

# Agiloft Summit

2019

## University Workbook

October 22

Mandalay Bay Resort | Las Vegas



# Contents

Power User: Power Up .....	6
Overview .....	7
Home .....	7
Dashboards .....	7
User Preferences .....	8
General Preferences .....	8
Personalized Left Panes .....	8
Keyboard Shortcuts .....	9
Searching .....	10
Quick Searches .....	11
Saved Searches .....	12
Table Views .....	13
Icons and Colors .....	14
Image with Versioning .....	14
Heat Bars .....	15
Quick Edit .....	15
Action Buttons .....	16
Summary Layout .....	16
Record views .....	17
General Recommendations .....	17
Common Area .....	18
Tabs and Text Headings .....	19
Conditional Visibility .....	20
Tab Visibility .....	22
Build to Suit: Custom EUI .....	23
Elements of a Webpage .....	24
Common EUI Macros .....	24
Automate Task Management .....	25
Overview .....	26

Tasks vs. Approvals .....	26
Structure that Supports Task Management .....	27
Guide to Linking Tasks to Other Tables .....	28
Working with the Tasks Table .....	28
Task Layout .....	29
Use Case.....	31
Working with the Task Templates Table .....	34
Task Template Layout .....	34
Use Case.....	36
Working with the Task Workflows Table .....	37
Use Case.....	37
Prerequisite Task Handling .....	39
Cloning a Task Workflow.....	40
Working with the Task Steps Table .....	40
Graphical Elements .....	42
Row Coloring.....	43
Notification Icons .....	43
Image with Versioning .....	44
Creating an Image with Versioning Field.....	44
Image Dimensions for Personnel Photos .....	45
Heat Bar .....	45
Create a Heat Bar .....	46
Show the Heat Bar .....	47
Singleton Check.....	47
Creating a Singleton Check .....	47
Relationship Diagram.....	47
How to Configure Relationship Diagram Fields .....	47
Defining Field Relationships.....	48
Resources.....	49
Status Bars .....	49
System Icons .....	49
Snagit Image Capture and Editing Software.....	50
Look and Feel .....	50
Take Control of Roles and Permissions .....	52
Groups .....	53

Starting from Default Groups.....	55
Group Permissions Wizard.....	55
Table Permissions Wizard .....	58
Teams .....	62
Teams Wizard .....	65
User Roles .....	71
Roles .....	71
Drill Down on Dashboards, Charts, and Reports .....	75
Dashboards .....	76
Navigation.....	76
Permissions.....	76
Create a Dashboard .....	76
Add and Remove Widgets.....	77
Widget Types .....	77
Dashboard Filters .....	78
Dashboard Layout .....	78
Dashboard Access .....	79
Reports .....	79
Report permissions .....	79
Creating a Chart/Report.....	80
Combined excel reports .....	82
Useful reports to have .....	83
Think Outside the Docx.....	84
Print Template Syntax Reference.....	85
Field Variables.....	85
View or Download Field Variables .....	85
Formulas and Conditions .....	86
Basic Formulas .....	86
Concatenate Strings.....	87
Dateformat .....	87
Format Formula Outputs .....	88
Conditional Text, Paragraphs, or Clauses.....	89
Ternary Operator .....	89
If-Then-Else Conditions to Display Field Variables .....	90
Conditional Paragraphs using \$startif .....	91



isEmpty .....	92
New line or insert line break .....	92
Merge Documents .....	93
Print Templates with eSignature Tags.....	93
DocuSign .....	93
Adobe Sign .....	95
Work Wonders with Rules .....	96
Rules .....	97
Creating, editing, and deleting rules .....	98
Rule Configuration .....	98
Actions .....	101
If-Then-Else Actions .....	102
Email Actions.....	104
Update Fields Actions .....	105
Validate Actions .....	107
Data Conversion Actions .....	107
Import and Export Actions .....	109
Delete Actions.....	110
Create a Delete Action .....	110
Browser Pop-up Actions.....	111
Script Actions .....	112
Report Actions .....	113
OCR Action .....	114
Convert Document Format Action .....	115
Merge Documents Action .....	116
Print Action .....	118
Linked Record Actions.....	120
Engage with Email and SMS .....	128
Introduction .....	129
Email Templates.....	129
SMS Templates .....	135
Configuring SMS Integration with Twilio.....	136
Non-Standard Implementations .....	137
Approve via Email from outside the Firewall .....	138
Receiving SMS .....	139

Parsing the Subject Line .....	139
Update Records via Email without Replying to System Generated Email .....	142
Handling Emails differently across the Organization.....	144
Care for Your Database .....	149
Table Field Limits .....	150
Working with History Size Limits.....	151
History Best Practices.....	151
Indexing .....	152
Creating Indexes .....	152
Considerations for Creating Indexes .....	153
Checking Indexing Progress.....	153
Synchronizing Indexes.....	154
Turning Off Indexing .....	154
How to Let Users Delete Records without Delete Permission .....	154
Mark for Deletion Button.....	155
Activity Logs Setup .....	156
Configure Activity Logs.....	157
Secure Your Organization .....	159
User Controls .....	160
Access Controls .....	160
Data Controls .....	160
Global Variables .....	160



# Power User: Power Up

---

# Overview

This document contains resources for the Agiloft Summit University "Power Up" presentation. It will include links to the Wiki when available, and sections from the Wiki as possible.

## Home

The Home Menu is found in the left pane when you log in to Agiloft. Clicking on the Home menu itself launches the last visited dashboard.

Wiki Resource: <https://wiki.agiloft.com/display/HELP/Home+Menu>

## Dashboards

Dashboards are the "home base" for Agiloft Power Users, and are published to Teams. Power users can automatically be directed to the default Dashboard defined for their Primary Team.

Wiki Resource: <https://wiki.agiloft.com/display/HELP/Dashboards>

## Dashboard Widgets

The dashboard is created from a collection of widgets, and the following kinds of widgets are supported:

- [Chart Widgets - https://wiki.agiloft.com/display/HELP/Chart+Widgets](https://wiki.agiloft.com/display/HELP/Chart+Widgets)
- [Hotlink Widgets - https://wiki.agiloft.com/display/HELP/Hotlink+Widgets](https://wiki.agiloft.com/display/HELP/Hotlink+Widgets)
- [Table Widgets - https://wiki.agiloft.com/display/HELP/Table+Widgets](https://wiki.agiloft.com/display/HELP/Table+Widgets)
- [Embedded Webpage Widgets - https://wiki.agiloft.com/display/HELP/Embedded+Webpage+Widgets](https://wiki.agiloft.com/display/HELP/Embedded+Webpage+Widgets)
- [Text Widgets - https://wiki.agiloft.com/display/HELP/Text+Widgets](https://wiki.agiloft.com/display/HELP/Text+Widgets)
- [Image Widgets - https://wiki.agiloft.com/display/HELP/Image+Widgets](https://wiki.agiloft.com/display/HELP/Image+Widgets)
- [Custom Summary Report Widgets - https://wiki.agiloft.com/display/HELP/Custom+Summary+Report+Widgets](https://wiki.agiloft.com/display/HELP/Custom+Summary+Report+Widgets)
- [Numerical Result Widgets - https://wiki.agiloft.com/display/HELP/Numerical+Result+Widgets](https://wiki.agiloft.com/display/HELP/Numerical+Result+Widgets)

Once a widget has been created it is stored in a widgets library and becomes available to any system dashboards, based on the user's team permissions.

## Dashboard Filters

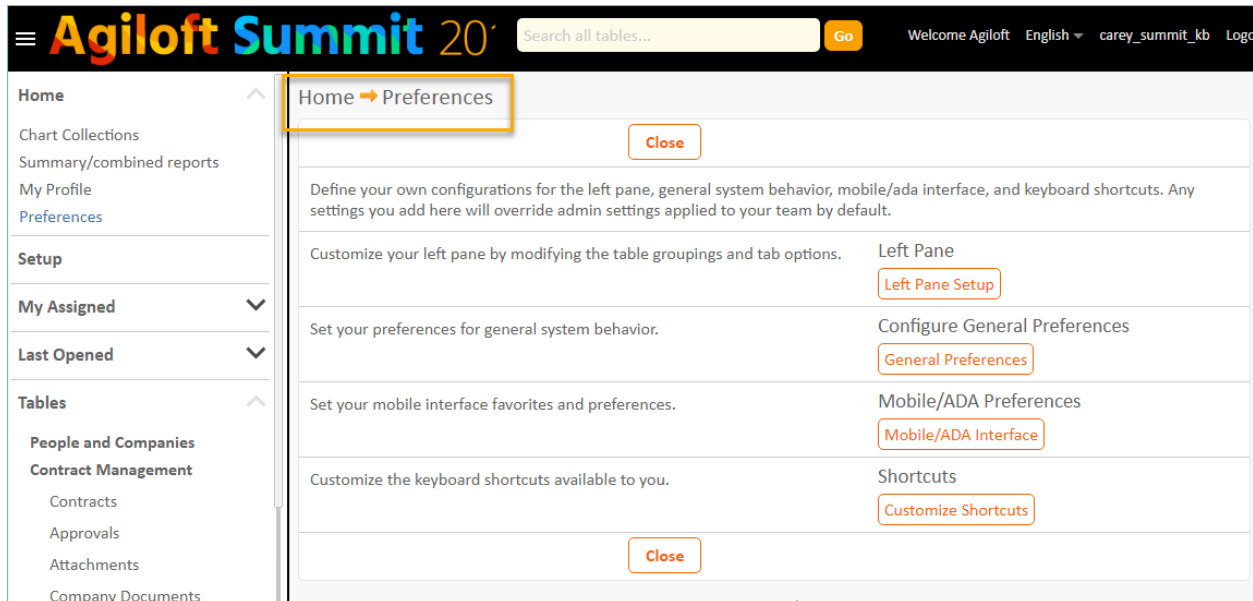
Wiki Resource: <https://wiki.agiloft.com/display/HELP/Dashboard+Filters>

Filters can be added as a widget on the [Dashboard \(https://wiki.agiloft.com/display/HELP/Dashboards\)](https://wiki.agiloft.com/display/HELP/Dashboards), which allow you to refine the fields in each widget by some defined search criteria. Each filter type works by defining a set of fields to filter the widget, along with additional search criteria. The filter setup allows you to choose all relevant fields in the current widget selection for the filter type. For example, a Time-based filter has access to all time fields in the available widgets, such as End Date, Start Date, and Hire Date.

Filters are created on the Filters tab of the Dashboard wizard. When a filter is applied to the dashboard, it removes all records that do not have the value in one of the fields specified in the filter, for all widgets that use the filter fields.

## User Preferences

The Preferences menu allows Power Users to customize behavior of certain system-wide features for your user account (if permitted by Group permissions).



Admins can turn on/off the ability for Groups to configure their own preferences via group permissions at Setup > Access > Groups.

## General Preferences

Admins can configure default General Preferences for Primary Teams via Setup Look and Feel > General Preferences.

Wiki Reference: <https://wiki.agiloft.com/display/HELP/General+Preferences>

Click Home > Preferences > General Preferences to adjust your system's behavior (if allowed by your group permissions).

General Preferences changes allow you to open records and searches in new browser tabs rather than new windows if possible, to display all records in a table or your last search, and more.

## Personalized Left Panes

The left pane is the primary navigation tool in the power user interface. You may see different menus depending on how your system is customized or due to your group permissions.

Admins can configure default Left Panes for Teams via Setup > Look and Feel > Left Pane Setup.

Wiki Resources: <https://wiki.agiloft.com/display/HELP/Navigating+the+Left+Pane>  
<https://wiki.agiloft.com/display/HELP/Left+Pane+Setup>

Menus can be collapsed and expanded by clicking on the menu names or using the minimize and maximize icons. To resize the width of the pane, drag it to the right or the left with the mouse; to hide it entirely, click the minimize icon on the Home menu.

The items in the left pane can be customized according to your own requirements. Navigate to Home > Preferences > Left Pane Setup, where you can hide and reorder items and define the order of tables.

At any time, you can return to the Home Page even when the left pane is closed, by clicking the company logo at the upper left of the screen.

## My Assigned

The My Assigned area of the Left Pane shows specific searches configured to display in this area. Each search can optionally display the number of records matching the search criteria (though this option will cause slower login times).

Wiki Reference: <https://wiki.agiloft.com/display/HELP/My+Assigned+Inbox>

An entry appears in the My Assigned section for each table when:

1. The user has permission to view records.
2. The My Assigned search is accessible, active and set to be visible in the My Assigned area for at least one of the user's groups. These options are set when the Saved Search is created or edited on the "Apply" tab.

## Last Opened

This section of the Left Pane can be configured in Left Pane Setup. It contains an option for the maximum number of records to display in the Last Opened list, as well as whether Last Opened records are opened in Edit mode.

## Keyboard Shortcuts

Keyboard shortcuts (also sometimes called hotkeys) allow power users to navigate through the Agiloft interface and perform labor-intensive tasks much more quickly.

The standard shortcuts are initially defined in a Default shortcut scheme available to all users. These default keystroke values may be changed by an administrator, who may also create additional schemes applied to specific teams by going to Setup > Look and Feel > Customize Shortcuts.

Individual power users may also edit the standard shortcut definitions to suit their own preferences at Home > Preferences > Customize Shortcuts.

Wiki References: <https://wiki.agiloft.com/display/HELP/Keyboard+Shortcuts>

To see all Shortcuts in Agiloft, you can go to Help > Shortcut Help.

Press Esc to close this page

General	Table Views
Alt + H Home screen	Alt + ] Next Page
Alt + S Global Search	Alt + [ Previous Page
Alt + / Shortcuts Help	Alt + N New Record
Alt + D Help Site	Alt + O Edit Record
Alt + M My Assigned	Alt + V View Record
Alt + L Last Opened	Alt + - Delete Record
Alt + T Select table	Alt + U Mass Edit
Alt + X Log out	Alt + 8 Cancel All In-Table edits
	Ctrl + Enter Finish All In-Table edits
	Up Cursor Up
	Down Cursor Down
	Left Cursor Left
	Right Cursor Right
	Enter Start Direct Edit
	Alt + Right Next Field to the Right
	Alt + Left Next Field to the Left
	Alt + E Send Email
	Alt + P Print Records
	Alt + W Edit Table View
Record form	
Alt + R Highlight Embedded Table	
Alt + [ Previous tab	
Alt + ] Next tab move	
Alt + K Field Selection	
Ctrl + Enter Save and Close	
Alt + C Save and Continue	
Alt + 8 Cancel	
Alt + I Select LF Record	

## Searching

Any table in your system can be searched. For instance, you may want to find a particular contract, or all contracts of a certain type, or you may be looking for particular text content within an attached document.

The quick **search block** lets you complete ad-hoc searches, and is used when you don't anticipate running the same search again.

**Saved searches** let you run the same search repeatedly and is typically reserved for searches you will need routinely. Saved searches have the added benefit of accepting more complex search criteria and offer more granular search results.

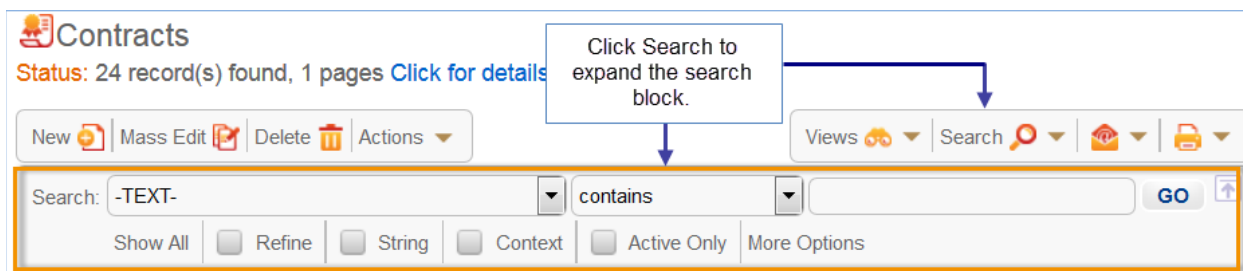
Wiki References: <https://wiki.agiloft.com/display/HELP/Searching>

<https://wiki.agiloft.com/display/HELP/Main+Search+Block>

<https://wiki.agiloft.com/display/HELP/Creating+Saved+Searches>

## Quick Searches

Quick searches are done from the **search block** above the table you wish to search. If the search block (screenshot below) is not visible, click the word **Search** on the action bar to expand the search block:



Administrators can configure Quick Search fields via Setup table > Indexes > Edit Quick Search Fields.

## Using the Search Block

The first input box is a drop-down to indicate which field in the table to search. If the default **-TEXT-** is selected, all fields will be searched, including the content within any attached files.

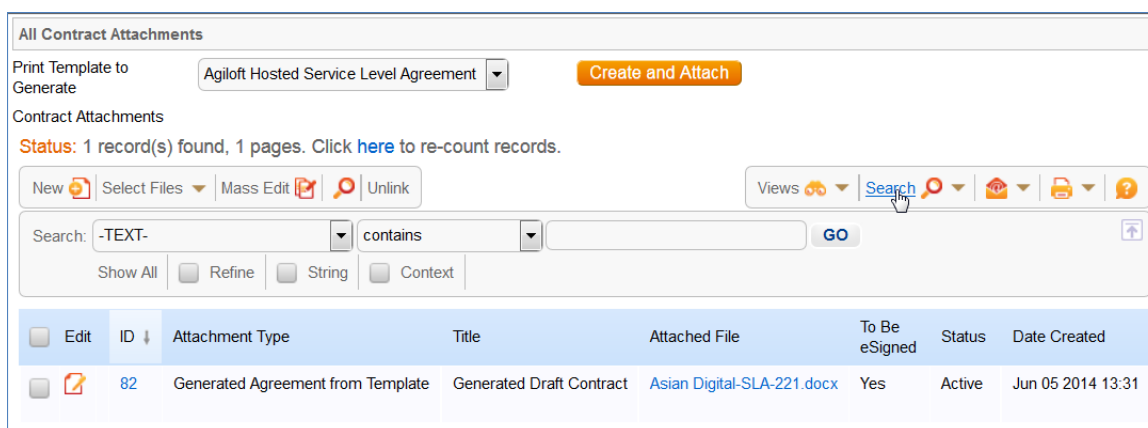
**NOTE:** If the default **-TEXT-** is selected, and you type in a number, Agiloft will search the ID field.

The second input box defines the **search operator**. Different operators are available depending on the type of field selected in the first box. For example, numeric or choice fields offer “greater than” and “less than” operators, but text fields do not.

In the third input box, enter the value you are searching for.

The most common ad-hoc searches will look something like “Contract Party Type equals Vendor” or “Contract Title contains Amazon.”

Quick searches can be performed on **related tables** using the search function on the associated action bar. For instance, the **Attachments** tab of each contract record contains the **Contract Attachments** related table of all documents attached to the contract. Note that the same search functionality is available here; again, clicking **Search** from the top-right of the action bar will expand the search block if needed. Please note that searching from a **related table** within a record is searching on the source table; in the screenshot below, the search will run on the attachments table to find content in documents related to that contract record.

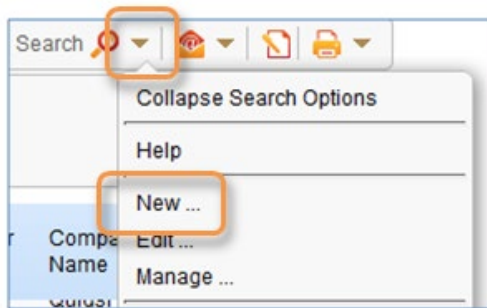


Edit	ID	Attachment Type	Title	Attached File	To Be eSigned	Status	Date Created
	82	Generated Agreement from Template	Generated Draft Contract	Asian Digital-SLA-221.docx	Yes	Active	Jun 05 2014 13:31



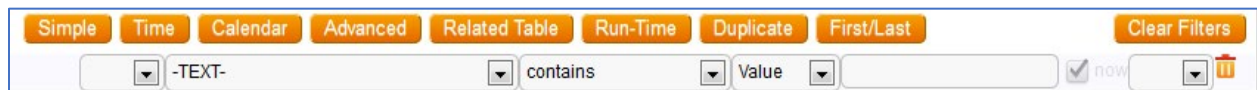
## Saved Searches

To create a **saved search**, hover over the down arrow next to the search icon and select **New**.



This opens the saved search wizard, which has options for **Simple** filters (most commonly used), as well as more advanced options.

Click **Simple** to add a filter:



Just like the quick search above, simple filters let you search any field for a particular value. Again, selecting -TEXT- will search through all available fields within the current table.

For more information on the other search types, please see our online tutorial [here](http://www.agiloft.com/emvideo.html?name=search) (<http://www.agiloft.com/emvideo.html?name=search>).

## Managing Saved Searches

Users in the Admin group, and those with permission to publish saved searches, see an **Apply** tab in the saved search wizard where they can decide which groups should have the search:

- Accessible
- Active
- Visible in left menu
- Visible in My Assigned

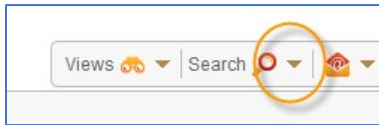
The **Apply** tab options have the following results:

1. Choosing to make a search **Accessible** gives user the ability to interact with the search in their own Manage Searches wizard, shown below.

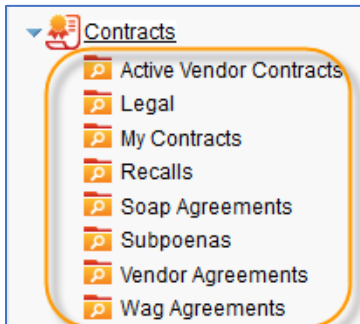
savedsearch					
Status: 41 record(s) found, 5 pages. Click <a href="#">here</a> to re-count records.					
Page: [Page 1] <a href="#">Next</a> <a href="#">Last</a> Go to page <input type="text"/> <a href="#">Go</a>					
<input type="checkbox"/> Edit Link	Name ↑	Filter	Active	Show in Left Pane	Show in My Assigned
<input type="checkbox"/>	Active Contract expiring within next 12 months (C)	Status='Active' and Contract future	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Active Contract expiring within next 90 days (C)	Status='Active' and Contract	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Active Customer Contracts	Contract Type Party Type='Ci	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Each staff, or **power user**, can adjust where their accessible searches appear by checking the **Active**, **Show in Left Pane**, or **Show in My Assigned** boxes, defined next. This lets users customize where searches appear to tailor the interface to their work preferences.

2. **Active** makes the search appear on the Search drop-down:

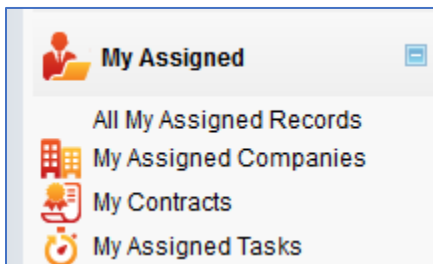


3. **Show in Left Pane** adds the search to the left pane, indented below the Table name:



*TIP: Searches are ordered here A-Z. To control the order of the searches, you can try prefacing their label with \* or numbers like 1, 2, 3 etc.*

4. **Show in My Assigned** adds the search to the **My Assigned** section in the left pane:



## Table Views

A View determines the way records are displayed within a table. Views allow different individuals and teams to have their own perspective on the same data in a table.

Wiki References: <https://wiki.agiloft.com/display/HELP/Views>

<https://wiki.agiloft.com/display/HELP/View+Wizard>

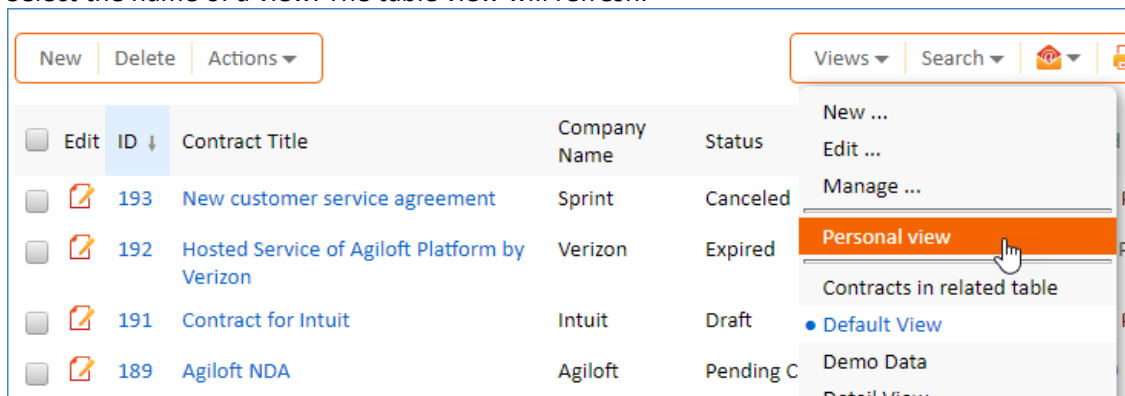
Views define the following types of display characteristics:

- Which fields are included and visible.
- Row ordering and colors.
- View and field widths, and number of records per page.
- Whether cell borders are included.

Views do not affect the data to be included in the tables. This is controlled by a [search](https://wiki.agiloft.com/display/HELP/Searching) (<https://wiki.agiloft.com/display/HELP/Searching>). Optionally, saved searches can define a preset view to use.

To switch between views...

1. Hover over **Views** in the action bar to see a list of available views.
2. Select the name of a view. The table view will refresh.



## Icons and Colors

You can visually represent the values of certain fields in a record by creating notification icons that correspond to the value(s) in a choice field. Whenever a record contains the specified value in the field, the icon will be displayed in the table view for that record. To set up notification icons, navigate to **Views > Edit** to open the View wizard, click the Order/Colors tab, and then click Define Notification Icons to open the Define Notification Icons window.

On the Fields tab, select one or more choice fields from the list. Then, on the Icons tab, select the icons to associate with the values in each field you selected and define the alt text. You can associate an icon with one value, several of the values, or all the values. The system comes with many icons to choose from, but you may also upload your own icons.

Wiki References: <https://wiki.agiloft.com/display/HELP/Notification+Icons>

In order to provide a visual aid to easily identify specific record attributes, records can be color-coded based upon the value of a selected choice field. In a table, navigate to **Views > Edit** to open the View wizard, then click the Order/Colors tab and click the Set Row Coloring button to open the Set Row Coloring window. On the Field tab, select the choice field that will serve as the trigger for the row coloring. Then, on the Colors tab, change the text size, font, color, and background color for each value of the choice field.

Wiki References: <https://wiki.agiloft.com/display/HELP/Row+Coloring>

## Image with Versioning

Image with Versioning fields store attached files that are displayed as pictures within the record. For example, an employee record might have a small photo of that employee. This data type supports versioning that stores past versions of image files, but you can disable the option if you don't want to use it or if you need to save space.

Only files with appropriate image extensions can be uploaded to this field type. You can also configure these fields to display the images in the table view, in their original format or cropped into circles, on the Options tab of the Field wizard.

Wiki Resources: <https://wiki.agiloft.com/display/HELP/Image+with+Versioning>

## Heat Bars

Heat bar fields show data as colored segments of one bar, useful for showing proportions, progress, and more. Heat Bars are built using either a Related Table or using multiple Numeric fields.

For example, you might include a Heat Bar field in Contracts to show data from the Approvals Needed related table. In the Default View of Contracts you could include the Heat Bar field to see the number of Approvals that are Pending, Queued and Require Change.

Wiki Resources: <https://wiki.agiloft.com/display/HELP/Heat+Bars>

## Quick Edit

The Quick Edit feature allows users to edit fields directly a table view. The admin can determine which users can enable or disable Quick Edit within particular tables by setting up the required permissions.

Before using Quick Edit, you must define two group permissions

(<https://wiki.agiloft.com/display/HELP/Views+and+Permissions#ViewsandPermissions-RequiredFieldSetup>), enable Quick Edit for each field, and then [create the view](https://wiki.agiloft.com/display/HELP/View+Wizard) (<https://wiki.agiloft.com/display/HELP/View+Wizard>).

Wiki Resources:

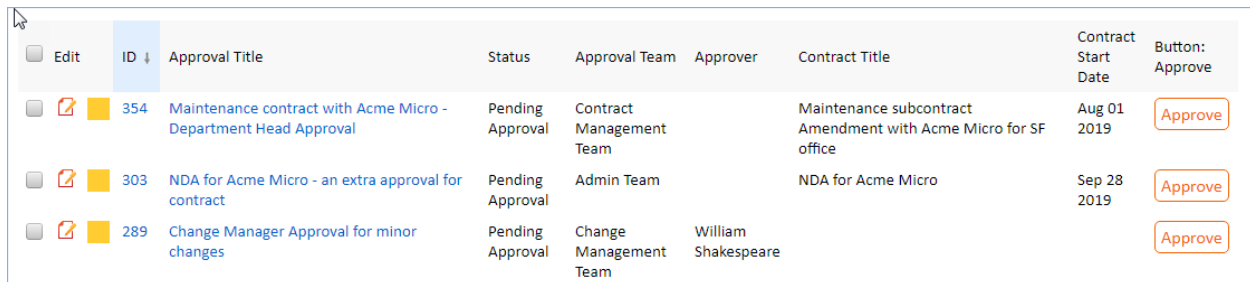
- <https://wiki.agiloft.com/display/HELP/Quick+Edit>
- <https://wiki.agiloft.com/display/HELP/Views+and+Permissions>
- <https://wiki.agiloft.com/display/HELP/Table+Permissions+Wizard>
- <https://wiki.agiloft.com/display/HELP/View+Wizard>








In a quick edit view, a pencil icon appears in the column heading above each editable field. Two buttons, Save Changes and Cancel Changes, appear in the table's action bar. You can also use Keyboard Shortcuts to save all changes and cancel all changes.

Edit	ID ↓	Company Name	Primary Role	Account Rep	Status
<input type="checkbox"/>	54	Thrashikeet Industries	Vendor	Agiloft Admin	Active
<input type="checkbox"/>	53	Chickens with Names	Vendor	Agiloft Admin	Active
<input type="checkbox"/>	52	Glenaire Garlic Farm	Vendor	Agiloft Admin	Active
<input type="checkbox"/>	51	Lost Coast Abalone Jewelry	Vendor	Agiloft Admin	Active
<input type="checkbox"/>	50	New Company Limited	Vendor	Agiloft System	Active

## Action Buttons

Action buttons have admin-level permissions by default, and will operate independently of a user's permissions. If a user can view the button, they can activate it with a few exceptions. Action Buttons may be included in Views.



 Edit	ID ↓	Approval Title	Status	Approval Team	Approver	Contract Title	Contract Start Date	Button: Approve
	354	Maintenance contract with Acme Micro - Department Head Approval	Pending Approval	Contract Management Team		Maintenance subcontract Amendment with Acme Micro for SF office	Aug 01 2019	
	303	NDA for Acme Micro - an extra approval for contract	Pending Approval	Admin Team		NDA for Acme Micro	Sep 28 2019	
	289	Change Manager Approval for minor changes	Pending Approval	Change Management Team	William Shakespeare			

View permissions for action buttons apply to both the record form and the table view, but they have a caveat when placed on the table view. **If users have permission to view an action button and it's placed on the table view, it will appear for every record, even if the user doesn't have permission to edit that record. If a user clicks an action button from the table view for a record they don't have permission to edit, the action won't run and the user will receive an error message.**

Admins can change this behavior by setting the [Always Show Action Button in Views](https://wiki.agiloft.com/display/HELP/Global+Variables+List#GlobalVariablesList-AlwaysShowActionButtoninViews) (<https://wiki.agiloft.com/display/HELP/Global+Variables+List#GlobalVariablesList-AlwaysShowActionButtoninViews>) global variable to No. This forces the system to evaluate users' permissions and only display action buttons on the table view for records that they also have permission to edit. However, this may impact system performance if you have many records or complex permissions.

