



AGILOFT REFERENCE MANUAL

HIGH LEVEL OVERVIEW

INTRODUCTION

Agiloft is a comprehensive system for automating business processes, managing relationships with customers, staff, contractors and suppliers and helping staff and managers gain insight into effective strategies for improving profitability.

The product is ready to use immediately. It is also fully extensible without requiring conventional software programming - meaning that you can add custom fields, tables and relationships between tables, as well as reports, workflows and business processes through your browser using a graphical user interface.

This guide introduces Agiloft concepts, describes how they work together to help your organization, and provides detailed examples of how you might customize it to meet your needs.

Customization and system administration is 100% wizard based, so this guide is reasonably short as it avoids duplicating lower level details that are explicitly described in these wizards. For example, the section in this guide that describes the integer field type does not list all the available options on the wizard, such as making the field required, limiting the maximum/minimum input value or providing default values, because these options are fully detailed in the wizard itself.

Agiloft is built on a few powerful concepts and once you understand them, it is relatively easy to master the entire system. These concepts are summarized in the next section.

See Also:

[Concepts](#)

[Business Example](#)

[Initial Implementation](#)

[Installation](#)

CONCEPTS

Concept	Brief Explanation
Rule	A statement of business logic that Agiloft executes automatically.
Field	A piece of information, such as a Telephone Number..
Record	A collection of field values.
Workflow	Workflow refers to the rules that apply when changes are made to a particular type of record.
Object (Table)	A table is the whole set of fields, relationships, and rules on which individual records are based. A KnowledgeBase typically has several different types of tables, such as Cases, Leads, Contracts, Users, and so on.
Sub-Table	Sub-tables are made up of specifically defined records within a larger table. Tables can be hierarchical and inherit fields and rules from their parents. For example Hardware Issue and Software Issue might be sub-tables of the Bug Table, in which case they would automatically contain all the fields from Bug as well as their own unique fields.
Group	Groups define the permissions that a user has within Agiloft. These group permissions define what records and fields a user may view or edit.
Team	Teams are used to identify working units of users, and are assigned as Primary Team as well as sub-teams. Primary Teams are used to control the interface that a user sees on login.
Report	A summary of record information that may be presented in graphical, HTML, text or Excel format.

Search

A filter applied to a table to bring up specific information. For example "Find all Support Contracts that are due to expire within the next 30 days".

View

A setting that determines what fields from a particular type of record are shown. For example, "Show the Summary, Severity and Assigned To fields of bug reports".

KnowledgeBase

The entire collection of objects, records, teams, groups, views, reports and associated data.

The next section describes how these concepts are applied to automating a business.

See Also:

[Business Example](#)

[Initial Implementation](#)

[Installation](#)

BUSINESS EXAMPLE

Let's take a company that:

- 1) Sells five different software products and consulting services and wants to provide web-based forms to its customers so that questions, bug reports and requests for products/services can be submitted online 24/7.
- 2) Has some customers that do not want to be bothered by emails, but prefer to login remotely to perform spot-checks on their open issues.
- 3) Has other customers who want a weekly status report on their open issues, but are each interested in seeing different information. For example, the service customers want to know when their particular project will be completed, while the product customers want to know about the status of their enhancement requests.
- 4) Has another set of customers who want to define their own custom reports that are distributed on the 15th of each month. Let's assume that this last set of customers is based in Japan and does not understand English.
- 5) The Sales Manager wants to be notified if the number of Leads that are predicted to close within the next 30 days drops below 20 or exceeds 45.

Just satisfying all these requirements keeps entire departments busy at some companies using error-prone manual processes.

Let's look at how it might be handled with Agiloft.

Action

The Engineering Manager creates Case sub-tables for each software product and consulting service.

The record types share some common fields such as Date Submitted and Severity, but also have their own unique custom fields such as Operating System for the software products and Contract Duration for the consulting services.

The Engineering Manager exports a NLS (National Language Support) file from Agiloft that contains these

Comments

This satisfies requirement 1.

When issues are submitted via the web interface, required fields, choice tables and pop-up explanatory text ensure that all the necessary information is collected via the web form and so eliminates rounds of telephone tag.

Creating half a dozen different sub-tables of an existing table, each with some custom fields, takes a couple of hours.

The NLS file only contains the

custom fields and passes it to a translator who converts the custom fields and input instructions into Japanese. For immediate turnaround of simple text, a machine translator can be used.

The System Administrator creates a Group called Customer (actually such a group is provided with the system) and specifies that members of this group be allowed to see Cases that they have submitted, but may not see cases that other people have submitted. He also specifies that members of the Customer group can create Reports.

The Support Manager creates a Team called "Weekly Report" and adds the customers who requested a weekly report to this team.

Next, she creates a search for "Open" Cases and creates a Weekly Status report that executes at 6am each Monday morning and distributes information as a graphical chart and HTML report to the Weekly Report team.

She specifies that the HTML output should be formatted using the Personal view for the recipient.

The Japanese customers login and create the custom reports they desire. Because their browsers are set to Japanese, the GUI wizards, status messages, field labels, etc. are automatically shown in

custom fields. Exporting the NLS file, translating a couple of dozen words and a few sentences of input instructions into another language can be accomplished in less than an hour.

This satisfies requirement 2.

Creating a custom Group takes about ten minutes.

This satisfies requirement 3.

Recipients receive information formatted according to their Personal View of the Cases table. The Group permissions automatically filter the output so that the report sent to each user only shows the records they are allowed to see.

Creating a custom Team and a Saved Search takes less than five minutes each. Assigning all the relevant customers to this team takes another couple of minutes using Mass Edit.

Creating the Weekly Status report takes ten minutes.

This satisfies requirement 4 and takes no additional company resources.

Japanese.

The Sales Manager defines a business rule to send email whenever the number of Leads with an expected close date of less than 30 days in the future drops below 20 and defines another rule to provide notification if the number rises above 45.

This satisfies requirement 5.

Creating a couple of custom business rules takes about twenty minutes.

SUMMARY

The entire set of business requirements has been satisfied in the space of a single morning and will operate 24/7 without burning up company resources.

COMMENTS

Much of the code in Agiloft is dedicated to making this all seem easy.

In the example above, the system was modified by an Engineering Manager, Sales Manager and System Admin, and because all Agiloft customization is wizard driven, it really is possible for the business managers themselves to keep the system up to date with the ever changing needs of their departments. Further, the Group permissions system can prevent the less skilled individuals from damaging the system. For example, someone might belong to a group that has authority to delete Reports that they created, but not those created by other users.

Of course, in many companies the business managers will work through a system admin to make the changes that they require. However, even administrators appreciate software that allows them to complete such tasks in real time.

INTEGRITY MANAGER

The Integrity Manager is one of the most powerful modules in Agiloft, it operates behind the scenes to ensure system integrity and provide self-management facilities that ensure that every part of the system coordinates with every other part.

For example, as soon as the custom field named Operating System was added to the Cases table, it automatically showed up as one of the options in the wizard used to define new Reports. To take another example - if the System Admin tries to delete the Search that looked for Open Cases - the system automatically finds whether the search is used by the Weekly Status report and provides options such as canceling the delete request, choosing another Search to take its place or deleting the Report.

CONCLUSION

Now that we have seen an example of how these components work together, let's examine them in more detail, together with some examples. The next section also introduces a handful of concepts that are not really central to the system but will be helpful when considering how to configure it to meet your particular needs.

See Also:

[Concepts](#)

[Initial Implementation](#)

[Installation](#)

[Record](#)

[Rule](#)

[Team](#)

[Group](#)

INITIAL IMPLEMENTATION

Agiloft works right out of the box, but of course it works best once you have configured it to meet the particular needs of your department. We provide flash presentations that will guide you through configuring your system.

The most important thing you can do at this point is to relax.

- You have purchased an award-winning product that can be fully configured to meet the needs of a typical department in a few days.
- Agiloft comes with an unconditional money back guarantee so if you are not happy for any reason, you can get a full cash refund any time within 3 months of purchase.
- Wizards will guide all your actions; the Integrity Manager guarantees system integrity and our support staff are always available to help.

So don't be afraid to experiment! If you want to try something really crazy, you can always export the KnowledgeBase in Agiloft format first to create a backup. But even if you forget this step and manage to delete everything, you will still be able to recover from one of the automatic backups via the admin console.

Try not to get bogged down worrying about details when you are planning your implementation. If you don't like something, you can change it with a few mouse clicks. In the old days it was so difficult to configure an enterprise system that organizations used to spend months planning and trying to anticipate every eventuality, followed by many more months in implementation only to find that by the time they were finished, the original business needs had changed.

Now you can make changes, see the effect and gather feedback from your users in real time. This process is so much faster than rapid prototyping and the result is so much more than a prototype, that it should really be called extreme-development.

The result of the prototyping phase is a fully functional, working system. We recommend developing it in a couple of stages, as detailed in the next section.

See Also:

[Implementation Stages](#)

[Stage 1](#)

[Stage 2](#)

[Concepts](#)

[Business Example](#)

IMPLEMENTATION STAGES

We recommend implementing Agiloft in stages as shown below, together with time estimates to help you plan your implementation.

The next section goes into detail about the steps involved in each stage.

Stage 1: Get familiar with the capabilities of the system

All the steps in this Stage are leading towards the point where you demonstrate the software to your colleagues and decide what business process to automate first.

When you have finished, you will be in a good position to answer any question that begins "In Agiloft can you...?".

This will take less than a day, but of course there is quite a difference between knowing what can be done and really understanding the best way of doing it.

Stage 2: Automate first business process

During this stage you put the software into full production use and demonstrate early success to your colleagues.

When you have finished, the software will be paying for itself.

Typical working time: 3-7 days. Elapsed time: 1-3 weeks

Stage 3-> Automate additional business process

As you automate each additional process, the value to your organization grows and because everything is working from the same KnowledgeBase, it is fully integrated with the preceding processes.

Typical working time per process: 3 days. Elapsed time: 1-2 weeks

See Also:

[Stage 1](#)

[Stage 2](#)

[Concepts](#)

[Business Example](#)

[Initial Implementation](#)

STAGE 1: GET FAMILIAR WITH THE CAPABILITIES OF THE SYSTEM

Click the following hotlink to see flash presentations that show how to create custom tables and custom table relationship, configure workflow, create custom charts, etc.



[Overview of Capabilities](#)

Then try the following steps for yourself. Do not worry about the details at this stage, the goal is just to reach the point where you can understand the broad capabilities of the system and can hold a meeting with your colleagues to decide what process to implement first.

Stage 1: Get familiar with the capabilities of the system	Time Estimate
Install the software	20 minutes
Review the flash presentations here	4 hours
Read this manual	1 hour
Configure incoming email/outgoing email	10 minutes
Customize the look and feel to include your company logo and colors	30 minutes
Create a custom table and add a couple of fields to it	20 minutes
Create a custom workflow for this table	10 minutes
Create a custom business rule for this table	10 minutes

See Also:

[Stage 2](#)

Agiloft Reference Manual

[Initial Implementation](#)

[Implementation Stages](#)

[Concepts](#)

[Business Example](#)

STAGE 2: AUTOMATE FIRST BUSINESS PROCESS

Now that you understand the capabilities of the system, you can help your colleagues choose which business process to automate first, and put it into full production use.

An outline of the steps is given below, followed by more detailed instructions.

Stage 2: Automate first business process	Time Estimate
Team Meeting and Demo - demonstrate the software to your colleagues, decide what business process you want to automate first.	2 hours
Create/Customize Tables - create custom forms for these business processes, and for the information that the system will handle.	4 hours
Create Teams - model company hierarchy and any customer/reseller relationships with teams	20 minutes
Create Groups - control access permissions for special groups of users if necessary. You may use the default groups, or create your own.	40 minutes
Configure Email - configure incoming email to automatically create and update records, setup outgoing email to send updates to end users.	20 minutes
Create Workflows - automate the flow of information in your business process and the assignment of tasks to particular individuals/teams.	2 hours
Create Business Rules - monitor system use and activity, and alert the appropriate people when necessary.	20 minutes
Create User Accounts - create user accounts or configure the system to use an LDAP or Active Directory server.	1 hour
Import Existing Data - import existing user data from Excel, ASCII or XML files.	1 hour

Demonstrate the Working System - gather feedback and make adjustments in real time while training staff to use the system effectively.	4 hours
Start Production Use - deploy the new Agiloft system in parallel with the existing system for the first week. Adjust as necessary based on user feedback.	1-2 weeks
Start Next Automation - hold a meeting now that you've got the first process running, discuss what you learned and determine how this can be applied to the next process.	2 hours

Let's look at each of the above actions in more detail.

Team Meeting and Demo

Demonstrate the software to your colleagues and decide what business process you want to automate first.

The ideal business process to automate is one that is reasonably self-contained (i.e. does not require integration with several other software systems), provides a clear and measurable benefit to your staff, managers or customers and occurs frequently enough that the overhead cost of automation is rapidly returned. You may wish to ask yourself "what issue is causing us the most trouble?"

It is easiest to start by automating one or two critical functions within a single department. The experience and success of this implementation will make it easy to roll the functionality out to additional departments later.

Agiloft's ease of use makes it tempting to involve the entire company immediately. However, your business managers' previous experiences with inflexible enterprise systems may lead to unnecessary over-thinking of the process steps. Since Agiloft can be modified in real-time, meticulous specifications are not needed immediately. We recommend allowing managers to experiment with project demonstrations so they can see for themselves how quickly and easily the program can be modified. Implementation will go much faster if all people involved understand how flexible the system is.

We recommend growing organically, starting from a position of proven success and user satisfaction. If your CEO wants an immediate system-wide deployment, remember that it will not be expected overnight. You can position the initial deployment as pure learning exercise and design the enterprise deployment in parallel to this trial implementation.

In order to provide a fairly comprehensive example, we will automate a complex, multi-step process, as summarized below, but you may want to start with only with two or three such steps.

1. Purchase Leads from a list vendor and import them into the system
2. Contact Leads and try to make a sale
3. Collect credit card information from new customers and share it with some members of the Accounting department
4. Forward new sales to the Accounting department for billing
5. Follow up each sale with a support call to confirm that the customer is happy and possibly make a follow-on sale
6. Convert happy customers into reference accounts
7. Remind new Customers that they can access an on-line dynamic FAQ 24/7 and get answers to questions directly via the web.

Create/Customize Tables

Create forms for the information that the system will handle for these business processes..

In our example, we have Leads, Customers and Answers. Talk to the sales people to find out what information is provided by the list vendor, what additional information is gathered about Leads on the phone, and what information is required for Customers. It's unnecessary to keep notes on paper, as it is quicker and easier to enter the information straight into the system.

Your system comes pre-configured with a Leads table, but there is nothing special about these pre-defined tables. If you are using Agiloft to address a very special need in your company, you may want to create custom tables and custom relationships between them. If you don't find a pre-made table that meets your needs, you can create one by selecting Setup/Tables. From this screen, you can create an entirely new table by selecting Object (at the top of the tree) and clicking New, or you can create a sub-type of an existing table by selecting the parent table's title, clicking New and letting the wizard guide you.

When considering whether to create a new table or a sub-table of an existing table, a good question to ask is simply: "What seems most logical?" For example, many companies think of sales leads as a particular type of Contact, so it is natural to create the Leads table as a sub-type of Contact. You can also create a Service Contract as a type of Contact, but it isn't logical since a Service Contract is a legal contract between two companies, and not a Contact which is usually a person.

Another good question to ask: "Is it reasonable to expect that I will need to turn something of type A into type B". For example, it is reasonable that Contacts may

turn into Leads and Leads will turn into Customers, but it seems unlikely that Leads will turn into Service Contracts. You might sell a Service Contract to a particular Lead, just as you might buy a Car from a Supplier, but the name of the customer is just one of the pieces of information that goes into the Service Contract.

Create Teams

Teams represent the company hierarchy and any customer/reseller relationships. For instance, you may wish to set up an Accounting Team and a Sales Team so that you can easily send notifications to everyone in each department.

We will set up an Accounting Team and a Sales Team so that we can easily send notifications to everyone in that department.

To set up teams, start with Setup/Access/Manage Teams. Create a new team by clicking New, copy an existing team by selecting its checkbox in the leftmost column and clicking Copy, or edit an existing team by clicking the Edit button and letting the wizard guide you.

Create Groups

Groups control access permissions for different groups of users. You may use the default groups, or configure your own.

Teams and Groups are quite different. Your CEO may be in a team at the very top of the company hierarchy, but that does not mean that the admin would necessarily grant him the access permissions necessary to configure Agiloft.

In order for the Accounting Department to bill the customer, they need access to essential customer information. However, not all of this information is relevant and some may be so sensitive that we only want to share it with particular individuals. By creating an Accounting Group we give the members of this group access only to data that is relevant to them.

Setting up groups is just like setting up teams, except that you start with Setup/Access/Manage Groups.

Configure Email

Configure incoming email to automatically create/update records and setup outgoing email.

Select Setup/Email/Configure Email Server to configure the SMTP server for outbound email. If you have installed on a Linux machine, there is probably an SMTP server installed already and you can just leave the configuration as "localhost". This step is not necessary if you are using the 30 day free trial on our server

Select Setup/Email/Configure Inbound Email to configure inbound email. When you first visit this screen, you will probably see an empty table, simply click New to configure a new inbound email account. You can create multiple accounts for a single table and apply different rules to each of them. For example, you might assign a priority of Medium to records created from email received from one account and tell high value leads about another address, with records received from that one automatically given a priority of Urgent.

Create Workflows

Workflows automate the flow of information as records are changed in your business process, and the assignment of tasks to particular individuals/teams.

In this example, we are purchasing Leads, qualifying them, converting them into Customers, billing those Customer and converting them into Reference Customers, so it is natural to create workflow states such as New Lead, Qualified Lead, New Customer, Customer - sent bill, Customer - received payment, Reference Customer

Select Setup/Workflow, select the Lead table and create the necessary states as described in the [Workflow](#) section.

Create Business Rules

Rules monitor activity and alert the appropriate people when necessary.

When Accounting bills a customer, they should change the state from New Customer to Customer Sent Bill, and when the customer pays, their state should be changed again to Customer Received Payment. Manual editing of these records opens a liability gap for things to go wrong or be forgotten: accounting may neglect to bill the customer or the customer may forget to pay.

Business rules can catch these types of issues automatically, before they become problems. First, we create a rule that looks for New Customer records that have not been modified in more than seven days. Next, we set it to send an email to the Accounting group reminding them to bill the customer.

- Select Setup/Rules/New/, select the table you use to store customer info (usually the Company table)
- Select Rule Type/At selected time intervals/Condition/create a New Search
- Create a Search with a Simple Filter of: Status=New Customer and a Time Filter of Modification Date>=7 days old
- In the Save Search Options tab, save it with a descriptive name such as "New customers that have not been billed" and select Finish
- Select Schedule and in Apply Rule, specify that the search should run Weekly on Mondays

- Select Action and select Create an Email Action that emails the Accounting Team. See the instructions in Email and the following flash presentation for details. Implement other business rules in our example in a similar fashion.
- Follow up each sale with a support call to confirm that the customer is happy and possibly make a follow-on sale. Create a rule based on a Saved Search that checks for Customers that are over 2 days old where the field "Contacted by Support" set to No. This rule could email a Support Engineer to suggest calling the customer to check in, and setting Contacted by Support to Yes.

Assign tasks to individuals whenever possible, since tasks assigned to a team are no one person's responsibility.

- Convert happy customers into reference accounts If the customer says that they are very pleased with the product during the above support call, they can be converted to Reference Customer immediately. If not, a call can be scheduled for later.
- Remind new Customers that they can access an on-line dynamic FAQ 24/7 and get answers to questions directly via the web

The same rule that schedules the support call can also email the customer pointing them to a dynamic FAQ page. The support engineer can follow up by leading the customer to this page and answering any questions. The result is that the customer is encouraged to use the self-help facilities, but knows that a friendly voice is available for emergencies. This conserves the support team as a resource and keeps the customer happy.

Create User Accounts

You may either create individual user accounts from scratch or configure the system to use an LDAP or Active Directory server.

When you create user accounts, be sure to set their group and team memberships correctly.

Verify that the access permissions are correct by logging in as a member of each group and team. Perform a few of that user's tasks to verify that they will be correctly notified by business rules or workflow when needed. Temporarily set their email address to one of your own accounts for this purpose. To help prevent confusion, use a separate email account for each user that you are testing.

Import Existing Data

You can import existing customer data from Excel, ASCII or XML files.

Select the table that you are going to be importing the data into, then select the Import tab. Suppose that the file containing all the new leads is on your local machine in tab-delimited form and you are going to be importing Leads.

Select Leads/Import/Data Source, Local hard drive/File Name, Browse, Upload/Data Format/Tab-delimited text/Table,Leads/Fields.

The dialog that appears shows sample data from the first line of the selected input file and allows you to choose which column is mapped to which field.

The Import tab under Setup is used for importing entire KnowledgeBases in Agiloft format, for example if you were restoring from backup or moving it to a different machine.

See the API Reference for information on scripting to import data at regular intervals.

It is easy to write a script to import data from an external file into Agiloft at regular intervals, see the API Reference for more information.

Demonstrate the Working System

Gather feedback on the system and make adjustments in real time.

You should now be familiar enough with the system capabilities to translate user requirements into the appropriate tables, rules and workflows in real time. Have the employee users concentrate on describing their needed features, and demonstrate to them the ways that the system can be configured to meet that need. It is inadvisable to let the end-users who are unfamiliar with the capabilities specify the configuration directly.

Start Production Use

Deploy the new Agiloft system in parallel with the existing system for the first week. Adjust as necessary based on user feedback.

The most important thing you can do at this stage is to stay in close contact with the users. Do not assume that everything is perfect because they are not complaining. Often users will not mention problems because they assume that they cannot be resolved easily. With Agiloft, almost all user needs can be met with ease and grace! If your users *do* come up with something that really stumps you - just call us! We are here to help.

See Also:

[Stage 1](#)

[Initial Implementation](#)

[Implementation Stages](#)

[Concepts](#)

[Business Example](#)

STAFF USER GUIDE

STAFF INTERFACE

The Staff Interface is for internal company personnel to create and update records, interact with customers, manage their agenda, receive notifications and undertake other activities typical within a fast-paced business environment.

The screenshot displays the Staff Interface with a search bar at the top. A callout box labeled "Multiple search capabilities" points to the search bar. The interface includes a navigation menu on the left with options like Home, Setup, Communications, Last Opened, Tables, Calendar, Chat Status, and Messaging. The main content area shows a "Support Cases" section with a status message: "Status: 18 record(s) found, 1 pages Click for details...". Below this is a search bar with "CR ID" entered and "contains" selected. The search results are displayed in a table with columns for Edit, ID, Summary, Priority, and Status.

Edit	ID	Summary	Priority	Status
<input type="checkbox"/>	426	My printer is not working	Critical	Open
<input type="checkbox"/>	424	Customer's keyboard not functioning correctly	Low	Assigned
<input type="checkbox"/>	415	problem with equipment	Low	Closed
<input type="checkbox"/>	414	Repair Ticket	High	Closed
<input type="checkbox"/>	413	Hard Drive not accessible - BSOD	Low	In Repair
<input type="checkbox"/>	412	Printed pages have streaks	Medium	Assigned
<input type="checkbox"/>	406	SQL Server query	Critical	Open
<input type="checkbox"/>	405	Calculation with fields from the same record	Low	Closed
<input type="checkbox"/>	404	How do we login to the alternative end user portal?	Critical	Closed
<input type="checkbox"/>	403	How do we login to the mobile interface?	Low	Reopened
<input type="checkbox"/>	402	How do I request a license for Agiloft?	High	Reopened
<input type="checkbox"/>	382	Reports are incorrect in Peoplesoft	Low	Open
<input type="checkbox"/>	356	My printer is not working although it shows that it is ready	Medium	Closed

This chapter makes a quick overview of features available in Staff Interface. All of them will be discussed in more details further on in following chapters. For a hands-

on tutorial of other important Staff Interface tools, such as Search, Login Block, or Left-Hand Pane, please review the flash presentations [here](#).

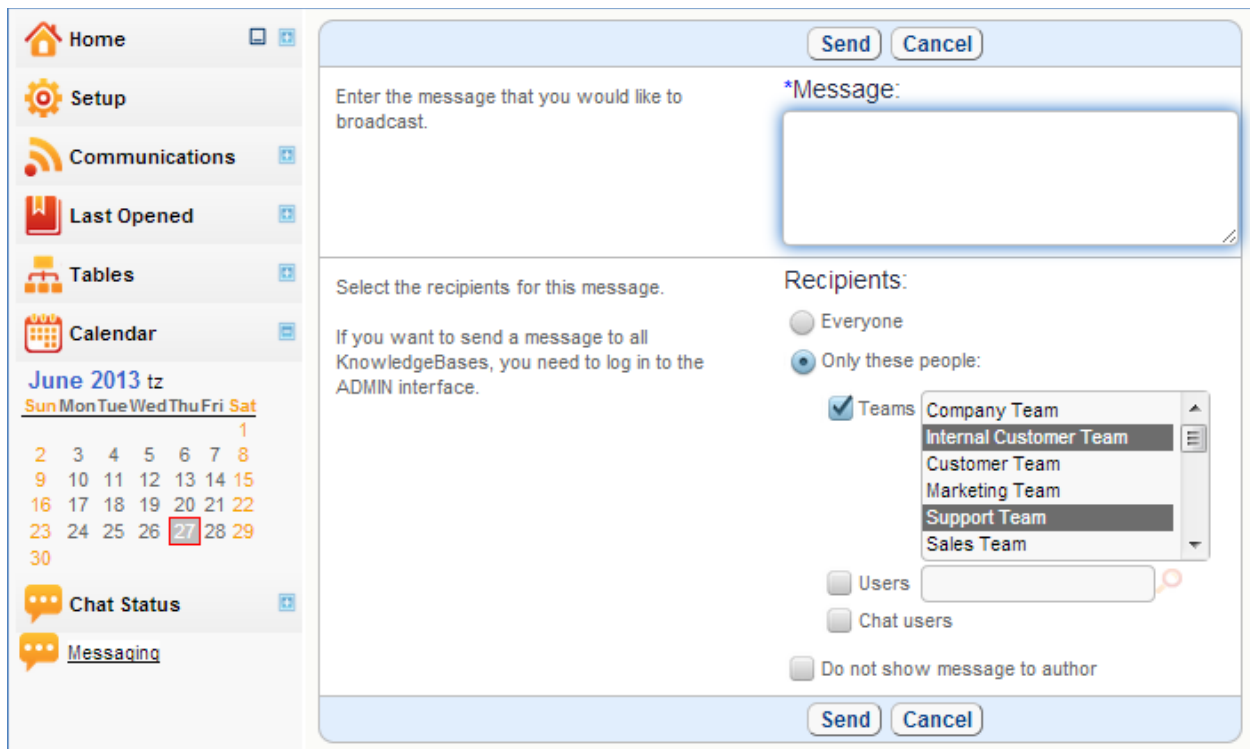
FULL TEXT SEARCH

The integrated Full Text Search (FTS) engine provides a fast natural language search capability with synonym support, to assist staff and end users in finding the information they need, in the database and in attached files

See more here: [Search Overview](#)

INSTANT MESSAGING

The instant messaging feature can be used to send messages to specific individuals or teams in real time.



MAIN TOOLBAR

The Main Toolbar provides tabs corresponding to the tables that you can access.

To create a new record in the Vertical Toolbar Layout, first choose the table on the Left Pane (under Tables Menu) and then click on the New in the table's Toolbar.

In the example below, the user is about to create a new Contact record using a data entry form.

Contacts
 Status: 35 record(s) found, 2 pages [Click for details...](#)
 Page: [Page 1] [Next](#) [Last](#) Go to page [Go](#)

Views

Quota String Context Active Only [More Options](#)

	Name	Contact Table	Company	Direct Phone
<input type="checkbox"/>	413 John Doe	Contact	Agiloft	5554334:434
<input type="checkbox"/>	412 Admin	Employee		
<input type="checkbox"/>	411 Guest End User	End User		
<input type="checkbox"/>	410 Jane Repairgal	Employee		888-555-1245
<input type="checkbox"/>	409 Jack Repairguy	Employee	Agiloft	333-123-4222
<input type="checkbox"/>	407 Repair Customer	End User	IBM	808-555-1234
<input type="checkbox"/>	406 Don Wilson	End User	State of California	
<input type="checkbox"/>	405 Jerry Brown	Contact	State of California	222-333-4444

TABLE VIEW

Support Cases
 Status: 18 record(s) found, 1 pages. [Click here](#) to re-count records.

Views

Search: equals, =

Refine String Context Active Only [More Options](#)

<input type="checkbox"/>	Edit	ID ↓	Summary	Priority	Status
<input type="checkbox"/>		426	My printer is not working	Critical	Open
<input type="checkbox"/>		424	Customer's keyboard not functioning correctly	Low	Assigned
<input type="checkbox"/>		415	problem with equipment	Low	Closed
<input type="checkbox"/>		414	Repair Ticket	High	Closed
<input type="checkbox"/>		413	Hard Drive not accessible - BSOD	Low	In Repair

A toolbar is displayed at the top of the table following a status line that shows the number of records and, if applicable, any search criteria.

At the top of each column in the table is the name of the field. You can sort the table by clicking on these field names, see [Sorting](#) for more details.

OPERATIONS ON THE TABLE: NEW

Simply click New to create another record.

One click record creation

Contacts
Status: 35 record(s) found, 2 pages.
Page: [Page 1] [Next](#) [Last](#) Go to page [Go](#)

New Mass Edit Delete Actions Views Search

New Contact
New Employee
New End User
Help

	Name	Contact Table	Company	Direct Phone
	Doe	Contact	Agiloft	5554334:434
	n	Employee		
<input type="checkbox"/>	411	Guest End User	End User	
<input type="checkbox"/>	410	Jane Repairgal	Employee	888-555-1245
<input type="checkbox"/>	409	Jack Repairguy	Employee Agiloft	333-123-4222
<input type="checkbox"/>	407	Repair Customer	End User IBM	808-555-1234

New record dialog layout and content obviously depend on the table for which you are creating the record, but they also depend on the permissions that have been given to the particular user.

Common Features

Explore some of the common features each New Record Wizard shares:

Create another record once you are done with the current one.

Helpdesk Case

Save Cancel Details Working Time Related Records History << >>

Save & New

ID 64

Status Open Priority Choose one

Active Action:
 email customer acknowledgement

Assigned Team Support Team Assigned Person

Some information is already filled in.

User Information

Submitter Name	Marva Lazinski	Submitter Login	salesmgr
Submitter Phone	333-444-5555	Submitter Email	example10@example.com
Submitter Cell Phone		Submitter Department	Sales

Case Information

*Type of Problem Choose one Subtype of Problem Choose one

Related to an Asset Yes No

*Summary

Problem Description

Required fields are marked.

Solution

Edit HTML plain text

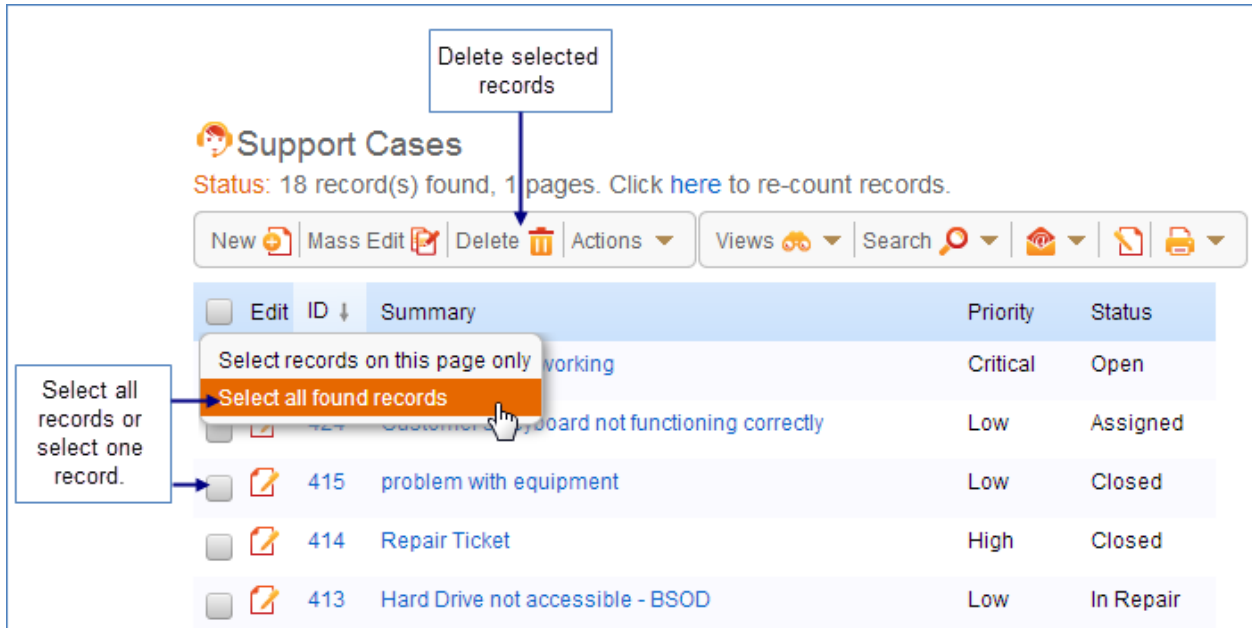
Enter text in plain text or HTML formats

Use the Lookup Icon to find and link other records to the current one.

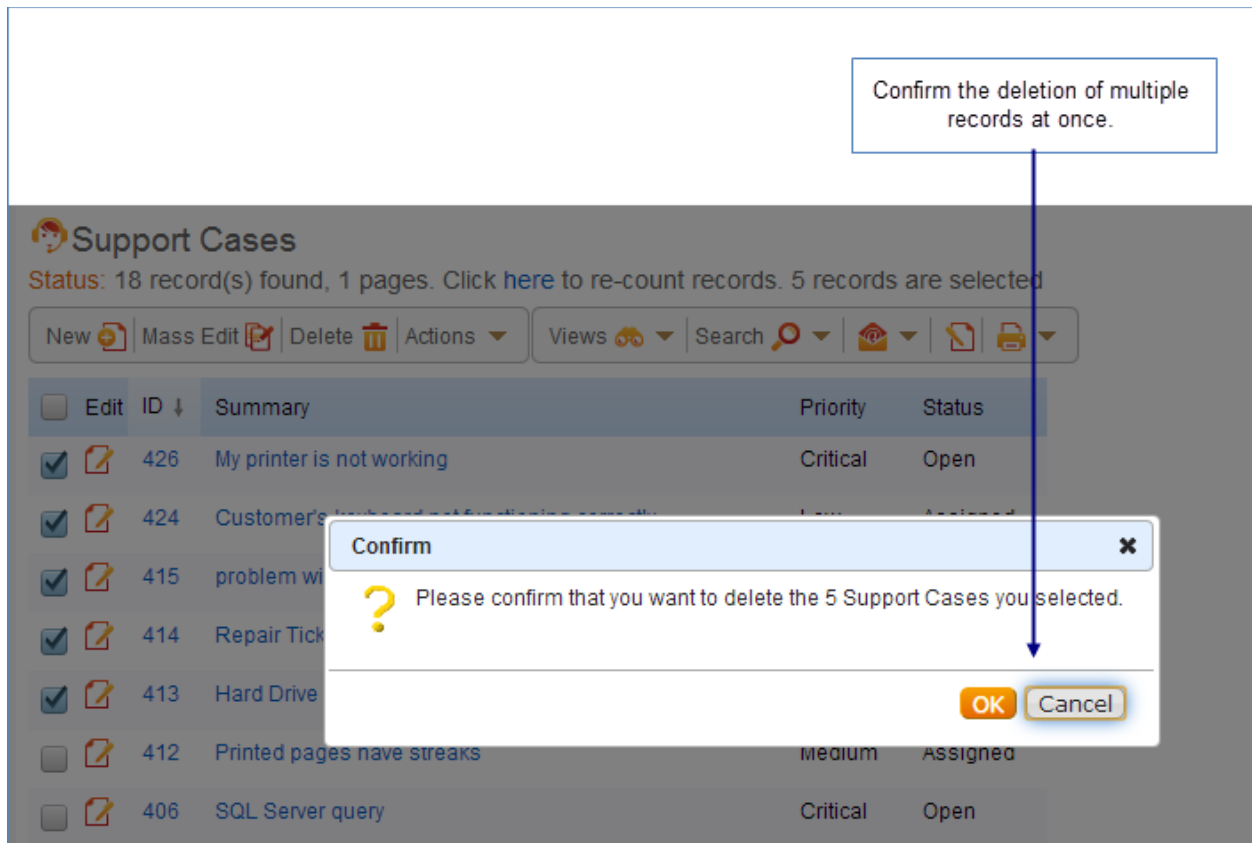
OPERATIONS ON THE TABLE: DELETE

You can delete records or one record by selecting a checkbox in the very left of the Table View.

You can even delete all records by using the "Select All" checkbox.




You will be asked to confirm your decision:



A progress window appears during record deletion. It shows the progress percentage and elapsed time.

Status message will be displayed to give you a feedback on your actions.

The Status message displays the number of deleted records upon success. It displays a warning message upon failure.





Support Cases

Status: 1 Support Case records successfully deleted. 18 record(s) found, 1 pages. Click [here](#) to re-count records.

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


	Edit	ID ↓	Summary	Priority	Status
<input type="checkbox"/>		426	My printer is not working	Critical	Open
<input type="checkbox"/>		424	Customer's keyboard not functioning correctly	Low	Assigned
<input type="checkbox"/>		415	problem with equipment	Low	Closed
<input type="checkbox"/>		414	Repair Ticket	High	Closed

OPERATIONS ON THE TABLE: EDITING

To edit a record, click on the  icon; to view the record, click on the  icon. Alternatively, you can use a table view that makes fields in the record 'active', meaning that clicking on that field will open the record for Edit or for View.

Note that when editing the record you have an option to jump directly to the next record

Save and open the next record




Support Case

Save | Cancel | Details | Working | Related Records | Contract | Time | Emails | History

ID	426	Status	Open	Assign to Me
Assigned Team	Support Team	Assigned Person		
Alternative Assigned Team or Person	Christopher Caldwell			

Contact Information

*Customer Name:  User Company:

Customer Phone: Ext. Cell Phone:

Sometimes staff technicians would like to be able to open up the next assigned record when they close a given record without stepping on each other's toes and getting a message that the record they have just selected to edit is locked by another staff user.

This has been made easier by the Save & New and Save & Next buttons from the drop-down on the Save button. When it is clicked, it saves the current record and instead of returning to the table view, it selects the next record displayed below the record that is being saved and opens it for editing. Cancel & New and Cancel & Next have also been added as options under the Cancel button drop-down.

If that record is currently locked by another user, or has just been deleted by another user, or no longer meets the search criteria (because it has been closed by another user or assigned to another user), it moves to the next record available below that.

This is ideally intended to be used with the Inbox or My Assigned search, so a staff user may run that search, pick the first ticket in the list and work on it, and then just proceed through the remaining assigned tickets by using the Finish and Next buttons.

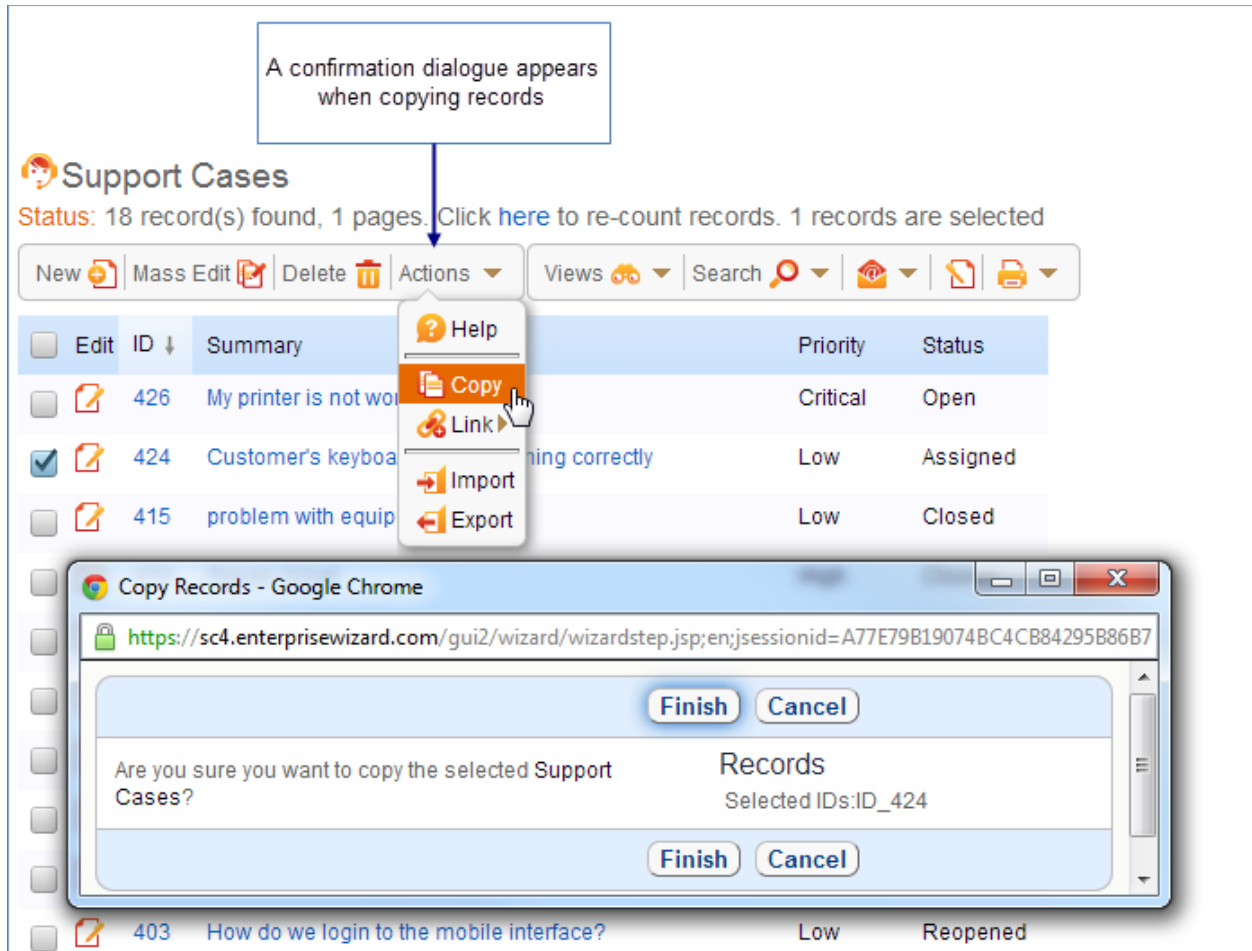
OPERATIONS ON THE TABLE: MASS EDIT

A Mass Edit operation can modify several records and fields simultaneously.

More here: [Mass Edit Overview](#)

OPERATIONS ON THE TABLE: COPY

Copying records can be quite handy if you, for example, just created a support case for one contact, and want to create another on their behalf, then you can copy the first one and edit the one copy operation creates for you. Please note, that everything about that original ticket will be copied, including the history.



OPERATIONS ON THE TABLE: LINK

Sometimes you need to indicate that specific items are actually connected based on some criterion. It is possible to link records by selecting their checkboxes and using Link operation. Once linked, records will be marked with their peer's Ids. You might want to modify your view to add Links field in.

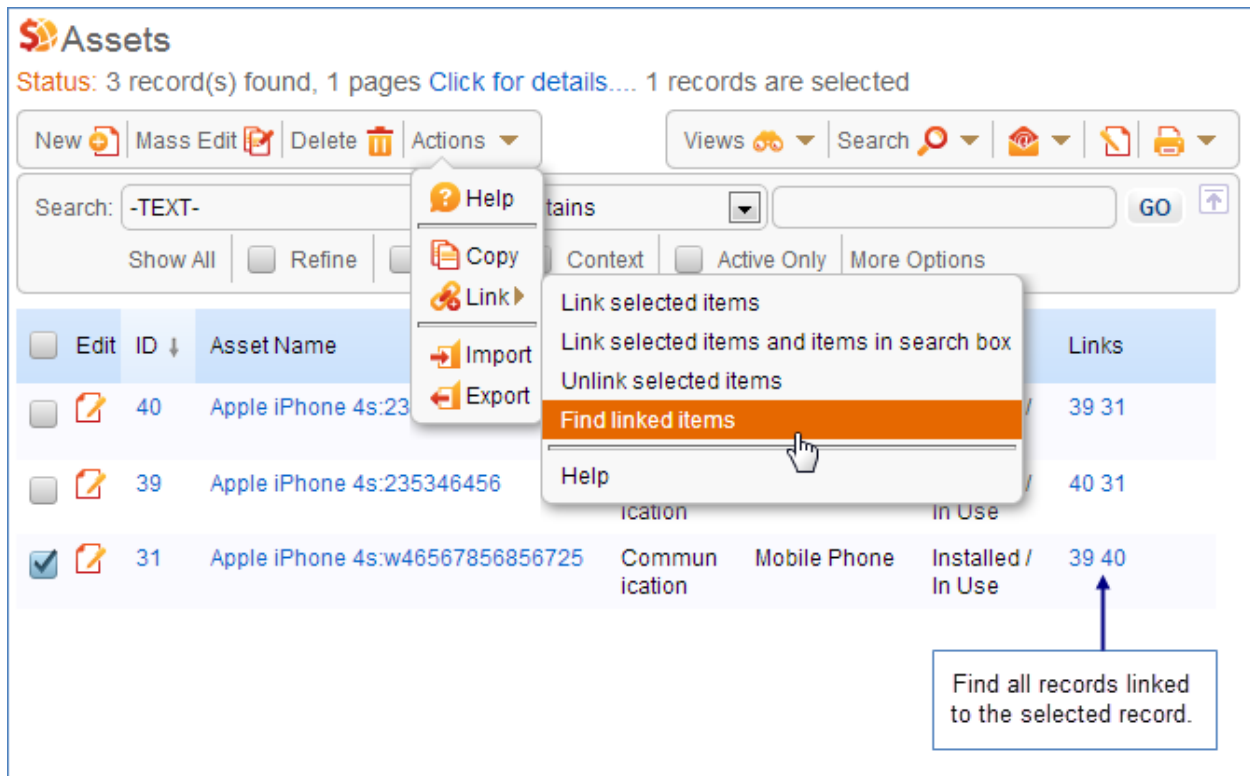
The screenshot shows the 'Assets' management interface. At the top, a status message reads: "Status: Items [31, 39, 40] have been linked". Below this, a search bar contains "-TEXT-" and a table lists assets. A dropdown menu is open over the 'Link' button, with the option "Link selected items and items in search box" highlighted. Annotations include:

- A box pointing to the status message: "Status message indicates which records have been linked".
- A box pointing to the 'Link' dropdown menu: "Link records by selection or by entering a known ID".
- A box pointing to the selection checkboxes in the table: "Select one or many records to link".

Edit	ID ↓	Asset Name	Status
<input checked="" type="checkbox"/>	40	Apple iPhone 4s:23	Installed / In Use
<input checked="" type="checkbox"/>	39	Apple iPhone 4s:235346456	Installed / In Use
<input type="checkbox"/>	37	Polycom 4690IP:234239347982	Communication Conference Phone Installed / In Use
<input type="checkbox"/>	31	Apple iPhone 4s:w46567856856725	Communication Mobile Phone Installed / In Use
<input type="checkbox"/>	30	Acer AcerPower S280:2436457568679	Computer Desktop Installed / In Use

"Links" field has all Ids of the items current record is linked with

You can always bring up all linked records by selecting current one and clicking "Find linked items"



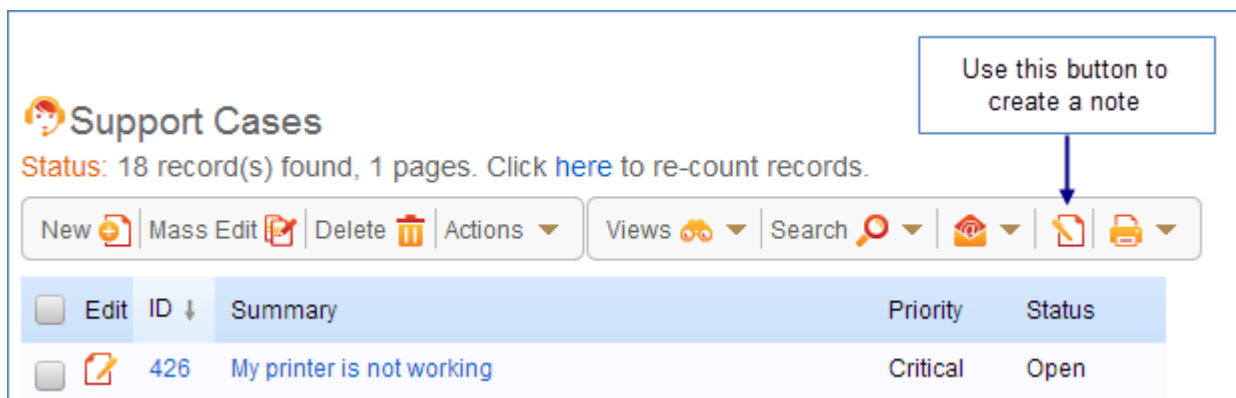
OPERATIONS ON THE TABLE: EMAIL

The user can select one or more records in a particular table and send them by email using Email Wizard.

Read more on Email here: [Email_Wizard](#)

OPERATIONS ON THE TABLE: NOTE

Both notes and emails can be stored in communications subtable, which is automatically created when you create a new table. Notes, once created, can be displayed within a record. A user adds a note by clicking on the appropriate icon to the right of the tabs when the record is open:



or in the Table View for chosen record or number of records


While creating the note you can use standard HTML editor to format your message nicely. You can narrow permissions for this note by either choosing Internal or Published type




Like in the standard Email wizard you can insert variables and an attachment.

