals are based to check whether their condition is still met, and updates the linked approvals based on the findings. See the field itself for more details.

Refresh Configuration Items

This saves and reopens for editing, which has the result of copying the CI's from the All CI's field into the Generate Tasks for CI's field.

Companies Table

Company Saved Searches

The default searches provided for this table are detailed below.

Saved Search Name	Search Description
My Company	Returns records where the Company Name and Company City match those of the logged-in user
Reference Accounts	Shows all records for companies who have agreed to
Customers	Company Roles includes "Customer"
Manufacturers	Company Roles includes "Manufacturer"
Partners	Company Roles includes "Partner"
Prospects	Company Roles includes "Prospect"
Vendors	Company Roles includes "Vendor"

Action Buttons

Find local Hotels

Purpose: Automatically searches Google Maps for hotels near the company.

On Click: Opens URL: "http://maps.google.com/maps?f=q&q=hotels" + "+loc:+" + urlEncode($street_address_1) + ",+" + urlEncode($city) + ",+" + $us_state

Map it

Purpose: Automatically searches Google Maps for the location of the company.

On Click: Opens URL: "http://maps.google.com/maps?f=q&q=" + urlEncode($street_address_1) + ",+" + urlEncode($city) + ",+" + $us_state

Refresh User

Purpose: Saves record and reopens for edit to set most recent user to person editing record.

On Click: Save and Open record for edit

Configuration Items Table

Configuration Item Type and Subtype Dependencies

The tables below show the current setup of dependencies between CI Type and CI Subtype. These dependencies can be changed, as can all the choices, to meet your needs. The dependencies are also shown in the spreadsheet fields_and_choices.xls.

CI Type values	CI Subtype (linked) values
Communication Device	Aircard Conference Phone Desk Phone Fax Machine IP Phone Mobile Phone Modem PDA enabled Cell Phone Video Conferencing Voice Gateway WiFi
Computer	Desktop Laptop Tablet Virtual Machine Windows Workstation
Computer Peripheral	Bar Code Scanner Camera Disk Drive Docking Station External Hard Drive Modem Monitor Mouse Peripheral - Other UPS Video Card

Data Center	Modem Other Rack Router Server Storage Switch UPS Virtual Machine Wireless Access Point
Imaging/Printing	Copier Fax Machine Label Maker Network Printer Personal Printer Scanner
Infrastructure Item	Cluster Firewall FTP Server Load Balancer Monitoring Device Rack Video Conferencing Virtual Machine Virtual Private Network Voice Gateway VPN Appliance Web Service Web Site

Network Hardware	IP Device IP Network IP Router IP Switch KVM Switch Load Balancer Monitoring Device Network Adapter Network Appliance Other Hardware Switch Virtual Machine Voice Gateway VPN Appliance WAN Interface
Server	Directory Server Email Server Firewall FTP Server Java Server Server Storage Management Server Virtual Machine
Storage Device	Storage Storage Device Storage Management Server Storage Switch Tape Library
Software Application	Application Server Database Application Database Instance Desktop Application Google Search Appliance Infrastructure Software Operating System

	Other Application Web Application Web Server Application Web Service Windows Service
Storage Device	Storage Storage Device Storage Management Server Storage Switch Tape Library
Video Conferencing	Camera Other (DVD, Mixer, Amp) Projector Screen Television Video Conference Controller Web Camera
Other	
None	

Rules and Actions

Edit: Push Information to Task Template to Convert for Change Request

Purpose: Creates tasks based on workflow or templates for a change request. This is how tasks are created linked to specific CI's from a change request. The template ID is pushed into the Configuration Item and this action updates the task template to push the CI information into it so it can be converted into a task.

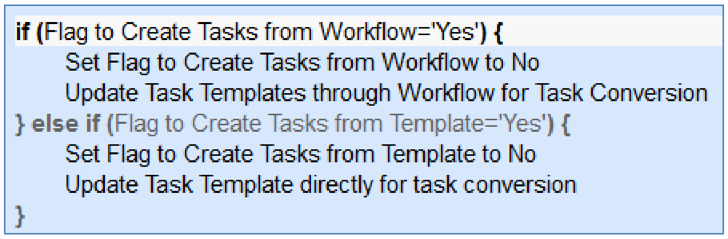
Runs: Edit by Web or API

Search Filter: R: Flag to Create Task from Workflow or Template is Yes

Priority: 5

Actions:

I: Trigger Task Conversion within Task Template



```
if (Flag to Create Tasks from Workflow='Yes') {
    Set Flag to Create Tasks from Workflow to No
    Update Task Templates through Workflow for Task Conversion
} else if (Flag to Create Tasks from Template='Yes') {
    Set Flag to Create Tasks from Template to No
    Update Task Template directly for task conversion
}
```

Edit: All Edit Actions

Purpose: Adds assets to Contracts when Latest Contract ID added to.

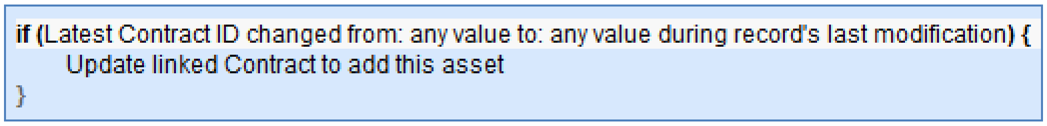
Runs: Edit by Web or API

Search Filter: None

Priority: 11

Actions:

I: Edit actions by api or web



```
if (Latest Contract ID changed from: any value to: any value during record's last modification) {
    Update linked Contract to add this asset
}
```

Contracts Table

Rules and Actions

The Contracts table has the following rules set up. Rules which are run based on a schedule (rather than those which are event-triggered) are identified by “TB”, for “Time-based”.

Create: All New Contract Actions

Purpose: This rule handles all update and linking actions upon creating a new record.

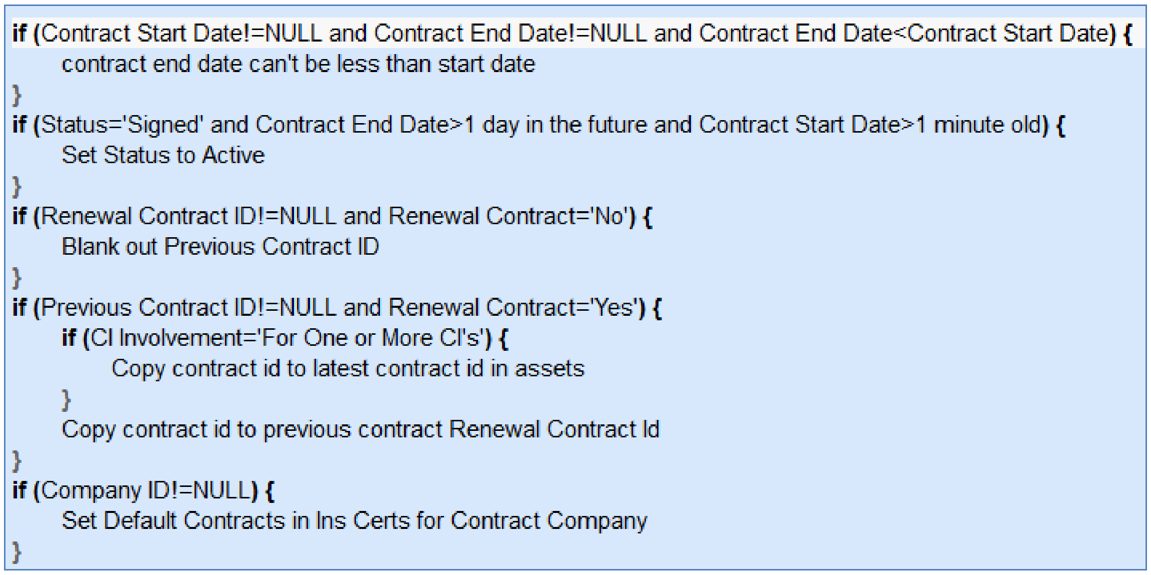
Runs: Upon creation by Web or API

Search Filter: None

Priority: 1

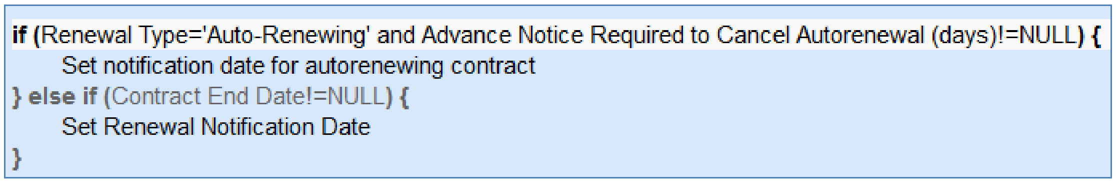
Actions:

I: All New Contract Actions



```
if (Contract Start Date!=NULL and Contract End Date!=NULL and Contract End Date<Contract Start Date) {
    contract end date can't be less than start date
}
if (Status='Signed' and Contract End Date>1 day in the future and Contract Start Date>1 minute old) {
    Set Status to Active
}
if (Renewal Contract ID!=NULL and Renewal Contract='No') {
    Blank out Previous Contract ID
}
if (Previous Contract ID!=NULL and Renewal Contract='Yes') {
    if (CI Involvement='For One or More CI's') {
        Copy contract id to latest contract id in assets
    }
    Copy contract id to previous contract Renewal Contract Id
}
if (Company ID!=NULL) {
    Set Default Contracts in Ins Certs for Contract Company
}
```

I: Set Renewal Notification Date based on renewal type



```
if (Renewal Type='Auto-Renewing' and Advance Notice Required to Cancel Autorenewal (days)!=NULL) {
    Set notification date for autorenewing contract
} else if (Contract End Date!=NULL) {
    Set Renewal Notification Date
}
```

Updated Field: Renewal Notification Date

Updated Value: $contract_end_date – ($days_in_advance_to_notify_for_renewal) DAYS

Edit: Updates by Party

Purpose: This rule handles notifications for Contracts that are updated by an outside party that does not have direct access to the system.

Runs: Upon edits by Email or Web

Search Filter: R: Updates by Vendor or Customer Team

Priority: 1

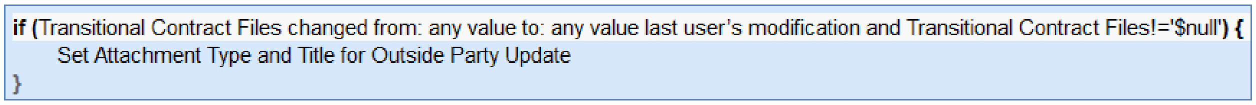
Actions:

E: Email Owner and Sender of Outside Party Update

Sent To: Contract Manager Team, Contract Manager, Sent to Party By.

Summary: This email notifies the relevant users that an outside party has updated the contract.

I: Handle Inbound Files



```
if (Transitional Contract Files changed from: any value to: any value last user's modification and Transitional Contract Files!='$null') {
      Set Attachment Type and Title for Outside Party Update
}
```

Edit: All Edit Actions by Web or API

Purpose: This rule handles the processes resulting from changes made by staff or rules.

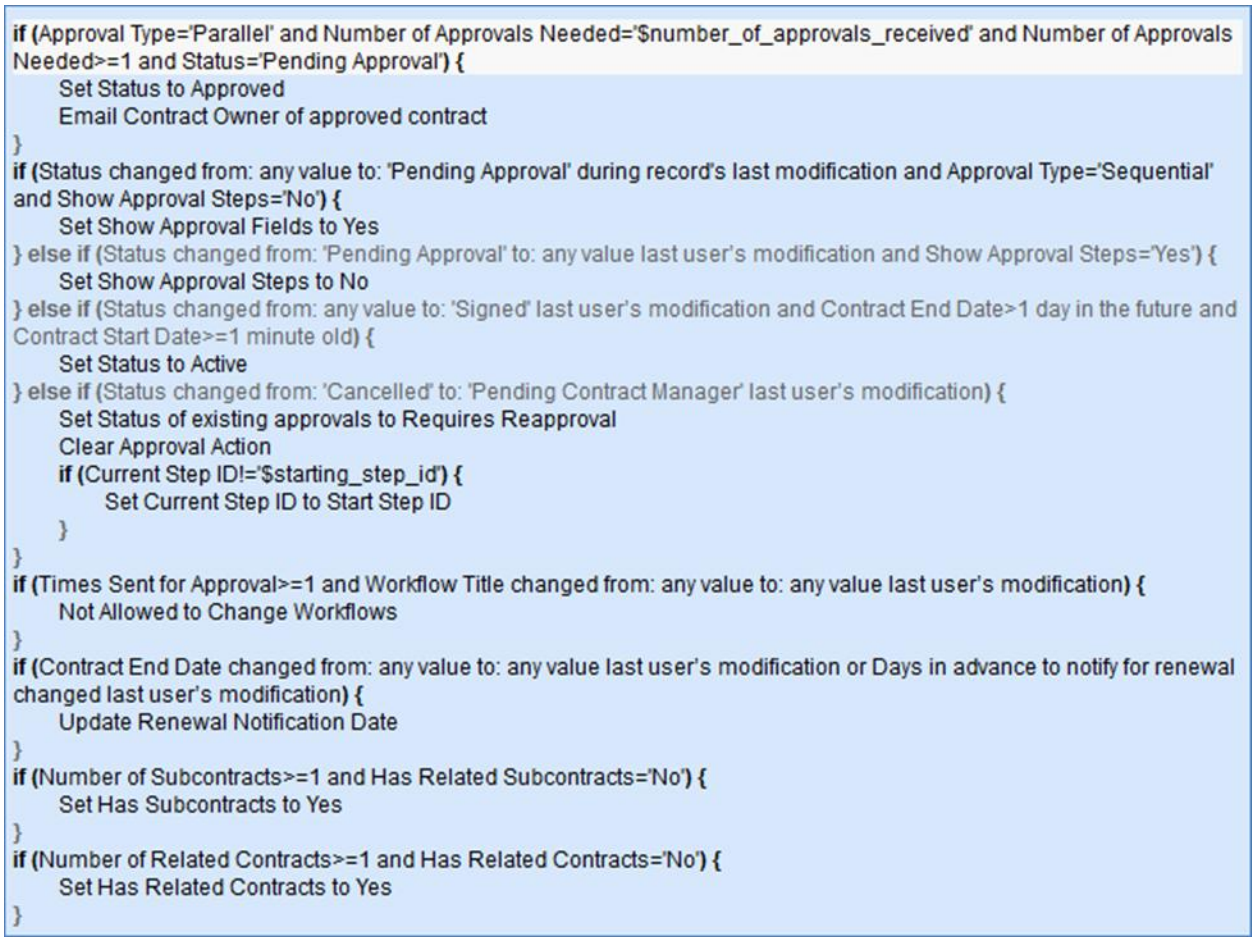
Runs: Upon edits by Web or API

Search Filter: None

Priority: 2

Actions:

I: All Edit Actions with API



```
if (Approval Type='Parallel' and Number of Approvals Needed='$number_of_approvals_received' and Number of Approvals
Needed>=1 and Status='Pending Approval') {
      Set Status to Approved
      Email Contract Owner of approved contract
}
if (Status changed from: any value to: 'Pending Approval' during record's last modification and Approval Type='Sequential'
and Show Approval Steps='No') {
      Set Show Approval Fields to Yes
} else if (Status changed from: 'Pending Approval' to: any value last user's modification and Show Approval Steps='Yes') {
      Set Show Approval Steps to No
} else if (Status changed from: any value to: 'Signed' last user's modification and Contract End Date>1 day in the future and
Contract Start Date>=1 minute old) {
      Set Status to Active
} else if (Status changed from: 'Cancelled' to: 'Pending Contract Manager' last user's modification) {
      Set Status of existing approvals to Requires Reapproval
      Clear Approval Action
      if (Current Step ID!='$starting_step_id') {
            Set Current Step ID to Start Step ID
      }
}
if (Times Sent for Approval>=1 and Workflow Title changed from: any value to: any value last user's modification) {
      Not Allowed to Change Workflows
}
if (Contract End Date changed from: any value to: any value last user's modification or Days in advance to notify for renewal
changed last user's modification) {
      Update Renewal Notification Date
}
if (Number of Subcontracts>=1 and Has Related Subcontracts='No') {
      Set Has Subcontracts to Yes
}
if (Number of Related Contracts>=1 and Has Related Contracts='No') {
      Set Has Related Contracts to Yes
}
```

TB: Notify of upcoming expirations (Disabled by Default)

Purpose: This rule controls the email notifications for upcoming expirations of contracts

Runs: At selected time intervals; every 2 Days at 3 AM

Search Filter: R: Renewal date is tomorrow

Priority: 0

Action:

E: Email Contract Owner about Renewal Date

Sent To: User in the Contract Manager field

Summary: Reminds the Contract Manager that the impending renewal date of a contract

TB: Daily Check for Start Date (Disabled by Default)

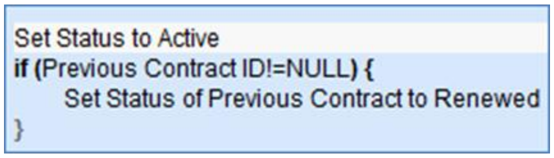
Purpose: This rule checks for contracts whose start date has arrived and changes the Status accordingly

Runs: At selected time intervals; every 2 days at 4 AM

Search Filter: R: Status is signed and Start Date is today or earlier

Priority: 4

I: Actions When Start Date Arrives



```
Set Status to Active
if (Previous Contract ID!=NULL) {
    Set Status of Previous Contract to Renewed
}
```

TB: Daily Check for Expiration Date (disabled)

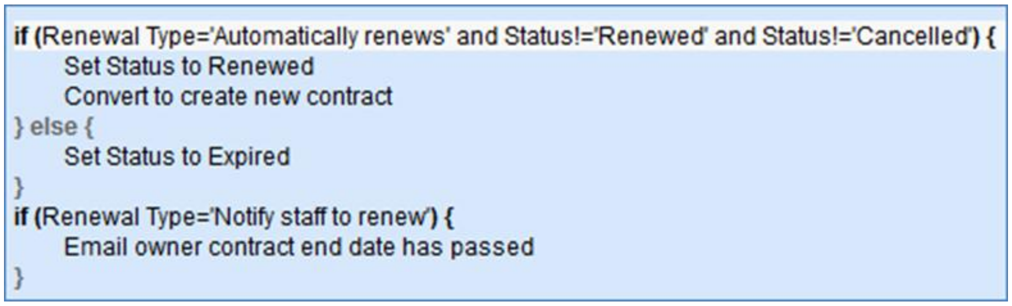
Purpose: This rule checks for contracts whose end date has arrived and takes the appropriate actions.