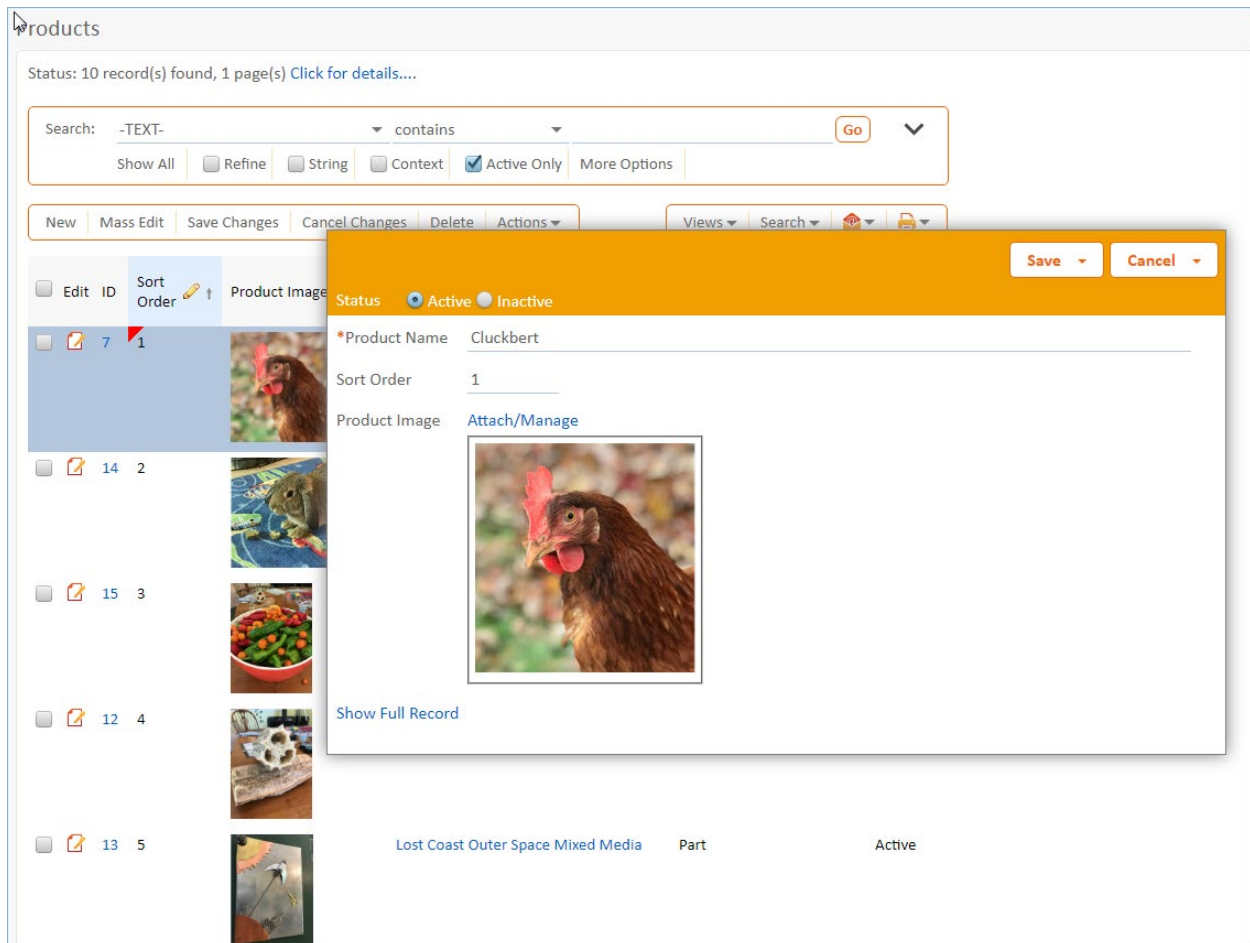
Wiki References: <https://wiki.agiloft.com/display/HELP/Action+Buttons>

## Summary Layout

Make it easier to check common record details and make simple edits from the table view with Summary layout popovers. Simply configure a Summary layout for the table and add View or Edit Popover to a field in the table view to make the Summary layout accessible to users.

Wiki Reference: <https://wiki.agiloft.com/display/HELP/Layouts>

Popovers are available in both the power user interface and the EUI, and they can also be used in dashboards and widgets. All popovers include a link at the bottom to the full record.



## Record views

The record view displays when users click a record on a table view, and they're taken to the record in View or Edit mode.

Wiki Resources: <https://wiki.agiloft.com/display/help/Layouts>

## General Recommendations

1. Use the Common Area for a limited number of critical fields. May display at the top or the left, depending on size of the fields.
2. Use Tabs and collapsible Text Headings to break up fields into logical chunks.
3. Keep required fields on the first tab.
4. Prioritize horizontal placement over vertical placement to allow users to use the Tab key to move from one field to the next.
5. Use conditional visibility / Extra Fields to Show to display only relevant fields.

## Common Area

The Common Area is a section that always shows in the record form, even when the user changes tabs. This way, you can make important fields easily accessible from every tab, both for reference and editing purposes. For example, Common Areas often include the record ID and name, a summary field, and a status field. Make sure the Common Area includes fields users frequently need, without making the area too large and overtaking the rest of the tab, or requiring users to scroll down a long sidebar.

The Common Area can be formatted as a horizontal section across the top of the form, a sidebar across the right or left side of the form, or not used at all.

To decide how to format the Common Area for a table, consider the fields you want to place there and the way users will interact with the form. For example, if your Common Area will function as a passive reference for the record ID, title, and creator, you might place it at the top so users can scroll up on any tab to see the information. In contrast, if your Common Area houses several action buttons that users might need to click at any time, you might place it as a sidebar so it is always visible and users can click the action buttons from anywhere.

## Sidebar Display Options

Showing the Common Area as a sidebar can require more trial and error when configuring your layout, especially if you originally designed the Common Area in the default position and want to move it.

Because the sidebar is narrower, field names and content might be too long or look too cluttered. Some common solutions include:

- Using the Common Area option to show field labels above the input box rather than beside it, especially if there's a lot of open space in the Common Area
- Moving fields with long names or long values out of the Common Area and onto another tab
- Changing the field label display in Look & Feel to distinguish it from the input text, such as using bold for field labels, as long as this change is appropriate to apply everywhere in your system

It's also more noticeable if a sidebar Common Area is disproportionate to the rest of the form.

- Sidebars should have a maximum width set at or near 30%
- Don't include so many fields that the sidebar requires scrolling

Finally, consider if and how you want to include action buttons.

- The sidebar Common Area is always visible, unlike the header Common Area. Any action buttons you include will be accessible from any tab at any point in the user's workflow. In some cases, this is beneficial, and makes it easy for users to escalate issues, create new tasks, email reminders, and more. In other cases, users are required to complete certain fields or tabs before clicking action buttons, and including them on the sidebar might set users off course.
- If you're pressed for space or you want all your Common Area fields to align, you might consider configuring action buttons in your Common Area to display as hyperlinks. These take up less visual space and align with other regular text field labels.

## Planning Layout Sections

It can be useful to plan or sketch your form before you begin building, especially for tables with many fields. Think about how users will use the form and what they need to accomplish, in what order. For example, if users need to complete all the fields on the first tab before using any action buttons, you might place the action buttons at the bottom of the first tab. In general, create a layout where users complete forms from top to bottom and move sequentially from the first tab to the last tab.

Here are some questions to guide your planning:

What is the workflow for users? What do they fill out first and last, and when do they use action buttons?

Should fields be concentrated in one or two tabs with more scrolling, or distributed among many tabs that must be navigated?

What information needs to be in the Common Area? Is that enough, or too much?

Which fields are required? Usually, these fields are placed on the first tab, so users fill them all out before clicking Save.

You don't need to place every field ahead of time, but you'll save yourself time if you have an idea of the sections and tabs you want to create, and how you plan to use the Common Area. At a minimum, consider whether to show the Common Area at the top of the form or move it to the right or left sidebar, and consider which fields you want to add to the Common Area to be available on every tab of the form.

## Tabs and Text Headings

Wiki Resources:

- <https://wiki.agiloft.com/display/help/Layouts#Layouts-CreatingaLayout>
- <https://wiki.agiloft.com/display/help/Visibility+Dependent+Fields>

Considerations:

- To maximize loading speed, avoid placing related tables on the first tab so that they can load in the background.
- Try to place all or at least most required fields on the first tab.
- Text headings help organize the form, so use them liberally. You may also want to have a text heading collapsed by default if the fields it contains aren't typically displayed to users. Note that if the form is refreshed, the default settings will be reapplied.
- If a user does not have permission to view any of the fields in a record tab, the tab will automatically disappear from view. If the visibility conditions are satisfied for even one field in a tab, the tab will be visible in the record layout. If the visibility conditions for any field are changed so that the user can view it, the tab will automatically become visible for them.
- When no fields within a Text heading are visible to a user, the text heading will also not be visible to the user.



# Conditional Visibility

Conditional field visibility simplifies the user experience by hiding/showing fields on the record form only when they are needed. Visibility dependence can be based on one or more conditions that combine to determine whether the field is visible. Each condition is based on a native or linked *choice* or *multi-choice* field having certain values. (Contract Type's "Extra Fields to Show" is an example of a multi-choice field controlling the visibility of fields within Contract.)

You can choose whether to show the field if **all conditions are met** or if **any of the conditions is true**.

Wiki Resources: <https://wiki.agiloft.com/display/HELP/Visibility+Dependent+Fields>

## Terminology


- **Dependent field** – A field whose visibility is controlled by the values held in one or more other fields – the parent fields – in the table.
- **Parent field** – A field whose values control the visibility of a dependent field. Parent fields are choice fields, multi-choice fields, or linked multi-/choice fields from other tables.
- **Condition** – Conditions are the set of allowed values in the parent field that cause the dependent field to appear. The condition editor is a pop-up window where you select fields and values.


## Setting Visibility Dependence Conditions

To begin, edit the dependent field by navigating to **Setup Table > Fields tab > Edit field**.

1. In the Field wizard, navigate to the Options tab. Scroll down to find the option Make the visibility of this field conditional.
2. Click Add Condition to open the condition editor.
  - a. From the drop-down, select the field that controls this field's visibility, the parent field.
  - b. Select the values that will allow this field to appear on layouts. Hold down Ctrl to select multiple values for a condition.
  - c. Click Finish to return to the field wizard.



3. To add another condition, repeat step 2. There is no specific limit on the number of conditions, other than the number of available parent fields.
4. To modify an existing condition, click the edit icon .

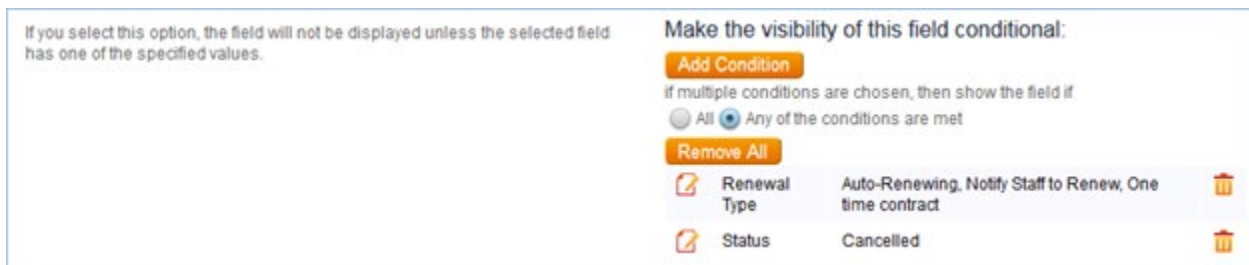
5. To delete an existing condition, click the delete icon . To remove all existing conditions for visibility dependence, click Remove All.
6. If multiple conditions are set, choose whether the field should be shown if All conditions are met, or if Any of the conditions are met.
7. Click Finish to apply your changes and return to the Table wizard.

## Use Case

Let's consider an example in contract management. Suppose the Contract table contains a date field, Contract End Date, which should be visible if the Renewal Type field has the value Auto-Renewing, Notify Staff to Renew, or One time contract. Contract End Date should also be visible if the Status is Cancelled, regardless of the Renewal Type.

To meet this requirement, add the two conditions on the Options tab of the Field wizard – one based on the Status field, a second based on the Renewal Type field – and choose the intersect option, 'Any of the conditions are met'. Note that in earlier releases of Agiloft, there was no way to do this without creating a third choice field, for example, Show End Date?, that was set by a rule based on the values of the first two fields, Renewal Type and Status.

Now, we can make dependent fields visible if **all** or **any** of several conditions is met:



If you select this option, the field will not be displayed unless the selected field has one of the specified values.



**Make the visibility of this field conditional:**

Add Condition

If multiple conditions are chosen, then show the field if

☐ All ☒ Any of the conditions are met

Remove All

Renewal Type	Auto-Renewing, Notify Staff to Renew, One time contract	
Status	Cancelled	

As with other field settings, System Admins or Table Admins can set visibility dependencies, provided their group has admin access to the tables containing the affected fields.

## Required Field Permissions

In order to see a visibility dependent field, users must have view permission for the dependent field itself. In addition:

- When a single visibility condition is set: Users must have view permission for the field that controls visibility, the parent field.
- When “all” of multiple conditions must be met: Users must have view permission for all of the parent fields.
- When “any” of multiple conditions must be met: Users must have view permission for **at least one** of the parent fields which is “true,” i.e., which is currently meeting the condition and allowing the dependent field to appear.

It is not necessary for a parent field to appear on the layout in order for a dependent field to recognize its value.

If any of the parent fields is itself hierarchically dependent, users must have permission to view the parents of those parent fields, i.e. the “grand-parent” fields.

## Multiple Visibility Condition Permission Examples

The graphic examples below show how different combinations of **view permissions** and **true conditions** affect users’ ability to see a dependent field.

### All conditions must be met

All conditions must be met	User has view permission?	Is condition met?
Parent A	✓	✓
Parent B	✓	✓
Parent C	✓	✓
Is dependent field visible?	YES	

All conditions must be met	User has view permission?	Is condition met?
Parent A	✓	✓
Parent B	✓	
Parent C	✓	✓
Is dependent field visible?	NO	

All conditions must be met	User has view permission?	Is condition met?
Parent A	✓	✓
Parent B		✓
Parent C	✓	✓
Is dependent field visible?	NO	

### Any condition must be met

Any condition must be met	User has view permission?	Is condition met?
Parent A	✓	
Parent B		✓
Parent C	✓	
Is dependent field visible?	NO	

Any condition must be met	User has view permission?	Is condition met?
Parent A	✓	✓
Parent B		✓
Parent C	✓	
Is dependent field visible?	YES	

Any condition must be met	User has view permission?	Is condition met?
Parent A	✓	
Parent B	✓	✓
Parent C	✓	
Is dependent field visible?	YES	

## Tab Visibility

If a user does not have permission to view any of the fields in a record tab, the tab will automatically disappear from view. If the visibility conditions are satisfied for even one field in a tab, the tab will be visible in the record layout. If the visibility conditions for any field are changed so that the user can view it, the tab will automatically become visible for them.



# Build to Suit: Custom EUI

---

# Elements of a Webpage

All webpages consist of 4 basic elements:

1. **HTML** – HTML defines the structure of a webpage. HTML is the content of the webpage, and some basic info about how it should look. HTML can define lists, links, tables, images, and more.
2. **CSS** – CSS defines how a webpage looks. CSS defines how all HTML elements and classes should look. Without CSS, each HTML tag has to be styled separately.
3. **Javascript** – Javascript is an event-based programming language. It often handles dynamic HTML content; making lots of rows on an HTML spreadsheet based off of some data for instance. It also handles complex animation and changing the look of a page on the fly (like 'Dark Mode'). Javascript is always handled by your browser.
4. **Serverside content** – Agiloft uses Velocity to get Agiloft tables and data. This is handled by the server before the page is sent to the browser.

## Common EUI Macros

EUI Macro	Usage
<b>#ew_create_record</b> (\$table, \$url, \$frame)	Create a record in a table. Can pass in extra default parameters or values for fields.
<b>#ew_forward</b> (\$eui_template)	Link to an EUI page from the EUI Templates table.
<b>#ew_image</b> (\$image_name)	Display an image from an Agiloft kb.
<b>#ew_table</b> (\$table, \$view, \$search, \$query, \$params, \$frame, \$style, \$showLabel)	Show a table with a given view and saved search. \$params allows for lots of different options such as showing the quick search bar.
<b>#ew_edit_record</b> (\$table, \$recordId, \$url, \$frame)	Edit a record from a table.
<b>#if()</b> <b>#end</b>	Allows for conditional formatting.
<b>\$request.getParameter()</b>	Allows for passing info between webpages. Great for changing how a page looks based on where you came from.
<b>#ewquery</b> (\$table, \$frame, \$controlID, \$params)	Uses the text in the element with the ID \$controlID to apply a filter to the table in \$frame.
<b>#ew_searches_list</b> (\$table, \$frame, \$control, \$style, \$params, \$selectedSearch)	Provides an HTML list of saved searches from a table. Great for quickly viewing relevant records in a table.



# Automate Task Management

---

## Overview

The Tasks table has records that can be linked to and appear within the records in other tables. The Tasks table holds individual tasks. The standard system is currently set up to link tasks to Service Requests, Change Requests, and Projects, as well as to Assets, 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. The ITIL system, for instance, also uses tasks in Release Management.

Fields related to task generation and a related Tasks table 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 Assets. Projects allow the same methods of task generation as service requests. Assets allow single tasks to be created and linked to the asset, and show all tasks that have been created from other records, such as change requests or service requests, that relate to that asset.

## Tasks vs. Approvals

Tasks are similar to approvals in their basic setup. They have workflow and templates tables where they can be predefined, and the workflows/templates are used to automatically generate a set of tasks to be done in a specified order within some other record, such as a contract or service request.

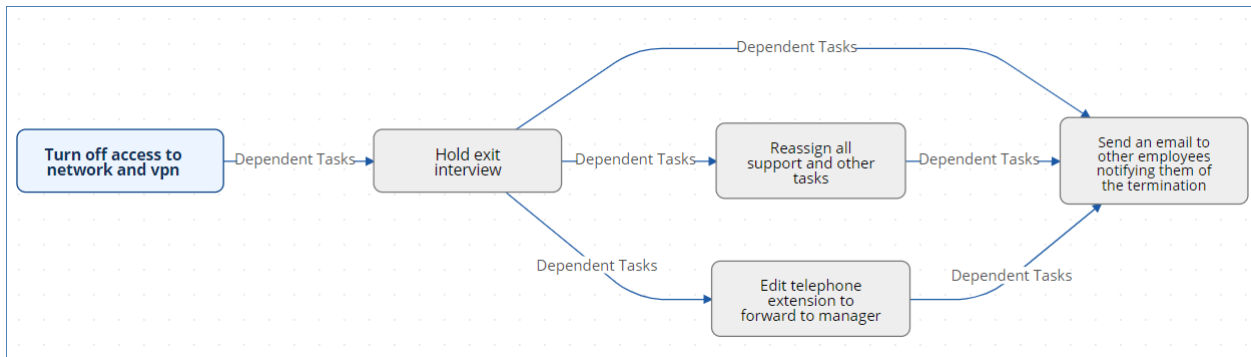
There are some differences in the kind of sequencing available between the two tables. Tasks have a more sophisticated relationship to each other than do approvals. Approvals have a step number, and all approvals of a given step number must be completed before the next step number approvals are assigned. They can combine parallel and sequential approvals, such as:

- 1 - Contract Manager Approval
- 1 - Submitter Manager Approval
- 2 - Finance Approval
- 3 - Risk Approval

Here, the approvals with step 1 are launched in parallel, and they must both be completed before step 2 is assigned, and step 2 must be done before step 3 is assigned.

Because the step number controls the sequencing, it is not necessary to relate specific approvals to other approvals in the setup, which makes approval workflows simpler to set up and maintain. But it is also not possible to have multiple branches of approvals operating in parallel but independently, as it is with tasks. For instance, it is not possible to say: "Once the Contract Manager Approval is done, launch step 2, even if the Submitter Manager Approval is not yet done."

Unlike approvals, tasks can be related to each other in more complex prerequisite structures, allowing for more sophisticated branching and triggering. A task diagram can show these relationships:

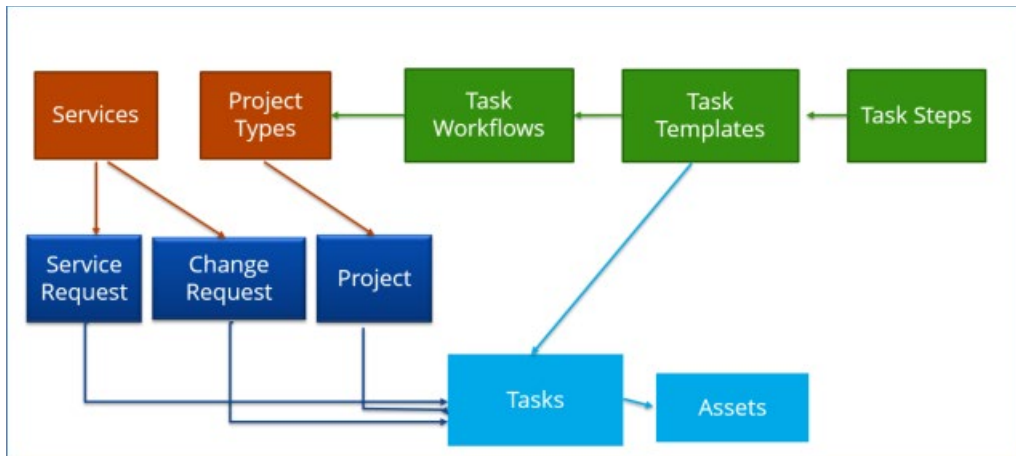


If an approval situation requires these complex relationships, it can make sense to use the prebuilt task structure instead of the approval structure to provide this flexibility. It is also possible to add a task type of Approval and mix approval tasks with other kinds of tasks in a specific sequence.

## Structure that Supports Task Management

The diagram below shows the general principles behind task management.

Task workflows and templates are the main building blocks to create the sequence of tasks or individual tasks that will be generated within a process table.



The Services and Project Types tables define the task generation method and any templates to be used:

\*Task Generation Method

- Predefined Task Workflow
- Choose one
- Predefined Task Workflow
- User Selected Tasks
- User Generated Ad Hoc Tasks



If one of the first two options are selected, an additional field defines whether to also enable ad hoc tasks.

Enable Ad Hoc Tasks?

☒ Yes ☐ No

If Predefined Task Workflow is selected, then a workflow must be chosen. If User Selected Tasks is chosen, then task templates are defined from which the user will select.

In a service request or a project, when the user selects the service or the project type, the task generation method is pulled in, along with the appropriate visibility conditions and task generation buttons.

When a task generation button is clicked, a rule action goes up through the service or project type to its workflow and/or task templates to get to the task templates that should be converted into tasks and triggers their conversion in such a way that the resulting tasks are linked to the service request or project.

As the tasks are completed, any dependent tasks are assigned, until all tasks have been completed.

The process table controls what happens when all tasks are done.

## Guide to Linking Tasks to Other Tables

Within the Tasks table, the field **Related to** indicates the type of record the task is associated with. If you need to use tasks within any additional tables, there are several steps to that configuration.

1. First, you can modify this field and add that table to the choice list.
2. Then, in the Task table, create a linked set of fields sourced 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.
3. Then, in the other table, create a related table field pointing to the Tasks table.
4. Then you will need to determine what kinds of task generation you want to support - ad hoc, workflow based, or user selected, or some combination. Depending on the requirement, you may need an intermediate table where that generation method is defined, or you may be able to use the existing Task Workflow and Task Templates tables, with the new "related to" value added to each of them, to define the task workflows that you need.
5. Then you will need to add the relevant fields to your process table to generate and manage the tasks, and update the task table rules to process tasks for this new table. You can use the fields and rules defined for Service Requests or Projects as a guideline.

## Working with the Tasks Table

The Tasks table is used within several other process tables to automate and track standard recurring or ad hoc tasks.

# Task Layout

We'll begin with an overview of the important fields on the Task layout.

## Task Details Tab

The main task screen has fields defining the nature of the task, such as Task Type, Task Summary, and the task Description. It also shows the task Status, the assigned team and person, and the Date Due.

The screenshot shows the 'Task Details' tab of a task management system. At the top, there are tabs for 'Task Details', 'Related Tasks', 'Related Info', 'Time', 'Emails', and 'History'. Below these are buttons for 'Save' and 'Cancel'. The task ID is 1034. The 'Related To' field is set to 'Change Request' and the 'Task Type' is 'Asset Change'. The 'Status' is 'Completed'. The 'Task Summary' is 'Update Linux Server with latest patches'. The 'Task Details' section includes fields for 'Auto-Complete?' (No), 'Assigned Team' (Security Team), 'Date Due' (May 01 2017 12:00), 'Actual Start Time', 'Sequence' (1), 'Assigned Person' (Agiloft System), 'Date Done' (Apr 22 2017 16:57), 'Actual End Time', and a 'Description' field with the text 'This task is generated if the server OS is one of Linux, AIX, Solaris, or HP-UX'. There are also 'Start Clock' and 'End Clock' buttons.

It also includes an area for managing the related asset, if any.

Tasks based on a template may be defined to have specific task steps, and if the task has such steps, they appear with checkboxes in the Working Notes section. There are no default rules enforcing that all checkboxes are checked before completing a task, but such a rule could be added.

The user can add working notes at any time, and these will be added to the running history of all working notes:

The screenshot shows the 'Working Notes' section. It has a title 'Working Notes' and a sub-section 'Task Steps' with a list of tasks: 'Perform Backup' (checked), 'Stop all Services', 'Install Patches', 'Reboot Server', and 'Check Services are running'. Below this is a 'Working Notes' text area. At the bottom, there is a 'Running Working Notes' section with a timestamp '[Admin System May 27 2016 09:09]' and the text 'Here are the first notes about work on this task.'

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

## 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, that is, those for which this task is a prerequisite:

**Prerequisite Tasks**

Enable Prerequisite Tasks: ☒ Yes ☐ No

Trigger Condition: ☒ When all tasks are completed

Add Task to Prerequisites:   
Remove Task from Prerequisites:

Prerequisite Tasks

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

Edit	ID	Task Summary	Status	Assigned Team	Assigned Person	Date Due
	956	Recover Personal Computing Equipment	Assigned	1st Level Support Team		Apr 10 2017 13:34
	954	Conduct Exit Interview	Completed	Internal Customer Team	Douglas Teller	

Number of Prerequisite Tasks: 2

Number of Completed Prerequisite Tasks: 1

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. When tasks are launched, any task without a prerequisite is automatically set to Assigned (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.

**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 be included in the request to which the task is linked.

The Emails and History tabs hold the standard fields.

## Use Case

Although Tasks may be linked to one of several other tables, they are typically created and processed in similar ways. Tasks are created automatically from templates, or manually by users, and several automated actions occur when tasks are created by either method.

## Tasks Created from a Template

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

Note that 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 template titles for the same reason.

## Ad Hoc/Manual Tasks

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 process 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.

## New Task Automation

The following is a summary of the rules and validations that run when new tasks are created, either manually or from templates:

- When a task is created manually, if it is related to a Project whose status is Completed or Canceled, 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 performed 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, for instance 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 user completes the task, they change the Status to Completed and save the record. If there is no value in the Date Done field, the system will put the current date/time into the Date Done field. Alternatively, the user may enter a time in that field directly.
- 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 an asset, 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				Actual End Time			
				<div>Start Clock</div>			
				<div>End Clock</div>			

- 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.

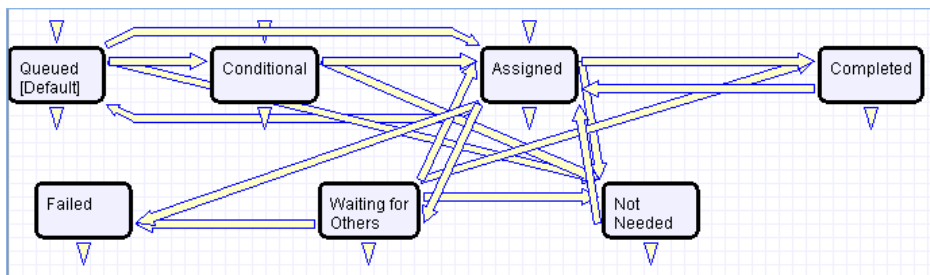
## Measuring Time for a Task

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:



## Disabled Time-Based Rules

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

- **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.

- **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.

## Working with the Task Templates Table

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. Task Templates 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 be independent of 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.

## Task Template Layout

The next sections outline the important fields and options on each of the Task Template record layout tabs.

## Template Details Tab

All the fields that users can fill out are on the Template Details tab:

*Task Usage	Default ▾	Auto-Complete?	<input type="radio"/> Yes <input checked="" type="radio"/> 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	<div></div>		
Attached Files	<a href="#">Attach/Manage</a> <div>Drag&amp;Drop files</div>		
Sequence	3		
<b>Prerequisite Tasks</b>			
Choose a Task to Add	▾	<a href="#">Add Task to Prerequisites</a>	
Prerequisite Task to Remove	▾	<a href="#">Remove Task from Prerequisites</a>	
Trigger Condition	When all tasks are completed ▾		

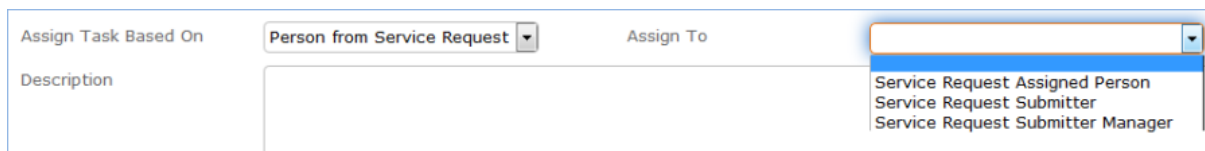
The Task Title may not contain commas; this prevents errors when running the task automation rules. The Related To field links Task templates to Service Requests, Change Requests, or Projects.

Task Usage may be Default or Conditional. If conditional, then a saved search condition based on metadata 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 the status is marked Assigned, by adding the defined number of working hours (for the assigned team) to the current time.

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 pre-configured in the Replacement Variables table. For instance, for a task related to a service request you will see these options:



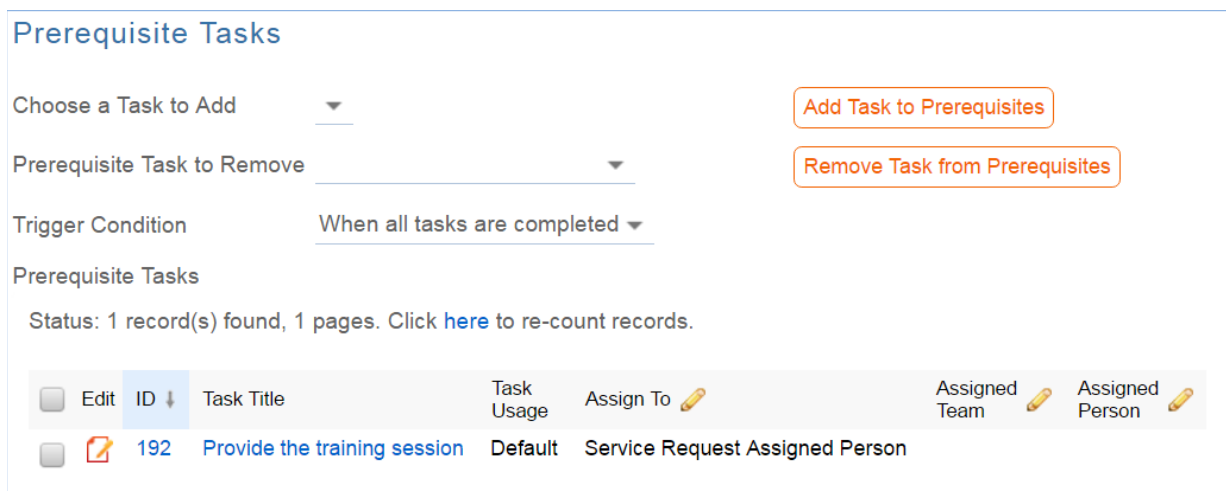
Assign Task Based On: Person from Service Request

Assign To: 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: [dropdown]

Prerequisite Task to Remove: [dropdown]

Trigger Condition: When all tasks are completed [dropdown]

[Add Task to Prerequisites](#) [Remove Task from Prerequisites](#)

Prerequisite Tasks

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

<input type="checkbox"/>	Edit	ID ↓	Task Title	Task Usage	Assign To	Assigned Team	Assigned Person
<input type="checkbox"/>		192	Provide the training session	Default	Service Request Assigned Person		

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

Create Task Step




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

Mass Edit

Save Changes

Cancel Changes

Views ▾

<input type="checkbox"/>	Edit	Step Number  ↑	Step Name 	Description
<input type="checkbox"/>		1	1st step	

To add steps, first check the Create New Checklist Item? checkbox, then fill out a Step Name and Step Number, then click Create Task Step. 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:

Save ▾

Cancel ▾

Template Details

Related Information

ID	191	Template Status	Active ▾
*Task Title	Agree with requester on what training is needed and		
Task Workflow Title	Training for Business Applications 		
Related Service	<input type="text"/>		

## 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 includes rules to strip out any commas from the task title, a rule to update the sequence of task templates in relation to other templates for the same workflow, and the rule to unlink inactive task templates.

In addition, the system tracks the service requests and projects for which a task template based on user selection has been generated so that users are prevented from generating the same task for the same parent record. Since change request tasks must be generated multiple times, one for each asset, 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 and any task steps are also cloned and linked to the new workflow.

Cloning reproduces all the elements of the original task template.

## Working with the Task Workflows Table

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 selected in services for Change Requests or Service Requests, and they are also associated with Project Types. They essentially predefine the tasks that should be completed 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.

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:

## Predefined Workflows

Create new Task Workflow

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:

### Task Workflow

Save ▼

Cancel ▼

Workflow

History

« »

\*Related To Project ▼ Status

\*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:

### Task Workflow

Save ▼

Cancel ▼

Workflow

History

« »

\*Related To Project ▼ Status Active ▼

\*Workflow Title workflow for customer implementation

Clone Workflow

---

#### Workflow Details

\*Description Used for customer implementations of Agiloft

Notes

---

#### All Task Templates

Create New Task Template

\*Enable Prerequisite Tasks ☒ Yes ☐ No

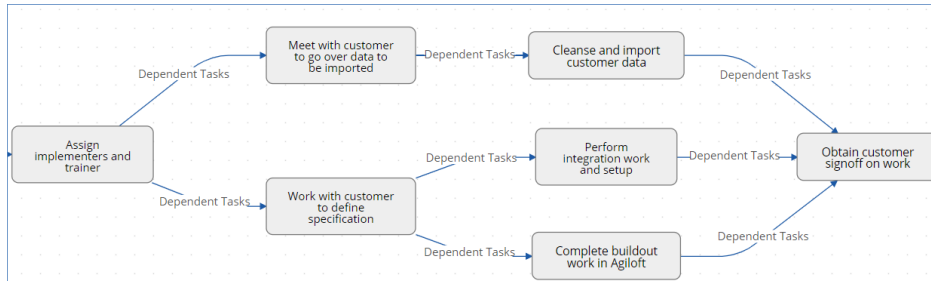
Clicking the Create New Task Template button saves the workflow and then brings up the new task template screen. See the [Task Templates](#) section for more information on how to create and fill out the task template form.