Once note is created it can be viewed in Related Info tab (View Record Icon/Related Info). Whole communication history will be displayed in the communication table:

Communications

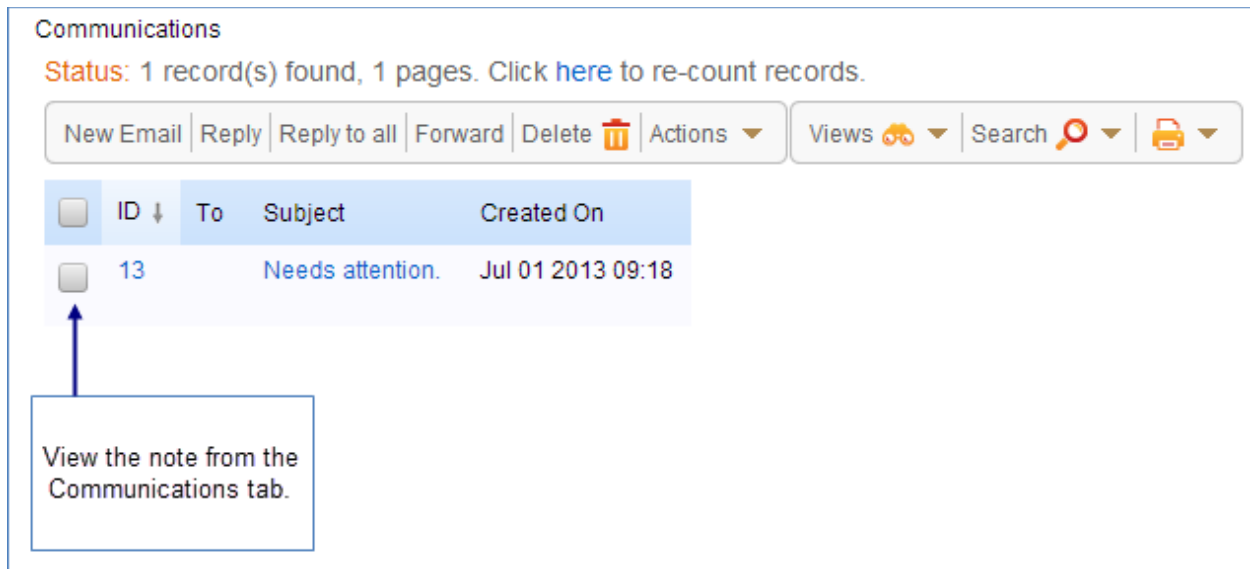
Status: 1 record(s) found, 1 pages. Click [here](#) to re-count records.

New Email | Reply | Reply to all | Forward | Delete  | Actions ▾

Views  ▾ | Search  ▾ |  ▾

<input type="checkbox"/>	ID ↓	To	Subject	Created On
<input type="checkbox"/>	13		Needs attention.	Jul 01 2013 09:18

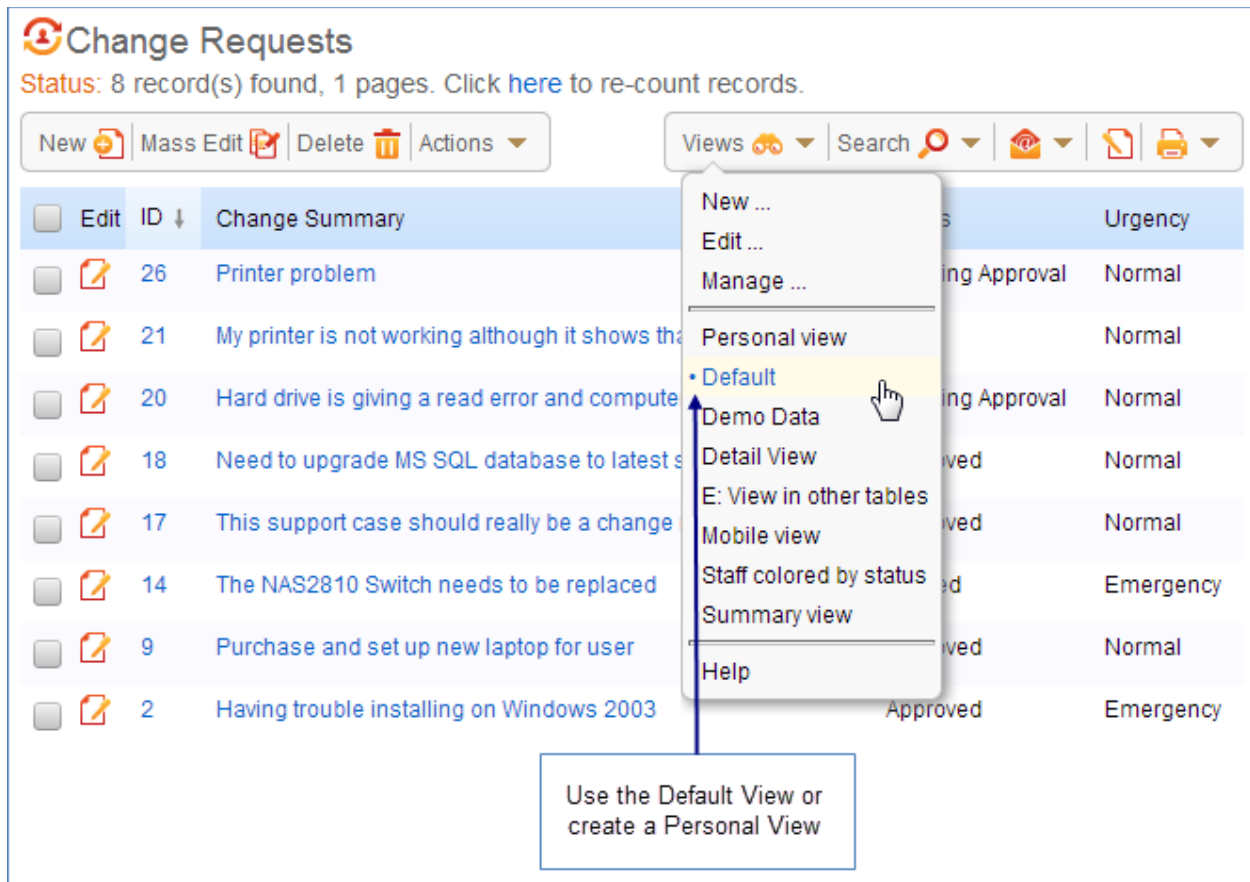
View the note from the Communications tab.



OPERATIONS ON THE TABLE: VIEWS

Use **Set View** to change what fields are shown and how many records are displayed per page. All Views are displayed by clicking Set View/Manage.

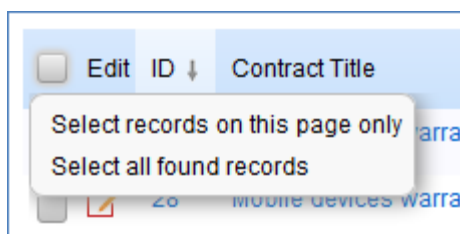
There you can find predefined views like "Summary view" and "Personal view".



Please note that "Personal view" is only visible and available to you and no other user, so it is safe to change whatever you feel like changing.

You can also choose to create a view that will be used by some other teams. For example, you can define a view that will be default for you and selected teams (See Set View/Manage/Edit View/Apply Tab)

Use the checkbox on the left of the record to select, or unselect, records for **Delete**, **Copy**, **Link**, **Mass Edit** and **Email**. Options have been added to allow the selection of all records or only those displayed on a page. When the "select all" box is clicked, a popup appears allowing the user to select just the records visible on the page, or all records found by the filter.



See more on Views: [Views Overview](#)

OPERATIONS ON THE TABLE: SORTING

The rows in a table are sorted by the Record ID number in descending order by default but you can sort them on any field by clicking on that Field Name.

The currently selected field is marked with ↓ or ↑ next to the Field Name for descending or ascending order. Clicking the field name or the ↓ (↑) icon again reverses the sort order.

OPERATIONS ON THE TABLE: PRINTING

Mousing over the printer icon from the table view of records, you can select New to create a print format which may contain HTML text with specific fields inserted wherever you like.

It is also possible to print records to named Word/PDF templates. From the table view of records, mouse over the print icon. The method of printing is for the user to select a record or records in a table, then to mouse over the printer icon and select the word template. This will cause the browser to either open the document in word or ask the user to save it, depending on his browser settings. All standard options are available: templates may be created, modified, and deleted. A Word template may be downloaded, corrected and uploaded.

The screenshot shows a web application interface for 'Change Requests'. At the top, it displays 'Status: 8 record(s) found, 1 pages. Click here to re-count records.' Below this is a toolbar with buttons for 'New', 'Mass Edit', 'Delete', 'Actions', 'Views', 'Search', and a printer icon. A dropdown menu is open over the printer icon, listing options: 'Table View', 'All Fields', 'New HTML Template', 'New Word/PDF Template', 'Manage...', and 'Help'. The table below has columns for 'Edit', 'ID', 'Change Summary', and 'Un'. The 'ID' column is sorted in descending order, indicated by a downward arrow. The table contains 8 rows of records with various details like 'Printer problem', 'My printer is not working...', 'Hard drive is giving a read error...', 'Need to upgrade MS SQL database...', 'This support case should really be a change request', 'The NAS2810 Switch needs to be replaced', 'Purchase and set up new laptop for user', and 'Having trouble installing on Windows 2003'.

<input type="checkbox"/>	Edit	ID ↓	Change Summary	Un
<input type="checkbox"/>		26	Printer problem	No
<input type="checkbox"/>		21	My printer is not working although it shows that it is rea	No
<input type="checkbox"/>		20	Hard drive is giving a read error and computer is inacc	No
<input type="checkbox"/>		18	Need to upgrade MS SQL database to latest service pack	Approved
<input type="checkbox"/>		17	This support case should really be a change request	Approved
<input type="checkbox"/>		14	The NAS2810 Switch needs to be replaced	Closed
<input type="checkbox"/>		9	Purchase and set up new laptop for user	Approved
<input type="checkbox"/>		2	Having trouble installing on Windows 2003	Approved

Everywhere in the interface you will find Help Icon that will bring help topic connected to the particular dialog. Use it when you want come clarification or to see some examples.

The screenshot shows the 'Contracts' interface. At the top, it says 'Contracts' and 'Status: 12 record(s) found, 1 pages Click for details...'. Below this is a table with columns: Edit, ID, Contract Title, and Status. The table contains several rows of contract records. A context menu is open over the table, showing options: Expand Search Options, Help, New ..., Edit ..., Manage ..., Show All, My Drafts, and P: All Customer contracts. A callout box points to the 'Help' option with the text 'Help in Table View brings up a corresponding help topic'.

Edit	ID	Contract Title	Status
<input type="checkbox"/>	29	Mobile devices warranty support	A
<input type="checkbox"/>	28	Mobile devices warranty support	A
<input type="checkbox"/>	26	Oracle Maintenance Support Contract	IE
<input type="checkbox"/>	23	Mobile devices warranty support	A
<input type="checkbox"/>	22	Apple desktop warranty support	A

Search

Overview

The search functionality provides the real power behind rules and workflow and is used interactively to find particular records.

- Automatic full text indexing provides fast searching through large tables of records *and* their attached files for whole words, phrases or strings within a word (click the "More Options" hotlink).

CALENDAR

The Calendar allows you to schedule events of a selected type, for instance meetings, phone conferences, vacations or announcements.

See more about Calendar here: [Calendar Overview](#)

CHAT

The integrated chat facility allows staff to communicate with multiple customers concurrently, reducing support costs and telephone charges.

See more about Chat here: [Chat Overview](#)

HOTKEYS

Agiloft has some keyboard hotkeys available to perform actions within the application from the staff interface and the end-user interface.

Hotkeys currently include:

End User Interface:

- Ctrl+Alt+N (New Support Case)
- Alt+X (Logout)

Staff Interface:

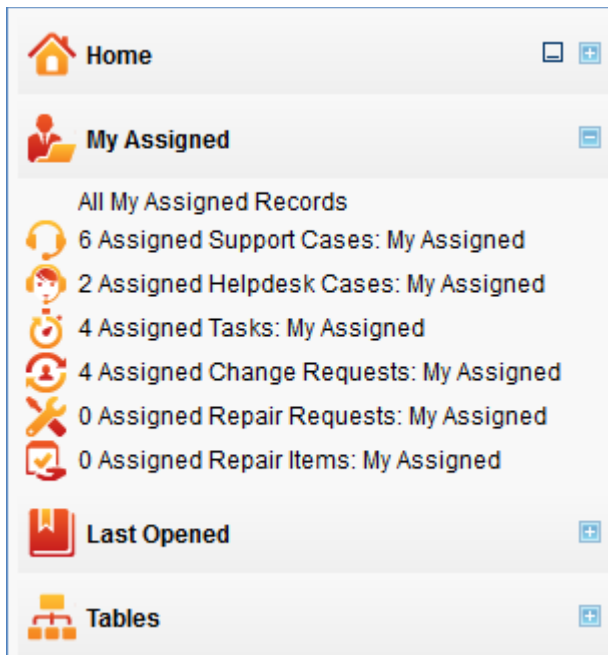
- Ctrl+Alt+N (Create Note)
- Ctrl+Alt+L (Create New Mail Record)
- Ctrl+Alt+{SPACE} (Displays current field's pop-up)
- Ctrl+Alt+Right arrow (Moves to the next form tab)
- Ctrl+Alt+Left arrow (Moves to the previous form tab to the left)
- Alt+{F1} (Displays Help)

Both End-User and Staff Interface:

- Ctrl+{ENTER} (Finish)
- Ctrl + Alt+{ENTER} (Finish and New)
- Shift+{ESC} (Cancel Edit)

INBOX

Inbox can be found either on the Left Pane or at the top right corner of the Staff interface (depends on your Layout preferences)



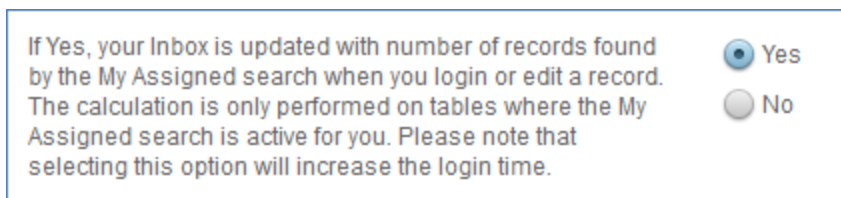
To a particular user the Inbox shows entries for tables that:

1. The user is allowed to view
2. Have search with name "My Assigned"
3. Have the My Assigned search both accessible and active for the user's group(s)

When one clicks the item in the Inbox - this search is executed.

The "All My Assigned Records" item runs the "My Assigned" searches for all available tables and shows only those that have not-empty results.

One can configure the Inbox to show the number of matching records for every participating table via Home -> Preferences.



Each individual "My Assigned" search can be modified in any way.

See Also:

[Flash Tutorials](#)



HOME PAGE

Home page is your personal space where you define your preferences, change your profile and set up your Dashboard.

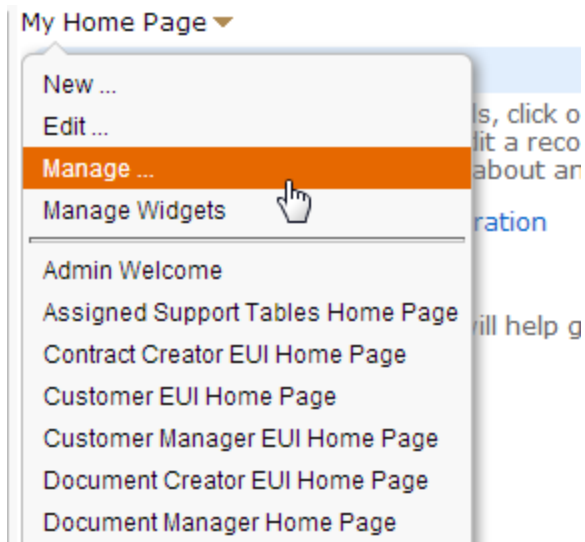
A customizable dashboard-style home page for staff users gives users a standard "home page" from which they can launch all of the actions of interest to them. It acts as a "Home" page in that it provides access to appropriate information immediately upon log-in (e.g. list of open assigned Incidents) and displays relevant charts and reports. The staff user can see high-level overviews of actionable information and quickly assess needed action. Internal hotlinks can be added to the home page to give direct create access to records in any table, views of any saved searches on any tables, and so on.

The Home page may be set in a user's preferences as his default starting page. The Home tab automatically displays the default Home page template for the user's Primary team unless a different default template has been selected previously by the user under the Preferences tab.

The home page is based on widgets and the following kinds of widgets are supported: charts, dashboards, internal hotlinks, table view with Saved Search applied, google gadgets, embedded webpage or image, and text.

New Group permissions on the General permissions tab controls which users can create their own home pages and their own widgets, and which can only select from among home pages and/or widgets developed by admins and made available to them.

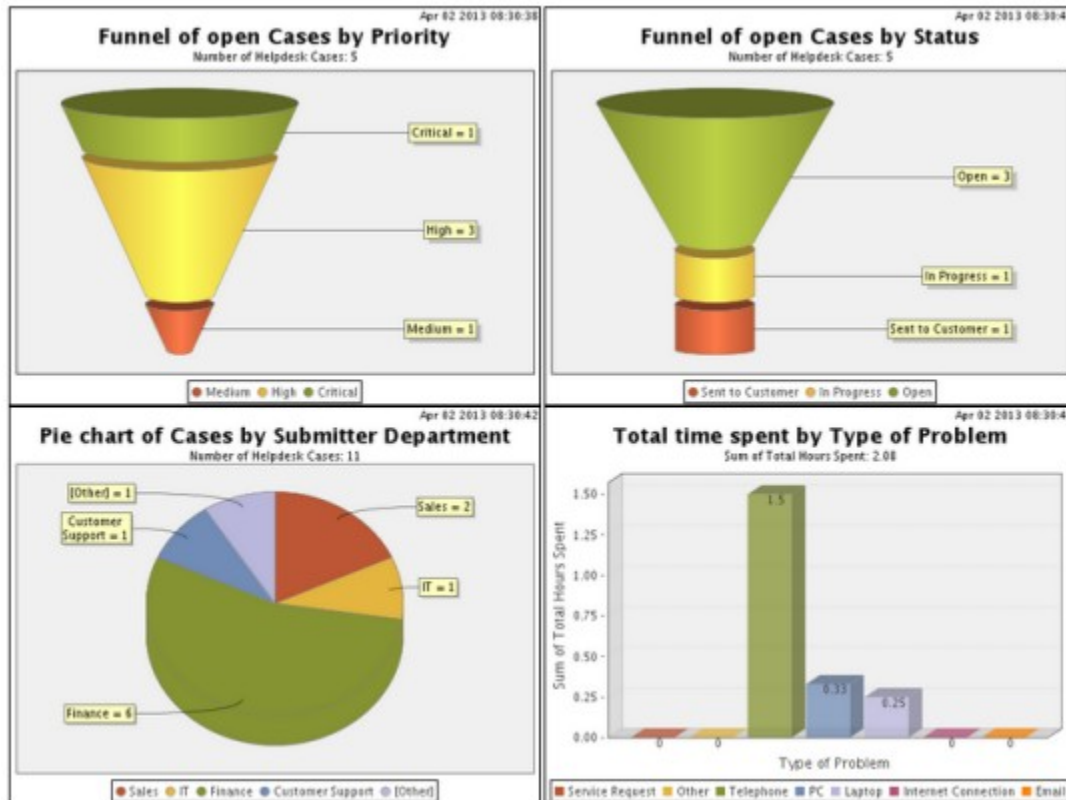
A variety of default home pages can be created by admins for different sets of users and applied as their default.



A flash tutorial showing how to create your own staff home pages will be available [here](#).

DASHBOARDS

Dashboard functionality allows any number of reports from any selection of tables to be combined into a single view. This is highly valued by people who want to get an overview of operations at a single glance.










OVERVIEW

- Location – Dashboards are accessed at Home/Dashboards
- Any number of different dashboards can be created

Dashboards

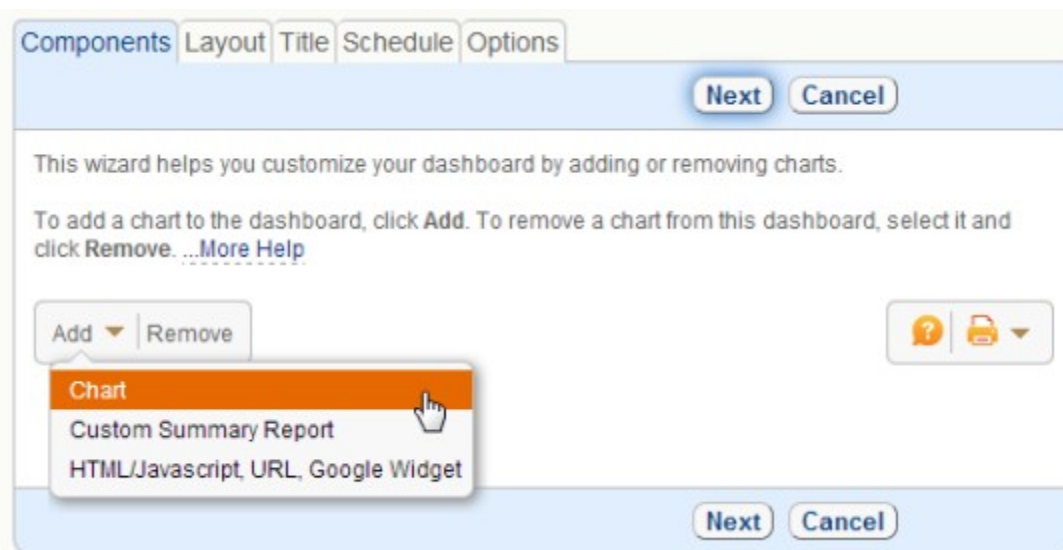
Status: 3 record(s) found, 1 pages. Click [here](#) to re-count records.

New		Delete	Views   		
Edit	View	ID ↓	Title	description	
		306	Sales: Opportunities Dashboard	Four charts centered on open opportunities.	
		305	Helpdesk Cases Overview	Charts showing open cases by status and priority and time spent by type of problem	
		304	Support Cases Overview	Shows trend of cases over past year, numbers created and closed, and time spent to close	

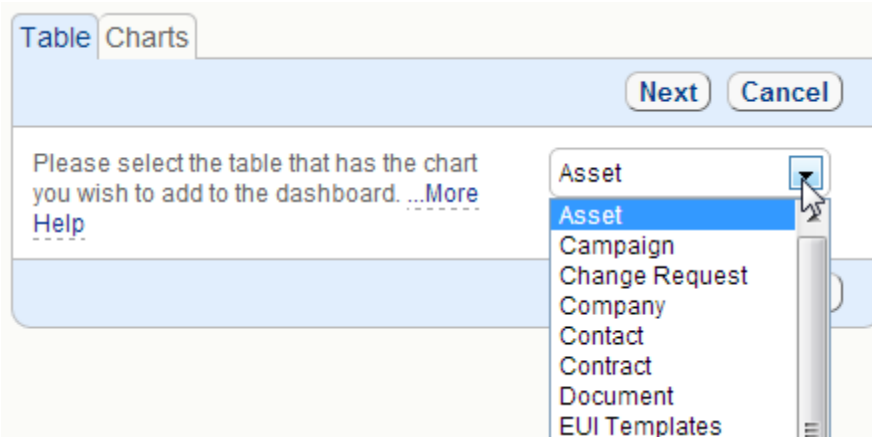
- Viewing the Dashboard – To run the dashboard reports and view them, click on the lookup icon
- You can drill down into the reports once they are on your screen by double-clicking on the chart
- Distribution – Dashboards can be automatically distributed via email in the same way as reports.

CREATING DASHBOARDS

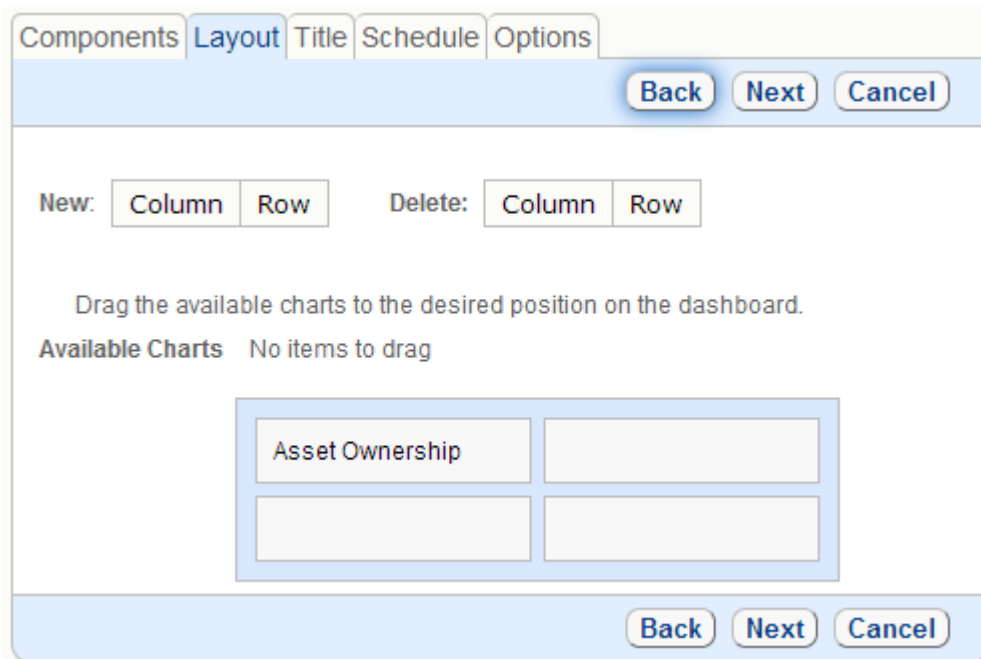
- Click New to bring up the wizard with a blank dashboard



- Click Add to invoke a pop-up wizard to add (pre-existing) charts. First select the table, then the chart in that table



- Layout – Controls where the charts are placed in the dashboard



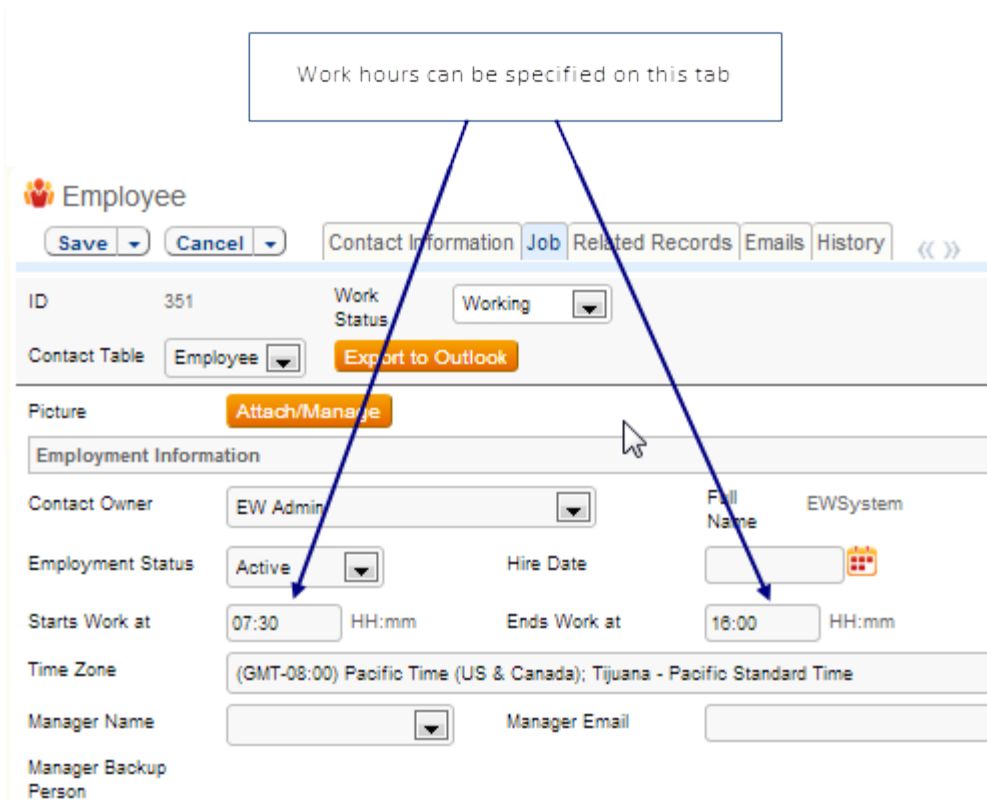
- Title – Provides a title and description
- Schedule – Controls how often the dashboard is automatically generated and written to the file system or emailed
- Options – Controls who can see the dashboard and whether the associated report should be generated together with the chart.

PROFILE

Profile displays your Employee table record. You need to specify your email address here, so system can send you notifications, reminders, reports etc. That's the place where you define your password and login name. Please note that depending on your permissions you can see more or less fields/tabs than in our example.

Text and field layouts depend on what has been configured for the Employee table

The screenshot displays the 'Employee' form interface. At the top, there are tabs for 'Contact Information', 'Job', 'Related Records', 'Emails', and 'History'. The 'Contact Information' tab is active. Below the tabs, the form is divided into sections: 'Contact Information' and 'User Information'. The 'Contact Information' section includes fields for First Name (Agloft), Last Name (Employee), Title, Department (Customer Support), Email, Direct Phone (415-232-4444), Ext., Cell Phone, and Email Pager. The 'User Information' section includes fields for Login (ewsystem), Password (Change Password...), Primary Team (Admin Team), Groups (admin, adminimport), LDAP User (No), DN, and User time zone (Choose one). A note is present next to the Teams field, stating: 'Note: Please include the Primary Team value in the Teams field. Admin Team, Agloft Implementation Team, Change Approval Team, Contract Manager Team, Document Approval Team, Professional Services Team, Sales Manager Team, Support Manager Team, System Admin Team'. The form also features 'Save' and 'Cancel' buttons at the top and bottom.



PREFERENCES

This tab allows you to customize behavior of some system wide features like a popup message every time you're closing the wizard, or whether you prefer to be forwarded to your Cases table or Dashboard once you're logged in.

Preferences ADA/Mobile Left Pane

Next Finish Cancel

If Yes, then you are prompted to confirm when you click Cancel to exit a wizard. Yes No

If Yes, then a new window is opened when you edit or create a new record. Yes No

If Yes, then a new window is opened when you view a record. Yes No

If Yes, then a new window is opened when you perform search for all tables. Yes No

If Yes, then any above options that are set to open in a new window will instead open in a new browser tab.
Note: Setting this option to Yes does not work on all browsers and if you do set it to Yes, you must also set your browser preferences to open in a new tab. In FireFox, select "Tools/Options/Tabs/Open new windows in a new tab instead".

These options specify what table you are taken to on login

Default table Dashboards

Default search Show All

Remember last search

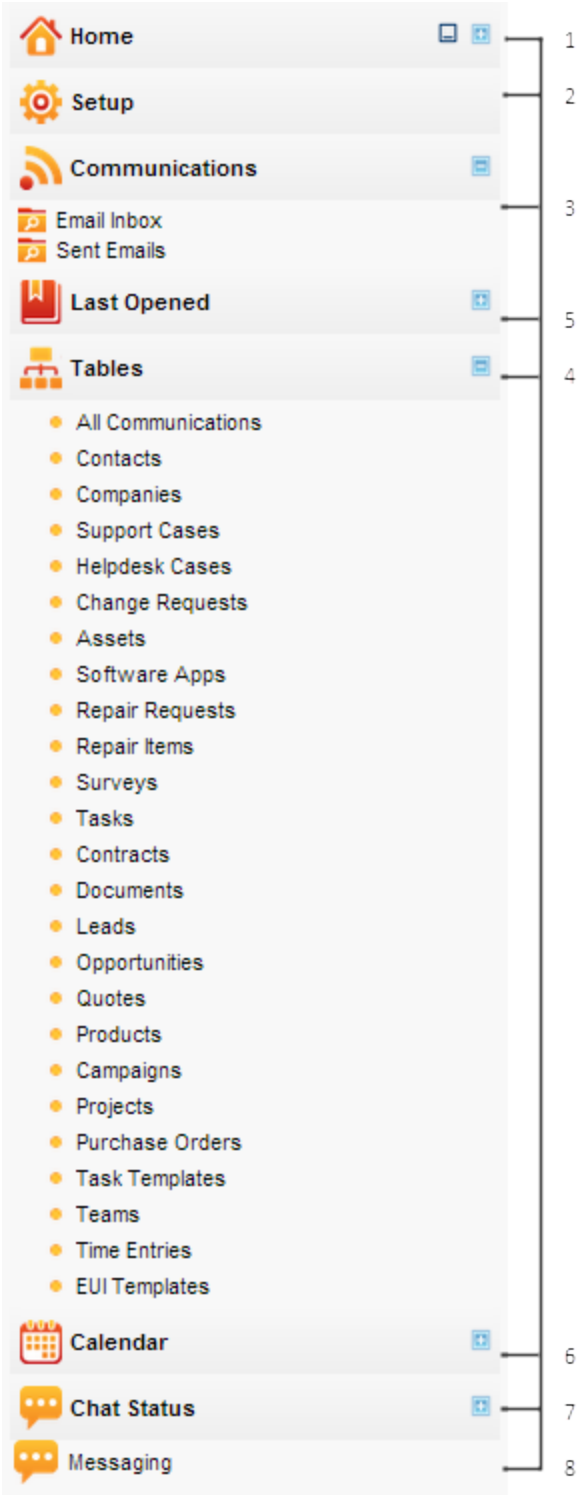
Last state

Last selected or default home page

VERTICAL TOOLBAR SETUP

In addition to the old Horizontal Toolbar a new vertical toolbar interface is now available. It is enabled by editing your default Look and Feel scheme (or creating a new one) with the 'Vertical Toolbar' option set on the first page of the Look and Feel scheme wizard.

Each element of the left pane may be collapsed and expanded by clicking on the main headings or using the minimize and maximize icons, and the whole left pane may be minimized by clicking the minimize icon on the Home tab. Its width can also be resized by dragging it with a mouse.



The Vertical Toolbar contains these main elements:

1. Home - this brings up the new Staff Home page. It contains the previous sub-options for Preferences, My Profile and Dashboards.

2. Setup - brings up a newly designed Setup page with links and description of all Setup options such as Tables, Rules, Access, System etc. Only Admin users see this tab.
3. Inbox - contains links to the global Inbox (All My Assigned Records), and separate inboxes for each table. Counting of the Inbox records can be enabled in user preferences (under Home/Preferences) so that the user can see in the Left Pane how many records are assigned to him in each table.
4. Tables - this brings up the vertical arrangement of the main tabs previously shown in the horizontal toolbar. The Table menu options are improved. Now a user may add his favorite Saved Searches to be displayed just below the table label for one-click access. Searches can be added to the left pane by clicking on Saved Searches/Manage and clicking the column to add the search to the left pane. Standard options like 'Charts/Reports', 'Import', 'Export' are also available under the table label. A 'Setup' menu option is added and visible to users with admin privileges so they can go directly to the setup table wizard from the left pane.

If there are subtables and the user clicks on the subtable link in the Left Toolbar then the records in the subtable are loaded. Also, the 'Setup' link below the table shows not the root table, but the corresponding subtable's setup.

5. Last Opened shows the 5 last opened records. It displays the first 25 characters of the field defined as the 'Summary' field for the table. Example: if the 'Company Name' field in the Companies table has the option 'Summary'=Yes then the Company Name value will be shown in the left Pane for the Last Opened company record. Mousing over shows the full Summary field. The icon for the table is displayed to the left of the summary field, so it is helpful to choose icons that identify the type of record.
6. Calendar - Shows the same Calendar display that was previously available on the left pane.
7. Chat status - This is the same as on the previous left pane.
8. Messaging - This is the same option that was previously available at the end of the horizontal toolbar for sending broadcast messages.

SEARCH


OVERVIEW





The search functionality provides the real power behind rules and workflow and is used interactively to find particular records.





- Automatic full text indexing provides fast searching through large tables of records *and* their attached files for whole words, phrases or strings within a word (click the "More Options" hotlink).
- Advanced search enables you to search individual fields for specific values or variables (such as \$my_name), relative date and times, calendar criteria (such as last month) or by when and how a field was last changed
- You can search for records that **ever** met the search criteria (historical search)
- You can search for synonyms and display the context within which the search term was found in the results.
- The Quick search box enables fast ad-hoc queries.
- Saved searches are available from drop-down for repetitive queries.
- You can configure the system to automatically take you to a particular table and execute a search when you login (click /Home/Profile for details)
- Pop-up searches allow you to find and import other records while editing a record

TABLE SEARCH BLOCK

The area under the main table tool bar is called a Table Search Block and is shown below. It can be hidden to save screen space by clicking on the Search icon. Click on it again to restore the Search Block.


 Support Cases
 Status: 18 record(s) found, 1 pages. [Click here](#) to re-count records.

New  Mass Edit  Delete  Actions 

Views  Search   

Search: contains GO

Show All Refine String Context Active Only [More Options](#)

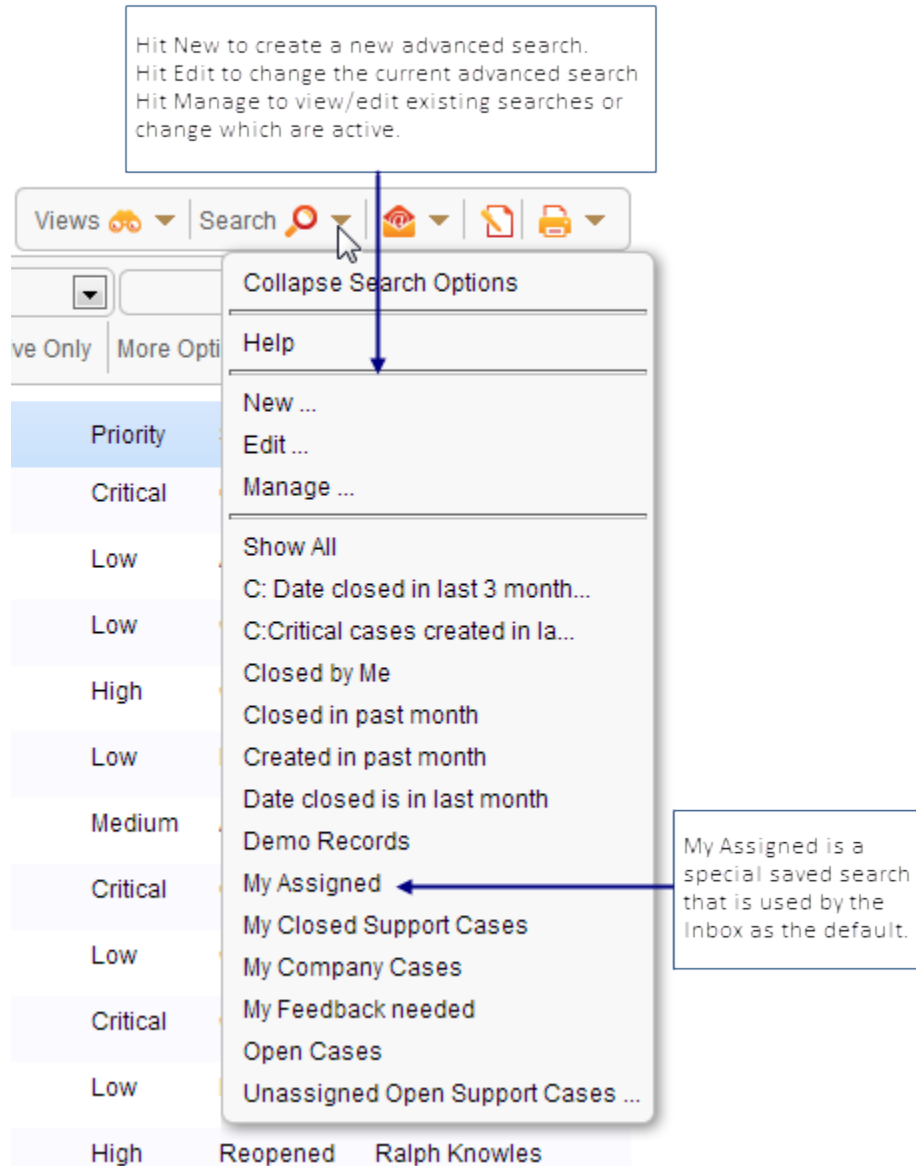
<input type="checkbox"/>	Edit	ID ↓	Summary	Priority	Status	Assigned Person
<input type="checkbox"/>		426	My printer is not working	Critical	Open	

- Enter text to find all records submitted by a user, or numbers to find records by ID. For example 10-15,22,23 will find records 10,11,12,13,14,15,20,23.
- The More Options hyperlink allows string searching and other options.

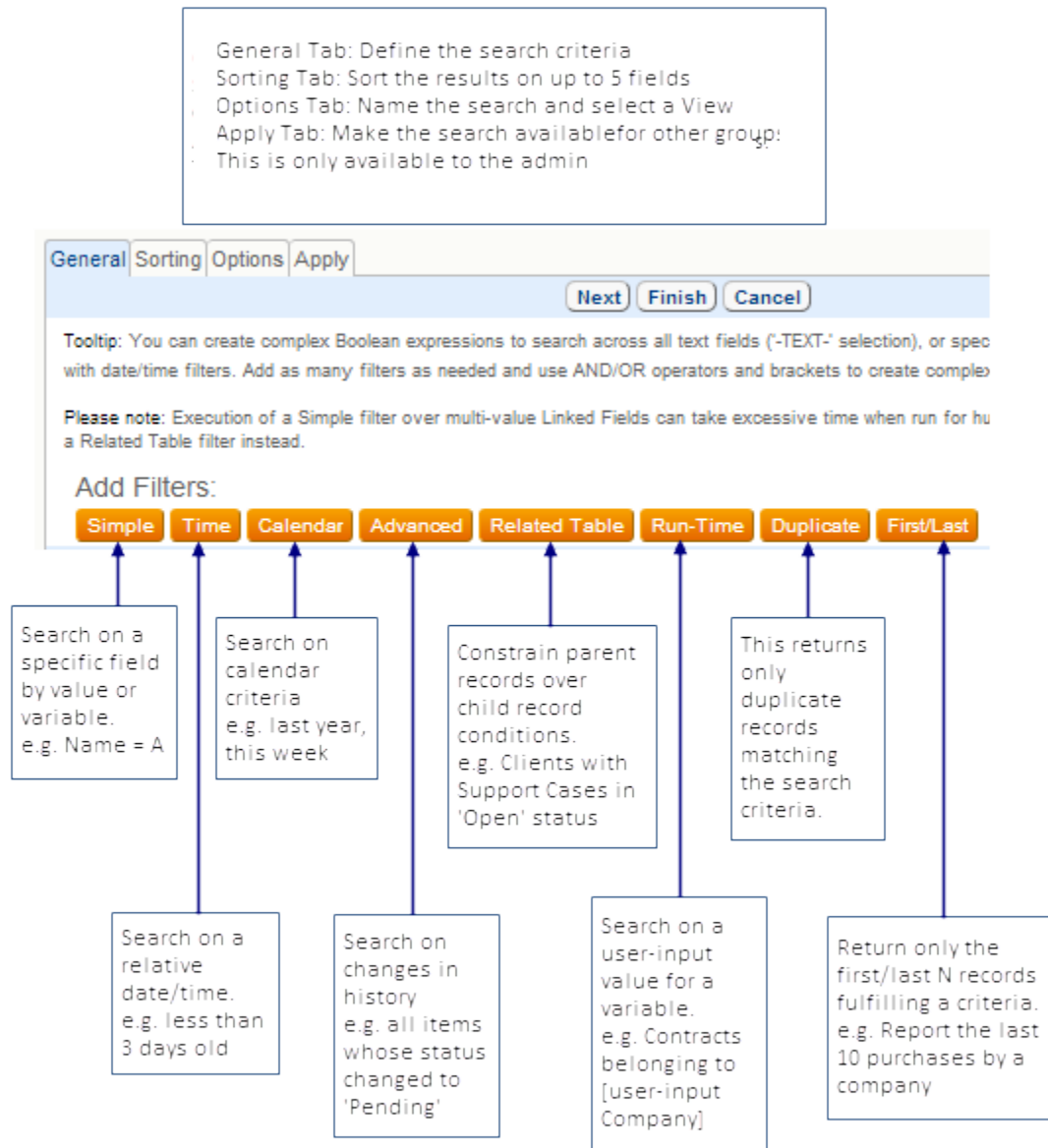
- Show All removes any filters and shows all the records.
- Refine narrows already executed search by new criteria. For example, you might search for Open issues, then enter IBM in the search box, click the Refine button and hit Go to find all the Open issues that contain the word IBM.
- Context shows context with search results. This is a very useful option, it shows where the specified phrase occurred (including within attached files) so that you can easily see which record is the one you want.
- Active only (defined in More Options) refines the search so that it only displays the "active" records. You might define "active" to be those records that are assigned to you. So, when you enter the text ABC, records containing ABC assigned to you are found.

SAVED SEARCH DROP-DOWN

This drop down list is invoked by moving the mouse over the Saved Search text in the search block.



ADDING A NEW SEARCH:



Also, Options tab introduces the ability to show the Total, Minimum, Maximum, and/or Average of selected fields in the records found by the search.

Any of these calculations options chosen will appear above the list of results (in the Status area) when running the search. If more than one calculation is shown, each will be displayed on its own line.

For example:

Status: 200 records found, 5 pages

Sum of Est Man Hours to Complete 130.0

Sum of Unbillable Hours 35

Max of Billable Hours 82

For the found records, show the following:

Minimum of the Latest Contract End Date

Maximum of the Date Created

Average of the Annual Revenue

USING WORKING HOURS IN SEARCHES:

- Relative Time searches (Time Filter) include an option to include only time that occurred during a team's working hours:

Limit to Working hours for:

team

team in the Account Rep field

- This is useful for escalations for companies that only include business hours in SLAs – for instance, if a ticket is submitted at 4:50pm and should be escalated after one working hour to a manager, using a search in which Date Created is more than 1 hour old and checking the option to include only the working hours of a team that has hours from 9 to 5pm will result in the ticket being found only when the search is run after 9:50am the next day

See Also:

[Saved Search Menu](#)

[Saved Search](#)

SAVED SEARCH

A complex search is created by applying one or more filters to data contained in tables.

Agiloft supports several types of filters, which can be further combined:

- Simple
- Time
- Calendar
- Advanced
- Related Table
- Run-Time
- Duplicate
- First/Last

If several filters are to be applied, the Boolean operators (And/Or) may be specified (leftmost column) and these expressions may be enclosed in brackets to reflect precedence.

The following types of filter may be applied:

Simple filter

Add Filters:

Simple Time Calendar Advanced Related Table Run-Time Duplicate First/Last

Status equals. = Value Closed

Next Finish Cancel

A simple filter is created from a combination of:

- A viewable field of a record. The available fields are alphabetically ordered on a drop-down list.
- A comparison operator (e.g. =, !=, >, <, etc)
- A value or formula.

The appropriate selection of operators and values is provided automatically based on which field was selected.

For example, selection of a choice field changes the dropdown list to the valid values for that field. Selection of a date field changes the dropdown list to a calendar of dates.

Choice fields are assumed to be in descending order listed.

Specifically, if the Priority list reads:

- Very High
- High
- Medium
- Low

a search for Priority \geq Medium will find Very High, High and Medium.

The same search for a Priority list of:

- Low
- Medium
- High
- Very High

returns records with a priority of Medium or Low.

Simple searches run on a default search field denoted by **-TEXT-**, the **-TEXT-** field searches all text fields within a record, including attached searchable files. Searchable files include .doc/x, .txt, and non-image .pdf.

Select the "Now" checkbox to search for records that currently have the specified value. De-select the checkbox to search for records that have fulfilled the criteria at any point in time.

For example, if the box is checked, a search for "Priority = High" returns records that have a priority of High at the present time.

If the "Now" box is unchecked, the search returns all records that have ever had a priority of High. Searches on History run slower than searches without History, but these can be very useful, especially when combined with the Advanced Search criteria.

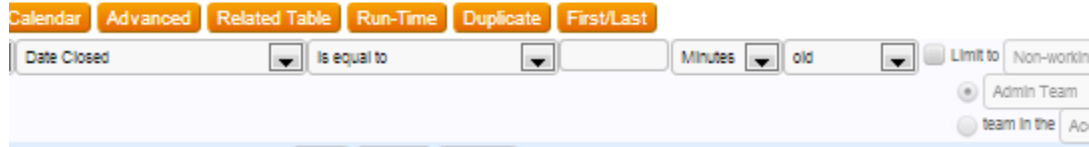
To illustrate, two hypothetical records are provided. All records start with a Priority of "Urgent".

- Record 1: The record is closed. The priority of the record remains "Urgent".

- Record 2: The record is closed. Later, a staff member returns to mark the priority as "Low".

A search for urgent priority records whose status changed from anything to Closed returns only Record 1. The same search with the "Now" box unchecked returns both records.

Time Filter



Time filters are used to search on periods of elapsed time relative to a specific field. The anchor point of a time filter is the field itself. The amount of time specified is an integer.

Example: Date Created is less than or equals to 5 hours old.

Because of the way the Agiloft system stores time, searches on a range of times are more easily handled in with the Calendar filter discussed in more detail below.

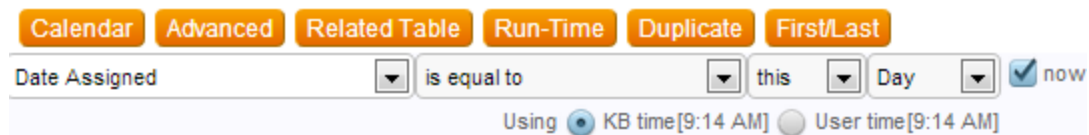
Example: Date Created is equal to last month.

Field selection includes all user viewable date, time, and date/time fields. The time units list includes past and future values ranging from Minutes to Years.

In addition to specifying a particular team, the Team in a particular field in the record may be chosen.

For example, if a ticket must be escalated after 5 working hours, instead of manually selecting all possible teams, the search parameter should be limited to the "Assigned Team" field. The filter dynamically works on whatever team value is in the field is specified.

Calendar Filter



Calendar filters are used to search on a range of time. The anchor point of a calendar filter is the range of time being searched. Options are provided for time

ranges of minute, hour, day, week, month, quarter, and year. The search returns results for any records whose day/time falls within the entire time range.

Example: Date Created is less than or equal to last month.

Because of the way the Agiloft system stores time, searches on time elapsed time are best handled through a Time filter.

Example: Date Created is greater than or equal to 8 hours old.

There is a further option to search on calendar times based on the internal KB time or the user's current time zone. The default calendar filter uses internal KB time.