Runs: At selected time intervals; every 2 days at 2 AM

Search Filter: R: Contract Expired Today

Priority: 5

I: Handle Expired Contract



```
if (Renewal Type='Automatically renews' and Status!='Renewed' and Status!='Cancelled') {
    Set Status to Renewed
    Convert to create new contract
} else {
    Set Status to Expired
}
if (Renewal Type='Notify staff to renew') {
    Email owner contract end date has passed
}
```

Action Buttons

Cancel Contract

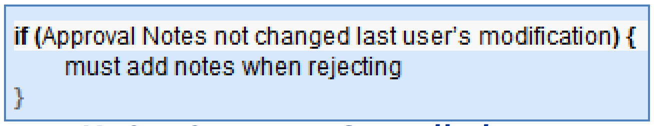
Purpose: This action button will cancel the contract and send an email to the contract requester. The button is only visible to Admins, Configuration Managers, Contract Owners, Service Managers, and Contract Managers.

On Click: Save record then execute actions.

Visibility: Only when Status is Active, Approved, Draft, Expired, Pending Approval, Renewed, or Signed.

Actions:

I: Validations for Cancellations and Reapprovals



```
if (Approval Notes not changed last user's modification) {
      must add notes when rejecting
}
```

U: Set Status to Cancelled

Updated Field: Status

Updated Value: Cancelled

E: Email Requestor Contract Cancelled

Email Template: Contract has been cancelled - Requester

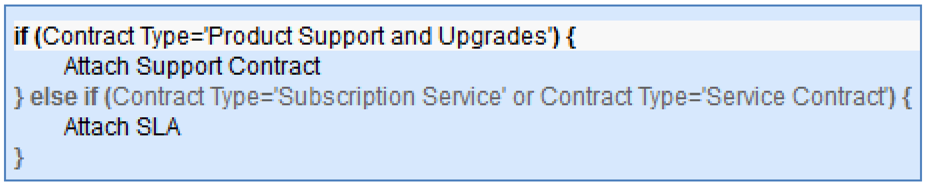
Create and Attach Contract

Purpose: On click, when Contract is for Product Support and Upgrades, runs an attached record action to create and Support Contract template. If the Contract is for Subscription Service or Service Contract, runs an attached record action to attach an SLA template; Visible only when Contract is for a Service Contract, Subscription Service or Product Support and Upgrades; Pop-up description: Use this button to create and attach a contract based on fields in this record.

On Click: Execute action

Visibility: Only when Contract Type is Product Support and Upgrades, Service Contract, or Subscription Service.

I: Create contract based on type of contract



```
if (Contract Type='Product Support and Upgrades') {
      Attach Support Contract
} else if (Contract Type='Subscription Service' or Contract Type='Service Contract') {
      Attach SLA
}
```

Action Specific Items: Attach Support Contract and Attach SLA work as advertised.

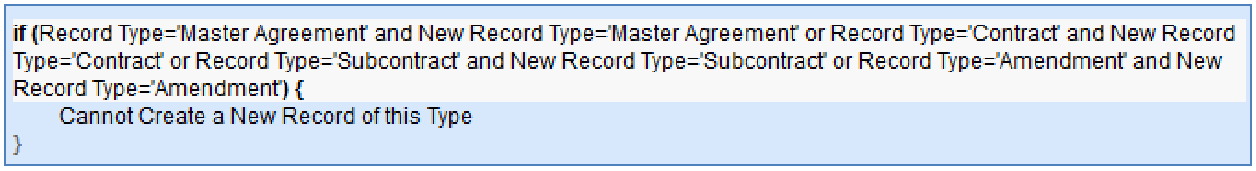
Create Related Contract

Purpose: This button creates a new contract, mapping several of the fields and showing the new contract on the screen. When the new contract is saved, any assets linked to the old contract will be linked to the new one. Visible only when the Status is Renewed.

On Click: Save record, then execute actions.

Actions:

I: New Contract Validations



```
if (Record Type='Master Agreement' and New Record Type='Master Agreement' or Record Type='Contract' and New Record Type='Contract' or Record Type='Subcontract' and New Record Type='Subcontract' or Record Type='Amendment' and New Record Type='Amendment') {
    Cannot Create a New Record of this Type
}
```

C: Convert to new contract

Conversion: Contract

Options: Interactive, showing user new record page

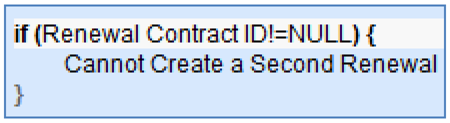
Create Renewal Contract

Purpose: Validates that another renewal contract doesn't already exist, sets certain fields correctly, then converts the current contract to create a new contract, then sets Is the New Contract a Renewal to No.

On Click: Execute actions.

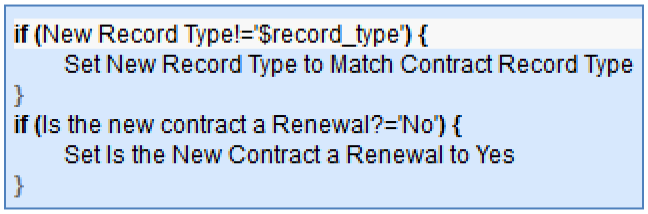
Visibility: Status is Active, Expired, or Signed

I: Contract Renewal Validations



```
if (Renewal Contract ID!=NULL) {
    Cannot Create a Second Renewal
}
```

I: Contract Renewal Actions



```
if (New Record Type!='$record_type') {
    Set New Record Type to Match Contract Record Type
}
if (Is the new contract a Renewal?='No') {
    Set Is the New Contract a Renewal to Yes
}
```

C: Convert to create new contract

Conversion: Contract

Options: Interactive, showing user the new record page

U: Set Is the New Contract a Renewal to No

Updated Field: Is the new contract a Renewal?

Updated Value: No

Mark As Signed

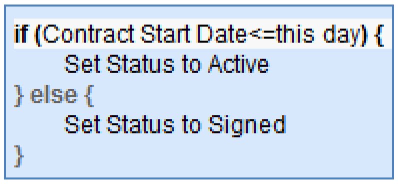
Purpose: Sets status to Active or Signed depending on if the Contract Start Date has passed or not.

On Click: Execute action.

Visibility: Status is Approved.

Actions:

I: Contract Signing Actions



```
if (Contract Start Date<=this day) {
        Set Status to Active
} else {
        Set Status to Signed
}
```

Review Completed

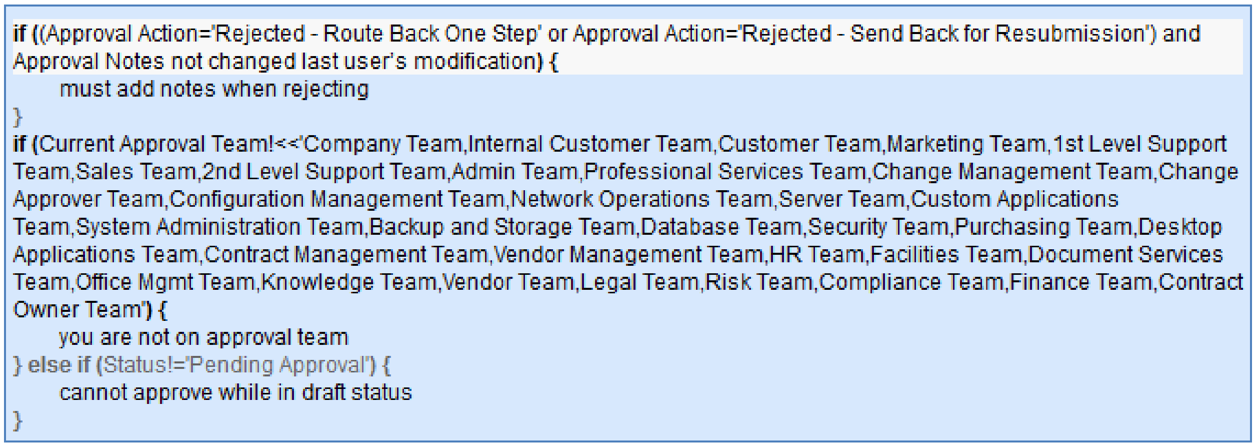
Purpose: Depending on the Approval Action, the button either marks the Contract as approved, sets it to the next approval step, or back to the preceding or first step, and sends notifications to the contract owner (in case of rejection) or to the next assigned team otherwise.

On Click: Save record, then execute actions.

Visibility: Show Approval Steps is Yes.

Actions:

I: Validations for review completed



```
if ((Approval Action='Rejected - Route Back One Step' or Approval Action='Rejected - Send Back for Resubmission') and
Approval Notes not changed last user's modification) {
        must add notes when rejecting
}
if (Current Approval Team!<<'Company Team,Internal Customer Team,Customer Team,Marketing Team,1st Level Support
Team,Sales Team,2nd Level Support Team,Admin Team,Professional Services Team,Change Management Team,Change
Approver Team,Configuration Management Team,Network Operations Team,Server Team,Custom Applications
Team,System Administration Team,Backup and Storage Team,Database Team,Security Team,Purchasing Team,Desktop
Applications Team,Contract Management Team,Vendor Management Team,HR Team,Facilities Team,Document Services
Team,Office Mgmt Team,Knowledge Team,Vendor Team,Legal Team,Risk Team,Compliance Team,Finance Team,Contract
Owner Team') {
        you are not on approval team
} else if (Status!='Pending Approval') {
        cannot approve while in draft status
}
```

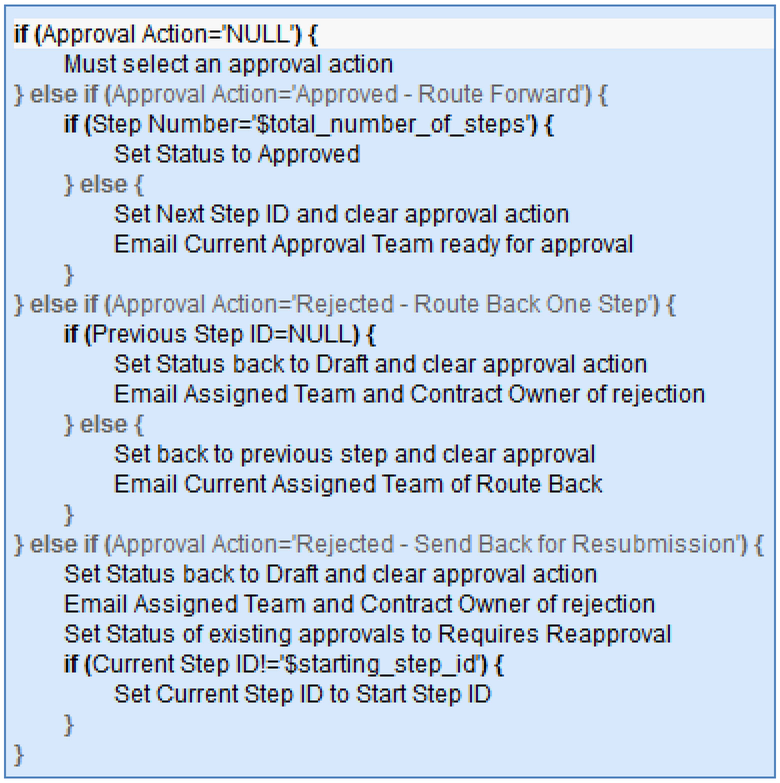
Note: Rule checking against $GLOBAL.my_teams

C: Create Approval Record

Conversion: Approval

Options: Silently with no confirmation, don't report errors.

I: Handle Review Complete Actions



```
if (Approval Action='NULL') {
    Must select an approval action
} else if (Approval Action='Approved - Route Forward') {
    if (Step Number='$total_number_of_steps') {
        Set Status to Approved
    } else {
        Set Next Step ID and clear approval action
        Email Current Approval Team ready for approval
    }
} else if (Approval Action='Rejected - Route Back One Step') {
    if (Previous Step ID=NULL) {
        Set Status back to Draft and clear approval action
        Email Assigned Team and Contract Owner of rejection
    } else {
        Set back to previous step and clear approval
        Email Current Assigned Team of Route Back
    }
} else if (Approval Action='Rejected - Send Back for Resubmission') {
    Set Status back to Draft and clear approval action
    Email Assigned Team and Contract Owner of rejection
    Set Status of existing approvals to Requires Reapproval
    if (Current Step ID!='$starting_step_id') {
        Set Current Step ID to Start Step ID
    }
}
```

Send Back for Reapproval

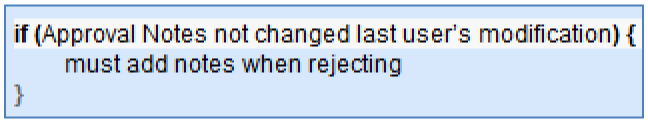
Purpose: Validates that notes are added, and then sets Step ID and Status to Start Step ID and Pending Approval or Requires Reapproval, as necessary. Emails Current Approval Team of reapproval.

On Click: Saves record then executes actions.

Visibility: Status is Approved.

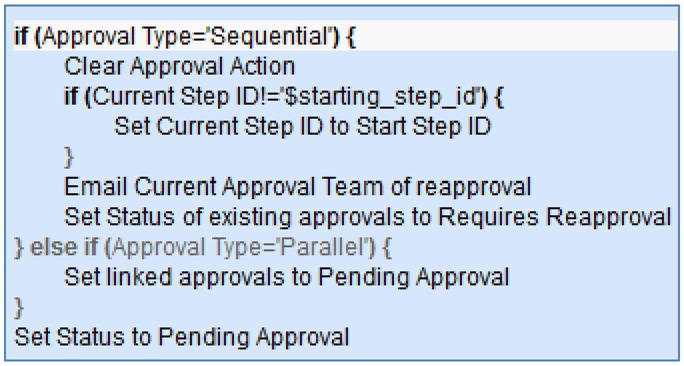
Actions:

I: Validations for Cancellations and Reapprovals



```
if (Approval Notes not changed last user's modification) {
    must add notes when rejecting
}
```

I: Send Back for Reapproval Actions



```
if (Approval Type='Sequential') {
    Clear Approval Action
    if (Current Step ID!='$starting_step_id') {
        Set Current Step ID to Start Step ID
    }
    Email Current Approval Team of reapproval
    Set Status of existing approvals to Requires Reapproval
} else if (Approval Type='Parallel') {
    Set linked approvals to Pending Approval
}
Set Status to Pending Approval
```

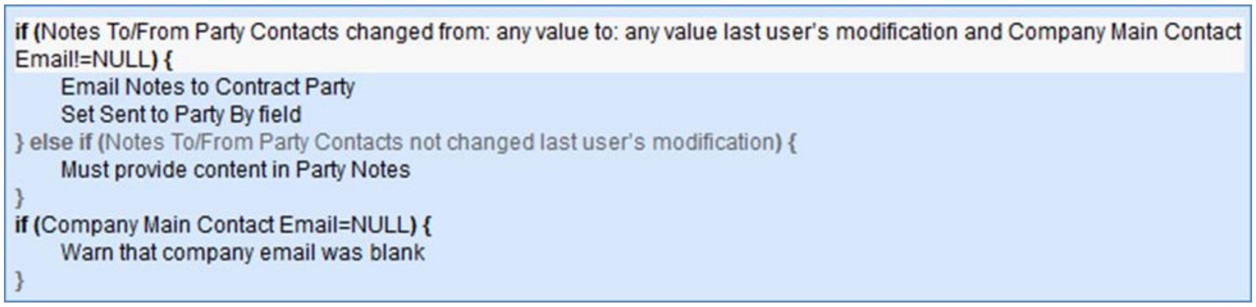
Send Contract to Party

Purpose: Validates that notes are provided to Party then emails notes to party. If party email does not exist, generates a popup that warns the user that the email was not sent because there was no address.

On Click: Save record, execute actions, then save again.

Actions:

I: Send to Party Actions



```
if (Notes To/From Party Contacts changed from: any value to: any value last user's modification and Company Main Contact Email!=NULL) {
    Email Notes to Contract Party
    Set Sent to Party By field
} else if (Notes To/From Party Contacts not changed last user's modification) {
    Must provide content in Party Notes
}
if (Company Main Contact Email=NULL) {
    Warn that company email was blank
}
```

Submit for Approval

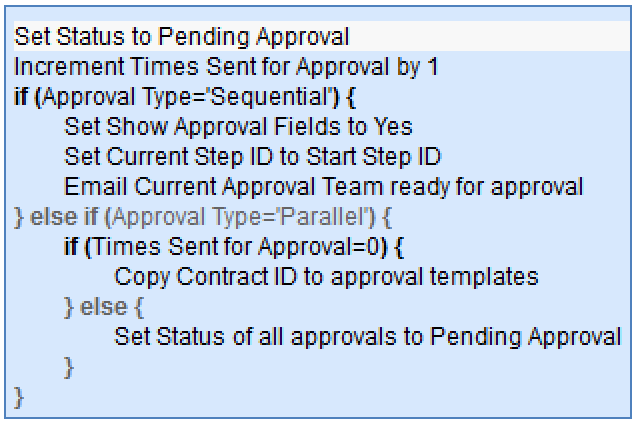
Purpose: On click, sets the User Status to Pending Approval then saves and closes the record; Visible only if Status equals Draft.

On Click: Saves record and executes action.

Visibility: Status is Draft or Pending Contract Manager.

Actions:

I: Run Submit for Approval Actions



```
Set Status to Pending Approval
Increment Times Sent for Approval by 1
if (Approval Type='Sequential') {
      Set Show Approval Fields to Yes
      Set Current Step ID to Start Step ID
      Email Current Approval Team ready for approval
} else if (Approval Type='Parallel') {
      if (Times Sent for Approval=0) {
            Copy Contract ID to approval templates
      } else {
            Set Status of all approvals to Pending Approval
      }
}
```

Submit for Review

Purpose: Emails Contract Manager Team of New Contract, then sets Status to Pending Contract Manager.