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.

Tasks in a workflow can be displayed in a diagram showing their relationships and dependencies:



As shown in the diagram above, it is possible to launch two separate threads and then bring them back into alignment by making their final tasks prerequisites of the next task.

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 and marked as not needed by a technician before the tasks are launched.

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



All Task Templates
















Create New Task Template

\*Enable Prerequisite Tasks ☒ Yes ☐ No

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

Save Changes | Cancel Changes | Remove Task

Views  | Search 

 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 without prerequisites, ID62 and ID61, that will be assigned right away. 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 another task is assigned provides a flexible model for complex task workflows.

The sequence number gives a general idea 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 it 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.

## Working with the Task Steps Table

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, and when there is no need to track the time it takes for each step to be completed. They can simplify workflows and reduce the overall number of tasks, while providing reminders about steps that should be done.

Individual task steps are linked to a specific task template and can only be used in that template. Task Steps are simple records with a Step Number, Status, Title, Description and link to the Task Template.

Task steps are created from within a task template, as shown below, or they can be created directly with the task template specified. There is no automation on this table.

They appear within task template records as a related table ordered based on the Step Number:

The screenshot shows a web interface titled "Task Steps". At the top, there are two radio buttons for "Use Task Checklist" (selected "Yes") and a checkbox for "Create New Checklist Item?". Below these are three buttons: "Mass Edit", "Save Changes", and "Cancel Changes". To the right is a "Views" dropdown menu. The main part of the interface is a table with the following structure:

<input type="checkbox"/>	Edit	Step Number	Step Name	Description
<input type="checkbox"/>		1	Check calendar for open time frame	
<input type="checkbox"/>		2	Discuss most urgent training needs with internal support staff	

At the bottom left, it says "Number of Active Steps" followed by the number "2".

When a task is created from the template, any task steps are displayed as a list of items with checkboxes, i.e.:

Task Steps	<input type="checkbox"/> Define the process tables
	<input type="checkbox"/> Define the background tables
	<input type="checkbox"/> Discuss workflows for approvals or tasks
	<input type="checkbox"/> Collect all field information
	<input type="checkbox"/> Setup Reports



# Graphical Elements

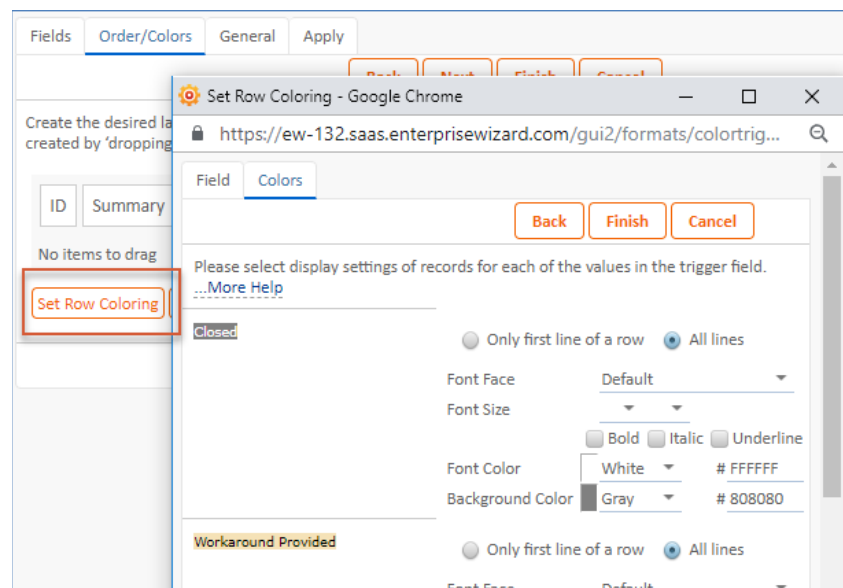
---

## Row Coloring

While Row Coloring and Notification Icons are not strictly Data Elements, their use requires data association.

In order to provide a visual aid to easily identify specific record attributes, records can be color-coded based upon the value of a selected choice field. In a table, navigate to **Views > Edit** to open the View wizard.

- **Order/Colors tab:** click the Set Row Coloring button to open the Set Row Coloring window.
- **Field tab:** select the choice field that will serve as the trigger for the row coloring.
- **Colors tab:** change the text size, font, color, and background color for each value of the choice field.

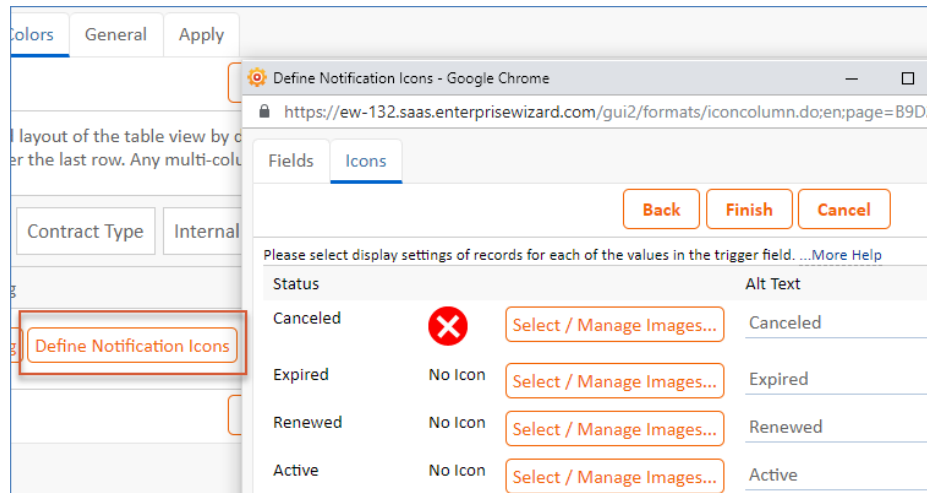


## Notification Icons

You can visually represent the values of certain fields in a record by creating notification icons that correspond to the value(s) in a choice field. Whenever a record contains the specified value in the field, the icon will be displayed in the table view for that record. To set up notification icons, navigate to **Views > Edit** to open the View wizard.

- **Order/Colors tab:** click Define Notification Icons to open the Define Notification Icons window.
- **Fields tab:** select one or more choice fields from the list.
- **Icons tab:** select the icons to associate with the values in each field you selected and define the alt text. You can associate an icon with one value, several of the values, or all the values. The system comes with many icons to choose from, but you may also upload your own icons.





*Less is More: when adding graphic elements, a minimalist approach to artistic or aesthetic matters may be more effective.*

## Image with Versioning

Image with Versioning fields store attached files that are displayed as pictures within the record. This data type supports versioning that stores past versions of image files, but you can disable the option if you don't want to use it or if you need to save space.

Only files with appropriate image extensions can be uploaded to this field type. You can also configure these fields to display the images in the table view, in their original format or cropped into circles.

## Creating an Image with Versioning Field

On the Options tab:

1. Choose whether to enable versioning. Select Yes to track changes to the image file, or No to save only the newest version of the file.
2. Choose whether to enforce check-out restrictions on editing. You can allow multiple users to freely edit at the same time; enforce check-out restrictions, but not prevent other users from concurrent editing; or enforce strict check-out restrictions so only one user can edit the file at a time.
3. Set whether to allow users to upload multiple files. If you allow multiple files and show this field in the table view, all the uploaded files are shown in the column, which can take up a lot of vertical space on each row.
4. Set limits on the minimum and maximum file size.
5. If desired, you can enforce image proportions to suit your form layout. **To prevent image distortion, it's best not to set both height and width unless all images will have the same aspect ratio.** You can set dimensions using pixels or viewport units.

Viewport units represent a percent of the viewable content area, meaning the browser window size without toolbars and buttons, and they're useful when working with lots of different display devices or sizes. The units are **vw**, and **vh**. They represent a percentage of the browser (viewport) dimensions and scale accordingly on window resize.

Let's say we have a viewport of 1000px (width) by 800px (height):

- **vw** - Represents 1% of the viewport's width. In our case 50vw = 500px.
- **vh** - A percentage of the window's height. 50vh = 400px.

You can use these units anywhere that you can specify a value in pixels, like in width, height, margin, font-size and more. They will be recalculated by the browser on window resize or device rotation.

6. Choose whether to show the image file in the table view:

- **No:** The field shows a hyperlink to the image file.
  - **Yes:** The image itself appears in the table view. You can enforce different image proportions specifically for the table view. For example, you might set an exact image width so that all the images fill the column space. You can also choose to crop images in a circle, in which case the corners are cropped. If you choose to crop images into a circle, make sure the image subjects are centered so they aren't cut off. There isn't an option for users to adjust the center point of the image.
7. If desired, add a search pop-up to the field to allow users to pull image files from existing records in the same table.

## Image Dimensions for Personnel Photos

Display image:

☐ At its original size

☐ With a width of \_\_\_\_\_ pixels

☒ With a height of 250 pixels

☐ With a width of \_\_\_\_\_ and a height of \_\_\_\_\_ pixels

*Note: the height of 250 pixels is only suggested when using a left-hand common area – otherwise this will be too large for normal personnel records.*

Display as image in table view:

☒ Yes

☐ No

Display image in table view:

☐ At its original size

☐ With a maximum width of \_\_\_\_\_ pixels ▼

☒ With a maximum height of 50 pixels ▼

☐ With a width of \_\_\_\_\_ and a height of 10 VW/VH ▼

Crop image as circle:

☐ Yes

☒ No

## Heat Bar

Heat bar fields show data as colored segments of one bar, **useful for showing proportions, progress, and more.** Heat Bars are built using either a Related Table or using multiple Numeric fields.

## Create a Heat Bar

Because a Heat Bar is a graphical representation of data from other fields, make sure you have already created the Related Table or the Numeric fields you want to show in your Heat Bar before you begin.

1. Select whether to use a Related Table or Multiple Numeric Fields to define your heat bar values. If you choose Related Table, choose the table from the drop-down list.
2. Click Next. Note that after you click Next, you can't change your heat bar type from Related Table to Multiple Numeric Fields or vice versa. If you selected the wrong one by mistake, you must click Cancel and start again.

If you're using a Related Table:

1. Select the field you want to use to segment the heat bar. For example, if your related table shows tasks, you might use a Status field.
2. Choose whether to use all values or to filter them with a saved search.
3. Choose an option for the color proportions. In some cases, you want to show the count of each field as a proportion of the total count; in other cases, it makes more sense to use the sum of a field value. For example, if your heat bar shows training information and the courses differ in length, you might want to show it as a proportion of the total hours the training will take, rather than the count of courses.
4. Select the calculated result fields to be created. You must create a calculated result field for each item you want to include in the heat bar.
5. Rename the new calculated result fields as needed. Remember that these will appear on the Fields tab with all other fields in the table.
6. Choose whether to run the calculation for all existing records. This will populate your heat bar for each existing record, which might take some time depending on the amount of existing records.

If you're using Multiple Numeric Fields:

1. Use Ctrl+Click to select all the fields you want to use in the heat bar.
2. Complete the Options and Permissions tabs as desired, then go to the Display tab.

Now that you've defined the data source for the segments, complete the Display tab:

1. Set the display settings for each segment by clicking Choose Color. Consider how the colors will look next to each other. Segments appear in the order of the fields here. You can click and drag to rearrange the order of the fields.
2. Choose when to show numerical values on the bar, if at all. A future enhancement will allow a user to identify the segment with a mouse-over.
3. Set the bar width and height. This might require some trial and error. Make sure to test your heat bar everywhere it will appear, including any views, dashboards, or widgets.

## Show the Heat Bar

You can include the Heat Bar in the record layout, table view, and more. **Heat bar values are refreshed whenever a record is saved.**

## Singleton Check

This data type shows a single checkbox, representing True when selected and False when cleared. **The Singleton Check is useful for controlling visibility-dependent sections of the record form, such as a checkbox labeled Approval Required that shows or hides the fields used to specify an approver.**

This data type can also be shown in the table view with a checkmark icon, so users can see the selection at a glance more easily than skimming a column of Yes and No.

## Creating a Singleton Check

On the Options tab:

1. Select the default value for the field. If you select True, the box is selected by default; if you select False, the box is cleared by default. If you don't set this option, the default value will be True. If you add a Singleton Check field to a table that already has records, the default value you specify here will be applied to all the existing records when you click Finish to create the field.
2. You can also use hierarchical dependency to set the field to True or False based on a Choice field. For example, you might set an Approval Required checkbox to always be False (clear) when a certain Task Type is selected. To do so, select the Choice field you want to use as a dependency, click Dependent Choices Wizard, and complete the hierarchical dependency setup in the wizard.

## Relationship Diagram

A relationship diagram field is a special data type that allows users to visualize the relationships between linked records in a tree diagram. The possible uses of the Relationship Diagram field are numerous, as it can model a large range of scenarios where there are dependencies and linked relationships between organizational assets, tasks, contracts, and other items.

## How to Configure Relationship Diagram Fields

You will need the following on the table where you configure the relationship diagram field:

1. A linked field set OR related table that identifies parent / upstream relationships.
2. A linked field set OR related table that identifies children / downstream relationships.
3. Arrow labels – You will choose a field or enter hard-coded text used to label the arrows in the diagram.

4. Item labels – You will choose a specific field to identify each item in the diagram item, typically a summary field. For example, you might use Contract Title for contracts or the Model Name/Number field for assets to label the item boxes.
5. Icon field - Optionally, you can choose an image field used to show icons next to each item name in the diagram. If you use this option, you can also choose the maximum icon size.

To begin, navigate to the Setup Table wizard for the table where you will add a relationship diagram.

1. **Options tab:** Choose the following:

- a. Select the linked relationship that identifies the parent records.
- b. Choose the relationship that identifies children records.
- c. Once you choose the linked set/related table to define the parent and/or child records, you can optionally add a saved search to further limit the items included in the diagram from the related table. You must also choose a field in the table that holds the Relationship Name, or enter a hard-coded label for the arrows.

2. **Display tab:** For the current item, its parent, and children, choose the field in the table that identifies the item. We recommend using a summary field like Title. Choose any additional display options, such as the arrow type, layout direction, and icon field. It is important to select the fields which will be used to display the name of the diagram items. If this selection is left undone, the diagram fields may be blank or display incorrectly.

Next, add the relationship diagram to the record layout. You can also add the diagram field to the table view. This can allow you to gain a quick overview of table relationships without going into the record.

## Defining Field Relationships

Because of the flexibility of relationship diagrams, there isn't a single default way to add field relationships. A KB could include several related tables which separately hold relationship information, enabling a very robust diagram setup that provides reporting on corporate assets.

On the other hand, a single table may be designed to hold all of the necessary fields, and a very simple relationship diagram may be set up with a small number of fields designed for a specific purpose. The examples below illustrate how to create a relationship diagram for Contracts, depicting parent and child contracts, all within the same table.

At a minimum, field relationships should include the following:

- The items to be modeled must exist in a relationship that would be useful for diagramming. For instance, whenever a related contract is added in the Contracts table, it creates a relationship to its parent contract.
- The relationship must be modeled using a linked field or related table. For instance, the Contracts table contains a related table to the Contracts table that depicts the relationship between the current contract and its parent contract.
- There must be a field holding relationships for the arrow labels, if you are not simply using text labels. Relationship fields must be available for both parent and child fields or tables. This should be a choice field and can be a part of the normal table setup.
- For example, a related Contract could use the Record Type field to label the arrows, which would produce results like Master Agreement for upstream relationships and Subcontract or Amendment for downstream contracts.

## Resources


See below for additional resources.

## Status Bars


A variety of Status Bars and icons can be found on the Admin Welcome Home page via the Status Bars PowerPoint presentation link.

### Resources and Downloads


The Standard System Documentation describes the out-of-the-box tables and features.




Standard System Documentation



Orange/Red Icon Set



Blue/Grey Icon Set

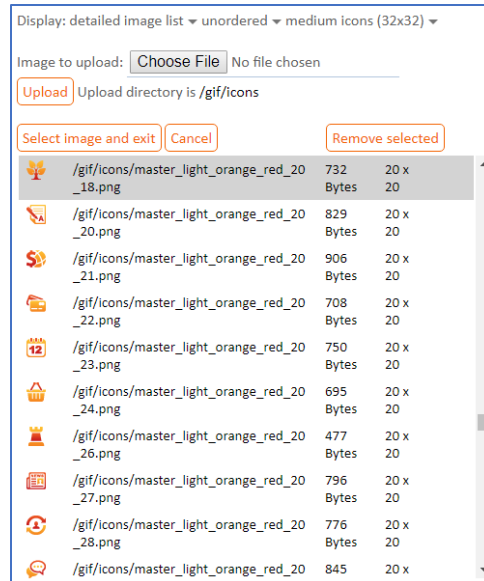


Status Bars

## System Icons

Icons may be uploaded in multiple places:

- Look and Feel wizard on the Icons or Content tab
- Field wizard for datatypes that allow an image hyperlink, i.e. Action Buttons or Relationship Diagram.



## Snagit Image Capture and Editing Software

<https://www.techsmith.com/screen-capture.html>

15 Day full version free trial – no credit card required

<https://www.techsmith.com/download/snagit/>

## Look and Feel

### Where to find look and feel settings

Power user look and feel: Setup > Look and Feel > Power User Interface

End user look and feel: Setup > Look and Feel > End User Schemes

Email look and feel: Setup > Look and Feel > Email Schemes

Required field style: Setup > Look and Feel > Required Field Display

Power user page headers: Setup > System > Edit Page Headers

Left pane groupings: Setup > Look and Feel > Left Pane Setup

### Applying look and feel schemes

Look and feel schemes can be applied to individual teams or to all users. There is also an option for whether to apply the look and feel scheme to new teams that are created after the look and feel scheme is finished.

## End user vs. power user look and feel schemes

The end user and power user look and feel wizards are similar, but there are some elements that are different between the two interfaces, and therefore different styles available to edit. For example, the Left Pane and Charts/Reports tabs is not in the end user look and feel wizard, because end users do not have a left pane and cannot access charts and reports.

## End user look and feel vs. CSS

The end user look and feel scheme applies to all of the elements that come from Agiloft: table views, record views, etc. The end user home page, headers, and menu items are controlled by CSS (the style.css EUI template record).

By default, the EUI Templates table is under the “System” heading in the left pane.

## Look and feel resources

<https://wiki.agiloft.com/display/HELP/Look+and+Feel>

<https://wiki.agiloft.com/display/HELP/End+User+Interface>





# Take Control of Roles and Permissions

---

## Groups

Group membership defines the access permissions each user has within Agiloft. For example, groups define which records a user may view, edit, or delete, and which fields within the record the user may view or edit. Strategic group configuration is essential to efficient permissions management.

Users can belong to many groups and enjoy the combined permissions of all the groups that they belong to. With this in mind, there are two possible strategies for group permissions: using groups as permission layers, which are combined to give appropriate permissions; or creating self-sufficient groups, so each user might be a member of just one group that contains all the permissions they need.

	Layer Groups	Self-sufficient Groups
<b>Pros</b>	Easier maintenance, where you can edit just one group to modify all users' permissions	Easier troubleshooting, where each user is in just one or two groups for you to check
<b>Cons</b>	Harder to troubleshoot, with many groups to check for issues	Harder to maintain, with many groups to modify to control overall permissions

Groups are designated as either End User or Power User, often called a staff user. The type of group determines which license type is required. End User groups cannot access the Power User Interface or edit records created by other users. By placing users who only need to create, or submit, records and use the FAQ only in an End User group, you limit their access and reduce licensing costs.

Power Users use an individual named or floating license and can access either interface and can perform all functions permitted to them. If a user is a member of both End User and Power User groups, the system logs the user in as a Power User with the associated access and license usage.

## Configuration Tips

- Every knowledgebase comes with several predefined groups. You can modify or delete them as needed, and you can create an unlimited number of new groups.
- If you aren't sure whether a group is being used, you can add a z to the front of the label to move the group to the bottom of the list. If you use this method, you can filter the Group field in the People table to show only groups whose Group Name does not contain z.
- Generally, you should not delete predefined groups with admin, guest, or anonymous in the name.
- It can be useful to create groups as early as possible in the process so the groups can be referenced in other configurations, such as restricting a group from editing a field when some specified condition is true. This saves time that would be spent later going back and adding these restrictions.
- Field permissions are controlled in the field directly and in group permissions. You can use either option, but sometimes one is preferable:
- If you're already working in a table, or if you're updating permissions in many groups for one field, it's usually easier to edit permissions in the field.

- Take advantage of the ability to copy permissions from another group. This can make permission management much simpler.
- It's usually best to keep Power Users and End Users in separate groups because they have different roles and access the system through different interfaces.

## Group Permissions

Group permissions control access to:

- saved searches;
- tables and records;
- views and reports.

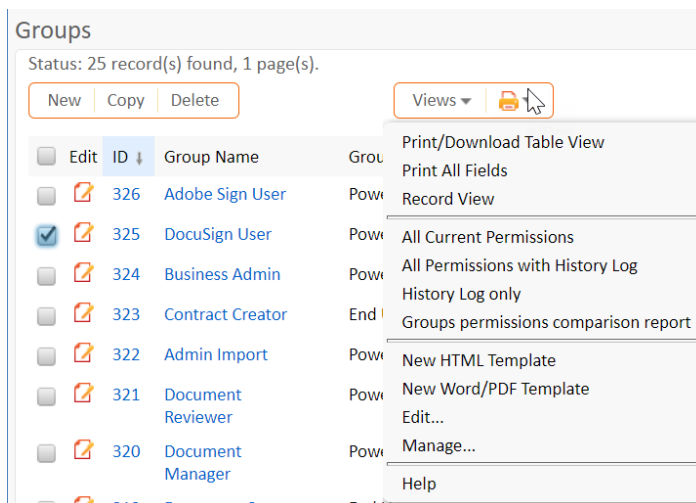
You can modify the permissions for each group in the Group Permissions wizard.

Configuring permissions for a new group can be time consuming, due to the fine level of access control offered by Agiloft. To simplify this process, copy an existing group with permissions similar to those you need for the new group, and then modify the copy to suit the new requirements. Groups are only used for access permissions. If users share a common function and need to be notified or assigned to issues collectively, they are placed in the same Teams.

## Printing Group Permissions

You may wish to print a group's permissions as either a hard copy or to a file to get a sense of what tables are active for each group. This allows you to quickly scan the existing settings for each table, and saves you the trouble of going through the wizard one screen at a time. When done customizing you can store these files as a reference of what permissions have been set up.

In **Setup > Access > Manage Groups**, mouse over the printer icon:



These options allow printing of permissions or a history log of changes. You can also print a comparison report that shows differences between selected groups.

## Starting from Default Groups

The system contains a variety of default groups preconfigured for certain applications, such as CRM. These groups are often adequate for a new implementation, and it's a good idea to become familiar with their permissions before taking the time to create new groups. If necessary, you can copy existing groups to create additional groups when you need more refined permissions.

The lists below provide examples of default End User and Power User groups.

*The admin group should never be renamed or it will not function as it should.*

## Default End User Groups

These groups use the unlimited end-user license:

- **Customer:** Users who are not employees and see only their own records.
- **Customer Manager:** Users who are not employees and see all the records submitted by users in their company. They may also be given other privileges.
- **Internal Customer:** Users who are employees and see their own and others' records.
- **Contract Creator:** Users who are employees and work primarily with contracts. They are able to create and edit their own contracts, as well as see others' contracts.
- **Guest:** Users who self-register from hyperlinks and create records like leads and support tickets. They do not have access to the rest of the interface. A hyperlink logs them in to the new record screen and logs them out when they click Save or Cancel.

## Default Power User Groups

These groups use individual licenses:

- **Admin:** Administrators who are customizing and maintaining the system.
- **Support Staff:** Technical staff who access external and internal support cases.
- **Contract Manager:** Users who are employees and work primarily with contracts. They have full control over contracts and companies.
- **Sales:** Users in sales who access leads, opportunities, and so on.
- **Marketing:** Users in marketing who access leads, opportunities, and so on.

## Group Permissions Wizard

The Group Permissions wizard, located at **Setup > Access > Manage Groups**, enables you to create new groups and manage the permissions of existing groups. To open the Group Permissions wizard, edit an existing group or click New to create a new group.

Groups			
Status: 29 record(s) found, 1 page(s). Click <a href="#">here</a> to count records again..			
<div> <div>New</div> <div>Copy</div> <div>Delete</div> </div>		<div> <div>Views ▾</div> <div></div> </div>	
<input type="checkbox"/> Edit	ID ↓	Group Name	Description
<input type="checkbox"/>	330	Professional Services	will be responsible for performing the professional sevicees and n
<input type="checkbox"/>	329	Sales Manager	Manager of sales
<input type="checkbox"/>	328	Support Manager	Manages the support staff
<input type="checkbox"/>	327	Support Staff	Support Staff

## General Tab

The General tab allows you to set the basic group permissions. Selecting Power User or End User affects which options are available.

Setup → Access → Manage Groups

General

Tables

History

Next

Finish

Cancel

This is the ID number of the group in the current KnowledgeBase.

Group Identifier  
257

Choose a name for the group. This name will uniquely identify the group in the current KnowledgeBase.

\*Group Name  
admin

Provide a description to help you remember the purpose of this group.

\*Description  
For administrators who can see and edit all records in all tables.

Select the general role of this group in the system.

☒ Power User
☐ End User

Staff groups can be given any access permissions and may access any interface. They use a staff license when they access the system.

End users require an end user license (which is generally an unlimited user license) and can only log in to the End User Interface. They can be given permissions to submit, edit and view records, but cannot edit records they do not own.

Select whether members of this group can reset their password by clicking **Send Password** on the login page. For security reasons, it is recommended that highly privileged groups, for instance Staff or Admin, are not allowed to receive passwords by email.

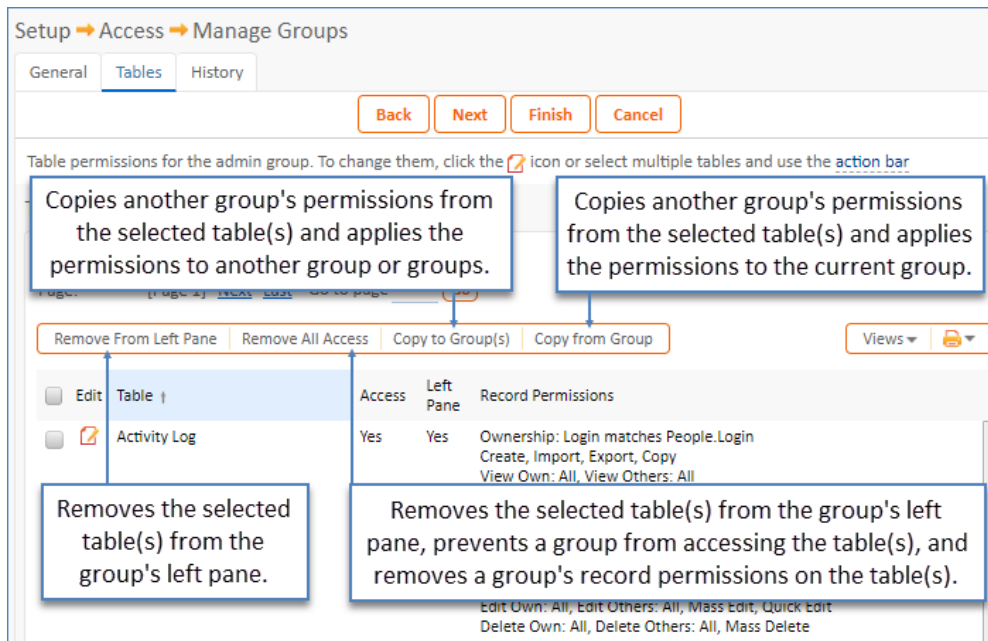
Allow Sending Password?  
☐ Yes  
☒ No

Select whether this group can modify the Left Pane.

Allow users to Personalize their left pane  
☒ Yes  
☐ No

## Tables Tab

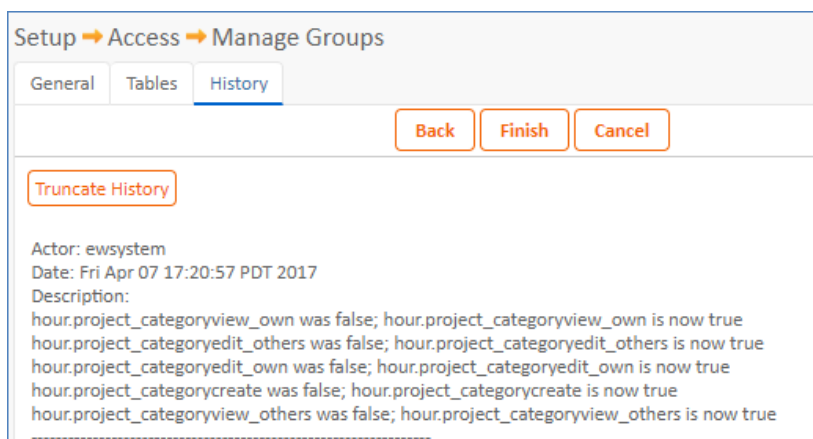
The Tables tab lists each table in the knowledgebase and shows an overview of the current group's access to each table. Editing a table opens the Table Permissions wizard where you can configure the group permissions for that table.



## History Tab

The History tab keeps a record of all changes made to the group's permissions. This is an essential feature when using Agiloft for applications that require complete audit histories for compliance, such as in governmental systems. It is also useful when multiple administrators may be editing the system so that other administrators can track any changes that have been made.

You can remove all history entries by clicking Truncate History. This is usually not necessary, but it may be helpful if you have completely redesigned a group's permissions and want to start with a new history.



## Table Permissions Wizard

The Table Permissions wizard enables you to configure the permissions for each table in a knowledgebase for a given group. The wizard is accessed from the Tables tab of the Group Permissions wizard by editing a table from the list. For more information on configuring table permissions for groups, see Set Group Table Permissions.

*Table permissions are saved individually, independent of changes made in the overall wizard.*

### General Tab

The General tab specifies high level security, such as whether members of the group can access the Setup wizards and whether the table will be displayed on the group's toolbar, or left pane.

General	Menu Permissions	Record Permissions	Field Permissions
<div>NextFinishCancel</div>			
The permissions you set using this dialog will apply to the table: Approval. You are setting permissions for the group: admin.			
A record is owned by the person whose own Login field matches the Creator Login field			
Select whether this group can access the table selected above. If set to No, group members won't be able to access the table, unless they are also members of a group that is granted access to it. It is usually not necessary to turn off access to a table - you can simply turn off the next option to show the table on the toolbar to hide it from this group. Setting this option to No will eliminate all field level and record level permissions that have been set up previously.		Allow access to the table? <input checked="" type="radio"/> Yes <input type="radio"/> No	
Select whether this group sees a tab on the toolbar giving them access to this table. If set to No, group members won't have a Toolbar tab corresponding to that table, unless they are also members of a group with the permission for it. You may still give users access to the table through linked fields. For instance, you might not want a tab visible on the toolbar for assets, but you may give staff (or customers) view access to their own assets so they can select a linked asset from a search lookup in their support case.		Show table on the Toolbar? <input checked="" type="radio"/> Yes <input type="radio"/> No	

Table access controls aspects such as creating a record via a Custom Hotlink, viewing a table's Reports, and selecting information from a Linked Field. In most cases, it's usually not necessary to remove table access. If you don't want users to use the table outside of the previous scenarios, you can effectively limit their access by removing the table from their toolbar by selecting No from the Show Toolbar option.

If the table has subtables, an additional option controls whether to apply the changes to the subtables.

Apply to subtables
<input type="radio"/> Do not apply changes
<input checked="" type="radio"/> Apply changes only
<input type="radio"/> Apply all permissions

## Table Permissions and Security

Security is built into the core of the system and enforced consistently. Users who do not have access to a table cannot see records and data from that table. This seems obvious, but it can have important implications for tables and permissions.

### Example

Consider the Assigned Team field in the Service Requests table. This field is displayed as a drop-down list containing each team name, and it's linked to the Team Name field in the Teams table. Users will not be able to choose a team from the drop-down list or even view the list at all unless they belong to a group with access to the Teams table. That is, they must be granted access to the Teams table and have view permission to the table's records and the Team Name field.

However, if the Assigned Team field in the Service Requests table already has a value selected, the user will be able to see the value if they have view permission to that field, even if they don't have access to the Teams table. They just won't be able to interact with the field by choosing a new value.

This is also an example of a situation when you might want to give a group access to a table but not show the table on the toolbar.

## Menu Permissions Tab

The Menu Permissions tab specifies group permissions for menu items like views and saved searches.