Advanced Filter

Advanced filters provide the ability to define a search for instances of a field changing. Settings can also search for a choice/numeric field changing from a set of specified values to another set of specified values. Users can select in which period of time the change must have occurred.

Example: Records whose priority changed from Low or Medium to High in the last two weeks.

Related Table Filter

Related table filters are used to run searches on linked fields. The filter will only run on an already existing Saved Search in the target table. There is no option to create a Saved Search for a different table through the active table.

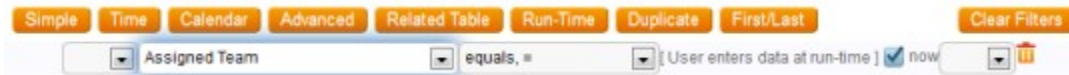
Example: Assets whose Contracts were renewed in the last 3 months.

In the example above, Assets and Contracts are two different tables. In order to run the search as described, a previous saved search in the Contracts table named

"Contracts renewed in the Last 3 Months" must already be in existence. The Asset table has linked fields to the Contracts table. Once the Asset table is open, a Related Table Filter can be applied to the linked fields from the Contracts table through the saved search of "Contracts renewed in the Last 3 Months."

While the same search can be accomplished using one or many Simple filters, for large tables a Related table filter is more efficient.

Run-Time Filter



Run-Time filters are used to return search results based on a user-input value. These filters are especially useful in the context of Charts and Reports. The user-input aspect provides flexibility and

Example: All Support Cases related to one company.

This search can be run in one particular instance using a simple filter, however if the search must be executed for each company, there must be a different saved search for each company. Instead of creating a different search for each individual company using a simple filter, a run-time filter should be used. This requires only one saved search to be created.

Duplicate Filter



Duplicate filters are used to returns results on searches that have more than one record fulfilling the criteria.

Example: Companies who have three or more expired contracts.

In that case, the field to be matched is "Contract Status" and the duplicate filter is set to report only if there are 3 or more matches in the Contract Status field.

There can be only one filter of either the Duplicate or the First/Last filter type.

First/Last Filter



First/Last Filters are used to return either the first or the last result of a search.

Example: The most recent contract signed by a specific partner company.

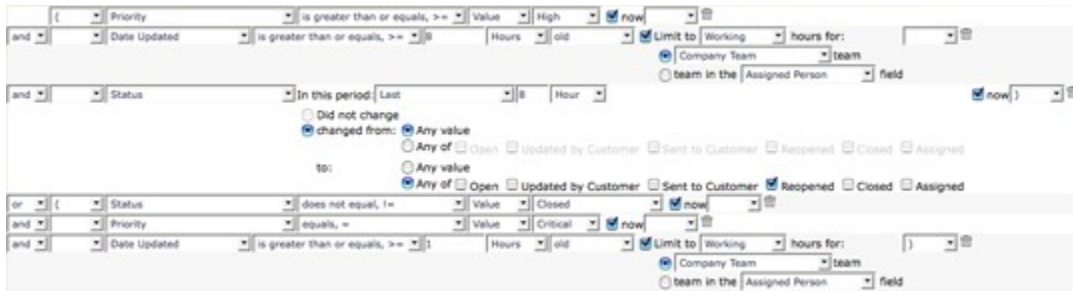
In that case, the field to be matched is "Date Signed" and the first/last filter is set to report on the first match.

The First/Last filter applies the order defined in the Sorting Tab.

Combinations of filters

Multiple selection criteria can be defined in few clicks to create a powerful search such as:

Records of High or higher priority that were last edited more than 8 hours ago according to Company team working hours and that were reopened at that time or those not yet closed of Critical priority last modified more than 1 hour ago according to Company team working hours



See Also:

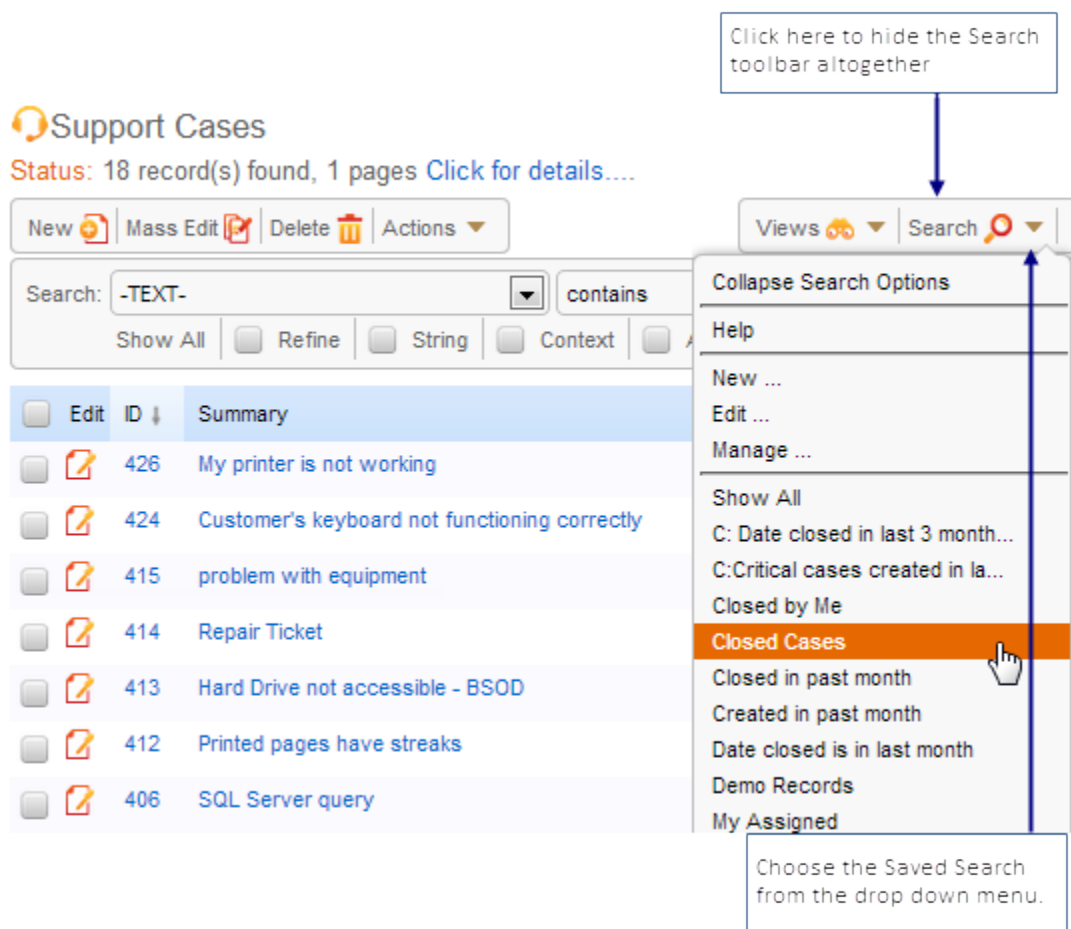
[Search](#)

[Saved Search Menu](#)

[Field](#)

[Record](#)

SAVED SEARCH MENU



Frequently used Searches can be saved and re-executed. Saved Searches are listed on the Saved Search menu to the right of the search input block. A Saved Search can optionally be made accessible to other users, so that they can execute it from their Saved Search menu.

The 'New' menu item invokes the Search wizard to create a new search.

The 'Edit' menu item invokes the Search wizard to modify the current saved search (the currently selected search is highlighted on the menu). In other words, to modify a search, first execute it and then select Edit.

The 'Manage' menu item provides access to viewing, editing, or deleting saved searches.

See Also:

[Search](#)

[Saved Search](#)

VIEWS

WHAT ARE VIEWS?

- The way in which multiple records from a table are displayed is controlled by a "View" that defines the display characteristics (but not the data included, which is defined by a search)
- Staff users can create their own views that only they can access, and they can switch views at any time.
- Admins can create views and apply them to teams of users, making them accessible or the default view for that table.
- Besides the possibility to choose the view explicitly from a drop-down, saved searches can replace the current view with a view defined in the search – this is particularly useful when viewing hierarchical data, as different sub-tables often have different optimum views showing different fields.
- Views can include color coding of rows based on a choice field, which can be populated by rules based on any criteria, enabling any color coding logic needed.

Table View Example:

The fields selected, their width, order and number of lines per record is defined by the view

Views are managed, created and selected from the Views drop-down

Support Cases
Status: 17 record(s) found, 1 pages [Click for details...](#)

New [icon] Mass Edit [icon] Actions [icon]

Edit View	ID ↓	Summary	Person
[icon]	424	Customer's keyboard not functioning correctly	Caldwell
[icon]	415	problem with equipment	Caldwell
[icon]	414	Repair Ticket	Caldwell
[icon]	413	Hard Drive not accessible - BSOD	Caldwell
[icon]	412	Printed pages have streaks	Caldwell
[icon]	406	SQL Server query	Caldwell
[icon]	405	Calculation with fields from the same record	Caldwell
[icon]	404	How do we login to the alternative end user portal?	Caldwell
[icon]	403	How do we login to the mobile interface?	Caldwell
[icon]	402	How do I request a license for Agiloft?	Caldwell
[icon]	382	Reports are incorrect in Peoplesoft	Caldwell
[icon]	356	My printer is not working although it shows that it is ready	Caldwell

Views [icon] Search [icon] [icon] [icon] [icon]

- New ...
- Edit ...
- Manage ...
- Personal view
- Color Coded Background by Priority
- Color Coded Font by Status
- Default End User
- Detail View
- E: Cases in Contacts
- FAQ View
- id, customer, company;Summary
- Mobile view
- Staff default
- staff home page
- staff view with time
- Staff with description
- Summary view
- Help

Closed Medium Christopher Caldwell

Views allow different individuals, teams and groups to have their own perspectives on the same data in a table.

A view may define which fields to display, the field width, field order, number of rows, coloring and the number of records displayed per page.

A Personal View and Summary View are always available. Changing your Personal View will have no effect on other users' Personal Views.

New - Invokes the Set View wizard to create a new table view.

Edit - Invokes the Set View wizard to edit the current view (the currently selected view is highlighted).

Manage - provides access to edit, delete or copy other views.

Select any other menu item to obtain an alternative view of the table.

VIEW WIZARD

Fields tab

Helps you to configure how many lines to display per record, should the edit/view icons be displayed, lets you choose the subset of fields to display and set width limit per each field.

- Show Edit icon
- Show View icon (if available)
- Show Icon Column

Max lines per record:

Max lines per linked field:

Field Name	Display	Column Width	Edit	View	Right Align
Account Number	<input type="checkbox"/>	Automatic	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Additional Notes	<input type="checkbox"/>	Automatic	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Assign to Me	<input type="checkbox"/>	Automatic	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Assigned Person	<input checked="" type="checkbox"/>	Automatic	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Assigned Team	<input type="checkbox"/>	Automatic	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Assigned Team Leader	<input type="checkbox"/>	Automatic	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Cell Phone	<input checked="" type="checkbox"/>	16	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Change Category	<input type="checkbox"/>	Automatic	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Order tab

Here you drag and drop fields in the order you want them and define the row coloring if any desired.

You can create a multi-row display by dragging fields onto a new line (below the current line). This is useful for fields such as "Solution" that may contain a lot of text.

Fields Order/Colors General Apply

Back Next Finish Cancel

Create the desired layout of the table view by dragging and dropping the field boxes displayed below. Single fields, taking the whole view width, may be positioned in the second row. Additional rows may also be created by 'dropping' selected fields after the last row. Any multi-column fields will appear without any field label.

ID	Priority	Assigned Person	Status	Date Created
Summary				
Customer Name				

Set Row Coloring Define Notification Icons

Back Next Finish Cancel

General

Here you provide a name for the view and control the number of records per screen,

*View Name

My New View

Deletable

- Yes
 No
-

Maximum View Width

70 characters

Minimum View Width

- None
 Same as Action Bar
 0 characters
-

Minimum Field Width

10 characters

Records Per Page

20 ▼

Display Cell Borders?

- Display borders around the cells of the table
 Do not display borders

Apply

This tab is only visible to those with administrator rights - they can make the view available to other teams. In a similar way the administrator can set a view as default for another team.

Make this view visible to:

Keep settings

Change settings

View is available for me

and for teams

- Admin Team
- Agiloft Implementation Team
- Change Approval Team
- Company Team
- Contract Creator Team
- Contract Manager Team
- Customer Manager Team
- Customer Team
- Document Approval Team
- Document Creator Team
- Internal Customer Team
- Internal Implementation Team
- Marketing Team
- Professional Services Team
- Repair Team
- Sales Manager Team
- Sales Team
- Support Manager Team
- Support Team
- System Admin Team

Select all

Deselect all

and their subteams

Any new teams that are created

MANAGING VIEWS

The Manage item on the Views drop-down gives you access to all views

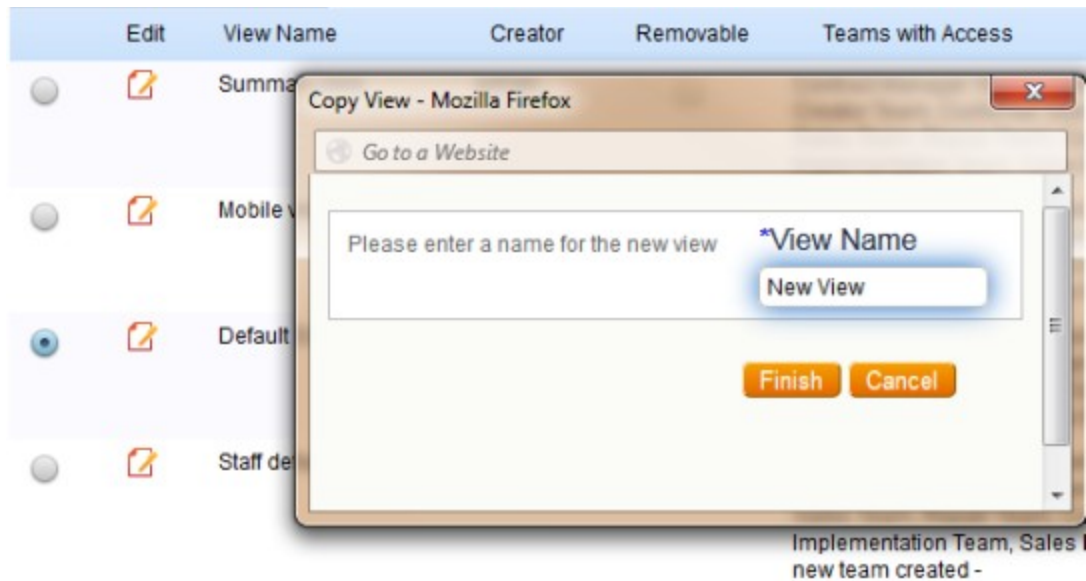
Agiloft Reference Manual

	Edit	View Name	Creator	Removable	Teams with Access	Default for Teams
<input type="radio"/>		Summary view	admin	<input type="checkbox"/>	Support Manager Team, Contract Manager Team, Sales Manager Team, Document Creator Team, Sales Team, Customer Manager Team, Marketing Team, Support Team, Change Approval Team, Internal Customer Team, Contract Creator Team, Customer Team, Admin Team, Professional Services Team, Agiloft Implementation Team, Internal Implementation Team, Document Approval Team, System Admin Team, Company Team	-- none --
<input type="radio"/>		Mobile view	admin	<input type="checkbox"/>	Support Manager Team, Contract Manager Team, Sales Manager Team, Document Creator Team, Sales Team, Customer Manager Team, Marketing Team, Support Team, Change Approval Team, Internal Customer Team, Contract Creator Team, Customer Team, Admin Team, Professional Services Team, Agiloft Implementation Team, Internal Implementation Team, Document Approval Team, System Admin Team, Company Team	-- none --
<input type="radio"/>		Default	admin	<input checked="" type="checkbox"/>	Support Manager Team, Contract Manager Team, Sales Manager Team, Document Creator Team, Repair Team, Sales Team, Customer Manager Team, Marketing Team, Support Team, Change Approval Team, Internal Customer Team, Contract Creator Team, Customer Team, Admin Team, Professional	Support Manager Team, Contract Manager Team, Sales Manager Team, Document Creator Team, Repair Team, Sales Team, Customer Manager Team, Marketing Team, Support Team, Change Approval Team, Internal Customer Team, Contract Creator Team, Customer Team, Admin Team, Professional

CUSTOM VIEWS

Using the Manage Views screen (Set View/Manage), the user can quickly create a new view on the basis of an existing view by using the Copy hotlink, if she is authorized to create a view based on her group membership.

For any field displayed in a view, the user can select whether clicking it will display Edit Record or View Record form for that record. If a view includes the Links field, clicking it will open the Edit Record form for the record linked to the selected record.



See Also:

[Views](#)

[Default Views](#)

[Row Coloring](#)

[Views and Permissions](#)

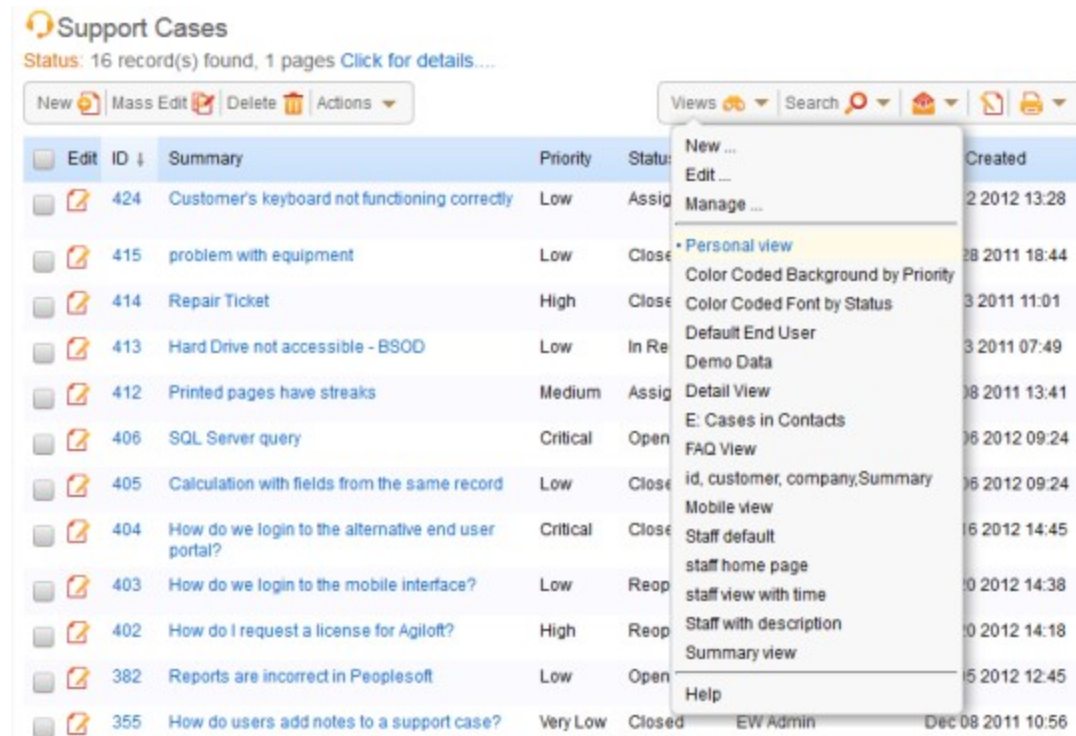
DEFAULT VIEWS

Views are specific to the tables for which they are created. When you create a table, three pre-defined views are created by default: Personal View, Mobile View and Summary View.

Personal View

If you change your own Personal View, it will have no effect on other users' Personal Views. By contrast, if you change any other view for which you have access, it will affect all users with access to it.

A user's personal view will not be overwritten with the default view for his team when the default view is modified. If a user is using the team default view and makes changes, then those changes are saved to the user's personal view. The user's personal view will not be changed if changes are subsequently made to the team default view, providing the ability for user's to have their own personal view and a team default view.



Mobile View

Mobile View shows only one field by default, the "Created By" field. The Mobile view cannot be deleted, but it can be edited. Mobile views are designed to be used when a user is logging into the system with a mobile device, e.g. iPhone, BlackBerry or

Android device or when a user is logged in over the web as an ADA compliant interface.

Summary View

Summary View shows only two fields by default: Record ID and the Creator Login field. As the Mobile view, the Summary view cannot be deleted, but it can be edited.

See Also:

[Views](#)

[Custom Views](#)

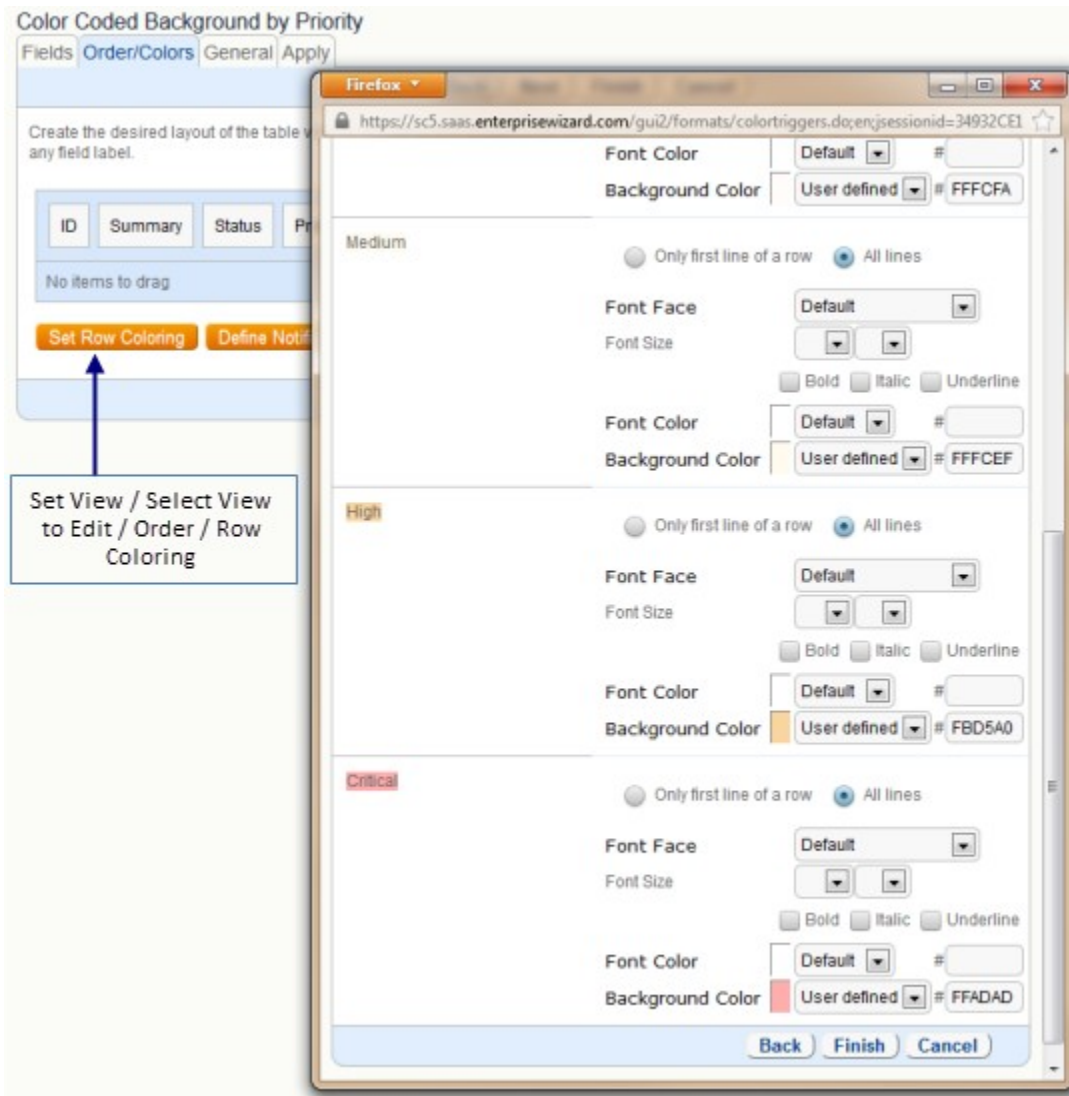
[Row Coloring](#)

[Views and Permissions](#)

[Teams](#)

ROW COLORING

In order to provide a visual aid to easily identify specific record attributes there is an ability (via Set View/Select View to Edit/Order/Row coloring) to customize the appearance of specific records based upon the value of a selected choice field. The text font, size, color and background color may be selected for each different value of the choice field.



For example, if the selected choice field is Priority then all Critical records may be displayed in Red, High displayed in Orange, etc.

Support Cases
 Status: 16 record(s) found, 1 pages. Click [here](#) to re-count records.

<input type="checkbox"/>	Edit View	ID ↓	Summary	Status	Priority	Assigned Person
<input type="checkbox"/>		424	Customer's keyboard not functioning correctly	Assigned	Low	Christopher Caldwell
<input type="checkbox"/>		415	problem with equipment	Closed	Low	Christopher Caldwell
<input type="checkbox"/>		414	Repair Ticket	Closed	High	
<input type="checkbox"/>		413	Hard Drive not accessible - BSOD	In Repair	Low	Christopher Caldwell
<input type="checkbox"/>		412	Printed pages have streaks	Assigned	Medium	Jack Repairguy
<input type="checkbox"/>		406	SQL Server query	Open	Critical	Ralph Knowles
<input type="checkbox"/>		405	Calculation with fields from the same record	Closed	Low	Ralph Knowles
<input type="checkbox"/>		404	How do we login to the alternative end user portal?	Closed	Critical	Christopher Caldwell
<input type="checkbox"/>		403	How do we login to the mobile interface?	Reopened	Low	Christopher Caldwell
<input type="checkbox"/>		402	How do I request a license for Agiloft?	Reopened	High	Ralph Knowles
<input type="checkbox"/>		382	Reports are incorrect in Peoplesoft	Open	Low	Christopher Caldwell

See Also:

[Views](#)

[Default Views](#)

[Custom Views](#)

[Look and Feel](#)

VIEWS AND PERMISSIONS

Access to views is controlled by group permissions. An admin user can allow selected groups to create and edit views, and optionally make them visible to other users. Users with the permission to publish views can assign them to users in other teams, and set the view as default for those users. To make a view accessible and/or the default view for other teams, click Set View/Select View to Edit/Apply. When a view is assigned as default for selected teams, it replaces any Personal view that the users in those teams have defined.

To see a list of views and their access rights, click Set View/Manage. In the displayed table of available views the 'Teams with Access' column shows the teams that are allowed to use the view.

	Edit	View Name	Creator	Removable	Teams with Access
<input type="radio"/>		Summary view	system	<input type="checkbox"/>	Support Manager Team, Con Customer Manager Team, Mi Team, Customer Team, Adm Document Approval Team, S
<input type="radio"/>		Mobile view	system	<input type="checkbox"/>	Support Manager Team, Con Customer Manager Team, Mi Team, Customer Team, Adm Document Approval Team, S
<input type="radio"/>		Default View	system	<input checked="" type="checkbox"/>	Support Manager Team, Con Customer Manager Team, Mi Team, Customer Team, Adm Document Approval Team, S
<input type="radio"/>		View in Assets	system	<input checked="" type="checkbox"/>	Support Manager Team, Con Customer Manager Team, Mi Team, Customer Team, Adm Document Approval Team, S
<input type="radio"/>		Detail View	ewsystem	<input checked="" type="checkbox"/>	Admin Team, Support Manag Internal Implementation Team
<input type="radio"/>		Personal view	ewsystem	<input type="checkbox"/>	-- none --

Even if a group (or individual user) does not have the permission to create or edit views, they can still edit and save their Personal View. Views are not used for setting [record level access](#) This is controlled by [Group](#) permissions.

See Also:

[Views](#)

[Default Views](#)

[Custom Views](#)

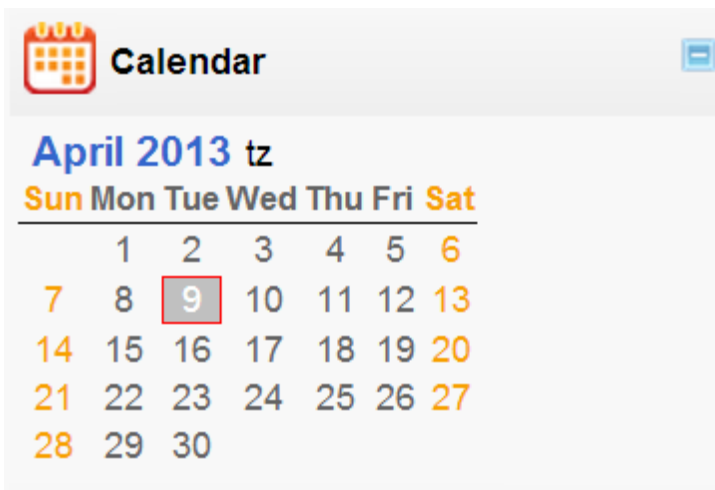
[Groups](#)

[Teams](#)

CALENDAR

The Calendar is a centralized resource that helps you manage your appointments, events and to-do items. The Calendar allows you to schedule events of a selected type, for instance meetings, phone conferences, vacations or announcements. The events you are creating may involve one or several users or teams. For instance, you can schedule a simple "To Do" item that references you only, such as "Call John at 2pm", or references multiple objects of different types, for example "Have a meeting with John, Victor and Mary in Conference room A at 2pm".

An event takes place at a predefined location, for example, Meeting Room or Phone, or may have no location at all, for example, a simple reminder like "Follow up on Whitney's proposal".



See Also:

[Calendar Entries](#)

[New Events](#)

Left Hand Pane Setup

CALENDAR ENTRIES

To pull up the Calendar, click on a date in the Calendar shown in the left hand menu.

Note that dialog includes Next and Previous links so you can move easily from one day to the next day, or the previous day.

Calendars are implemented as standard tables so you can add fields and create links to other tables if necessary.

You can also synchronize your schedule with Microsoft Outlook via Export/Import buttons.

Show for:

- All
- User
- Team

◀◀ Apr 2013 ▶▶

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				

tz ▼ Today

Current KB Time:
09 Apr 2013 09:01

[Subscribe to EW Calendar](#)

[Manual Export to MS Outlook](#)

[Manual Import from Outlook](#)

Day	Week	Month	3 Months	Year
:30				
3:00 am				
:30				
4:00 am				
:30				
5:00 am				
:30				
6:00 am				
:30				
7:00 am				
:30				
8:00 am				
:30				
9:00 am				
:30				
10:00 am				
:30				
11:00 am				

See Also:

[Calendar](#)

[New Events](#)

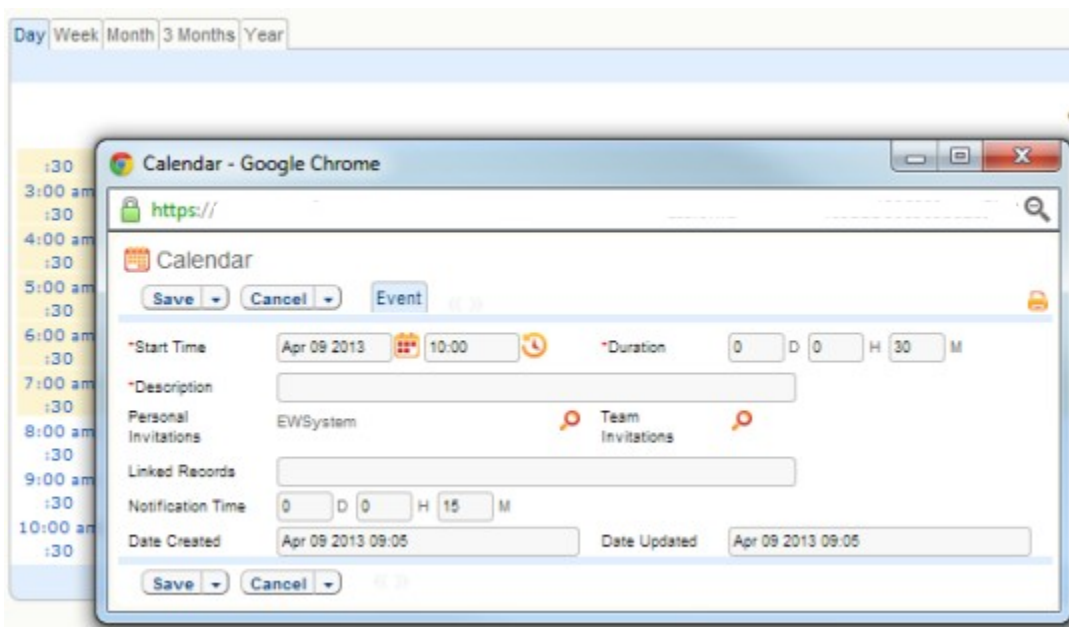
NEW EVENTS

To schedule an event, click on a time in the Day view of the Calendar. This launches the Event Wizard.

When you create an event, you become a meeting coordinator for that event. An invitation will be created first as a pop-up dialog and then emailed.

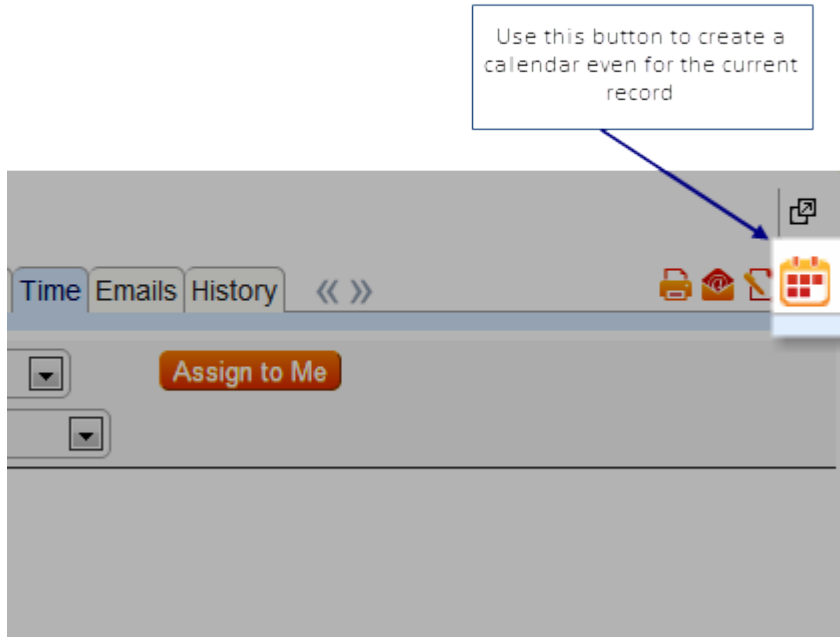
When scheduling an event, you can quickly see the schedules of the objects that are related to that event. For instance, if you click on the user, you will see his or her schedule. If you click on the conflicting event, you will be shown the event details.

Each event has a start time and end time, and the system checks that no conflicting events occur. For instance, the system will make sure the Room is available at the requested time and all the Required Attendees are available when an event of type Meeting is created.



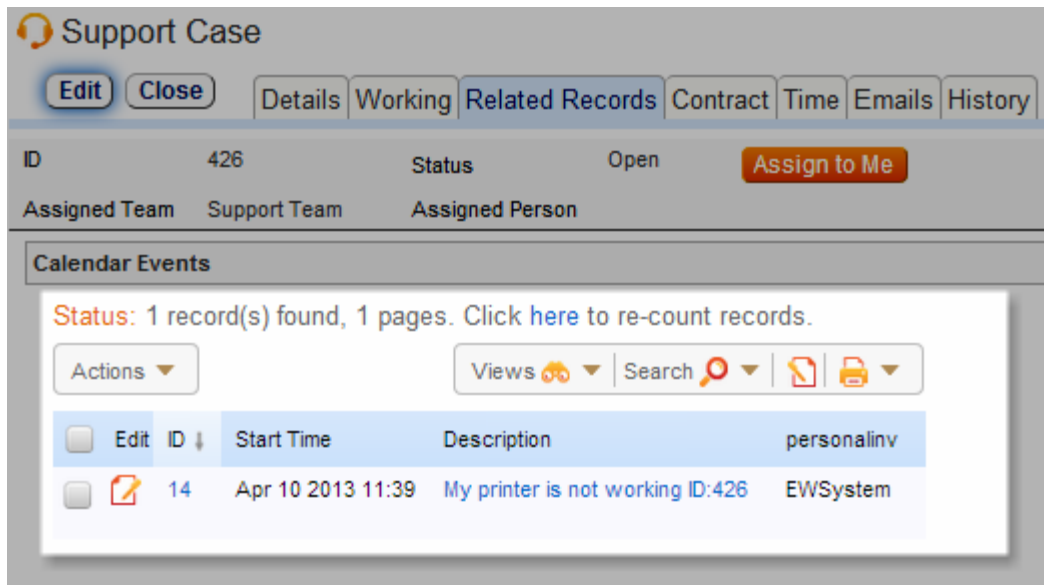
Adding an Event for the Specific Record

You can use the calendar icon in a table to create a calendar event (In the Edit mode for the record, find an icon in the top right corner)



This event could be viewed in the record from which it was created. A link is set up in both directions, so that in the calendar record view you could see the linked item as well.

This makes calendar entries available to the related table data type so you can create an embedded table in the record layout for any other table that for a given record will show all calendar entries related to this particular record.



See Also:

[Calendar](#)

[Calendar Entries](#)

CALENDAR SETUP

The Calendar can be fully configured for each Knowledgebase to suit your particular needs. In addition to creating calendar entries for the built-in objects, like User or Meeting Room, you can also set schedules for custom tables, for example, create schedules of equipment that is used at trade shows.

The admin user can customize the default Calendar objects, like Meeting Places, Event Types, Default Notifications, Acceptance or Decline URLs, etc. These settings are available by clicking Calendar/Event Wizard/Administration.

Tables that support Calendar entries have Events tab displayed when a new object (record) is created or edited.

See Also:

[Calendar](#)

[Calendar Entries](#)

[New Events](#)

[KnowledgeBase](#)

MASS EDIT

A Mass Edit operation can modify several records and fields simultaneously.

For example, the 'Status' field in a particular set of records can be assigned the value 'Closed' . A more complex example might be to assign all records resulting from a Search filter a Status of 'Closed' , assign 'Sep 15 2004' as the date in the 'Closed on' field and append the string 'Resolved in Release 2.4' to the 'Comment' text field.

Records to be updated are selected in the leftmost selection box. The Mass Edit Wizard is invoked from the toolbar menu in a Table view. (For more information on using table views, see Table View.)

Helpdesk Cases

Status: 11 record(s) found, 1 pages. Click [here](#) to re-count records. 5 records are selected

The screenshot shows the 'Mass Edit - Google Chrome' window. The URL is <https://sc4.enterprisewizard.com/gui2/do/massedit;enjsessionid=CA3DB046931D8069>. The wizard has three tabs: 'Select Fields', 'Update', and 'Confirm'. Below the tabs are 'Next' and 'Cancel' buttons. The 'Select Fields' section is active, showing 'Helpdesk Case Fields'. It includes a dropdown menu for 'To edit linked records select the linked field: ---Select one---' and a message 'Please select the fields to be mass edited.' with a help icon. There are 'Select All' and 'Clear All' buttons. Below this is a section titled 'The fields to modify' with a grid of checkboxes for various fields.

The fields to modify		
<input type="checkbox"/> Additional Notes To/From Customer	<input type="checkbox"/> Alternative Assigned Team/Person	<input type="checkbox"/> Asset ID
<input type="checkbox"/> Asset Model Name/Number	<input type="checkbox"/> Asset Name	<input type="checkbox"/> Asset Owner Name
<input type="checkbox"/> Asset Responsible Team	<input type="checkbox"/> Asset Subtype	<input type="checkbox"/> Asset Tag
<input type="checkbox"/> Asset Type	<input type="checkbox"/> Assigned Person	<input type="checkbox"/> Assigned Team
<input type="checkbox"/> Assigned Team Leader	<input type="checkbox"/> Change Category	<input type="checkbox"/> Change Summary
<input type="checkbox"/> CR Conv ID	<input type="checkbox"/> CR conversion related to	<input type="checkbox"/> CR ID
<input type="checkbox"/> CR Spawned from HD ID	<input type="checkbox"/> CR Status	<input type="checkbox"/> Date Closed
<input type="checkbox"/> Date Created	<input type="checkbox"/> Date Updated	<input type="checkbox"/> Deletable
<input type="checkbox"/> Demo Data	<input type="checkbox"/> Downloadable File(s)(delete)	<input type="checkbox"/> FAQ Topics

Just like changes from the Edit form, Mass Edit changes are reflected in the History record.

If you do not see a Mass Edit link on the Toolbar for the selected table, you probably do not belong to a group with permission to edit multiple records at once (set on the Permission tab of the Groups wizard)

See Also:

[Mass Edit Wizard](#)

[Mass Edit Permissions](#)

[Field](#)

[Record](#)

MASS EDIT PERMISSIONS

Access to mass editing records in a table is controlled by group membership. The groups that are not authorized to mass edit records in a table will not see the Mass Edit hotlink on the toolbar for that table.

To grant a group the permission to mass edit records in a single operation, click Setup/Access/Manage Groups and select Group to edit. Click Table tab to select the table, then switch to Permissions tab and enable the "Mass Edit multiple records in a single operation" permission.

Allow this group to

Edit their own Companies

All

Only those that satisfy the saved search:

All Companies Alphabetical

Edit other people's Company

All

Only those that satisfy the saved search:

All Companies Alphabetical

Mass Edit multiple records in Company in a single operation

Link multiple records using the Link menu

Print records using the printer icon

See Also:

[Groups](#)

[Record Level Permissions](#)

[Field Level Permissions](#)

[Reports and Permissions](#)

[Mass Edit](#)

[Mass Edit Wizard](#)

MASS EDIT WIZARD

Select Fields **Update** Confirm

Back Next Cancel

You can enter new field values directly, or use the Formula radio button to set field values from

Please enter new values for these fields

Date of Purchase

Treat the above as Standard text A formula

Quote Addressed to

Treat the above as Standard text A formula

Round Robin Assignment

Team Leader of Admin

Reimport source record

Use the Company of use

more info

if the field allows multiple values

Choose the first record

Choose all records

Calendar - ...

https://sc4.enterprisewizard.co

◀◀ Apr 2013 ▶▶

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				

tz Today

The Select Fields tab allows you to select which fields to change. Fields which are 'greyed out' may not be modified.

An input field is displayed on the Update tab for each change. The input field may have an optional input aid, for instance, a Choice list for choice fields or Calendar for Date fields. In the example above, Calendar is displayed as the input aid for the 'Start Date' date field.

The Update tab also provides options for 'search and replace' string substitution and for using formulas and variables. For example, if you had fields Cost, Price and Quantity, you could set the Cost to Price * Quantity.

Note, that you can use search and replace even for fields with data types like "email".

Note that for multi-choice fields you can choose whether the new value should overwrite or append to the old value.

Select Fields **Update** Confirm

Back Next Cancel

You can enter new field values directly, or use the Formula radio button to set field values from existing formula

Please enter new values for these fields

Related To Opportunity Contact Support Case
 Helpdesk Case Change Request Asset

Treat the above as Standard text
 A formula

The new value should Overwrite old values with new value
 Append new choices to existing value

Back Next Cancel

See Also:

[Formula](#)

[Mass Edit](#)

[Field](#)

FORMULA

Formulas perform calculations on selected fields. This feature can be used via the Rules interface, or through Mass Edit, or may be attached to an Action Button in a table.

Formulas can be input directly, for example \$amount*10, where "amount" is a field name, or by using the Formula Wizard. To launch the Formula Wizard, select "A formula" radio button next to the appropriate input field and click Formula Help.

The first page of the Formula Wizard provides detailed instructions on usage and available selection of values, fields and functions available. The formula can be edited subsequent to function and field selection.

String Replacement

Text string replacement may replace, or append to, an existing string.

To replace part of a text string, delimit the text to be replaced (search string) and the replacement text with a slash ("/").

For example:

/dog/cat/

changes '**Good dog! Your dog-food is in the dog dish**'

to '**Good cat! Your cat-food is in the cat dish**'

Regular expressions can be used to specify the search string. Some examples of regular expressions are given below:

Expression	Matches...
sep [ae] r [ae] te	separate, seperate, seperete <i>or</i> separete
\bcat\b	<i>cat as an entire word, rather than part of another word, as in catalog</i>
Dec (ember) ?	Dec <i>or</i> December
cat dog mouse fish	cat, dog, mouse <i>or</i> fish

variable [0-9] a	variable0a, variable1a, etc ...variable9a
th (is) *	th, this, thisis, thisisis, ...
this+	this, thiss, thisss, ...

You can find more information on usage of regular expressions at <http://www.regular-expressions.info>

Ternary operator

For conditional statements, use the ternary "a?b:c" operator, which works much like an if-then-else action.

It works like this: If statement 'a' is true, value 'b' is returned, otherwise (i.e. if statement 'a' is false) value 'c' is returned.

For example:

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

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.

EMAIL

EMAIL WIZARD

The user can select one or more records in a particular table and send them by email using Email Wizard. To invoke the Email Wizard, select one or more records and click on Email Icon below the table. A pop-up window appears as shown below. The same window appears if you click the Email Icon when editing a record.

The main part of the Email Wizard is an HTML editor which allows you to create/edit and format text, insert pictures and hyperlinks and even tables into the email. You can send email in HTML or plain text. The top part contains additional menu options that allow you to add an attachment, call Formula Wizard to insert a formula or select field variables to be included in the email.

You can also send several records by email at once. To do it, select a few records and click the Multiple Email Options menu option. Select whether the system will send one email per Case or collate results into a single email per each recipient. The Multiple Email Options tab appears only on the Outgoing Email dialogues that can result in multiple emails, i.e. for rules, workflow actions, and the email icon from a table view.

Send Cancel

Insert Format Multiple Email Options

Linked Records	4 Tasks selected (IDs: 7, 8, 9, 10)
*From address	Inbound email account for Task
*Reply to	Email address of the individual who last updated the Task
*To	<input type="checkbox"/> Users: <input type="checkbox"/> Teams: <input checked="" type="checkbox"/> User Assigned Fields: Person <input type="checkbox"/> Email Fields: <input type="checkbox"/> Address in Field: <input type="checkbox"/> Link to Team: <input type="checkbox"/> Addresses: <input style="width: 150px;" type="text"/>
	CC one copy of the email to Users:
Subject:	<input type="text" value="New Task #\${id} has been assigned to you"/> Formula Help <input checked="" type="checkbox"/> Include encrypted ID information in subject line to enable replies to update correct record <input type="checkbox"/> Enable field parsing for any reply to this email <input type="checkbox"/> Include Hidden Code for Truncating replies
Message to import	<input type="text"/>

HTML text

Context #0
Fonts
A↑
B
I
U
Link
Text

Align
Indent
Outdent
Decrease Indent
Increase Indent
Table
Calendar
Image
Code
Undo
Redo

Please see the task `#{id}` about `formula($title)`.

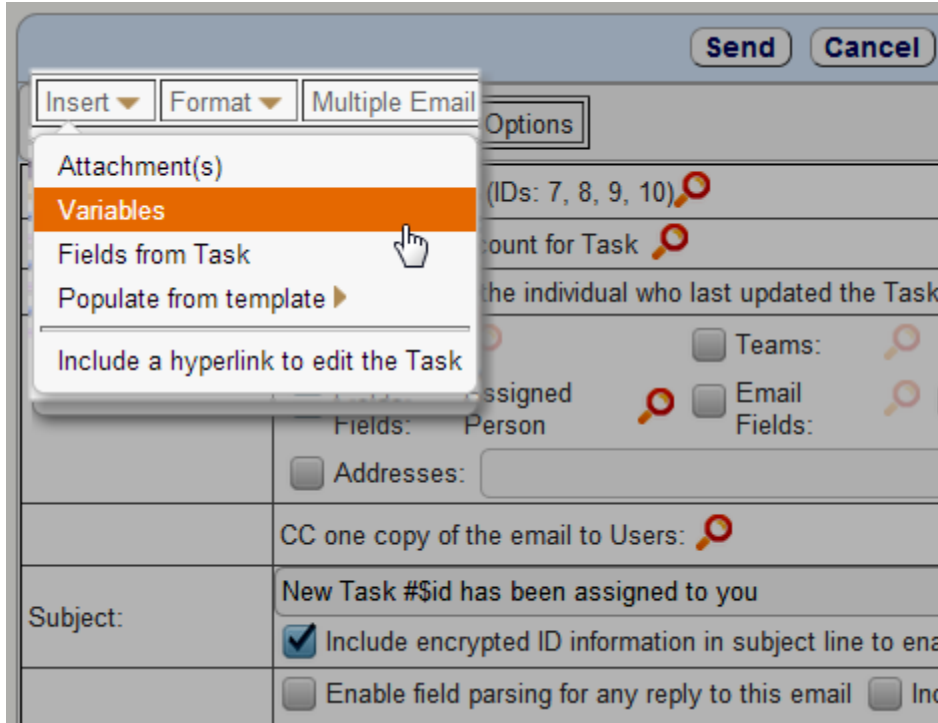
Click [here](#) to edit the task directly.

HTML Source HTML

Send Cancel

Attachments: No Files Attached Fields Included: Based on Recipient's Access Permissions

"Insert" option in top left corner allows you to:




- insert attachment
- create a formula based on available fields from current table, global variables and functions
- insert fields from current table and linked fields from other tables
- use existing template for the email

See Also:

[Communication Tables](#)

COMMUNICATIONS TABLES

- Each table has a related communications table that has several default fields, group permissions, layouts etc. like any other table. You can use the table wizard for these tables to edit group access to communications, change the layout when viewing a communication record, modify field labels, and so on.
- Each Communications table contains a linked field to the ID field of the related table, so communications are displayed as a Related Table data type in the other table.
- All outbound emails and all inbound emails that update an existing record are stored as entries in the related Communications table.
- A Communications table can be accessed directly through a toolbar tab for troubleshooting by changing group permissions to show the table on toolbar - this allows you to see all communications in that table in one view.
- **To Search through Communications** : First select a record, then click the email icon from the record view, then click the lookup next to Message to Import box. This will bring up a search dialogue allowing you to search the related Communications table.