On Click: Saves record, executes, actions, then saves record again.

Visibility: Status is Cancelled or Draft.

Actions:

E: Email Contract Manager Team of New Contract

Email Template: Contract has been submitted for review.

U: Set Status to Pending Contract Manager

Updated Field: Status

Updated Value: Pending Contract Manager

Documents Table

Rules and Actions

All Edit Validations

Purpose: This rule handles all validation actions when a Document is edited.

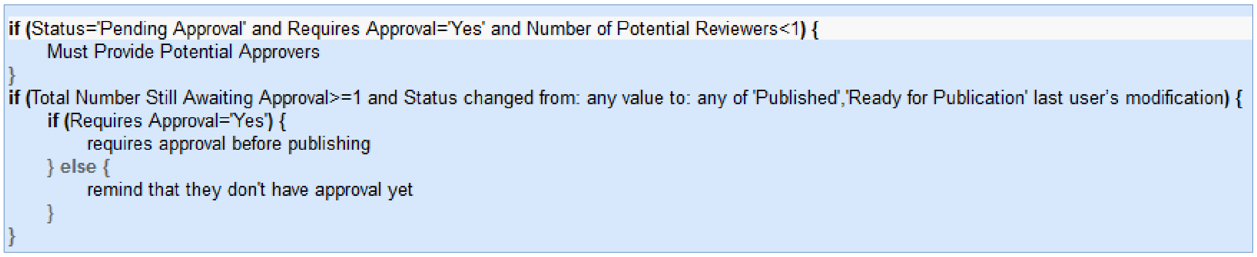
Runs: When edited by email or web.

Search Filter: None.

Priority: 5

Actions:

I: All Edit Validations



```
if (Status='Pending Approval' and Requires Approval='Yes' and Number of Potential Reviewers<1) {
    Must Provide Potential Approvers
}
if (Total Number Still Awaiting Approval>=1 and Status changed from: any value to: any of 'Published','Ready for Publication' last user's modification) {
    if (Requires Approval='Yes') {
        requires approval before publishing
    } else {
        remind that they don't have approval yet
    }
}
```

Edit: Last Approval Received (API Enabled)

Purpose: This rule updates the Status and notifies the Submitter and Document Manager that the document is ready for publication.

Runs: When edited by email, web or API.

Search Filter: R: Final Approval just received

Priority: 5

Actions:

U: Set Status to Ready for Publication

Updated Field: Status

Updated Value: Ready for Publication

E: Email Submitter Document Ready for Publication

E: Email Document Mgr that Document is Ready for Publication

Edit: All Edit Actions without API

Purpose: This rule handles all edit actions, including email notifications to Submitter and Document Manager of Status changes, and updates linked approval records.

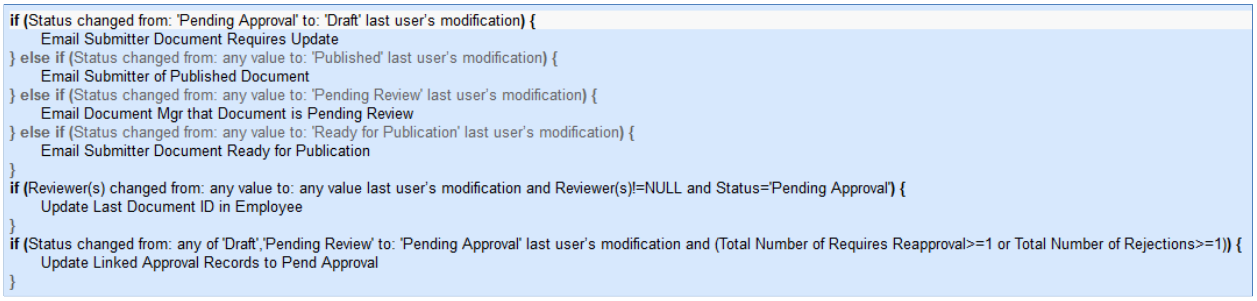
Runs: When edited by email or web.

Search Filter: None.

Priority: 1

Actions:

I: All Edit Actions



```
if (Status changed from: 'Pending Approval' to: 'Draft' last user's modification) {
    Email Submitter Document Requires Update
} else if (Status changed from: any value to: 'Published' last user's modification) {
    Email Submitter of Published Document
} else if (Status changed from: any value to: 'Pending Review' last user's modification) {
    Email Document Mgr that Document is Pending Review
} else if (Status changed from: any value to: 'Ready for Publication' last user's modification) {
    Email Submitter Document Ready for Publication
}
if (Reviewer(s) changed from: any value to: any value last user's modification and Reviewer(s)!=NULL and Status='Pending Approval') {
    Update Last Document ID in Employee
}
if (Status changed from: any of 'Draft','Pending Review' to: 'Pending Approval' last user's modification and (Total Number of Requires Reapproval>=1 or Total Number of Rejections>=1)) {
    Update Linked Approval Records to Pend Approval
}
```

Create: All Creation Actions

Purpose: This rule handles all creation actions, and emails the Document Manager if the initial Status is Pending Review.

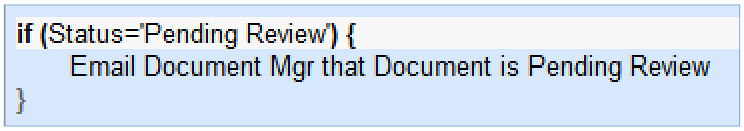
Runs: When created by email, web or API.

Search Filter: None.

Priority: 1

Actions:

I: All Creation Actions



```
if (Status='Pending Review') {
    Email Document Mgr that Document is Pending Review
}
```

Edit: Handle Rejection of an Approval (Web or API)

Purpose: This rule handles all actions if an Approval is rejected.

Runs: When edited by web or API.

Search Filter: R: Has one or more rejections and status is not draft

Priority: 4

Actions:

U: Set Status to Draft

Updated Field: Status

Updated Value: Draft

E: Email Submitter Document Requires Update

Sent To: Submitter

Template Name: Your Document Requires and Update

L: Set Document Rejected field to Yes in linked Approvals

Chain: Document ← Approvals through fields: Document ID, Document Title, etc.

U: Set Document Rejected to Yes

Updated Fields: Document Rejected, Requires Update (Approvals table)

Updated Value: Yes

Edit: Refresh Concurrent Approvals if Total Number of Approvals Changes (web, API)

Purpose: This rule handles updates to approval records when the Total Number of Approvals changes.

Runs: When created or edited by Web or API.

Search Filter: R: Total Number of Approvals Changed

Priority: 20

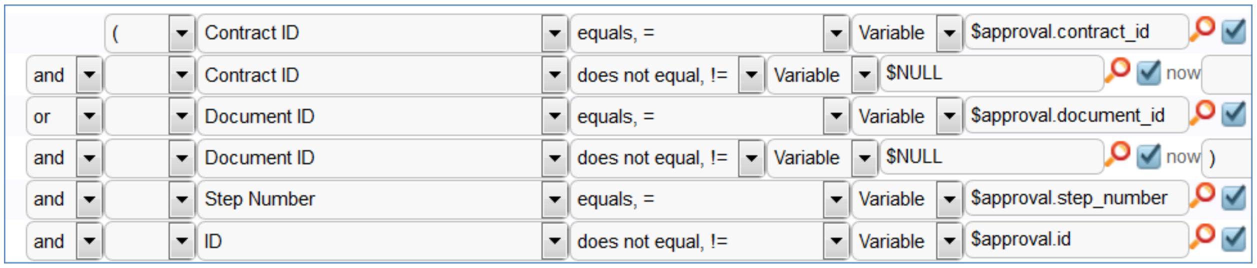
Actions:

U: Refresh Concurrent Approvals

Chain: Document ← Approvals through Document ID

Updated Field: Concurrent Approvals

Updated Value: The records that result from the search MT: Approvals for same contract or document with same step number



	(	Contract ID	equals, =	Variable	$approval.contract_id	✓	
and		Contract ID	does not equal, !=	Variable	$NULL	✓ now	
or		Document ID	equals, =	Variable	$approval.document_id	✓	
and		Document ID	does not equal, !=	Variable	$NULL	✓ now	)
and		Step Number	equals, =	Variable	$approval.step_number	✓	
and		ID	does not equal, !=	Variable	$approval.id	✓	

Action Buttons

Submit for Review

Purpose: This button will change the status of the record to Pending Review and save the record

Visibility: Only visible when Status is Draft or Retired

Submit for Approval

Purpose: This button will change the status of the record to Pending Approval following validation, set Last Document ID field in Employee record and save the record.

Visibility: Only visible when Status is Pending Review

Cancel

Purpose: This button will change the status of the record to Cancelled and save the record

Visibility: Only visible when Status is Draft

Publish

Purpose: This button will change the status of the record to Published and save the record

Visibility: Only visible when Status is Ready for Publication

Publish without Approval

Purpose: This button will change the status of the record to Published following validation and save the record

Visibility: Only visible when Status is Pending Review

Mark Retired

Purpose: This button will change the status of the record to Retired and save the record

Visibility: Only visible when Status is Published

Return to Submitter

Purpose: This button will change the status of the record to Draft, email the submitter to update the document and save the record

Visibility: Only visible when Status is Pending Review

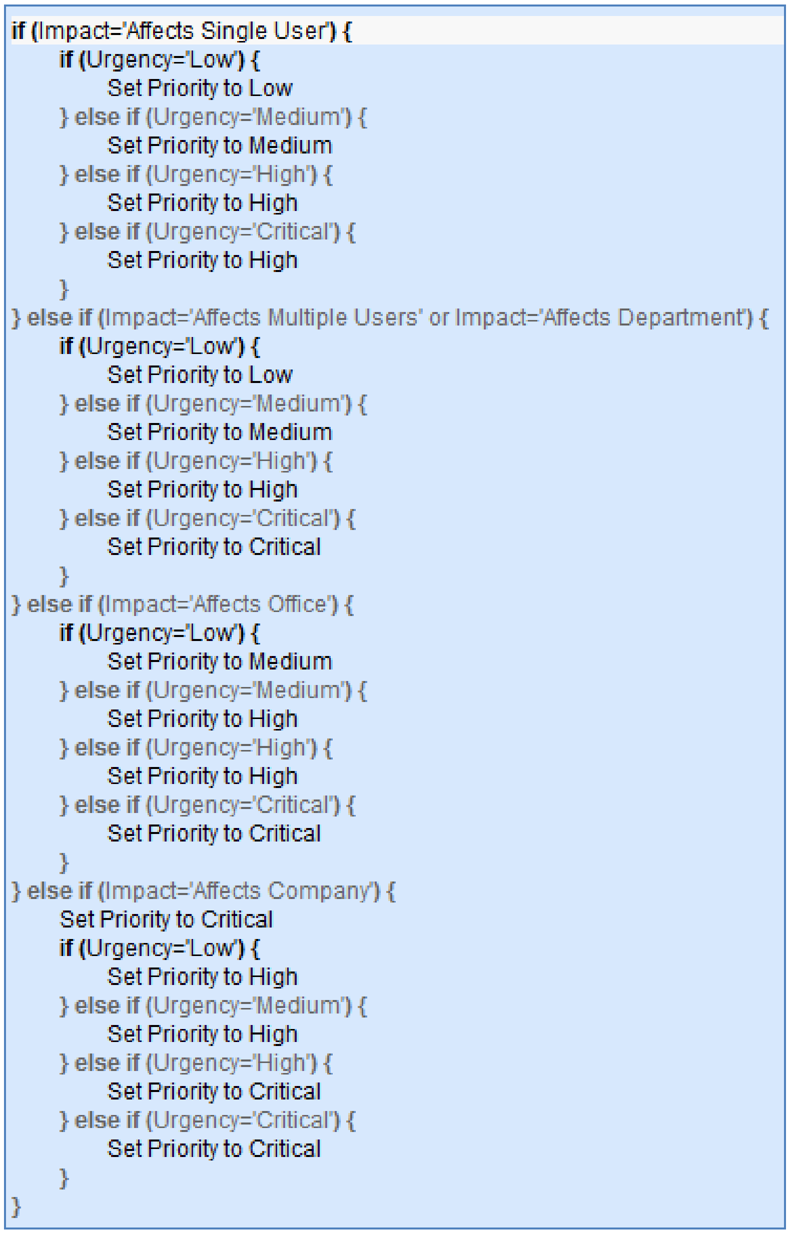
Incidents Table

Rules and Actions

Set Priority on New Tickets

Saved Search: R: Priority is Empty

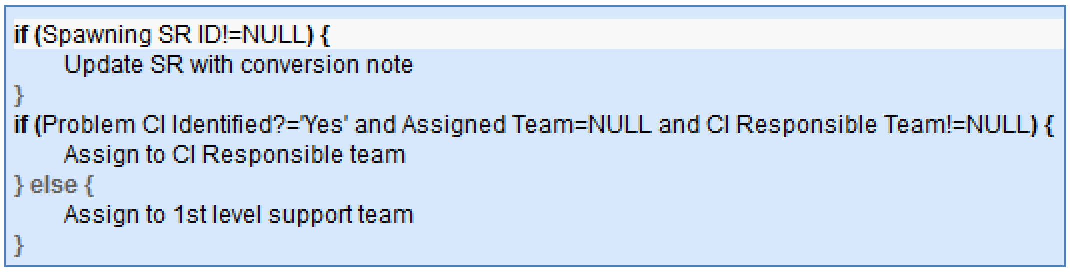
I: Set Priority



```
if (Impact='Affects Single User') {
        if (Urgency='Low') {
                Set Priority to Low
        } else if (Urgency='Medium') {
                Set Priority to Medium
        } else if (Urgency='High') {
                Set Priority to High
        } else if (Urgency='Critical') {
                Set Priority to High
        }
} else if (Impact='Affects Multiple Users' or Impact='Affects Department') {
        if (Urgency='Low') {
                Set Priority to Low
        } else if (Urgency='Medium') {
                Set Priority to Medium
        } else if (Urgency='High') {
                Set Priority to High
        } else if (Urgency='Critical') {
                Set Priority to Critical
        }
} else if (Impact='Affects Office') {
        if (Urgency='Low') {
                Set Priority to Medium
        } else if (Urgency='Medium') {
                Set Priority to High
        } else if (Urgency='High') {
                Set Priority to High
        } else if (Urgency='Critical') {
                Set Priority to Critical
        }
} else if (Impact='Affects Company') {
        Set Priority to Critical
        if (Urgency='Low') {
                Set Priority to High
        } else if (Urgency='Medium') {
                Set Priority to High
        } else if (Urgency='High') {
                Set Priority to Critical
        } else if (Urgency='Critical') {
                Set Priority to Critical
        }
}
```

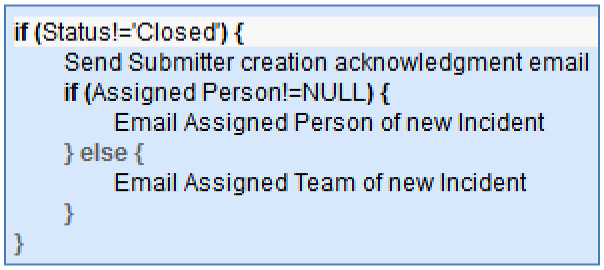
Incident - All Creation Actions

I: Conditional Creation Actions



```
if (Spawning SR ID!=NULL) {
        Update SR with conversion note
}
if (Problem CI Identified?='Yes' and Assigned Team=NULL and CI Responsible Team!=NULL) {
        Assign to CI Responsible team
} else {
        Assign to 1st level support team
}
```

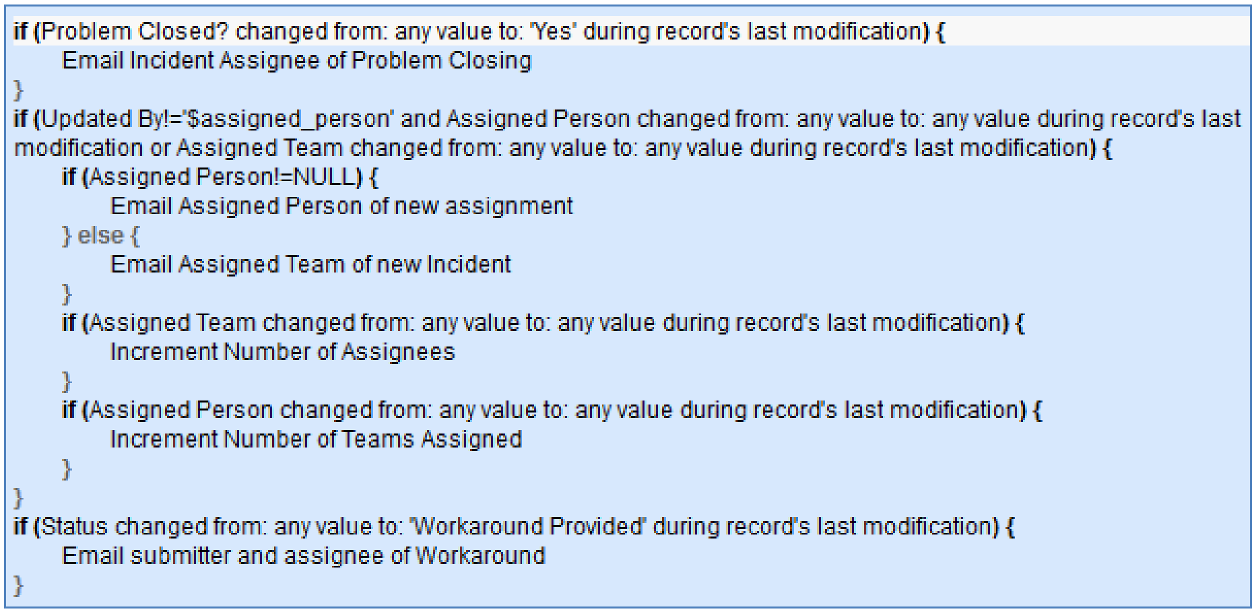
I: Notify Customer and Assignee



```
if (Status!='Closed') {
        Send Submitter creation acknowledgment email
        if (Assigned Person!=NULL) {
                Email Assigned Person of new Incident
        } else {
                Email Assigned Team of new Incident
        }
}
```