General	Menu Permissions	Record Permissions	Field Permissions
<div>BackNextFinishCancel</div>			
The permissions you set using this dialog will apply to the table: People. You are setting permissions for the group: admin.			
Select whether this group can create, modify and delete saved searches. <a href="#">...More Help</a>		Allow creating/editing/deleting saved searches?	
		<div><input type="radio"/> Don't allow</div> <div><input type="radio"/> Allow for their own saved searches</div> <div><input checked="" type="radio"/> Allow for all saved searches</div>	
Select whether this group can create, edit and delete views. <a href="#">...More Help</a>		Allow creating/editing/deleting Views?	
		<div><input type="radio"/> Don't allow</div> <div><input type="radio"/> Allow for their own views</div> <div><input checked="" type="radio"/> Allow for all views</div>	
Select whether members of this group can make views visible to other users.		<input checked="" type="checkbox"/> Allow publishing Views	

## Record Permissions Tab

The Record Permissions tab specifies the group's record permissions for the table. These include creating, deleting, editing, viewing, mass editing, quick editing, and viewing FAQs for the records.

Setting record-level permissions with the Table Permissions Wizard is best suited for defining permissions that a particular group has for multiple tables. If you want to set record-level permissions



that multiple groups have for a particular table, you can use the Table wizard to set which groups can view, edit, and delete records in that table.


The screenshot shows the 'Record Permissions' tab of a table wizard. At the top, it says 'The permissions you set using this dialog will apply to the table: People. You are setting permissions for the group: admin.' Below this, a red box labeled '1' highlights the ownership setting: 'A record is owned by the person whose own Full Name field matches the Contact Owner field'. The main area is divided into four sections, each with a red box and a number: '2' for 'Create People' (Import multiple People from a file, Copy People), '3' for 'Delete their People' and 'Delete other people's People' (with a dropdown for 'Drafts' and 'Mass delete multiple People in a single operation'), '4' for 'Edit other people's People' (with a dropdown for 'Active Employees Rename'), and '5' for 'Mass Edit multiple records in People in a single operation', 'Quick Edit records from the Table View', 'Link multiple records using the Link menu', and 'Print records using the printer icon'. A red box labeled '6' highlights 'Export multiple People to a file', 'Show conversion button on menu action bar', and 'Allow interaction with conversion rules'.

1. The table's ownership settings are displayed at the top of the screen. For edit, delete, and view permissions, each permission is based on whether the group can act on records that they own, records that other people own, or both. You can change a table's ownership settings on the Permissions tab of the Table wizard.
2. Permissions for editing, deleting, and viewing records can all be controlled with separate saved searches. To allow group members to interact with every record in the table for a given permission, select All.
3. For a user to edit a record, they must first have view permission for that record. Even if the edit permission options are selected, users won't be able to edit records if they can't view them.

4. The Edit other: permission is only available for Power User groups. End User groups can only edit their own records.
5. Edit permissions have several additional options:
  - a. Mass Edit: controls whether the group can edit multiple records at once.
  - b. Quick Edit: sets whether the group can edit records directly from the table view.
  - c. Link multiple records: controls a group's ability to link records together using the **Actions > Link** action on a table's action bar.
  - d. Print records: determines whether a group can print selected records, either to a physical printer or PDF.
6. View permissions also have a few additional options:
  - a. Export multiple: controls whether a group can export records from the table.
  - b. Show conversion button: determines whether a group sees the Convert button on the table's action bar.
  - c. Allow interaction: defines whether a conversion action that is supposed to bring up the new record screen will actually do that. If the permission is not granted, and a user in this group runs a conversion action, it will just run in the background and create the record without showing it to the user.

## Record Permissions Summary

The options selected on the Record Permissions tab are displayed in Record Permissions column on the Tables tab of the Group Permissions wizard. The permissions are grouped into one line each and are separated by commas. For instance, View Own: All means that the All checkbox was selected. View Own: SS means that the Saved Search checkbox was selected.

<input type="checkbox"/> Edit	Table ↑	Access	Left Pane	Record Permissions
<input type="checkbox"/>	 Contract	Yes	Yes	Ownership: Requester ID matches People. Create, Import, Export, Copy View Own: All, View Others: All Edit Own: All, Edit Others: All Delete Own: All, Delete Others: All, Mass Delete

If the permission has not been selected, it will not appear in the list. This allows you to scan the table view to see what permissions have been granted without having to read through all the permissions.

## Field Permissions Tab

The Field Permissions tab allows you to set the group permissions for each field in a table, which are broken down by records that one owns and records that other people own. This enables you to set a very granular level of permission for the table for each group.

Setting field-level permissions with the Table Permissions Wizard is best suited for defining permissions that a particular group has for multiple fields. If you want to set field-level permissions that multiple groups have for a particular field, you can use the Table wizard to set how groups can interact with that field.

General **Menu Permissions** Record Permissions **Field Permissions**

Back Finish Cancel

The permissions you set using this dialog will apply to the table: Project. You are setting permissions for the group: admin.

"View own" sets the ability to view the field in records that the user owns (typically records that he/she created).  
 "Create" permissions refer to the ability to set the field value when creating new records. A group must also have View own permission to see  
 "Edit own" sets the ability to edit the field in existing records that the user owns.  
 "View other" sets the ability to view the field in records owned by other users.  
 "Edit other" sets the ability to edit the field in records owned by other users.

Fields that are shown as indented are part of a set of linked fields whose edit and modify permissions are dependent on the primary field in the set. If create and modify permissions are on for the set, any field that can be v

Field	View own	Create	Edit own	View other	Edit other
Attached Contract	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Attached Documents	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Billing Notes	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Business Contact Email	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Business Contact Phone	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Client Business Contact	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Button: Add Time	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Local fields have permissions that are set independently of one another.

Linked field sets have create and edit permissions that are dependent on the primary linked field, while view permissions may be set independently.

## Related Tables and Field Permissions

Field permissions are especially important when using Related Tables. In many cases, you may want to prevent certain groups from adding or removing records from a Related Table. To do so, remove a group's Edit permissions for the fields in the linked set in the source table. This prevents the Lookup icon from working to link new records, and it automatically removes the Unlink button from the Related Table's action bar.

### Example

Consider a Related Table of Support Cases contained in a Company record. To prevent a group from adding or removing records from the Related Table, remove the Edit permissions in the Support Cases table for all the fields included in the linked set to the Company table.

Another way to prevent users from unlinking records in a Related Table is to navigate to the Permissions tab of the Field wizard for the Related Table and deselect the option that allows users to unlink records.

## Teams

Teams represent sets of users who share a role or work on projects together. They are often used for sending notifications, assigning records to groups of users, and configuring localization, such as setting working hours and available languages. This makes teams different from groups, which determine the permissions each user has throughout the system. Teams help model your business processes and the company's entire hierarchy or supplier/customer relationships. The system comes preconfigured with several teams, and you can create any number of custom teams.

- Team membership controls:
  - working hours
  - group email distribution
  - team-specific views
  - which Look and Feel scheme is used
  - access to searches
- Teams can have subteams. For example, Sonya may be a member of the West Coast Sales Team, which is in turn a member of the US Sales Team.
- Teams can be hierarchical. For example, email notifications sent to a parent team will also go out to all subteam members.
- Teams serve different functions for Power User and End User groups:
  - Power User teams are used primarily to identify functional units to whom records and chat requests might be assigned, emailed, and so on.
  - End User teams are used primarily to enable distinct branding in the end user interface (EUI). Customers on different teams can see a completely different EUI color scheme, logo, etc.

## Team Membership

Team membership is independent of group membership, so several users from one team can belong to different groups and vice-versa. Users may belong to one or several teams, one of which should be set as their primary team.

The primary team determines the user's default interface and Look and Feel scheme, so users need a primary team before they can log in to the system. Team membership also controls how rules and notifications are applied to users. For example, you can schedule an announcement to be delivered to members of a selected team or have an email sent to them in response to some event.

Team membership can be adjusted from a user's record or from the Custom Fields tab of the Teams wizard.

## Teams Table

The Teams table makes it possible to mass edit teams, delete multiple teams at once, import and export teams, and perform some other general table functions.

Home

Chart Collections

Summary/combined reports

My Profile

Preferences

Tables

People and Companies

Contract Management System

All Communications

EUI Templates

Fiscal Years

Replacement Variables

Teams

Charts/Reports

Setup Teams

Teams

Status: 48 record(s) found, 2 page(s) [Click for details....](#)

Page:

[Page 1] [Next](#) [Last](#) Go to page  [Go](#)

New

Delete

Actions

<input type="checkbox"/>	Edit	ID	Team Name	Description
<input type="checkbox"/>		389	1st Level Support Team	First Level Support Team
<input type="checkbox"/>		391	2nd Level Support Team	Second Level Support Team
<input type="checkbox"/>		392	Admin Team	M-F 8-5pm, Used for system notifications about
<input type="checkbox"/>		403	Backup and Storage Team	Asset Backup and Storage Team
<input type="checkbox"/>		440	Change Advisory Board	Change Advisory Board
<input type="checkbox"/>		398	Configuration Management Team	Asset Team
<input type="checkbox"/>		446	Contract Creator Team	For end user contract requesters

The Teams table may be hidden by default. To unhide the table, go to **Setup > Tables**, click Team, and then click Unhide.

Teams may be deleted from the Teams table, but users need to be reassigned to another team if the deleted team is their primary team. When practical, rename rather than delete unused default teams to avoid being asked to unlink data dependent on the team.

## Working with Fields in the Teams Table

The Teams table does have certain limitations due to its special hard-coded fields for language, working hours, and so on. When viewing the Table wizard for the Teams table, you are only able to edit and add new fields to the General tab. The General tab then appears as the Custom Fields tab when viewing the Teams wizard, and it changes its name to General when clicked. Any fields you create are placed on this tab. Fields on the General/Custom Fields tab are also the only fields you're allowed to mass edit, import, and export.

Team

Save Cancel General

General information:

Backup Team Leader Team Pager

Include Team in Assigned Team Dropdowns

☒ Service Request
 ☒ Incident
 ☒ Problem

☒ Change Request
 ☒ Purchase Request
 ☒ Contracts/Documents

☒ Item Requested
 ☒ Asset
 ☒ Task

☒ Service
 ☐ Sales Tables
 ☐ External Support

Common Area \* General Hidden

General information:

Backup Team Leader Team Pager

Include Team in Assigned Team Dropdowns

Primary Team Members

Primary team members

Team Members

Team members

## Teams Wizard

The Teams wizard is used to create and edit teams. You can set a team's working hours, define the date and time format displayed to the team, schedule events for the team, and adjust a variety of other options.

To access the Teams wizard:

1. Go to **Setup > Access > Manage Teams**, or click the Teams table on the left pane.
2. Edit an existing team, or click New to create a new team.

## General Tab

The General tab allows you to define general information about the team, such as the team name, parent team, team leader, team description, and refresh rates. Note the following details when setting the parent team and the team leader:

- The parent team defines hierarchy and is used as a template for new teams. Teams are hierarchical in the sense that changes made to the parent team will also be applied to the subteams below it in the hierarchy. Moreover, an email sent to parent team will also be sent to its subteams, their subteams, and so on, down through the hierarchy. Never use the Company team or any other internal team as the parent team of an end user team, or an email sent to the Company team will inadvertently go to external end users.
- The Team Leader field can be used as a user linked field. It can be populated as a linked field along with the Team Name field into another table, and then the team leader for a team can be

emailed using rules. The team leader functions as the head of a team and does not need to be a member of the team.

Setup → Access → Manage Teams

General
Events
Working Hours
Languages
Formats
Custom Fields

Next
Finish
Cancel

The team's ID is an integer number used to uniquely identify the team in the current KnowledgeBase. The team ID has been pre-populated for you.	Team Identifier 449
Enter the name of the team. This name will identify the team in the current KnowledgeBase.	*Team Name Support Team
Select a parent team for your team. ?  A parent team can be used as a template for the new team. When changing a parent team's settings, you can have the changes applied to the teams down the hierarchy. ...More Help	Parent Team Company Team
Select a team leader The team leader does not need to be a member of the team.	Team Leader <input checked="" type="radio"/> None <input type="radio"/>
Enter a brief description of the team to help you remember its purpose.	*Team Description Support Team
This option specifies how often the chat messages are refreshed for users for whom this is their Primary team	Message Refresh Rate <input checked="" type="radio"/> Every 5 Minutes <input type="radio"/> Every 10 Seconds
This option specifies how often the table view is refreshed for users for whom this is their Primary team	Table View Refresh Rate <input checked="" type="radio"/> Never <input type="radio"/> Every 5 Minutes
This option specifies how often the table view checks locked records for users for whom this is their Primary team	Table View Locks Refresh Rate <input type="radio"/> Never <input checked="" type="radio"/> Every 1 Minutes
This option specifies how often to check notifications (broadcast, calendar etc) for users for whom this is their Primary team	Notifications Refresh Rate <input type="radio"/> Never <input checked="" type="radio"/> Every 1 Minutes

## Refresh Rates

Teams have several types of refresh rates:

- Message Refresh Rate:** determines how often Agiloft checks for new chat requests. By default, it is set to every 5 minutes, but you should select a greater frequency for teams that you plan to

receive chat messages. Use the Chat Refresh Rate global variable in the Admin Console to allow the selection of a maximum refresh frequency greater than every 10 seconds.

- **Table View Refresh Rate:** specifies how often the system refreshes the table view. It should typically be set to Never unless you are running a time-sensitive support operation. If users are working actively in the system, the table view will refresh every time they edit or create a record anyway. The downside of setting a value here is that the user will never be perceived as inactive, since their browser will keep sending a refresh message to the server. If this value is less than the Disconnect Due to Inactivity global variable, the user will never be automatically timed out by the system. This is especially important if they are using a floating license, because the license may not be freed up unless the user actively logs out.
- **Table View Locks Refresh Rate:** specifies how often the system checks for locked records.
- **Notifications Refresh Rate:** specifies how often the system checks for new notifications, such as from calendar events.

## Events Tab

Use the Events tab to schedule meetings, seminars, training sessions, and other events for whole teams.

Setup → Access → Manage Teams

General **Events** Working Hours Languages Formats Custom Fields

Back Next Finish Cancel

This wizard helps you schedule events, for instance meetings, announcements or vacations, for the team. ...!

Show for:  
☐ All  
☐ User  
☒ Team

Admin Team

◀◀ Apr 2019 ▶▶

Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

PST ▼ Today

Current KB Time:  
25 Apr 2019 14:07

Subscribe to Calendar

Manual Export to MS Outlook

Manual Import from Outlook

Day Week Month 3 Months Year

◀ April 25, 2019 ▶

12:00 am	
:30	
1:00 am	
:30	
2:00 am	
:30	
3:00 am	
:30	
4:00 am	
:30	
5:00 am	
:30	
6:00 am	
:30	



## Working Hours Tab

On the Working Hours tab, you define the hours that the team works. Working hours have some special uses: you can define rules to run only during a team's working hours, as well as run searches for elapsed time that exclude the nonworking hours of a specific team. You may wish to set the most common working hours for one or more teams and then use the Same hours as [TEAM NAME] option to copy those hours to subsequent teams.

The screenshot shows the 'Setup → Access → Manage Teams' interface. The 'Working Hours' tab is selected, with other tabs like 'General', 'Events', 'Languages', 'Formats', and 'Custom Fields' visible. Navigation buttons 'Back', 'Next', 'Finish', and 'Cancel' are present. The main content area has a text box on the left explaining the purpose of the tab. On the right, under the heading 'Working Hours', there are two radio button options: 'Same hours as' (selected) and 'As defined below'. The 'Same hours as' option is currently set to 'Company Team'. Below this, a table lists the days of the week with corresponding start and end time dropdowns. The times for Monday through Friday are set to 08:00 and 17:00, while Sunday and Saturday are currently blank.

Day	Start Time	End Time
Sun		
Mon	08:00	17:00
Tue	08:00	17:00
Wed	08:00	17:00
Thu	08:00	17:00
Fri	08:00	17:00
Sat		

You can also set working hours for teams that work into the next day. This is useful for companies with teams outside their home time zone, or for companies with 24/7 staffing, especially if shifts do not begin or end exactly at midnight. If a team's defined working hours end earlier in the day than they begin, Agiloft automatically interprets the end as being in the following day.

### Example

*Sunday: 22:00 to 11:00 will mean 22:00 Sunday to 11:00 Monday.*

*Monday: 22:00 to 11:00 will mean 22:00 Monday to 11:00 Tuesday.*

Working hours for teams outside the Agiloft server's time zone must be adjusted to take this disparity into account. For instance, if a team's working hours are 8 AM to 5 PM Central Time, and the Agiloft server is located two time zones away in the Pacific Time Zone, set this team's hours 2 hours earlier: 6 AM to 3 PM Pacific.

## Languages Tab

The Languages tab defines which languages are available to the team.

Setup → Access → Manage Teams

General Events Working Hours Languages Formats Custom Fields

Back Next Finish Cancel

Which languages are available to this team?

Select languages

- Urdu
- French
- Arabic
- 简体中文
- Brasileiro

## Formats Tab

The Formats tab defines date and time formats displayed to primary members of this team. The default for a new team is to show dates and times with this format: Jan 15 2014 17:00:00, with seconds displayed.

Setup → Access → Manage Teams

General Events Working Hours Languages Formats Custom Fields

Back Next Finish Cancel

Time Format:

☐ 12 Hours ☒ 24 Hours

Time format separator:

☐ Space ☒ Character :

☐ Display Seconds

Date Format:

Month: ☒ First 3 characters, e.g. *Oct*

☐ All characters, e.g. *October*

☐ 2 digits, e.g. *10*

Year: ☒ 4 digits, e.g. *2010*

☐ 2 digits, e.g. *10*

Order: ☒ Month Day Year

☐ Day Month Year

☐ Year Month Day

Date format separator:

☒ Space ☐ None ☐ Character

*It typically looks best to turn off the display of seconds. The default, built-in teams display with this format: Jan 15 2014 17:00, without seconds displayed.*

## Custom Fields Tab

The Custom Fields tab, which turns into another General tab when clicked, contains custom fields added to the Teams table. By default, this includes Backup Team Leader, Include Team in Assigned Team Dropdowns, Team Pager, and related tables showing people for whom the team is their primary team and those for whom it is one of their teams. You can use the related tables to create and add new users to teams. Note that adding a user to a new primary team removes them from their current primary team.

New fields in the Teams table that have been added to the layout will appear here. This is the only tab of the Teams wizard whose layout can be edited.

Team

Save

Cancel

General

<<

>>

Collapse All

Used as a filter for linked Team fields on the People table and sub-tables.

Status

Active

\*Allow emails to be sent to this team

Yes

General information:

Backup Team Leader

Team Pager

Include Team in Assigned Team Dropdowns

☐ Service Request

☐ Incident

☐ Problem

☐ Change Request

☐ Purchase Request

☐ Contracts/Documents

☐ Item Requested

☐ Asset

☐ Task

☐ Service

☐ Sales Tables

☐ External Support

Primary Team Members

Primary team members

Status: 1 record(s) found, 1 page(s). Click [here](#) to count records again..

New

Views

<input type="checkbox"/>	Edit	ID ↓	Full Name	Login	Primary Team	Teams
<input type="checkbox"/>		282	Christopher Caldwell	support	Support Team	Backup and Storage Team Server Team 1st Level Support Team...

## Include Team in Assigned Team Dropdowns

The Include Team in Assigned Team Dropdowns field is used in linked field filters on various tables to determine which teams are shown. When you create a new team, be sure to check the boxes for tables where you want the team to appear. If you don't check the box for a specific table, the team won't appear in the Assigned Team drop-down for that table.

© Copyright 2019 Agiloft Inc.

70

Include Team in Assigned Team Dropdowns	<input type="checkbox"/> Service Request	<input type="checkbox"/> Incident	<input type="checkbox"/> Problem
	<input type="checkbox"/> Change Request	<input type="checkbox"/> Purchase Request	<input checked="" type="checkbox"/> Contracts/Documents
	<input type="checkbox"/> Item Requested	<input type="checkbox"/> Asset	<input checked="" type="checkbox"/> Task
	<input type="checkbox"/> Service	<input type="checkbox"/> Sales Tables	<input type="checkbox"/> External Support

## User Roles

When you begin to customize the system, think about the different types of users and how their role affects the access they need.

Agiloft users belong to TEAMS and GROUPS. Groups set the level of access to tables, records, and fields. Team settings affect other parts of the interface such as the color scheme, available views, and the default home page. Teams also define working groups of users and can receive emails that go to every member of the team.

Users in multiple groups receive the superset of those groups' access settings. Users can also belong to multiple teams, but must always have a Primary Team to set important defaults. For easier maintenance, we recommend keeping the number of groups relatively small.

Teams and Groups can be added or removed in the Employee table by Roles. This allows users to add only Groups and Teams that they have permission to add. Roles also flag whether or not the Employee uses an assigned license.

## Roles

Roles allow staff users to change a user's group and team, but restricts the staff user to only Roles they have permission to grant. (e.g. Business Admin cannot grant someone Admin access). Roles are associated with teams and groups, so adding or removing a role will also add or remove the user's group and team. Roles also have a flag, Uses Assigned License, which the Employee table links to and refreshes on a rule after every save.

## Default Roles

Below are the default roles. You can delete or add more roles by going to the Roles Table.

Role Name	Associated Group	Associated Team	Uses Assigned License
Contract Manager	Contract Manager	Contract Management Team	Yes
Change Approver	Approver	Change Approver Team	Yes
Change Manager	Change Manager	Change Management Team	Yes

Role Name	Associated Group	Associated Team	Uses Assigned License
Backup and Storage Team Member	Base ServiceDesk	Backup and Storage Team	Yes
Change Advisory Board Member	Change Manager	Change Advisory Board	Yes
Clause Library Manager	Business Admin	Clause Library Team	Yes
Compliance Team approver	Approver	Compliance Team	Yes
Configuration Manager	Configuration Manager	Configuration Management Team	Yes
Contract Creator	Contract Creator	Contract Creator Team	No
Contract Owner	Contract Owner	Contract Owner Team	Yes
Custom Applications Service Desk Team	Base ServiceDesk	Custom Applications Team	Yes
Database Team	Base ServiceDesk	Database Team	Yes
Desktop Applications Team	Base ServiceDesk	Desktop Applications Team	Yes
Document Creator	Document Creator	Document Creator Team	No
Document Manager	Document Manager	Document Management Team	Yes
Document Reviewer	Document Reviewer	Document Reviewers Team	Yes
Facilities Team	Base ServiceDesk	Facilities Team	Yes
Finance approval team	Approver	Finance Team	Yes
HR Team	Base ServiceDesk	HR Team	Yes

Role Name	Associated Group	Associated Team	Uses Assigned License
Internal Customer for service desk	Internal Customer	Internal Customer Team	No
Internal Signer for contracts	Internal Customer	Internal Signer Team	No
IT Executive Board for change approvals	Approver	IT Executive Board	Yes
Legal Team Approver	Approver	Legal Team	Yes
Marketing Team	Marketing	Marketing Team	Yes
Network Operations Team	Base ServiceDesk	Network Operations Team	Yes
Project Manager	Project Manager	Project Manager Team	Yes
Purchasing Team	Procurement Group	Purchasing Team	Yes
QA Team	Base ServiceDesk	QA Team	Yes
Risk Approval team for contracts	Approver	Risk Team	Yes
Sales Person	Sales	Sales Team	Yes
Security Team	Base ServiceDesk	Security Team	Yes
Service Manager	Service Manager	Service Management Team	Yes
System Administrator	Base ServiceDesk	System Admin Team	Yes
Vendor Manager	Base ServiceDesk	Vendor Management Team	Yes
Business Admin	Business Admin	Admin Team	Yes

Role Name	Associated Group	Associated Team	Uses Assigned License
Base Service Desk group only	Base ServiceDesk		Yes
1st Level Support	Base ServiceDesk	1st Level Support Team	Yes
2nd Level Support	Base ServiceDesk	2nd Level Support Team	Yes
admin	admin	Admin Team	Yes



# Drill Down on Dashboards, Charts, and Reports

---



# Dashboards

The Dashboards screen is the landing page of the Power User Interface, and consists of a highly customizable view of widgets showing an overview of record progress and saved searches related to you. Widgets can display system data in the form of table views, charts, and numerical results; and filters can be used to refine the system data in real-time.

The Dashboard may be set as the default starting page for individual users at Left Pane Preferences > General Preferences > Login Preference.

## Navigation

To access the dashboards in a knowledgebase, click the Home link in the left pane.

To view other dashboards, click the dashboards drop-down list at the top of the page and select one of the items.

## Permissions

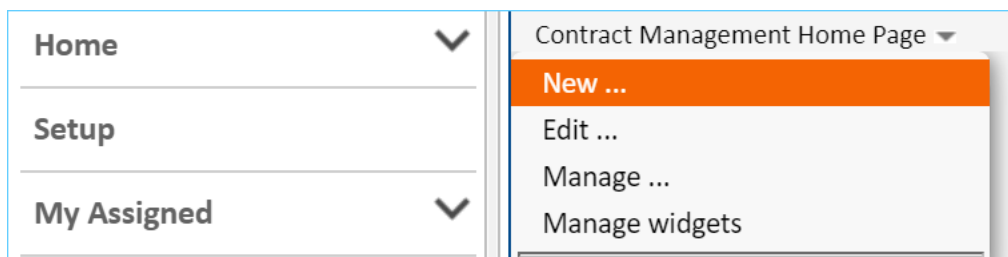
The dashboards drop-down allows you to create new dashboards and manage existing ones on the system. This is based on your user permissions via Setup > Access > Manage Groups > [select Group to edit] > General tab

You can select whether the group should be allowed to manage chart collections, dashboards, and widgets.

Dashboard views also enforce user view permissions. If a user does not have view permissions for a table or field that is used in a widget, it will not be included in their view of the widget on the dashboard. The widget will show a blank or null value.

## Create a Dashboard

In the home page dropdown, click New to create a new dashboard.



**New** - opens the Dashboard wizard to create a new dashboard.

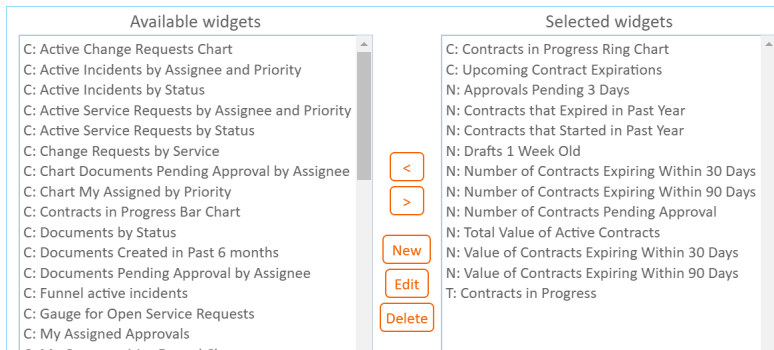
**Edit** - open the Dashboard wizard to edit the current dashboard.

**Manage** - opens a view of all dashboards to which you have access for editing or deleting.

**Manage widgets** - opens a view of all widgets to which you have access for editing or deleting.

## Add and Remove Widgets

1. In the **General** tab of the Dashboard wizard, add a name and description.
2. In the Widgets section, select the widgets to add to the Dashboard or create new widgets.
3. The list of available widgets will include charts from all tables to which you have access, as well as some useful predefined widget types.
4. Select a widget and click the right arrow to include it in the current dashboard.
5. Or, click New and walk through the setup to create a new widget.



## Widget Types

The dashboard is created from a collection of widgets, and the following kinds of widgets are supported:

- **Chart Widgets**
  - Display graphical charts on the Dashboard, and allow you to refine them in real-time and open the full chart or a table when clicked.
- **Hotlink Widgets**
  - The Hotlink widget type displays a vertical list of hotlinks to perform various functions in the system.
- **Table Widgets**
  - Displays a view of table records on the dashboard, allowing you to view and edit the records directly. The widget can use saved searches and apply any of your available table views. It can also optionally include the action bar, the search bar, and the status line.
- **Embedded Webpage Widgets**
  - Takes a URL and displays it in the space allocated to the widget. This can be useful if you wish to create a fully custom, branded home page for users in the Power User Interface.
- **Text Widgets**
  - Can either display plain text, or use the HTML editor from the Text field, including the WYSIWYG editor and the Source HTML.

- **Image Widgets**
  - The Image widget displays an image which you can upload in the Options tab of the wizard, with an optional caption beneath the image.
- **Custom Summary Report Widgets**
  - Highly flexible, allows you to insert variables, including Table Variables, and Formulas from multiple tables into the HTML editor but requires some HTML knowledge for complex designs.
- **Numerical Result Widgets**
  - Shows a single number calculated from a Formula that can use Table Variables. Because this widget is small sized, it can be helpful to group several into one area of the dashboard for an overview of some key system metrics.

## Dashboard Filters

Filters are created on the **Filters** tab of the Dashboard wizard. When a filter is applied to the dashboard, it removes all records that do not have the value in one of the fields specified in the filter, for all widgets that use the filter fields.

## Filter Types

- **Time based** - Operate on time-related fields in the tables used in the widgets on the layout.
  - *Example:* Using **Date Created** field to find records created this week, month, or year
- **User based** - Operate on Person-related fields in the tables used in the widgets on the layout. Uses a lookup to the People table, where you select a record that filters the widget fields based on the value contained in the source field.
  - *Example:* Using **Created By** field to find records created by a person selected from the People table
- **Team based** - Operate on Team-related fields in the tables used in the widgets on the layout. Uses a lookup to the Teams table and operates on the Team Name field.
  - *Example:* Using **Assigned Team** field to find records assigned to a specific team
- **Linked source field** - Uses linked fields from a source table used in one of the widgets on the dashboard layout. Uses a lookup on the linked source table, where you select a record containing the linked source field.
  - *Example:* Using **Project Name** from Projects table as the linked source field to find tasks associated with the project

## Dashboard Layout

The **Layout** tab of the Dashboard wizard allows you to define the layout of widgets on the screen and the dashboard background. The layout uses a responsive tile grid that adapts to the size of the screen.

**Available Widgets:** Click a widget from this list to add it to the layout, and drag it around to place it in relation to the other widgets. If you make a widget size smaller or larger in the layout editor, the contents will automatically attempt to center and resize.

**Vertical Sizing:** You can choose whether to fit all widgets onto one screen, or whether to allow scrolling.

**Autoarrange Widgets during Placement:** While moving widgets around the screen in the layout editor, enabling this option allows you to choose whether they automatically move to another area of the screen to fill the space.

## Dashboard Access

The **Apply** tab lets you select which teams should be able to view this dashboard, which teams should have this dashboard as their default page

## Reports

A report is a summary of table information presented in graphical, HTML, text, or Excel format. A saved instance of a chart or report can include multiple formats, and charts and reports can be combined into dashboards and chart collections to give users information at a glance and provide a launch point for daily tasks. Reports are useful for managing and monitoring work, tracking data and business trends, and representing information visually or succinctly.

You can locate Charts/Reports in the left pane under the table the report is from.

## Report permissions

Charts and reports apply user permissions, so they can be shared across the organization without granting unintentional access to sensitive information. They can be scheduled for automatic distribution to individual users and teams, and users can also run them interactively as needed according to their permissions.

Group permissions determine users' access to the data included in the chart or report, which means that the same report automatically shows a unique set of data for each user group, depending on their permissions. If a user doesn't have permission to view a field that is included in a report, the field is not shown when they view the report. When saved searches restrict a user from seeing a certain record or field, the record is not included in the report, or the field is shown as a field header with no values.

*Setup > Access > Manage Groups > Select Group name > Edit Table > Menu/Record/Field permission tabs*

The levels of permission control include:

- Allow creating/editing/deleting reports
- Allow publishing saved reports
- Allow using reports
- Access to records subject to group's record permissions
- Access to fields subject to group's field permissions

You can also distribute charts and reports by email, or write them to the hard drive for viewing in a web browser or other program.

## Creating a Chart/Report

*Charts Reports under the relevant table > click New*

Reports and charts are created using the same wizard so you can develop them in parallel, or quickly create charts from existing reports and vice versa. The wizard contains different tabs depending on the report format. Initially, only the General and Type tabs are visible, but after selecting the report output, the relevant wizard tabs appear.

### General tab

The General tab is where you name your chart or report, select which table or subtable it applies to, give it a description, and select the output format.

For graphical charts, you can select the Use JavaScript Charts option to allow users to select and hide data from the X Axis, and to automatically resize the chart based on screen dimensions.

### Type tab

There are three basic types of report: standard, elapsed time analysis/SLA, and trend analysis. If you aren't reporting on an Elapsed Time field or on historical trends over time, use the Standard type.

- **Standard:** Shows values from fields in records, with the potential to calculate averages or totals of field values and to sort and group down to an unlimited number of levels.
- **Elapsed Time Analysis/SLA:** Only appear if the current table contains a field with an Elapsed Time data type. Select the elapsed time field and the record field to run the report against, such as the average length of time a record spends in each status, or how long records are assigned to a team or person.
- **Trend Analysis:** Shows historical trends over time, such as the number of open support tickets that existed at the start of each month.

### Filter tab

You can filter the records included in reports by a date range for record creation or modification, and by more complex custom criteria using a saved search.

### Grouping/Summary tab

The Grouping/Summary tab controls how the information in the report is grouped, which view will be used to display it, and the summary information to include.