 Support Case

Details
Working
Related Records
Contract
Time
Emails
History
<<

ID	426	Status	Open	<input type="button" value="Assign to Me"/>
Assigned Team	Support Team	Assigned Person		

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

<input type="checkbox"/>	ID ↓	To	Subject	Created On
<input type="checkbox"/>	8	example@example.com	Your Support Case #426 has been updated	Apr 03 2013 14:14
<input type="checkbox"/>	7	chris@example.com, example7@example.com, johndoe.ew@gmail.com	Customer has updated Case #426	Apr 03 2013 14:11
<input type="checkbox"/>	6	example@example.com	Your Support Case #426 has been created	Apr 03 2013 14:08
<input type="checkbox"/>	5	chris@example.com, example7@example.com, johndoe.ew@gmail.com	New Support Case #426 has been assigned to the...	Apr 03 2013 14:08

See Also:

[Email Wizard](#)

REPORTS

A report is a summary of table information that may be presented in graphical, HTML, text or Excel format.

Reports can be run interactively or scheduled for automatic distribution to individual users and Teams. They can also be written to the hard drive and accessed over the web without even logging into Agiloft.

For a list of reports available for a table, select its tab and click the Charts/Reports subtab.

Agiloft comes with dozens of built-in charts and reports that provide visual snapshots of any aspect of your business:

- expected sales revenues by region,
- deals closure rates by industry,
- open case numbers per support,
- person or leads conversion rates against sales method;

are just some of the many pre-defined charts designed to keep you on top what is happening.

Group permission access rules are respected in any data displayed in reports, consequently the same report automatically shows the data tailored for each user or group. There is also a permission flag in the group permission wizard to allow/disallow a group from being able to publish reports.

Reports and charts creation are integrated within the same wizard enabling you to develop them in parallel, or to quickly create charts from existing reports and vice versa. Graphical, text, HTML or Excel files may be generated by choosing the Report Type in Charts-Reports/Type. Note that a report can be exported as comma-delimited or tab-delimited text.

CUSTOM REPORTS

Reports can optionally be configured to show records filtered based on custom criteria or pertaining to data created or modified within a selected date range.

You can optionally display the records as they existed

- At their creation time, or
- At the time they met some criteria, or
- At the time the report is executed.

These customizations are available at Charts-Reports/Filter.

You can also configure the report to include dynamic variable values, for instance \$Company in the report title. The variables are automatically set to the corresponding value for the user to whom they are being emailed when the report is generated.

See Also:

[Reports_Wizard](#)

[Groupings](#)

[Appearance](#)

[Saved Search](#)

[Object](#)

[Flash Tutorials](#)



REPORTS WIZARD

The first screen of the wizard controls whether you see wizard tabs for charts, reports or both

Support Cases → **Charts/Reports**

General | Type

Next Cancel

Created by ewsystem

This chart/report will apply to the subtable of the **Support Cases** table that you select here.

Select All to include all subtables in the **Support Cases** table. [...More Help](#)

This chart/report should apply to:

Please enter a title for this chart.

Title:

Enter a description of this chart to help you remember its purpose.

Description:

If this option is selected, this chart will be accessible to all users who are allowed to view the table. Otherwise, it will only be visible to its creator.

Published

Your chart/report can use a combination of formats to present data in different ways. For instance, you can have a graphical chart, HTML-based or text report, or a downloadable Excel spreadsheet. Each of these formats will be placed on its own tab.

Output format:

Integrated Chart/Report

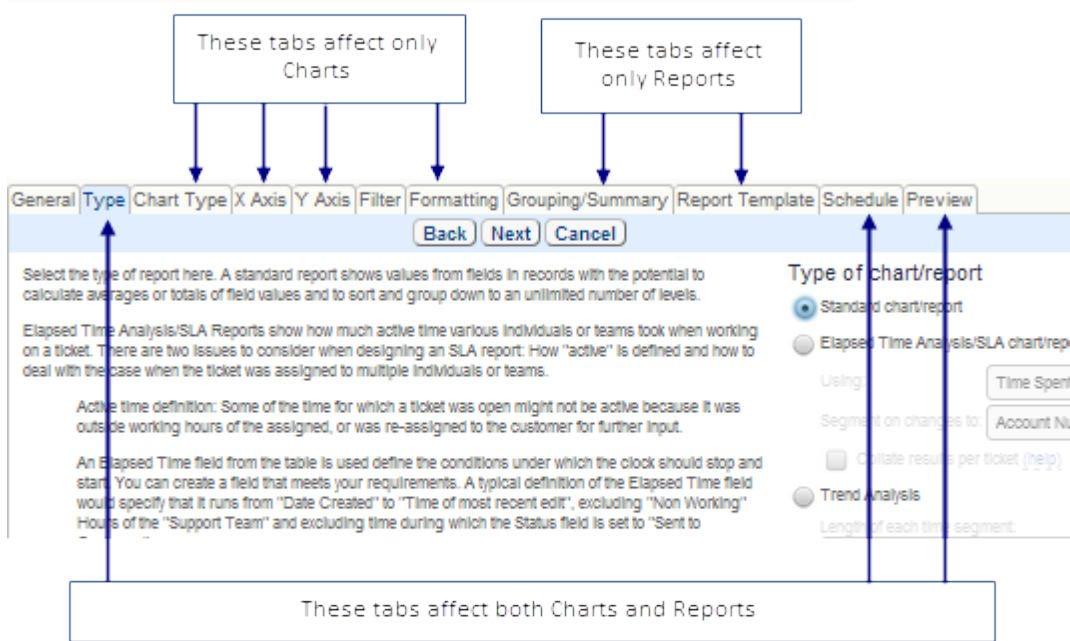
- Graphical chart
- HTML
- Excel
 - 2007(.xlsx)
 - 97(.xls)
- Text report
 - Tab
 - Comma delimited
- Custom Summary Report

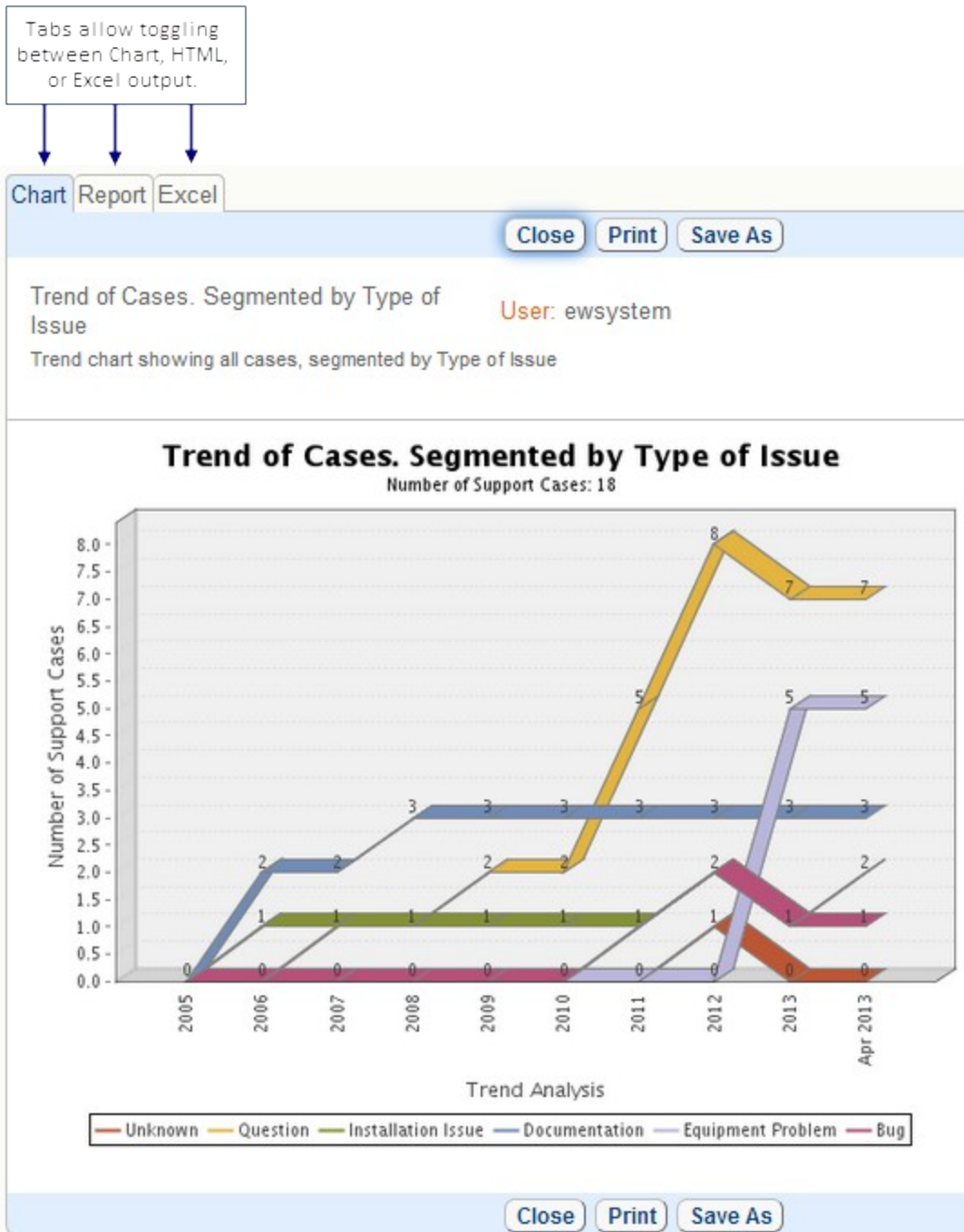
Next Cancel

These checkboxes determine which tabs are shown in the Charts and Reports setup.



Most tabs are common to both charts and reports





Date Range

Controls whether the report shows all the records from the table or just those that fall in a particular date range.

Chart Type

Selects the type of graphical chart (i.e. pie chart, segmented bar chart, etc)

Chart type:

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

X Axis

Controls what is shown as the X axis on a graphical chart and the first level of grouping for the associated report.

X (Horizontal) Axis

Field: ▼

Linked Tables:

▼

Field in Table Selected Above

▼

Sort bars by

Content

Number of items in the group

Number of subgroups

Summary value ▼

Results order

Ascending Descending

Sorting options for column bar charts and segmented bar charts allow the user to sort the results X-axis data according to the same options that are on the grouping screens.

Y Axis

Controls what is shown as the Y axis on a graphical chart and the second level of grouping for the associated report.

Y (Vertical) Axis

The number of Support Cases
 Total of the Time Spent
 Display Choice rounded to 1 digits
 Total obtained by
 All Time Entries added to All Cases for This Customer
 Formula Help
 Use custom label: Total of the Time Spent

For each segment, show:

- Total of the Time Spent
- percentage of Total of the Time Spent in that segment

It is possible to show chart data in percentages

Filter

Allows the user to apply a Saved Search that selects particular records such as Critical issues (within any Date Range defined on the first tab).

Saved search can be edited from this tab:

Relative dates:

Start Date: Date of Earliest Support Case

End Date: Today

Absolute dates:

Start Date:

End Date:

Based on

Date Created History

The chart/report shows records

- All Support Cases
- Created during selected date range
- Created or modified during selected date range
- Created, updated or simply existed during date range

Saved Search filter:

None

Saved Search

The chart/report shows:

- historic data
 - current data
-

Formatting

Controls details of the look and feel for charts


Title

Shows the title/description of the report and whether it is "published". Certain groups may only see published reports.

Grouping/Summary

Controls how the information in the report is grouped, what view is used to display it and what summary metrics are included.

Summary information:

New Delete	
Edit	Summary information
<input type="checkbox"/>	 sum(Time Spent)

Show records?

HTML

- Do not show records
- Show records using view selected below


Excel

- Do not show records
- Show records using view selected below

View for showing record:

Show not more than records in each grouping

Group data by:

New Delete				
	Edit	Grouping	Summary Information	Page Break
<input type="checkbox"/>		Type of Issue	sum(Time Spent)	No

The information for each record is show based on the chosen View

Average of Total Hours Spent : 2.936

Type of Issue : Question

Average of Total Hours Spent : 1.131

Priority : Low

Average of Total Hours Spent : 0.125

ID	Priority	Status	Assigned Person	Date Created	Customer Name	Summary
403	Low	Reopened	Christopher Caldwell	Jul 20 2012 14:38	EWSsystem	How do we login to the...
405	Low	Closed	Ralph Knowles	Oct 06 2012 09:24	Margery Wood	Calculation with fields...

Priority : Medium

Average of Total Hours Spent : 0.333

ID	Priority	Status	Assigned Person	Date Created	Customer Name	Summary
356	Medium	Closed	Christopher Caldwell	Sep 25 2012 09:08	John Suzuki	My printer is not...

Report Template

Controls the look and feel of reports

Schedule

Controls the distribution of charts and reports via email and allows them to be written to the hard drive.

Preview

Allows you to preview changes made in the wizard, prior to saving them with the Finish button.

See Also:

[Reports](#)

GROUPINGS

Reports can include an overall summary, displayed at the beginning of the report. They can also include any number of groupings and each grouping may have an optional summary displayed on a separate line above the grouping.



A grouping allows you to see a particular aspect of the data included in the report. For example, you might want to group the records based on their Priority and calculate the average cost for each set of data with a certain Priority. This 'grouping-specific' data can also be included with the summary information for the whole report.

To create a new grouping or summary, click Charts-Reports/Grouping-Summary.

Grouping/Summary Details

- Grouping-Controls how the information in the report is grouped. The first two levels of grouping for a combined chart/report are defined by the X and Y Axis, for example:

Group data by:

New Delete				
	Edit	Grouping	Summary Information	Page Break
<input type="checkbox"/>		Type of Issue	avg(Total Hours Spent)	No
<input type="checkbox"/>		Priority	avg(Total Hours Spent)	No

- Click New in the Group table above to add additional groupings:

Grouping | **Sorting**

Next **Finish** **Cancel**

Please select a field by which Support Cases will be grouped.

Group by

Field:

Linked Tables:

Field in Table Selected Above

Each grouping can have optional summary lines displayed on top of the grouping. To add a summary line for this grouping, click [New](#). ...[More Help](#)

Summary Information

Edit Summary information

No summaries defined

Create a printer page break before each new group

Insert Page Break

Yes No

Next **Finish** **Cancel**

- Click the Edit icon next to each Grouping and click New in the Summary table that is shown for each grouping to add/change what summary metrics are shown for the group:

Summary

Finish **Cancel**

Select a function for the summary.

The number of Support Cases

Total of the All Cases for This Customer

Display rounded to digits

1st option displays count of selected records.

Total obtained by

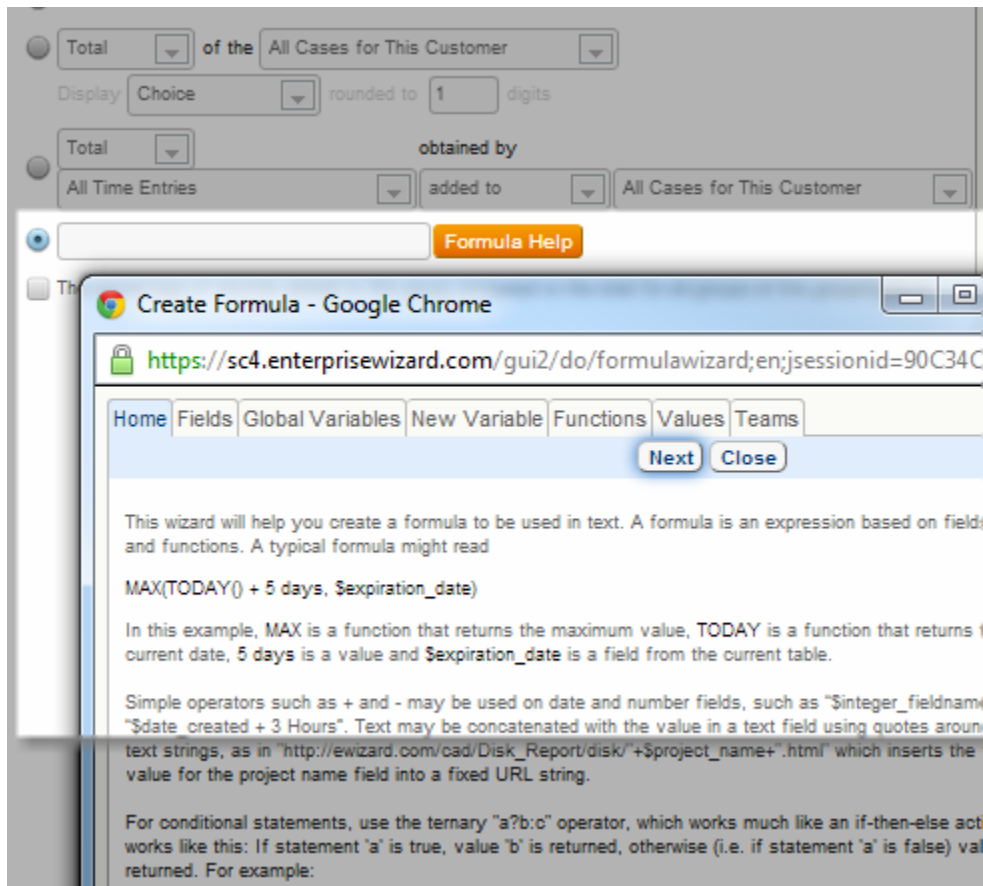
added to

2nd option displays the result of a

Formula Help

The percentage of records related to this group compared to the total for all groups at this grouping level

- If none of the counts/averages/totals provide the exact summary information you need, click the Formula Help button and define the appropriate formula using the Formula wizard



If you include several groupings, their order in the Grouping/Summary tab is important: your data will be first be grouped on the first criteria that you specify, then on the next criteria that you specify, and so on.

For each grouping, you can specify a sorting method, for instance, sort alphabetically, by number of items in the group, or using a summary value like sum (balance_due).

To select a sort method, click Charts-Reports/Grouping-Summary/New/Sorting.

See Also:

[Reports](#)

[Appearance](#)

[Saved Search](#)

[Object](#)

APPEARANCE

You can set the appearance of the report using the built-in HTML Editor available under the "Report Template" tab and/or use an existing view by clicking Charts-Reports/Grouping-Summary.

You can customize the appearance for each grouping. If you click [Edit Summary Appearance], the HTML editor will open in a separate window with a template loaded.

See Also:

[Reports](#)

[Groupings](#)

[Saved Search](#)

[Object](#)

REPORTS AND PERMISSIONS

Access to reports is controlled by group membership. Users can see the reports that have been published (Charts-Reports/Title), and users may optionally be granted the permission to create or modify reports. Users with the permission to publish reports can display the information in the desired format and have the report emailed to users who need to see it, even if those users do not have access to Agiloft itself.

Reports include a Schedule tab so that they can be individually configured for automated distribution to specified individuals or teams.

See Also:

[Groups](#)

[Field Level Permissions](#)

[Record Level Permissions](#)

[Mass Edit Permissions](#)

SECURITY

- Reports use the user's group permissions when extracting data, thus any information to which he does not have access is automatically excluded. This maintains security and of course, it also means that different users may see different data when they view a given report online or receive it via email.
- Admins can check how a chart will look to a particular user, using the Preview tab shown below:

Preview chart/report for:

Current user

User:

- If you enter a login on this tab, the Preview button shows the data that both you and the specified user are allowed to see.

See Also:

[Groupings](#)

[Appearance](#)

[Reports](#)

[Permissions](#)

[Reports Wizard](#)

CHAT

Agiloft contains a support chat interface that can be made available to end users. When enabled, it is typically displayed as a tab in the EUI. When a user requests a chat, a notification pops up for any staff on the team assigned to chat requests, asking them to accept the request.






Chat can only be initiated from End User Interface. Staff users can only reply to chat requests submitted by End Users.

To turn on Chat Interface:

Go to Setup > Chat and edit the default session

Chat

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

New  ▾		Delete 		Views  ▾			
<input type="checkbox"/>	Edit	ID ↓	Summary			Project ID	
<input type="checkbox"/>		97	Chat with our Support Staff			97	

The key options are:

- Select teams to notify when a request is received
- Decide whether to have a popup suggesting support chat when a user logs in

GENERAL TAB

This Tab is used to

- Provide a title for a session. This title will appear on the chat popup window seen by customers, so you may want a title like "Chat with a Support Rep" or "Chat about a Sales Question"
- Define a session nickname that will be displayed to staff members when a chat request is received so they know which session it is from. This should be a short phrase or word to save space.
- Select the way you want this session to be launched. It may be started directly by the user clicking a tab in the end user interface (if it is the only session launched by that tab), by a popup window that appears as soon as the user logs in to the end user interface, or from a button at your website outside of Agiloft. If you choose a tab in the EUI and have multiple chat sessions on the same tab, when the user clicks

the Chat tab, he will see a list of all available session types and can click on which one he wants to join. If you have only one session defined, clicking the tab will immediately launch a request for that session.

- Choose whether to show the chat tab or the popup only when a member of one of the responsible staff teams is logged in or only during the working hours of specific teams.

***Title:**

Chat with our Support Sta

***Session Nickname:**

Chat with our Support Sta

***Launch the Chat session from:**

- A Tab in the EUI labeled:
- Also launch a popup window upon login
- Web page outside the program

Tab (and optional Popup) should be active:

- Always
- Only if a chat staff team member is logged in
- During the working hours of the teams:

Company Team
Internal Customer Team
Customer Team
Marketing Team
Support Team
Sales Team

ACCESS TAB

In Access Tab you define what end user groups are allowed to send chat request, what staff users are permitted to reply to such requests and what groups should be notified when a customer logs into a chat session.

When a user requests a chat session, the session only begins when a staff member responds to the request. The normal sequence is: A user requests chat; staff are notified; a staff person responds; chat starts. This prevents the user from waiting in an empty chat room.

There is also the role of Moderator that has the authority to silence individuals in the chat session or remove them entirely

Show incoming chat request popup to:

All members of these teams simultaneously:

- Company Team
- Internal Customer Team
- Customer Team
- Marketing Team
- Support Team
- Sales Team

round robin to members of these teams:

- Company Team
- Internal Customer Team
- Customer Team
- Marketing Team
- Support Team
- Sales Team

Wait seconds before notifying the next person in the round robin set

the standard end user interface with a status message reading:

the alternative end user portal to the EUI Template named:

the following URL (outside of EW):

After seconds without a staff response, redirect the person who initiated the chat to:

the standard end user interface with a status message reading:


the alternative end user portal to the EUI Template named:

the following URL (outside of EW):

Moderator is

Person who starts chat

First member of group to login

User 

There is no moderator

APPEARANCE TAB

In this tab, define whether and how much of the end user's look and feel scheme is applied to the chat window. You can include just the color scheme without the header and footer html, or you can include everything.

From this tab you can also set the refresh frequency for the chat page, and the field to display as the Nickname in the chat window.

Include End-User HTML:

- Use end user's default color scheme
- Include headers and footers from that scheme

Refresh Rate

2 seconds

Choose field to display:

Full Name

Attach/Manage

1. [SampleSound.wav](#)

See Also:

Left Hand Pane Setup

TABLE IMPORT/EXPORT

Table Import/Export permits information transfer between Agiloft tables, text files and spreadsheets. The Import/Export wizard is invoked via Setup/Import. The wizard implicitly operates on the currently selected Table and provides control of the transfer format and field selection.

Setup → Import

Data Source | File Name | Data Format | Options

Next Cancel

Select the source of the data to be imported. ?

Local hard drive
The import file is stored on your local computer.

TMP directory
The import file is stored on the server in /usr/local/EnterpriseWizard/tmp.

Home Directory
The import file is stored in /usr/local/EnterpriseWizard/data/backups/demo for screenshots on the server.

Note: Use Rules or Setup/Sync to setup automatic, scheduled imports. Use Reports or Setup/Sync to automate exports.

Next Cancel

The source or target data location may be the local disk, the TMP directory or the server home directory. A table may be imported or exported directly to, or from, an Excel spreadsheet, or in tab-delimited (.txt) or comma-separated (.csv) format.

Setup → Import

Data Source | File Name | **Data Format** | Options

Please select the data format of the import file.

If you are importing from a standard SupportWizard formatted export file, choose the SupportWizard option.

If you are restoring from a backup, choose the Agiloft option.

If you are importing SupportWizard data from a .tgz file created by a SupportWizard multi-file ASCII Export or a .tgz file that includes attached files, choose the option ".TGZ file with multiple text files/attachments". This .tgz file should contain not only the data files but also any attached files that you want to be imported, stored in the /files subdirectory.

If you are importing from a Salesforce database, choose the Salesforce Data option. Other formats should be automatically selected for you.

Data Format:

- Salesforce Data
- Tab-delimited text
- .TGZ file with multiple text files/attachments
- Other text file
- Excel spreadsheet
- XML file

"Update imported fields only in matching records" option allows to update fields for matching records and append it if no existing records found. There is also another option for the Update and Replace modes: if no matching is found you can either import new record or skip it.

Data Source	File Name	Data Format	Options	Records	Fields
<input type="button" value="Back"/> <input type="button" value="Next"/> <input type="button" value="Cancel"/>					
Sheet name: Sheet1					
<p>There are five ways to import new records:</p> <ol style="list-style-type: none"> 1. You can append these records to the existing records in the table 2. You can look for matching records (using the ID or some other unique field to identify a match) and completely replace the existing record with the new record 3. You can skip matching records so that records in the import file that match existing records are ignored, i.e. the existing record is not modified at all 4. You can update the imported fields only in the matching records, leaving the other fields in the records untouched 5. You can delete all records first and then run the import (if the person running the import does not have delete permission, the records will be appended). Note that deleting all records first will greatly add to the time taken by the import 			<p>Act on Existing Records:</p> <ul style="list-style-type: none"> <input checked="" type="radio"/> Append new records <input type="radio"/> Replace matching records <input type="radio"/> Skip matching records <input type="radio"/> Update imported fields only in matching records <input type="radio"/> Delete all records before importing 		
<p>If you have chosen the second or third option above, choose the field that will identify a matching record. If you fail to include this field in your import file, the records will be appended.</p>			<p>Match records:</p> <p>By the <input type="text" value="ID"/> field</p>		
<p>Choose Create as new record mode to add records for the rows that don't have a matching record</p>			<p>If no matching record is found:</p> <ul style="list-style-type: none"> <input type="radio"/> Skip record <input checked="" type="radio"/> Create as new record 		
<p>In case there are two or more records in source file with the same value in "Match By" field, import can use all records or just the first one.</p>			<p>Duplicates:</p> <ul style="list-style-type: none"> <input type="radio"/> use all records <input checked="" type="radio"/> use first record 		
<p>Choose whether append only text fields must be overwritten with new data or not.</p>			<p>Append Only Text Fields</p> <ul style="list-style-type: none"> <input checked="" type="radio"/> Append new data <input type="radio"/> Overwrite with new data 		
<input type="button" value="Back"/> <input type="button" value="Next"/> <input type="button" value="Cancel"/>					

Field selection is accomplished via a field selection list. When Importing, Agiloft attempts to perform default mappings based on field type, but typically the user needs to manually match all import fields to existing table fields prior to import.

Setup → Import

Data Source | File Name | Data Format | Options | Records | **Fields**

Back Finish Test Cancel

Sheet name: Sheet1

Id	Sample Data	Column
1	Description	Asset Name
2	Very Low Factors	Asset Subtype
3	Hours - Very Low	Asset Tag
4	Low Factors	Asset Type
5	Hours - Low	CPU
6	Medium Factors	Current Contract End Date
7	Hours - Medium	Current Contract ID
8	High Factors	Current Contract Start Date
9	Hours - High	Current Contract Status

Back Finish Test Cancel

These tabs are only shown for users in groups with permission to Import/Export multiple records to a file. This permission setting is defined on the Permissions tab for the group. Users also need permission to 'create record' to import and 'view record' permission for export.

System displays a progress message about what sheet and row is processing during record importing. Now the user can leave the import page and view some records, then return back to the import and if the import is still in progress he will be shown a progress screen.

See Also:

[KnowledgeBase Import/Export](#)

[Import/Export Using Backup Files](#)

END USER GUIDE

END USER INTERFACE

The End User Interface provides the customer access to self help, assisted help, to their user account information and to any data that the user has been authorized to create or access.

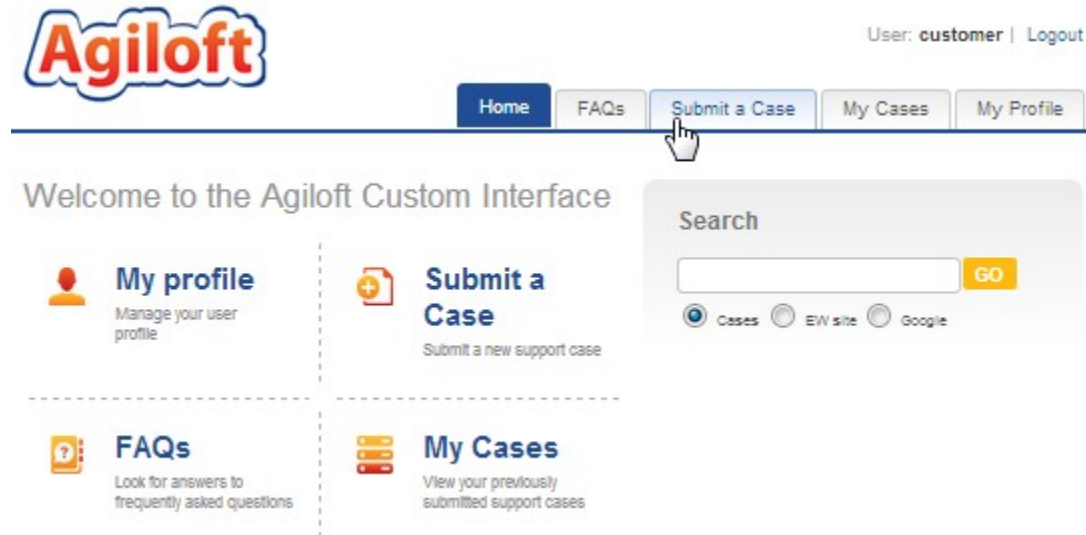
The End User Interface allows you to:

- Create new records by selecting a **New <Record Name>** tab on the Toolbar. When end-users have permission to create records in only a subtable of a main table, they will only see the name of the subtable in the New "Table" tab, and clicking that tab in the end-user interface will take them directly to the appropriate form, instead of an intermediate screen.
- View or edit existing records using the **My Items** menu.
- View or update your user account information using **My Profile** page.
- Chat with a staff representative in real time using **Chat** page.
- Browse frequently asked questions by clicking **Search FAQ** , and provide customer feedback on their usefulness (when this option is enabled).
- Receive broadcast notifications from admins about important system changes.

Depending on your user account rights, some of these features might not be available.

NEW RECORDS

Clicking on a **New <Record Name>** tab on the Toolbar opens a record entry form for that record type, allowing you to create a new entry, for example, New Question or New Contact. If several record types are available for the tab, you can select the one you need from the menu when you hover the mouse over the tab. In the example below, the user is about to log a new PC Hardware Issue.



MY ITEMS


The **My Items** menu, located on the right hand side of the Toolbar, provides access to records of the selected type, for example, questions or support contracts. Any selection from the **My Items** menu takes you to the **My Items** screen, showing the Search Block followed by a list of records that are relevant to your user account.

To view a subset of records that contain certain keywords, enter the keywords and click Find. You can also search for records IDs and specify a range of IDs, for instance 1, 3, 4-7 will return a list of records with IDs of 1, 3, 4, 5, 6 and 7.

If the table has filters (called Saved Searches), you can switch between them using the **Show** drop down list. The current filter is displayed above the list of records as shown in the example below.



Home FAQs Submit a Case **My Cases**

Saved Searches: Show all records Text:

 **Support Cases**
 Status: 9 record(s) found, 1 pages.

<input type="checkbox"/>	ID ↓	Summary	Type of Issue
<input type="checkbox"/>	432	new ticket	Enhancement Request
<input type="checkbox"/>	429	Regressive bug	Bug
<input type="checkbox"/>	424	Customer's keyboard not functioning correctly	Equipment Problem
<input type="checkbox"/>	415	problem with equipment	Equipment Problem
<input type="checkbox"/>	413	Hard Drive not accessible - BSOD	Equipment Problem
<input type="checkbox"/>	412	Printed pages have streaks	Equipment Problem
<input type="checkbox"/>	382	Reports are incorrect in Peoplesoft	Bug
<input type="checkbox"/>	356	My printer is not working although it shows that it is ready	Question
<input type="checkbox"/>	352	How can I learn how to administer the EW system?	Question

If you need to create a new filter, click **New Search** . For more information on using Saved Searches, see [Saved Search](#).

To view or edit a record, click the "view" icon  or the "edit" icon  next to the appropriate record.

To sort on a column, click the column title. To reverse the sorting order, click it a second time.

MY PROFILE

The **My Profile** tab allows you to edit your own user record. It is only shown for users whose group permissions allow them to edit their own record in the Contacts table.

SEARCH FAQ

The **Search FAQ** feature allows you to search for answers to frequently asked questions in an FAQ table. If several FAQ tables are available, hover the mouse over **Search FAQ** tab and then click the table you need on the drop-down menu.

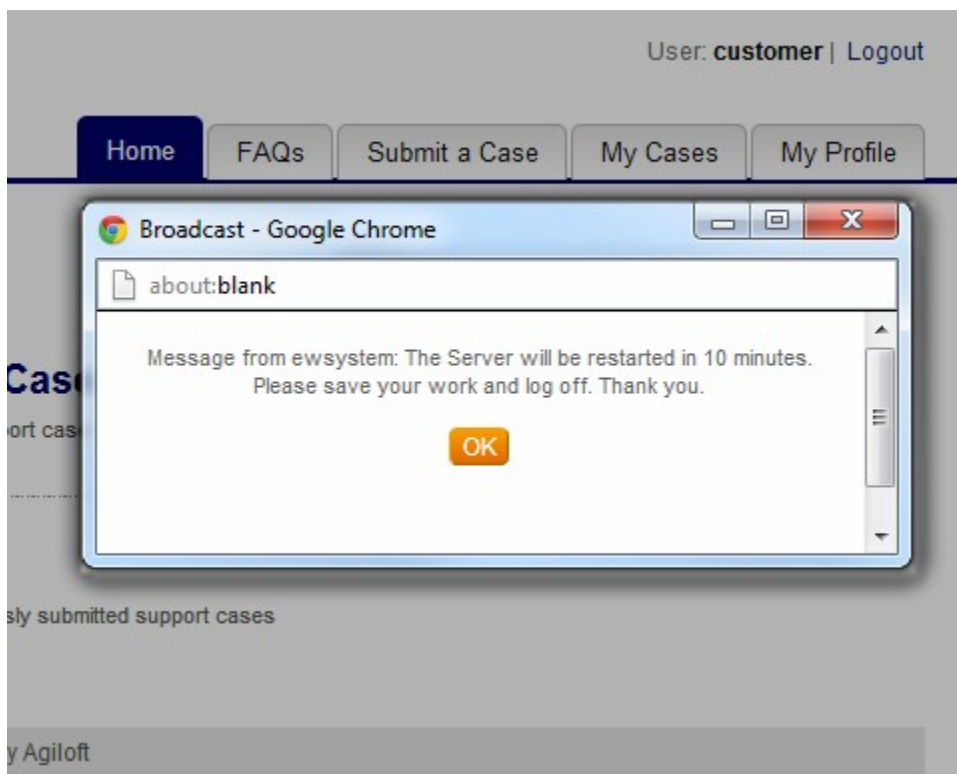
The Search Block on the Search FAQ page is very similar to that used in [My Items](#) pages. The few differences are as follows:

- **Powerful search functionality:** allows you to use 'AND' or 'OR' operators, specify an exact phrase or use natural language, which instructs the engine to find words in different grammar forms and/or their synonyms.
- **Several sorting methods:** Most Useful sorting, Best Match sorting and Newest sorting. The "Most Useful" sorting is achieved based on the feedback the answer received from other users who viewed it.

If an answer has optional Comment or Rating inputs, users can provide feedback on how useful an answer was. To rate an answer, select one of the available grades on the Rating drop-down, enter an optional Comment and click Submit.

MESSAGES/BROADCASTING

The messaging function allows the broadcast of messages to all users logged into the system. End users are notified via pop-up. The message appears to users within 1 minute of broadcast.



See Also:

[End User Interface Access and Permissions](#)

[End User Interface Setup](#)

[Staff Interface Overview](#)

ADMINISTRATOR GUIDE

ADMIN CONSOLE

KNOWLEDGEBASE

The entire collection of records, teams, groups, views, reports and the associated data.

An entire KnowledgeBase (KB) can be exported (in XML format) as a single file and imported onto another server. A given Agiloft installation may include multiple KB's, but each KB is logically independent. The system administrator of one KB cannot overwrite another.

See Also:

[Administrator Console](#)

[Licensing](#)

[Installation](#)

[Initial Implementation](#)

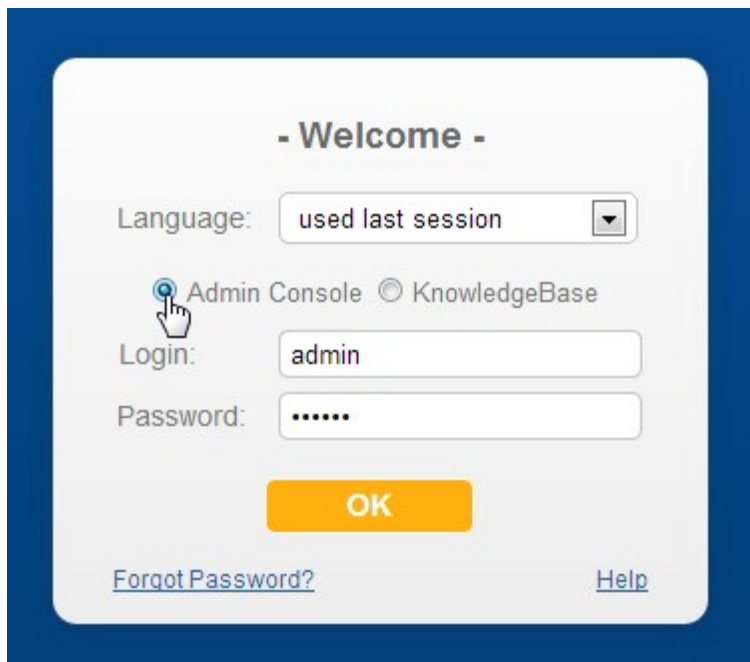
[Concepts](#)

ADMINISTRATOR CONSOLE

The Administrator Console is only available for "purchased" or dedicated server installations of Agiloft.

This console provides access to functions affecting all the KnowledgeBases on a specific server. You can create, edit or delete KB's, enable debugging for selected KB's, set the default SMTP server, create a backup schedule for each KB and set licensing that applies to all KB's. The Administrator Console also provides access to security sensitive functions that are not accessible from within a particular KB, even to the admin group.

Access the Administrator Console from the normal login screen by toggling from "KnowledgeBase" to "Admin Console".



The screenshot shows a login window titled "- Welcome -" with a blue border. It contains the following elements: a "Language:" dropdown menu set to "used last session"; two radio buttons for "Admin Console" (selected) and "KnowledgeBase"; a "Login:" text box containing "admin"; a "Password:" text box with masked characters; an orange "OK" button; and two links at the bottom: "Forgot Password?" and "Help".

The KnowledgeBases tab allows you to create a KB from the Default KB template (Setup/KnowledgeBases/New) or create a KB from the template of an existing KB (Setup/KnowledgeBases/Copy). You can also edit a KB's table structure or delete the entire KB.

The Import tab allows you to import a KB from a local file or the server. Importing a KB can result in a new KB being created, or an existent KB being overwritten.

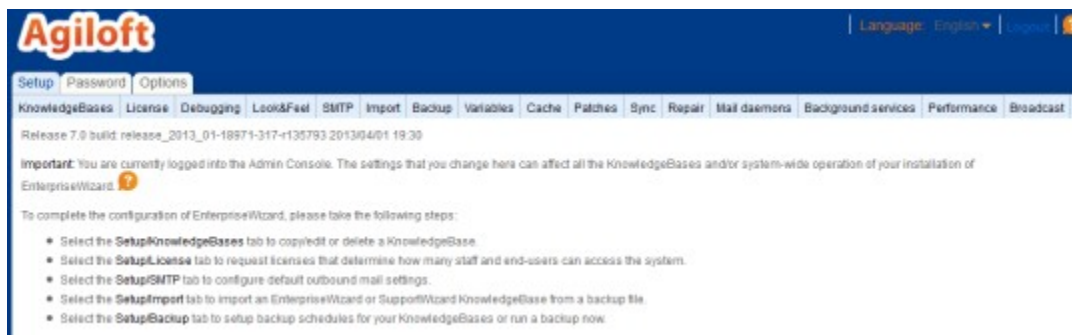
The License tab allows you to install licenses for a particular KB or for all KB's in your Agiloft installation. These licenses are used after there are no more KB-specific licenses left.

The Debugging tab allows you to turn on debugging for specific functions and specify how the debugging data should be saved.

The Server Caches tab allows you to delete the server's cache, which might be useful for debugging purposes.

The Email/SMTP tab is used to set the default SMTP server used for sending outbound email for all of your KnowledgeBases.

The Backup tab allows you to set a custom backup schedule for each KB, specifying where the backup files should be stored and how often old backup files should be deleted. You can also backup the current state of the selected KB by clicking Setup/Backup/Run Backup Now.



See Also:

[KnowledgeBase](#)

[KnowledgeBase Import/Export](#)

[Email](#)

[Licensing](#)

TABLES

OBJECT (AKA TABLE)

An object is the whole set of fields, relationships, and rules on which individual records are based.

The objects in Agiloft have a hierarchical structure, and objects inherit fields and rules from their parent objects.

Objects are implemented as tables in a relational database, so to keep things simple, the program and this manual usually refer to them as tables.

Some facts about tables:

- A Table holds records that contain similar kinds of information stored in fields with a variety of attributes (text, choice, number etc.)
- Top level tables are displayed as tabs on the horizontal toolbar.
- Tables can have multiple levels of sub-tables that inherit everything from their parent(s).
- Data from one table can be displayed in another table through linked fields in one-to-one, one-to-many and many-to-many record relationships.
- Access to tables is controlled by group permissions. New tables are only available to the admin group until permissions are given to others.
- All tables have the full range of functionality available to them: fields, layouts, searches, views, workflows, rules, charts/reports, permissions, email, etc.

For example, it is possible to define an Asset Table with fields such as Cost, Purchase Date, and Repair Date, and then a sub-table of Asset named Computer. In this example, the Computer sub-table would automatically inherit the Cost, Purchase Date, and Repair Date fields from its parent and might have additional fields such as CPU.

Unlike most other systems, Agiloft lets you define custom table types and relationships between them.

For example, you might define a Support Contract table and an Asset/Computer table. Introducing a linked field relationship would allow each computer record to contain information about its corresponding support contract.

To create a new table:

1. Go to Setup/Tables.
2. Select the table you wish to be the parent of the new table, or click Table Tree to create an entirely new table.
3. Click the New button.
4. Follow the Table Wizard to complete the process.

To edit an existing table, select it in step 2 and click Edit.

Table Appearance

The manner in which table data is displayed is controlled by the View, as described in Table View.

See Also:

[Quick How-To Guide](#)

[Field](#)

[Record](#)

[Business Example](#)

WHAT ARE VIEWS?

- The way in which multiple records from a table are displayed is controlled by a "View" that defines the display characteristics (but not the data included, which is defined by a search)
- Staff users can create their own views that only they can access, and they can switch views at any time.
- Admins can create views and apply them to teams of users, making them accessible or the default view for that table.
- Besides the possibility to choose the view explicitly from a drop-down, saved searches can replace the current view with a view defined in the search – this is particularly useful when viewing hierarchical data, as different sub-tables often have different optimum views showing different fields.
- Views can include color coding of rows based on a choice field, which can be populated by rules based on any criteria, enabling any color coding logic needed.

Table View Example:

The fields selected, their width, order and number of lines per record is defined by the view

Views are managed, created and selected from the Views drop-down

Support Cases
Status: 17 record(s) found, 1 pages [Click for details...](#)

New Mass Edit Actions

Edit View	ID ↓	Summary	Person
	424	Customer's keyboard not functioning correctly	Caldwell
	415	problem with equipment	Caldwell
	414	Repair Ticket	Caldwell
	413	Hard Drive not accessible - BSOD	Caldwell
	412	Printed pages have streaks	Caldwell
	406	SQL Server query	Caldwell
	405	Calculation with fields from the same record	Caldwell
	404	How do we login to the alternative end user portal?	Caldwell
	403	How do we login to the mobile interface?	Caldwell
	402	How do I request a license for Agiloft?	Caldwell
	382	Reports are incorrect in Peoplesoft	Caldwell
	356	My printer is not working although it shows that it is ready	Caldwell

Views Search

- New ...
- Edit ...
- Manage ...
- Personal view
- Color Coded Background by Priority
- Color Coded Font by Status
- Default End User
- Detail View
- E: Cases in Contacts
- FAQ View
- id, customer, company, Summary
- Mobile view
- Staff default
- staff home page
- staff view with time
- Staff with description
- Summary view
- Help

Closed Medium Christopher Caldwell

Views allow different individuals, teams and groups to have their own perspectives on the same data in a table.

A view may define which fields to display, the field width, field order, number of rows, coloring and the number of records displayed per page.

A Personal View and Summary View are always available. Changing your Personal View will have no effect on other users' Personal Views.

New - Invokes the Set View wizard to create a new table view.

Edit - Invokes the Set View wizard to edit the current view (the currently selected view is highlighted).

Manage - provides access to edit, delete or copy other views.

Select any other menu item to obtain an alternative view of the table.

VIEW WIZARD

Fields tab

Helps you to configure how many lines to display per record, should the edit/view icons be displayed, lets you choose the subset of fields to display and set width limit per each field.

- Show Edit icon
- Show View icon (if available)
- Show Icon Column

Max lines per record:

Max lines per linked field:

Field Name	Display	Column Width	Edit	View	Right Align
Account Number	<input type="checkbox"/>	Automatic	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Additional Notes	<input type="checkbox"/>	Automatic	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Assign to Me	<input type="checkbox"/>	Automatic	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Assigned Person	<input checked="" type="checkbox"/>	Automatic	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Assigned Team	<input type="checkbox"/>	Automatic	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Assigned Team Leader	<input type="checkbox"/>	Automatic	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Cell Phone	<input checked="" type="checkbox"/>	16	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Change Category	<input type="checkbox"/>	Automatic	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Order tab

Here you drag and drop fields in the order you want them and define the row coloring if any desired.

You can create a multi-row display by dragging fields onto a new line (below the current line). This is useful for fields such as "Solution" that may contain a lot of text.

Fields Order/Colors General Apply

Back Next Finish Cancel

Create the desired layout of the table view by dragging and dropping the field boxes displayed below. Single fields, taking the whole view width, may be positioned in the second row. Additional rows may also be created by 'dropping' selected fields after the last row. Any multi-column fields will appear without any field label.

ID	Priority	Assigned Person	Status	Date Created
Summary				
Customer Name				

Set Row Coloring Define Notification Icons

Back Next Finish Cancel

General

Here you provide a name for the view and control the number of records per screen,

*View Name

My New View

Deletable

Yes

No

Maximum View Width

70 characters

Minimum View Width

None

Same as Action Bar

0 characters

Minimum Field Width

10 characters

Records Per Page

20 ▼

Display Cell Borders?

Display borders around the cells of the table

Do not display borders

Apply

This tab is only visible to those with administrator rights - they can make the view available to other teams. In a similar way the administrator can set a view as default for another team.

Make this view visible to:

Keep settings

Change settings

View is available for me

and for teams

- Admin Team
- Agiloft Implementation Team
- Change Approval Team
- Company Team
- Contract Creator Team
- Contract Manager Team**
- Customer Manager Team
- Customer Team
- Document Approval Team
- Document Creator Team**
- Internal Customer Team**
- Internal Implementation Team
- Marketing Team
- Professional Services Team
- Repair Team
- Sales Manager Team
- Sales Team**
- Support Manager Team
- Support Team
- System Admin Team

Select all

Deselect all

and their subteams

Any new teams that are created

MANAGING VIEWS

The Manage item on the Views drop-down gives you access to all views

Agiloft Reference Manual

	Edit	View Name	Creator	Removable	Teams with Access	Default for Teams
<input type="radio"/>		Summary view	admin	<input type="checkbox"/>	Support Manager Team, Contract Manager Team, Sales Manager Team, Document Creator Team, Sales Team, Customer Manager Team, Marketing Team, Support Team, Change Approval Team, Internal Customer Team, Contract Creator Team, Customer Team, Admin Team, Professional Services Team, Agiloft Implementation Team, Internal Implementation Team, Document Approval Team, System Admin Team, Company Team	-- none --
<input type="radio"/>		Mobile view	admin	<input type="checkbox"/>	Support Manager Team, Contract Manager Team, Sales Manager Team, Document Creator Team, Sales Team, Customer Manager Team, Marketing Team, Support Team, Change Approval Team, Internal Customer Team, Contract Creator Team, Customer Team, Admin Team, Professional Services Team, Agiloft Implementation Team, Internal Implementation Team, Document Approval Team, System Admin Team, Company Team	-- none --
<input type="radio"/>		Default	admin	<input checked="" type="checkbox"/>	Support Manager Team, Contract Manager Team, Sales Manager Team, Document Creator Team, Repair Team, Sales Team, Customer Manager Team, Marketing Team, Support Team, Change Approval Team, Internal Customer Team, Contract Creator Team, Customer Team, Admin Team, Professional	Support Manager Team, Contract Manager Team, Sales Manager Team, Document Creator Team, Repair Team, Sales Team, Customer Manager Team, Marketing Team, Support Team, Change Approval Team, Internal Customer Team, Contract Creator Team, Customer Team, Admin Team, Professional

TABLE RELATIONSHIPS

Many-to-many Relationships

Suppose you need a many-to-many relationship:

- A contact may have many opportunities
- An opportunity may have many contacts

Process

1. First create an entry of type "Link to Selected Fields" with multiple values enabled in one table
2. Then create a Related Table data type in the other table.
3. Result: The display looks the same in both tables though created by different data types
4. If a record is created in the Related table, the bi-directional links are created automatically

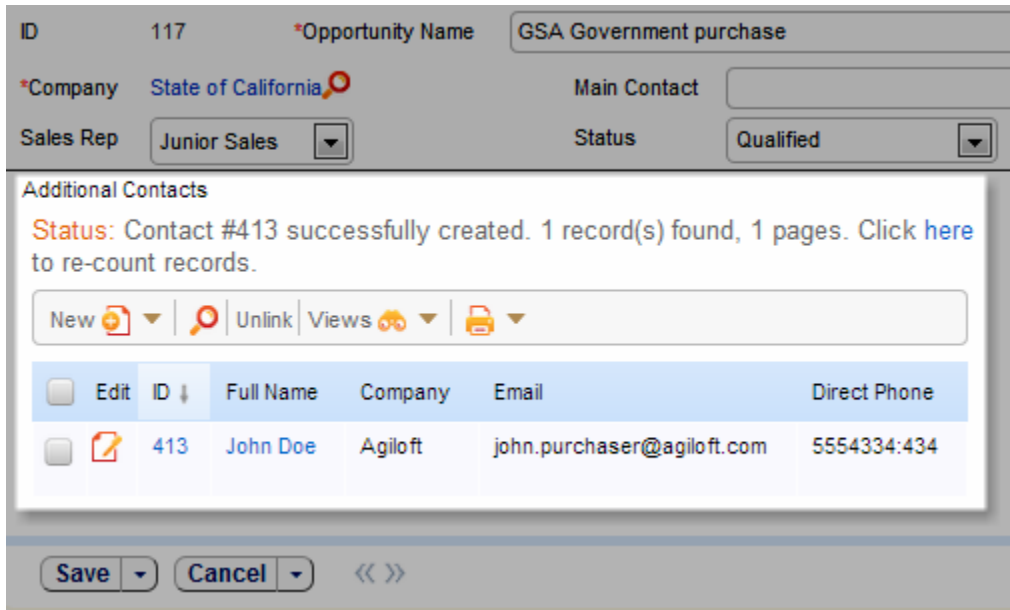
Many-to-Many Example

Here is what the resulting tables looks like.