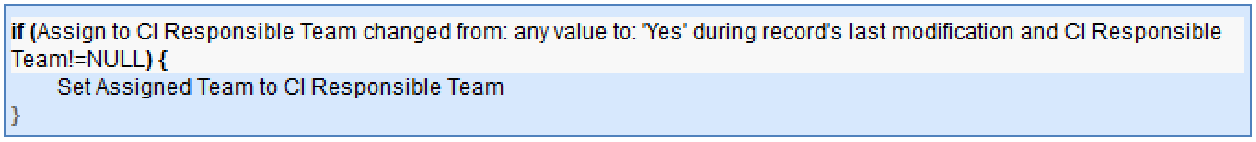
Incident - Edit actions (API Enabled)

I: Notification Actions for Incidents



```
if (Problem Closed? changed from: any value to: 'Yes' during record's last modification) {
        Email Incident Assignee of Problem Closing
}
if (Updated By!='$assigned_person' and Assigned Person changed from: any value to: any value during record's last
modification or Assigned Team changed from: any value to: any value during record's last modification) {
        if (Assigned Person!=NULL) {
                Email Assigned Person of new assignment
        } else {
                Email Assigned Team of new Incident
        }
        if (Assigned Team changed from: any value to: any value during record's last modification) {
                Increment Number of Assignees
        }
        if (Assigned Person changed from: any value to: any value during record's last modification) {
                Increment Number of Teams Assigned
        }
}
if (Status changed from: any value to: 'Workaround Provided' during record's last modification) {
        Email submitter and assignee of Workaround
}
```

I: Conditional Assignment Actions



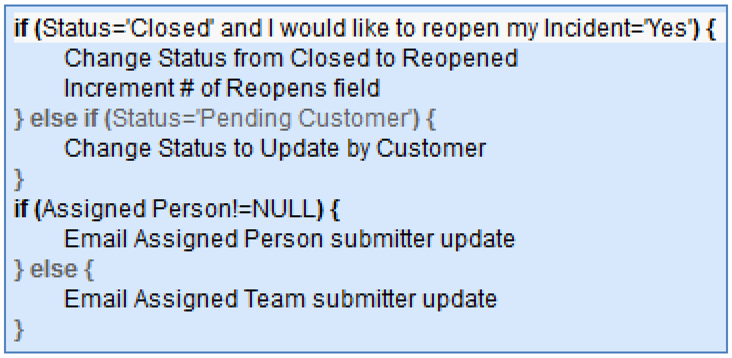
```
if (Assign to CI Responsible Team changed from: any value to: 'Yes' during record's last modification and CI Responsible
Team!=NULL) {
    Set Assigned Team to CI Responsible Team
}
```

Incident Customer Update Actions

Run upon Edit, no API

Saved Search: Updated by Submitter

I: Submitter Update Actions



```
if (Status='Closed' and I would like to reopen my Incident='Yes') {
    Change Status from Closed to Reopened
    Increment # of Reopens field
} else if (Status='Pending Customer') {
    Change Status to Update by Customer
}
if (Assigned Person!=NULL) {
    Email Assigned Person submitter update
} else {
    Email Assigned Team submitter update
}
```

Action Buttons

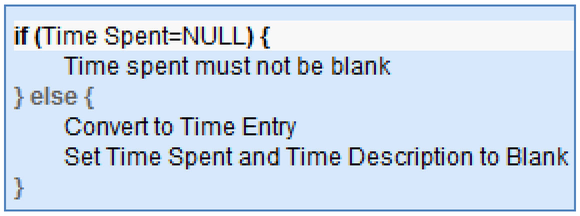
Add Time

Purpose: Triggers action to convert Time Spent and Time Description fields to a Time Entry record linked to the current Change Request record, then blanks out Time Spent and Time Description.

On Click: Save record, then execute action.

Actions:

I: Add Time



```
if (Time Spent=NULL) {
    Time spent must not be blank
} else {
    Convert to Time Entry
    Set Time Spent and Time Description to Blank
}
```

Convert to Change Request

Purpose: Saves changes to the Incident and maps some of its fields into a new Change Request.

On Click: Save record, then execute action.

Actions:

C: Convert to Change Request

Conversation: Change Request

Options: Interactive, showing user the new record page

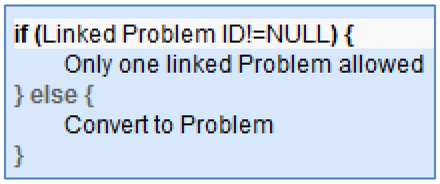
Convert to Problem

Purpose: Save before and after conversion, prevent conversion if the problem ID already has a value.

On Click: Save record, execute action, save record again.

Actions:

I: Convert to Problem if first conversion



```
if (Linked Problem ID!=NULL) {
        Only one linked Problem allowed
} else {
        Convert to Problem
}
```

Convert to Service Request

Purpose: Silently convert to service request.

On Click: Execute action, then save record.

Actions:

C: Convert to Service Request from Incident

Conversion: Service Request

Options: Silently with no confirmation

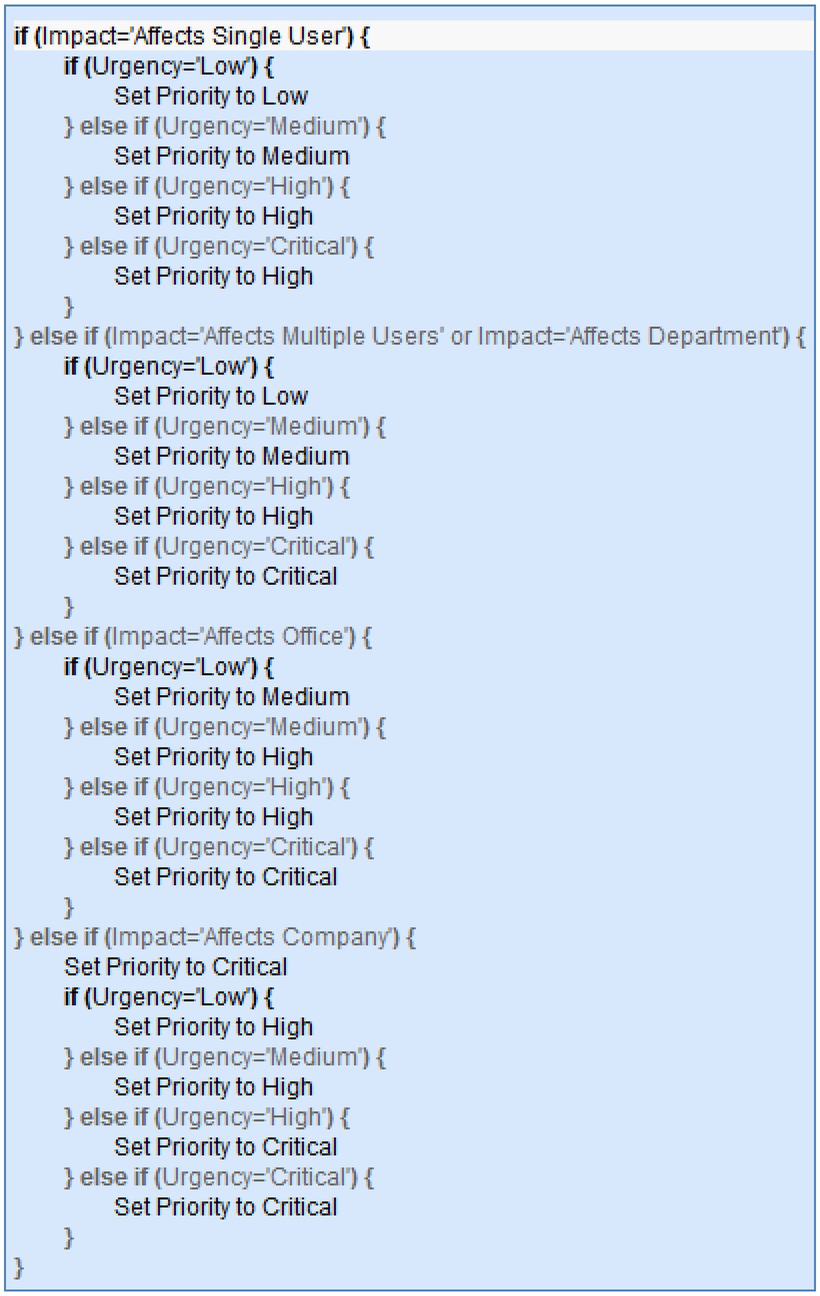
Set Priority

Purpose: Automatically sets priority based on impact and urgency.

On Click: Execute action.

Actions:

I: Set Priority



```
if (Impact='Affects Single User') {
        if (Urgency='Low') {
                Set Priority to Low
        } else if (Urgency='Medium') {
                Set Priority to Medium
        } else if (Urgency='High') {
                Set Priority to High
        } else if (Urgency='Critical') {
                Set Priority to High
        }
} else if (Impact='Affects Multiple Users' or Impact='Affects Department') {
        if (Urgency='Low') {
                Set Priority to Low
        } else if (Urgency='Medium') {
                Set Priority to Medium
        } else if (Urgency='High') {
                Set Priority to High
        } else if (Urgency='Critical') {
                Set Priority to Critical
        }
} else if (Impact='Affects Office') {
        if (Urgency='Low') {
                Set Priority to Medium
        } else if (Urgency='Medium') {
                Set Priority to High
        } else if (Urgency='High') {
                Set Priority to High
        } else if (Urgency='Critical') {
                Set Priority to Critical
        }
} else if (Impact='Affects Company') {
        Set Priority to Critical
        if (Urgency='Low') {
                Set Priority to High
        } else if (Urgency='Medium') {
                Set Priority to High
        } else if (Urgency='High') {
                Set Priority to Critical
        } else if (Urgency='Critical') {
                Set Priority to Critical
        }
}
```

Insurance Certificates Table

Rules and Actions

TB: Handle Insurance Certificate Expiration (Disabled by Default)

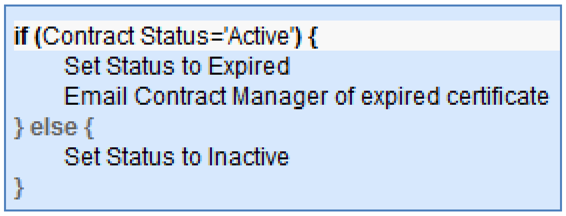
Purpose: Sets the status to Expired for contracts that were Active and past their expiration dates, and catches any other contracts past expiration and sets their status to Inactive.

Runs: Every 2 days at 2 AM

Saved Search: Valid Certificates Expiring Today

Actions:

I: Handle Certs Expiring Today



```
if (Contract Status='Active') {
      Set Status to Expired
      Email Contract Manager of expired certificate
} else {
      Set Status to Inactive
}
```

TB: Update Contracts (daily) (disabled)

Purpose: This rule updates the linked field of Contracts to find all contracts related to the insurance certificate.

Runs: Daily at 1:00 AM

Search Filter: R: Status is Valid

Priority: 0

Action:

U: Set Default Contracts

Updated Field: Contracts

Updated Value: Use the Insurance Certificate to Contract: Status is greater than or equal to Active and Company matches search, and choose all records**.**

Items Requested Table

Action Buttons

Assign to User

Purpose: Update Configuration Item record with appropriate data.

On Click: Execute action, then save record and open for editing

Action:

L: Assign to User – linked update of CI

Chain: Item Requested/CI Name -> Configuration Item

Linked Action: U: Copy Req. Login to CI User Login, Update Status and Ownership

U: Copy Req. Login to CI User Login, Update Status and Ownership

Updated Fields: Ownership, Status, User Login

Updated Values: Assigned to an Internal User, Pending Install, "$item_requested.requester_login"

Leads Table

Rules and Actions

Lead Conversion

Purpose: Runs conversion of Lead to Company, Opportunity, and Contact records when Status of Lead record is set to "Converted."

Runs: Edit by Email, Web or API

Search Filter: Just Converted

Priority: 2

Actions:

C: Convert to Company, Opportunity, Contact

Conversions: People, Opportunity, Company

Options: Silently with no confirmation

Campaign actions

Purpose: Automatically sets campaign name from name of lead.

Runs: Create by Email, Web, API

Search Filter: R: No campaign but user entry

Priority: 1

Actions:

U: Update Campaign from How did you hear

Field Changed: Campaign Name

Value Changed to: $lead_source

Action Buttons

Find Local Hotels

Purpose: Searches Google Maps for hotels near the lead.

On Click: Opens URL: "http://maps.google.com/maps?f=q&q=hotels" + "+loc:+" + urlEncode($street_address_1) + ",+" + urlEncode($city) + ",+" + $us_state

Map It

Purpose: Searches Google Maps to find the location of the lead.

On Click: Open URL: "http://maps.google.com/maps?f=q&q=" + urlEncode($street_address_1) + ",+" + urlEncode($city) + ",+" + $us_state

People Table

Rules and Actions

Edit: Approval Record Creation

Purpose: Performs conversion of Employee records to Approval records for documents approval and updates fields keeping track of Approvals that have been created.

Runs: Edit by Web or API

Search Filter: R: Last Doc ID Text Changed and All Docs does not contain Last Doc ID

Priority: 1

Actions:

C: Convert Employee to New Approval

Conversion: Approval

Options: Silently with no confirmation

U: Add to All Document IDs Converted

Fields Changed: All Document IDs Converted

Value Changed to: $last_document_id_text+”,”

Action Buttons

Export to Outlook

Purpose: Downloads the user's information to vcf format which will

Allow it to be saved and opened in MS Outlook

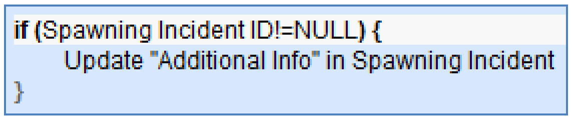
On Click: Open URL: "/gui2/vCard;jsessionid=?id=" + urlEncode($id)

Problems Table

Rules and Actions

Rule: Problem - Creation actions

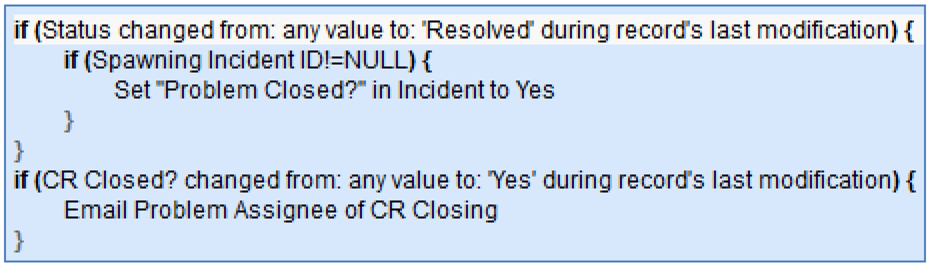
I: Conditional Creation Actions



```
if (Spawning Incident ID!=NULL) {
    Update "Additional Info" in Spawning Incident
}
```

Rule: Problem - Edit actions (API Enabled)

I: Notification actions



```
if (Status changed from: any value to: 'Resolved' during record's last modification) {
    if (Spawning Incident ID!=NULL) {
        Set "Problem Closed?" in Incident to Yes
    }
}
if (CR Closed? changed from: any value to: 'Yes' during record's last modification) {
    Email Problem Assignee of CR Closing
}
```

Action Buttons

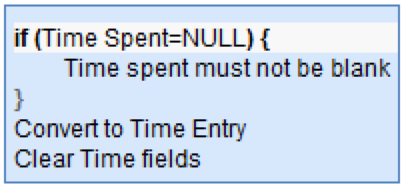
Add Time

Purpose: Triggers action to convert Time Spent and Time Description fields to a Time Entry record linked to the Service Request, then blanks the fields.

On Click: Save record, then execute action.

Actions:

I: Add Time



```
if (Time Spent=NULL) {
      Time spent must not be blank
}
Convert to Time Entry
Clear Time fields
```

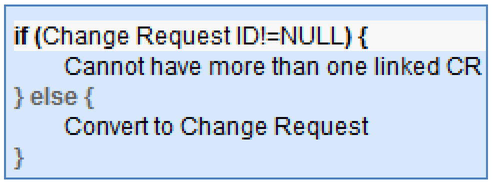
Convert to Change Request

Purpose: Validates the problem has no change request associated with it, then converts to a change request.

On Click: Execute action, then save record.

Action:

I: Convert to CR (if first conversion)



```
if (Change Request ID!=NULL) {
      Cannot have more than one linked CR
} else {
      Convert to Change Request
}
```

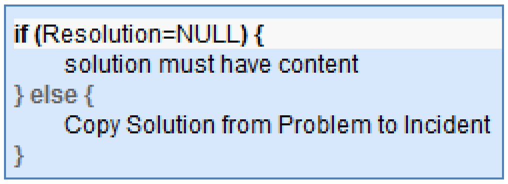
Copy Solution to Incident and Close It

Purpose: Validates a resolution exists, then copies the resolution from the problem to a linked incident.

On Click: Execute action.

Action:

I: Copy Solution field to related Incidents



```
if (Resolution=NULL) {
      solution must have content
} else {
      Copy Solution from Problem to Incident
}
```

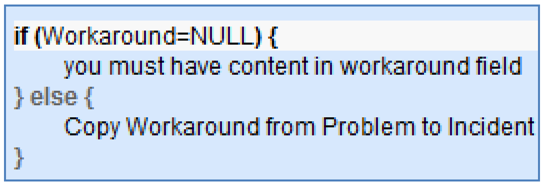
Copy Workaround to Incident

Purpose: Copies the contents of the Workaround field in the Problem record to the Resolution fields of linked Incidents. Sets the "Workaround Provided?" field in the Problem record to Yes.

On Click: Execute actions.

Actions:

I: Update Linked Incident with Workaround



```
if (Workaround=NULL) {
    you must have content in workaround field
} else {
    Copy Workaround from Problem to Incident
}
```

U: Set Workaround Provided to yes

Updated Field: Workaround Provided

Updated Value: Yes

End Diagnosis Clock

Purpose: Sets Diagnosis Finish Time to “now.”