## Report Template tab

The General tab is where you name your chart or report, select which table or subtable it applies to, give

## Chart Type tab

For graphical charts, the Chart Type tab determines the visual output. These are the available chart types:

- Multiple Axes Chart
- Column Bar Chart\*
- Segmented Column Bar Chart\*
- Line Chart\*
- Segmented Line Chart\*
- Pie Chart\*
- Multiple Pies Chart
- Ring Chart\*
- Funnel Chart
- Multiple Gauges Chart
- Gauge Chart

*\*JavaScript Supported*

## X Axis & Y Xxis tabs

For graphical charts, these tabs control what is shown on the X (horizontal) and Y (vertical) axes of the report. The X axis determines the first level of grouping if there is an associated report, and the Y axis determines the second level.

## Drill Down tab

You can add drill-down features to JavaScript charts so that if the user clicks on a segment of the chart, a new chart opens with details about that segment.

## Formatting tab

For graphical charts, use this tab to configure the chart's appearance, including:

- Three dimensional effects
- Whether to display numeric values
- Legends
- Titles
- Generation dates
- Chart size

## Report Template tab

The Report Template tab contains an HTML editor, where you can customize HTML reports. Use the WYSIWYG editor or edit the source HTML to control the appearance, colors, and formatting of HTML reports.

## Customized Excel tab

Excel reports output a downloadable Excel spreadsheet that contains the defined data set and any templates, charts, tables, or other elements added to the customized Excel template. Excel reports use the Report wizard's filters and groupings/summaries, and once the report parameters have been defined in the rest of the wizard:

1. Click Create/Download New Excel File
2. Open the excel file containing the data pulled by the report and placed into the data sheet and begin to customize it. The columns in the data sheet are determined by the report's defined view, and the records included are determined by the report parameters.

For example, you might customize the template to include a PivotTable analyzing the data in a separate sheet tab from the data sheet tab, and a PivotChart to display the results at a glance.

The charts and PivotTables refer to this worksheet, so it's important to preserve the layout and format of the Data worksheet. For example, if someone updated the view for this report and removed one of the columns, several of the charts would break.

3. Once the excel has been customized, upload it back under this tab by clicking Choose File > select file > click Upload Customized Excel File.

## Schedule tab

You can set a schedule for when the system automatically runs and distributes the report. Reports and charts can be run at a specified frequency and time and distributed by email or written to a file folder on the server, if you have access to it.

## Preview tab

Use the Preview tab to test your configuration and make sure it looks and works the way you want it to. You can test it with the current user or specify another user to make sure the report shows appropriate data for users with different permissions.

## Combined excel reports

Excel reports can pull data from multiple tables. You can then use PivotTables and charts to analyze the data in any combination. To create an Excel report with data from multiple tables, configure an Excel report in each table with the data you want to use, and then create a combined report that uses those reports as data sources.

1. In the left pane, expand the Home section and click Summary/combined reports.

2. Click New.
3. Give your report a title and description. Make sure the description lists the tables you're using to provide source data.
4. Select the Excel output format.
5. Go to the Select Reports tab and click New.
6. In the Combined Report wizard, select the table and then select the Excel report you created. Click Finish.
7. Repeat step 6 for each table and Excel report you want to include. Each one you select will create a separate Data worksheet in the Excel file.
8. The reports are sent to the Excel template in the same order they're listed here. If you want to change the order, click and drag to move the reports.
9. Go to the Customize Excel tab.
10. Click Create/Download New Excel File and proceed to the Customize the Excel Template section.
11. When you finish your Excel template, configure the Schedule and Apply tabs as needed.

## Useful reports to have

### Agiloft Admin

Number of users in each group  
New users from this quarter  
Recent communications

### CLM

Number of contracts assigned per owner  
Approver response time  
Document source and print templates used  
Total number of contracts submitted per department  
Number of contract tasks and due dates  
Number of active contracts assigned to contract owners

### Service Desk

Number of completed tickets last quarter by agent  
Average time to close each month, run time by assignee

### Project Management

Number of active contracts assigned to project managers  
Active customer projects by target completion date  
Active projects by status





# Think Outside the Docx

---

# Print Template Syntax Reference

The following is a reference to assist you when creating print templates. The syntax required for inserting field variables, formulas, conditions, and other elements are listed below.

## Field Variables

Field variables can be inserted into print templates so that the field's value is displayed in the document when the print template runs.

The general form for field variables is \$FieldLabel. For example, in the People table the variable for the Login field is \$login.

You can construct chains of field variables to find values in other tables linked to the current record. These are in the form \$Field1.Field2 where Field2 references a field in another table linked through Field1. Field variable chains must be constructed manually by locating the appropriate field names in each table.

## Syntax

\$formula(\$field)

\$formula(\$field1.field2)

## Examples

\$formula(\$login)	Insert the Login value.
\$formula(\$account_rep.backup_person)	Insert the name of the Account Rep's backup person.
\$formula(\$related123123123)	Insert a related table from the current record.

Note that if you are putting dollar signs (\$) into a print template to format values in a number field, you can enter a backslash (\) to ensure that the dollar sign does not get eliminated by the system while it is processing the variables and formulas in the print template. For example, \\$25,000 will appear as \$25,000. You can configure the system so that it does not eliminate the backslash by changing the value of the [Keep slash prior escaped dollar sign](#) global variable to Yes.

## View or Download Field Variables

A list of field variables for the current table can be displayed when creating or editing a print template.

- Click Show Field Variables to open the Formula Help wizard with a list of the Fields, Global Variables, and Functions.
- Click Download Template to save an MS Word file which lists the field name and field label for each field in the current table.

## Formulas and Conditions

Formulas and conditions are used in print templates to calculate totals, insert different fields based on the values, and insert or delete text when a condition is met. A longer list of available formulas can be found in the Formula Help window, available from the [Mass Edit Wizard](#) and other system locations.

The following formulas and examples represent the most common uses and will help you create complex print templates.

## Basic Formulas

Standard mathematical operations can be used with numeric data types such as integer, floating point, and calculated result fields.

## Syntax

\$formula(\$equation)

## Examples

\$formula(\$contract_amount*\$discount_percentage)	Multiply the CONTRACT AMOUNT by the DISCOUNT PERCENTAGE
--	---

<code>\$formula(\$total_cost - \$refund_amount)</code>	Subtract the REFUND AMOUNT from the TOTAL COST
--	--

## Concatenate Strings

The concatenate strings command allows you to combine field values with text strings, field variables, or other formulas that use variables. For example, if you use `$formula(concat("x",$field_name,"z",...))`, the "x" and "z" placeholders are where you insert text strings, field variables, or formulas.

## Syntax

```
$formula(concat("x",$field,"z",...))
```

The variables may be text strings, field variables, or other formulas and variables. Text strings must be surrounded in double quotation marks (""), while variables and formulas do not.

If the first piece is a text string, you can use shorthand to combine variables and strings:

```
"x"+$variable+"z"+...
```

## Examples

Example formula	Output
<code>\$formula(concat("Your account representative is ",\$account_rep,"."))</code>	Your account representative is Hector Gomez.
<code>\$formula(concat(\$company_name," Support Contract"))</code>	Agiloft Support Contract.
<code>""+\$field1*\$field2+" Total"</code>	2,142 Total
<code>\$formula(concat(\$contract_start_date," to ",\$contract_end_date))</code>	09/01/19 to 08/31/2020

## Dateformat

`Dateformat()` is used to display a date/time field in a particular format, often for localization. `Dateformat()` takes two arguments: the desired pattern format, and the field variable. Date time patterns are indicated with a series of letters that represent elements such as month, day in month, day of week, year, hour, and minute.

## Syntax

`dateformat("output pattern","$field")`

Date/time Pattern	Example output
"MM/dd/yy"	05/31/16
"yyyy/dd/MM"	2016/22/09
"MMMMM"	July (name of month)
"d"	10 (day of month)

## Examples

The following results are expected when `$contract_start_date` evaluates to February 10, 2016 at 01:00:00.

Formula	Sample output
<code>\$formula(dateformat("d",\$contract_start_date))</code>	10. INSERTS WHICH DAY OF THE MONTH THE CONTRACT STARTS.
<code>\$formula(dateformat("MMMMM",\$contract_start_date))</code>	February. INSERTS THE FULL TEXT NAME OF THE MONTH.
<code>\$formula(dateformat("yyyy",\$contract_start_date))</code>	2016. INSERTS THE YEAR OF THE CONTRACT START DATE.
<code>\$formula(dateformat("yyyy",NOW()))</code>	INSERTS THE CURRENT YEAR WHEN THE PRINT TEMPLATE RUNS.

## Format Formula Outputs

`format()` is used to determine the output format of a formula based on the display characteristics of a chosen field. It is most commonly used to set the appropriate display for the output of a formula that includes Currency fields.

## Syntax

```
$formula(format("$table_name.field",$value))
```

In the syntax above, `$table_name.field` is the reference to the field whose formatting you want to use. For instance, `$contract.dollar_amount` could indicate using the \$1.00 standard US currency display format. The formatting field can be referenced from any table. The `$value` may be any valid formula such as a field or formula including multiple field variables.

## Examples

The following example formulas will help you create your own formulas.

Formula	Output
<code>\$formula(format("contract.display_field",\$contract_amount))</code>	Inserts the contract amount field formatted with the display characteristics of Display_Field.
<code>\$formula(format("quote.base_currency",\$total_amount*\$discount_percentage))</code>	Multiplies the Total Amount times the Discount Percentage and formats the output based on the Base Currency field's settings.
<code>\$formula(format("pricing_plan.monthly_price",((\$number_of_users*16) + (\$number_of_units*5))))</code>	Formats this formula using the Monthly Price field in the Pricing Plan table: multiply the number of users times 16 and add five times the number of units.

## Conditional Text, Paragraphs, or Clauses

A common requirement is whether to show words, multiple lines, or paragraphs of text based on the values in a record or a condition based on those values. Conditions can use a full range of logical operators such as `==` (equivalency), `!=` (not equal), `>`, `<`, `>=`, `<=`.

## Ternary Operator

Short conditional statements can be inserted with the ternary operator `"a ? b : c."` This works like an if-else statement: if the condition "a" evaluates true, then insert "b;" if "a" is false, insert "c." You must provide all parameters for the ternary operator to function, even if one parameter simply inserts a blank space.

## Syntax

`$formula($condition ? "True Output" : "False output")`

## Example

In this example the statement compares the variable `$n` to the value 1. If it is 1, "`$n is 1`" is shown. If it's not 1, "`$n is not 1`" is shown.

`($n==1) ? "$n is 1" : "$n is not 1"`

## Additional Examples

Formula	Explanation / Output
<code>\$formula(\$authorized_contract=="Yes" ? "'Authorized contractor' means a third party contracted to develop or assist with the development of an Authorized Application." : "")</code>	Inserts the sentence "'Authorized contractor' means a third party contracted to develop or assist with the development of an Authorized Application." if the AUTHORIZED CONTRACT field value is Yes; otherwise, a blank space is inserted.
<code>\$formula( isEmpty(\$company_id.fax) ? "" : "Fax: " + \$company_id.fax)</code>	If the company's linked Fax field is empty, insert nothing. If the Fax field is not empty, insert "Fax: 555-111-2134."
<code>\$formula(\$country=="USA"&amp;&amp;\$city=="Redwood" ? "The company is located in Redwood, USA" : "")</code>	Inserts the sentence "The company is located in Redwood, USA" if the COUNTRY is USA and the CITY is Redwood; otherwise, a blank space is inserted.

## If-Then-Else Conditions to Display Field Variables

The `$if` statement evaluates a condition to decide which field variable to display. It is similar to the ternary operator, but the true and false outputs may only be field variables or the `$merge` command with an attached file field.

## Syntax

```
$if(condition) ? $fieldTrue : $fieldFalse
```

The second (else) field variable may be left out to display blank if the condition is not met.

```
$if(condition) ? $fieldTrue
```

Or, the `$merge($filefield)` command may be used as an output:

```
$if(condition) ? $merge($filefield) : $fieldFalse
```

## Examples

<pre>\$if (\$discount_applied == "Yes") ? \$percentage_discount</pre>	If a discount is applied, show the Percentage Discount field.
<pre>\$if (\$contract_amount &gt; 100000) ? \$related123456 : \$customer.related345678</pre>	If the contract amount is greater than \$100,000, show the field <code>\$related123456</code> ; else, show the field <code>\$related45678</code> in the linked customer record.
<pre>\$if (\$include_appendix == "Yes") : \$merge(\$attached_file)</pre>	If an appendix should be included, merge the file held in the Attached File field.

## Conditional Paragraphs using \$startif

Use the `$startif()` command in a print template if some paragraphs of text should only be included under certain conditions. For example, suppose a certain liability clause only appears in a printed contract when the contract amount is over \$50,000. The conditional text comes after the `$startif($condition)` and is followed by `$endif` to close the statement.

## Syntax

```
$startif($condition)
```

```
Paragraph 1
```

```
Paragraph 2
```

```
$endif
```



## Examples

<pre>\$startif(\$contract_amount &gt; 50000) Contracts over \$50,000 require an additional insurance certificate. \$endif</pre>	<p>If the CONTRACT AMOUNT is over 50,000, the paragraph appears in the final document.</p> <p>If the CONTRACT AMOUNT is less than 50,000, the paragraph is deleted.</p>
<pre>\$startif(\$extended_warranty=="Yes") Your extended warranty is in effect until \$formula(\$warranty_end_date). \$endif</pre>	<p>If the EXTENDED WARRANTY field is Yes, insert the sentence listing the WARRANTY END DATE.</p>
<pre>\$startif((\$service_name ~= "Installation")&amp;&amp;(\$flag = "Yes")) Lorem ipsum dolor sit amet. \$endif</pre>	<p>If the Service Name field contains "Installation" and the Flag field is set to Yes, then show the paragraph.</p> <p>This example combines conditions using &amp;&amp; and extra parentheses surrounding the conditions.</p>
<pre>\$startif(find("Amended", concat("", \$contract_updates)) Lorem ipsum dolor sit amet. \$endif</pre>	<p>Checks if the value "Amended" is found in a multi-value choice field "Contract Updates". The concat function gets a string representation of the values in the multi-value field, and if it contains the value "Amended", returns the text below.</p> <p>To handle cases where none of the values are selected, you can add the following after the startif:</p> <pre>   isEmpty(\$contract_updates))</pre> <p>isEmpty returns "true" and outputs the text if none of the values are selected.</p>

## isEmpty

The isEmpty operator checks whether a field has a value or not, and is formatted as follows:

```
isEmpty($field)
```

This returns true when the \$field is null or empty and false if the field has a value.

## New line or insert line break

To insert a new line or a line break, use \n within a print template or field value formula. This can help start new paragraphs in certain cases.

# Merge Documents

The `$merge` command can be used to merge multiple files held in a single field, or files held in multiple fields into one final document.

Using the `$merge($fieldname)` command within a print template will merge all attached .docx/.html files from the field in the current record into the resulting print template's .docx file. Documents held in a versioned file field will be merged in the order in which they were uploaded or appended to the field.

## Syntax

To merge multiple documents held in a single file field with versioning or multiple files enabled:

```
$merge($fieldname)
```

To merge multiple documents held in separate file fields, use the format:

```
$merge($field1) $merge($field2) $merge($field3) ...
```

This makes it possible to compile a large PDF or Word document from several attachments.

The `$merge` command must be placed within the body of the print template. This function will not work if the `$merge` function is placed within the header or footer of the template, or within a table.

## Print Templates with eSignature Tags

Setting up Word-based print templates in Agiloft with the appropriate eSignature (DocuSign or Adobe Sign) tags is a key part of the integration, as this enables Agiloft to transfer signer information and required signature placement to the eSignature API. By automating document creation with print templates, the user who generates the contract document and the envelope does not have to manually place signature tags and related information for each signer in the eSignature platform.

## DocuSign

DocuSign Tags in Agiloft print templates use the following syntax:

```
$docusign:TagType:FieldName:RoleName
```

## Tag Types

For each DocuSign tag, you need to specify the tag type to determine how the field behaves. Some fields automatically populate with the signer's information, while others require the signer to enter information or make a choice.

The following tag types can be used in a DocuSign tag:

- Approve
- Checkbox
- Company

- DateSigned
- Decline
- FullName
- InitialHere
- SignHere
- Text
- Title

## Using Tag Types

In most cases, simply combine a tag type with the appropriate role name to create a DocuSign tag. For example:

- A DocuSign tag to indicate the signature for a signer whose role name is "InternalSigner1" would appear as \$docusign:SignHere::InternalSigner1.
- Add a DocuSign "Decline" button with the syntax \$docusign:Decline::Signer1.

Other commonly used tag types include DateSigned and InitialHere. A signature block that includes the DocuSign tags for the signer's signature, initials, and the date signed, as well as the Agiloft variables for the signer's full name and title, might look like this:

Signature: \$docusign:SignHere::InternalSigner1

Name: \$internal\_signer\_full\_name

Title: \$internal\_signer\_title

Date: \$docusign:DateSigned::InternalSigner1

Any DocuSign tags can be formatted with white text so that the tag text does not appear in the final document. In the Standard System KB, the example Hosted Service Level Agreement print template contains three sets of DocuSign Tags on the last page of the document.

Remember that if the signer fields are native to the Contract table, and are pulled into the print template directly, you must have all of them filled out before creating and attaching a document from the print template. In the example above, you would need to have the fields that correspond to the variables \$internal\_signer\_fullname and \$internal\_signer\_title filled in on the contract record before creating the document from the print template. In the standard Agiloft template KBs, these are the Internal Signer and Internal Signer Title fields under the Signature tab of the Contract record.

If these fields are not filled in before the contract document is created from the print template, the information will not show up in the resulting document. Similarly, the email address fields for the signers must be filled in before the DocuSign envelope is created, or the DocuSign Recipient records will not have the appropriate email addresses for the signers.

Signers			
Internal Signer	CAB Approver	Internal Signer Email	
Internal Signer Title	Change Approver Exec	Internal Signer ID	445

## Using Field Names

When there are multiple tags containing the same tag type and signer, they can cause overlapping signatures and other problems on the final document. For example, a signature field that appears multiple times in a document will create overlapping if their syntax is identical. To distinguish between tags in this situation, use the `FieldName` part of the tag syntax to give each tag a unique value.

To prevent the problem of duplicated tags, use different field names for each tag that will be used in the template.

### Example

```
$docusign:SignHere:unique1:Customer1
```

```
$docusign:SignHere:unique2:Customer1
```

## Adobe Sign

Adobe Sign Tags in print templates use the following syntax:

```
(<Required>)(<ReadOnly>):Field Name:ES Identifier:Role:Field Type:(:Rules)
```

## Tag Elements

- **Required:** Asterisks before tags act as optional flags that make the field required. If a field is marked as required, the signer must fill the field. If the field is not required, users have the option to leave the field blank during signing.
- **ReadOnly:** Exclamation marks before tags act as optional flags that make the field read-only. The data in read-only fields cannot be modified by users during the signing process.
- **Field Name:** Refers to fields within Agiloft tables that are then pulled into the Adobe Sign envelope.
- **ES Identifier:** The text string "\_es\_" is an Adobe Sign specific string that is a required element of Adobe Sign tags. If the string is not included, certain Adobe Sign features will not be activated.
- **Directive:** Variables that define the type of data allowed in the field. Some commonly used directives include:
  - **Role:** Defined role of the referenced user.
  - **Field Type:** Adobe Sign form field.
  - **Rules:** Other advanced rules.

A signature block that includes the Adobe Sign tags for the signer's signature, initials, and the date signed, as well as the Agiloft variables for the signer's full name and title, might look like this:

```
Signature: {{Sig_es_:signer1:signature}}
```

```
Name: {{N_es_:signer1:fullname}}
```

```
Title: {{Ttl1_es_:signer1:title}}
```

```
Date: {{Dte1_es_:date}}
```



# Work Wonders with Rules

---

# Rules

A rule is a statement of automated business logic that Agiloft executes.

For example: "Email the sales manager whenever a new lead arrives" or "Do not allow a bug report to be deleted unless it has been closed"

Agiloft provides a GUI for defining a wide variety of rules, and custom scripts may be used when the necessary functions are not supported by the GUI. For example a script might "verify the registration key provided by the customer against the list of valid keys kept in an external database".

There are three kinds of rules available:

## Event-specific Rules

- This type of rule is applied when a record is created, edited or deleted. In a typical KB, most rules are event-specific.

## Time-based Rules

- Time based rules are applied at specified time intervals. This type of rule can be used to automate actions that should occur when nothing is happening. For example, "Notify the support manager if an open case of severity Critical or greater from a customer with a valid support contract has been left open for more than 1 day"
- On a shared hosted Agiloft server, time-based rules have a minimum run interval of ten minutes. Decreasing this interval increases server load, so 10 minutes is the default on shared servers.
- Users with a dedicated server running their own instance of Agiloft may decrease this interval. The Admin console options allow the administrator to set the shortest interval used for time-based rules.
  - To change the minimum interval option, log in to the Admin console, and go to Knowledgebases. Select the knowledgebase, click Edit, and click the Options tab. Select a Frequency for "Shortest time interval allowed for time-based searches (in minutes)".

The screenshot shows the Agiloft Admin console interface. At the top, the Agiloft logo is on the left, and 'Language: English' and 'Logout' are on the right. Below the logo, there are tabs for 'Setup', 'Password', and 'Options'. Under 'Options', there are sub-tabs for 'KnowledgeBases', 'License', 'Debugging', 'Look&Feel', 'SMTP', 'Import', 'Backup', 'Variables', 'Cache', and 'Patches'. The 'KnowledgeBases' sub-tab is active, and within it, the 'Options' sub-tab is selected. The main content area shows a form with two fields: 'Shortest time interval allowed for time-based searches (in minutes)' and 'Frequency'. The 'Frequency' field is a dropdown menu currently set to '5'. Below the form, there are 'Apply' and 'Cancel' buttons. At the bottom right, a message states 'The FM infrastructure has already been added'.

## Aggregate Rules

- Aggregate rules are applied based on the number of table entries meeting a specified condition at specified time intervals. For example, "Notify the Sales Manager if the number of Leads expected to close next month is less than 20 or more than 40".
- In all cases additional conditions may be applied using a saved search. The Schedule tab allows you to control when the rule is active, for instance to limit it to the working hours for a particular team.

By default, only one time-based or aggregate rule is allowed to run in a KB at any one time. You can modify this by customizing the Max Timer Rules per KB global variable. Note that time-based and aggregate rules run sequentially when they are not allowed to run simultaneously.

## Creating, editing, and deleting rules

To create, edit, or delete rules, go to **Setup > Rules**.

It is not necessary to delete a Rule if you think you might need it in future.

### To deactivate a rule:

1. Navigate to **Setup > Rules**.
2. Select the rule in question and open it for editing.
3. Click Action.
4. Select all actions in Active Actions window.
5. Move them to Inactive Actions.

## Rule Configuration

Rules can be named to make them easier to group in the Rule wizard. For instance, it is useful to group the time-based rules by using the syntax Ticket TB - Name of rule for a time-based rule on the Ticket table.

Rules can easily get out of hand and grow in number, and if they are all done separately, it can be hard to control their order of execution. The best practice is to group automation into a smaller number of rules. In general, any table with lots of automation will typically have the following "parent" rules:

Rule 1:	all new record actions all new validations in an if-then action all new conditional actions in an if-then action any other separate actions on new records
Rule 2:	All edit validations. Usually customers who start setting up validation criteria end up having several of them and so we group them all into one rule. If there is only a single edit validation, then it can go in one of the other rules below.

Rule 3:	Edit actions when power users update.
Rule 4:	Edit actions when end user updates.
Rule 5:	Parent-child handling. If complex, this may need its own separate rule with all actions in an if-then; if not, it might go into one of the above rules.
Rule 6 - infinity:	<p>Rules that must be defined separately from other ones in order to be triggered by changes made by the others, or rules that should not include API when the rule they might otherwise go in includes it, or which must include API when a possible containing rule omits it.</p> <p>For instance, a rule based on customer updates should never include API changes because since other rules run currently in name of last updater, if any other rule fires, the system will think the customer has updated again and will re-trigger that rule. That is why it is usually suggested to separate customer update rule actions from staff edit actions. There may be some staff edit actions that must be separate too because of a need to include or exclude API triggers.</p>
Rule 7 - infinity:	Time based rules - It is usually preferable to combine them whenever possible into if-then conditions, but sometimes it is cleaner to make a separate rule for each one, unless there are 10s or hundreds of them, as long as they are named so you can sort them and see them all together, i.e. named for "table name TB - description".

The basic goal is to have as few rules as possible per table. Otherwise, a complex system will have hundreds of rules which can make sorting laborious, and they become much more difficult to troubleshoot, prioritize, and document.

## Priority Order

- When you create a rule, you assign it a priority number.
- When you create multiple actions for a rule, you move them up or down to define their order of execution.
- Rules execute in priority order relative to other rules with the rule with the lowest number executing first. For example, a rule with a priority of 1 executes before a rule with a priority of 2.

A given rule can execute multiple actions and these actions execute in the order that they are listed.

### *Example*

*Create a rule for new tickets, then add an action that changes the Assigned to field based on values in the ticket, then add a second action to the same rule that emails the assignee and a third action that emails the customer the fields of the ticket, including the assignee field.*



Where possible, try to combine rules for a given table into a few, or even a single if-then-else rule. A few if-then-else rules are both more efficient and easier to manage than a swarm of independent rules.

## Order of Execution in Saving

Many checks are run on a record when it is saved. There may be field validation, rules, workflow actions and guards, and custom scripts run as a result of saving. It is important to understand the order that these actions execute. In order, they are:

1. Form validation is run on any validated fields.
2. Dynamic default values are updated, the Updated By and Date Updated fields are populated.
3. Rules based on record modification are executed in priority order.
4. Workflow guards are executed. If they fail, the record is returned to its former pre-saved condition.
5. Workflow actions are executed.
6. Any scripts associated with rule or workflow actions are executed with the associated rule or workflow, and if their validation fails, the record is returned to its pre-saved condition.

Please note that, by default, you will see a warning before running a rule that matches more than 100,000 records. If you choose to run such a rule, it will only affect the first 100,000 matching records. You can change this behavior at both the Admin or KB level by setting new values for the following global variables:


- Max Rule Records Limit
- Max Rule Records Warning

## Rules that Trigger other Rules

Changes made by one Rule can trigger the execution of other rules.

Edits made by a Rule are defined as API actions, and do not change the Updated By field.

Prevent a Rule from being triggered by the actions of other Rules by unchecking the API box which is

found at **Setup > Rules > edit rule > Condition tab** .

### Example

*In a normal business process, Agiloft sends email to the assignee of a customer ticket every time the ticket record is updated. This rule would send an email every time the customer updates their record, but also when the record is changed by any other means. Say, for instance, that the ticket aged past a certain SLA goal, so the system updated it to increase the severity. On this edit, the assignee would get an email. If the increase in severity then triggered several other rules, the assignee would get an email for every single one of these rule-triggered edits. That can add up to a lot of email to dig through!*

*To prevent the tech being overwhelmed with autogenerated emails, the admin edits the Rule, unchecking API so that an email notification is not sent after a rule-triggered (API) change.*

*Alternately, the admin may create a single, more complicated if-then-else rule to replace the several simpler rules.*

## Applying Rules

The Condition tab of the Rules wizard contains the options to define when the rule should be applied. For example, if you select Edited and Web, the rule will only be applied if an Approval was edited using the Web interface.

The possible triggers for a rule include:

- **Email** - all updates from inbound email.
- **Web** - user interactions with records through the web portal; REST and Web Service calls, because the Web Service call creates a user session and performs edits on behalf of a named user.
- **API** triggers include:
  - Record edits by a rule or workflow action
  - Automatic calculation field updates
  - Synchronization actions, including automatically triggered updates such as creation of Twitter responses
  - Record creation in cases where the user does not interact with the record form, for example during data imports from files and silent conversion actions.

Rules based on the Activity Logs table

The Activity Log table can be used as the trigger for rules. For example, if you wanted to run a rule that updates a user record when they log in, or that notifies you if someone changes a workflow, you can now create a rule to do that when the relevant record is created in the Activity Log.

## Actions

Actions are used in rules and action buttons to automate repeated actions in the system, like sending notifications or updating related fields and records. When you create an action for a rule or a workflow

transition, it is added to the library of actions for that table and is then available for reuse by other rules and workflow transitions.

The Actions wizard can be accessed from any of the following paths:

- In the Actions tab of the Table wizard.
- When creating a rule, in the Action tab of the Rule wizard.
- In the Fields tab of the Table wizard, when creating a new Action Button field type, select **Execute Actions > Add Action**.
- From **Setup > Rules > new or edit > Action tab**.

Each action type is displayed as a button on the screen. Click **Create <action type>** to open its specific wizard. The wizard details will vary by action type.

## If-Then-Else Actions

If-Then-Else actions are used to execute a variety of actions based on different criteria. They function as a container action that holds other actions. You define criteria that control when one or multiple actions run, and the If-Then-Else action groups those actions into a single action. If-Then-Else actions are the most commonly used action.

**If-Then-Else actions are used for:**

- Reducing the number of rules in the system by grouping similar actions together in a single rule where they can all be seen and maintained.
- Providing a quick, GUI based programming tool.
- Allowing admin users to define multiple conditions, each of which can run different actions of any type when the condition for the "if" clause is true.
- Embedding multiple "ifs" within each other and separating or combining them with "and" and "or" operators to create conditional statements.

### Example

Consider a rule that runs on a request table such as Support Cases when a new record is created. It could contain the following If-Then-Else action:

*If the Assigned Person field does not equal null*

*Then email the Assignee about the new case.*

*Else, if the Assigned Team field does not equal null,*

*Then email the Assigned Team the new case.*

The screenshot shows a dialog box with two tabs: 'General' and 'Details'. The 'Details' tab is active. At the top, there are three buttons: 'Back', 'Finish', and 'Cancel'. Below these is a text block explaining the dialog's purpose: 'This dialog helps you create complex If-Then-Else actions. Select the line where you want to edit or insert a statement. The buttons are context-sensitive and show what actions are allowed based on the cursor position. The Empty block . . . cannot be removed or edited, it is used as a place-holder for selection. Please see the tutorials at our website for detailed directions.' Below this text is a row of buttons: 'Add Action', 'Add If', 'Add Elseif', 'Add Else', 'Add Comments', 'Move Up', 'Move Down', 'Edit', and 'Delete'. At the bottom, there is a code editor showing the following code:

```
if (Assigned Person!=NULL) {  
    email assignee new case  
} else if (Assigned Team!=NULL) {  
    Email Assigned team new case  
}
```

*In the above example, an Email action will be executed only if the Assigned Person or Assigned Team fields are not empty.*

## Creating an If-Then-Else Action

You can find the Actions wizard with several different navigation paths, but the easiest is to start by selecting the setup menu for the table where you wish to create your action.

1. Select the Actions tab.
2. Click Create If-Then-Else Action.
3. On the General tab, name your action and give it a description.
  - a. Note: Once your action is saved the system automatically adds an I: before your given title to distinguish the action as If-Then-Else.
4. Navigate to the Details tab and use the Add buttons to create your unique If-Then-Else criteria.
  - a. You can modify existing actions or create a new one.
5. Click Finish.

### Best Practice Tip

*When creating If-Then-Else actions, provide a short description of the action's purpose in the Description field. For example, you might say, "Handles team assignments based on Category." This can be helpful if you inherit someone's system, are working with someone else, or want to quickly know the action's purpose without examining the conditions.*

## Useful Tips

Before creating an If-Then-Else action, it is useful to refer to the following list:

- You cannot move an “if” up to the top level of your statement once it has been created. However, you can move actions up and down. It is best to think through your logic carefully before building it.
- Placing “ifs” at the correct level can sometimes be a challenge. Until you are familiar with the editor, we suggest checking the screen after each Add If to make sure it is in the correct order.
- Searches within an If-Then-Else action will not find changes made by earlier actions within the same If-Then-Else action.
  - For example, suppose the first clause action in an If-Then-Else container sets the Assigned Person to John Smith, and a later “if” condition says “if the Assigned Person equals John Smith, then email John Smith that he has been assigned.” The second search will not find the record where John Smith was just assigned. Changes made by an If-Then action are saved in memory and executed after all the ifs have been evaluated.

## Email Actions

Email actions are used to send an email to a user, email address, or team of users based on a template. The email template is composed with the Email wizard, just like the one used to send an email from a record or table view. For more information about creating email templates, see [Sending Emails](#).

### *Useful to Know*

*Emails within email actions can be addressed to multiple recipients who are selected based on the User, Team, or Email fields in the record. They can also be addressed to specific, hard-coded email addresses, but this method is not recommended.*

*Emails can contain specified fields from the record, attachments, and multiple hyperlinks, which allow the recipient to log in and either edit or view the record.*

*Emails can be sent in either HTML or plain text format.*

*When an Email action is run on several records, the results can be sent individually or collated into a single email containing all of the relevant records.*

*Emails sent to a team are sent to the individual members of that team, using the permissions of each individual. However, if the "Allow emails to be sent to this team" field in the team's record is set to No, the email is not sent at all.*

## Create an Email Action

You can find the Actions wizard with several different navigation paths, but the easiest is to select Setup [TABLE NAME] from the left pane for the table where you wish to create your action.

1. Select the Actions tab in the Table wizard.
2. Click Create Email/SMS Action.

3. On the General tab, name your action and give it a description. Once your action is saved the system automatically adds an E: before your given title to distinguish the action as Email or SMS.
4. Choose whether you want to send the message through email or SMS message.
5. Navigate to the Message tab and either select an existing template or create a new template.
6. Click Finish.