First, here is a view of the "Opportunities" table shown in the Contact record of John Customer.

The screenshot shows a CRM interface for a contact record. At the top, there are tabs for 'Contact Information', 'Sales Progress', 'Opportunities', and 'Activities'. The 'Opportunities' tab is selected. Below the tabs, there are fields for 'ID' (413), 'Contact Table' (Contact), 'Date Created' (Apr 10 2013 08:42), and 'Date Updated' (Apr 10 2013 08:43). Below these fields is a table titled 'Opportunities'. The table has a status bar indicating '1 record(s) found, 1 pages. Click here to re-count records.' Below the status bar is a toolbar with buttons for 'New', 'Mass Edit', 'Delete', 'Unlink', 'Actions', 'Views', and 'Search'. The table has columns for 'Edit', 'ID', 'Opportunity Name', 'Company', 'Status', 'Sale Value', and 'Created By'. The table contains one record with ID 121, Opportunity Name 'New Opportunity from IBM', Company 'IBM', Status 'Qualified', Sale Value '\$450,000', and Created By 'EWSsystem'. At the bottom of the table, there are 'Save' and 'Cancel' buttons and a pagination control showing '<<' and '>>'.

Next, here is what the "Additional Contacts" table looks like in an Opportunity record:



CASCADING TABLE RELATIONSHIPS

Links can be connected through multiple tables using default values

- For instance, in a support case you might have a set of linked fields from the company table (support info), to the asset table (company assets), and the contact table (user info). The user table would have linked fields from the company table (the user's company).
- You can set up defaults for these field sets that pull in both the company contract fields and the assets based on the company field of the user who submits the record

See Also:

[Linked Fields](#)

DEFAULT TABLES IN DEMO KB

Shown below is a list of the tables and sub-tables that are typically included with a new installation. Agiloft makes it easy to add/remove tables and your custom tables will behave exactly like the default tables with all the functionality of Workflow, Business Rules, Charts/Reports. They can also be linked to other tables.

Contacts

Holds contacts who do not have a login to the system. Note that Agiloft prevents deletion of the "guest" user.

Contacts/End Users

Holds customers and sales prospects who have a login for system access.

Contacts/Employees

Holds employees and contractors who have a login for system access.

Support Cases

Holds support issues submitted by customers.

Helpdesk Cases

Holds support issues submitted by company employees.

Assets

Holds hardware, software and other assets that are tracked.

Activities

Holds activities and tasks, often related to other tables, for example an Activity might be to configure and install some new hardware.

Leads

Holds sales leads that were imported from an external source, or from potential customers who self-registered.

Companies

Holds customer, supplier and reseller companies.

Opportunities

Holds information about deals, linked to the Contacts and Companies tables.

Billing

Holds billing information, linked to other tables. For example a Support Case may contain a list of billing records that show the activities taken by each member of staff to resolve the issue and their times.

Communications

There is one Communications table linked to each of the other tables. It holds all the emails and notes for that table.

Calendar

Hold individual and team scheduled events such as appointments and calendars. May be linked to other tables to provide automatic prompts.

For more details on the default setup, please check the [Demo KB Spec](#)

When a Default Table isn't Enough

When you customize the system, you often have the choice of creating new top level tables, new sub-tables or simply adding fields to the top level table with the visibility dependant on choice field(s). The pros and cons of each of these options are explored in the "See Also" pages shown below.

See Also:

[Sub-tables vs. Choice Fields](#)

[Separate Table vs. Sub-table](#)

[Reference Other Table vs. Create Independent Fields](#)

TABLE CREATION WIZARD

This wizard is used to create tables or edit existing ones. For example, if you select Setup/Tables, select the "Table Tree" and click New, the following wizard is shown:

Company

General Graphics Fields Layout Action Bar Workflow Rules Permissions Indexes Conversion History API

Next Finish Cancel

This wizard will help you to create a new table, or edit an existing table in the Agiloft table hierarchy. ?

The label is the identifying name of the table that is seen by staff and end-users. It can be modified at any time and like all "labels" in the system, it is NLS compliant, meaning that it can be translated into multiple languages and each user will see the label in his native language. The label is also used for individual records. ...More Help

*Table Label
Company

The Plural Table Label is used when referring to multiple records in a table. ?
For example: Customer Accounts.

*Plural Table Label
Companies

By default, a Plural Table label is constructed by adding an "s" to the table label.

The internal KnowledgeBase name of the table. Staff and end users do not see this name and it cannot easily be changed once it has been set. ...More Help

*Table Name
sw_1651company

Logical Table Name
company

This option controls whether it is possible to delete this table. This option is used to protect tables that are critical to your business processes. ...More Help

Deletable
 Yes
 No

Select special search to use when subtype filtered in top tabs

Use predefined filter
 Use custom search

Create Search

Create linked Communications Table.

Linked Communications Table exists.

Omit the table label from members of the following groups when they view the table or create or edit records in it.

admin
adminimport
Anonymous
Contract Creator

Next Finish Cancel

General Tab

This tab lets you set the table's label and provides default values for the name and the plural

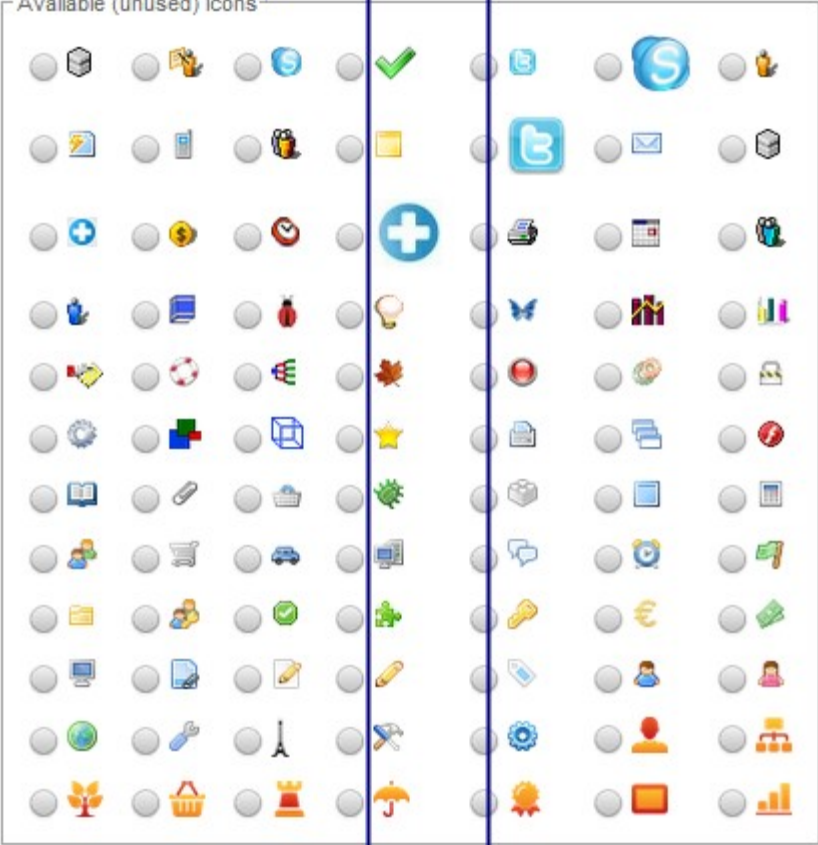
Graphics tab

This tab lets you choose or upload an icon for the table.

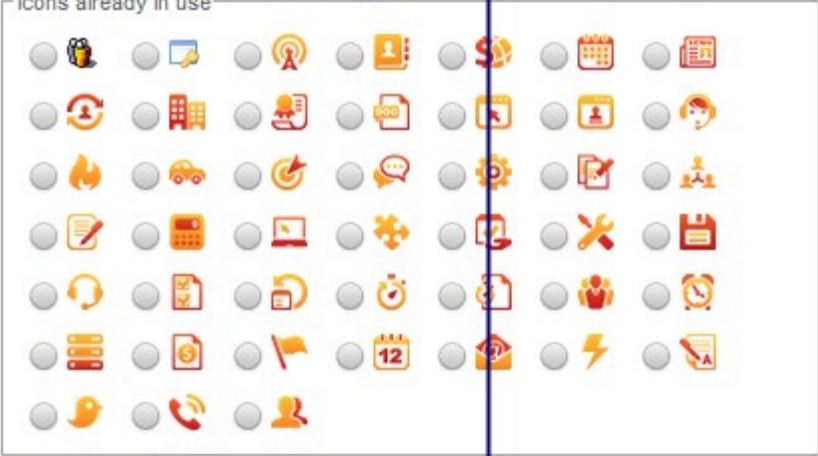
This tab displays the available icons, the used icons, and a button to upload a custom icon.

*Current icon 🏢

Available (unused) icons

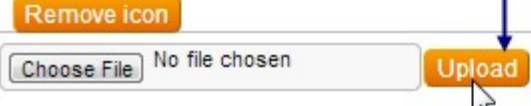


Icons already in use



Remove icon

Choose File No file chosen Upload



Fields Tab

- This is where the bulk of table work is done. Fields are listed alphabetically by label and may be created, edited, and deleted from this screen.
- Relationships between tables are created and edited here through special field types
- Creating a field does not make it show up in a form unless you both add it to the Layout and give groups permission to view it

Click New to create a field.
Click Edit to change an existing field.

Linked fields or related tables are identified in the data type column.

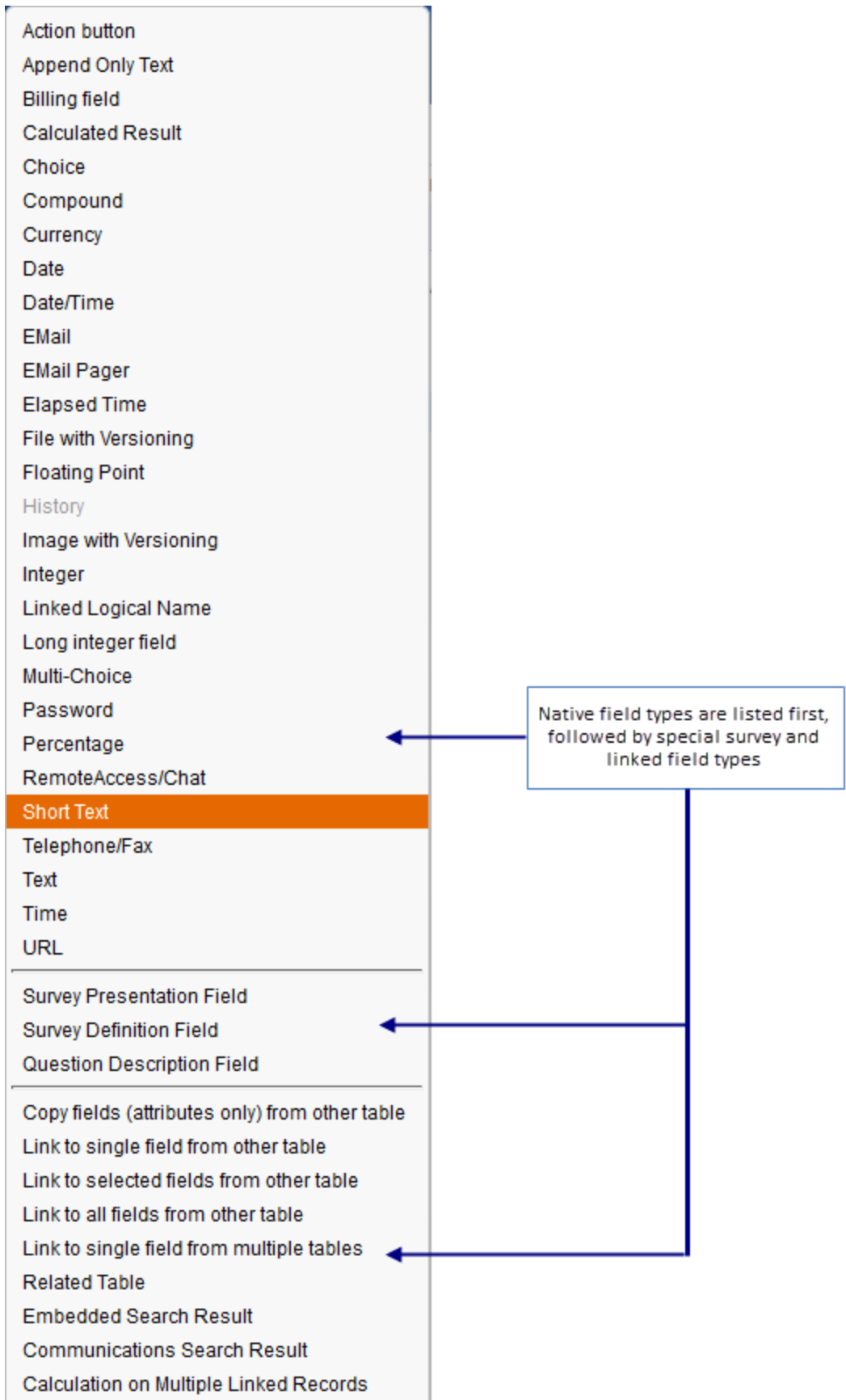
New Move Delete

Print Field Documentation

	Edit	Field label	Field name	Field DbName	Data Type	Default	Required
<input type="radio"/>		Backup Team Leader	backup_team_leader	backup_team_leader	Link to: selected fields from other table Employee: full_name/Full Name		
<input type="radio"/>		Communications	unify_messaging	unify_messaging	Communications Search Result		
<input type="radio"/>		Demo Data	demo_data	demo_data	Choice	No	
<input type="radio"/>		Description	description	sw_description	Text		*
<input type="radio"/>		Include Team in Assigned Team Dropdowns	team_role	team_role	Multi-Choice	Tasks	
<input type="radio"/>		Link to Team Leader Contacts	team_leader	full_name	Link to: selected fields from other table Contact: full_name/Full Name		
<input type="radio"/>		Owned By	owned_by	owned_by	Link to: single field from other table Contact: _login/Login		
<input type="radio"/>		Primary team members	related1217578723124	related1217578723124	Linked data from another table		
<input type="radio"/>		Team members	related1217578723233	related1217578723233	Linked data from another table		
<input type="radio"/>		Team Name	_name	sw_name	Text		*
<input type="radio"/>		Team Pager	team_pager	team_pager	E-Mail Pager		
<input type="radio"/>		type	type	type	Type of the SW object		

Back Next Finish Cancel

Types of data Fields



Each data type may have options specific to it, but many options are the same for most types:

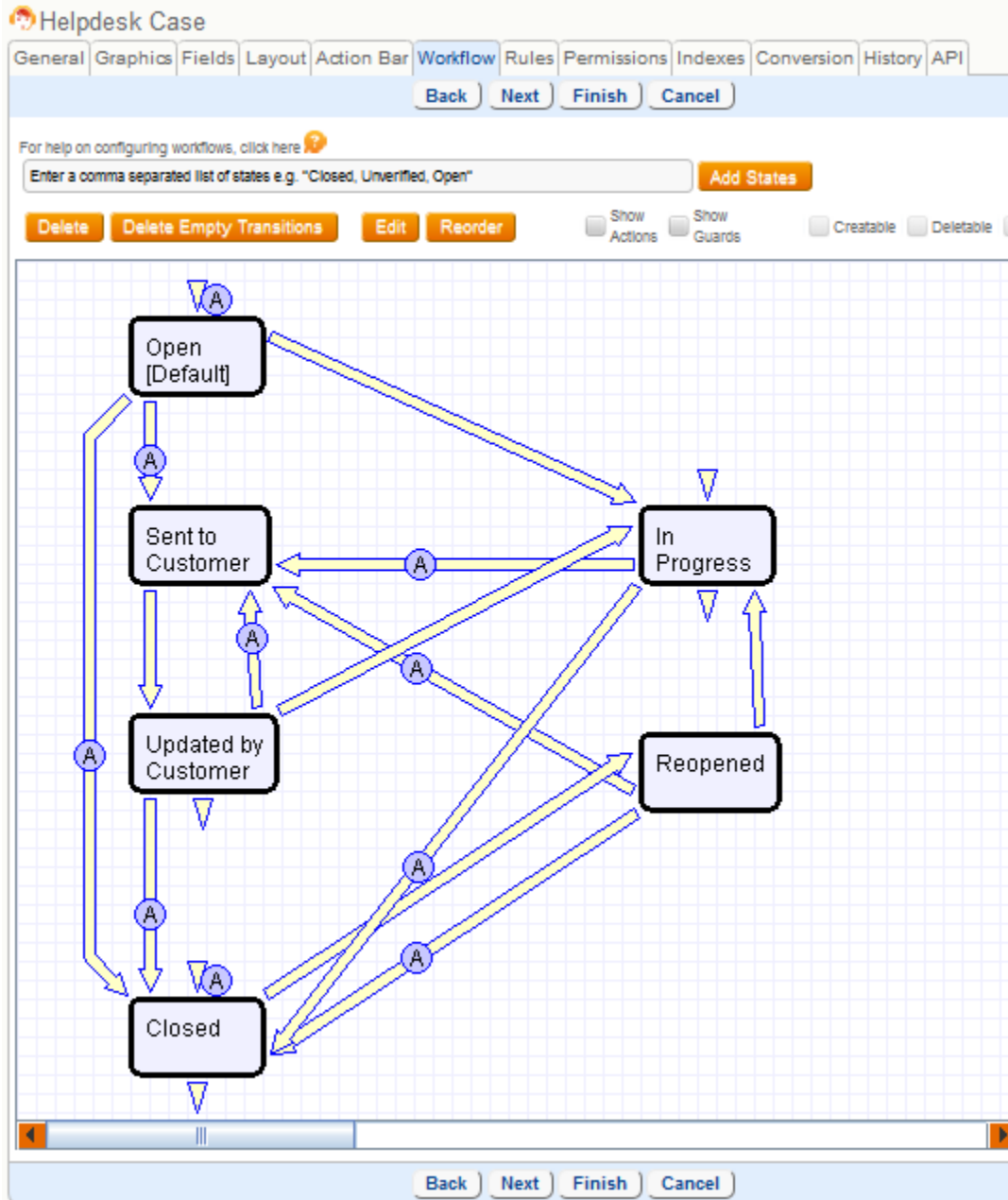
- General – Provides the name of the field and basic type information
- Options – Defines if field is dependent, required, what type of input is used
- Permissions – Allows an existing field to be copied to set permissions for all groups for new field
- Display – Defines further display characteristics, input and popup instructions and placement

Rules and Workflow Tabs

Create or view rules and workflow specific for this table directly from the Table Wizard.

The screenshot shows the 'Company' table wizard with the 'Rules' tab selected. The 'Rules' tab contains a table of defined rules for the 'Company' table. The table has columns for Edit, Rule ID, Comment, Rule Type, Conditions, Name of Action(s), and Actions Descriptions. Two rules are listed:

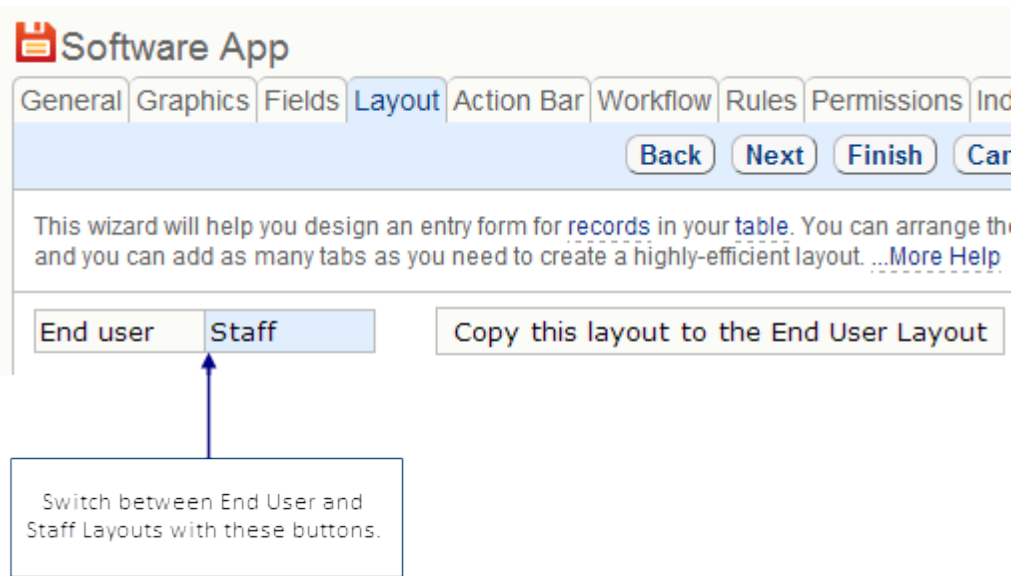
Edit	Rule ID	Comment	Rule Type	Conditions	Name of Action(s)	Actions Descriptions
	6526	TB Demo Data Update: Update date fields by one month each month so reports have data	At selected time intervals.	Saved Search: Demo Records	Update Demo Dates	Update action: Update Demo Dates Update fields: Reference as of: Original Start Date, Date Created, Date Updated New values: \$reference_as_of + 1 months, \$original_start_date + 1 months, \$date_created + 1...
	6492	Create/Edit: Copy Address to Billing	When a Company is created/edited/deleted.	Apply Rule: When a Company is Created/Edited	Update Billing from Address	If (Billing same as shipping) { Copy info to Billing address } { Update action: Copy to billing Update fields: Billing US State Long, Billing Zip or Postal Code, Billing US State, Intl State or Province, Billing Country, Billing...



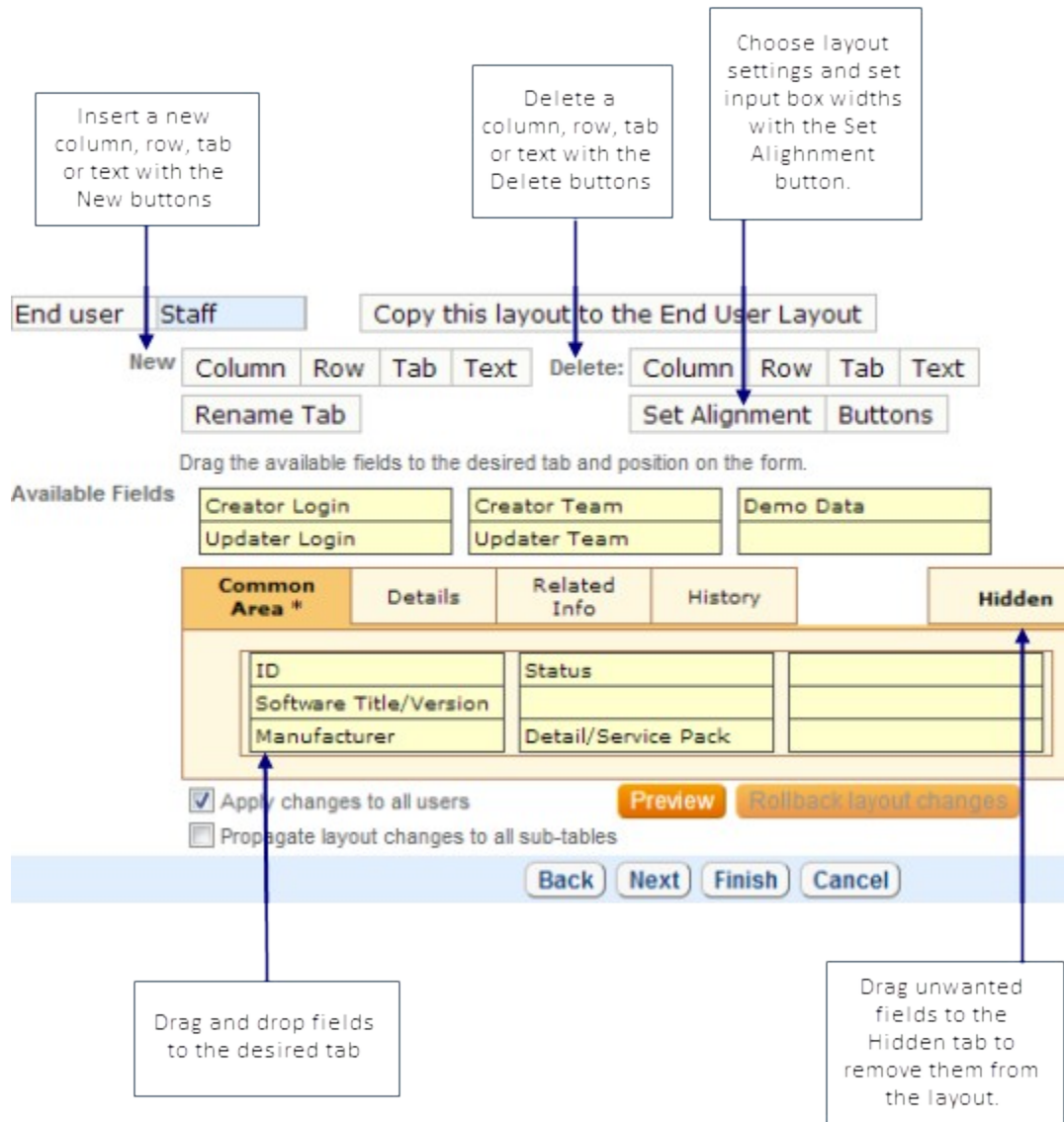
Layout

Note that fields will not appear until you add them to a layout

There are two distinct layouts for every table: one for staff and the other for the EUI:



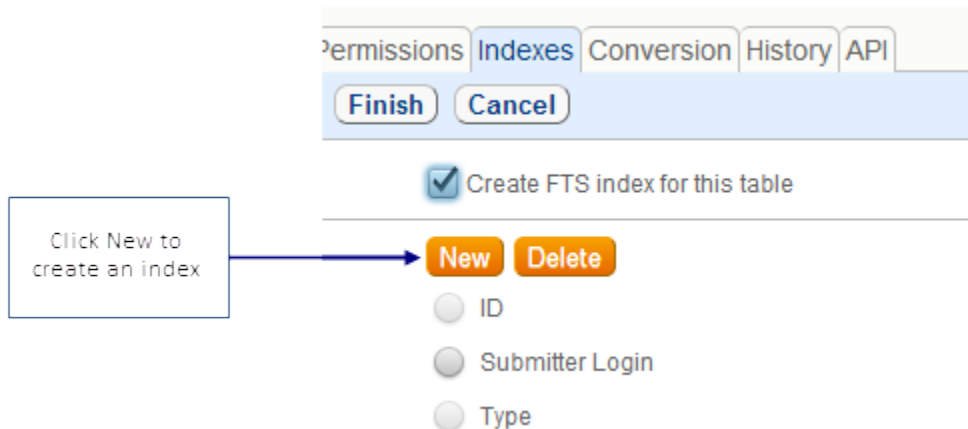
Layout Tab



Please note a special section for dragging-and-dropping fields - Common Area. This section will appear as the first tab and you can drag and drop fields into it just as you would with tabs normally. When viewing a record using the layout, any fields in the Common Area will be shown above the other tabs, near the title, rather than underneath the tabs where the other fields are shown.

A good use of the Common Area is to put critical fields that are relevant to multiple other tabs there, such as ID, customer name, etc. It is possible to put all types of fields to the Common Area, including Linked Fields, required fields, dependent fields and so on. It is possible to create headers for the Common Area.

Indexes



You can create indexes for specific fields of a table that are important so that speed will be improved when running a saved or advanced search on those fields.

It is also possible to turn off the full text indexing on tables in which it is not relevant or useful, and this will speed performance on those tables (communications tables, for instance).

To set up a field index, go to Setup/Tables/select a table/Edit/Indexes and click the New option. Typically you will want to index only a single field at a time, but it is also possible to create a compound index by choosing multiple fields. When you save the index setting, the system will take the time to index the selected field immediately.

Indexing fields such as Status and/or Assigned to, as an example, can speed the performance of saved searches based on those fields.

Record Ownership

Record ownership is defined in the Permissions Tab

Permissions Indexes Conversion History API

Finish Cancel

Permissions, for this table. You can assign granular permissions, such as create, view, edit, or delete records in this table. ?

Option:

Hide the table from all users

*Field used to define ownership

Submitter Login

*Matching field in Contacts table

Login

Which records are "My Own" ?

- Generally the owner of a record is defined as a single person, either an end user or account rep
- However, end users can only edit "their own" records, so if they need to be able to edit others' records, ownership needs to be defined more broadly – i.e., the Company field of the record matches the Company field of the user
- If "my own" is defined more broadly, you may need to add record permission filters to prevent some people from seeing all of "their own" records

Example. : Privileged end users at a company can edit others' records, but normal users should only see and edit their own – you can apply a filter in the group permissions for which of "my own" records a user can view or edit

Permissions Tab-Record Permissions

- Permissions control in this tab is not as fine as through the Setup/Access/Manage Groups menu. Do your initial permission setup here, but always refine them from Setup/Access/Manage Groups.
- This tab defines only record level permissions for the table – field level permissions are defined in Manage Groups screens

Conversion

Conversion is a very useful feature – it allows you to auto-create new records in one or more other tables, mapping values from fields in the current table.

- It can be used to convert leads into contacts, companies, and opportunities
- It can be used to generate new records from a set of base records on a regular basis (monthly to do's from a set of scheduled recurring tasks)
- It can be used to reassign records from the wrong table to the right table (i.e. if customers submit an issue to the wrong table)
- It allows you to collect information in one table and then populate it to several other tables automatically

The maximum number of possible conversion mappings is 99 per table.

Setting up Conversion Rules

- Example – let's set up conversion of our reseller orders into a company and end user record
- Keep in mind:
 - To retain links between tables, it is important to convert the source table for linked fields before the target table. For example, if the Company table contains links to the Contact table, convert Company first, then Contact if the Contact table contains a link back to fields in Company.
 - If you want to preserve links during conversion, be sure to map the linked fields with the same values from source table.
 - Conversion can be done manually by a user, or automatically via a Rule. (In a Rule, when the record meets some criteria the conversion is run). If conversion is run by a user, the new record can be edited as well as converted from the original source.

Conversion Permissions

Many customers use action buttons with conversion rules to enable staff to perform record conversion (creation of a record in another table with fields mapped from the current record). They do not want users to see the Conversion button on the menu

bar, which brings up a different kind of dialogue, but they want users to be able to interact with the rule conversion action, for instance, by being able to fill out the new record fields as part of the conversion.

We added a new permissions option on the Record permission tab of the Group permission wizard and separated conversion permissions into two options:

Show conversion button on menu action bar

Allow interaction with conversion rules

If the option 'Allow interaction with conversion rules' is checked, then all 4 kinds of rule action conversions work no matter what the user's new record create permissions are (if option 3 on the conversion wizard is selected, the new record screen appears with fields based on user's group permissions, just as it did in the previous release).

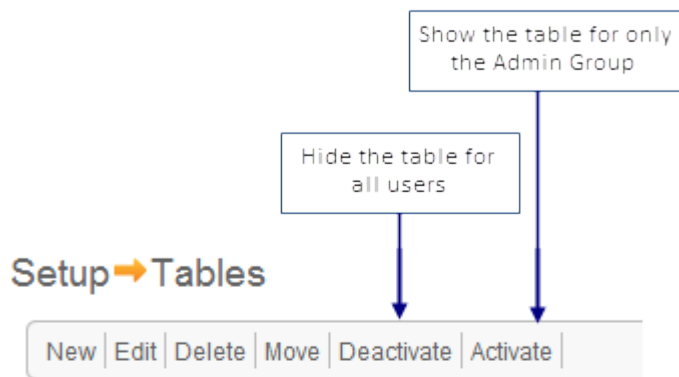
But if allow interaction with conversion rules is set to No, then when a conversion action is run by a rule, the following logic is applied, depending on the option selected for the behavior of the conversion action:

- silent conversion - runs silently, record is created in background without user's knowledge
- conversion dialogue - runs silently, creates record in background, brings up confirmation when done
- new record page - runs silently, creates record in the background, brings up confirmation when done
- automatically with confirmation - does the conversion and shows the result

Deactivate/Activate Table

Once you've added the table, you might want to hide it later at some point.

Table may be hidden from all users by using either the Deactivate button on the table tree or the option 'Hide from all users' on the permission tab of the table wizard. This option cannot be simply deselected. If a table has been hidden from all users, it may be unhidden from only the admin group by using the Activate button on the table tree wizard. To make it visible to other groups, Setup/Access/Manage Groups must be used. Admin should edit the appropriate group, select the desired table and scroll down on the Table tab to the option 'Show Table on a Toolbar Tab'.



To create a new table, select the top of the Table Tree if it

Agiloft tables are arranged in a hierarchy so that sub-table

To create a sub-table, select the parent whose properties
custom fields that you specify. To edit an existing table, s
To hide an existing table from all users, select it and click
To turn the toolbar tab on for the admin group, select it ar

- Table Tree
 - Acct Period
 - Activity Log
 - Agiloft Request
 - Asset
 - Billing
 - Calendar
 - Campaign
 - Change Request
 - Chat Log

VIEWS AND PERMISSIONS

Access to views is controlled by group permissions. An admin user can allow selected groups to create and edit views, and optionally make them visible to other users. Users with the permission to publish views can assign them to users in other teams, and set the view as default for those users. To make a view accessible and/or the default view for other teams, click Set View/Select View to Edit/Apply. When a view is assigned as default for selected teams, it replaces any Personal view that the users in those teams have defined.

To see a list of views and their access rights, click Set View/Manage. In the displayed table of available views the 'Teams with Access' column shows the teams that are allowed to use the view.

	Edit	View Name	Creator	Removable	Teams with Access
<input type="radio"/>		Summary view	system	<input type="checkbox"/>	Support Manager Team, Con Customer Manager Team, Mi Team, Customer Team, Adm Document Approval Team, S
<input type="radio"/>		Mobile view	system	<input type="checkbox"/>	Support Manager Team, Con Customer Manager Team, Mi Team, Customer Team, Adm Document Approval Team, S
<input type="radio"/>		Default View	system	<input checked="" type="checkbox"/>	Support Manager Team, Con Customer Manager Team, Mi Team, Customer Team, Adm Document Approval Team, S
<input type="radio"/>		View in Assets	system	<input checked="" type="checkbox"/>	Support Manager Team, Con Customer Manager Team, Mi Team, Customer Team, Adm Document Approval Team, S
<input type="radio"/>		Detail View	ewsystem	<input checked="" type="checkbox"/>	Admin Team, Support Manag Internal Implementation Team
<input type="radio"/>		Personal view	ewsystem	<input type="checkbox"/>	-- none --

Even if a group (or individual user) does not have the permission to create or edit views, they can still edit and save their Personal View. Views are not used for setting [record level access](#) This is controlled by [Group](#) permissions.

See Also:

[Views](#)

[Default Views](#)

[Custom Views](#)

[Groups](#)

[Teams](#)

RECORD

A record is a collection of field values.

For example a particular "Lead" record might include a name, address, and telephone number.

Each record is represented as a row in a database table.

Agiloft supports linking records between tables, for example you might create a Company table and a Support Contract table and then create fields such as "Support Contract:Type" and "Support Contract:ResponseTime" in the Company table to specify which type of Support Contract that company has purchased and what the required response time is for that contract.

It is also possible to reference records that exist in tables outside Agiloft, including databases on remote machines that support standard JDBC connections.

To create a record, mouse-over the top-level tab for the table, and choose one of the New options from the drop-down that appears.

See Also:

[Object](#)

[Field](#)

[Record Level Permissions](#)

[Search](#)

[View](#)

[Mass Edit](#)

FIELDS

A field is both a place to input a specific type of information, and the information itself that is then stored in that part of a record.

Agiloft supports fields of many different data types, such as text, single choice list, multi-choice list, integer, floating (decimal) point, telephone, email, URL, attached file, date, time, date/time, elapsed time, currency, and so on, and each field can be provided with validation rules appropriate to its type. For instance an integer field might require a value between 1 and 100. A full list of field types is provided in [Field Types](#).

Each field has a "field name" and a "field label". The field name is the name of the column in the database table. The field label is the value that users see through the GUI. Labels can contain spaces, for example Customer Name.

Agiloft provides full National Language Support, meaning that field labels can be translated into multiple foreign languages so the user will see labels in the language specified by their browser settings.

To review the fields for a particular Table and add, modify, or delete fields, go to Setup/Tables/, select the Table to edit, then select the Fields tab. Mouse-over "New" to view the drop-down of all available data types to choose from.

A new progress window with progress indicators will pop-up when an administrator adds a new column to a table. This window will give administrators better indications of duration and progress of this operation.

There is a set of Default Fields for each table:

New Move Delete				
Edit	Field label	Field name	Field DbName	Data Type
	Communications	unify_messaging	unify_messaging	Communications Search Result
	Created By Creator Login Creator Team	_1880_full_name creator_login creator_team	_1880_full_name creator_login creator_team	Link to: selected fields from other table Contact: full_name/Full Name _login/Login _106_sw_description/Primary Team
	Date Created	date_created	date_created	Date/Time
	Date Updated	date_updated	date_updated	Date/Time
	Deletable	deletable	deletable	Choice
	Demo Data	demo_data	demo_data	Choice
	History	history	history	History
	ID	id	id	Auto-Increment
	Type	type	type	Type of the SW object
	Updated By Updater Login Updater Team	_1888_full_name updater_login updater_team	_1888_full_name updater_login updater_team	Link to: selected fields from other table Contact: full_name/Full Name _login/Login _106_sw_description/Primary Team

See Also:

[Field Types](#)

[Object](#)

[Record](#)

[Field Level Permissions](#)

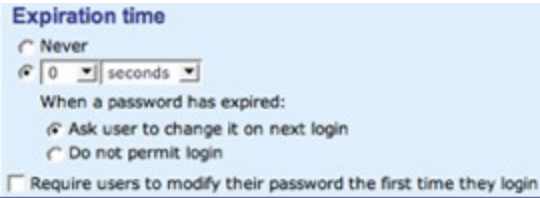
[Linked Fields](#)

FIELD TYPES

Agiloft supports a wide range of built-in field types, and it also allows you to create your own complex fields by uniting several fields, linking the fields to other fields in the same or different table or imposing a specific format on a built-in field type. Each of the available field types is described below:

Field Type	Description
Action Button	<p>This field type is actually a button placed on the page that performs a specified action, such as running a script, updating a record or opening a URL.</p> <p>Action buttons can do almost anything. For example, an action button can take the contents of the current record, pass them to a script and open a new browser window at the URL returned to the script. This URL might show the status of a FedEx shipment or provide directions to a company using Google maps.</p>
Append Only Field	<p>This is a text field that date/timestamps the notes and includes the user name every time the record is saved. Each time you open the record for editing, the previous notes appear before the textbox and are read-only, and the additional notes textbox is blank for new entries. So you can always add new notes, but not edit previously saved notes.</p> <p>When you're setting up a field like this, you can choose to display the entries in chronological order with the input box at the bottom, or in reverse chronological order with the input box at the top.</p> <p>A field like this is useful for including only the "latest additional notes" in an automatic email. For example, in an exchange where a tech and a customer communicate via the additional notes field, automated email can be set up to send only the "latest" update to the notes in the email instead of the entire correspondence.</p>
Billing	<p>There is a default billing table generally hidden from view, available to all tables. Adding a field of type Billing Field to a table pulls the fields from the Billing table into the table as a set of linked fields with multiple values enabled, so users can create billing records from within the current table. It also adds two calculated fields to hold the total billable and non-billable hours. Access permissions to the underlying billing table are required</p>

	for users to create billing records from another table.
	Billing Enhancements coming: auto-timer, auto-creation of billing records in background, more complex billing table structure allowing mapping of fields from parent tables into billing sub-tables so that reports run from billing table can include all relevant information from parent records
Choice	Choice fields provide the user with a predetermined set of choices. An example of a choice list is the list of countries or Yes/No. You can select an existing choice field or create a new choice fields by selecting this field type. You can edit existing choice fields and see where they are used using Setup/System/Manage Choice Lists.
Compound	Compound fields store the result of joining some fields together and displaying them as a single unit. An example of a compound field is the Full Name field, which is the compound of Title, First Name, and Last Name.
Currency	A currency field stores some money value in one of the national currencies;for instance \$30.65 is interpreted as thirty US dollars and sixty-five cents.
Date	A date field stores a date in one of the selected formats. An example of a date field's value is May 1, 2004.
Date/Time	A Date/Time field stores some date and specific time on that date. An example of a date/time value is May 1, 2004, 3:30pm.
Elapsed Time	An Elapsed Time field stores the amount of time. An example of the elapsed time field's value is 3 days, 8 hours, and 24 minutes.
Email	An Email field stores one or more e-mail addresses. E-mail fields can be configured to check e-mail addresses for validity. For example, you can require that all e-mail addresses contain the "@" character in them.
Email Pager	An Email Pager field stores the e-mail address of a pager. E-mail sent to a pager is sent as plain text, and all HTML formatting (if any) is automatically discarded.
Floating Point	Floating Point fields are used to store decimal numbers, such as 2.71828183

File	File fields are used to store attached files. For example, a sales proposal or a snapshot of a screen might be attached to a record.
History	A history field is created for every new table automatically. It is critical to store data used by reports and rules and you should NEVER delete it. This provides a full audit trail of all changes made to a record with a snapshot capability that shows what a record looked like after any set of changes.
Image	Image 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.
Integer	Integer fields are used to store integer numbers, such as 7,194 or 2435.
Multi-Choice	Multi-Choice fields are like choice fields, but allow the selection of 0 or many items from the list. An example of a multi-choice field is Additional Groups, you might use a multi-choice field to describe someone's Skills.
Password	<p>A Password field stores passwords encrypted using one-way hash encryption. Passwords are shown as asterisks in the GUI. Administrators may set an option on password fields that will require the user to reset the password the first time the user logs in.</p> 
Percentage	A percentage field can be useful to estimate the likelihood of a deal closing or completeness of the assignment. Note, that if user enters a % in the input box, it will be omitted.
RemoteAccess/Chat	RemoteAccess/Chat field is type of Action Button (see above) that allows you to either create a Remote Desktop connection or to start a Chat session from WITHIN the record view. For example: imagine that for 90% of the cases before updating the record you want to be able to verify some important details with your boss. Add a chat button to your layout and you can talk to anyone without leaving the Edit dialog

Short Text	A Short Text field stores up to 255 characters of text.
Telephone/Fax	A Telephone/Fax field stores a phone or fax number using a predefined format.
Text	A Text field can store at least 4,000,000 characters of text, the actual limit may be higher, depending on the underlying database.
Time	A Time field stores the time of day, for example 5pm or 3:30am.
URL	A URL field stores a web site address using http://... format.
Survey Presentation Field	Stores an instance of a questionnaire or survey for a given record. It is used to define which survey definition applies, to display a set of questions defined by the selected survey definitions, and to store the combined questions and answers.
Survey Definition Field	Allows users to select the questions for a particular questionnaire, define their order, visibility dependencies, and requirements.
Question Description Field	Allows users to define questions and answer types in a normal table record.
Copy fields (attributes only) from another table	<p>You can copy one or more fields into the current table from another table in this or any other Agiloft KnowledgeBase. This is useful for quickly adding fields that you have already created in another table or fields from built-in tables designed with this purpose in mind. This operation copies the fields with their attributes, but does not import any data or maintain any relationship with the source table.</p> <p>To copy fields from a table in another Agiloft KnowledgeBase, you must be a member of the admin group.</p>
Link to single field	A "Link to single field" links to only one field in another table. We recommend using "Link to selected fields" for most cases, as Selected Fields can be expanded in later editing, whereas Single Fields can not.
Link to all fields in another table	This field type links to all of the fields in the selected table. See Linked Fields and Reference Other Table vs. Create Independent

	<p>Fields for more details.</p>
<p>Link to selected fields from another table</p>	<p>This field type links to the selected fields from two or more tables. See Linked Fields and Reference Other Table vs. Create Independent Fields for more details.</p>
<p>Link to a single field from multiple tables</p>	<p>This field type links to the selected fields from two or more tables. An example of such a field is the field "Assigned To" which can either be an individual member of the User table, or a Group (i.e. a member of the Group table).</p>
	<p>It is possible to construct a multi-choice field from different data types, for example an integer, a text field and a floating point field. In such a case, it is represented as a text field.</p> <p>See Linked Fields and Reference Other Table vs. Create Independent Fields for more details.</p>
<p>Related Table</p>	<p>This field type shows one of the above link fields from the other direction. For example, you may have a field such as Assigned To in the Cases table that links to the Login field in the user table. This field type will allow you to create a table within each User record that shows all the Cases that the user is Assigned To. See Linked Fields for more details.</p>
<p>Embedded Search Result</p>	<p>The Embedded Search Result data type is a field that displays the results of search on some table. For example this can automatically display records in the Standard Solutions table based on keywords entered in the current record. The field allows a special filter to be configured to limit the records being shown by some criteria. This filter allows usage of fields from the "parent" table (the one where the Embedded Search Result field is located) in the filter conditions. This filter is not otherwise accessible from Left Panel or via Active Saved Searches for security reasons.</p>
<p>Calculation on Multiple Linked Records</p>	<p>This field provides the result of a calculation of two or more numerical fields.</p> <p>Fields may be from other tables. If a field from another table is used, Agiloft creates a link relationship. If the field in the other table is modified, the data in existing tickets or objects will update dynamically.</p> <p>Important! If you want calculated data to remain stable, all needed fields should be added to the current</p>

WMI (Windows Management Instrumental)

table using a link relationship.

Example: This could be used to provide a quote for a service call based on the technician's hourly rate and the time scheduled, or to calculate the sum due for work on a particular case.

This field type enables the user to capture various kinds of information from a Windows based user that can be accessed through the WMI specification.

The implementation uses ActiveX controls in IE for accessing the WMI classes and properties. Because of the ActiveX technology used, the properties can only be accessed through the Internet Explorer browser and are not applicable for browsers that don't support ActiveX controls (like Firefox).

To set up a WMI field, go to Setup/Tables/choose the table/Edit/Fields, and select New/Windows Management Instrumental Tech data type. Fill out the wizard to determine the classes and options you want to use.

The sample knowledgebase includes a table named WMI Sample that contains common WMI field definitions and these can be copied into your current table using Setup/Table/Edit Table/Fields/New:Copy Fields (attributes only) from other table

Please note, that if WMI fields are hidden by dependency conditions, they do not run the corresponding Active X components to poll the user's machine until they become visible.

Please read more about WMI on [here](#)

MISC TABLES/FIELDS CONSIDERATIONS

- Editing the linked field settings for a subtable in a hierarchical table setup will change the settings for all other subtables in that hierarchy.
- Fields cannot be converted from one data type to another. However, it is usually possible to create a new field of the desired data type and use the Mass Edit "Formula" feature to assign the old field's values to the new one.
- The "Type" field is only a label. When building dependent choice fields the "Type" field can not be the parent for a dependent choice field. For example, you can not have some options display if the subtable is SubTableA, and display other options if it's SubTableB.

- It is possible to set dynamic field defaults based on occurrence of results of changes in other fields values, for example a date field might be set to the date that another field value changed. However, these are sometimes counter-intuitive and better handled via rules instead.
- You can delete any fields or tables not marked as undeletable. Often it is safest to just hide fields via permissions or layout instead, because deleting fields used by searches or reports may break them. If you try to delete a field used by linked fields you will get a warning wizard from the Integrity Manager that walks you through the deletion process.

See Also:

[Field](#)

[Object](#)

[Record](#)

[Field Level Permissions](#)

[Linked Fields](#)

Surveys and Questionnaires

CHOICE LISTS

A choice list is used to store a pre-defined set of values either in the form of radio buttons or as a drop down list. A choice list may be as simple as Yes/No values, or a single value from a choice of values (e.g. the severity of a problem), or a multi-choice list where multiple values may be selected (e.g. a list of states in which the company has service centers).

A field can be based on a choice list as a single-choice or multi-choice value. A single value field just has one value from the list, where as a multi-choice field can have multiple values.

Choice lists can be used to drive dependant fields that are only shown if the choice is set to a particular value. For example, Steps to Duplicate Bug might be shown if the Category choice field is set to Bug, but not if it is set to Enhancement Request.







Choice lists can also control the values shown for other choice lists. For example, the list of Version Numbers might be different if the user chooses Product A rather than Product B. These are referred to as Hierarchical choices.

Example:

Set items for the choice list:

New **Delete** **Hide** **Unhide**

Choice List Items:

-  Opportunity
-  Contact
-  Support Case
-  Helpdesk Case
-  Change Request
-  Asset

Managing Choice Lists

Choices

Status: More than 100 record(s) found. Click [here](#) to count records..