On Click: Execute action.

Action:

U: Set Diagnosis Finish Time to Now

Updated Field: Diagnosis Finish Time

Updated Value: NOW()

Set Solution Finish Time

Purpose: Sets Solution Finish Time to NOW().

On Click: Execute action.

Action:

U: Set Solution Finish Time to Now

Updated Field: Solution Finish Time

Updated Value: NOW()

Set Solution Start Time

Purpose: Sets Solution Start Time to Now

On Click: Set Solution Start Time to Now

Action:

U: Set Solution Start Time to Now

Updated Field: Solution Start Time

Updated Value: NOW()

Start Diagnosis Clock

Purpose: Sets Diagnosis Start Time to NOW().

On Click: Execute action.

Action:

U: Set Diagnosis Start Time to Now

Updated Field: Diagnosis Start Time

Updated Value: NOW()

Projects Table

Rules and Actions

Edit by Web: All edit actions: trigger tasks, cancel tasks when cancelled, etc.

Purpose: Validates, triggers, and cancels tasks.

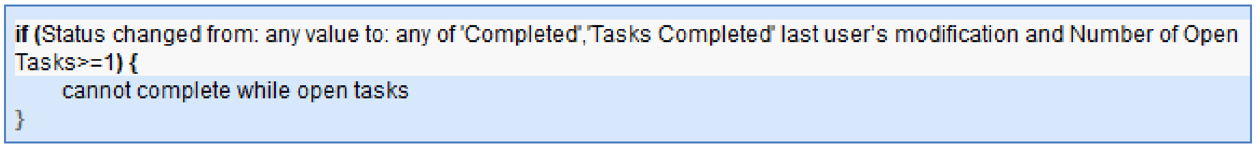
Runs: On Edit by Web.

Search Filter: None.

Priority: 2

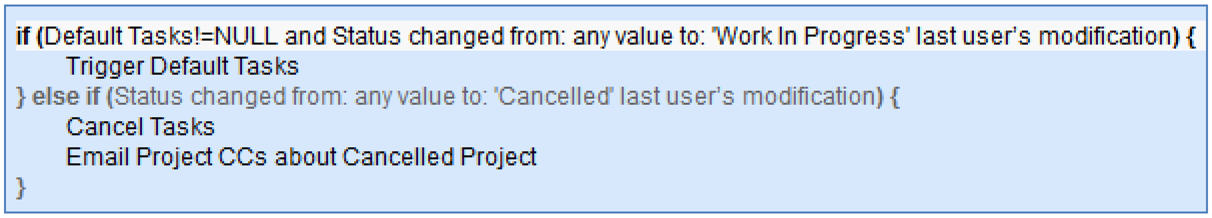
Actions:

I: All Edit Validations



```
if (Status changed from: any value to: any of 'Completed','Tasks Completed' last user's modification and Number of Open
Tasks>=1) {
      cannot complete while open tasks
}
```

I: All Edit Actions



```
if (Default Tasks!=NULL and Status changed from: any value to: 'Work In Progress' last user's modification) {
      Trigger Default Tasks
} else if (Status changed from: any value to: 'Cancelled' last user's modification) {
      Cancel Tasks
      Email Project CCs about Cancelled Project
}
```

TB: notify Project Manager if hours exceed authorized hours

Puprose: Checks on authorized hours and hours worked and sends notification if authorized hour are exceeded.

Runs: Every day.

Search Filter: R: total hours for client project exceeds authorized hours

Priority: 1

Actions:

U: Set Status to Assigned

Fields Updated: Status

Updated To: Assigned

E: Email PM project has exceeded hours

Email Template: Project $formula($project_name) has exceeded authorized hours

Edit by Web or API: Handle completion or addition of tasks and status changes

Purpose: Opens or completes projects depending on number and state of tasks linked to project.

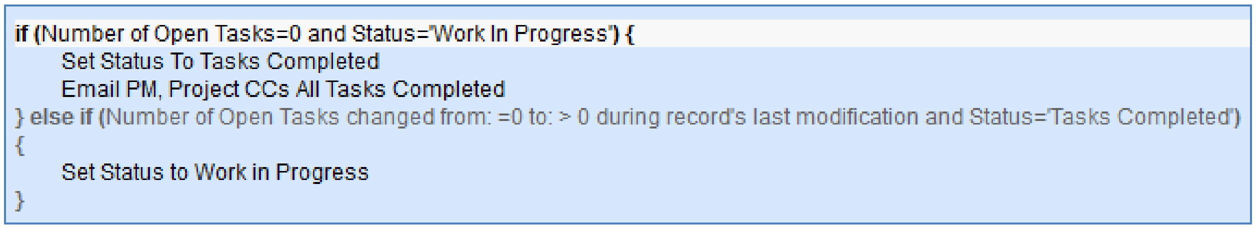
Runs: Edits by Web or API.

Search Filter: R: Has Tasks and the number of open tasks changed

Priority: 1

Actions:

I: Handle completion or adding tasks



```
if (Number of Open Tasks=0 and Status='Work In Progress') {
    Set Status To Tasks Completed
    Email PM, Project CCs All Tasks Completed
} else if (Number of Open Tasks changed from: =0 to: > 0 during record's last modification and Status='Tasks Completed')
{
    Set Status to Work in Progress
}
```

Create: All Project creation actions – trigger tasks

Purpose: Notify Project Manager and CCs of new project, and begin default tasks.

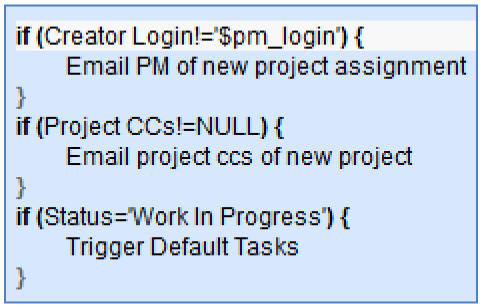
Runs: Creation by Email, Web, or API:

Search Filter: None.

Priority: 1

Actions:

I: All project create actions



```
if (Creator Login!='$pm_login') {
    Email PM of new project assignment
}
if (Project CCs!=NULL) {
    Email project ccs of new project
}
if (Status='Work In Progress') {
    Trigger Default Tasks
}
```

Project Types Table

Rules and Actions

Create/Edit by Web or API: Update Has Task Templates

Purpose: Checks to see if Task Templates exist in this Project Type and sets the flag field Has Task Templates to Yes if so.

Runs: Create/Edit including rules by Web or API

Search Filter: R: Has task templates is no and number of templates is 1 or more

Priority: 10

Actions:

U: Set Has Task Templates to Yes

Field Changed: Has Task Templates

Value Changed to: Yes

Purchase Orders Table

Action Buttons

Calculate Value

Purpose: Multiplies the Hours by the Rate, and puts the result in the field "Total Value".

On Click: Execute action.

Action:

*U: Total Value = Hours * Rate*

Updated Field: Total Value

Updated Value: $hours * $rate

Purchase Requests Table

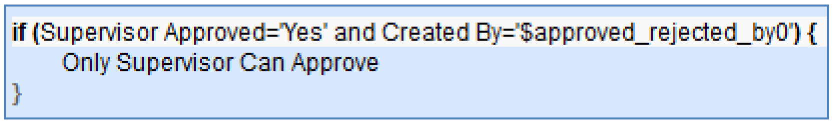
Rules and Actions

Rule: Purchase Request - New Request Actions

Condition and Schedule

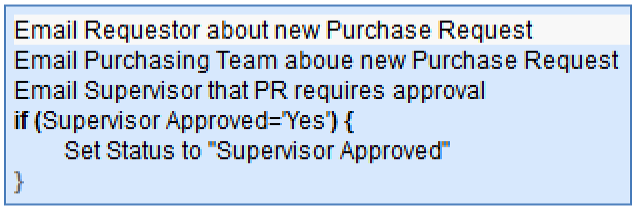
When a Purchase Request is Created by Web, Priority 1

I: Validation for New Requests



```
if (Supervisor Approved='Yes' and Created By='$approved_rejected_by0') {
      Only Supervisor Can Approve
}
```

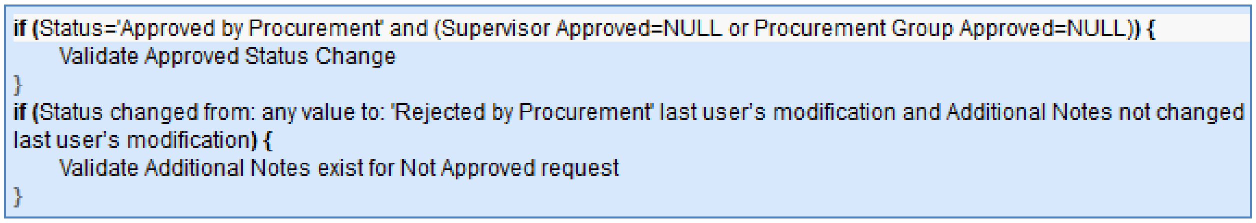
I: New Purchase Request Actions



```
Email Requestor about new Purchase Request
Email Purchasing Team aboue new Purchase Request
Email Supervisor that PR requires approval
if (Supervisor Approved='Yes') {
      Set Status to "Supervisor Approved"
}
```

Rule: Purchase Request - Edit Validation Actions

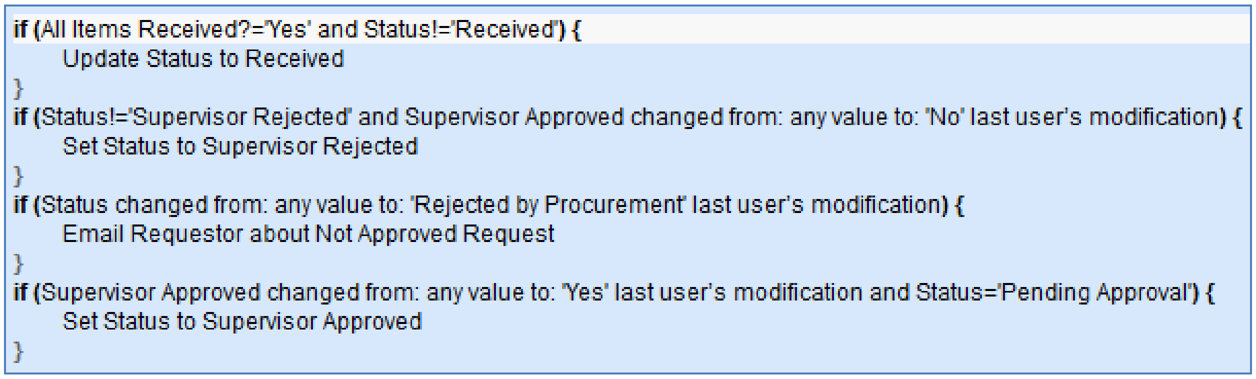
I: Actions Based on Edit Validations



```
if (Status='Approved by Procurement' and (Supervisor Approved=NULL or Procurement Group Approved=NULL)) {
      Validate Approved Status Change
}
if (Status changed from: any value to: 'Rejected by Procurement' last user's modification and Additional Notes not changed
last user's modification) {
      Validate Additional Notes exist for Not Approved request
}
```

Rule: Purchase Request - All Staff Edits

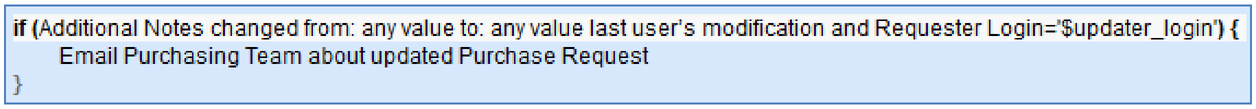
I: Staff Edit Actions



```
if (All Items Received?='Yes' and Status!='Received') {
      Update Status to Received
}
if (Status!='Supervisor Rejected' and Supervisor Approved changed from: any value to: 'No' last user's modification) {
      Set Status to Supervisor Rejected
}
if (Status changed from: any value to: 'Rejected by Procurement' last user's modification) {
      Email Requestor about Not Approved Request
}
if (Supervisor Approved changed from: any value to: 'Yes' last user's modification and Status='Pending Approval') {
      Set Status to Supervisor Approved
}
```

Rule: Purchase Request - All End User Edits

I: Actions Based on End User Edits



```
if (Additional Notes changed from: any value to: any value last user's modification and Requester Login='$updater_login') {
      Email Purchasing Team about updated Purchase Request
}
```

Service Requests Table

Rules and Actions

Rule: Create: All new Service Request actions

Purpose: This assigns the SR to the correct team, based on the CI or the Service responsible team. It sends an acknowledgement email to the customer and emails the assigned person or team.

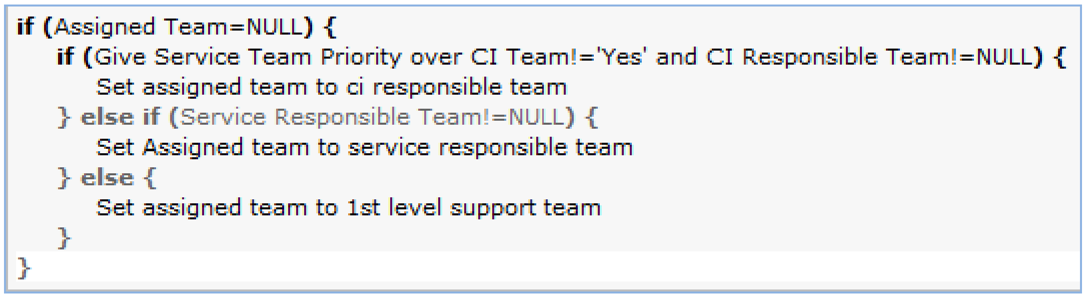
Runs: On creation by Email, Web, or API

Search Filter: None

Priority: 2

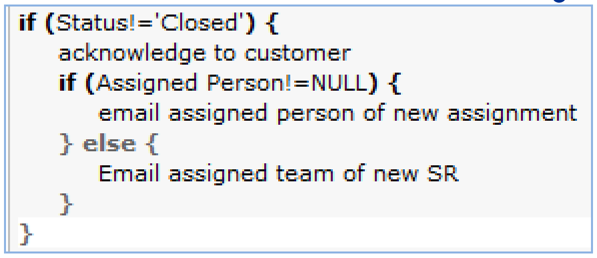
Actions:

I: Assign Service Request



```
if (Assigned Team=NULL) {
    if (Give Service Team Priority over CI Team!='Yes' and CI Responsible Team!=NULL) {
        Set assigned team to ci responsible team
    } else if (Service Responsible Team!=NULL) {
        Set Assigned team to service responsible team
    } else {
        Set assigned team to 1st level support team
    }
}
```

I: Send emails to customer and assignee



```
if (Status!='Closed') {
    acknowledge to customer
    if (Assigned Person!=NULL) {
        email assigned person of new assignment
    } else {
        Email assigned team of new SR
    }
}
```

Rule: Edit: All Customer Update Actions

Purpose:

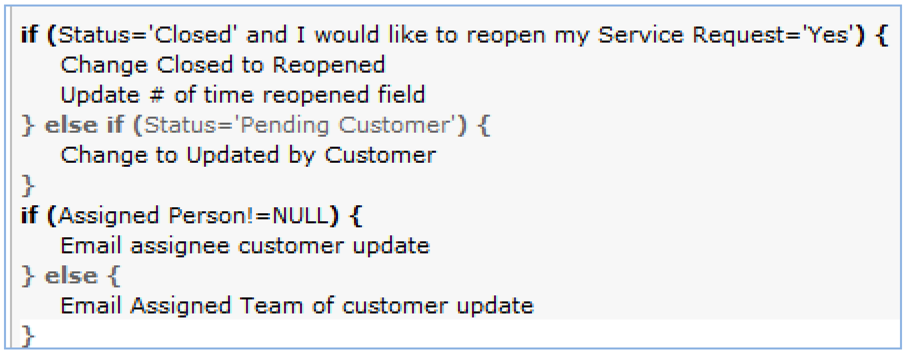
Runs: On edit by email or web

Search Filter: Updated by Customer (Updated by=Customer Name and Submitter team contains customer)

Priority: 1

Actions:

I: User Update Actions



```
if (Status='Closed' and I would like to reopen my Service Request='Yes') {
    Change Closed to Reopened
    Update # of time reopened field
} else if (Status='Pending Customer') {
    Change to Updated by Customer
}
if (Assigned Person!=NULL) {
    Email assignee customer update
} else {
    Email Assigned Team of customer update
}
```

Rule: All edit validations and other actions

Purpose: Handles validations – if setting status to Pending Customer, must add notes. If changing the status to Closed, all tasks must be completed and you must enter text in the Resolution field.

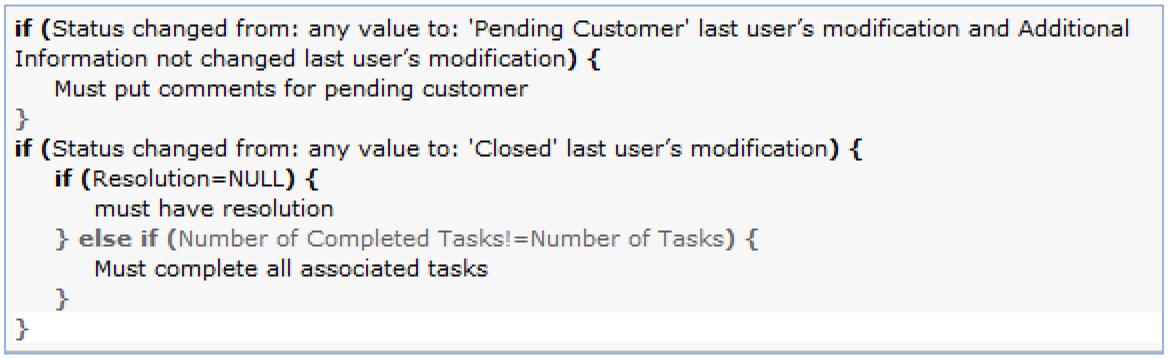
Runs: On Edit by Web only

Search Filter: None

Priority: 2

Actions:

I: All edit validations by staff



```
if (Status changed from: any value to: 'Pending Customer' last user's modification and Additional
Information not changed last user's modification) {
    Must put comments for pending customer
}
if (Status changed from: any value to: 'Closed' last user's modification) {
    if (Resolution=NULL) {
        must have resolution
    } else if (Number of Completed Tasks!=Number of Tasks) {
        Must complete all associated tasks
    }
}
```

Rule: Edit: Tasks Just Completed

Purpose: Notifies the assigned person or team when all tasks have been completed.