### Example

When creating an Email action, you can create a message using an existing template by going to **Insert > Populate from template** and choosing the template. For example, choosing the "Email Contract Owner about renewal date" action results in a message that already contains predefined text and variables, as well as a link to update the contract record. When the action runs, the variables are populated with their appropriate values.

The screenshot displays the 'Edit Email Template' window. At the top, there are buttons for 'Next', 'Finish', and 'Cancel', along with tabs for 'Insert', 'Options', and 'Heading'. The main form area includes fields for 'From' (set to 'Default outbound demo\_itil\_test@agiloft.com'), 'To' (set to '>Internal Contract Owner'), and 'Subject' (set to 'Contract for \$company\_name is coming up for renewal'). The 'Content' field contains a message: '[The Contract for \$formula(\$contract\_type) described in more detail below has a renewal type of \$renewal\_type so it will either expire or auto-renew on \$formula(\$contract\_end\_date). Please take the appropriate action. Click [here](#) to update the record directly.]'. A 'Formula Help' button is located next to the subject field. The 'Attachments' section shows 'No Files Attached'. At the bottom, there are buttons for 'Next', 'Finish', and 'Cancel', and a 'Fields Included' section listing variables: 'Renewal Type, ID, Status, Contract Description, Contract End Date, Company Name, Contract Start Date'.

## Update Fields Actions

Update Fields actions are used to modify one or more fields in the selected record or a linked record in another table. You can modify field values in several different ways, such as with standard text, a saved search, a regular expression, or a formula. The available options depend on the type of field being modified.

### *Example*

*Update Fields actions can be used in many different ways:*

*Assign records to the appropriate team.*

*Increment or decrement number fields.*

*Add or subtract time from date/time fields.*

*Change the status of a record when specific conditions are met.*

*Manage the process flow between related tables, such that if an update is made in one table, fields can then be updated in the related tables.*

*Populate a linked field based on search criteria.*

*Update records in a related table when some condition is met.*

## Regular Expressions in Update Field Actions

You have the option to use a regular expression (regex) in the Formula field of Update Field actions. This type of replacement also accepts special characters. For example, regex `'/;+/\n/` replaces `;"` with a line break. If you're unfamiliar with regex, use one of the other options for updating field values or seek out one of the many online resources for using regex.

## Create an Update Fields Action

You can find the Actions wizard with several different navigation paths, but you'll commonly just click Setup [TABLE NAME] on the left pane for the table where you wish to create your action.

1. Click the Actions tab in the Table wizard.
2. Click Create Update Fields Action.
3. On the General tab, name your action and give it a description. Once your action is saved, the system automatically adds a U: before your given title to distinguish the action as an Update Fields action.
4. Navigate to the Fields tab and select the fields you want to update. By default, only fields on the current table are available fields. To update fields from a linked set, select a value from the linked field drop-down; this updates both the local fields listed and the linked fields in the drop-down. You can select a field in the linked table, or select another linked field to access another linked table in the chain. If you select a one-to-many link from the Referring Tables section of the drop-down list, all linked records from that table are updated, regardless of which referring table you select.
5. On the Values tab, select a new value for the field.
6. On the Errors tab, choose how you want the system to respond if there is a system error while the field is being updated.
7. Click Finish.

## Validate Actions

A Validate action is used to generate an error message and either reject or allow the user to accept the change if the record meets the search conditions specified on the Condition tab. The admin who creates the action provides the action name, a description of the purpose of the validation action, then writes a custom error message that users will see when the validation is triggered. The error message should explain to the user what to do in order to save the record.

## Create a Validate Action

You can find the Actions wizard with several different navigation paths, but the easiest is to start by selecting the setup menu for the table where you wish to create your action.

1. Select the Actions tab.
2. Click Create Validate Action.
3. Name your action and give it a description and an error message.
4. Select whether the action should prevent record saving, or allow the user to save by confirming the dialog. Once your action is saved the system automatically adds a V: before your given title to distinguish the action as Validate.

## Data Conversion Actions

Data Conversion actions are used to automate the conversion of a record into another record using predefined conversion rules. They are frequently used to take data from one record and map that data to create one or more new records, or update one or more existing records. The records can be in the same table that contains the action or a different table. Data Conversion actions can run in several ways, each permitting different levels of visibility and/or control to the user who initiates the action. For more information, see Data Conversion.

### *Examples*

*Automatically map the Time Spent and Work Description fields from a change request record into the Time Entries table when the change request record is saved.*

*Convert task templates selected in a project into new task records that are linked to the project.*

*In a contract, allow users to click an action button to create a renewal contract record and bring up the new contract form to fill out additional fields.*

## Create a Conversion Action

You can find the Actions wizard with several different navigation paths, but the easiest is to click Setup [TABLE NAME] on the left pane for the table where you wish to create your action.

1. Select the Actions tab in the Table wizard.
2. Click Create Data Conversion Action.



3. On the General tab, name your action and give it a description.

*Once your action is saved, the system automatically adds a C: before your given title to distinguish the action as Data Conversion.*

1. On the Conversion tab, choose an existing conversion mapping or create a new one. The Conversion tab shows all the conversion mappings that have been configured from that table to any other table. Clicking New opens the Data Conversion wizard, where you can create new mappings for your action.
2. On the Options tab, choose how the conversion should run:
  - **Silently, with no confirmation:** The records are created without any user intervention or confirmation. This option is used with action buttons and background conversion performed by rules.
  - **Interactively, showing the conversion dialog and results:** The Conversion Dialog opens before the conversion runs. This allows users to confirm the conversion and edit any fields that are configured for editing during conversion. After the conversion runs, a confirmation screen appears that lists the hyperlinked IDs of the newly created record(s). This option is used primarily when converting to multiple new records to allow edits of certain fields for each record. This is similar to clicking the Convert button on the action bar.
  - **Interactively, showing the new record page:** The conversion runs and opens the new record(s) for users to complete, edit, and save. This option is usually used when the conversion is triggered by an action button that brings up the new record(s) for completion.

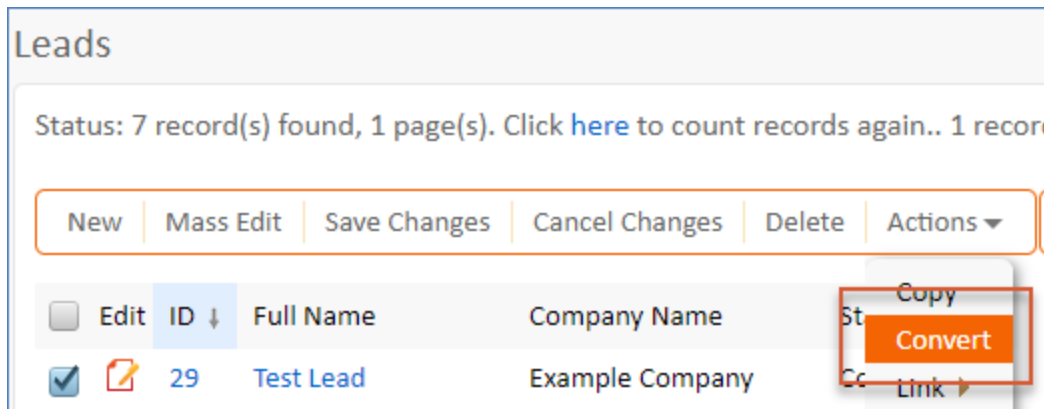
*When running a conversion with this option selected, a user's permissions determine the fields they can see and edit.*

- **Automatically, but showing the confirmation screen:** The conversion runs and a confirmation screen appears that lists the hyperlinked IDs of the newly created record(s). This option is used mainly for action buttons that convert in the background but report the results.
3. Click Finish.

## Using the Convert Button

In addition to running Data Conversion actions with an action button or in the background via a rule, they can also be run by a user selecting a record in the table view and clicking Convert on the action bar. This brings up a Conversion Dialog for completing the conversion, making it similar to using the "Interactively, showing the conversion dialog and results" option if running the conversion from an action button. The visibility of the Convert button is controlled by a group permission.

By default, the Convert button runs all conversion mappings defined for the table. However, you can also give the user the option to choose which conversion mappings run.



## Import and Export Actions

Import actions are used for automating imports and updates from another system, and export actions are used to export information from your Agiloft system into a new file. Both Import and Export actions are typically initiated by time-based rules that run after hours, which helps mitigate the impact on the system of a large-sized import or export.

### Example

*If real-time sync is not needed, you can automatically import changed data from another system into Agiloft by using an Import action to update or create the relevant records. Conversely, you can schedule a nightly Export action to export data from Agiloft for importing into another system.*

## Import Actions

When you create an import action, you configure options with a wizard that's very similar to the normal import wizard used for manually importing data. If your goal is to synchronize data between systems, which is often the case, place the import action in a time-based rule in the table into which you want to import data.

To create an Import action:

1. Go to Setup [Table] and go to the Actions tab.
2. Click Create Import Action.
3. On the General tab, name your action and give it a description. Once your action is saved, the system automatically adds a T: before your given title to distinguish the action as Import.
4. Navigate through the remaining tabs of the Import wizard. Note that you cannot import data from a local hard drive with an Import action.
5. Click Finish to save the Import action.

## Export Actions

The wizard for Export actions is identical to the Export wizard for manual, one-time exports.

To create an Export action:

1. Go to Setup [Table] and go to the Actions tab.
2. Click Create Import Action.
3. On the General tab, name your action and give it a description. Once your action is saved, the system automatically adds an X: before your given title to distinguish the action as Export.
4. Navigate through the remaining tabs of the Export wizard. Note that you cannot export data to a local hard drive with an Export action.
5. Click Finish to save the Export action.

## Delete Actions

Delete actions are used to automatically delete table records or history entries. They can be run with a time-based or event-specific rule, and they are particularly useful for deleting excess history records created by other time-based rules or older, unnecessary records.

### Examples

*Once a month, you want to delete unpublished records older than three years.*

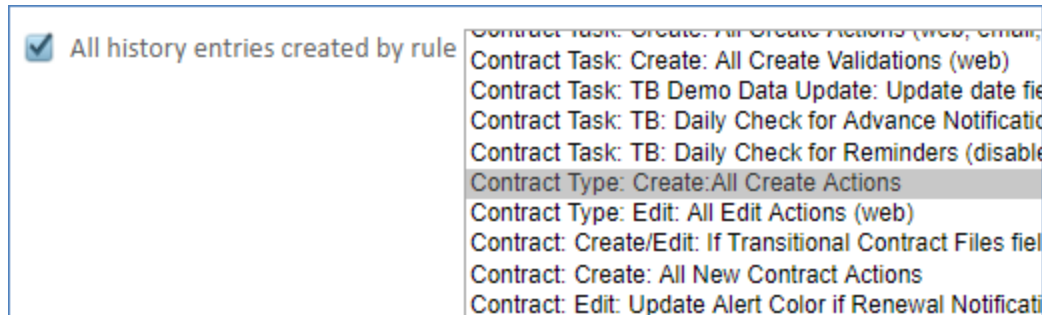
*You want to automatically delete the history records generated by a particular time-based rule that are older than one week, such as from a rule that updates a count every hour.*

*You want to automatically delete records that have been marked as spam by power users, but you don't want to give these users permission to delete the records themselves. You could, however, allow users to prevent records from being deleted by allowing them to change the spam designation before the rule runs that contains the Delete action.*

## Create a Delete Action

You can find the Actions wizard with several different navigation paths, but the easiest is click Setup [TABLE NAME] on the left pane for the table where you wish to create your action.

1. Select the Actions tab in the Table wizard.
2. Click Create Delete Action.
3. Name your action and give it a description. Once your action is saved, the system automatically adds a D: before your given title to distinguish the action as a Delete action.
4. Choose to either delete records or delete history entries. If you delete history entries, you have several additional options:
  - **All but the most recent [VALUE] history entries per record:** Prevents newer history entries from being deleted. With this option selected, you can also select the "Do not delete the first history entry for record creation" option. This prevents the first history entry from being deleted, which is created when the record is created.
  - **All history entries created by rule:** Deletes history entries that are created by the selected rule. To the right of this option is a list of all rules in each table of the knowledgebase. Hold down the Ctrl key to select multiple rules.



When deleting history entries created by a rule, you can refine which entries are deleted by checking the "Limit deletion to those that are more than [VALUE] days old" option. With this option checked, only delete entries that are older than the desired number of days are deleted. This is useful to limit database size in cases where tables have a lot of rules and history tables can become overly large. Large history tables affect backup times and storage size, and it is best to keep only history records that are actually useful.

*Time-based rules can cause many history entries to add up over time, so it's often useful to select this option when using time-based rules.*

- **All history entries more than [VALUE] [CHOICE] old:** Deletes history entries that are older than a desired number of days, weeks, months, or years. With this option selected, you can also select an additional option to prevent the first history entry from being deleted when a record is created.
- **Delete empty history entries...were modified:** Deletes history entries where no tracked fields were modified.

5. Click Finish.

Once created, the action can be selected in rules, action buttons, and workflows.

## Browser Pop-up Actions

Browser Pop-up actions trigger a pop-up window displaying a custom message in a logged in user's browser. The pop-up appears to the users specified in the action only if they are currently logged in to the system. Pop-ups can be run from a time-based or event-specific rule or an action button.

They are typically used to notify a user who saved a record that a follow-up action has been completed or to notify staff members when a new record of a specific type is received, such as a high priority support ticket.

The popup includes a custom message and button to view the record that triggered the popup, as well as a button to close the popup.

## Create a Pop-up Action

From the Actions tab within the Table wizard or while adding an action to a rule:

1. Click Create Browser Pop-up Action.
2. Name your action and give it a description. Once your action is saved the system automatically adds a B: before your given title to distinguish the action as Browser Pop-up.

3. Choose recipients to receive the pop-up message. Pop-ups generated by the action can be displayed to a set of hard-coded users or to users held in a user linked field in the record, such as Assigned Person, the team in the Assigned Team field, or the person who initiated the action that resulted in the pop-up.
4. Create a message that will appear in the pop-up window. This message may contain HTML formatting and may also include any field variables from the record that triggered the popup.
5. Click Finish.

#### *Best Practice Tip*

*The default size of the popup is rather small. To control the size of the popup window, add the following text at the bottom of your custom message within the html editor, while in source HTML mode:*

```
<script>window.resizeTo(600,450)</script>
```

*This will provide a window 600 x 450 pixels in size; you can use custom numbers to achieve your preferred size.*

## Script Actions

Scripting is an advanced option that allows you to execute a script when an action is triggered. Scripts can be executed under any set of conditions.

The action defines the file name and location of a script; the condition under which the script is run is defined by the rule's search criteria or when an action button is pressed. Script actions are not often needed because most actions can be accomplished with the default action types.

#### *Example*

*A Script action runs when a new Training Signup record is created, which signs a new person up for training. The script copies two master KBs to create the Training KBs for the trainee, populates their passwords and login information, and then updates the KB record and Training Signup record with the information needed for a trainee to login.*

## Create a Script Action

You can find the Actions wizard with several different navigation paths, but the easiest is to select Setup [TABLE NAME] on the left pane for the table where you wish to create your action.

1. Select the Actions tab in the Table wizard.
2. Click Create Script Action.
3. Name your action and give it a description. The name of the Script action must match the name of the script file. Once your action is saved the system automatically adds an S: before your given title to distinguish the action as Script.
4. Click Finish.
5. Locate and upload your script to one of the following server file installation directories, based on your server type:
  - For Linux servers: /usr/local/agiloft/data/[KB]/scripts

- For Windows servers: /agiloft/scripts

*For security reasons, it is not possible to upload the file to the server from the browser. You must have direct access to the server.*

## File Extensions

The following extensions designate how the script executes:

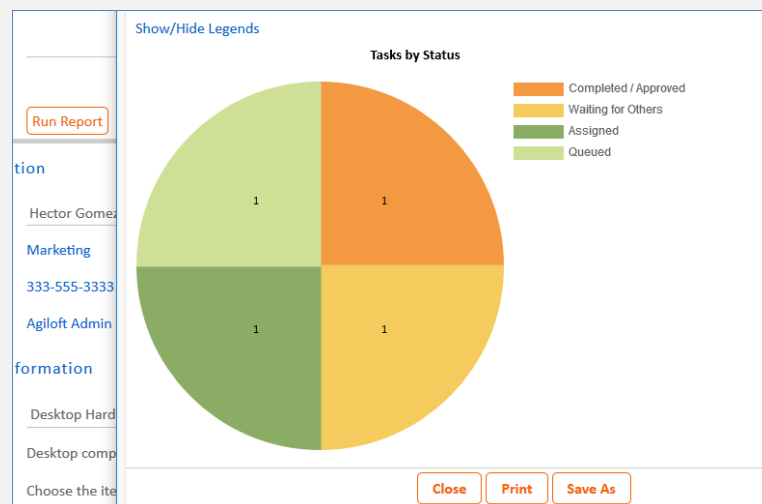
- .pl – Executed by the Perl interpreter
- .jar or .class – Dynamically loaded and run as a java class
- .bsh – Executed as BeanShell scripts
- .py – Executed as Python scripts
- .exe or any unrecognized extensions – Treated as executables

## Report Actions

Report Actions run reports directly from records, using the information from the record as a run-time input.

### Example

*For example, you could add a Report Action to an action button in the Service Requests table that runs a report on the service request's associated tasks. When a user clicks Run Report, the service request is passed in automatically and the user doesn't see the run-time input at all.*



## Create a Report Action

You can find the Actions wizard with several different navigation paths, but the easiest is to start by selecting the setup menu for the table where you wish to create your action.

1. Select the Actions tab.
2. Click Create Report Action.
3. Give your action a name and description.
4. Select the type of report, either table-specific or combined.

5. Either select an existing report from the drop-down list or create a new one. You can also use the Edit button to make changes to an existing report once you select it.

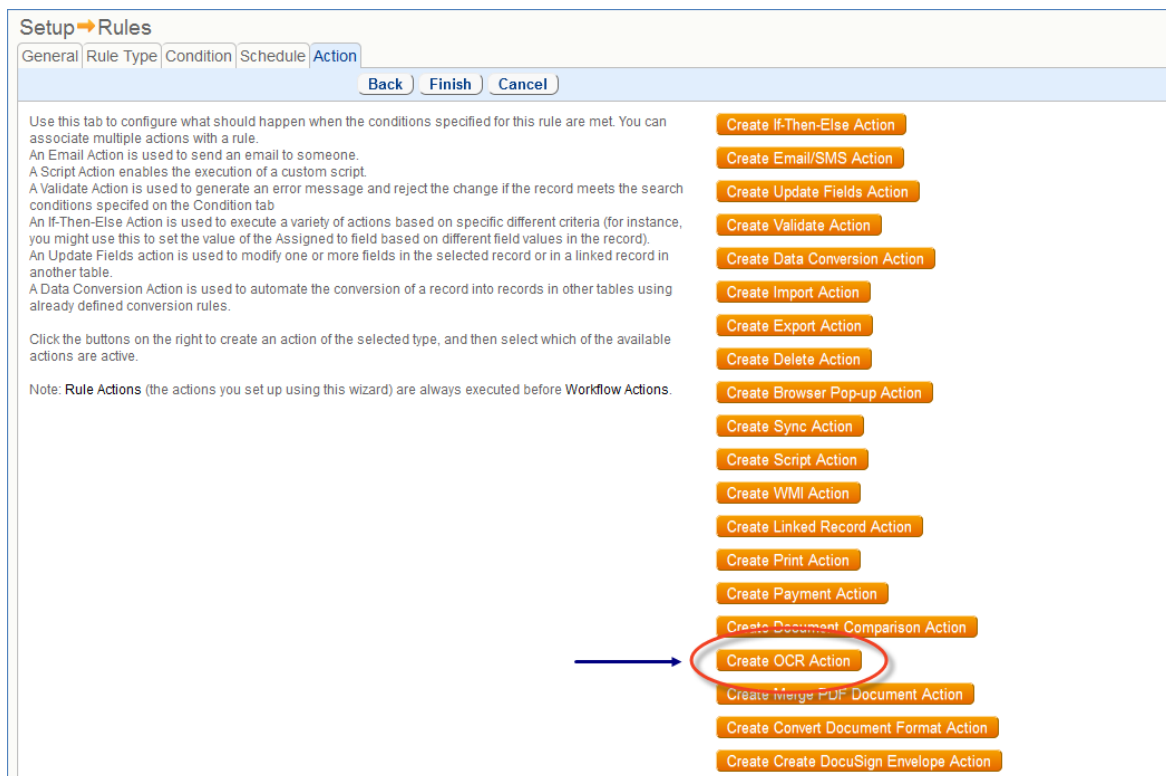
*It's best to use reports with run-time filters when your report action will be used in an Action Button. This way, when a user clicks the Action Button from a record, that record is automatically passed in to the run-time filters behind the scenes, and the user sees a report tailored to the context of the record.*

6. Choose how you want the output to appear. You can show the report in a new window or browser tab, or you can attach the report as a file to another field.
7. Choose which permissions you want to use. You can use the permissions of the user clicking the button, or select a user from the drop-down menu and always apply that user's permissions.  
Tip: If you're using the report action in a rule, it's best to select a user whose permissions will be used each time.
8. Click Finish.

## OCR Action

Optical Character Recognition (OCR) takes a PDF file from an attached file field, performs OCR on it, and creates a readable PDF that can be added to an attached file field. This creates a layer of readable text within the PDF that can be searched for keywords, and copy/pasted into another document.

The OCR action type allows users to choose select the most suitable way to perform optical character recognition (OCR) on attached files within the system. OCR Actions are prefixed with O: in the list of available actions.



## Create an OCR Action

1. In the Action window, click Create OCR Action to open the Action configuration screen.

General Options

Back Finish Cancel

Select the file field that will provide the source file(s) to be OCR'd. Note that if versioning is on in this file field, only the current version(s) of files will be OCR'd.

Source  
Attached File

Select the target location for the OCR'd file(s). We recommend using a second file field to store the OCR'd files, because this way the original file can be used for emailing, printing, and viewing, while the OCR'd files are used primarily for searching. In some cases, the OCR'd file may be much larger than the original file, which may prevent it from meeting email size limit restrictions. Note that files that do not need to be OCR'd or that will not benefit from OCR'ing will never be copied into the target field.

Store OCR'd file(s):

☐ In the original field

☐ Replace original files

☐ Append OCR files

☒ In File field: OCR File

☒ Overwrite/Update existing files

☐ Append OCR file(s)

Choose the text to append to the original file name when a file is OCR'd. All OCR'd files will add this string to the end of the file name

Append to file name:  
\_ocr

Back Finish Cancel

2. From the Source drop-down, choose from the available file fields in the current table.'
3. Under Store OCR'd file(s), choose how to store the OCR-converted files:
  - a. **In the original field:** Adds the converted file into the same field as the original.  
Replace original files: Replaces the original file with the converted one.  
Append OCR files: Appends the converted file to the field, while keeping the original.
  - b. **In File field:** Selecting a file field from the drop-down list will save the converted file into that field.  
Overwrite/Update existing files: Replaces the original file in the selected file field with the converted file.  
Append OCR file(s): Appends the converted file to the selected field, while keeping the original.
4. The Append to file name field allows you to define the wording that will be appended to the file name of the converted file. This may include field variables such as \$ID.
5. Click Finish to save the action.

## Convert Document Format Action

The Convert Document Format action can either take an MS Word document and convert it to a PDF, or take a PDF and convert it to MS Word format. The system can use either a built-in PDF converter, or Hosted MS Word Service to convert the document to PDF or to Word.

Conversion from PDF to Word uses the Aspose library API for the conversion by default, or you can use the Word Services Integration API to do the conversion.

*PDF to Word conversion uses a best effort approach, and cannot be guaranteed to be 100% accurate, particularly for complex PDF files. This is true whether you use Aspose or Microsoft services.*



## Create a Convert Document Format action

1. In the Action window, click Create Convert Document Format Action.
2. Enter an Action Name and Description.
3. For Document Conversion Type, choose whether to Convert MS Word to PDF or Convert PDF to MS Word.
  - a. **Source File:** Choose which file field holds the source file
  - b. **Destination File:** choose which file field will hold the converted file, and whether to append or overwrite any files.
4. If your Word Services have been configured, select the "Use Hosted MS Word Service to convert documents" checkbox.

The screenshot shows a configuration window for creating a 'Convert Document Format' action. It is divided into several sections:

- \*Action Name:** A text input field containing 'Convert to PDF'.
- Description:** A large, empty text area for providing a description.
- Document Conversion Type:** A dropdown menu currently set to 'Convert MS Word to PDF'.
- Source File:** A dropdown menu currently set to 'Test File 1'.
- Destination File:** A dropdown menu currently set to 'Merged File'. Below this are two radio button options: 'append the file to this field' (which is unselected) and 'overwrite any existing files in this field' (which is selected).
- Use Hosted MS Word Service to convert documents:** A checkbox that is checked.

5. Click Finish to save the action.

## Merge Documents Action

The Merge Documents action allows you to merge multiple Microsoft Word or PDF files into one file, and convert Word files to PDF. This action takes the existing documents in an attached file field and

combines them into one output file, depending on the configuration settings chosen by the user. Other file types such as images or .xlsx files are skipped.

The Merge Documents Action wizard consists of the following tabs:

## General

Add a name and description for the action.

## Data Source

- Select the source field to use for the merge action. The source can be a file field, a linked file field from another table, or a file field in a related table.
- Choose whether to save the document in another file field or open it in the browser window.
- Set a formula to create a name for the file.

## Options

General	Data Source	Options
<div>Back Finish Cancel</div>		
<p>Specify the format of the merged document.</p> <p>If the selected format of the merged document is PDF, then files with extension ".docx" and ".pdf" will be considered for merge. If the selected format of the merged document is MS Word, then only files with the extension ".docx" will be considered for merge.</p>		<p>Output Format: Select the format of the merged document</p> <p>PDF ▼</p>
<p>Define whether each document should start on a new page in the resulting merged file. Note: The option to concatenate documents without inserting any page breaks are applied while 1) Merging MS Word files into a single file 2) Merging MS Word files into a single file and converting that into a PDF.</p>		<p>Page Breaks:</p> <p><input checked="" type="radio"/> Start each document on a new page <input type="radio"/> Concatenate documents without inserting any page breaks</p>
<p>Select this checkbox if you want to include MS Word files while merging documents into PDF format.</p> <p>Select this option if you want to generate a merged MS Word document from each consecutive set of word documents before converting to pdf. This will enable consecutive pagination if the first document in the set has page numbers. It will also use the first file's style definitions for the set of word documents, making the converted pdf more consistent. Note that you may merge a combination of word and pdf documents, so only the word documents that are consecutive will be merged together, and they may be interrupted by pdf files and then another set of word files that are merged.</p> <p>Select the second option if you want to convert attachments of type MS Word document to PDF format before merging documents</p>		<p>Merge or Convert First:</p> <p><input checked="" type="checkbox"/> Include MS Word files during merge</p> <p><input checked="" type="radio"/> Merge consecutive MS Word documents before converting to PDF format <input type="radio"/> Convert MS Word documents to PDF format before merging</p> <p>Choose an option for PDF conversion</p> <p><input checked="" type="radio"/> Use MS word installed on this Agiloft server <input type="radio"/> Use Hosted MS Word Services on a remote Agiloft server</p>

- **Output Format** - Select whether to output the merged file as PDF or Word (.docx) format. Note that if the option chosen is 'MS Word (.docx)', the 'Merge or Convert First' section will not appear, as this deals with how to manage Word to PDF conversion.
- **Page Breaks** - Decide how to manage page breaks in places where two documents are merged. The documents can either be merged seamlessly without page breaks, or can insert a page break between each document. Page breaks can be managed while merging multiple Word documents into a single Word document, or while merging Word files into a single document and then converting into PDF.
- **Merge or Convert First** - Choose whether to include or ignore Word files during the merge. If Word files are included, you can also choose whether to merge the Word files consecutively

before performing the conversion, or individually converting each Word file before merging as a PDF.

- Choosing to merge consecutive Word files will give the appearance of a single document by numbering the pages across documents consecutively, and will take the first document's style definitions and header and footer for all the following documents.
- If a Merge Documents action contains a mixture of Word and PDF documents, the consecutive Word documents may be interrupted by PDF files, and if any Word files come after the PDF, they will take the next Word file as the first of its sequence.
- **Choose an option for PDF conversion** - Word to PDF conversion requires Hosted Microsoft Word API functionality to work; choose whether the conversion should use the current server, or a remote server.

#### Example

*If you want to create a single PDF from a group of five Word documents from a file field, such that the final merged output will have consecutive page numbers and appear as a single consistent document:*

#### Prerequisites

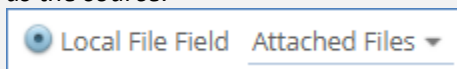
*Word API Hosted Services must be set up correctly in the knowledgebase.*

#### Steps

*In the Merge Documents Action wizard:*

*In the General tab, add a name and description for the action.*

*In the Data Source tab, select the local file with versioning field with multiple files enabled to use as the source.*



*Choose where to open or store the file output, and set a name for the document using the formula field.*

*In the Options tab:*

*Set the format of the merged document as "PDF".*

*Choose to concatenate the documents without inserting any page breaks.*

*Select the option to include MS Word files during the merge, and to merge consecutive MS Word documents before converting to PDF format.*

*Choose to use MS Word Services installed on this Agiloft server.*

*Save the action in an action button and place it on the layout. When the action is run on the attached file field with multiple Word files, the resulting file will be a merged PDF output.*

## Print Action

Print formats can be used to generate a PDF or Word document from a print template and attach it automatically to a record.

When this action runs, the print template selected in the Format drop-down list is executed, a file is created, and the file is attached to the defined field in the manner selected. This action creates history records and can trigger other actions like sending the file by email.

Print actions are commonly used in action buttons to automatically generate documents and attach them to a record.

## Prerequisites

- These steps assume you have already created the print template you want to use.
- Before you begin, create a File with Versioning field to store the documents created by your action button.

## Creating the Fields and Action Button

1. Log in to Agiloft as an admin and navigate to the Table wizard for the table where you want to show the action button.
2. On the Fields tab, create an action button:
  - a. Label the action button.
  - b. Select the Execute Action checkbox.
  - c. Click Add Action.

The screenshot shows a configuration form for an action button. It has several sections: 1. '\*Field label' with the text 'Add Project Report' and a 'Use HTML editor' button. 2. '\*Field name' with the text 'Add\_Project\_Report'. 3. '\*The field is displayed as' with two radio buttons: 'Text hyperlink Create and Attach' (selected) and 'Image hyperlink Select / manage...'. 4. 'Before executing actions:' with three checkboxes: 'Do nothing' (selected), 'Save record', and 'Validate required fields' (checked). Below these are 'Run rules and update default values' (checked) and 'then'. 5. 'then' section with 'Execute Actions' (checked) and a 'No items to drag' area containing 'Add Action', 'Remove', and 'Add Wait Point' buttons. 6. 'And' section with a dropdown menu showing 'Do Nothing'.

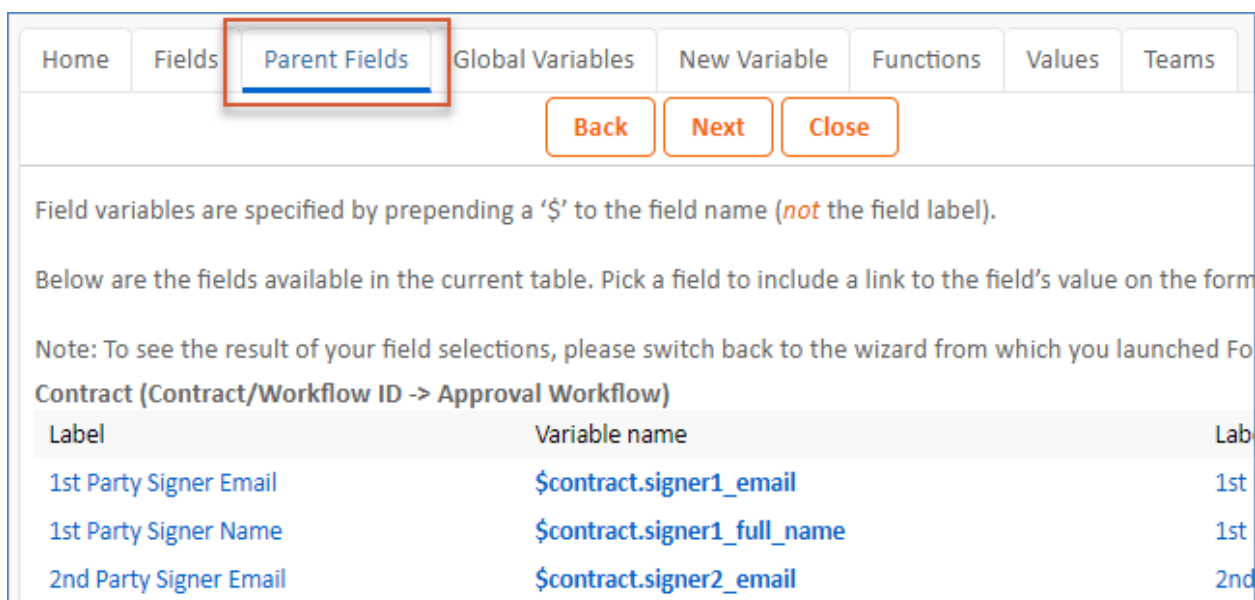
- d. In the pop-up window, click Create Print Action.
3. Create the action as follows:
    - a. Name the action.
    - b. In the Format field, select your print template.
    - c. Select Store printed document in File Field and select your File with Versioning field. Choose whether to append or overwrite any other files stored in the field.
    - d. In Define the default file name, use Formula Help to configure the automatic naming convention. It is generally helpful to include the record ID and date in the file name.
  4. Finish setting up the action button and Save it. Continue organizing the layout and permissions as needed.