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

<input type="checkbox"/>	Edit	Choice ID ↓	Choice Name	Used In
<input type="checkbox"/>		15887	document_priority	Documents:Priority,
<input type="checkbox"/>		15885	audiences	Documents:Audience,
<input type="checkbox"/>		15884	document_formats	Documents:Published Format,
<input type="checkbox"/>		15883	document_subtypes	Documents:Document Subtype
<input type="checkbox"/>		15882	document_types	Documents:Document Type,
<input type="checkbox"/>		15881	document_access_levels	Documents:Document Access,
<input type="checkbox"/>		15880	working_status	Contacts:Working Status,
<input type="checkbox"/>		15879	faq_topics	Helpdesk Cases:FAQ Topics,
<input type="checkbox"/>		15878	time_entries_related_to	Tasks:Time Related to,Time...
<input type="checkbox"/>		15876	choice_test	

Admin users may modify choice lists via Setup/System/Manage Choice List

This brings up a table that includes a Used In column which shows table/field uses for each choice list

If you put your choices in a reasonable order, they help records be searched more efficiently. For example if you chose Critical, Very High, High, Medium and Low for your values in a field named Priority then a subsequent search for Priority > =High would result in the selection of Critical and High Priority records only. It is the order of the choices in the list, not their labels that drives this behavior.

Clicking the edit icon for a choice list will bring up a wizard that allows values to be added or removed from the list and existing entries can be re-ordered using 'drag and drop'.

Existing entries can be changed by editing the choice value.

If a Choice list is deleted, then fields that used it must be deleted to maintain system integrity. The Integrity Manager wizard will systematically review each instance and request that you confirm each deletion is acceptable.

See Also:

[Field](#)

[Record](#)

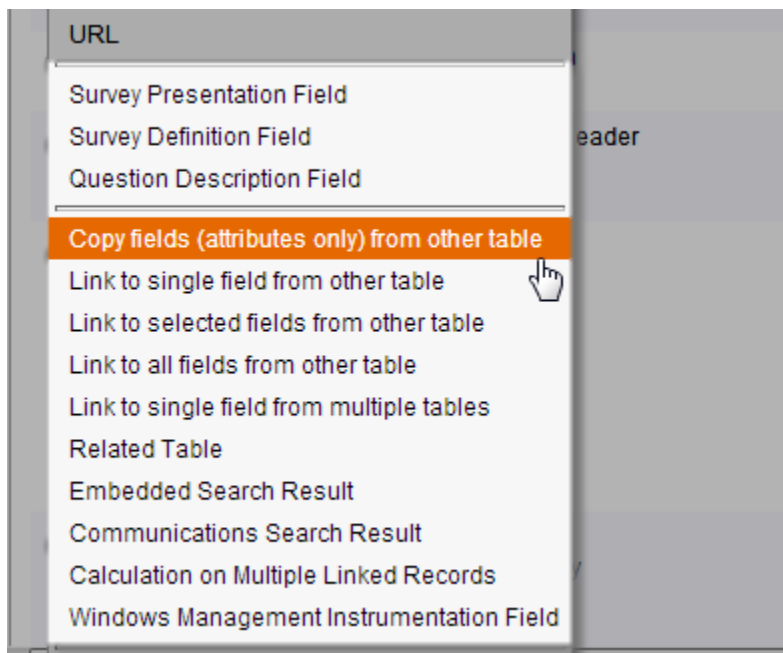
[Object](#)

LINKED FIELDS

Linked fields is a special data type that automatically pulls the values in the fields from one table (called "source table") into another table. Linked fields help avoid duplication and ensure that records dynamically reflect any changes that occurred in other places within the infrastructure. For example, a Case table would include linked fields from a source table, typically the User table, which provide access to the submitter's full name, telephone number, email address etc. from the values in his User record.

Linked fields can be configured to automatically update when the value in the source table changes or to save the old value. In either case, they retain the display attributes, permissions, and size restrictions of the source fields. You can copy or link fields from another record in the current table, another table within the current KnowledgeBase or even an external database. Fields that have been linked from another table are marked as "Link to" in the data type column (Setup/Tables/select Table to edit/Fields).

Several different kinds of Linked fields appear at the bottom of the "New" drop-down on the Fields tab in the Table wizard (Setup/Tables/select Table to edit/Fields).



- **Link to single field.** Links to one field in another table. Seldom used because you will often find you want to add more fields later and have to redo it.
- **Link to selected fields.** The most common way to pull in fields from a single record in another table (i.e. user information in a case or asset). Can also be used to pull in multiple fields from records as part of a multi-to-multi relationship. In this case it is displayed as an embedded table rather than as separate fields.

- **Link to all fields.** Pulls all fields in and if new fields are added to source table they are automatically added to the set of target fields.
- **Single field from multiple tables.** Used for the Assigned To fields to allow an issue to be assigned to a Team or an individual (Team: Team Name and Employee: Full Name) and for other situations in which a single field should display values from more than one table.
- **Related table.** Shows one of the above relationships from the other direction. For example, when the Assigned To field is used to assign each Issue to a user, a Related table shows all the Issues that were assigned to her.

See Also:

[Link to Single Field](#)

[Link to Selected Fields](#)

[Link to All Fields](#)

[Link to Single Field from Multiple Tables](#)

[Related Table](#)

[Linked Fields Display](#)

[Linked Fields Access Permissions](#)

[Linking to Other Databases](#)

[Copied Fields](#)

[Fields](#)

[Field Level Permissions](#)

[Flash Tutorials](#)




Linked Fields Wizard

Table Tab

You can import one or more fields into the current table from another table within the current KnowledgeBase, another Agiloft KnowledgeBase or an external database table. You can also import fields from another Agiloft KnowledgeBase table or from an external database table.

In this tab you specify the source table for your record. For another Agiloft KnowledgeBase, you must provide a login and password for a member of the admin group. To connect to an external database, you must add a database descriptor to the application server.

Table	Mapping	Fields	Permissions	Options	Display
<input type="button" value="Next"/> <input type="button" value="Cancel"/>					
<p>You can import one or more fields into the current table from another table within the current KnowledgeBase, another Agiloft KnowledgeBase or an external database table. ...More Help</p> <p>You can also import fields from another Agiloft KnowledgeBase table or from an external database table. For another Agiloft KnowledgeBase, you must provide a login and password for a member of the admin group.</p> <p>To connect to an external database, you must add a database descriptor to the application server. </p>		<p>Select table:</p> <p><input checked="" type="radio"/> Agiloft table in the current KnowledgeBase:</p> <p>Acct Period <input type="button" value="v"/></p> <p><input type="checkbox"/> Include fields from sub-tables</p> <p><input type="radio"/> Table from another database:</p> <p>Database Name: java:/SWDictionaryDS <input type="button" value="v"/></p> <p>Login: <input type="text"/></p> <p>Password: <input type="text"/></p> <p><input type="button" value="Get Tables"/> <input type="button" value="v"/></p> <p>Primary Key field: <input type="button" value="v"/></p>			
<p>Use the Admin Notes field to put your comments about the purpose and details of the field's configuration and usage.</p> <p>These notes can then be included in the automated Field Documentation printout.</p>		<p>Admin notes</p> <div style="border: 1px solid #ccc; height: 100px; width: 100%;"></div>			
<p>If this checkbox is selected the Admin Notes shown above will be included in the automated Field Documentation printout.</p>		<p><input checked="" type="checkbox"/> Include in Field Documentation</p>			
<input type="button" value="Next"/> <input type="button" value="Cancel"/>					

Mapping

There are various options on how exactly you can draw the record from another table.

First thing you need to think about is how many records you are going to store in your linked field.

Typically imported fields are drawn from a single entry in the source table. For instance, the contact information fields are populated from a single user record. However, you can allow the

values from multiple records to be populated into the imported fields. For instance, you might have an entry in the Contacts table named "Opportunities" that links to all the Opportunity records related to that Contact.

The wizard provides several options.

Allow the imported fields to hold multiple values from the Acct Periods table?

Yes, fast search

Yes, save space

No

Choosing Yes will enable the values from multiple source records to be placed in a single Contact.

When you enable multiple values, the imported records are shown in a table view rather than as individual fields, similar to the display used for a "Related Table" field. The difference between enabling multiple values here and choosing a Related Table field type is that each individual source record must be manually selected here. A Related Table automatically shows all records from a source table that contain links within them to the current table record (for instance, all assets owned by the selected Company). So you will generally only use the current data type if you want to be able to manually select specific records from the other table (for instance, you want to have multiple assignees for a record) or if you are setting up a many-to-many relationship (such as many opportunities for each contact and many contacts for each opportunity).

If you request "Fast Search" the values are actually copied into the record so that the Full Text Search engine can index them. The Fast Search option is generally not recommended because it greatly increases the table size and can result in the size of individual records exceeding limitations in the underlying database.

Next you need to decide whether you want users to directly enter the data into the field or you want them to look up the record from the source table.

Allow entries not in source table

-
- Do not update
 - Update matching fields
 - Automatically Update in the background
 - Automatically Update synchronously

If option "Allow entries not in source table" is enabled, the user may enter text directly (including data that does not correspond to any entry in the source table) or may select a record from the source table. If it is disabled, the user must pick a value from the source table.

Update section of this tab gives you different options on the way synchronization will work for your record.

Do not update option. The values of linked fields in target records are not updated when their source record changes. This option is computationally efficient. If necessary, the records can always be updated using Mass Edit.

Update matching fields The values of linked fields in target records are updated when the source record changes for those fields records whose old values match the old field values in the source record. In other words, if the user manually entered different data for certain fields, the manual changes will not be overridden. This option can be computationally expensive.

Always update All linked fields in target records are updated when their source record changes, including fields whose values were set manually. This option is not recommended because it can be computationally expensive and overrides manual changes made when the data was entered.

Fields

On the Fields tab you actually choose the fields to be copied.

Do not worry if you get an error message about a field already existing. You cannot have two fields with the same label in a given table, but you can change the field labels that will display on the table you are importing the fields into by typing into the text boxes on the right.

Field name	Field label
<input type="checkbox"/> Accounting Period	Accounting Period
<input type="checkbox"/> Created By	Created By
<input type="checkbox"/> Creator Login	Creator Login
<input type="checkbox"/> Creator Team	Creator Team
<input type="checkbox"/> Date Created	Date Created
<input type="checkbox"/> Date for Reporting	Date for Reporting
<input type="checkbox"/> Date Updated	Date Updated
<input type="checkbox"/> Deletable	Deletable
<input type="checkbox"/> Demo Data	Demo Data
<input type="checkbox"/> ID	ID
<input type="checkbox"/> Type	Type
<input type="checkbox"/> Updated By	Updated By
<input type="checkbox"/> Updater Login	Updater Login
<input type="checkbox"/> Updater Team	Updater Team

[View Field Definitions in Acct Periods Table](#)

Permissions

The Permission tab is pretty standard although it has a specific bulk option to copy permissions from the source table for the copied fields.

To view these permissions, click View Permissions. You have the option to replace these permissions with those copied from a field in the current table. To copy the permissions from a field in the current table, select it and click Copy Permissions.

The Copy Permissions operation sets the permissions for all of the selected fields. If you want to assign different permissions to some of the fields, you will need to go to Setup/Access/Work with Groups and edit the group permissions directly for the affected groups.

Allow group to view the field in their own Contracts:

A dropdown menu with a light gray background and a dark gray header. The header contains the text 'admin' and a small upward-pointing triangle. Below the header, the menu is open, showing four options: 'adminimport', 'Anonymous', and 'Contract Creator'. The 'adminimport' option is highlighted with a dark gray background. To the right of the options is a small icon of a list with three horizontal lines, and a downward-pointing triangle at the bottom.

Allow group to edit the field when creating new Contracts:

A dropdown menu with a light gray background and a dark gray header. The header contains the text 'admin' and a small upward-pointing triangle. Below the header, the menu is open, showing four options: 'adminimport', 'Anonymous', and 'Contract Creator'. The 'adminimport' option is highlighted with a dark gray background. To the right of the options is a small icon of a list with three horizontal lines, and a downward-pointing triangle at the bottom.

Allow group to change the field when editing their own Contract's:

A dropdown menu with a light gray background and a dark gray header. The header contains the text 'admin' and a small upward-pointing triangle. Below the header, the menu is open, showing four options: 'adminimport', 'Anonymous', and 'Contract Creator'. The 'adminimport' option is highlighted with a dark gray background. To the right of the options is a small icon of a list with three horizontal lines, and a downward-pointing triangle at the bottom.

Allow group to view the field in others' Contracts:

A dropdown menu with a light gray background and a dark gray header. The header contains the text 'admin' and a small upward-pointing triangle. Below the header, the menu is open, showing four options: 'adminimport', 'Anonymous', and 'Contract Creator'. The 'adminimport' option is highlighted with a dark gray background. To the right of the options is a small icon of a list with three horizontal lines, and a downward-pointing triangle at the bottom.

Allow group to change the field when editing Contracts created by others:

A dropdown menu with a light gray background and a dark gray header. The header contains the text 'admin' and a small upward-pointing triangle. Below the header, the menu is open, showing four options: 'adminimport', 'Anonymous', and 'Contract Creator'. The 'adminimport' option is highlighted with a dark gray background. To the right of the options is a small icon of a list with three horizontal lines, and a downward-pointing triangle at the bottom.

Options

To select the default value for your field choose an option in the first section.

Set default value for the imported fields based on:

The values from the record(s) in which the
Accounting Period field matches the value in the
Approval Notes field in the current Contract
Choose the record in which the value of the
---Select one---
field is less than all other matching records

run the Add asset ID to linked assets action when the Approval Notes field is first set or edited

The values for the user in the Choose one field of the user who changes the Choose one field

None

Selecting the first option imports the values from the record(s) in the source table in which a specific field matches a value in the current table. This is primarily useful as a default if the field being used for selection is automatically populated, generally from another set of imported fields. In this case the fields being populated by this rule must come after the other set of imported fields.

If there might be more than one record in the source table that matches the default criterion, you can choose the logic to be used for selection here.

The second to last option allows you to choose a default Company from which the imported fields will be autopopulated.

It is possible to have matching done against fields of different data types. For example, it is possible to configure a default value in table 1 based on matching a choice field in table 1 with a short text field in table 2.

Choosing "None" will leave the imported fields empty until a value is manually selected for one of the imported tables, at which time the others will be populated based on that record.

Next section:

Please select the desired behavior:

- Once the field has a value, never overwrite it
 Update the field whenever the underlying condition changes

Require the user to choose record(s) to be imported?

- Yes
 No
 Conditionally, when the

field has any of these values:

Require unique value in this table?

- Creator Teams
 Dates Created
 Dates for Reporting
 Deletables
 Demo Datas

Deletable

- Yes
 No

The user creating a new Contact may be forced to choose a Company (other than the default Company, if one is defined) to be imported into the current Contact by selecting Yes here. The user may use the input aid next to any imported field to make a selection.

If you select No and do not define a default value, the imported fields may be left empty.

Also it is possible to set 'Require unique value' attribute for individual fields from a single-value set of Linked Fields. This is useful if you want to be sure that a particular source record is only linked to one record in the current table, not more. It is clearly best used with fields that have a unique value in the source table, though it is not limited to such fields.

Next:

Restrict the Acct Periods from which the values may be imported?

Use all values

a saved search:

Make the visibility of this field conditional:

Only show this field if the field

contains one of these values:

You may want to restrict the records available for import to those that match specific criteria. It is often desirable to be able to match multiple fields in the current record against multiple fields in the source record. A special kind of saved search (called multi-table search) can have multiple rows that find specific values or that match a field in the current record with a field in the source record.

For instance, if you were creating an assignment field drawn from the Teams table, you might only want to include certain teams. You can use an already existing search as the filter, or you may create a new search.

Note that there is an option that allows you to further restrict the records from which the values may be imported.

This meets the need to provide a limited set of records to choose from based on some value in the current record, for instance the company of the user submitting the record.






A typical use of this might be when a user from a particular company submits a ticket, show him only the linked products or locations or assets for his own company by matching the company field in the ticket to the company field in the source table. Previously this had to be limited by group permissions and the filter couldn't be used for staff people who had full permissions.

If you want to use a matching field search criterion, select the field in the source table from the field drop-down, choose an operator, and choose variable rather than value from the next drop-down, then click on Formula help. Click on the Parent Field tab to find the field in the current table that you want to match against and to insert it into the search. It will insert the table name.fieldname into the search box. You can add any other criteria to be used from the source table and save the search.

Display

Use next tab to specify how your fields will be displayed.

Select the input aid for each field:

Creator Team	Hyperlinked Box with lookup	▼	
Date Created	Hyperlinked Box with lookup	▼	
Date for Reporting	Hyperlinked Box with lookup	▼	
Deletable	Hyperlinked Box with lookup	▼	
Demo Data	Hyperlinked Box with lookup	▼	

Each field imported from another table may be provided with a user input aid to the right of the field label. This allows users to select a valid record from the other table. Selecting a value from a drop-down list or a box with popup next to a field will populate the values for all imported fields from the other table.

For each imported field, select whether the input is a drop-down list of values (List of Values), a field input box with a pop-up search window (Box with popup), or just a normal input box (Box only).

Choosing option "Source field display type with lookup" causes the field to behave like a regular choice field in a non-strict link situation and allows choice selection, looking up and importing a record in a strict link situation.

If Box only is selected, the field can not be used to define a source record whose fields will be imported.

A drop-down list is usually best if there are fewer than 100 unique entries. If you select Auto, the system will automatically choose a drop-down list if there are fewer than 100 entries and will use a search box if there are more. As a sanity check to prevent the browser from crashing, a search popup is always used (even if drop-down list is specified) if there are more than 5000 entries. The input box width will be the same as the field width in the source table. The display characteristics for the imported fields can be changed individually by clicking on the Edit icon next to the field.

Note that a user will only see the drop-down list or pop-up icon if she has view permissions for the field in the Company table. Otherwise she will see a normal input box (box only) if the field can be manually edited, or no input box whatsoever.

Select a view to be displayed when a lookup search returns the search results. This view should help your technicians or users identify the correct record to be imported.

Select the view:

Display Acct Period using the user's default View for Acct Period

Display Acct Period using the

Default

Def
 Default
 Demo Data
 Mobile view
 Summary view

This field

All linked fields

All text fields

See Also:

[Linked Fields](#)

[Link to Single Field](#)

[Link to Selected Fields](#)

[Link to All Fields](#)

[Link to Single Field from Multiple Tables](#)

[Related Table](#)

[Linked Fields Display](#)

[Linked Fields Access Permissions](#)

[Linking to Other Databases](#)

[Copied Fields](#)

[Fields](#)

[Field Level Permissions](#)


[Flash Tutorials](#)



Linked Fields Display

Several methods are available for displaying Linked Fields. The display of a linked field depends on the type of the linked field and on whether it allows just one value or several values to be entered. In general, fields linked to the other table are provided with a user input aid to the right of the field box allowing the user to select a valid value from the source table's field.

Examples:

Box with lookup: Last Name: 

List of Values: Last Name: ▼

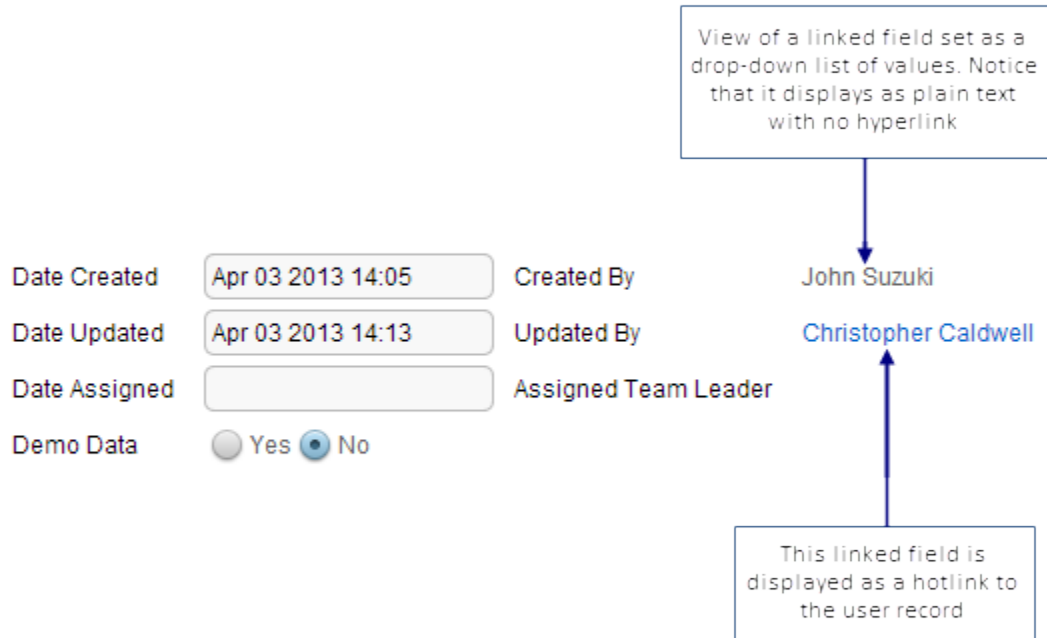
Box only: Last Name:

View Only: Last Name: Admin

One Value Fields

For fields that can only hold one value from the source table, you can choose between the display of a drop-down list of values (" Value List"), a field input box with a pop-up search window (" Box with pop-up"), or a regular input box (" Box only"). For an example of these input aids, see [Link to Single Field](#).

When a linked field has been filled with a value, it is displayed as a hotlink when viewing or editing the ticket and clicking on the hotlink opens up a view of the linked record.



There are two exceptions to this rule when editing a ticket:

1. If you chose to display the field as a drop-down list or
2. to display the field as a simple text box without any input aid,

in neither case is it displayed as a hotlink.

The reason for (1) is obvious-HTML drop down lists simply cannot act as hotlinks. The reason for (2) is that if the text were displayed as a hotlink without an input aid, there would be no way for the user to change it. Of course, in View mode, the linked fields are always displayed as hotlinks

Plain text View Only Fields

Additionally linked fields can display as plain text or a text box with lookup.

The plain text only option allows long text strings from linked fields to be displayed without the underlying link to the source record. This also allows linked fields containing URLs to redirect clicks to the URL and not to the linked record.

The plain text with lookup is a good option for fields where a default value is supplied but may need to be changed, as in the case where a staff user creates a record on behalf of an end user. This box allows you to begin typing over the current value to input a search, instead of clicking search and inputting search terms on a separate screen.

More than One Value Fields

If the linked field can hold more than one value from the source table, the field is displayed as a table within the current record.

In this case Time is a separate tab inside the table Support Case:

Support Case

Save Cancel Details Working Related Records Contract Time E

ID 426 Status Open
Assigned Team Support Team Assigned Person

All Time Entries

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

New Unlink Views

<input type="checkbox"/>	Edit	ID ↓	Done On	Done By	Work Description	Hours Spent
<input type="checkbox"/>		58	Apr 03 2013	Christopher Caldwell		1.50

Total Hours Spent 1.5 **Calculate**

In another example, you might have a table with all software programs owned by your company and import the names of all software you have loaded on a given computer from that table. This could be implemented using a series of checkboxes, or displaying the records that you imported from the source table. When displaying records, click Magnifying Glass to invoke the Search Popup, which allows you to add a link to the selected records, or click "Unlink" to remove the link to the selected records from the source table.

See Also:

[Linked Fields](#)

[Link to Single Field](#)

[Link to Selected Fields](#)

[Link to All Fields](#)

[Link to Single Field from Multiple Tables](#)

[Linking to Other Databases](#)

[Related Table](#)

[Copied Fields](#)

[Fields](#)

[Field Level Permissions](#)

Linked Fields Access Permissions

By default, linked or copied fields have the same group access permissions as the fields in the source table. You can view those permissions by clicking Setup/Tables/select Table to edit/Fields/select Linked data type to edit/Permissions and clicking View Permissions. You have the option to replace those permissions by copying them from a selected field in the current table.

Table	Mapping	Fields	Permissions	Options	Display
<input type="button" value="Back"/> <input type="button" value="Next"/> <input type="button" value="Cancel"/>					
<p>Note: This screen is view-only because you are editing a linked field set. Please use the Setup/Access/Groups wizard to edit the access permissions of the individual fields in the set.</p>					
<p>The selected groups will be able to see the field when viewing Email Clicks that they created or that were created on their behalf. The other groups will not see the field at all.</p>			<p>Allow group to view the field in their own Email Clicks:</p> <div style="border: 1px solid gray; padding: 2px;"> <input type="text"/> <ul style="list-style-type: none"> adminimport Anonymous Contract Creator </div>		
<p>The selected groups will be able to see and edit the field when submitting a new Email Click. The other groups will not see the field at all.</p>			<p>Allow group to edit the field when creating new Email Clicks:</p> <div style="border: 1px solid gray; padding: 2px;"> <input type="text"/> <ul style="list-style-type: none"> adminimport Anonymous Contract Creator </div>		
<p>The selected groups will be able to change the field when editing their own Email Clicks. The other groups will see the field during editing if they have the view permission but will not be able to edit it.</p>			<p>Allow group to change the field when editing their own Email Click's:</p> <div style="border: 1px solid gray; padding: 2px;"> <input type="text"/> <ul style="list-style-type: none"> adminimport Anonymous Contract Creator </div>		
<p>The selected groups will be able to see the field when viewing other people's Email Clicks. The other groups will not see the field at all.</p>			<p>Allow group to view the field in others' Email Clicks:</p> <div style="border: 1px solid gray; padding: 2px;"> <input type="text"/> <ul style="list-style-type: none"> adminimport Anonymous Contract Creator </div>		
<p>The selected groups will be able to change the field when editing Email Clicks created by others. The other groups will see the field during editing if they have the view permission but will not be able to edit it.</p>			<p>Allow group to change the field when editing Email Clicks created by others:</p> <div style="border: 1px solid gray; padding: 2px;"> <input type="text"/> <ul style="list-style-type: none"> adminimport Anonymous Contract Creator </div>		
<input type="button" value="Back"/> <input type="button" value="Next"/> <input type="button" value="Cancel"/>					

When fields are copied, their permissions are copied from the source table by default. To view these permissions, click View Permissions. You have the option to replace these permissions with those copied from a field in the current table.

To copy the permissions from a field in the current table, select it and click Copy Permissions.

The Copy Permissions operation sets the permissions for all of the selected fields. If you want to assign different permissions to some of the fields, you will need to go to Setup/Access/Work with Groups and edit the group permissions directly for the affected groups.

The selected groups will be able to see the field when viewing a table that they created or that was created on their behalf. The other groups will not see the field at all.

When a field is linked to a field in another table, the users must be given View Others' permissions to the field in the source table, in order to be able to use it (for instance, in order to view the currently selected value or select a new value for the linked field).

See Also:

[Linked Fields](#)

[Link to Single Field](#)

[Link to Selected Fields](#)

[Link to All Fields](#)

[Link to Single Field from Multiple Tables](#)

[Related Table](#)

[Linked Fields Display](#)

[Linking to Other Databases](#)

[Copied Fields](#)

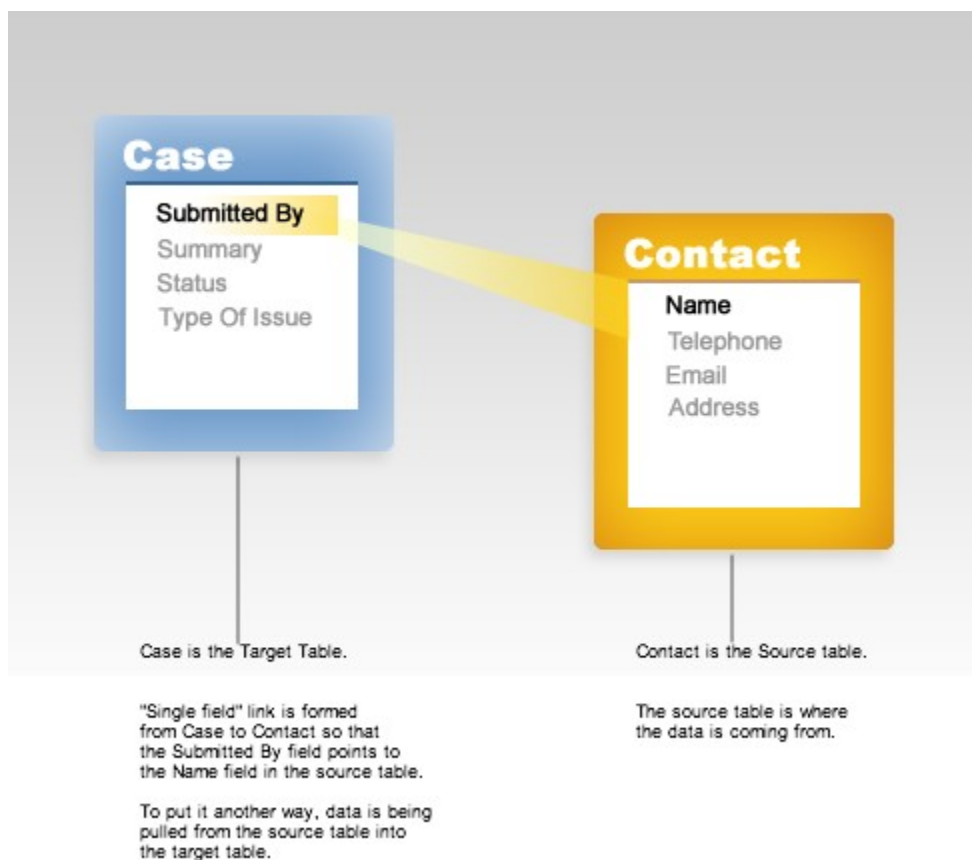
[Fields](#)

[Field Level Permissions](#)

Link to Single Field from Another Table

This data type is used to add a single field to a table that pulls its values from a single field in another table.

For example, you might add a multi-choice field called *Submitted By* into the Case table and link it to the *Name* field in the Contact table. The *Submitted By* field in the Case table will take values from the *Name* field in the Contact table. The relationship between the two tables is shown in Fig. 1 below.



The field type "Link to Selected Fields from Another Table" is usually preferable to "Link to Single Field from Another Table" because it allows greater flexibility when editing later. Unlike Link to a Single Field, Link to Selected Fields can be expanded to multiple fields as needed. Use Link to a Single Field only when you are sure that you will not need other fields later.

The input box for the linked field is determined by the display option you select when setting up the field, accessed by clicking Setup/Tables/, selecting the Table to edit, clicking /Fields/New/Link to Single Field from Another Table/Display. The input box is also determined by the data type of the field in the source table. For example, a linked field of type "Date/Time" displayed in a Box with Popup will have

Date input, Time input and a Magnifying glass. Click the magnifying glass icon to invoke the Search Popup.

The screenshot shows a form with the following fields and values:

- Status: Planned
- *Title: (empty)
- Submit for Review: (button)
- Publication Date: Oct 20 2012 10:35
- ID: 12
- Rush: No
- Document Type: Choose one
- *Requires Approval: Yes
- Priority: Choose one

The same linked field displayed as a List of Values will have a drop-down list on the Record form.

The screenshot shows a form with the following fields and values:

- Status: Cancelled
- *Title: Corporate dress code
- Submit for Review: (button)
- Assigned Person: Paula Ling
- Next Milestone Date: (calendar icon)
- Approval Team: Document Approval
- Approval Status: Approved (radio), Rejected (radio)
- Approved/Rejected By: System
- Date Approved/Rejected: (empty)

The 'Publication Date' field has a search popup menu open, displaying a list of dates and times:

- Apr 10 2013 11:39
- Feb 14 2012 15:40
- Jan 09 2012 13:54 (highlighted)
- Jul 04 2012 11:50
- Jul 23 2012 11:53
- Jul 28 2012 13:58
- Oct 17 2011 14:40
- Oct 20 2012 10:35
- Sep 27 2011 10:48

The Search Popup allows you to quickly select one or several values from the source field and populate them into the linked field input.

Depending on the configuration of a linked field, the user may be able to manually enter the value into the linked field input, without using the Search Popup.

See Also:

[Linked Fields](#)

[Link to Selected Fields](#)

[Link to All Fields](#)

[Link to Single Field from Multiple Tables](#)

[Related Table](#)

[Linked Fields Display](#)

[Linked Fields Access Permissions](#)

[Linking to Other Databases](#)

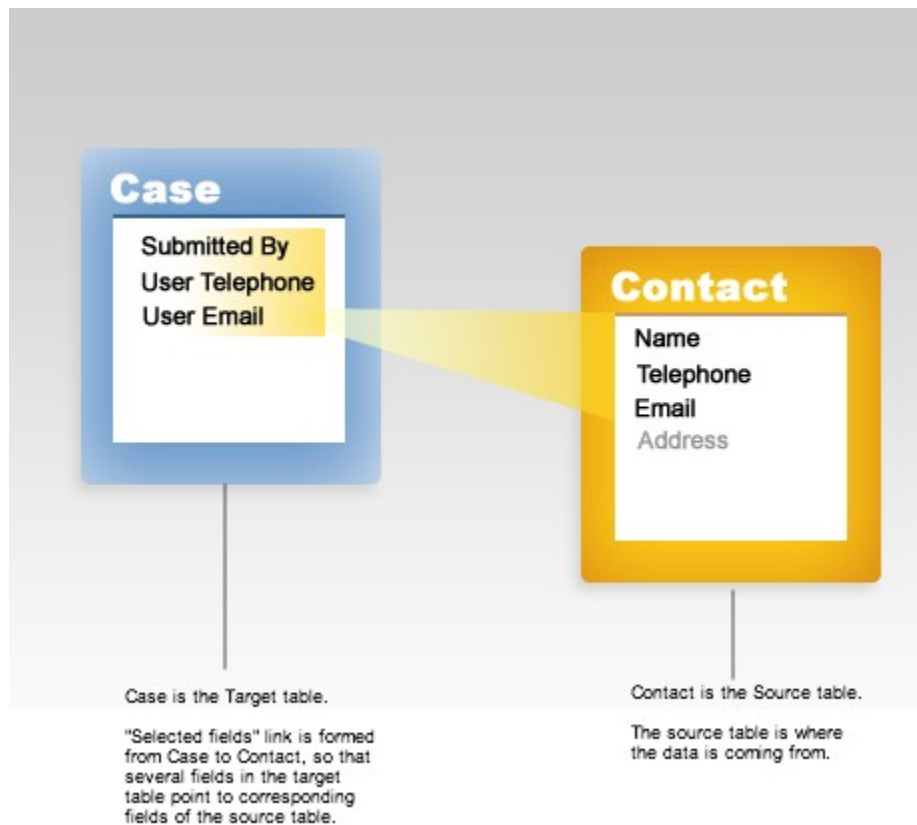
[Copied Fields](#)

[Fields](#)

[Field Level Permissions](#)

Link to Selected Fields from Another Table

Multiple fields from a source table may be linked to the current table in one operation. For example when a user submits a Case, you may want to automatically include his or her name, telephone number and email address. The data would be pulled automatically based on the user's login and could also be manually imported by selecting a different user in the Name field.



Let's assume that a Company table needs contracts information. It can be pulled from Contract table. This is how it looks when you setup fields for Company table (Setup/Tables/Edit/Fields tab):

Contract	contract_description	contract_description	Link to: selected fields from other table
Description	latest_contract_end_date	latest_contract_end_date	Contract:contract_description/Contract
Latest Contract	contract_is_for	contract_is_for	Description
End Date	latest_contract	latest_contract	contract_end_date/Contract End Date
Contract is For	latest_contract_id	latest_contract_id	contract_is_for/Contract is For
Latest Contract			contract_title/Contract Title
Latest Contract			id/ID
ID			

When viewed in Staff interface, the pulled fields may be displayed like this (subject to Layout configuration):

 **Contract**

ID	28	Our Company is the	Client
Internal Contract Owner	EW Admin	Status	Pending Approval
Company Name	AT&T	Company Main Contact	
Street Address 1	32 Avenue of the Americas	City	New York
US State	NY	State or Province	
Country	USA	Zip or Postal Code	10013-2412

See Also:

[Linked Fields](#)

[Link to Single Field](#)

[Link to All Fields](#)

[Link to Single Field from Multiple Tables](#)

[Related Table](#)

[Linked Fields Display](#)

[Linked Fields Access Permissions](#)

[Linking to Other Databases](#)

[Copied Fields](#)

[Fields](#)

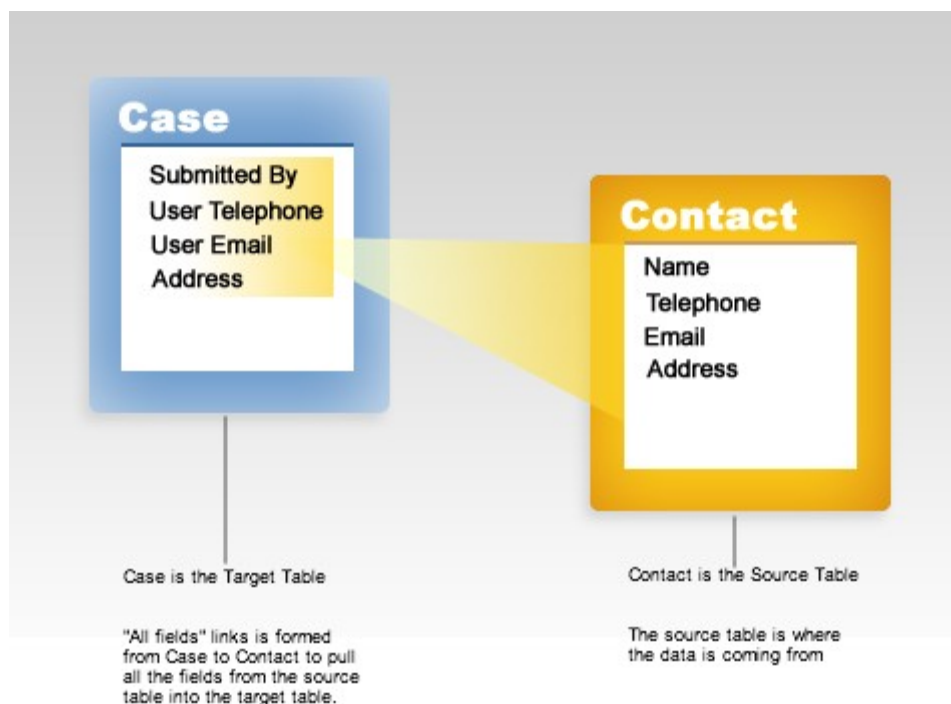
[Field Level Permissions](#)

Link to All Fields from Another Table

This data type allows you to link all fields from a source table into the selected table. For example, you might place all of the Account table fields in the Customer table.

This data type keeps track of any fields that might be subsequently added to the source table. The new fields will automatically be added to the group of linked fields in the target table.

History fields, Email History fields, and other Linked data fields are not included in the "All fields From another table" operation.



See Also:

[Linked Fields](#)

[Link to Single Field](#)

[Link to Selected Fields](#)

[Link to Single Field from Multiple Tables](#)

[Related Table](#)

[Linked Fields Display](#)

[Linked Fields Access Permissions](#)

[Linking to Other Databases](#)

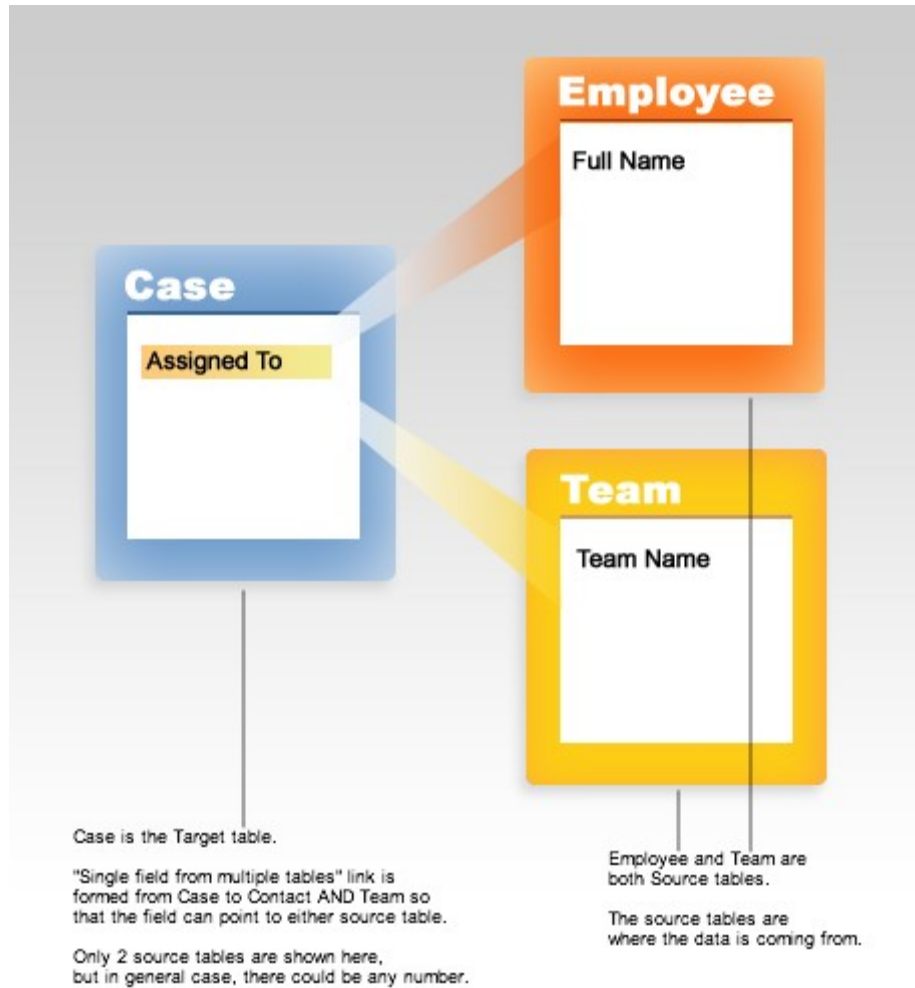
[Copied Fields](#)

[Fields](#)

[Field Level Permissions](#)

Link to Single Field from Multiple Tables

This data type is used to create a field that pulls its possible values from more than one source table. For example, you might have the *Assigned To* field in the Case table pull its values from the Contact table as well as from the Team table so you can assign not only to individuals but also to whole teams.



When you setup this field for Case table (Setup/Tables/Edit/Fields tab) add Source tables one by one in the first tab of the Field Wizard:

Table Mapping Field Permissions Options Display

Next Cancel

This dialog helps you select source tables for the [linked field](#).

Select a table and click **Add Source Table** to add it to the list of source tables. You can add as many source tables as needed. [...More Help](#)

You can also import values for a field in another Agiloft KnowledgeBase table or from an external database table.

For another Agiloft KnowledgeBase, you must provide a login and password for a member of the admin group. For an external database table, you must provide a valid login and password. [...More Help](#)

Select table:

Agiloft table in the current KnowledgeBase:

Agiloft Request

Table from another database:

Database Name:

Login:

Password:

Get Tables

Add Source Table **Remove Source Table**

---Selected tables---

Acct Period
Company
Contract

Use the Admin Notes field to put your comments about the purpose and details of the field's configuration and usage.

These notes can then be included in the automated Field Documentation printout.

Admin notes

If this checkbox is selected the Admin Notes shown above will be included in the automated Field Documentation printout.

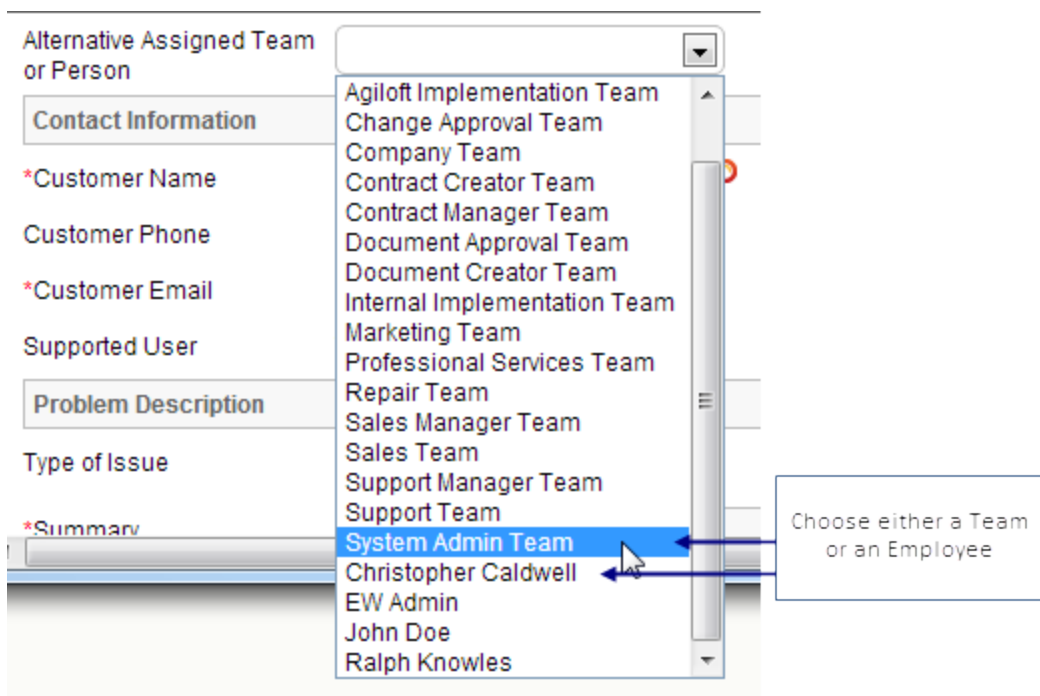
Include in Field Documentation

Next Cancel

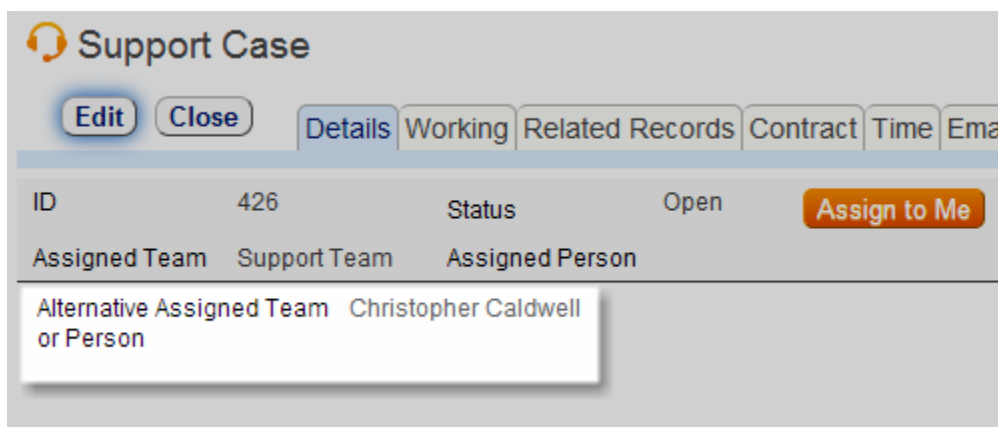
Once fields is created, the fact that information has to be pulled from multiple tables, will be displayed like this:

Alternative Assigned Team or Person	_1905_full_name	_1905_full_name	Link to: single field from multiple tables Employee: full_name/Full Name Team: _name/Team Name
Assign to Me	assign_to_me	assign_to_me	Action button
Assigned Person	assigned_person	assigned_person	Link to: selected fields from other table Employee: full_name/Full Name
Assigned Team Leader Assigned Team	assigned_team_leader assigned_team	assigned_team_leader assigned_team	Link to: selected fields from other table Team: team_leader/Team Leader

The drop-down field is often the best way to present such relationship when editing/creating the Case. Note that Assigned To field accepts both Team or Full Name value.



This is how it looks in Staff interface when you view Case record.



See Also:

[Linked Fields](#)

[Link to Single Field](#)

[Link to Selected Fields](#)

[Link to All Fields](#)

[Related Table](#)

[Linked Fields Display](#)

[Linked Fields Access Permissions](#)

[Linking to Other Databases](#)

[Copied Fields](#)

[Fields](#)

[Field Level Permissions](#)

Linking to Other Databases

In order to link to tables in another database, you must first tell the application server about that database by adding a datasource descriptor file.

Agiloft only supports external tables with single primary key field which must be of exact numeric type (eg bigint/integ)

See examples of datasource configurations in *EW_HOME/jboss/server/sw/deploy/ew-ds.xml* as well as [JBoss AS Config DataSources](#). More detailed information is available in [Configuring JDBC DataSources](#) document from JBoss AS documentation.

Example for MSSQL:

```
<no-tx-datasource>
<jndi-name>DATABASE_NAME</jndi-name>
<min-pool-size>10</min-pool-size>
<max-pool-size>20</max-pool-size>
<blocking-timeout-millis>7500</blocking-timeout-millis>
<idle-timeout-minutes>5</idle-timeout-minutes>
<connection-url>jdbc:sqlserver://
SQL_SERVER_HOSTNAME:1433;database=DATABASE_NAME;
SelectMethod=direct;lastupdatecount=true;responseBuffering=adaptive;
lockTimeout=900000;packetSize=512;applicationName=EW</connection-url>
<driver-class>com.microsoft.sqlserver.jdbc.SQLServerDriver</driver-class>
<user-name>USER_NAME</user-name>
<password>PASSWORD</password>
<type-mapping>MS SQLSERVER2000</type-mapping>
</no-tx-datasource>
```

To prevent the datasource file from being overwritten on update give it a name matching one of the following patterns:

- **ewcustom*-ds.xml*
- *ew_datasource_*-ds.xml*

Example: *ew_datasource_my_external_db_ds.xml* or *my_db_ewcustom-ds.xml*

Jboss AS picks up the changes to this file immediately, so it is advisable to stop the server process before making changes to this file and start again only after changes are complete.