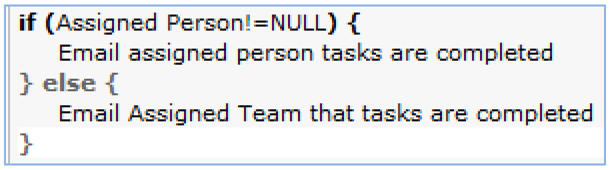
Runs: On Edit by Web or API

Search Filter: R: Completed Tasks just changed to all

Priority: 2

Actions:

I: Notify Assigned team or person of completed tasks



```
if (Assigned Person!=NULL) {
    Email assigned person tasks are completed
} else {
    Email Assigned Team that tasks are completed
}
```

Rule: Edit: Status Change Actions

Purpose: Sets the value for whether the SR was solved within the SLA to No if it exceeded the SLA time; sets the I would like to reopen my request to No. If the Assigned person was blank, sets it to the person who closed the request.

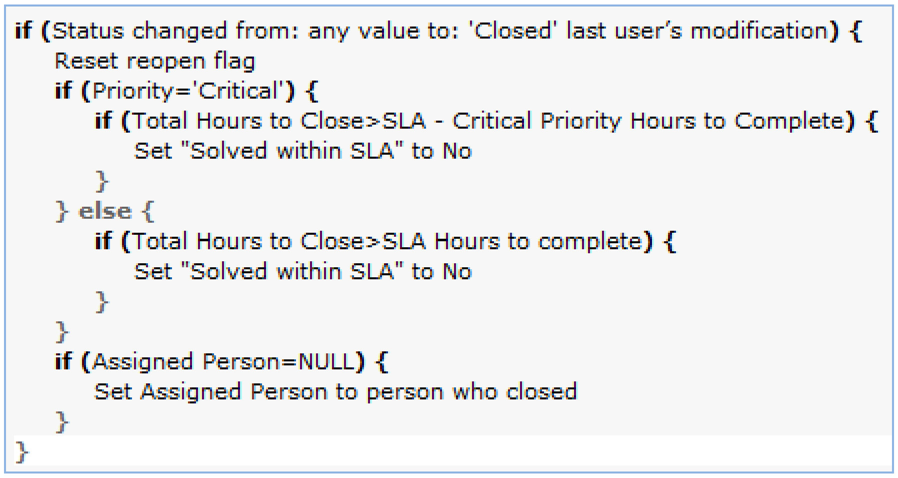
Runs: On Edit by Web only

Search Filter: R: Status changed last user mod

Priority: 3

Actions:

I: All Status Change Actions



```
if (Status changed from: any value to: 'Closed' last user's modification) {
    Reset reopen flag
    if (Priority='Critical') {
        if (Total Hours to Close>SLA - Critical Priority Hours to Complete) {
            Set "Solved within SLA" to No
        }
    } else {
        if (Total Hours to Close>SLA Hours to complete) {
            Set "Solved within SLA" to No
        }
    }
    if (Assigned Person=NULL) {
        Set Assigned Person to person who closed
    }
}
```

Rule: Assigned team or person changed

Purpose: Notifies the new assignee if the assigned person or team changes

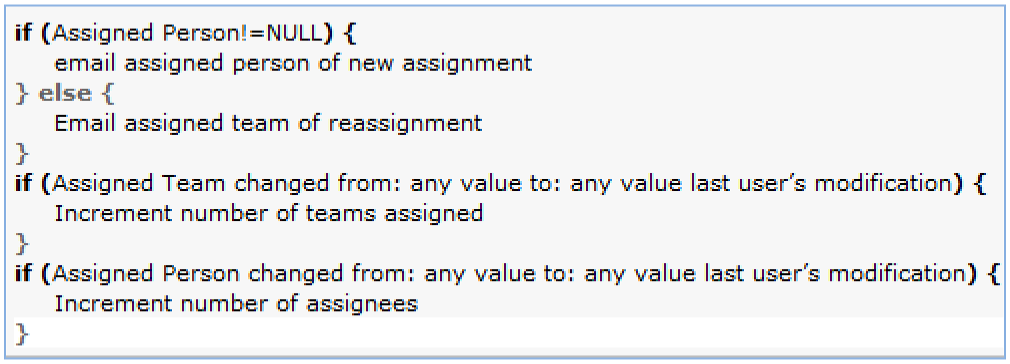
Runs: On Edit by Email, Web, or API

Search Filter: Assigned person or team changed

Priority: 2

Actions:

I: Notify team or person of new assignment



```
if (Assigned Person!=NULL) {
    email assigned person of new assignment
} else {
    Email assigned team of reassignment
}
if (Assigned Team changed from: any value to: any value last user's modification) {
    Increment number of teams assigned
}
if (Assigned Person changed from: any value to: any value last user's modification) {
    Increment number of assignees
}
```

Action Buttons

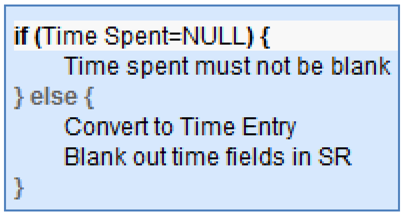
Add Time

Purpose: Triggers conversion mapping Time Spent and Time Description fields onto a new Time Entry record linked to this Task then blanks those fields.

On Click: Execute action.

Action:

I: Validate and Create Time Entry



```
if (Time Spent=NULL) {
    Time spent must not be blank
} else {
    Convert to Time Entry
    Blank out time fields in SR
}
```

Clone

Purpose: Makes a copy of the Service Request, currently unused.

On Click: Execute action.

Action:

C: Clone to Service Request

Conversion: Service Request

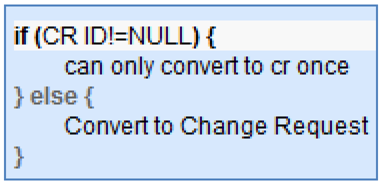
Convert to Change Request

Purpose: Convert to create a new Change Request, then save and close record.

On Click: Execute action then save record.

Action:

I: Convert to Change Request If First One



```
if (CR ID!=NULL) {
    can only convert to cr once
} else {
    Convert to Change Request
}
```

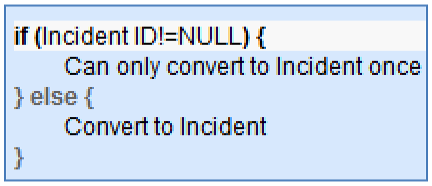
Convert to Incident

Purpose: Validates that Problem has not been converted to Incident before, then saves and converts to an Incident.

On Click: Execute action then save record.

Action:

I: Convert to Incident If First Conversion



```
if (Incident ID!=NULL) {
    Can only convert to Incident once
} else {
    Convert to Incident
}
```

Escalate

Purpose: Sets Assigned Team to 2nd Level Support Team and Priority to Critical.

On Click: Execute action.

Action:

U: Set Assigned Team to 2nd Level Support Team and Priority to Critical

Updated Fields: Assigned Team, Priority

Updated Values: 2nd Level Support Team, Critical

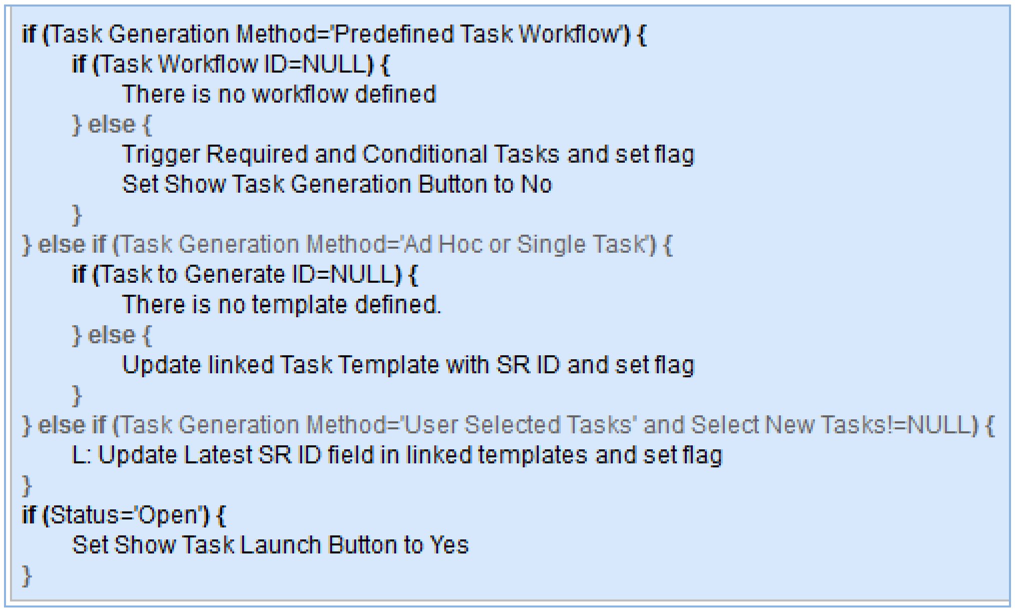
Generate Tasks

Purpose: generates the tasks based on the service details.

On Click: Executes action.

Action:

I: Create Task Actions



```
if (Task Generation Method='Predefined Task Workflow') {
    if (Task Workflow ID=NULL) {
        There is no workflow defined
    } else {
        Trigger Required and Conditional Tasks and set flag
        Set Show Task Generation Button to No
    }
} else if (Task Generation Method='Ad Hoc or Single Task') {
    if (Task to Generate ID=NULL) {
        There is no template defined.
    } else {
        Update linked Task Template with SR ID and set flag
    }
} else if (Task Generation Method='User Selected Tasks' and Select New Tasks!=NULL) {
    L: Update Latest SR ID field in linked templates and set flag
}
if (Status='Open') {
    Set Show Task Launch Button to Yes
}
```

Support Cases Table

Rules and Actions

Assignee Change by someone else

Purpose: Informs assignee when assignee is changed

Runs: Edit by Email, Web, or API

Search Filter: Assignee change

Priority: 1

Actions:

E: Email assignee

Template: Case #$id assigned to you: "$summary"

SC – All new case actions

Purpose: Inform assignee or assigned team of new case.

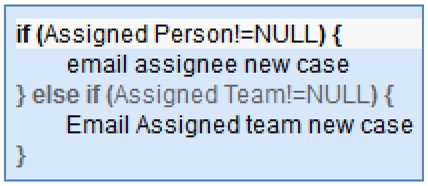
Runs: Create by Email, Web, or API

Search Filter: None

Priority: 2

Actions:

I: Email Assigned team or person of new case



```
if (Assigned Person!=NULL) {
      email assignee new case
} else if (Assigned Team!=NULL) {
      Email Assigned team new case
}
```

SC – All Customer Update Actions

Purpose: Handles status changes and email notices from interactions with customer.

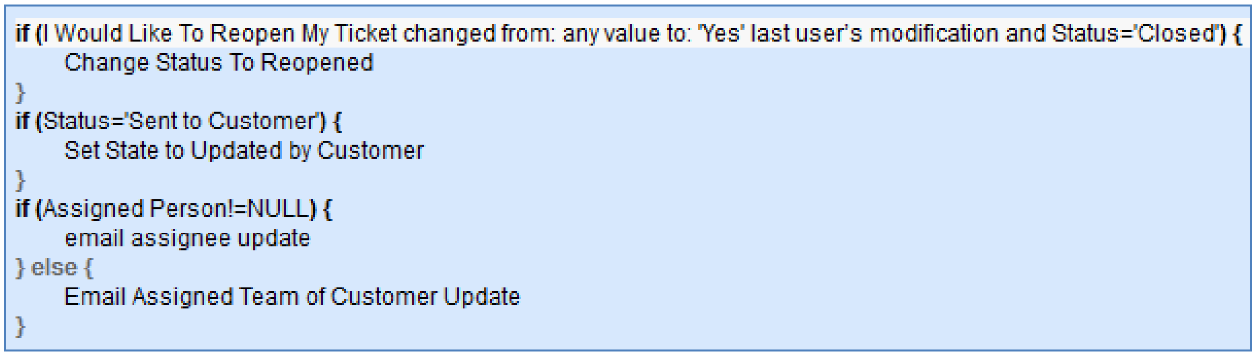
Runs: Edit by Email or Web

Search Filter: R: Updater Team is customer team

Priority: 3

Actions:

I: Customer Update Ticket Actions



```
if (I Would Like To Reopen My Ticket changed from: any value to: 'Yes' last user's modification and Status='Closed') {
      Change Status To Reopened
}
if (Status='Sent to Customer') {
      Set State to Updated by Customer
}
if (Assigned Person!=NULL) {
      email assignee update
} else {
      Email Assigned Team of Customer Update
}
```

Action Buttons

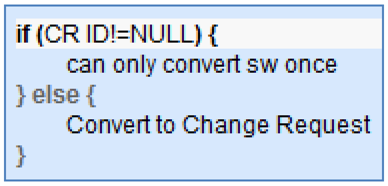
Convert to Change Request

Purpose: Check to see if Change Request already exists for this Support Case. If not, converts to create a Change Request from Support Case.

On-Click: Do nothing, execute actions, then save and open record for view.

Action:

I: Convert to change request if not yet done



```
if (CR ID!=NULL) {
      can only convert sw once
} else {
      Convert to Change Request
}
```

Tasks Table

Rules and Actions

Create: All create validations

Purpose: Restricts addition of tasks to legal statuses and warns the user if the due date is earlier than the time it is created (the user may pass through this warning). Note that this will not run on tasks created by API.

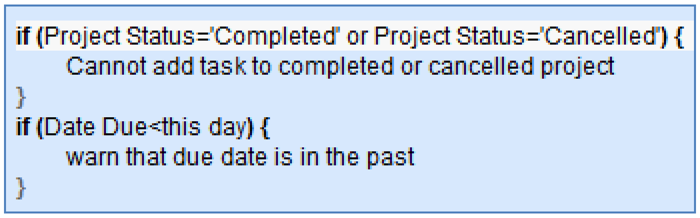
Runs: Create by Web.

Search Filter: None

Priority: 1

Actions:

I: All create validations



```
if (Project Status='Completed' or Project Status='Cancelled') {
        Cannot add task to completed or cancelled project
}
if (Date Due<this day) {
        warn that due date is in the past
}
```

Create: All Creation Actions

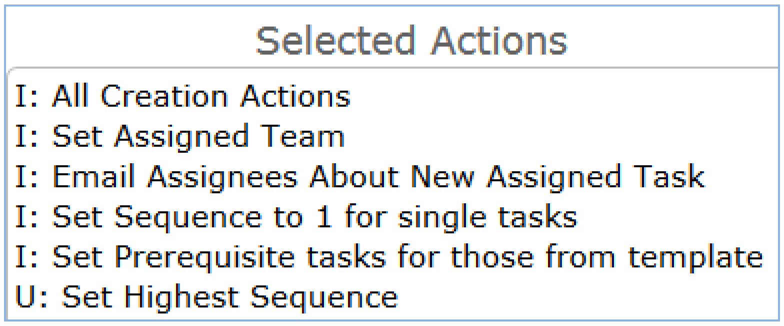
Purpose: Sets due dates, assigns project manager and teams, and emails assignees of assignation.