## Linked Record Actions

Linked Record actions provide a way to copy information from one record into linked records that are one or more levels away. To do so, the action takes values from the current record and inputs them into one or more records that are linked through field relationships.

Linked record actions are used as containers for other actions, particularly Update Field actions, allowing direct access to fields that are linked through several levels of relationships. It's possible to update linked records using a normal Update Fields action, but in that case the update must be a fixed value or variable from the record being updated; the action can't copy a value from the current record. Linked Record actions can copy a value, such as the record ID, from the current record into linked records three levels away without creating intermediary linked fields, which triggers a relationship between those records that did not previously exist.

This allows you to pass values directly from the parent table for updating, which is not possible with an Update Field action. In other words, the table from which the action is running can use its own values in the linked field. In the Formula Help wizard for actions that are contained by a Linked Record action, the Parent Fields tab contains values from the current table, as well as values that belong to other tables in the chain of linked fields.



Home Fields **Parent Fields** Global Variables New Variable Functions Values Teams

Back Next Close

Field variables are specified by prepending a '\$' to the field name (*not* the field label).

Below are the fields available in the current table. Pick a field to include a link to the field's value on the form.

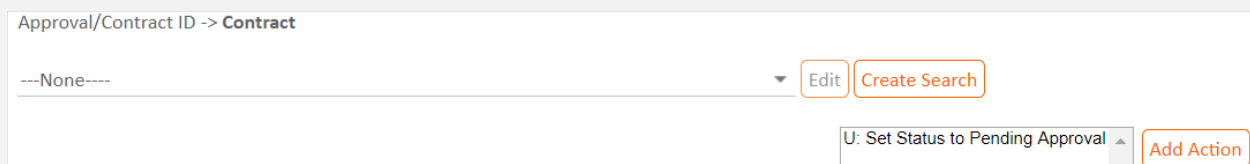
Note: To see the result of your field selections, please switch back to the wizard from which you launched Formula Help.

Contract (Contract/Workflow ID -> Approval Workflow)

Label	Variable name	Label
1st Party Signer Email	\$contract.signer1_email	1st
1st Party Signer Name	\$contract.signer1_full_name	1st
2nd Party Signer Email	\$contract.signer2_email	2nd

### Example

The Approvals table contains a linked field set to the Contracts table. This linked field set is used to create the "Set Contract Status to Pending Approval LR" Linked Record action in the Approvals table.



Approval/Contract ID -> Contract

---None---

Edit Create Search

U: Set Status to Pending Approval Add Action

On the Actions tab of the Linked Record Action wizard, clicking the Contract table in the chain shows the "Set Status to Pending Approval" Update Fields action. When triggered, this action sets the status of the linked Contract record to Pending Approval.

*When the status changes to Pending Approval, the "All Contract Approval Edit Actions" If-Then-Else action in the Approvals table runs a series of further actions related to the Approval.*

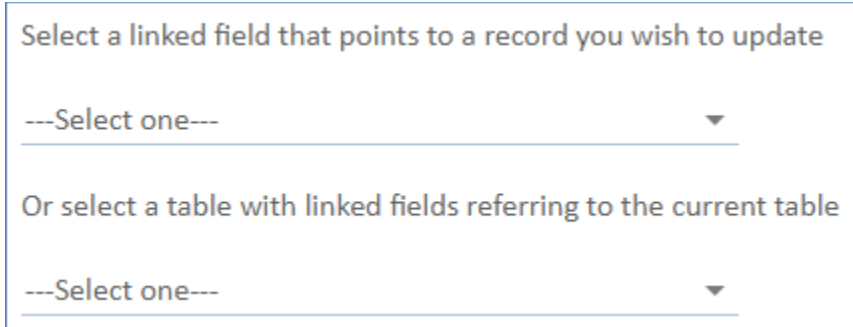
Linked Record actions are also commonly used to:

- Run an Email action from the other table to addresses held in that table that includes field values from the parent table.
- Run an If-Then-Else action to apply other conditions before updating a subset of the found records.

## Create a Linked Record Action

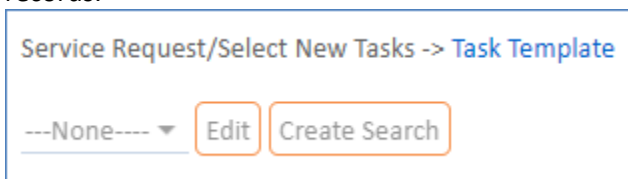
You can find the Actions wizard via several paths, but the easiest is to start by clicking Setup [TABLE NAME] for the table where you wish to create your action.

1. Select the Actions tab in the Table wizard.
2. Click Create Linked Record Action.
3. On the General tab, name your action and give it a description. Once your action is saved the system automatically adds an L: before your given title to distinguish the action as Linked Record.
4. Navigate to the Chain tab and select a linked field you want to update. The top drop-down menu contains active links, or relationships in which the current record pulls in one or more records from another table. The bottom drop-down menu contains passive links, or relationships in which records in another table have a link to one or more records in the current table.



*In most cases, choosing a linked relationship from the top drop-down menu will update a single linked record, while choosing a "referring" item such as a related table will update multiple records that have been linked to the current record.*

5. Navigate to the Actions tab and select the table hyperlink above the drop-down menu where you would like to perform your actions and, if desired, choose a saved search to filter the records.



6. Click Add Action to choose the actions to be executed on the found records. You can either select an existing action or create a new one. Actions may include any action type but are generally limited to If-Then-Else, Update Fields, and Email actions.
7. Click Finish.

## Use Cases

Let's take a look at two extended use cases to demonstrate how a Linked Record action may be used.

### Linking a Service Request to a User Record

In this example, suppose that you want each user record in the People table to show the last completed service request that the user submitted. To do so, you create a Linked Record action in the Service Requests table that adds the service request ID to the Last Completed Service Request field in the submitter's user record whenever one of the user's service requests is closed. You can initiate the Linked Record action with an action button, or a rule that closes the service request.

### Selecting the Table with the Linked Field

Assume that you've already created the Last Completed Service Request linked field. On the Chain tab of the Linked Record action wizard, select the table with the linked field that points to the Service Requests table. In this case, it's the People table and the field we just mentioned.

Remember, the top drop-down list contains active links, or linked fields in the current table pointing to another table. The bottom drop-down list contains passive links, or linked fields in other tables that point to the current table.

Select a linked field that points to a record you wish to update, or select a table with linked fields referring to the current table, creating a chain of linked updates:

---Select one---

Or select a table with linked fields referring to the current table:

---Select one---

---Select one---

- Change Requests from Spawning SR ID
- Documents from SR Additional Information,SR Assign
- Employees from Last Completed Service Request
- External Users from Last Completed Service Request
- Incidents from Service Request ID,Service Request St
- Incidents from Spawning SR ID
- People from Last Completed Service Request**

## Viewing the Relationship Chain

After making the above selection, the wizard displays a relationship chain. The chain shows the parent table, Service Requests, the current table, People, the selected linked fields, and the direction of the link.

Each time you select a linked relationship, both drop-down menus are updated with new linked relationships based on your selection. This allows you to work your way down through those links to the target table.

Create a chain of linked records to operate with.

Service Request <- People/Last Completed Service Request

## Emailing the Submitter's Department Head

Suppose you also want to email the submitter's department head that the service request was completed. Use the bottom drop-down list to select the linked set in the Department table that includes the Department Head field. When you do so, see that the relationship chain grows.

Create a chain of linked records to operate with.

Service Request <- People/Last Completed Service Request <- Department/Department Head Cell Phone Head, Department Head ID

## Creating Actions

On the Actions tab, notice the relationship chain and select each table in the chain to define an action to run at that level. After selecting the People table, create an Update Fields action to populate the Last Completed Service Request field with the service request ID. You can also select the Department table and create an Email action to email the department head that the service request has been completed, including the service request ID in the email.

## Using the Formula Wizard

When creating your Update Fields action, you can select the option on the Values tab to treat the new values as a formula. This lets you use the Formula wizard to insert values from the current record. With the Email action, you can directly click Formula Help in the email template to do the same.

Once in the Formula wizard, navigate to the Parent Fields tab to see a list of fields from the parent table, which is the Service Requests table in this example. To insert the service request ID, click the appropriate variable name.



Label	Variable name
Additional Fields to Show	\$helpdesk_case.show_dependent_fields_i
Employee Name	\$helpdesk_case.new_employee_name
Enable Ad Hoc Tasks	\$helpdesk_case.enable_ad_hoc_tasks
ID	\$helpdesk_case.id

## Using the Linked Record Action

Finally, you can create an action button or a rule in the Service Requests table that closes the service request and runs the Linked Record action. The Linked Record action then updates the Last Completed Service Request linked field in the People table with the Service Request ID value from the current service request. It also runs the Email action from the Department table and uses the Service Request ID value from the current service request.

## Generating Approvals Based on a Workflow

This use case is a real example from the Contracts table, and it's slightly more involved than the previous use case. When a contract needs approvals from other users, a user can click a Create Approvals action button to generate approval records based on the approval workflow associated with the contract. The approval workflow is the set of approvals a contract must go through before the contract is formally approved. For each approval, a record is generated that allows the appropriate user to approve or reject the contract at that stage in the approval workflow.

### Using the Create Approvals Button

The Create Approvals action button contains an If-Then-Else action that checks to see if the contract has a Workflow ID. The Workflow ID field in a Contract record is a linked field that links to the Approval Workflows table, which contains the set of approval workflows for each contract type. Depending on the Workflow ID, the contract goes through a specific approval workflow based on the contract type.

When a user clicks the Create Approvals action button and the If-Then-Else action finds that a Workflow ID has not been defined for the contract, it runs a Validate action with an error message. Otherwise, if the Workflow ID has been defined, it runs a Linked Record action.

Add Action
Add If
Add Elseif
Add Else
Add Comments
Move Up

```

if (Workflow ID=NULL) {
    There is no workflow defined
} else {
    Trigger Required and Conditional Approval Creation
}

```

## Selecting the Table with the Linked Field

When creating the above Linked Record action in the Contracts table, first create a chain to the Approval Workflows table using the Workflow ID field. In the Linked Record action wizard, select that field and table from the top drop-down list on the Chain tab. Remember, the top drop-down list contains linked fields in the current table that point to another table.

This begins the chain and creates the relationship to the Approval Workflows table. This step is necessary because this relationship allows you to create a further relationship to the Approval Templates table, which you do next.

The Approval Templates table is the table from which new approval records are generated. You need to make sure the correct approvals are generated for the contract type and that they're linked to the current contract. The Approval Templates table also contains its own Workflow ID field, and this field links to the Approval Workflows table, allowing you to create the relationship chain you need.

---Select one---

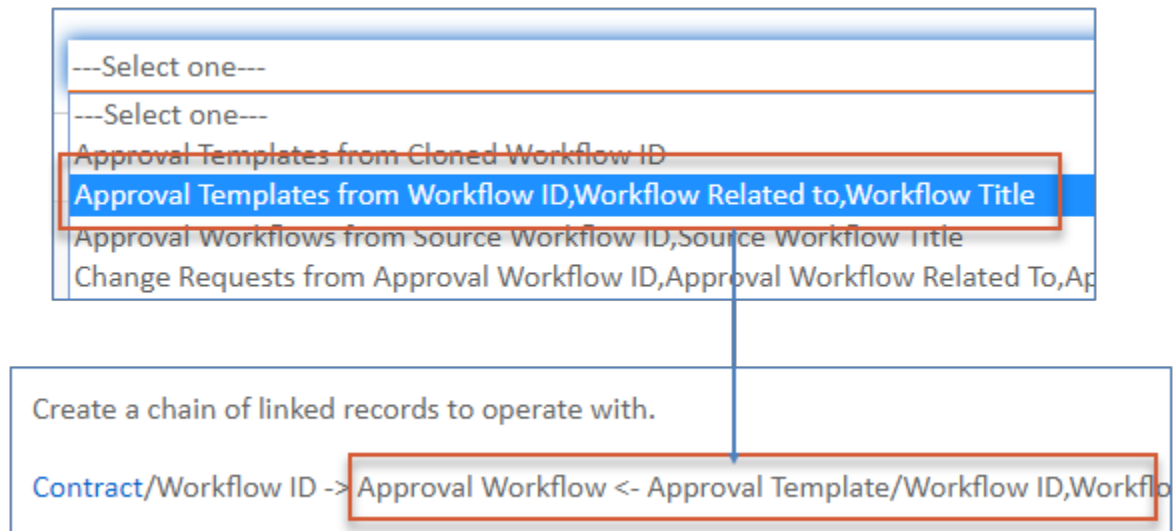
- Requester Email links to People
- Requester ID links to People
- Requester Name links to People
- Updater Email links to People
- Updater Team links to People
- Uploaded Attachment Type ID links to Attachment Type
- Workflow ID links to Approval Workflow**
- Workflow Title links to Approval Workflow
- Zip or Postal Code links to Location

Create a chain of linked records to operate with.

Contract/Workflow ID -> Approval Workflow

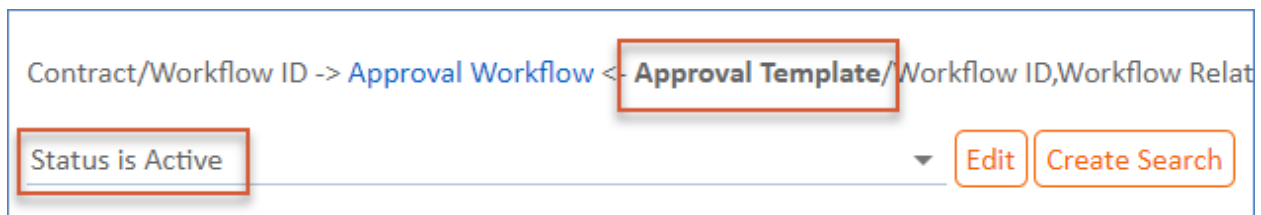
## Selecting the Next Table with the Linked Field

Instead of the top drop-down menu, this time you use the bottom drop-down menu because the linked relationship goes in the other direction: The Workflow ID field in the Approval Templates table links to the Approval Workflows table, the current table. As might be obvious, you choose the Approval Templates table with the linked field set containing the Workflow ID field from the list.



## Selecting a Saved Search

Now that you have your chain of linked records, you could add the action that starts the approvals process. However, in this case it's better to first create a saved search to return only approval records that have an Active status. You don't want to generate approvals that are inactive.



## Adding an Update Fields Action

After selecting the saved search, create and add an Update Fields action to the Linked Record action. When the Create Approvals action button that we discussed earlier is clicked, this action runs on the Approval Templates table so that a rule can ultimately generate the approvals. The action updates two fields. It pushes the contract ID from the current contract into the Latest Contract ID field in the approval templates that make up the approval workflow. This links those approval templates to the current contract.

*Remember, to use a value from the current record, select a variable from the Parent Fields tab of the Formula wizard. In this case, that variable is "\$contract.id," which gets its value from the contract's ID field.*

The other field the Update Fields action updates is the Trigger Approval Creation field, which it changes to Yes. This field, as the name suggests, acts as a trigger that a rule uses to generate the approval records by using a Conversion action.

Please enter new values for these fields

Latest Contract ID

\$contract.id

Treat the above as

☐ Standard text

☒ A formula

☐ Reimport source record matching current

☐ Use the 

MT: Company in Contract: Match

more info

Edit

Create New Search

☒ Choose the first record returned in sort order

☐ Choose all records that match the

Trigger Approval Creation

Yes

Treat the above as

☒ Standard text

☐ A formula



# Engage with Email and SMS

---

# Introduction

This first portion of this handout, Email and SMS Setup, includes information about Email and SMS Templates and how to configure SMS Service with the Agiloft built-in Twilio integration. The information provided in the first three main sections will walk you through the various options available and, along with the Agiloft Wiki articles, will aid you in setting up or improving the way you work with email and SMS communications.

The second portion of this handout, Non-Standard implementations, provides an overview of some alternative ways that you can set up Agiloft in order to meet the requirements of your organization. The information provided in this section will hopefully give you some ideas about how you make Agiloft work for your organization, whether or not something is expressly included in the demonstration system.

## Email Templates

### Where are they used?

1. Direct Email from All Communications table
2. Direct Email from related table of Communications within a record in another table
3. Direct Email from the Email icon at the top-right of the record
4. Direct Email from Email Icon in related table
5. Via Email Template, auto-send via action/action button
6. Via Email Template, pre-populated/allow user to edit via action button

### How are they accessed?

1. Setup>Email and SMS>Go to Rule/Workflow Emails>[Table]>Templates tab
2. [Table View] Email Icon>Manage Templates
3. Setup [Table]>Actions>[Email/SMS Action with Send Email selected]>Message tab
4. New Email>Insert>Populate from template

## Setting Up Email Templates

### Message Tab

#### From

Defines the address the recipient sees when they receive the email.

- Default Outbound Address for [Table]
  - o Defined at Setup>Email and SMS>Configure Outbound Email

- Inbound Account for [Table]
  - o Provides a lookup to select from all inbound accounts for this table
  - o Defined at Setup Email and SMS>Configure Inbound Email
- Email Address of the individual who last updated the [Table]
  - o Will use the email address of the last user who updated the email
  - o If that user does not have an email address in their profile, no email will be sent
  - o If the template is not triggered by an action button, or if the action button does not save first, the “From” address *may not* be the user who triggered the email
- Address from the Contact field
  - o Provides a lookup to select from a list of contacts defined in the [Table] record
  - o For instance, if you select Assigned Person, the email will come from the email address associated with the Assigned Person
  - o If that user does not have an email address in their profile, no email will be sent
- Address from the Email field
  - o Provides a lookup to select from a list of email fields defined in the [Table] record
  - o For instance, if you select Requester Email, the email will come from the email defined in the Requester Email field
  - o If the email field is blank, no email will be sent
- Address from the Ordinary field
  - o Provides a lookup to select from a list of all email and text fields defined in the [Table] record
  - o The options here *may or may not* be email addresses
  - o If the field is blank, or if the value is not an email address, not email will be sent
  - o Not recommended
- My email address
  - o User interacting with template/sending the email
- Address
  - o User defined email address

## Reply-to

Defines the address which will be populated if the user clicks “reply”

- Same Options
- You may use different “From” and “Reply-to” addresses
- This may be useful if you want the email to come from the Support Rep/Contract Manager, etc., but still want replies to be directed to a defined inbound email account

## To, CC, BCC

By default, only “To” is visible. You may add CC and/or BCC by hovering over “To” and selecting the desired options.

- Users
  - o Explicitly select from all users in the system
- Teams
  - o Explicitly select from all teams in the system
- User Fields
  - o Select from a list of contacts defined in the [Table] record

- For instance, send the email to the Main Contact
- Email Fields
  - Select from a list of email fields defined in the [Table] record
  - For instance, send the email to the address defined in the Support Email field
- Address in Field
  - Select from a list of all email and text fields defined in the [Table] record
  - Not recommended
- Link to Team
  - Select from a list of teams defined in the [Table] record
- Addresses
  - User defined email address

## Subject and Content

The Subject and Content areas allow for plain text or html and can contain variable formulas.

- By default, email templates will use HTML; you may change to plain text by hovering over HTML and selecting Plain Text
  - If using HTML, you may access the Source HTML by hovering over HTML and selecting Source HTML, which may make adjusting the content simpler for those comfortable with HTML
- Formulas may be added by clicking “Formula Help” or Insert>Variables
  - This will bring up a new window which includes the various formulas you can add; some examples include:
    - Values from fields in the source record:
      - Contract Expires on \$contract\_end\_date
    - Values from fields in records related to the source record:
      - Email Signature: \$formula(\$contract\_owner\_id.email\_signature)
- After selecting a formula, you will be provided a prompt asking if the formula should be added to the Subject Line or the Email Body
- You may also add formulas manually if you know the field names
  - When adding to the subject line, do not wrap in \$formula()
    - Check on this
  - When adding to the email body, it is not required to wrap in \$formula(), but it will prevent unexpected results due to trailing punctuation, etc.

## Attachments

You may add fixed attachments to the Attachments field. You may also add attachments defined within the [Table] record.

## Insert Dropdown

- Variables
- Populate from Template
  - Allows you to replace the content of the template based on the content of another
  - Templates active for your group will appear in the dropdown
  - All templates you have access to are visible under “Search...”



- Message to import
  - Allows you to insert the content of an email which has already been sent; any existing content will be retained
  - The content from the other email will be inserted to the location of your cursor
  - All emails you have access to will be available
- Fields from [Table]
  - Any fields selected here will display either at the footer of the email content or within an attached text file
  - Only fields which are on the layout may be selected
  - You may include:
    - All Fields
      - Will send all fields regardless of recipient's view permissions
    - Based on Recipients Access Permissions
      - Will send all fields for which the recipient has view permissions
    - Selected Fields
      - Will send selected fields regardless of the recipient's view permissions
      - test
  - You may define whether or not to include only those fields which have a value
- Include a Hyperlink to edit the [Table]
  - You may define a hyperlink which will (a) direct the recipient to access the record where the email originated and/or (b) update the value of a choice field and redirect a user to a defined URL
  - The hyperlink may be defined to open the record in view or edit mode or to immediately redirect a user to
    - If the hyperlink is defined to open the record for edit, and the recipient does not have edit permission to that record, the hyperlink will fail
  - The hyperlink may be configured to set field values for choice fields
    - This may be used in approval emails to Approve contracts from within the email, within support emails to allow customers to close the ticket without accessing the ticket directly, etc.
  - You will define where the user is redirected:
    - If hyperlink is opening for edit, defined separately for Finish and Cancel
    - If hyperlink is open for viewing, defined for close (or edit then finish)
    - When immediate redirect, defined upon button click
      - Most useful to redirect to a Custom URL with a splash page "Thank you for [performing action]"
    - Options include:
      - Default exit URL (defined at Setup>System>Manage Global Variables>Exit URL)
      - Custom URL
      - Leave the user logged in (for view/edit)
      - User's default interface with this table selected for viewing (for immediate redirect)

## Options Tab

### CC one copy of the email to Users (only for manually sent emails)

This option allows you to CC emails to the specified user(s). It is typically used when creating an email template, rather than when sending an individual email.

### Collate Results (only for manually sent emails)

Select whether the system will send one email per Case or collate results into a single email per each recipient.

For instance, if you are sending emails to the person assigned to an item, and if multiple results are found for each assignee, then Collate Results will collect all the Cases for that user and send them as a single email.

For example, if records 1, 3 and 5 are assigned to OddJob and records 2, 4 and 6 are assigned to EvenSteven, then OddJob will receive one email with data from 1, 3 and 5 and EvenSteven will receive an email with data from 2, 4 and 6.

### Include encrypted ID in subject line

This option controls whether an encrypted ID is added to the subject line of the email; for example, the subject line might contain ID:[9oehqw/790578/33557]. If the recipient responds to the email, the system will parse this text and update the appropriate record.

### Enable field parsing for replies

This option controls whether the text of any emails sent in response should be parsed in order to update individual fields in the record. For example, if the email body contained text "Priority: High", that could set the value of the Priority field. The help manual provides details of the format required in HTML and plain text emails.

Add example

### Include Hidden Code for Truncating replies

When an email is replied to, the response often includes the original email, so the length of the message grows with each exchange. This is cumbersome and unnecessary since the original emails are already stored and new responses can be concatenated. This option inserts a hidden code in the email that allows the system to recognize and discard the text from the original outbound email when the response arrives.

### Heading Tab (only for saved templates)

- Name
- Description

- Published
  - This option allows users with permission to view published templates to see this template. In addition to viewing their own templates, some groups have access to all templates and others to only published templates. These permissions are set on the Menu-Specific tab of the group permissions wizard.
- Make this active for
  - This option determines whether the template appears in the drop-down list of active templates for those with permission to view it.  
To make this template active for users who can view only published templates, you must also publish the template in the option above.  
Even if the template is not active in the drop-down list, those with permission to view it can select it in the Manage Templates dialog.

## Update via Export/Import and Mass Edit

Email and SMS Templates can be updated via data import. If there is a common change which you need to make to a number of template records, it will be more time efficient to perform that update via export/import or mass edit.

With import/export, you can export all fields within the template record

With mass edit, the following fields are available:

- Message Body
- Message Subject
- Published
- Template Active For (Teams)
- Template Name

## Gotchas

If the from, reply-to point to a blank address, no email will be sent. If the To, CC, BCC point to a blank address, no email will be sent to those without a properly defined email address

For emails sent to users who are not in the system which include a hyperlink, the hyperlink will log the user in as the Anonymous user

## Related Wiki Articles

Email Templates: <https://wiki.agiloft.com/display/HELP/Email+Templates>

Inbound Email: <https://wiki.agiloft.com/display/HELP/Inbound+Email+Accounts>

Email Parsing: <https://wiki.agiloft.com/display/HELP/Email+Parsing>

Exporting Data: <https://wiki.agiloft.com/display/HELP/Exporting+Record+Data>

Importing Data: <https://wiki.agiloft.com/display/HELP/Importing+Record+Data>

Mass Edit: <https://wiki.agiloft.com/display/HELP/Mass+Editing+Records>

# SMS Templates

Much the same as Email Templates, but fewer options are available.

## Where are they used?

1. Direct SMS from All Communications table
2. Direct SMS from related table of Communications within a record in another table
3. Direct SMS from Email Icon in related table
4. Via SMS Template, auto-send via action/action button

## How are they accessed?

5. Setup>Email and SMS>Access SMS Templates>[Table]>Templates tab
6. [Table View] SMS Icon>Manage Templates
7. Setup [Table]>Actions>[Email/SMS Action with Send SMS selected]>Message tab
8. New SMS>Insert>Populate from template

# Setting Up SMS Templates

## Message Tab

### From

Defines the number the recipient sees when they receive the SMS message. You may select from accounts configured at Setup>Email and SMS>SMS Services [Configure]

### To

- Users
  - o Explicitly select from all users in the system
- Teams
  - o Explicitly select from all teams in the system
- User Fields
  - o Select from a list of contacts defined in the [Table] record
  - o For instance, send the SMS to the Main Contact
- Phone Fields
  - o Select from a list of phone fields defined in the [Table] record
  - o For instance, send the SMS to the address defined in the Requester Phone field
- Phone no's
  - o User defined phone numbers

## Insert Dropdown

- Variables
- Populate from Template
  - o Allows you to replace the content of the template based on the content of another
  - o Templates active for your group will appear in the dropdown
  - o All templates you have access to are visible under “Search...”
- Message to import (only for manually sent SMS)
  - o Allows you to insert the content of an SMS which has already been sent; any existing content will be retained
  - o The content from the other SMS will be inserted to the location of your cursor
  - o All SMS records you have access to will be available

## Heading Tab (only for saved templates)

- Template Name
- Description
- Published
  - o This option allows users with permission to view published templates to see this template. In addition to viewing their own templates, some groups have access to all templates and others to only published templates. These permissions are set on the Menu-Specific tab of the group permissions wizard.
- Make this active for
  - o This option determines whether the template appears in the drop-down list of active templates for those with permission to view it.  
To make this template active for users who can view only published templates, you must also publish the template in the option above.  
Even if the template is not active in the drop-down list, those with permission to view it can select it in the Manage Templates dialog.

## Related Wiki Articles

SMS Templates: <https://wiki.agiloft.com/display/HELP/Adding+SMS+Templates>

Sending SMS: <https://wiki.agiloft.com/display/HELP/Sending+SMS+Messages>

## Configuring SMS Integration with Twilio

Prod: <https://www.twilio.com/>

Test/Dev: <https://www.twilio.com/try-twilio>

1. Sign up for account
2. Note the following pieces of information:
  - a. Account ID (SID) – Located on Dashboard
  - b. Authentication Token – Located on Dashboard

- c. Either/or
    - i. From Number (number, short number, alias associated with the account)
    - ii. Message ID (MSID)
3. Within Agiloft, Setup>Email and SMS>SMS Services [Configure]>New
4. Provide an Account Name, Account Label, and Account Description
5. Enter the SID, Authentication Token, From or MSID
6. Usage = Mass Communication
7. Enter a phone number in the field "Send a test SMS message from this account to" and click Send Test SMS to test the configuration
8. Click Finish
9. Navigate to Setup>Email and SMS>Define Default Outbound Account
10. Select a default account from the drop-down menu
11. Optionally, enter a valid number to test the service and click Send Test SMS
12. Click Finish
13. If using multiple SMS Accounts, you may define which account to use by default for each table at Setup>Email and SMS>Configure Outbound SMS>[Table]>From Options
14. Select Use this Account, select the appropriate account, and click Finish
  - a. By default, the option "Use the default SMS account for outbound communications" is selected

## Related Wiki Articles

Configuring SMS: <https://wiki.agiloft.com/display/HELP/Configuring+SMS+Service>

## Non-Standard Implementations

I am assuming that you all know the basics of Email and SMS setup and usage. As a brief overview:

- Email communications can be set up for two-way communication with outbound emails largely sent via templates and inbound email captured through inbound email accounts
- SMS communications can be setup for outbound communications only, sent largely from templates
- Outbound emails and SMS can be triggered manually or via actions
- Outbound emails will contain an encrypted ID in the email header and optionally within the subject line. This allows for replies captured by an inbound email account to be associated with the appropriate record in the appropriate table
  - Note – the encrypted ID in the email header *may be removed by the recipient's email client* and if so, Agiloft will only have the subject line to work with
- If no matching record is identified (email is not a reply, or encrypted ID not included in subject line and removed from the email header), the inbound email account may be configured to allow new records to be created by email
- Inbound email accounts can be configured to set the value of one or many choice fields to some explicit value

# Approve via Email from outside the Firewall

I have a customer who is self-hosted and has Agiloft installed on a server which is behind a firewall and inaccessible from outside their corporate network.

This same customer also has executive staff who are often out of the office on business or pleasure and meant to be approving contracts from wherever they may be.

So, how can we allow users to Approve a contract they are unable to access?

## Out-of-the-box Overview

In the out-of-the-box set up, Approval emails include a hyperlink to approve from email.

From the user perspective, once they click the link they are directed to some pre-defined URL.

In practice, clicking this link logs the user in, set the value of a field called Approved by Email to Yes, logs the user out, and redirects them to a defined exit URL.

## Mailto

The approve by email hyperlink will not work in this case because the user will not be able to access Agiloft unless they are inside the corporate network.

For this customer, we replaced the standard hyperlink with a mailto link. You have seen mailto links, these are found on most company websites on the contact us page.




Clicking on a mailto link will open your default email client and pre-populate the To address. Mailto links can be setup to pre-populate the To, CC, BCC, Subject, and Body.

The mailto link we used for this customer opens a new email which populates the **To**, **Subject**, and **Body**. The Approval email has three separate links: one each for Approve, Require Changes, and Reject. The example below is for Approve:

```
<a href="mailto:approvals@company.com?subject=Contract%20#$formula%28$contract_id%29%20requires%20approval%20ID%3A%5Bwsg7dy%2F0000%2F$formula%28$id%29%5D&amp;body=%2D%2DApproval%3B%20en%3B%2D%2D%0D%0aStatus%3AApproved%0D%0aApproval%20Notes%3A%0D%2D%2DApproval%2D%2D" target="_top">Approved</a>
```

Note that the contents of the subject and body are URL encoded.

The Contract ID and Approval ID are populated by variables in the template so that when clicked, the link will open an email like the below:

 Send	To...	<a href="mailto:approvals@company.com">approvals@company.com</a>
	Cc...	
	Bcc...	
Subject		Contract #1354 requires approval ID:[wsg7dy/0000/6815]
--Approval; en;-- Status:Approved Approval Notes: --Approval--		

The user must only (a) add Approval notes by typing into that section (optional) and (b) click send.

The inbound account at [approvals@company.com](mailto:approvals@company.com) has email parsing turned on and will set the Status and Approval Notes based on the inbound email.

## Receiving SMS

When we first announced our integration with SMS via Twilio, one of my customers asked that I setup inbound SMS in order to generate tickets when a user send a text message to a particular number.

SMS integration is limited to outbound message sending; we do not have a way to accept inbound sms in the same way we have Inbound Email accounts.

So, how can we accept inbound SMS when the SMS integration does not include this functionality?

There are a number of platforms which will deliver received SMS messages to a given email address. In order to meet the customer's requirement, they contracted with a 3<sup>rd</sup> party to have SMS forwarded to an email address. In Agiloft, we set up an Inbound Email account. When tickets are created via this email address, we set Communication Method to SMS. This allows for their response to the person raising the ticket to occur via SMS as well.

## Parsing the Subject Line

A customer calls one day and tells me that they need to email scanned PDFs, directly from the scanner, and have those PDFs be associated with the correct contract record. This customer has a fairly standard Contract/Attachment relationship where multiple Attachments may be linked to a single Contract record.

## Out-of-the-box Overview

In the standard, out-of-the-box setup, inbound email to the Contract table will convert any attached files to create Attachment records linked to the Contract. In this case, the customer will not be replying to any email, but their scanner does allow them to define the subject line.