See Also:

[Linked Fields](#)

[Link to Single Field](#)

[Link to Selected Fields](#)

[Link to All Fields](#)

[Link to Single Field from Multiple Tables](#)

[Related Table](#)

[Linked Fields Display](#)

[Linked Fields Access Permissions](#)

[Copied Fields](#)

[Fields](#)

[Field Level Permissions](#)

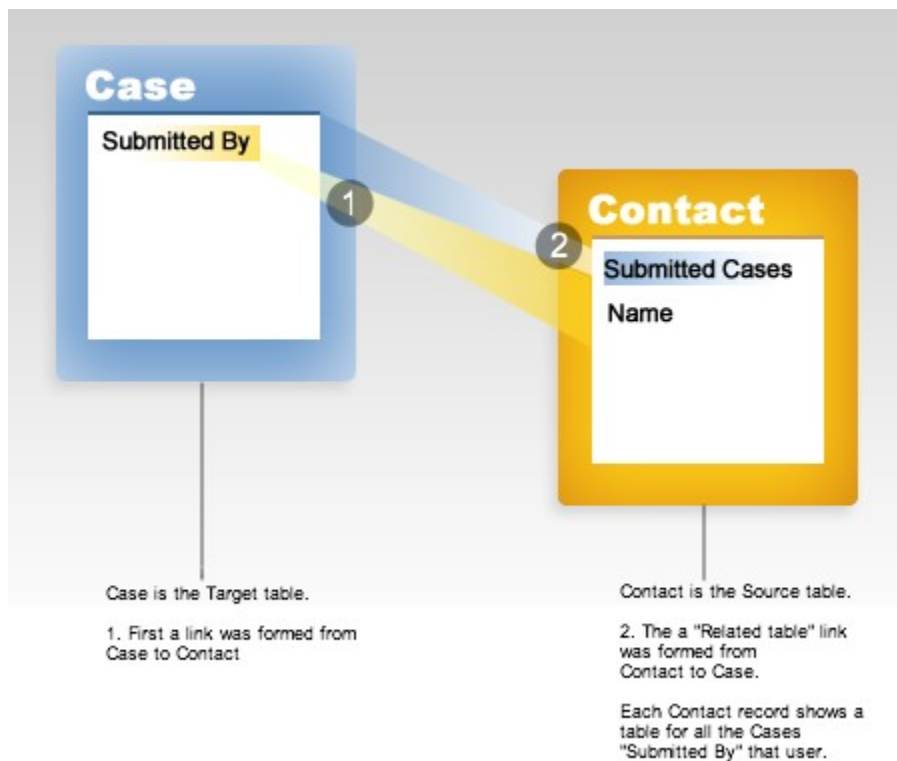
Related Table

Related Tables are used to show records outside the current table that reference or are dependent on a particular source record. You could consider this the inverse of linked fields: putting a Related Table in a source record shows all external records in the selected outside table that reference your source record. For instance, you may wish to view all of a particular Contact's help Case records, so in that Contact record you would put a Related Table linking to Cases.

To create a Related Table, there must first be other tables referencing your source table data; at least one field of your source table must be defined in another table as a Linked Field.

For example, to create a Related Table in the Contact table that will pull in all of a customer's cases, there must be a "Link to a Field from Another Table" field in the Case table that references the Contact table. In this example, the *Submitted By* field in the Case table might be linked to the *Name* field in the Contact table.

To create a Related table, go to Setup/Tables/, select a Table to edit, go to /Fields/New and choose 'Related Table". The wizard will ask you to specify which relationship you want to use for your Related Table. Using the scenario above, you would need to select "Case: Submitted By-> Name" on the General tab of the Related Table Wizard.



Here is an example of a Contact/End User record that shows multiple tables related to the End User table. In other words, these Related tables show all the places where this End User record is being referenced:

 End User

Save Cancel Contact Information Account Info **Activities** Support Items H

ID 250 **Export to Outlook**

Open Activities

Open Tasks/Activities

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

New Mass Edit Delete Unlink Actions Views Search

<input type="checkbox"/>	Edit	ID ↓	Title	Status	Assigned Person	Date Due
<input type="checkbox"/>		7	Draft proposal	Incomplete	EW Admin	Aug 12 2012 17:00
<input type="checkbox"/>		2	Set up time for onsite visit	Incomplete	EW Admin	Apr 16 2012 09:00

Past Activities

Completed Tasks/Activities

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

New Unlink Views

<input type="checkbox"/>	Edit	ID ↓	Title	Status	Assigned Person	Date Due
<input type="checkbox"/>		9	Writeup	Done	Christopher Caldwell	Jul 02 2012 12:30

Opportunities

Opportunities

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

New Mass Edit Delete Unlink Actions Search

<input type="checkbox"/>	Edit	Opportunity Name	Status	Sales Rep	Estimated Close Date
<input type="checkbox"/>		Additional Licenses for IBM	In Evaluation	Jack Richards	Jun 28 2012
<input type="checkbox"/>		New Computer parts for IBM	Closed Won	Jack Richards	Jul 01 2012

Quotations

Quotes

See Also:

[Linked Fields](#)

[Link to Single Field](#)

[Link to Selected Fields](#)

[Link to All Fields](#)

[Link to Single Field from Multiple Tables](#)

[Linked Fields Display](#)

[Linked Fields Access Permissions](#)

[Linking to Other Databases](#)

[Copied Fields](#)

[Fields](#)

[Field Level Permissions](#)

Copied Fields (Attributes) from Another Table

Another special data type that can be selected when creating fields is the "Copy fields (attributes only) from other table" This option enables you to easily add predefined groups of fields to other tables in a single operation, without having to redefine the fields each time. For example, you could use the Credit Card or Mailing Address fields in several tables within the current KnowledgeBase without having to redefine them for each table.

See Also:

[Linked Fields](#)

[Link to Single Field](#)

[Link to Selected Fields](#)

[Link to All Fields](#)

[Link to Single Field from Multiple Tables](#)

[Related Table](#)

[Linked Fields Display](#)

[Linked Fields Access Permissions](#)

[Fields](#)

[Field Level Permissions](#)

TEAMS

Whereas Groups control the access rights or permissions the user has within the system, Teams help to encapsulate business processes, and because teams can have sub-teams, they can elegantly model the company's entire hierarchy and supplier/customer relationships. For example, Joss might be a member of the West Coast Sales team, which is in turn a member of the US Sales Team.

Team membership enables assignment of working hours, email distribution, access to team-specific views, Look-and-Feel Schemes, searches, rules, etc.

OVERVIEW

- Teams serve different functions for staff and customer groups
 - Staff teams are used primarily to identify functional units to whom records and chat requests might be assigned, emailed, and so on.
 - Customer 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.
- Users have one primary team and can belong to any number of additional teams. A user's primary team determines his default look and feel schemes and default view of each table
- Teams can be hierarchical and email notifications sent to a parent team will also go out to all sub-team members
- Teams can be assigned working hours and calendar events

The system comes pre-configured with several teams, and you can create any number of custom teams in your KnowledgeBase. To add a user to a team, open the User form for editing (Contacts/{select Customer or Employee to edit}) and check the appropriate teams, also making the Primary team selection.

STARTING FROM DEFAULT TEAMS

- These are often adequate for a new implementation and can be copied to create additional groups where more refined distinctions in permissions are necessary.

WHAT NEW TEAMS ARE NEEDED?

- There are several default teams that match the default groups (Support Team, Sales Team, Customer Team, Internal Customer Team, etc).
- You will need new staff teams for each set of users to whom records may be assigned or emailed as a group.

- You will need new customer teams if you are going to support multiple brandings (ie different look and feels) or different default views

EDITING A TEAM LATER

The process for editing is same as for creating it: click the Edit icon next to the team. Then change the name or description, add calendar items, or change working hours and save your changes.

Team membership is independent of group membership, so that 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 their Primary team to be set before they can login. Team membership also controls how business 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.

Using teams, you can define working hours for your users during weekdays and holidays. Different teams can be assigned custom working hours to accommodate customers from different time zones. By default, the working hours for the team are the same as the working hours for the built-in Company team.

The Team Wizard (Access/Manage Teams/Select Team to edit/Events) lets you optionally define the team Calendar where you can schedule events, such as meetings, vacations or announcements. For more information on events, see [Calendar](#).

Teams table is also accessible from the main toolbar. This makes it possible to mass edit teams, delete multiple teams at once, import and export teams, and perform some other general table functions.

However, there are still limitations on the teams table due to its special hard coded fields for language, working hours, and so on. At the moment, only custom fields added to the table may be edited and created through the top level tab.

On the other hand, mass edit, import and export only allow access to the basic standard team fields: Team Name, Description, Parent Team, and Team Leader.

The Teams table is accessible from the Main Toolbar on the left

Teams
Status: 20 record(s) found, 1 pages. [Click here](#) to re-count records

New Delete Actions

<input type="checkbox"/>	Edit	ID ↓	Team Name	Description
<input type="checkbox"/>		407	Repair Team	Default team assigned
<input type="checkbox"/>		406	Internal Implementation Team	For Admins of EW resp it
<input type="checkbox"/>		405	Agiloft Implementation Team	Team responsible for it
<input type="checkbox"/>		404	Document Creator Team	Document Creator Tea external users, then ct
<input type="checkbox"/>		403	Contract Creator Team	Contract end user who own contracts. Can be
<input type="checkbox"/>		402	Change Approval Team	Default team to approv
<input type="checkbox"/>		401	Customer Manager Team	Customer Manager Tea
<input type="checkbox"/>		400	Contract Manager Team	Contract Manager Tea
<input type="checkbox"/>		399	Support Manager Team	Support Manager Tea

Teams may be deleted, but users need to be reassigned to another team if it is their Primary team.

See Also:

[Groups](#)

[Look and Feel](#)

[Rule](#)

[Workflow](#)

[Calendar](#)

[Saved Search](#)

[View](#)

[Flash Tutorials](#)



TEAMS WIZARD

Click Setup/Access/Manage Teams to create new teams, delete teams or edit the attributes of an existing team.

General Tab

The Parent team defines hierarchy, if any. Refresh rates for chat messages are defined on team basis.

The Team leader field in the Team table can be used as a normal user linked field. It can be populated as a linked field along with the Team Name into another table, and then the team leader for a team can be emailed using rules.

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
408

Enter the name of the team. This name will identify the team in the current KnowledgeBase.

***Team Name**

Select a parent team for your team. 

Parent Team

None

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](#)

Select a team leader
The team leader does not need to be a member of the team.

Team Leader

None

Enter a brief description of the team to help you remember its purpose.

***Team Description**

This option specifies how often the chat messages are refreshed for users for whom this is their Primary team

Message Refresh Rate

Every 5 Minutes

Every Seconds

This option specifies how often the table view is refreshed for users for whom this is their Primary team

Table View Refresh Rate

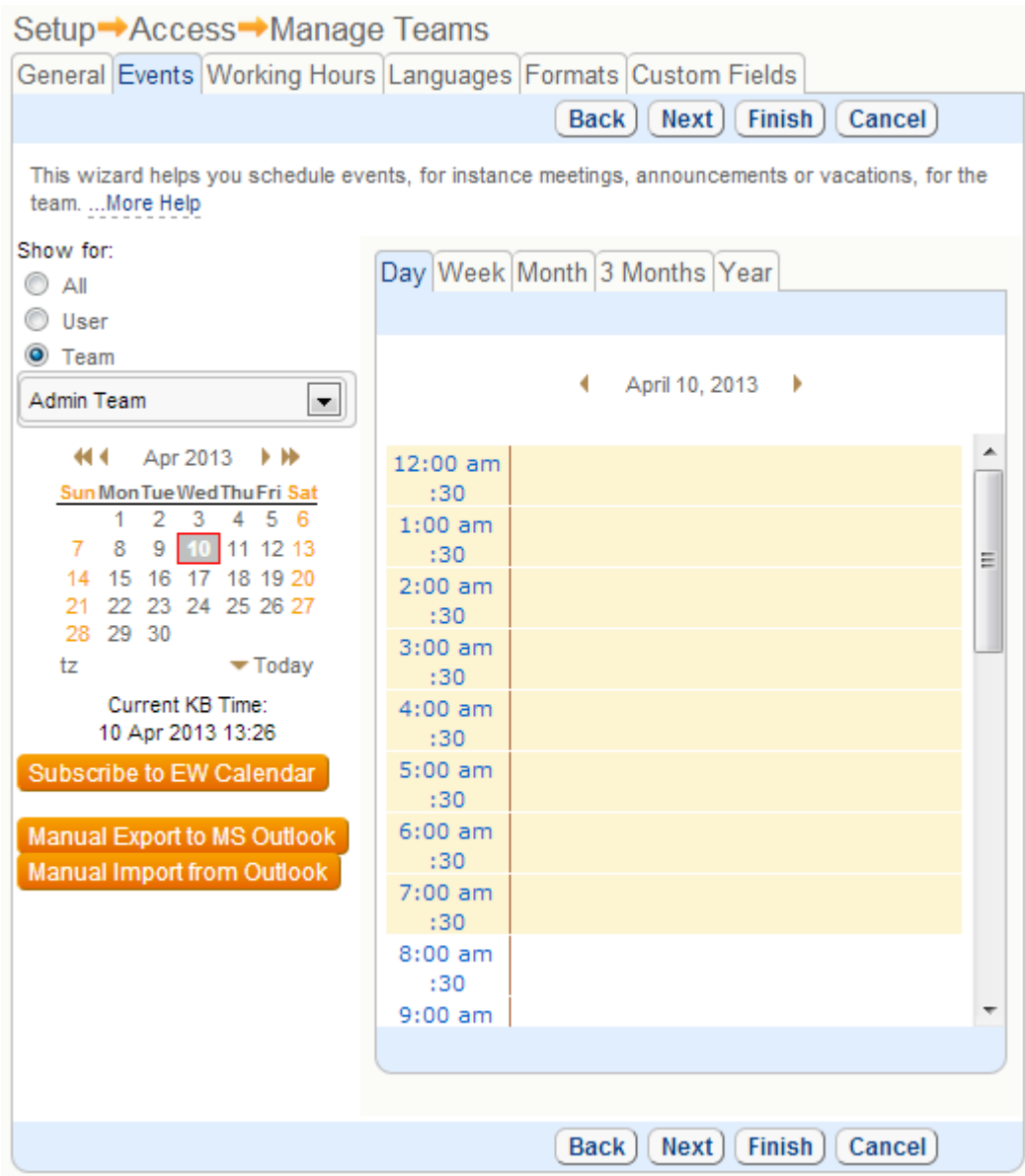
Never

Every Minutes

Next Finish Cancel

Events

Events can be scheduled here for whole teams as well as through regular calendar entry.



Working hours

Working hours have some special uses - you can define rules to run only during a team's working hours and run searches for elapsed time that exclude the nonworking hours of a specific team. You may wish to create dummy teams to define different sets of working hours.

Setup → Access → Manage Teams

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

Back Next Finish Cancel

Select the working hours for the team during the week days. If you accept the default setting, the working hours for the team will be the same as the working hours for the Company Team.

Working Hours

Same hours as
Admin Team

As defined below

Sun	08:00	-	17:00
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	08:00	-	17:00

Back Next Finish Cancel

Working hours for teams that work into the next day may be set without difficulty. This is useful for companies with teams outside their home time zone, or 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. For example:

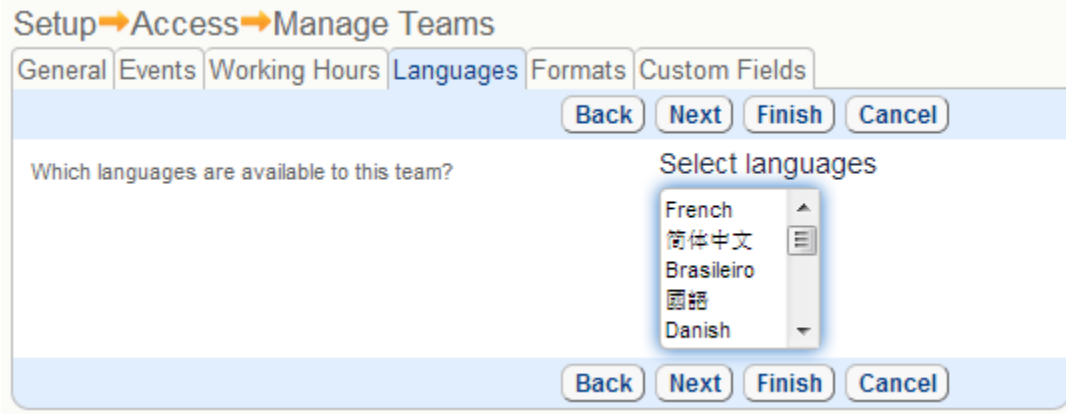
Sunday: 22:00 to 11:00 will mean 22:00 Sunday to 11:00 AM Monday

Monday: 22:00 to 11:00 will mean 22:00 Monday to 11:00 AM Tuesday

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

Languages

Choose language for the team:



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.

Users can belong to many groups and enjoy the combined permissions of all the groups that they belong to.

Every KnowledgeBase is provided with several pre-defined groups. You can modify or delete them as needed and create an unlimited number of new groups.

A group can be either a Customer group or Staff group. Customer groups cannot access the Staff Interface or edit records created by other users. By placing users who only need to submit records and use the FAQ in a Customer group, you limit their access and reduce licensing costs. Staff users use an individual named license and can access either interface and can perform all functions permitted to them. The type of group you put a user in determines his license usage.

STARTING FROM DEFAULT GROUPS

- There are several default groups pre-configured for CRM applications. These are often adequate for a new implementation and can be copied to create additional groups where more refined distinctions in permissions are necessary.
- Never rename the admin group or it will not function as it should.

Default Customer/End User Groups

These groups use the unlimited end-user license:

- Customer – external customer – members of this group only see their own records.
- Customer Manager – external – members of this group see all the records submitted by users in their company and may be given other privileges.
- Internal Customer – used for employees who are end users (for Helpdesk applications).
- Guest – this group is used primarily in hyperlinks to enable self-registration and creation of records such as leads and support tickets, without giving the user access to the rest of the interface. A hyperlink logs them in to the new record screen, then logs them out when they hit save or cancel.
- Self-registered user – a group that can be used for users who create their own account. This group can be given restricted access privileges until the user is validated and moved into the customer group.

Default Staff Groups

These groups use individual licenses:

- Admin – this is for administrators who are customizing and maintaining the system.
- Support Staff – for technical staff accessing external and internal support cases.
- Sales – accesses leads, opportunities, and so on
- Marketing – same as Sales group

GROUP PERMISSIONS

Group permissions control access to saved searches, the left hand menu bar, Views and Reports. For example, you can allow a group to create, modify, delete, execute or assign searches in the selected table. This function is accessed via Setup/Access/Manage Groups/, selecting the Group to edit, /Edit/Table/, selecting the Table to edit, /Searches tab.

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 team(s).

See Also:

[Record Level Permissions](#)

[Field Level Permissions](#)

[Teams](#)

RECORD LEVEL PERMISSIONS

Record level access permissions define what types of record (i.e. what tables) group members can view, edit or delete.

Record level permissions can be based on saved searches, giving them greater flexibility and utility. For example, you might create a saved search "Member of Same Company" that finds all records created by users with a Company field that matches yours. (i.e. company=\$my_company). You can then specify that group members can View or Edit records that satisfy this saved search.

There are two ways to set record level access permissions.

1) Edit the group and set what records that group can view, edit and delete – this is useful when setting the permissions that a particular group has for multiple tables. Select: Setup/Access/Manage Groups/, select a group to edit, /Table/, select a table to edit, /Permissions.

Note that saved-search based access permissions can only be defined by editing the group.

2) Edit the table and set what groups can view, edit and delete records – this is useful when setting the permissions that multiple groups have for a particular table. Select: Setup/Tables/, select table to edit, /Edit/Permissions.

See Also:

[Groups](#)

[Field Level Permissions](#)

[Mass Edit Permissions](#)

[Reports and Permissions](#)

FIELD LEVEL PERMISSIONS

Field level access permissions are only relevant for tables that group members are allowed to view or edit. They control what fields within the record can be viewed or edited in records that the user has created and in records that other people have created.

There are two ways to define field level access permissions.

1) Edit the group and set what fields that group can view/edit – this is useful when setting the permissions that a particular group has for multiple fields. Select: Setup/Access/Manage Groups/select group to edit/Table/select table to edit/Field Permissions.

2) Edit the field and set what groups can view/edit that field – this is useful when setting the permissions that multiple groups have for a particular field. Select: Setup/Tables/select table to edit/Fields/select field to edit/Permissions.

See Also:

[Groups](#)

[Record Level Permissions](#)

[Mass Edit Permissions](#)

[Reports and Permissions](#)

CREATING NEW GROUPS

Creating new groups can be done at the beginning or late in the implementation. To create a group go to: Setup/Access/Manage Groups. Here is the overview of all tabs in Group Wizard:

Setup → Access → Manage Groups

Overview | General | Table | Menu-Specific | Permissions | Field Permissions | History

Next Finish Cancel

This is the ID number of the group in the current KnowledgeBase. **Group Identifier**
318

Choose a name for the group. This name will uniquely identify the group in the current KnowledgeBase. ***Group Name**
Internal Support

Provide a description to help you remember the purpose of this group. ***Description**
The internal support group is responsible for all tickets related to inter-company incidents.

Select the general role of this group in the system. **?** Staff Customer

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

Customers can only log on to the End-User Interface and they use an End User License (which is generally an unlimited user license). They can be given permissions to submit, edit and view records, but cannot edit records they do not own. [...More Help](#)

Next Finish Cancel

Overview

The Overview tab specifies the name/description of the group and whether it is a staff or customer group.

General Tab

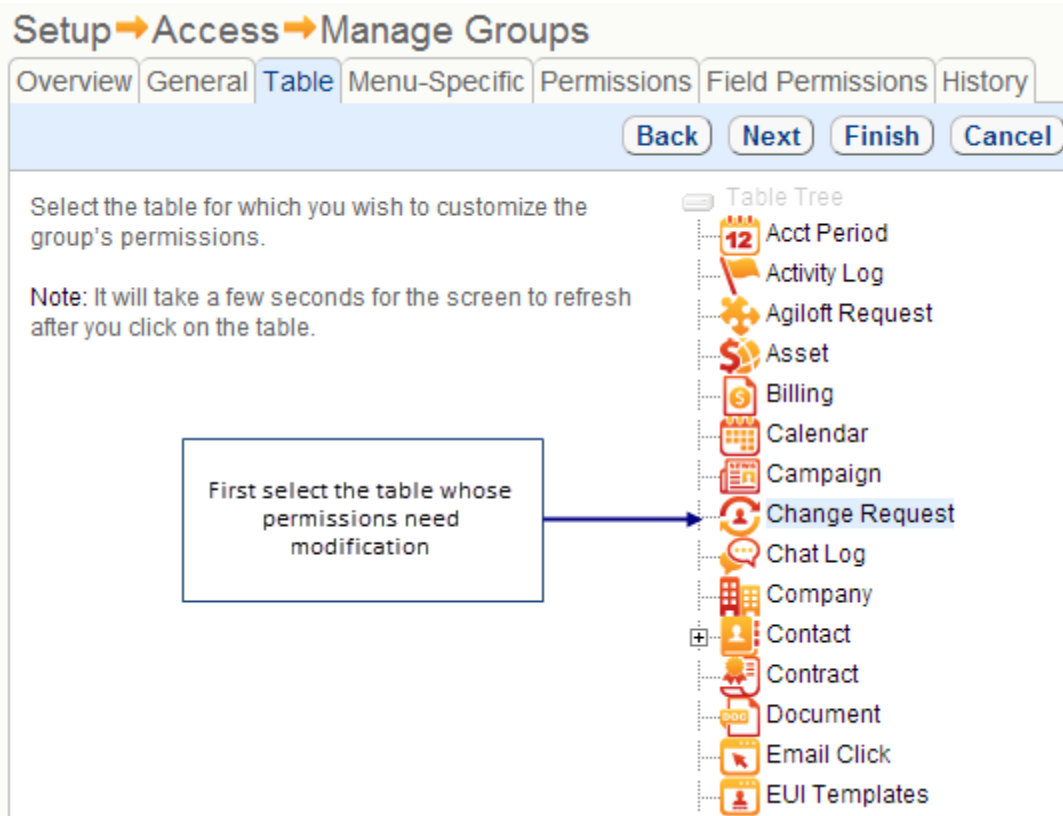
The General tab specifies high level security, such as whether members of the group can access the Setup wizards.

Groups that can access the Setup wizards can modify group permissions so this access is generally only given to the admin group.

The General tab also specifies whether group members can be sent a new password via email if they forget their password, whether they can modify the left-hand pane, export/import the entire knowledgebase and whether they can send Broadcast messages.

Table Tab

The remaining tabs will affect the permissions for the table that you select here.



Select the table whose permissions you wish to modify. All tabs following this selection (either by clicking "Next" or clicking a tab to the right of "Tables") will apply to this selected table.

<p>Select whether any permission changes you make should propagate to the sub-tables of the selected table. ...More Help</p>	<p>Apply to Sub-tables</p> <p><input checked="" type="radio"/> Do not apply changes</p> <p><input type="radio"/> Apply changes only</p> <p><input type="radio"/> Apply all permissions</p>
<p>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.</p>	<p>Allow access to the table?</p> <p><input type="radio"/> Yes</p> <p><input checked="" type="radio"/> No</p>
<p>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.</p>	<p>Show table on the Toolbar?</p> <p><input type="radio"/> Yes</p> <p><input checked="" type="radio"/> No</p>

Apply to Subtables

This option controls whether changes that you make to the selected table apply only to itself, or to itself and any subtables it contains. The third option will copy *all* permissions, not just those that have just been edited, overwriting any custom or differing permissions granted to the subtable.

Allow Table Access

This option controls whether members of the group can access the table at all. If you select "No", and then click later settings tabs for this table without selecting a new one, you will notice that all of the settings are grayed out.

Show Table in Toolbar

This option controls whether the table's tab is shown in the toolbar. This is not the same as denying a group access to a table.

Group Access and Toolbar Tab permissions are not the same. Creating a record via a [Custom Hotlink](#), viewing a table's Reports, or selecting information from a Linked Field require access to a table. However, if you do not want to give users permissions outside of these specific scenarios, you can effectively limit their

access by removing the table's tab from their toolbar by selecting "No" from the "Show" options.

Menu-Specific Tab

The Menu-Specific tab controls whether users can create, edit, or delete Searches, Views, all Templates, or Reports. A few examples are provided below.

Search permissions are controlled as below.

Setup → Access → Manage Groups

Overview General Table **Menu-Specific** Permissions Field Permissions History

Back Next Finish Cancel

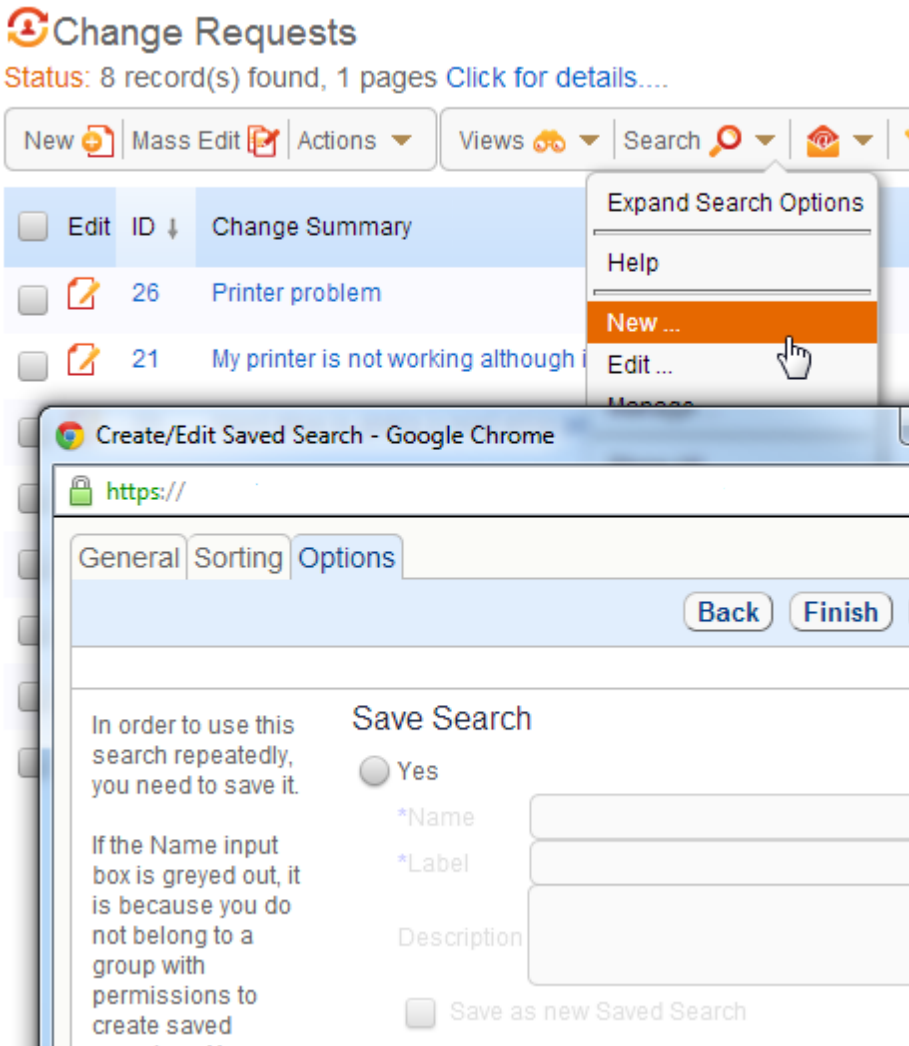
The permissions you set using this dialog will apply to the table: Change Request

Select whether this group can create, modify and delete saved searches. [...More Help](#)

Allow
creating/modifying/deleting
Saved Searches?

- Don't allow
- Allow for their own saved searches
- Allow for all saved searches

If permission to create/modify/delete Saved Searches is removed, users may still search but the ability to save a search is disabled.



The Views subsection is similar to the Searches subsection and controls whether users can save Views. It also controls whether they can make those views accessible to other users.

Select whether this group can create, edit and delete views. [...More Help](#)

Allow creating/editing/deleting Views?

- Don't allow
- Allow for their own views
- Allow for all views

Select whether members of this group can make views visible to other users.

Allow publishing Views

The Reports subsection is also similar to the Searches subsection, with the additional option to control access to reports by whether the "published" checkbox

has been selected. For more information on publishing reports for general viewing, see the section on the Reports Wizard.

Select whether this group can create, modify and delete Reports based on the Change Request table.

Allow creating/modifying/deleting Reports?

- Don't allow
- Allow group to modify/delete saved reports that they created
- Allow group to modify/delete all saved reports

Select whether this group can access none of reports, all reports or just **published** reports.

Allow using Reports?

- Don't allow
- Allow using **published** Reports only
- Allow using all Reports

Sometimes, the admin wants to allow a group to create reports but not be able to publish them.

- Allow group to publish saved reports

Other menu-specific permissions controlled in this tab involve E-mail templates and Print templates. Those subsections are analogous to the controls for the Searches subsection.

Permissions

The Permissions Tab allows fine individual control over record creation, mass-creation (import), editing, mass-editing, deletion and mass-deletion.

Setup → Access → Manage Groups

Overview | General | Table | Menu-Specific | **Permissions** | Field Permissions | History

[Back](#) [Next](#) [Finish](#) [Cancel](#)

The permissions you set using this dialog will apply to the table: Change Request

A record is owned by the person whose own Full Name field matches the Requester field

Select what the group is authorized to do with this table.

<p>Allow this group to</p> <ul style="list-style-type: none"> <input type="checkbox"/> Create Change Requests <ul style="list-style-type: none"> <input type="checkbox"/> Import multiple Change Requests from a file <input type="checkbox"/> Copy Change Requests 	<p>Allow this group to</p> <ul style="list-style-type: none"> <input type="checkbox"/> Delete their Change Requests <input type="checkbox"/> Delete other people's Change Requests <ul style="list-style-type: none"> <input checked="" type="radio"/> All <input type="radio"/> Only those that satisfy the saved search: <div style="display: flex; align-items: center;"> <input type="text" value="C: Approved in completion date order"/> Edit Create New Search </div> <input type="checkbox"/> Mass delete multiple Change Requests in a single operation
<p>Allow this group to</p> <ul style="list-style-type: none"> <input type="checkbox"/> Edit their own Change Requests <ul style="list-style-type: none"> <input checked="" type="radio"/> All <input type="radio"/> Only those that satisfy the saved search: <div style="display: flex; align-items: center;"> <input type="text" value="C: Approved in completion date order"/> Edit Create New Search </div> <input type="checkbox"/> Edit other people's Change Request <ul style="list-style-type: none"> <input checked="" type="radio"/> All <input type="radio"/> Only those that satisfy the saved search: <div style="display: flex; align-items: center;"> <input type="text" value="C: Approved in completion date order"/> Edit Create New Search </div> <input type="checkbox"/> Mass Edit multiple records in Change Request in a single operation <input type="checkbox"/> Link multiple records using the Link menu <input type="checkbox"/> Print records using the printer icon 	<p>Allow this group to</p> <ul style="list-style-type: none"> <input type="checkbox"/> View their own Change Requests <ul style="list-style-type: none"> <input checked="" type="radio"/> All <input type="radio"/> Only those that satisfy the saved search: <div style="display: flex; align-items: center;"> <input type="text" value="C: Approved in completion date order"/> Edit Create New Search </div> <input type="checkbox"/> View other people's Change Requests <ul style="list-style-type: none"> <input checked="" type="radio"/> All <input type="radio"/> Only those that satisfy the saved search: <div style="display: flex; align-items: center;"> <input type="text" value="C: Approved in completion date order"/> Edit Create New Search </div> <input type="checkbox"/> View FAQs for the Change Requests <ul style="list-style-type: none"> <input checked="" type="radio"/> All <input type="radio"/> Only those that satisfy the saved search: <div style="display: flex; align-items: center;"> <input type="text" value="C: Approved in completion date order"/> Edit Create New Search </div> <input type="checkbox"/> Export multiple Change Requests to a file <input type="checkbox"/> Show conversion button on menu action bar <input type="checkbox"/> Allow interaction with conversion rules

[Back](#) [Next](#) [Finish](#) [Cancel](#)

Typically the user's "own" records are those that he created, however the definition of the ownership can be changed in Setup/Table/Edit wizard. For example, a record can be "owned" by everyone in a particular Company.

Setup → Access → Manage Groups

Overview | General | Table | Menu-Specific | **Permissions** | Field Permissions | History

Back Next Finish Cancel

The permissions you set using this dialog will apply to the table: Team

A record is owned by the person whose own Full Name field matches the Team Leader field

Select what the group is authorized to do with this table.

Allow this group to

- Create Teams
 - Import multiple Teams from a file
 - Copy Teams

Allow this group to

- Delete their Teams
- Delete other people's Teams
- All
- Only those that satisfy the saved search:
 - Support or Staff Team
- Mass delete multiple Teams in a single operation

Allow this group to

- Edit their own Teams
 - All
 - Only those that satisfy the saved search:
 - Support or Staff Team
- Edit other people's Team
 - All
 - Only those that satisfy the saved search:
 - Support or Staff Team

Allow this group to

- View their own Teams
 - All
 - Only those that satisfy the saved search:
 - Support or Staff Team
- View other people's Teams
 - All
 - Only those that satisfy the saved search:
 - Support or Staff Team

Allow this group to

- View FAQs for the Teams
 - All
 - Only those that satisfy the saved search:
 - Support or Staff Team
- Export multiple Teams to a file
- Show conversion button on menu action bar
- Allow interaction with conversion rules

Back Next Finish Cancel

Access permissions can be

Security is built into the core of the Agiloft system and enforced consistently. If a user does not have access to a table, they cannot see entries from that table. If a field from the table appears as a drop down list in another table (for example the Assigned To field), the user will not be able to choose any values for that field until they have view access to the table. This is an example where you might want to give a group access to a table, but not show the table on the main toolbar.

Field Permissions Tab

Setup → Access → Manage Groups

Overview General Table Menu-Specific Permissions **Field Permissions** History

Back Next Finish Cancel

The permissions you set using this dialog will apply to the table: Helpdesk Case

“View own” sets the ability to view the field in records that 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 the field when creating it.
 “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 set by the first field in the set (which is not indented). You can modify the view permissions for the individual fields in the set. If create and modify permissions are on for the set, any field that can be viewed can theoretically be modified (unless its display type is view only).

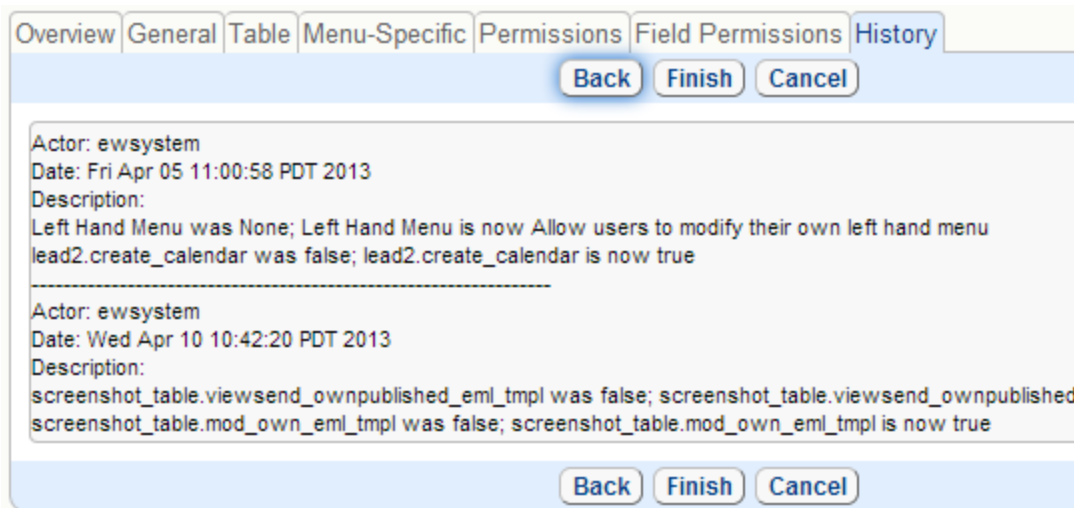
Field	View own	Create	Edit own	View other	Edit other
Additional Notes To/From Customer	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
All Tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
All Time Entries	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Alternate - Assign to Me	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Alternative Assigned Team/Person	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Asset ID	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Asset Model Name/Number	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Asset Name	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Asset Owner Name	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Asset Responsible Team	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Asset Subtype	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Asset Tag	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Asset Type	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Assigned Person	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

These fields are part of a linked field set. Their edit and modify permissions are set by the first field in the set (which is not indented). You can modify the view permissions for the individual fields in the set.

These fields are independent.

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 a handy feature to have when multiple administrators may be editing the system.



Knowing Which Tables Need Work

Since you can't see everything at once in the wizard, you may not know which tables you need to adjust for a given group.

You may wish to print the group's permissions either hardcopy 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.

On the Setup/Access/Manage Groups screen, mouse over the printer icon:

The options allow printing of permissions and/or history of log of changes. A comparison report may be printed showing differences between any selected groups.

Groups
Status: 20 record(s) found, 1 pages. Click [here](#) to re-count records. 1 records are selected

New Copy Delete

Views

<input type="checkbox"/>	Edit	ID	Group Name ↑	Description
<input type="checkbox"/>		257	admin	admin
<input type="checkbox"/>		312	adminimport	adminimport
<input type="checkbox"/>		303	Anonymous	Group for Anonym sent to a user wh
<input checked="" type="checkbox"/>		307	Contract Creator	Internal employee view all contracts
<input type="checkbox"/>		308	Contract Manager	Internal employee
<input type="checkbox"/>		263	Customer	Customer - end user
<input type="checkbox"/>		301	Customer Manager	Customer Manager - will be able to view project information relate projects
<input type="checkbox"/>		306	Document Creator	Can create documents and edit their own - customer of document
<input type="checkbox"/>		309	Document Manager	People who can approve and publish documents
<input type="checkbox"/>		286	Guest	Used in hyperlinks to allow users to create new lead, new user, n without seeing rest of gui

Print records

- All Current Permissions
- All Permissions with History Log
- History Log only
- Groups permissions comparison report
- New HTML Template
- New Word/PDF Template
- Edit...
- Manage...
- Help

See Also:

Left Hand Pane Setup

RULES AND WORKFLOWS

Workflow refers to the rules that apply when changes are made to a particular type of record.

For example: "When a bug report is created in a state of Open, email Engineering, when it moves to In Testing, email QA, when it is Closed, email the person who submitted it"

The various stages (i.e. Opened, In Testing, Closed) that the record can be in are referred to as states and the workflow engine allows the creation of guards that control how the states may be changed. For example "Do not allow the State to be changed from Open to In Testing unless the 'Cause of Bug' field has been filled in"

Changes in state are referred to as state transitions and the rules that are applied when a workflow changes from one state to another are referred to as actions.

So, in the example above, we have:

States: Open, In Testing, Closed

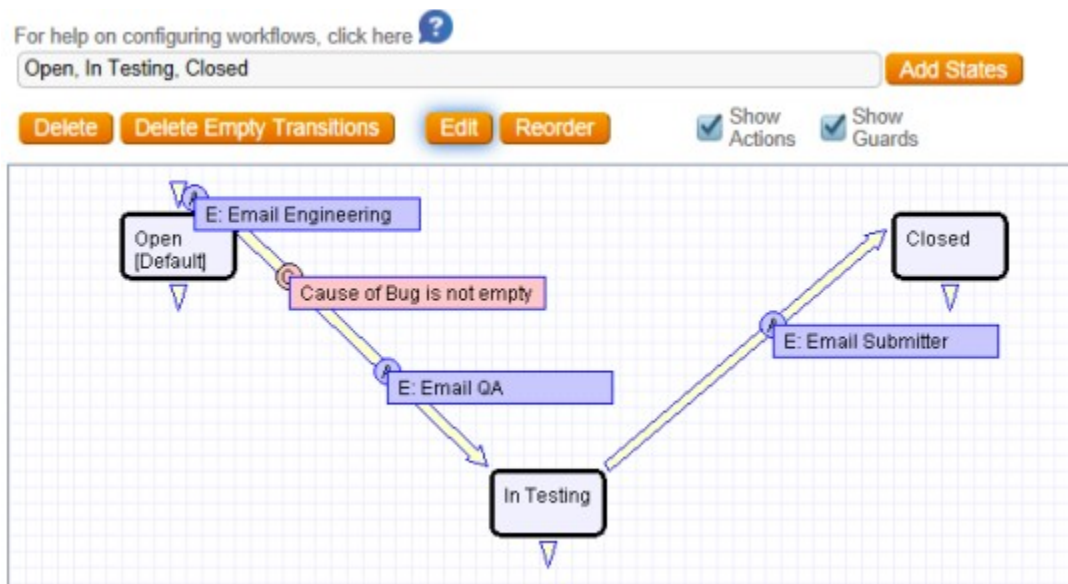
State Transitions: Open-> In Testing, In Testing-> Closed

Actions: email Engineering for creation/Open transition (i.e. creation of record)

email QA for the Open-> In Testing transition

email submitter for the In Testing-> Closed transition

Guards: Do not permit Open-> In Testing transition if "Cause of Bug" field is empty



All this is a lot easier in practice, using the GUI, than it seems on paper.

Agiloft provides a graphical interface that makes it easy for the administrator to see and edit the workflows that apply to a particular type of record.

Tables inherit the workflows from their parents. Consider the following hierarchy:

- Contact**
- + **Lead**
- + **Customer**

If you create a workflow for Contact, then select Lead and look at its workflow, you will see the Contact workflow that you just created has been applied to it as well.

However, workflows do not have to be the same between parent and child tables. For example, you can augment the workflow for Lead by adding additional states, state-transitions or guards. This will not alter the workflow for Contact, nor will it break the linkage to the workflow for Contacts - if you add additional states/transitions/actions to Contacts, they will automatically show up in Leads.

In brief, sub-table workflows don't contain a copy of the parent workflow, they contain the parent workflow plus their own additions to it.

If you are in a sub-table (Lead in the example above) and try to delete one of the state transitions or edit one of the actions that was inherited from the parent, the system will warn you that this change will affect the parent workflow.

See Also:

[Business Example](#)

[Actions](#)

[Rule](#)

[Record](#)

[Object](#)

[Flash Tutorials](#)



WORKFLOW WIZARD

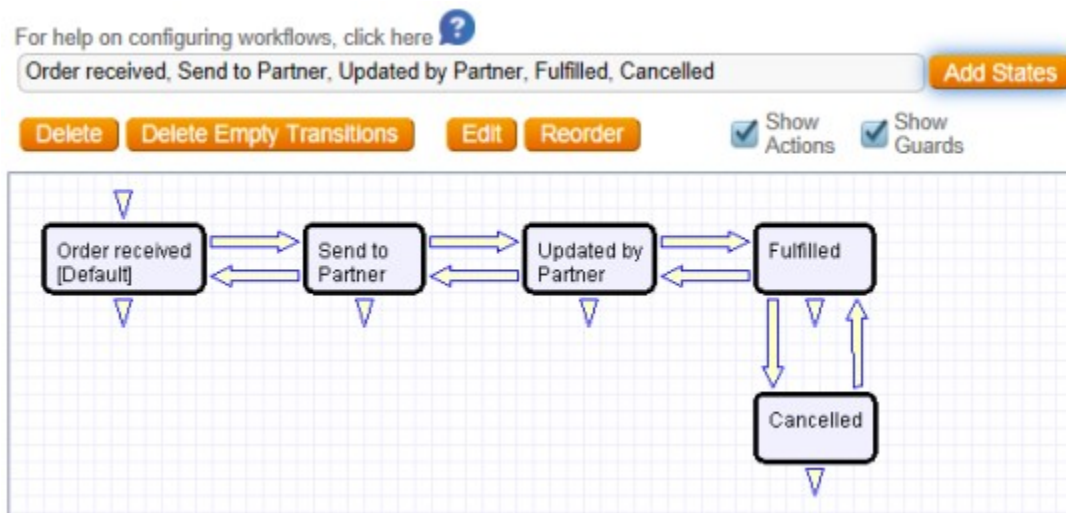
To create or edit a workflow, go to Setup/Workflow/Select table to edit/Edit.

Once you have created and saved your table's Workflow, Agiloft automatically creates a choice field containing all the workflow states you indicated.

You need to have Java installed to work with the workflow screens - if it's not there, the program automatically installs it

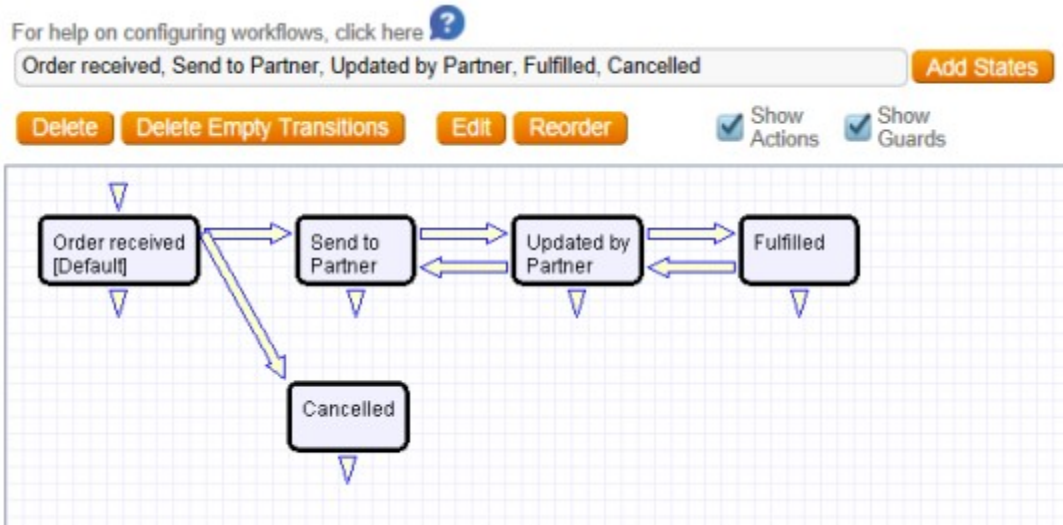
Adding States

Example: Say we have some states in a choice field: Order received, Sent to Partner, Updated by Partner, Fulfilled, and Cancelled. Type them in the text entry box and click the Add States button. States are added with simple (but not necessarily realistic) transitions set automatically, as shown below:



- Add state transitions by selecting a state's corner and dragging to another state (for example: Order received -> Cancelled).
- Arrange your workflow chart however you like by selecting a state and dragging it to another location.
- Delete transitions by selecting the line between the two states and clicking Delete (for example: Send to Partner -> Order received). This will prevent the record from being moved from the first state of the transition to another.
- Select a state and click checkboxes to make one of the states default, the record creatable or deletable in each state.

Your workflow might now look something like this:



To make a workflow state accessible when creating a ticket use check-box "Creatable" on the right side of the wizard. The default state is always creatable. The system can be configured to allow deleting record only in certain state and refuse to do so in others. Use Deletable check-box for this. By default any state is designated as "deletable".

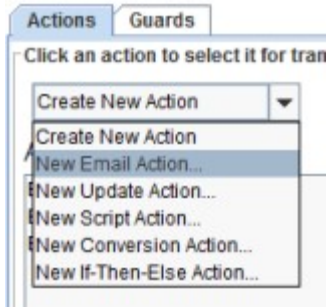


Adding Actions

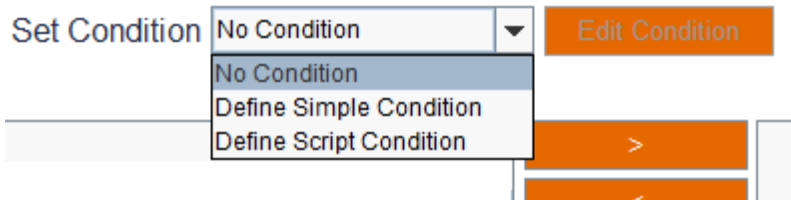
To create an action triggered by new record creation, click the triangular caret above the state and add the action there. If no caret is available, you must first make this a creatable state.

Example: You need to add an action to send an email when an order is received.

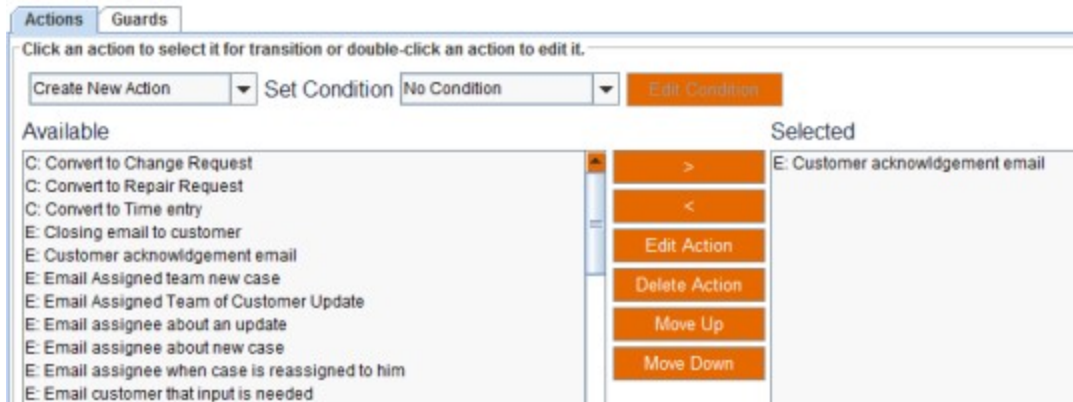
- Double click on the caret to bring up the action dialogue, then choose Action from the Create New Action drop-down.