Runs: Create by Email, Web, API:

Search Filter: None

Priority: 1

Actions:

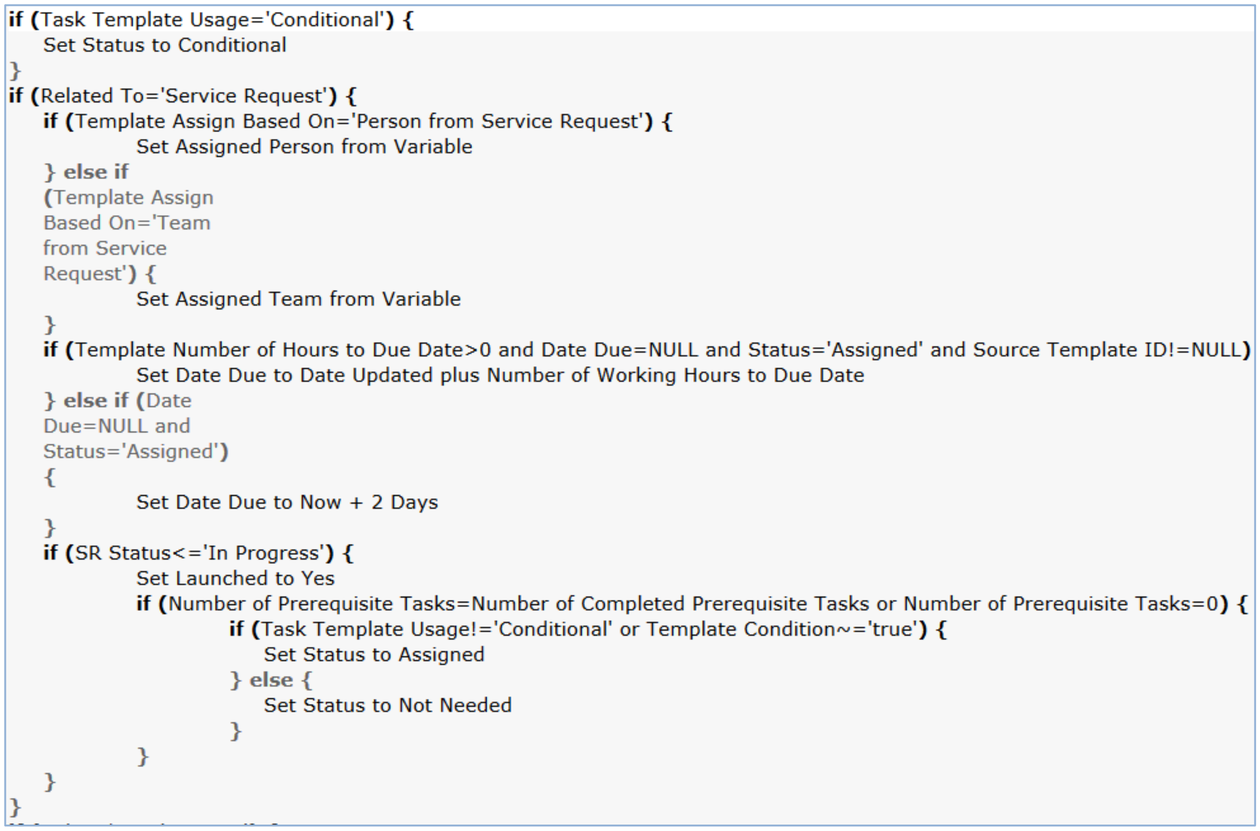


Selected Actions

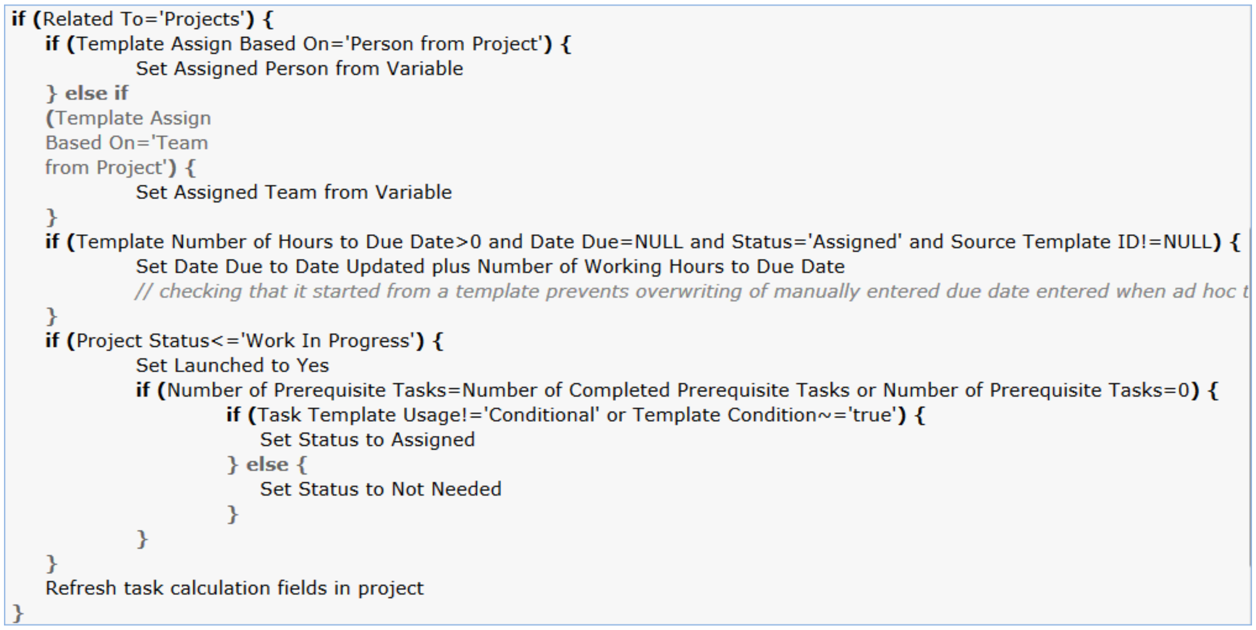
I: All Creation Actions
I: Set Assigned Team
I: Email Assignees About New Assigned Task
I: Set Sequence to 1 for single tasks
I: Set Prerequisite tasks for those from template
U: Set Highest Sequence

I: All Creation Actions

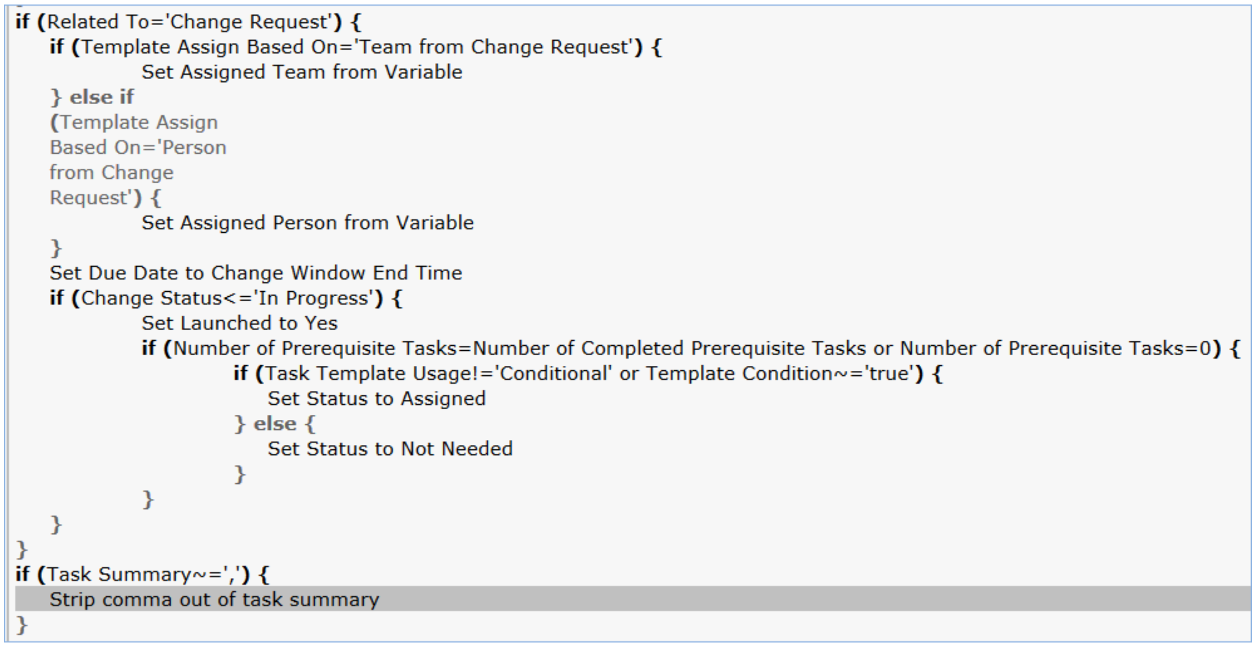
This action has several parts based on the type of request the task is associated with. In general, it does not permit a comma in the title and strips it out because the comma breaks the prerequisite task handling. It also sets the Status to Conditional for any task whose Task template usage is Conditional. The other actions handle setting the assignment and due dates based on the context:



```
if (Task Template Usage='Conditional') {
    Set Status to Conditional
}
if (Related To='Service Request') {
    if (Template Assign Based On='Person from Service Request') {
              Set Assigned Person from Variable
    } else if
    (Template Assign
    Based On='Team
    from Service
    Request') {
              Set Assigned Team from Variable
    }
    if (Template Number of Hours to Due Date>0 and Date Due=NULL and Status='Assigned' and Source Template ID!=NULL)
              Set Date Due to Date Updated plus Number of Working Hours to Due Date
    } else if (Date
    Due=NULL and
    Status='Assigned')
    {
              Set Date Due to Now + 2 Days
    }
    if (SR Status<='In Progress') {
              Set Launched to Yes
              if (Number of Prerequisite Tasks=Number of Completed Prerequisite Tasks or Number of Prerequisite Tasks=0) {
                        if (Task Template Usage!='Conditional' or Template Condition~='true') {
                           Set Status to Assigned
                        } else {
                           Set Status to Not Needed
                        }
              }
    }
}
```



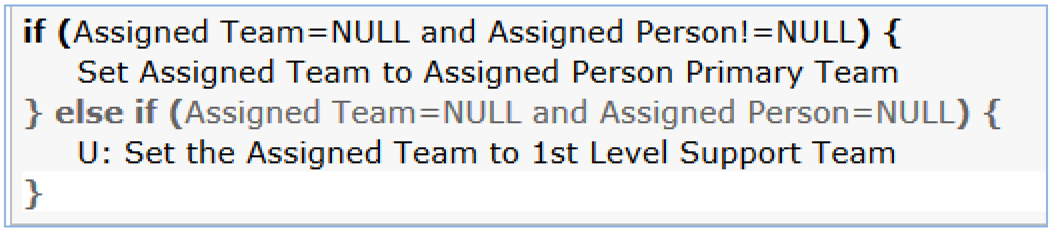
```
if (Related To='Projects') {
    if (Template Assign Based On='Person from Project') {
              Set Assigned Person from Variable
    } else if
    (Template Assign
    Based On='Team
    from Project') {
              Set Assigned Team from Variable
    }
    if (Template Number of Hours to Due Date>0 and Date Due=NULL and Status='Assigned' and Source Template ID!=NULL) {
              Set Date Due to Date Updated plus Number of Working Hours to Due Date
              // checking that it started from a template prevents overwriting of manually entered due date entered when ad hoc t
    }
    if (Project Status<='Work In Progress') {
              Set Launched to Yes
              if (Number of Prerequisite Tasks=Number of Completed Prerequisite Tasks or Number of Prerequisite Tasks=0) {
                        if (Task Template Usage!='Conditional' or Template Condition~='true') {
                           Set Status to Assigned
                        } else {
                           Set Status to Not Needed
                        }
              }
    }
    Refresh task calculation fields in project
}
```



```
if (Related To='Change Request') {
    if (Template Assign Based On='Team from Change Request') {
            Set Assigned Team from Variable
    } else if
    (Template Assign
    Based On='Person
    from Change
    Request') {
            Set Assigned Person from Variable
    }
    Set Due Date to Change Window End Time
    if (Change Status<='In Progress') {
            Set Launched to Yes
            if (Number of Prerequisite Tasks=Number of Completed Prerequisite Tasks or Number of Prerequisite Tasks=0) {
                    if (Task Template Usage!='Conditional' or Template Condition~='true') {
                        Set Status to Assigned
                    } else {
                        Set Status to Not Needed
                    }
            }
    }
}
if (Task Summary~=',') {
    Strip comma out of task summary
}
```

I: Set Assigned Team

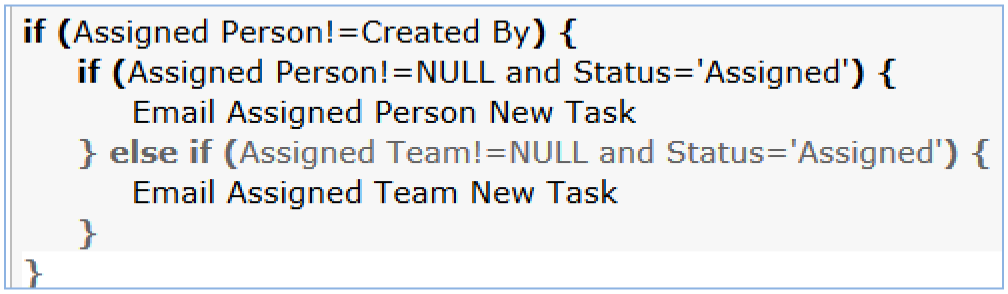
Purpose: This sets the assigned team to the primary team of the assigned person if the task has been assigned to a specific person from the request. If for some reason the system has tried to assign to a specific person and that person is not defined so there is no assigned team or person, then the system sets the Assigned Team to the 1st Level Support Team:



```
if (Assigned Team=NULL and Assigned Person!=NULL) {
    Set Assigned Team to Assigned Person Primary Team
} else if (Assigned Team=NULL and Assigned Person=NULL) {
    U: Set the Assigned Team to 1st Level Support Team
}
```

I: Email Assignees About New Task

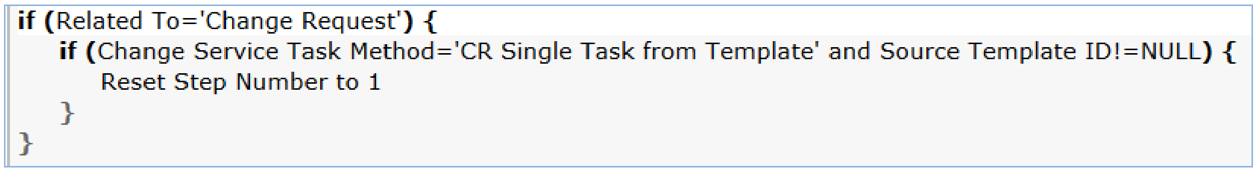
The assignee is only emailed if the task is already launched and set to a Status of Assigned. Otherwise the notification is sent when the status does change:



```
if (Assigned Person!=Created By) {
    if (Assigned Person!=NULL and Status='Assigned') {
        Email Assigned Person New Task
    } else if (Assigned Team!=NULL and Status='Assigned') {
        Email Assigned Team New Task
    }
}
```

I: Set Sequence to 1 for single tasks

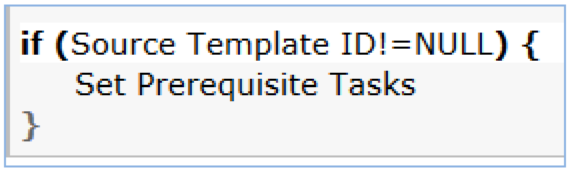
Purpose: For change request tasks that may be drawn from a task template that is part of a task workflow and that might therefore have a higher sequence value, this rule sets the Sequence value to 1:



```
if (Related To='Change Request') {
    if (Change Service Task Method='CR Single Task from Template' and Source Template ID!=NULL) {
        Reset Step Number to 1
    }
}
```

I: Set Prerequisite tasks for those from template

Purpose: If this is a task that is based on a task template that might have prerequisite task templates, it runs an action to set the Prerequisite tasks based on those whose source template IDs match the Prerequisite IDs in this task's source template.



```
if (Source Template ID!=NULL) {
    Set Prerequisite Tasks
}
```

U: Set Highest Sequence

Purpose: This action sets the linked field to the Prerequisite Task to this task that has the highest sequence value. This is used to set this task's sequence in the next rule to 1 higher than that value.

Create/Edit: Update Sequence based on Highest Sequence

Purpose: This sets the Sequence value for a new task to the next highest number after all of its prerequisite tasks to indicate where in the general sequence it will be due.

Runs: Create/Edit (including edits made by other rules during creation) by Web, API:

Search Filter: R: Highest Sequence is not empty and changed last modification

Priority: 2

Actions:

U: Set Sequence to Highest Prereq plus 1

Edit: All Edit Actions (API enabled)

Purpose: Handles updating a task when it is launched to a status of Assigned – sets the due date and notifies the assignees; When a task is completed, handles the launching of any tasks for which it was the prerequisite, sets the alert color, and notifies the appropriate people, and also handles pushing any working notes into the running notes fields in the related record.

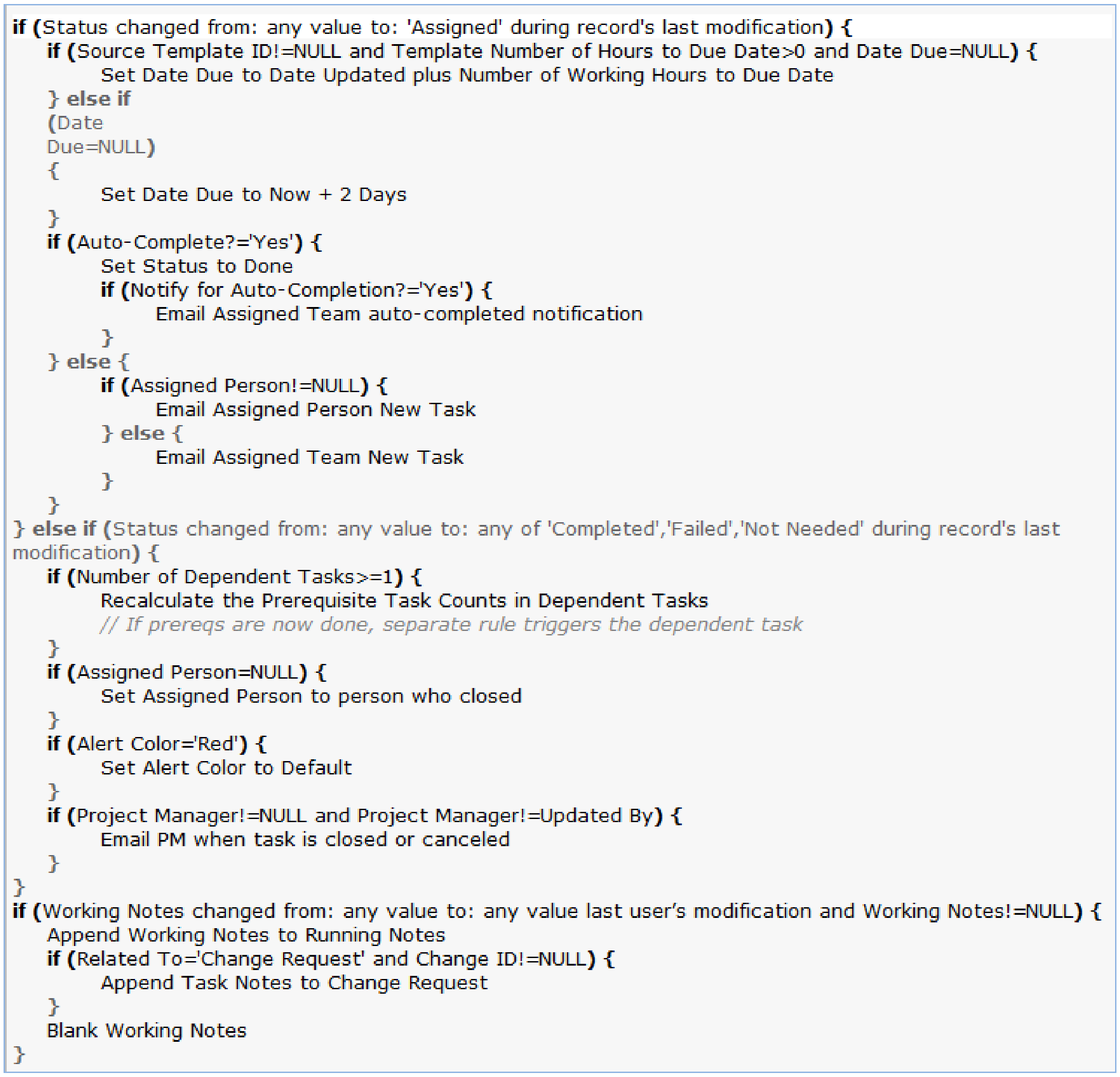
Runs: Edit by Web or API

Search Filter: None

Priority: 4

Actions:

I: All Edit Actions (API enabled)



```
if (Status changed from: any value to: 'Assigned' during record's last modification) {
    if (Source Template ID!=NULL and Template Number of Hours to Due Date>0 and Date Due=NULL) {
        Set Date Due to Date Updated plus Number of Working Hours to Due Date
    } else if
    (Date
    Due=NULL)
    {
        Set Date Due to Now + 2 Days
    }
    if (Auto-Complete?='Yes') {
        Set Status to Done
        if (Notify for Auto-Completion?='Yes') {
            Email Assigned Team auto-completed notification
        }
    } else {
        if (Assigned Person!=NULL) {
            Email Assigned Person New Task
        } else {
            Email Assigned Team New Task
        }
    }
} else if (Status changed from: any value to: any of 'Completed','Failed','Not Needed' during record's last
modification) {
    if (Number of Dependent Tasks>=1) {
        Recalculate the Prerequisite Task Counts in Dependent Tasks
        // If prereqs are now done, separate rule triggers the dependent task
    }
    if (Assigned Person=NULL) {
        Set Assigned Person to person who closed
    }
    if (Alert Color='Red') {
        Set Alert Color to Default
    }
    if (Project Manager!=NULL and Project Manager!=Updated By) {
        Email PM when task is closed or canceled
    }
}
if (Working Notes changed from: any value to: any value last user’s modification and Working Notes!=NULL) {
    Append Working Notes to Running Notes
    if (Related To='Change Request' and Change ID!=NULL) {
        Append Task Notes to Change Request
    }
    Blank Working Notes
}
```

Edit: Assign tasks when number of completed prerequisites meets criteria (API)

Purpose: This updates dependent tasks when their prerequisites are completed, checking if they are conditional, whether the condition is currently met.