As we know, inbound email will be associated with a specific record if, and only if, the encrypted ID is present in the subject line or the email header. While the encrypted ID can be manually typed, it presents a high chance of user error if the format is entered incorrectly.

So, how can we create Attachment records by email and have them linked to the correct Contract record?

## Inbound Email Account at Attachment table

- Subject maps to Inbound Email Subject (short text)
- Attached files map to Attached File (file with versioning)
- Record creation is allowed

## Create Rule at Attachment Table (runs on email only)

- U: Parse Contract ID from Email Subject

Parsed Contract ID -> replace( \$inbound\_email\_subject, "^.\*ID:\\s?([\\^\\V]\*\\V\\d\*\\V)?(\\d\*).\*\$", "\$2")

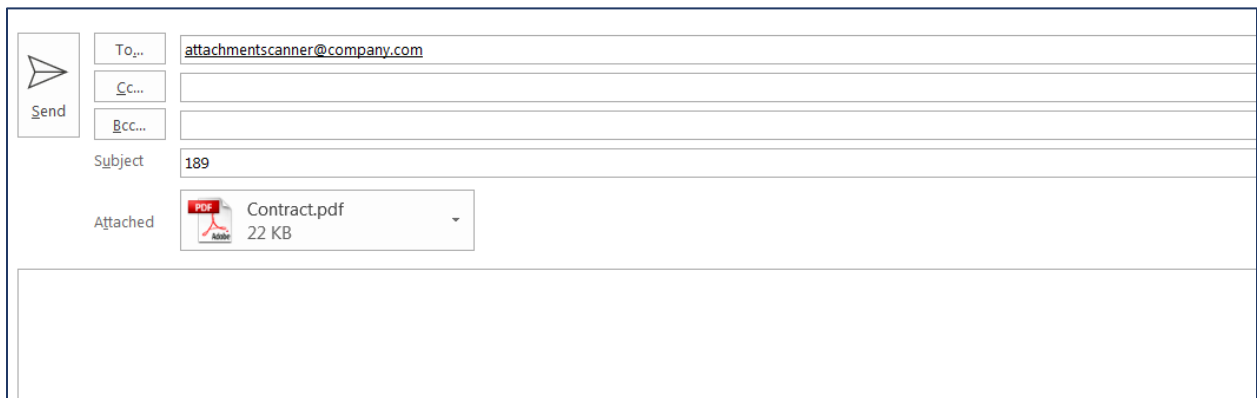
\*Uses regex to extract the ID in case the format is ID: [Table Key/All Comms ID/Record ID]

- U: Set Contract ID based on Parsed Contract ID

Contract ID -> \$parsed\_contract\_id (set in update fields action above)

## Process Flow

1. User sends an email to [attachmentscanner@company.com](mailto:attachmentscanner@company.com) with scanned attachment and enters the subject as the Contract ID



The screenshot shows an email composition window. On the left is a 'Send' button with a paper plane icon. To its right are input fields for 'To...', 'Cc...', and 'Bcc...'. The 'To...' field contains the email address 'attachmentscanner@company.com'. Below these fields is the 'Subject' field, which contains the text '189'. Under the 'Subject' field is the 'Attached' section, which shows a PDF icon, the filename 'Contract.pdf', and the size '22 KB'. Below the attachment information is a large empty text area for the email body.

2. Inbound email creates a new Attachment record

Attachment

Save
Cancel
Attachment
Contract Info
Emails
History
<< >>

Collapse All
ID 139

Attachment Information

\*Attached File Attach/Manage  
1. Contract.pdf

Include in Approval Packet Yes To Be eSigned No

Contract Information

Inbound Email Subject 189

Parsed Contract ID

Contract ID Contract Status

Contract Title

Contract Party Name

Contract Start Date Contract End Date

Contract Requester Name Internal Contract Owner

Contract Description

Contract Department

Contract Amount

- When the new Attachment record is created, Agiloft parses the ID from the subject line, and establishes the link to the correct Contract record

Attachment

Save
Cancel
Attachment
Contract Info
Emails
History
<< >>

Collapse All

ID139

Attachment Information

\*Attached FileAttach/Manage
1. Contract.pdf

Include in Approval PacketYes
To Be eSignedNo

Contract Information

Inbound Email Subject189

Parsed Contract ID189

Contract ID189
Contract StatusPending Contract Manager

Contract TitleAgiloft NDA

Contract Party NameAgiloft

Contract Start DateSep 01 2019
Contract End DateSep 01 2029

Contract Requester NameHelen Jensen
Internal Contract OwnerHelen Jensen

Contract DescriptionId like a standard NDA for this client

Contract DepartmentHR

## Update Records via Email without Replying to System Generated Email

One of my customers is a public facing organization who handles complaints issued by members of the public regarding services received by private organizations; a public advocate. The users who are submitting complaints do not have direct access to Agiloft and they may have one or many open complaints open in parallel.

While implementing their system, an interesting requirement arose. As they cannot expect members of the public to follow a restrictive process, Agiloft needs to be able to correctly process incoming email so that it is associated with an existing ticket or creates a new ticket, regardless of whether or not the email is in reply to some Agiloft generated email. Imagine the complainant emailing in additional information – of course we want this associated with their existing case and not creating a new case.

The encrypted ID is not included in the subject line as they do not want the public to believe they are receiving a system-generated email.

So, how can we process an inbound email, which is not a direct reply to an Agiloft generated email, and have it either be associated with an existing case if one exists or create a new case if this is a new complaint?

## Email Templates From Ticket to Complainant

All email templates which send email to complainants include hidden text. Within the email template, we include the below in a small, white-faced font:

- Initially, we used email parsing and included the text below:  
--Ticket ID Details; en;--  
Ticket: \$id  
--Ticket ID Details--
- But they have since moved away from email parsing and now include only the below:  
Ticket ID:\$id
- Using the below to parse the Ticket ID:  
Parsed Ticket ID -> `replace($body,"(?s)^.*Ticket ID:\\s*([0-9-]+).+$", "$1")`  
\*Uses regex to extract the ID from directly after "Ticket ID:"

They moved away from inbound email parsing due to some email clients/user settings converting "-" (two short dashes) into "—" (single long dash). Email parsing is very specific in the way the formatting needs to be set up and any deviation in syntax will result in a failure to parse the email as expected.

## Inbound Email Table

We have set up an Inbound Email table to pre-process emails received by one of many Inbound Email accounts. The fields in the Inbound Email table mimic the All Communications record.

Email Details	
From	consumer@memberofpublic.com
To	consumerline@governmentorg.gov
CCs	
Attached Files	<a href="#">Attach/Manage</a> 1. image001.png 2. image002.png
Subject	Regulations for Robotic Customer Calling
Body	<p>Dear Public Advocate,</p> <p>I want to check if there is regulations around robotic customer call to landlines.Is it possible and acceptable to have a robotic calling service members of the public?</p> <p>Looking forward to hearing from you,</p> <p>Kind Regards,</p> <p>Q. Public</p>

The purpose of this table is to pre-process emails received by the system. All public facing "contact us" pages for this organization include the Inbound Email account. All emails sent tot his address will create a new record in the Inbound Email table.

The Inbound Email account maps the From, To, CC, Subject, Body, and Attachments to the matching field.

When a new record is created in the Inbound Email table, a single rule runs which attempts to parse the email for the Ticket ID.

If there is a match, the system will establish a link to the existing Ticket and forward the email to the Inbound Email account associated with the Ticket table.

If there is no such match, the record will appear in the Support queue. Members of the support staff will review each email and determine if the email should be:

- Converted to a new Ticket
- Associated with an existing Ticket
- Treated as SPAM requiring no processing

## Email Template from Inbound Email to Ticket

When an email is identified as being associated with an existing ticket, the system forwards the email from the Inbound Email table to the Ticket table. The Email Template is set as follows:

- From: Address from the Email field "From"
- Reply-to: Address from the Email field "From"
- To: Inbound Email table for Ticket
- CC: *Excluded to avoid double emailing anyone in CC*
- Subject: \$subject
- Body: From: \$formula(\$from)  
To: \$formula(\$to)  
CC: \$formula(\$ccs)  
  
\$formula(\$body)  
  
--Ticket ID Details; en;--  
Date Last Email Received from Consumer: \$date\_created  
Email Subject: \$subject  
--Ticket Details--
- Fields Included: \$attached\_files

## Handling Emails differently across the Organization

For one of my multi-national customers, we were onboarding a new regional office (Parent Organization). For this customer, one of their goals is to consolidate processes across regions as much as possible, but complete and forced standardization faces some hurdles (1) legal requirements differ across countries and (2) the larger the change in process for any particular office the lower the adoption rate.

As part of their ongoing process, this customer gathers business owners from the various offices a couple times a year to discuss their internal processes, how each office is using Agiloft, and any new requirements.

Note that multiple offices share a single instance of Agiloft.

During one of these meetings, a discussion arose in which it was clear that we needed to handle email differently for each of the offices. Each office has their own email address which emails should be sent from. Some offices want replies to be processed by Agiloft, others require that any replies be sent to a

group inbox which Agiloft does not have access to, and some, thrilled by the idea, decided they need both.

It is agreed that given the number of offices, having a separate inbound email accounts and email templates for each option would not be tenable.

So, how can we handle email behavior differently for different areas of an organization without greatly expanding the number of templates which need to be created and maintained?

## Out-of-the-box Overview

In the out-of-the-box CLM module, we have the following setup:

- On the Email tab, we have the option to send an email to internal contacts, external contacts, or both
- There are three templates, one for each option above
- Replies to those emails are captured by an Inbound Email Account, and linked to the Contract where the initial email originated

## Options Defined for each Office

Agiloft holds a record of each regional office (Parent Organization). Within each Parent Organization, we configure the following options:

Email Addresses			
Include Encrypted ID in Subject Line	<input type="text" value="Yes"/>		
Default "From" Email	<input type="text" value="company_contracts@company.co"/>	Default "Reply to" Email	<input type="text" value="inboundemail@company.com"/>
Default "Forward to" Email	<input type="text" value="contracts_table@company.com"/>	Default "Bounce" Email	<input type="text" value="contract_admin@company.com"/>

## Email Templates

We have reduced the necessary templates to two, one which includes the encrypted ID and one which does not. With the exception of this option, the templates match:

For sending emails to colleagues or clients, each office can configure for themselves the From and Reply-to address. This allows each office/division to use their own from address, define if replies to the email should come back into Agiloft or sent to a group email box, as well as whether or not to include the encrypted ID.

Not visible here (because the font face is colored in white), is the following hidden text:

```
--Parsed Info; en;--
Parsed Contract ID from Email Body:$formula($id)
--Parsed Info--
```

Also not visible, all attachments created by an Agiloft print template include the Contract ID in a specific format as a DOCVARIABLE: {DOCVARIABLE contract\_id=\$contract\_id}. This is not visible to the recipient of the attachment.

Both of these will be used later when any replies are captured by Agiloft.

## Inbound Email Table

Similar to the case above, we created a new Inbound Email table, which mimics the All Communications record.

ID	53
From	john.doe@customer.com
To	company_contracts@company.com
CCs	jane.doe@customer.com
Attached Files	<a href="#">Attach/Manage</a> 1. <a href="#">Contract.docx</a>
Email Subject	RE: Please review the attached contract ID: [asd9y/16054/10942]
Email Body	Hi John,  Thank you for sending the contract over for review. Please find a redlined copy attached. I will be out of the office starting Thursday, but you can include Jane on any updates and she will handle in my absence.  Best Regards,

The purpose of this table is to pre-process emails received by the system. If the Reply-to address matches the system defined Inbound Email account for the Inbound Email table, replies will create a new record in this table.

The Inbound Email account maps the From, To, CC, Subject, Body, and Attachments to the matching field.

The From address establishes a link to a user in the system, including the office they are associated with. This sets the "Forward To" and "Bounce" addresses.

Further, the Inbound Email table has two "helper" fields:

- Encrypted ID Start, set by default to "ID: [Table Key/2016/"
- Parsed Contract ID from Email Body

When a new record is created in the Inbound Email table, two rules run in sequence:

- The first:
  - o Attempts to parse the Contract ID from:
    - Subject Line via the encrypted ID
    - Attached File via the DOCVARIABLE
    - Email Template via email parsing using the hidden text
  - o If the Contract ID is identified, the link to the contract is set
  - o If the Contract ID is not identified, the email is forwarded to the Default "Bounce" Address
  - o The "Email Subject to Forward" is set by appending "Contract ID]" to the value in Encrypted ID Start
- The second:
  - o Forwards the email to the Default "Forward To" Address
    - This includes the constructed encrypted ID so the email will be attached to the appropriate Contact record
    - The From, Subject, Body, etc. all match the original email so the user reviewing the email at the Contract is unaware of any pre-processing
    - The CC is not included in the forwarded email



- Note that the Default “Forward To” address may include multiple addresses, so the initial email may be distributed to the Inbound Email account for Contracts as well as any group inboxes

## Inbound Email Accounts

We set up two Inbound Email Accounts, one for the Inbound Email table to receive replies to be preprocessed; the other for the Contracts table.

If the Default “Reply-To” address for an office is defined as the Inbound Email account for the Inbound Email table, Agiloft will process it.

Once processed, if the appropriate office can be identified, the system will forward the email to the Default “Forward” address. If that forward address is defined as the Inbound Email account for the Contracts table, the email will be displayed within the related Contract record.



# Care for Your Database

---

# Table Field Limits

This topic records the known limitations for columns in a table, i.e. how many fields of which size you can fit in a table before you run into database limitations.

## Column and Size Limits

There are two kinds of limit.

- The first is the total column size limit, based on the lengths and data types of the fields in the table.
- The second limit is the individual record size limit, called the page size, which is the size of a single record that is being saved.

## Total Column Size Limit in MySQL Database: 65,535 bytes

- **Once you hit this limit, you cannot add another field to the table.** You will need to delete fields, reduce their length.
- **Short Text** fields are classified as VARCHAR datatype, and their space is calculated at  $(n*3)+2$ , where n is the maximum length of the field. So if you create a short text field with a maximum of 255 characters, it will actually take up 767 bytes! Avoid making short text fields that need a high maximum size value.
- **Linked fields** take up the same amount of space as they would in their own table, so if you have a short text linked field defined as 255 maximum, it will take up 768 in your target table as well. If you have a choice field, it will take up 2-4 bytes. A link to a multichoice field such as contract type extra fields to show will take up 767 bytes.
- **Text and Append Only Text** fields are classified as MEDIUMTEXT datatype and they do not vary in size. They will always be ~41 bytes regardless of maximum length. This is because they store only a pointer in the current table and the rest is stored/indexed in another table. So if you know you are going to have a large number of text fields, you can save a lot of space by using the text data type, if the field needs to be longer than say, 40 characters. The downside is that they take a little longer to load, which could be noticeable if you have a very large number of them in a record. They also cannot be defined to be unique.
- **Multichoice fields** - these take 767 bytes, avoid them when you are concerned with size limits.
- **Single Choice, Integer, Date, Floating fields** - these are extremely tiny, 2-8 bytes, and you can have a very large number of them.
- **Link to Single Field with multiple values** - these take only 41 bytes and store the remainder elsewhere as a MEDIUMTEXT field.
- **Link to Team Name field** - this takes 602 bytes, as the source field in the teams table is hard coded to be 200 characters in length, use sparingly.
- **Attached File field:** an attached file field is stored as MEDIUMTEXT and uses ~41 characters, whether it holds one or multiple values, and whether it is local or a linked field holding multiple values (such as Files to Send).

## Page Size Limit for both databases: about 8060 bytes

- Limit of how large a record can be if all fields are filled to their capacity.
- You will get warnings about approaching or surpassing this limit, but these are warnings only, and do not prevent you from adding fields
- If an individual record goes above this limit, you cannot save the record.
- Text/Append Only Text fields use 38+3 bytes for this limit, non-variable.

## Field Best Practices when you expect a lot of fields

- If a Short Text field is not absolutely necessary (unique, minimum length) and the maximum length is expected to be larger than 30-40, use a text field instead.
- If a Short Text field is necessary, select the smallest possible max char length. Remember that it will be multiplied by 3.
- Use Choice fields over Multi-Choice fields. Choice fields take 4 bytes while Multi-Choice fields take 767 bytes.
- Links to Teams take 602 bytes, so use only when needed.
- Don't worry about link to single field with mve or multivalue file fields, as they are not space consuming.

## Working with History Size Limits

Occasionally we get an error that a new field cannot be added to History because the size limit has been reached for history. Below is information about the impact of data types and their column size on history:

- When a new field is added to a table, it is also adding **two** fields to History. One that takes up the same space as the field itself, and a secondary field which takes up 20 bytes for linked fields and text/short text fields not part of a LF+SV. For all other types, the secondary field is the same size taking up double the space of the field type.
- When a field is **deleted** from the system, it frees up the same number of bytes in history as in the table itself, and also frees up the extra space that was added in the History table.
- When tracking linked sets in history, it is not necessary to track every field in a linked set that you are running advanced filters on, only the ones that the advanced filters are actually based on. For example, in a linked set from the Person table, you may need to track the Full Name field but not the Email or Phone Number fields.

## History Best Practices

- When adding a lot of new fields, particularly in a system that already has a lot of data, turn off "automatically add fields to History tracking", then add the fields, and then choose the ones needing to be tracked afterward. This is particularly useful when close to or at the size limit.

- In a large table, we will almost always track fields selectively, rather than tracking them all.
- In a linked set, only track the fields you may need to use an advanced search on, and don't track the rest of them.
- Remember that If something is not tracked by History, it cannot be shown in History, nor can it be used in an advanced filter.
- Be careful removing fields that may have rules based on changes to them.
- **Always track in History:**
  - Date Updated
  - Status
  - Any fields being used in advanced saved search filters
  - Flag fields used in rules
  - Date Created - surprisingly, this seems to be needed and can break rules if removed
- **Often removable from History:**
  - Extraneous linked fields (Phone Number, Title, etc.)
  - Fields that will never be searched for changes or needed for advanced filters
  - Type (unless you have subtables)
  - Creator Linked Set
  - Team fields that you don't need for advanced filters, as they are large.
  - Multichoice fields not used in advanced filters
- **Never to be tracked in History:**
  - ID

## Indexing

Indexes are meant to improve saved or advanced ad-hoc search speed for specific fields. The system automatically performs a full text index of all data in the system on all tables, including in attached files. The Indexes tab in the Table wizard allows administrators to create additional indexes for specific fields that are used very frequently in searching to speed up those searches. From the Indexes tab, you can also change the fields that appear in the drop-down of the main search block.

*It is not usually necessary to add a custom index, but it's worth understanding when they're useful so you can recognize when a custom index can help search performance in your knowledgebase.*

## Creating Indexes

You should only turn on indexing for specific tables if you are comfortable working with databases, have a good feel for the makeup of the data, and know which searches are most frequently used. If a certain field is often used for searching, and the number of matching records is much lower than the number of records in the knowledgebase, you can improve search speed by adding a database index for that field.

For example, adding an index to the Assigned To field will increase performance if there are 100 records assigned to a particular team or individual in a database of 100,000 records. However, if there are 50,000 records assigned to that team or individual, adding indexing will not help at all and may actually hurt performance.

To set up a field index:

1. Navigate to **Setup > Tables**.
2. Select the table where you want to add an index and click Edit.
3. Click the Indexes tab.
4. Click New.
5. Select the field you want to index and click Finish.

When you save the index setting, the system takes the time to index the selected field immediately.

Typically, you would only index a single field at a time, but it is possible to create a compound index by choosing multiple fields. For example, if you have a frequently-used saved search that uses both the Status and Assigned to fields, and the number of matching records is much lower than the number of records in the knowledgebase, indexing can improve the performance of that saved search.

## Considerations for Creating Indexes

If you are unsure whether adding an index is right for your situation, contact Agiloft consulting services to discuss the options or to request assistance.

Because database indexing that is implemented poorly can adversely affect knowledgebase performance, it is important to be careful and consider the following:

- Database locking makes the table unavailable while you are adding an index. For this reason, consider adding indexing outside of working hours.
- Inappropriate or excessive indexes will actually hurt performance. In general, tables should not have more than 7 or 8 indexes.

## Checking Indexing Progress

Indexing can take several hours for tables with hundreds of thousands of records or several days when a knowledgebase has been copied or moved and indexing is processing for many tables. You can visit a URL, specific to your server and your knowledgebase, to check the progress of indexing. The following is the basic format:

`http://[SERVER]/gui2/tests/ftsstatus.jsp?kb=[PROJECT_NAME]`

Change the server and project name accordingly. For example:

`http://server13.enterprisewizard.com/gui2/tests/ftsstatus.jsp?kb=Demo`

*While indexing is still in progress, you can use string searching as a workaround until indexing is complete. See Search Options on the Main Search Block page for a description of string searching.*

## Synchronizing Indexes

Indexes may sometimes become out of sync with the system database, which can prevent the index from working properly. This can occur if the system is restarted while an index is being created or deleted. You can use the Check Indexes button and Synchronize Indexes button on the Indexes tab to identify and address synchronization issues:

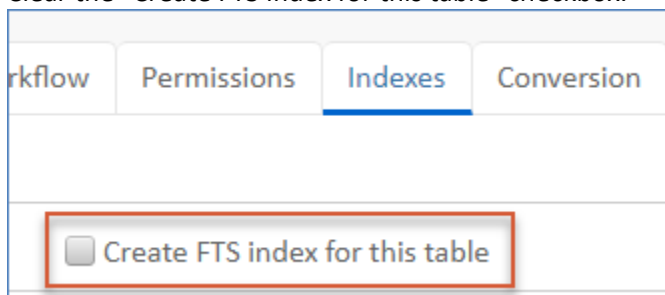
- The Check Indexes button compares the indexes found in the database with the indexes recognized by Agiloft and reports discrepancies. The button opens the report in a new window that lists the index name, the indexed fields recognized by Agiloft, the indexed fields found in the database, and the index status.
- The Synchronize Indexes button syncs the table's indexes with the database. The amount of time to complete this process depends on the number of records in the table. For example, a table containing one million records takes about five minutes to sync. You only need to sync with the database if the Check Indexes button reports a problem, or if you otherwise suspect that the database is out of sync, such as a log file message that reports a synchronization issue.

## Turning Off Indexing

The indexing process is resource intensive and should be turned off for tables where it is unnecessary, such as tables that do not require instantaneous searches of their content. The All Communications table is a good example of where indexing should be turned off because searching on this table is not common. Turning off indexing means that you cannot choose to make searches on certain fields faster, but it will improve the overall search performance of the entire table.

To turn off field indexing:

1. Navigate to **Setup > Tables**.
2. Select the table where you want to turn off indexing and click Edit.
3. Click the Indexes tab.
4. Clear the "Create FTS index for this table" checkbox.



5. Click Finish.

## How to Let Users Delete Records without Delete Permission

The permission to delete records is often reserved for admin level users and access should be given with discretion. This brief tutorial demonstrates how to:

- Configure an action button that allows users to mark records for deletion, so they do not require full delete permission.
- Create a time-based rule and Delete action that regularly delete the records marked by users with the action button.

This configuration is best used in situations when data recovery is critical, when accidental record deletion is likely otherwise a serious concern. Depending on the table and the degree to which record contain linked data, deleting records can be resource intensive and appear slow while the system repairs and deletes links. A time-based rule to delete marked records can therefore shift the performance load to off-peak hours.

## Mark for Deletion Button

First, create an action button that users can access from a record view.

Before starting your button configuration, ensure that your table includes a Pending Deletion status.

1. From the Fields tab of the Table menu, create a new action button.
2. Label the button Mark for Deletion.
3. Under Before executing actions, choose Do nothing and then Execute Actions. Click Add Action.

## Update Field Actions

Your action button requires one primary Update Field action to change the status of the record.

Configure the field with the following parameters:

1. In the Action wizard, select Update Field Action.
2. On the General tab, name your action Set Status to Pending Deletion.
3. On the Fields tab select Status and then click Next.
4. Under the Status drop-down, select Pending Deletion. Click Finish.
5. In the Action wizard, ensure your new action is selected and click Finish.

## Delete Action

In addition to the action button, your configuration also requires a delete action that deletes marked records after a certain number of days. After a user clicks the Mark for Deletion button that you created above, the admin can choose to change the status, which prevents the selected records from deletion.

1. Name your rule TB: Daily: Delete records pending deletion.
2. On the Rule Type tab, choose At selected time intervals.
3. On the Condition tab, create a saved search with the following conditions:
  - a. Status is equal to Pending Deletion.



- b. Date Updated is greater than 30 days old.

- c. Name your search R: Pending Deletion more than 30 days.

4. On the Action tab, click Create Delete Action.
5. In the Delete Action wizard, name your action Delete Attachment record and choose Delete the record itself. Click Finish.
6. Ensure that your new action is selected and click Finish to complete your rule.

## Use Case Considerations

The configuration described above is a basic guideline for a delete button of this nature. However, it is important to consider the table where your button exists and how the records in that table are used. When a user clicks the Mark for Deletion button, they aren't truly deleting the record yet and it will continue to be used in other actions and rules if the system is set up to do so.

For example, this button could be configured on the attachments table in a contract management solution. For this use case, the admin should consider rules affecting the Attachments table and where attachments are included in the system or sent by email. If, for instance, an approval packet is emailed with every contract and documents are pulled into the packet from the attachments table, the admin needs to create or edit a rule and any helper fields to prevent the record marked Pending Deletion from being included in the packet.

## Activity Logs Setup

Activity logs maintain a record of any specified statistics of system usage, which can assist with auditing your system. You can view current activity logs in the Activity Logs table.

It's often useful to create rules based on the events you track. For instance, you could create a rule to notify you when someone changes a workflow, rule, or user record. You can also set up reports for activities that pose a security concern. For instance, you could create a report for failed logins that alerts you regarding potential attempts to breach the login screen.

*The Activity Logs table is not visible by default. To add the table to your left pane, complete these steps:*

1. Navigate to **Setup > Tables** and select Activity Log.
2. Click Unhide.
3. Navigate to **Home > Preferences > Left Pane Setup > Tables** and select Activity Log.
4. Click Add and then click Finish.

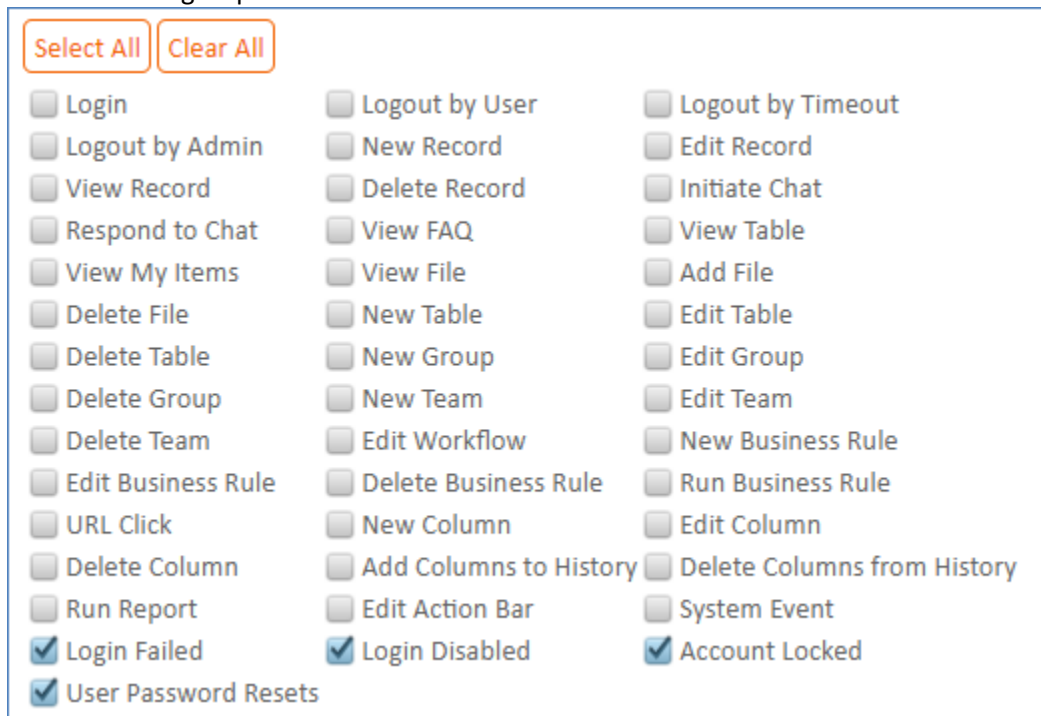
# Configure Activity Logs

You can configure your activity logs by creating different audit rules. Audit rules define which system activity is tracked and how long activity logs are maintained. To view existing audit rules, navigate to **Setup > System > Configure Activity Log**. You can create new audit rules by clicking New to open the Audit Rules wizard.

The Audit Rules wizard is a single screen that allows you to choose to track various system events and create saved searches that determine which users' actions are logged. Many different kinds of events can be tracked: login failures, action bar edits, record views, and others. The system creates records in the Activity Log table when the events you select meet the saved search criteria.

To create an audit rule:

1. Go to **Setup > System > Configure Activity Log**.
2. Click New to open the Audit Rules wizard.
3. Name the audit rule and select a language for the audit report.
4. Create or select a saved search that determines which users actions are logged.
5. Select which events are logged for the users meeting the saved search criteria. An event is only logged if it selected from this list and is executed by a user who meets the saved search criteria. For example, you might want to only log specific actions from admin users, such as when they edit a team or group.



The screenshot shows a configuration window for audit rules. At the top left are two buttons: "Select All" and "Clear All". Below them is a grid of 30 system events, each with a checkbox. The events are organized into three columns. The first column contains 10 events, the second column contains 10 events, and the third column contains 10 events. The events are: Login, Logout by Admin, View Record, Respond to Chat, View My Items, Delete File, Delete Table, Delete Group, Delete Team, Edit Business Rule, URL Click, Delete Column, Run Report, Login Failed, User Password Resets, Logout by User, New Record, Delete Record, View FAQ, View File, New Table, New Group, New Team, Edit Workflow, Delete Business Rule, New Column, Add Columns to History, Edit Action Bar, Logout by Timeout, Edit Record, Initiate Chat, View Table, Add File, Edit Table, Edit Group, Edit Team, New Business Rule, Run Business Rule, Edit Column, Delete Columns from History, System Event, and Account Locked. The checkboxes for "Login Failed", "User Password Resets", "Login Disabled", and "Account Locked" are checked.

<input type="checkbox"/> Login	<input type="checkbox"/> Logout by User	<input type="checkbox"/> Logout by Timeout
<input type="checkbox"/> Logout by Admin	<input type="checkbox"/> New Record	<input type="checkbox"/> Edit Record
<input type="checkbox"/> View Record	<input type="checkbox"/> Delete Record	<input type="checkbox"/> Initiate Chat
<input type="checkbox"/> Respond to Chat	<input type="checkbox"/> View FAQ	<input type="checkbox"/> View Table
<input type="checkbox"/> View My Items	<input type="checkbox"/> View File	<input type="checkbox"/> Add File
<input type="checkbox"/> Delete File	<input type="checkbox"/> New Table	<input type="checkbox"/> Edit Table
<input type="checkbox"/> Delete Table	<input type="checkbox"/> New Group	<input type="checkbox"/> Edit Group
<input type="checkbox"/> Delete Group	<input type="checkbox"/> New Team	<input type="checkbox"/> Edit Team
<input type="checkbox"/> Delete Team	<input type="checkbox"/> Edit Workflow	<input type="checkbox"/> New Business Rule
<input type="checkbox"/> Edit Business Rule	<input type="checkbox"/> Delete Business Rule	<input type="checkbox"/> Run Business Rule
<input type="checkbox"/> URL Click	<input type="checkbox"/> New Column	<input type="checkbox"/> Edit Column
<input type="checkbox"/> Delete Column	<input type="checkbox"/> Add Columns to History	<input type="checkbox"/> Delete Columns from History
<input type="checkbox"/> Run Report	<input type="checkbox"/> Edit Action Bar	<input type="checkbox"/> System Event
<input checked="" type="checkbox"/> Login Failed	<input checked="" type="checkbox"/> Login Disabled	<input checked="" type="checkbox"/> Account Locked
<input checked="" type="checkbox"/> User Password Resets		

6. Define for how long the system retains entries created by the audit rule. If an audit rule is deleted, any records created by the rule are also deleted.
7. Click Finish.

*Log files generated from activity logs and other system activity are deleted with the Logeraser utility. By default, log files are deleted after 30 days. To change Logeraser's default settings, you must have access to the file directory on the server.*



# Secure Your Organization

---

## User Controls

- Passwords – password strength and reset controls
- Groups – carefully control user permissions
- Deactivate Users – safely removing users

## Access Controls

- SSO – use SSO through IdP for safe user access
- Hotlinks – authentication control for email hotlinks
- 2FA – enforce 2FA logins for privileged users
- IP Restrictions – control where system can be accessed from
- Web Services – control who can utilize web services

## Data Controls

- Text fields:
  - Security Check
  - Data masking
- File Scanning
- Export Controls

## Global Variables

- MANY security variables available for restricting/allowing access – use with care!

## About Agiloft

Over 3 million users at organizations ranging from small enterprises to U.S government agencies and Fortune 100 companies depend on Agiloft's top rated product suites for Contract Management, Service Desk, Custom Workflow, and more. Agiloft specializes in automating processes that are too complex for competing vendors. Our best practice templates and adaptable technology ensure rapid deployment and a fully extensible system. For more information, visit <https://www.agiloft.com>.