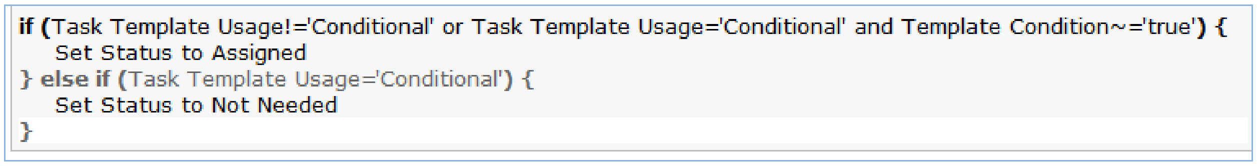
Runs: Edit by Web, API:

Search Filter: R: Prerequisite tasks just completed and status is queued or conditional

Priority: 1

Actions:

I: Assign or Mark as Not Needed based on condition



```
if (Task Template Usage!='Conditional' or Task Template Usage='Conditional' and Template Condition~='true') {
    Set Status to Assigned
} else if (Task Template Usage='Conditional') {
    Set Status to Not Needed
}
```

Edit: If task title has a comma, strip it out.

Purpose: Strips any commas that have been added to the task title. This is necessary because the task title needs to be compared against the prerequisite task titles and if they contain a comma, it breaks the comparison.

Runs: on Edit (including edits made by other rules during record creation) by Web or API

Search Filter: R: Task Summary contains comma

Priority: 40

Actions:

U: Strip comma out of task summary

TB: (Disabled) Notify of upcoming task

Purpose: Notifies appropriate parties of pending tasks.

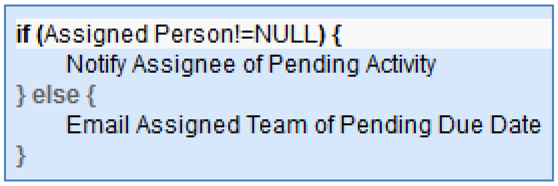
Runs: Daily

Search Filter: R: Upcoming Tasks

Priority: 3

Actions:

I: Notify assigned person or team of due date



```
if (Assigned Person!=NULL) {
     Notify Assignee of Pending Activity
} else {
     Email Assigned Team of Pending Due Date
}
```

TB: (Disabled) Set alert color field

Purpose: Updates helper Alert Color field to red when criteria met.

Runs: Daily

Search Filter: R: Status not done, cancelled, or failed and due date is passed and not red

Priority: 30

Actions:

U: Set Alert Color to Red

Fields Changed: Alert Color

New Value: Red

Action Buttons

Button: Add Time

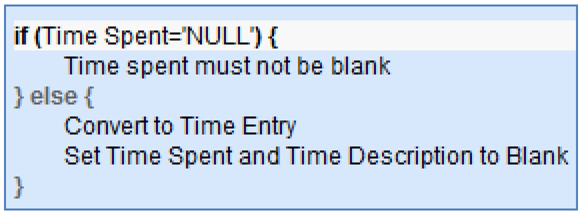
Purpose: Validates that Time Spent has a value, then converts Time Spent and Time Description to a Time Entry and blanks out those fields.

On Click: Save record, then execute action.

Visibility: If Enable Prerequisite Tasks is Yes.

Action:

I: Add Time



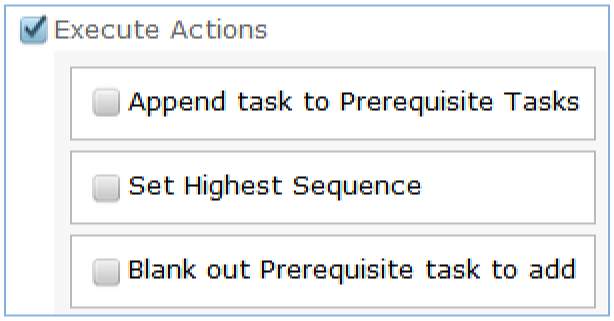
```
if (Time Spent='NULL') {
    Time spent must not be blank
} else {
    Convert to Time Entry
    Set Time Spent and Time Description to Blank
}
```

Button: Add to Prerequisites

Purpose: Adds the selected task from the same request as a Prerequisite task, then sets the highest sequence in the current record, then blanks out the Prerequisite task to add field.

On Click: Do nothing, then execute action, then do nothing. The new prerequisite task shows up below in the table of prerequisite tasks.

Action:


☑ Execute Actions

☐ Append task to Prerequisite Tasks

☐ Set Highest Sequence

☐ Blank out Prerequisite task to add

Button: End Clock

Purpose: Sets the Work actual end time to Now.

On Click: Execute action, then save and reopen for editing

Visibility:

Action:

U: Set Work Actual End Time to Now

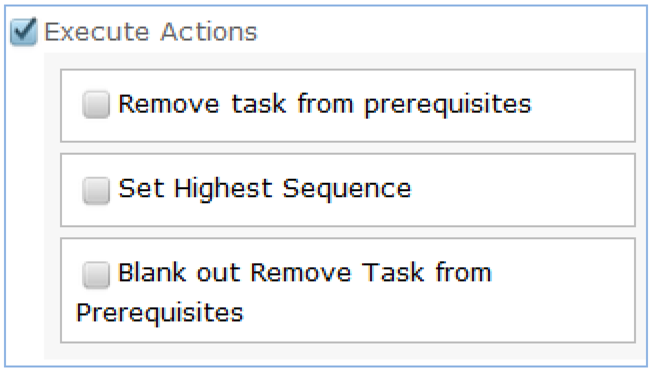
Button: Remove from Prerequisites

Purpose: Removes the selected task from the prerequisites and recalculates the highest sequence value, then blanks out the selected task.

On Click: Execute action, then does nothing

Visibility: If Enable Prerequisite Tasks is Yes.

Action:


☑ Execute Actions

☐ Remove task from prerequisites

☐ Set Highest Sequence

☐ Blank out Remove Task from Prerequisites

Button: Set CI to In Planned Downtime

Purpose: Sets the Operational Status of the linked CI to Planned Downtime

On Click: Saves, then executes action, then saves and reopens for edit.

Visibility: Task Type=CI Change and Status=Waiting for Others, Assigned

Action:

U: Set CI to In Planned Downtime

Button: Set CI to Working

Purpose: Sets the Operational Status of the linked CI to Working

On Click: Execute action, then save and reopen for editing.

Visibility: Task Type=CI Change and Status=Waiting for Others, Assigned

Action:

U: Set CI to Working

Button: Start Clock

Purpose: Sets the Work Actual Start Time to Now

On Click: Execute action, then save and reopen for edit.

Visibility:

Action:

U: Set Work Actual Start Time to Now

ButtonAB: Reassign Task

Purpose: This is a button to put on the action bar associated with the Inactive Tasks related tables to make it easy to move a Task back into the active Tasks related table by setting its Status to Assigned.

On Click: Execute action then save

Visibility:

Action:

U: Set Status to Assigned

ButtonAB: Remove

Purpose: Used to remove a task from use in a particular record (service request, project, or change request). The task is set to Not Needed and moved into the Inactive Tasks related table. It is still kept linked to the request in case it needs to be resuscitated.

On Click: Execute action then save

Visibility:

Action:

U: Set Status to Not Needed

Set Date Done

Purpose:

On Click: Execute action.

Visibility:

Action:

Purpose: Sets the Date Done field to current date/time.

On Click: Execute action.

Visibility: Status is Done.

Action:

U: Set Date Done to Now

Updated Field: Date Done

Updated Value: NOW()

Task Templates Table

Rules and Actions

Create: All Creation Actions

Purpose: If a new task template has been created from a service, the task template is linked back to the service. This is primarily for change request tasks. If there are prerequisite tasks, this sets the Highest Prerequisite Step Number to the one with the highest sequence value.

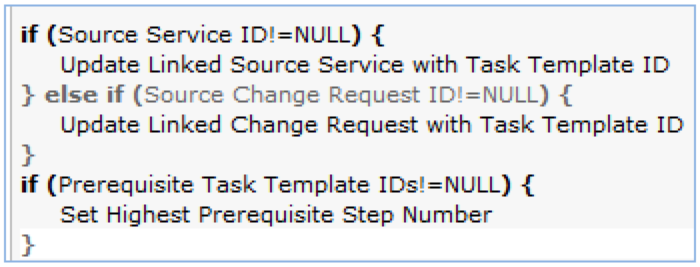
Runs: On creation by web or API

Search Filter: None

Priority: 1

Actions:

All Creation Actions



```
if (Source Service ID!=NULL) {
    Update Linked Source Service with Task Template ID
} else if (Source Change Request ID!=NULL) {
    Update Linked Change Request with Task Template ID
}
if (Prerequisite Task Template IDs!=NULL) {
    Set Highest Prerequisite Step Number
}
```

Create/Edit: - check for comma in task title and warn

Purpose: Removes any comma in the task title so it doesn't break the automation, then gives the user a browser popup telling them the comma has been removed in case they want to edit the wording.

Runs: on Creation or Edit by Web.

Search Filter: R: Task Title contains a comma

Priority: 15

Actions:

U: Remove comma from task title

B: Commas are not allowed

Create/Edit: Set Sequence Based on Prerequisites

Purpose: Sets the sequence of the task based on any prerequisites, to the highest prerequisite sequence plus 1.

Runs: on Create or Edit, including edits by other rules, by Web or API

Search Filter: R: Highest prerequisite step is not empty (i.e. there is a prerequisite)

Priority: 2

Actions:

U: Set Step Number to Highest plus 1

Edit: Generate Tasks when triggered from Project

Purpose: Generates task from the template and copies the Project ID into a field to track that this template has been converted for this project (to avoid regenerating it).

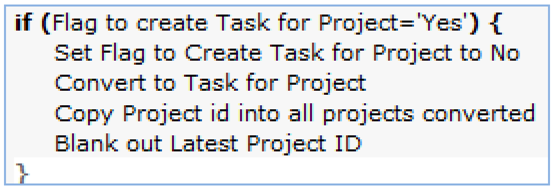
Runs: Edit by Web or API, including edits by other rules during creation

Search Filter: Project ID has changed and not yet converted

Priority: 1

Actions:

I: Create Task for Project



```
if (Flag to create Task for Project='Yes') {
    Set Flag to Create Task for Project to No
    Convert to Task for Project
    Copy Project id into all projects converted
    Blank out Latest Project ID
}
```

U: Copy Project ID text into all projects converted

Field Changed: All Project IDs Converted

Value Changed to: $last_project_id_text +”,”

Edit: Generate Tasks when triggered from Service Requests

Purpose: Generate tasks and perform upkeep on All SRs converted field.

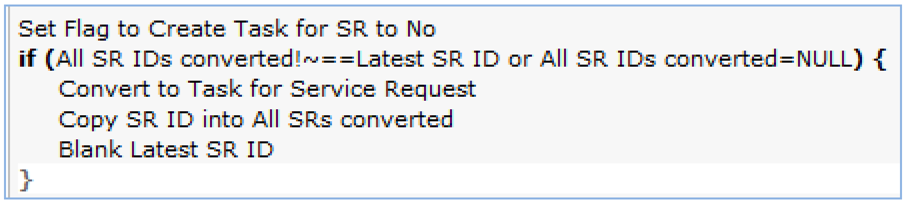
Runs: Edit by Web or API, including edits by other rules during creation

Search Filter: Flag to create task from SR is Yes

Priority: 2

Actions:

I: Create task for Service Request



```
Set Flag to Create Task for SR to No
if (All SR IDs converted!~==Latest SR ID or All SR IDs converted=NULL) {
    Convert to Task for Service Request
    Copy SR ID into All SRs converted
    Blank Latest SR ID
}
```

C: Convert to Task

Conversion: Task

Options: Silently with no confirmation.

U: Copy SR ID text into All SRs converted

Field Changed: All SR IDs converted

Value Changed to: $last_sr_text +”,”

Edit: Generate Tasks when triggered from Change Request

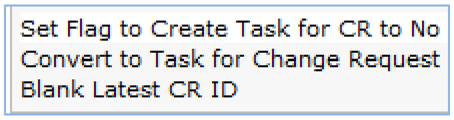
Purpose: Generates the task for a change request.

Runs: Edit by Web or API

Priority: 5

Search Filter: LF: Flag to Create Task for CR from Template is Yes

I: Create Task for Change Request



```
Set Flag to Create Task for CR to No
Convert to Task for Change Request
Blank Latest CR ID
```

Edit: Other edit actions - unlink inactive task template, handle cloned templates

Purpose: Removes the task template from a workflow or linked service if it is made inactive, so it will no longer be generated for that workflow or that service. If a task workflow is cloned, and the task template is marked as needing to be cloned, then this creates the cloned task template.

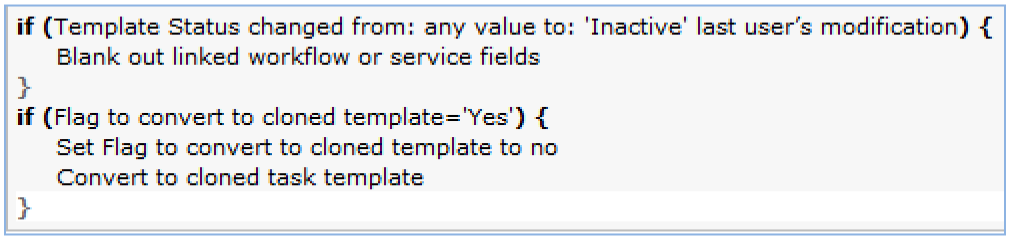
Runs: On Edit by Web or API, including edits made by rules during creation

Search Filter: None

Priority: 4

Actions:

Handle other edit actions



```
if (Template Status changed from: any value to: 'Inactive' last user's modification) {
    Blank out linked workflow or service fields
}
if (Flag to convert to cloned template='Yes') {
    Set Flag to convert to cloned template to no
    Convert to cloned task template
}
```

Task Workflows

Rules and Actions

Create: Trigger task template conversion for cloned workflow

Purpose: When a new workflow is created as a clone, this triggers the conversion of the original workflows task templates to new ones linked to the new workflow.

Runs: On Creation by Web or API

Search Filter: R: Source Workflow ID is not empty and has task templates

Priority: 1

Actions:

L: Trigger Conversion in task templates for cloned workflow

Create: All Create Actions

Purpose: After the workflow has been saved, this action sets the title from the original workflow with the time and date (the workflow title must be unique).

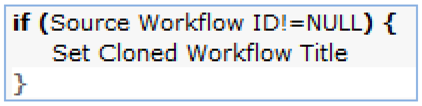
Runs: On Creation by Web or API

Search Filter: None

Priority: 2

Actions:

I: All Create Actions



```
if (Source Workflow ID!=NULL) {
    Set Cloned Workflow Title
}
```

Edit: refresh task template prerequisites as number of tasks changes for cloned

Purpose: This is the rule that ensures that the new task templates created for a cloned workflow are linked to each other properly in a prerequisite-dependent task relationship

Runs: On Creation/Edit by Web or API, including edits made by other rules on creation

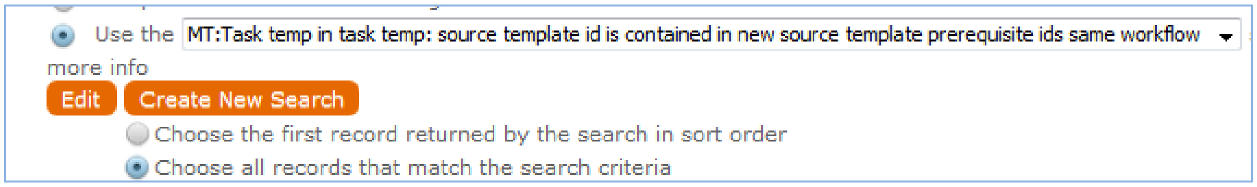
Search Filter: Number of tasks changed and cloned workflow (as each task template is added for a cloned workflow)

Priority: 2

Actions:

U: Refresh task template prerequisites

This sets the prerequisites based on a saved search:



[✓ Use the] MT:Task temp in task temp: source template id is contained in new source template prerequisite ids same workflow

more info

Choose the first record returned by the search in sort order
[✓ Choose all records that match the search criteria]