- For this action you'll need to specify that an action should only be executed if search condition is met. A condition is a guard applied to an action rather than a state transition and prevents the action from occurring if its criteria are not met. Double click the caret to open the action



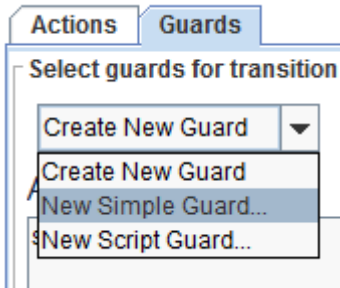
- Select the action with the mouse, then mouse-over the No Condition drop-down and select Define Simple Condition
- "Move Up" and "Move Down" buttons are available in the workflow's transition wizard to specify the order actions and guards execute in.



Adding Guards

Example: Suppose you need to add a guard to the transition to fulfilled that prevents it from being changed to Fulfilled unless the field labelled Shipped Date has been given a value.

- Double click the transition, and click the Guard tab, then mouse over Create New Guard and select New Simple Guard:



- You can define the guard based on search criteria for other fields, for example, you might create a guard to specify that the Solution field is not empty.

Note: The guard condition must be true for the state change to be allowed.

Saving the Workflow

Be sure to click the Finish button to save and apply changes. No matter how complete the diagram looks, it is not saved until you click Finish.

Click the Finish button to bring up another dialogue that allows you to define which groups can view workflow actions and override them. Select the groups you want to give these privileges to and click Finish again. Now the Workflow changes are saved

Override Workflow Actions

Finish
Cancel

Please select the groups that should have control over the execution of this workflow.

If 'visible' is checked, members of the group are informed that the action or guard is about to execute.

If 'optional' is checked, the group is able to cancel the action or guard.

Overriding:	Visible	Optional
admin	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Customer	<input type="checkbox"/>	<input type="checkbox"/>
Support Staff	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Internal Customer	<input type="checkbox"/>	<input type="checkbox"/>
Guest	<input type="checkbox"/>	<input type="checkbox"/>
Selfregistered User	<input type="checkbox"/>	<input type="checkbox"/>
Sales	<input type="checkbox"/>	<input type="checkbox"/>
Customer Manager	<input type="checkbox"/>	<input type="checkbox"/>
Professional Services	<input type="checkbox"/>	<input type="checkbox"/>

Now, if you go back to the Fields section of your table setup, you will see a field called "Workflow State" or "wfstate". The choices in this field automatically map to those in the Workflow diagram you just set up; there is no need to create a separate choice field.

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 when the necessary functions are not supported by the GUI, custom scripts may be used. 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.

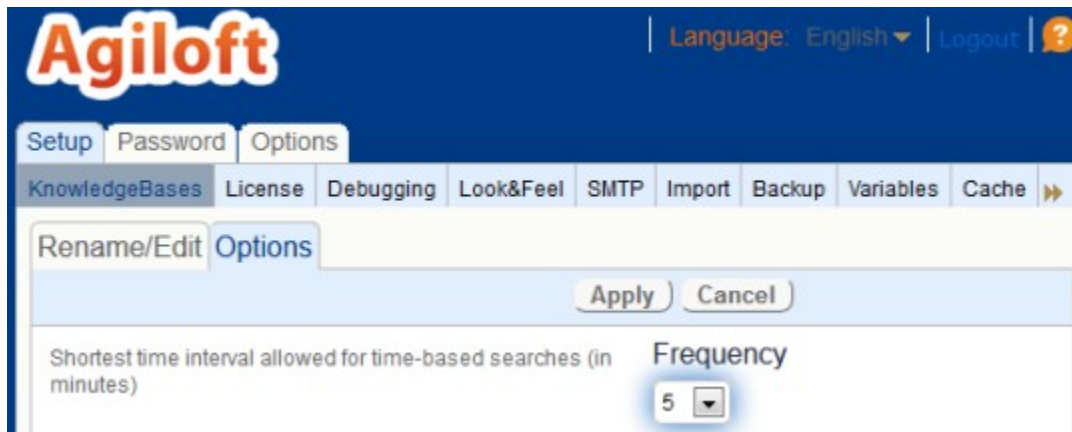
Time-based Rules

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)".



Aggregate based 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).

See Also:

[Actions](#)

[Rules Misc](#)

[Concepts](#)

[Record](#)

[Field](#)

[Workflow](#)

[Flash Tutorials](#)



RULES MISC

Edit/Delete Rule

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 de-activate a rule:

1. Go to the Setup/Rules
2. Select the rule in question and open it for editing
3. Switch to the Action tab
4. Select all actions in Active Actions window
5. Move them to Inactive Actions

Rules 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.

Rules Triggering 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 in /Setup/Rules/Condition.

Apply Rule:

When a Contract is

Created

Edited

Include edits made by other rules during record creation

Deleted

Only when the change occurs by means of:

Email

Web

API

Uncheck the API box to prevent a rule from being triggered by the actions of another rule.

→

Practical 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.

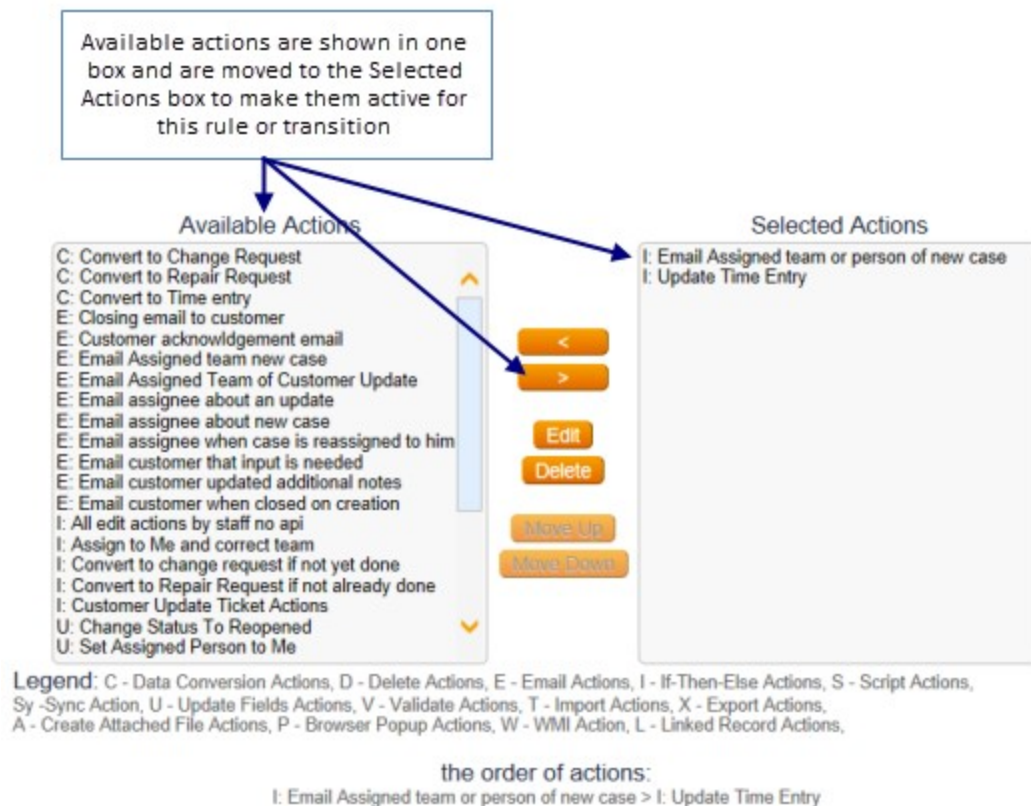
Rules based on the Activity Logs table

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 provide the automation to the system.

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.



Below is a list of types of actions available for rule creation:

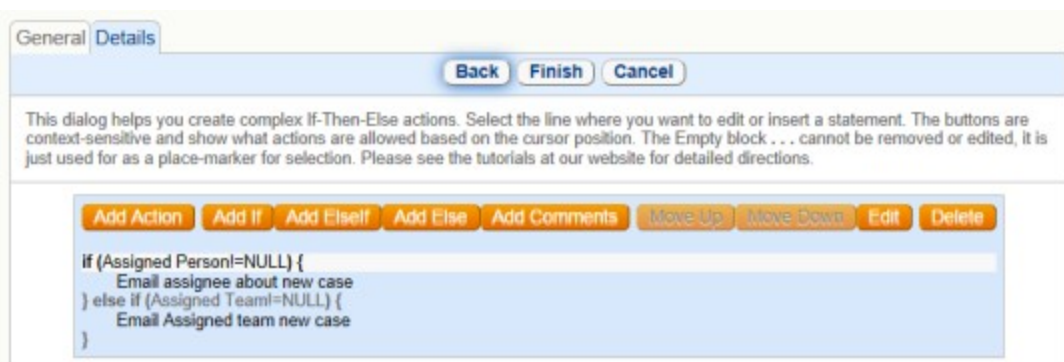
- If-Then-Else Action
- Email Action
- Update Fields Action
- Validate Action
- Data Conversion Action
- Import Action
- Export Action
- Delete Action

- Browser Popup Action
- Sync Action
- Script Action
- Create WMI Action
- Create Linked Record Action
- Create Attached File Action

If-then-Else Action

An If-Then-Else Action is used to execute a variety of actions based on specific different criteria.

For instance:



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

When running a time based rule, it is strongly suggested that as many conditions as possible be put in a Saved Search filter, not in the If-Then-Else action. The filter runs much faster than the If-Then-Else action against every record can slow down system performance.

Email Action

An Email Action is used to send an email to someone.

While creating the email action you will have to create an actual email message. That's done in the Email Wizard similar to the one that you would use to send an email from the record.

Note that you can populate your message from the template. For example choosing [Contract for \$company_name needs to be renewed] will result in the text like this:

The screenshot shows an email composition interface. At the top, there are 'Send' and 'Cancel' buttons. Below them are tabs for 'Insert', 'Format', and 'Multiple Email Options'. A dropdown menu is open under 'Populate from template', showing several options: 'Contract #\${id} has been submitted for approval', 'Contract for \$company_name has been approved', and 'Contract for \$company_name needs to be renewed' (which is highlighted in orange). Below the dropdown is a search bar. The email fields are populated with variables: the subject is 'Contract for \$company_name needs to be renewed', and the HTML text area contains a message template: 'The Contract for \$formula(\$contract_is_for) described in more detail below is due for renewal notification. Please take the appropriate action to cause the renewal email and quote to be sent. Click [here](#) to update the record directly.' At the bottom, there are 'Attachments: No Files Attached' and a list of 'Fields Included: Company Name, Contract End Date, Renewal Type, Status, Contract Title, Contract Start Date, Company Main Contact, Contract is For, ID, Contract Description, Asset Name'. 'Send' and 'Cancel' buttons are at the very bottom.

As you can see text contains variables that will be evaluated upon the action execution time.

Update Action

An Update Fields action is used to modify one or more fields in the selected record or in a linked record in another table. In the action wizard you will need to choose fields you want to be modified and you can enter new field values directly, or use the Formula radio button to set field values from existing fields or a formula.

There is an option to use a regexp (regular expression) in update field actions. This type of replacement also accepts special characters. Example: use regexp `'/;+/\n/` to replace `';` with a line break.

Validate Action

A Validate Action is used to generate an error message and reject the change if the record meets the search conditions specified on the Condition tab. Basically you will need to provide a name for the validation action, describe the purpose of the validation action and come up with the message you want the user to see in the status area that explains what he must do to be able to save the record.

Data Conversion Action

A Data Conversion Action is used to automate the conversion of a record into a record or records in another table using predefined conversion rules. The Data Conversion Wizard guides you through field mapping and the different options for executing conversion actions.

Import Action

There are many situations in which it is desirable to run a regularly scheduled import/update of records in a table. Typically synchronizing users with some other system might be done this way. The rule action type "Import" can be run as part of a time based rule. To enable this, create an "Import action" rule on the table into which you want to import records. As you define the action, you will walk through the normal import wizard, map the fields from a sample file, define the location of the import file and so on.

In Import Action Wizard:

- Specify path of the import file,
- Upload a sample so that fields may be mapped. The first row of the file should contain the same column names as all future versions of the file. The field mapping from this sample file will be used for all future imports. If the first row headings in your import file change, you will need to upload another sample and remap the fields.
- Define the data type of the file. Automatic import supports tab delimited ASCII, standard ASCII formats or delimiters with characters other than tab. To use another delimiter, on the action's Data Source tab, define the character that separates each field, and the character that encloses each field.
- Set up a confirmation that emails a log file after the import action has taken place. Type any email addresses into the box separated by commas. The log file is a text file containing any status messages generated by the import action.

Export Action

There are many situations in which it is desirable to run a regularly scheduled export of records in a table. Export actions within rules allow running regularly scheduled exports of records in a particular table.

The action is defined using an identical wizard to that of a regular one-time export.

In Export Action Wizard:

- Specify destination path of the export file. The file may be exported to the TMP directory on the server, the home directory on the server, any other directory on the server, or directly to an ftp server.
- Define the data type of the file. Automatic export supports tab delimited ASCII, standard ASCII formats or delimiters with characters other than tab. To use another delimiter, on the action's Data Source tab, define the character that separates each field, and the character that encloses each field. Excel and XML files are also supported.
- Select the fields from within the record to export. By default, all of the fields in a record will be marked for export.
- Set up a confirmation that emails a log file after the import action has taken place. Type any email addresses into the box separated by commas. The log file is a text file containing any status messages generated by the import action.

Delete Action

Rules can now run a new action called a Delete action that automatically deletes the found records. This is useful for automatically deleting records marked as spam by staff or for automating timed cleanout of old records, for instance "Delete all records older than 3 years and not published as FAQs".

Browser Popup Action

It may often be advantageous to provide pop-up notifications in addition to the default e-mail notifications set up by Agiloft. A Browser Pop-Up action will display a message for logged in users. This message format supports plaintext or HTML formatting where needed. The Browser Popup action wizard also controls the users to which the pop up notification is displayed.

Sync Action

The Sync Action allows a sync to be formed as part of a rule. This may consist of syncing with an external system or syncing an entity set within the Agiloft system. The Sync functionality is currently in Beta. Additional information may be found on the Agiloft website.

Script Action

Scripting is an advanced option and allows you to execute a script on the action.

Make sure that name of script points to a file in your script directory. 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 and install the file in the directory {Agiloft Installation Directory}/data/{kb name}/scripts. If the script has an extension of .pl, it will be automatically executed by the Perl interpreter. If the script has an

extension of .jar or .class, it will be dynamically loaded and run as a java class. Files with extension .bsh will be executed as BeanShell scripts. Files with an extension of .exe or any unrecognized extensions will be treated as executables.

WMI Action

Windows Management Instrumentation integration is provided through WMI actions for use with asset discovery and polling.

Linked Record Action

This type of rule action enables the copying of information from the current record into linked records down to one or more levels. The main purpose of this action type is to enable one to copy a value from the current record into one or more linked records. It is always possible to update linked records using a normal Field Update action, but the update had to be to a hard coded value - it couldn't copy a value from the current record. With this action, you can copy a value such as the record ID from a record to linked records three levels away without creating intermediary linked fields, thus triggering a relationship between those records that did not previously exist.

As an example, let's use Service Request table to manage new employee setup requests. When a new employee setup request is created, the user selects from multiple linked template tasks, and when the service request is saved, the ID of the request is copied into a field in the linked template task records called Last Request ID, a conversion rule is run to create new tasks from those templates that map that ID into a field linked back to the Service Request so the new tasks are linked to the service request and show up there immediately.

To configure a linked record update, select a linked field that points to a record you wish to update. If desired, you may select a linked field in that table that points to a record in another table, creating a chain of linked updates.

To define the action to run on those records, select a node in the chain and then select/add actions to be executed at this level. For instance, you can add an Update Fields action that will update a field in the linked record by copying any value from the current record.

Attached File Action

Print formats can be used to generate a pdf or word document and attach it automatically to a record.

When this action is run, the print format is executed, a file created, and the file is attached to the defined field in the manner selected. This is a regular action and will create a history record and be able to trigger other actions, like sending the file by email.

DATA MODELING

There are many ways in which Agiloft can be configured to address the special needs of individual companies. This section reviews some of the most important decisions and the pros and cons of various approaches.

You probably will not have to make any difficult decisions because the application ships pre-configured for various uses and configuration is included in the fixed price implementation package, but it is valuable to know what options are available when you are considering additional uses for the software. These options are all related to data modeling, which is a fancy way of saying, "how is information represented", as illustrated by the following examples and discussion.

Some of these examples do not have a right/wrong answer – it is bit like asking whether a car is better than a minivan – the answer depends on how many people/bicycles/dogs/etc you are planning to transport. The following rules will provide guidance if you are faced with difficult decisions.

1) Keep things as simple as possible from the user's perspective. A good way to judge how simple two alternative solutions would be is to ask how difficult each would be to describe them to a colleague. The solution that is easiest to describe is usually the best.

2) The most logical representation is usually the best. This is really another aspect of rule (1), and just means that you should represent information the way that you think about it. For example, if, in your business, Suppliers are always individuals that you work with directly (for example the gardener, janitor, etc), you probably think of them as people, in which case it is reasonable to represent Supplier as a type of Contact. If however, your suppliers were Companies, you would create Supplier as a type of Company.

3) Represent information as explicitly as possible. For example, if you have a table for collecting bug reports, it is better to have a field named Reproducible (typically a Yes/No choice field) and another field Steps to Reproduce Issue than just to have a field "Steps to Reproduce Issue" which may be left blank if the problem is not reproducible.

See Also:

[Sub-tables vs. Choice Fields](#)

[Separate Table vs. Sub-table](#)

[Reference Other Table vs. Create Independent Fields](#)

SUB-TABLES VS. CHOICE FIELDS

Example

A financial services company could represent high net worth individuals as a Sub-table of Customer, or have a Yes/No flag named High Net Worth flag in the Customer table with some additional dependant fields if the value is Yes.

Decision

Create a separate Sub table for each type of data or add a choice field to an existing table and create dependant fields for any additional information.

Pros of separate sub-tables

Each sub-table can have a separate workflow.

Each sub-table shows only the fields that are relevant to it, for example, there is no need to have a High Net Worth flag in the High Net Worth table.

There is no need to specify that the additional fields are dependant.

Each sub-table can have a separate layout.

Each sub-table can have separate group access permissions.

Pros of choice field/dependant field approach

The fewer tables there are, the easier it is for users to find the right one for entering data.

The flexibility of having separate layouts, access permissions and workflows can mean higher maintenance overheads, though Agiloft tries to minimize this through options to add new fields or change access permissions to a table and all its sub-tables in one operation.

Recommendation

If you think of the information as being fundamentally different (for instance apples versus oranges), especially if each type of information has several different fields, then create separate sub-tables.

If you think of the information as being similar (for example, red apples and green apples) and anticipate applying the same workflows to each, then use choice fields and a few dependant fields if necessary.

See Also:

[Data Modeling](#)

[Separate Table vs. Sub-table](#)

[Reference Other Table vs. Create Independent Fields](#)

SEPARATE TABLE VS. SUB-TABLE

Example

A hardware manufacturer might create top level tables (i.e. separate tabs) named Customer and Reseller or a table named Company, with sub-tables named Customer and Reseller.

Decision

Create a top level table or create a sub type of an existing table.

Pros of separate tables

Each table is totally independent.

Pros of sub-tables

Agiloft supports instant conversion of records from one sub table type to another. For example a record of type Company/Reseller can be converted to Company/Customer in an instant by changing the type field. But it is not possible to convert from one top level table type to another without setting up a conversion operation. For example, a Contact cannot be turned into Company without conversion rule or conversion action.

It is easier to maintain sub-tables than independent tables.

Sub-tables do not take an additional tab on the screen.

Recommendation

Use sub-tables unless the types of information are really different. A good question to ask is "Is XXX a kind of YYY" ? If the answer is Yes, you can use sub-tables.

See Also:

[Data Modeling](#)

[Sub-tables vs. Choice Fields](#)

[Reference Other Table vs. Create Independent Fields](#)

REFERENCE A TABLE VS. CREATE INDEPENDENT FIELDS

Problem Statement

The Name of Primary Contact field in the Company table could be a simple field or it could reference a record in the Contacts (i.e. a [Link to single field from another table](#) field)

When is the former better than the latter?

Pros of referencing another table

If there are multiple references to the same piece of information, it is easier to maintain it one place, especially if that information is likely to change. You might create a Link to Multiple Fields from Single Table reference to pull in the person's Name, Telephone Number and Email address from the Contact table. If the Telephone number changed in the master record in the Contact table, that change would automatically be reflected in all the records that reference it.

It is easier to keep the data consistent. When users enter freeform text, there are likely to be minor variations in spelling, capitalization, etc.

Pros of creating an independent field

It is less work to setup the system.

It is easier for users to enter a small piece of information than use a pop-up wizard to look up the reference in another table.

Recommendation

If the information has multiple fields that are always linked, is accessed frequently, or is likely to change and must be kept up to date across several tables, then store it in its own table and create a reference to it. If the information is just being used in one place, then create an independent field.

A good question to ask is "If the information is stored in an independent field, how many times are we going to re-enter it?" For instance, a user's contact info stays constant and is needed for every ticket they submit, so it would be efficient to make this a reference to their record on the Contact table. But the name of a CEO of a Company could reasonably be held as an independent field in the Company table.

See Also:

[Data Modeling](#)

[Sub-tables vs. Choice Fields](#)

[Separate Table vs. Sub-table](#)

ACCESS METHODS

LDAP/MICROSOFT ACTIVE DIRECTORY

LDAP Introduction

- LDAP provides a central repository of user information, passwords, and other data
- Microsoft Active Directory is an LDAP implementation with proprietary extensions
- A mapping is created between standard or custom LDAP attributes and Agiloft fields in the Employee table (including group and team membership) to allow synchronized logon
- Integrating Agiloft successfully with LDAP will most likely require additions to your LDAP database.

So, LDAP can hold any kind of data, but it is usually used to provide a central repository of user information and passwords. This allows other enterprise applications to check passwords against a single LDAP repository, rather than each application storing them individually. This reduces maintenance costs and improves security.

In addition to login names and passwords, Agiloft allows other information, such as email addresses, groups, teams and custom fields to be mapped between LDAP attributes and their equivalents in Agiloft.

The use of LDAP for more than login/passwords provides some interesting challenges for Agiloft as detailed below:

1) Both LDAP and Agiloft may contain custom attributes (aka fields), Unfortunately, there is no way to find all the custom attributes in an LDAP database, other than by reading the users one-by-one and noting their attributes. For LDAP databases with hundreds of thousands of users, this could take hours.

Agiloft resolves this problem by searching the first 1,000 users for attributes, and asking the administrator to nominate one user that contains the additional attributes that must be mapped. The actual values of the attributes in this user do not matter, they just need to exist. For example, if the user has a Telephone Number attribute of 0, the system will add Telephone Number to the list of mappable attributes.

Workflow rules that send email to a Team need to know what users are part of what teams. The Agiloft system can instantly search its own tables with an SQL query to find all the matching users, but there is no way to perform an equivalent search on the LDAP database. Instead it would be necessary to read every LDAP user in turn to determine whether they were a member of that team. For large LDAP repositories, this would be very slow.

This class of issues is resolved as follows: When a user logs into Agiloft using LDAP authentication, a dummy entry is created for them in the Agiloft user table and the data cached in this entry is refreshed from LDAP each time they subsequently login. This allows the system to rapidly find Team members and avoid unnecessary calls to the LDAP server. Of course, password information is not cached, so centralized password control is maintained.

This strategy allows Agiloft to automatically restrict Team emails to those LDAP users who actually use Agiloft. If you want all users to receive email, you can set it up to automatically sync with LDAP at regular intervals and use it at least once, but some administrators see this as an advantage since users who do not use Agiloft may not expect to receive email from it.

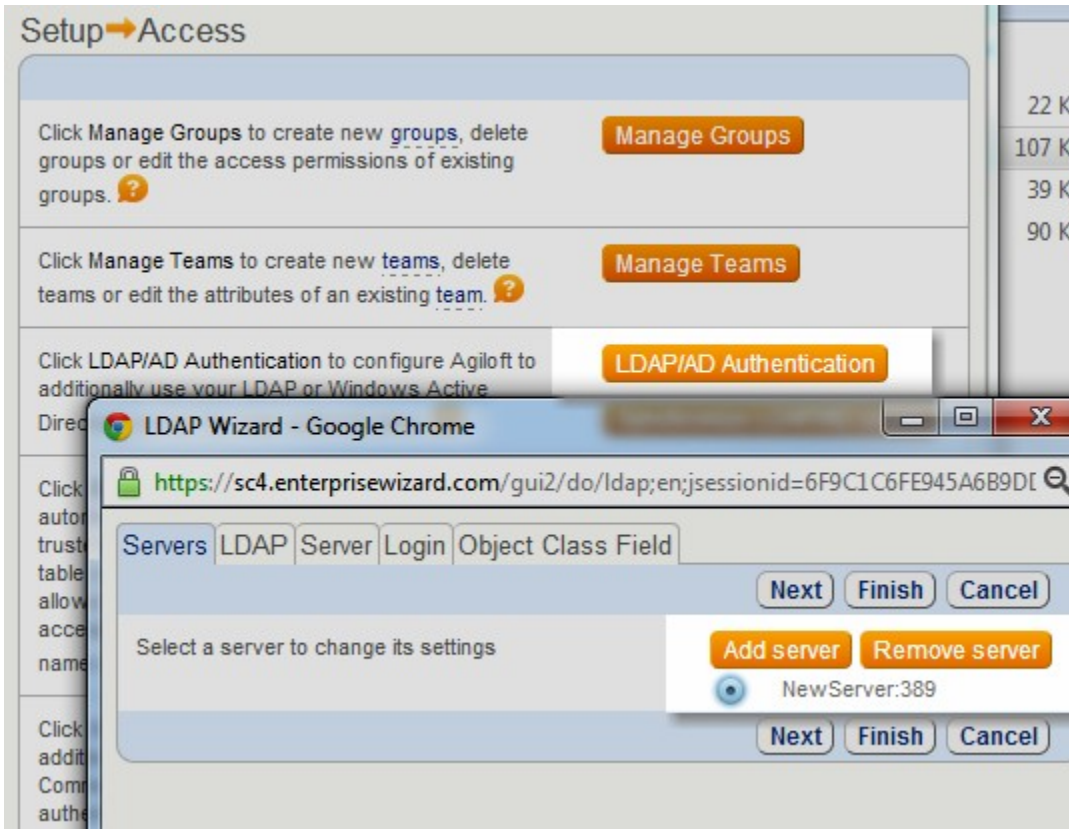
LDAP Integration

LDAP can be used by Agiloft in three ways:

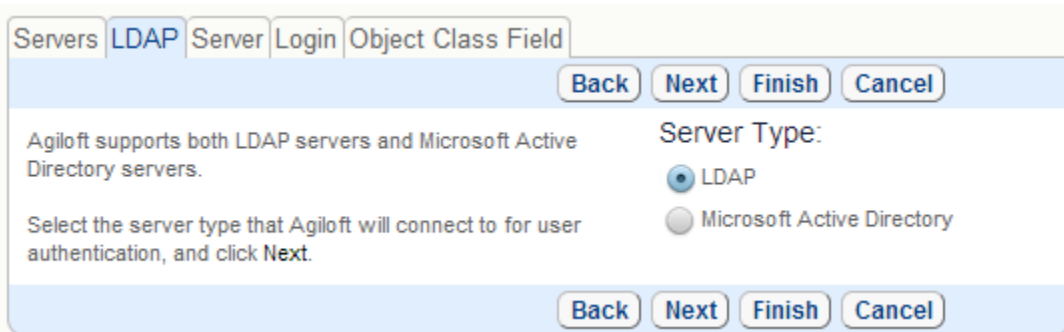
- Synchronize: Agiloft will import LDAP users automatically at a specified interval, or manually using the GUI. LDAP users will be Agiloft users even if they have never logged in to Agiloft.
- At login: LDAP will be queried for a specific user when they attempt to log in to Agiloft.
- Single Sign-on (SSO): Using a special hyperlink and an ActiveX control, LDAP will be queried with a user's domain login credentials so that they can access Agiloft without an explicit login.

LDAP Mapping Wizard

- To define the LDAP mapping go to Setup/Access and click LDAP/AD Authentication.
- Add the server or select the existing one. New server is always added with the temporary name and a default port (389). Once the server is selected these settings can be changed on the tab Server.



- Next choose the server type. Use "LDAP" for non-Microsoft servers.



- Enter the name of your company's LDAP server and the base directory. Common schemes for base directories:
- Domain: "dc=mycompany, dc=com" specifies each domain component of mycompany.com in order.
- Geographical: "c=US" specifies the country. This is a throwback to LDAP's X.500 roots, which was potentially one big global hierarchy.

This is the standard LDAP TCP port. SSL LDAP servers are usually on poart 636.

Servers LDAP **Server** Login Object Class Field

Back Next Finish Cancel

Type the name or IP address of the computer that hosts your LDAP server. (Example: *ldap.mycompany.com*)

*Server Name
10.64.0.101

If the LDAP server uses a non-default port, enter it here.

*Port
389

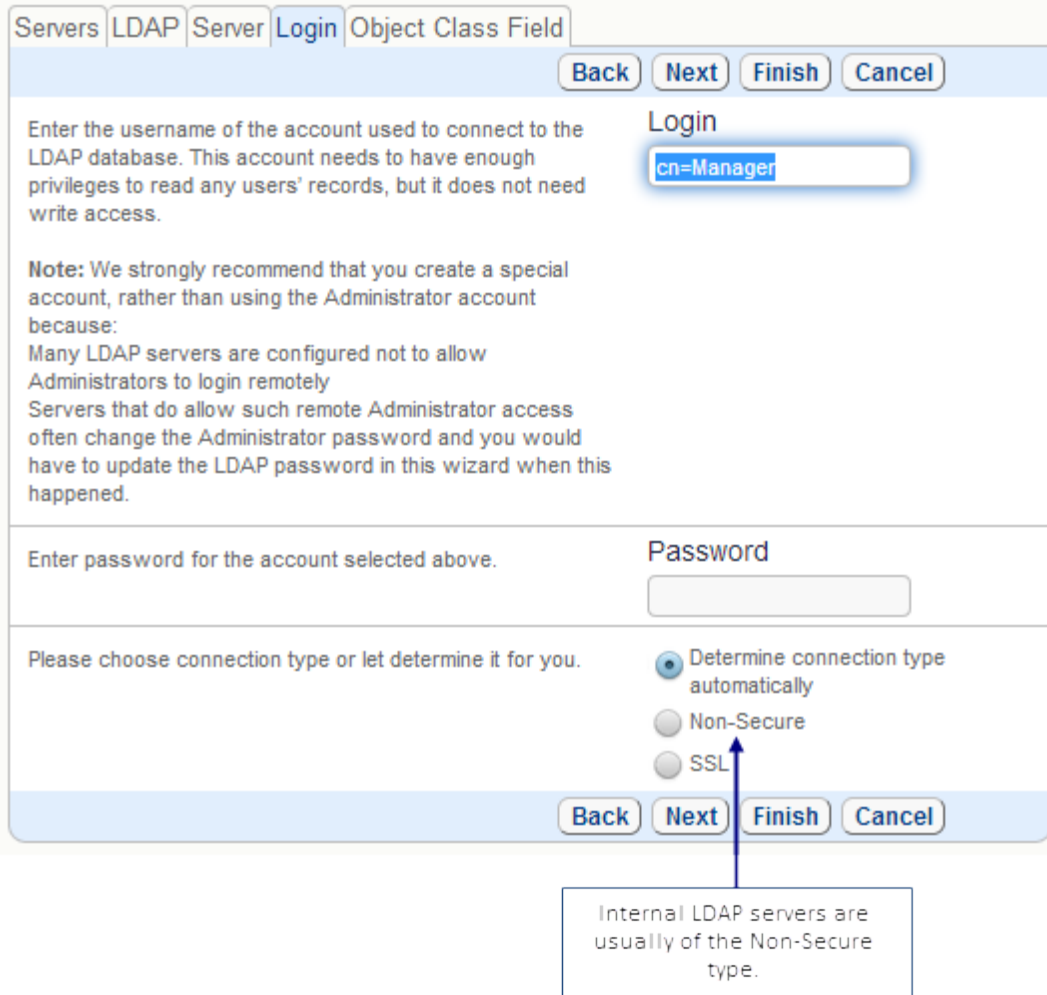
For example, you might specify 636 for Secure LDAP.

Enter the base directory on the LDAP server. Base directory is also known as the base DN, and it usually corresponds to the domain name. Example: dc=company, dc=com

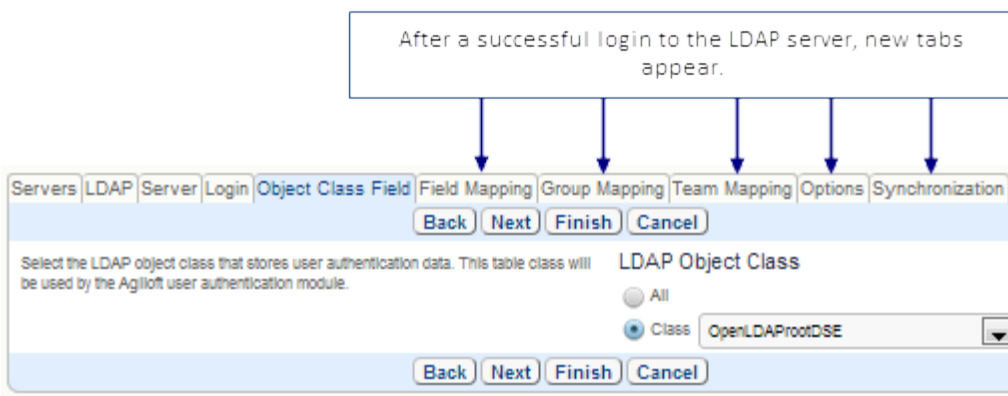
*Base Directory
dc=agiloft,dc=com

Back Next Finish Cancel

- Enter the login authentication info for your LDAP server and the connection type:



- Object classes separate LDAP attributes into logical categories. The "user" class is often used for entries that have the appropriate user authentication attributes. If you want to restrict Agiloft logins to certain users, you could add a custom objectClass such as "EWusers" to these users and specify it here.



Agiloft Reference Manual

- LDAP provides no easy method for finding all attributes in use, so Agiloft scans all attributes in the first 1000 records
- In addition, the administrator can nominate one user that contains any additional attributes for which mappings are needed
- Specify desired mappings of LDAP attributes here.
- If the specified Group attribute contains an Agiloft group name, the user is added to that group.
- If the specified Team attribute contains an Agiloft team name, the user is added to that team.
- LDAP users are always assigned to a designated primary team.

The screenshot shows a web-based configuration interface for LDAP field mapping. At the top, there is a navigation bar with tabs: Servers, LDAP, Server, Login, Object Class Field, Field Mapping (selected), Group Mapping, Team Mapping, Options, and Synchron. Below the navigation bar are four buttons: Back, Next, Finish, and Cancel. The main content area is divided into three sections:

- LDAP Group attribute:** The label "Select the object class that will be mapped to Agiloft groups." is followed by a dropdown menu containing "objectClass".
- LDAP Team attribute:** The label "Select the object class that will be mapped to Agiloft teams." is followed by a dropdown menu containing "title".
- Primary Team:** The label "Select primary team for the LDAP users." is followed by two radio button options: "Admin Team" (which is selected) and "First value of LDAP Team attribute".

- Agiloft fields in the Employee table are populated by LDAP attributes according to these mappings. You will likely want to add some custom LDAP attributes to your user entries for this purpose. Common mappings:

LDAP	Agiloft
sAMAccountName	Login
givenName	First Name
sn (surname)	Last Name
name	Full Name
mail	Email

Please specify the mapping between each Agiloft attribute and a corresponding LDAP attribute.

Actions Taken	--none--
Annual Sales Quota	--none--
Assets	--none--
Cell Phone	--none--
Change Requests Submitted	--none--
Comments	--none--
Communications	--none--
Completed Tasks/Activities	--none--
Contact History	--none--
Date Created	--none--
Date Updated	--none--

- At the bottom of the "Field Mapping" tab, you can specify that you want to have Agiloft store certain user attributes instead of storing them in LDAP.

Some businesses prefer to use LDAP to store usernames and passwords, but store other user-related info in Agiloft. This setting controls where Agiloft will look to get user fields that are not covered by the above mapping.

Additional User Fields

None
 Obtain values from user of same name
 Get values of the user:

- Group mapping allows LDAP groups to be granted group privileges in Agiloft. The default group catches users who are not in an appropriate LDAP group. You can also choose not to update groups and teams from LDAP after the first synchronization or login of a user. When LDAP synchronization is run for the first time, user records will be created with the default groups and teams values you select in the next option. After that first import, you may edit the group membership in the user record of Agiloft at any time, and LDAP will never overwrite this value when the user logs in. This setting also affects the Primary Team field.

Agiloft Reference Manual

Group Mapping Options:

- Do not update groups from LDAP after first user creation/login. Use settings from user record instead
- Map LDAP attributes to EW groups and refresh group data from LDAP each time the user logs in

Choose default EW group:

None (refuse login) ▼

Group Mapping

- Show the first groups that are found
- Find groups that the first users belong to [Refresh Group List](#)

LDAP Group

Agiloft Group

- Similar to Group Mapping, this tab assigns LDAP groups to Agiloft Teams. The setup will likely be very similar.

Team Mapping Options:

- Do not update the Teams field from LDAP after first user creation/login. Use settings from user record instead
- Map an LDAP attribute to the Agiloft Teams field and refresh data from LDAP each time the user logs in

Choose default EW team:

None (refuse login) ▼

Team Mapping

LDAP Team

Agiloft Team

- Show the first teams that are found
- Find teams that the first users belong to [Refresh Team List](#)

- Here you choose when and how often you want to automatically synchronize with LDAP.


These two settings are used for manual sync.

Automatically import all LDAP users into system when ldap setup is complete
 Do not import users automatically

Never
 Every days
 Weekly on
 Monthly on day of the month

Start time

- If you chose "Do not import users automatically", or chose to automatically synchronize "Never", or simply want an update from LDAP before the scheduled time, you can at any time click "Synchronize LDAP/AD users". If you do not synchronize, LDAP users are only created in Agiloft if they log in.

Click [LDAP/AD Authentication](#) to configure Agiloft to additionally use your LDAP or Windows Active Directory server for user authentication. 

[LDAP/AD Authentication](#)

[Synchronize LDAP/AD users](#)

- Instead of or in addition to copying LDAP data to an Agiloft table, you can choose to query LDAP using an ActiveX control. An explicit Agiloft login is not required.

Setup → Access → Single Sign-On

General

Finish
Cancel

Select whether the system will attempt to automatically validate login/password combinations against the internal user table or an LDAP/Active Directory server.
 This facility uses an Active X control so IE 8.0+ must be used as the browser, the server must be included in the list of trusted sites and the user's Windows login name must be the same as their Agiloft login. If these conditions met, the user can instantly login using a URL such as
https://SERVER_NAME:/gui2/sso.jsp?autoLogin=true&project=KB_NAME&State=Main

Enable LDAP Single Sign-On

 Yes
 No

Select whether the system will use Google™ Single Sign-On (SSO) Service to authenticate users. If SSO is enabled, the user can instantly login by clicking on a hotlink such as
https://SERVER_NAME:/gui2/sso?project=KB_NAME&State=Main. If they are logged into Google as John.Doe@gmail.com and *Use full Google account name* is checked, the system will look for a user with login John.Doe@gmail.com. Otherwise the system will look for a user with login John.Doe. If a matching user is found, they will be logged in as that user.

Enable Google Single Sign-On (SSO)

 Yes
 No
 Use full Google account name as a KB login name (recommended)

Enter the trusted domain name, so that users coming from this domain can use single sign-on. This option is most useful if the system is within your firewall.

Windows Domain Name

 Domain name

Or enter a range of trusted IP-addresses, so that users coming from these addresses can use single sign-on. This option is very useful if you are accessing the system from across a firewall / NAT since from the perspective of the system all your users will appear to come from a single IP address and it can also be used if the system is within your firewall.

IP addresses range

 IP range
 From:
 To:

If single sign-on authentication is enabled it is possible to login from a user's desktop without entering a password. This may be considered a security risk for highly privileged users such as members of the admin group. This selection list allows you to exclude the members of such groups from using this facility.

Excluded Groups

adminimport

Anonymous

Contract Creator

Select this option if you would like the single sign-on utility to validate the password used to login against the password in the MS Active Directory database. Technical note: The validation is performed by comparing an encrypted hash of each password.

Authenticate users against Windows Domain Controller

Finish
Cancel

See Also:

[Groups](#)

[Teams](#)

USER ACCESS

Location of User Data

User information (including login, password and other data) has its source in one of two places:

- A sub-table of the Contact table in Agiloft – users are imported or manually created in the appropriate table
- In LDAP or Microsoft AD, which some companies use to store all their user information in one place so that they can have only set of data to maintain and one login/password that is used to authenticate against by disparate systems

Even if the user is authenticated through LDAP or ADS, Agiloft creates a user record for him that it is then able to use in rules and other parts of the system as if the user were a native user, so effectively all users are stored in the system

How do Users Access the System?

- There are several ways to login:
 - Through a custom login page with a login block that may sit on the corporate web site
 - Through an autologin hyperlink or button that contains a login/password and other parameters (KB name, Table, State, Search, Record ID and so on) and that may be encrypted and time limited if it is sent in an outbound email
 - Through a single sign-on methodology – the user is logged into a corporate intranet or web portal already, and he clicks a hyperlink that passes his user information into the system to authenticate him without his having to enter a login again
 - Through the system login page
- The system login is generally located at: <https://hostname/gui2/login.jsp> For certain servers <http://hostname/gui2/login.jsp> or <http://hostname:8080/gui2/login.jsp> may be used.
- Custom login blocks can be added to any web page using standard HTML like this:

```
<form method= "post"  
  
action="http://www.example.com/gu  
i2/login.jsp" >
```

```

        <input type= "hidden"
name= "KeyID"
                value="0" >
        <input type= "hidden"
name= "state"
                value= "Main"
>
        <input type= "hidden"
name= "project"
                value=
"KnowledgeBaseName" >
        <input type= "hidden"
name="exiturl"
value="http://www.example.com" >
        <input type= "hidden"
name= "loginurl"
value="http://www.example.com" >
        <table border= "0" width=
"90%" >
                <tr><td
align=left>Username:</td>
                        <td> </td>
                        <td>
                                <input
type="text" size= "30"
maxlength= "50"
name= "user"
value= "admin">
                                        </td>
                        </tr>
                <tr><td
align=left>Password:</td>
                        <td> </td>
                        <td>
                                <input
type="password" size= "30"
maxlength= "50"
name= "passwd" >
                                        </td>
                        </tr>
                <tr>
                        <td align=left>
</td>
                                <td> </td>
                                <td>
                                        <input
type= "submit"
value= "Login" >

```

```
        </td>
      </tr>
    </table>
  </form>
```

To result in:

Sending/Resetting Password

- To allow users to be sent a new password if they forget their password, first assign them to a group that permits this on the General tab of the group permissions.
- If you are using a custom login page, add a Send Password link or button to your login page that calls up another custom html page.

Reset Password Page

- The HTML Code needed on the reset password page is as follows. You can add additional instructions or information for the user and insert this into your own website look and feel scheme:

```
<form
action="http://[serverhostname]/g
ui2/resetPassword"
method=post>
  Login:<input type=text
name=login size=25
maxlength=50><br>
  or email address:<input
type=text name=email size=25
maxlength=50><br>
  <input type=hidden
name=project value=[KB Name]>
  <input type=submit
value=Go>
</form>
```

- Using this function does not mail the user his current password, as this could create a security problem. Instead, it changes the password to a random string of characters and emails that. The user must then login and re-set his password from inside the system.

EXIT AND LOGIN URL'S

When users log out they are taken to the ExitURL, which can be contained in the login block or hyperlink and/or defined as a global variable. By defining this global variable, you can ensure that users are never taken to the system login screen.

When users are logged out automatically for any reason (incorrect login or password, timeout, etc.) they are shown a link to log back in that contains a URL defined as the LoginURL

You will need to set up both of these URLs as global variables that point to your main login or support page, once you know where that will be

Setting up Global Variables

- Go to Setup/System/Manage Global Variables
- There should already be variables defined for ExitURL and LoginURL
- Edit each of these variables and type in the full URL to the page you want users to be taken to instead of the default page (which is a page at our website explaining how to do this)

Hyperlinks Included in Emails

Emails can include a hyperlink that logs a user back into the system. In order for the system to generate this hyperlink, it has to know it's own URL. If the OS/Webserver is configured in some non-standard way, this may be impossible for Agiloft to discover, resulting in incorrect hyperlinks.

There is a global variable that can be set to correct this situation. If it does not already exist, go to Setup/System/Manage Global Variables and create a new variable of type URL with the name: hotlinkServerRoot (please use this name exactly) and type in the full hostname used to access your Agiloft system, e.g. "http://server17.agiloft.com". The system will add the rest of the link e.g. /gui2/login.jsp automatically.

SINGLE SIGN-ON

You can allow users to login automatically by configuring single sign-on. If configured, user can simply click on a hotlink to be logged into the system using the login from their Windows session and verified against LDAP.

Setup → Access → Single Sign-On

General

Finish **Cancel**

Select whether the system will attempt to automatically validate login/password combinations against the internal user table or an LDAP/Active Directory server.
This facility uses an Active X control so IE 8.0+ must be used as the browser, the server must be included in the list of trusted sites and and the user's Windows login name must be the same as their Agiloft login. If these conditions met, the user can instantly login using a URL such as `https://SERVER_NAME:/gui2/sso.jsp?autoLogin=true&project=KB_NAME&State=Main`

Enable LDAP Single Sign-On
 Yes
 No

Select whether the system will use Google™ Single Sign-On (SSO) Service to authenticate users. If SSO is enabled, the user can instantly login by clicking on a hotlink such as `https://SERVER_NAME:/gui2/sso?project=KB_NAME&State=Main`. If they are logged into Google as John.Doe@gmail.com and "Use full Google account name" is checked, the system will look for a user with login John.Doe@gmail.com. Otherwise the system will look for a user with login John.Doe. If a matching user is found, they will be logged in as that user.

Enable Google Single Sign-On (SSO)
 Yes
 No
 Use full Google account name as a KB login name (recommended)

Enter the trusted domain name, so that users coming from this domain can use single sign-on. This option is most useful if the system is within your firewall.

Windows Domain Name
 Domain name

Or enter a range of trusted IP-addresses, so that users coming from these addresses can use single sign-on. This option is very useful if you are accessing the system from across a firewall / NAT since from the perspective of the system all your users will appear to come from a single IP address and it can also be used if the system is within your firewall.

IP addresses range
 IP range
 From:
 To:

If single sign-on authentication is enabled it is possible to login from a user's desktop without entering a password. This may be considered a security risk for highly priviledged users such as members of the admin group. This selection list allows you to exclude the members of such groups from using this facility.

Excluded Groups

Select this option if you would like the single sign-on utility to validate the password used to login against the password in the MS Active Directory database. Technical note: The validation is performed by comparing an encrypted hash of each password.

Authenticate users against Windows Domain Controller

Finish **Cancel**

This feature uses an Active X control so IE 5.0+ must be used as the browser, the server must be included in the list of trusted sites and and the user's Windows login name must be the same as their Agiloft login. If these conditions met, the user can instantly login using a URL of

http://SERVER:8080/gui2/sso.jsp?autoLogin=true&project=KB_NAME&State=Main

Enter the trusted domain name, so that users coming from this domain can use single sign-on. This option is most useful if the system is within your firewall.

Or enter a range of trusted IP-addresses, so that users coming from these addresses can use single sign-on. This option is very useful if you are accessing the system from across a firewall / NAT since from the perspective of the system all your users will appear to come from a single IP address and it can also be used if the system is within your firewall.

If single sign-on authentication is enabled it is possible to login from a user's desktop without entering a password. This may be considered a security risk for highly privileged users such as members of the admin group. This selection list allows you to exclude the members of such groups from using this feature.

HOTLINKS

Overview

Hotlinks allow actions to be automated. For example, a hotlink can be created that logs the user into Agiloft and opens a new Bug Report record and sets the Severity field to Medium, hides the GUI so that the users never even realize that they have entered Agiloft and returns them to a URL of your choosing when they click the Finish button.

Autologin Hotlinks

- You can create a hotlink that will log a guest user in to any screen, such as the New User form, the New Ticket form, the New Lead form, etc. without showing them any of the interface
- Staff users can create their own hotlinks to take them to different saved searches, different tables and so on

Hotlink Examples

Login to Demo in Progress KB to self-register and create a New Customer record without seeing the rest of the gui:

```
<a href="http://www.example.com:8080/gui2/login.jsp?keyID=0
&user=register&password=register
&project=Demo%20in%20Progress&state=New:contacts.customer&
Table=Contacts&GUI=No/eui">
http://www.example.com:8080/gui2/login.jsp?keyID=0&user=register
&password=register&project=Demo%20in%20Progress
&state=New:contacts.customer&Table=Contacts
&GUI=No/eui</a>
```

Login to Podaac KB as the FAQ user to the end user interface:

```
<a href="http://www.example.com:8080/gui2/login.jsp?KeyID=0
&project=podaac&user=faquser&password=faquser
&State=enduser">
http://www.example.com:8080/gui2/login.jsp?KeyID=0&
project=podaac&user=faquser&password=faquser
&State=enduser</a>
```

Login to Agiloft KB as myself to submit a new bug report:

```
<a href="http://www.example.com:8080/gui2/login.jsp?keyID=0
```



```
&user=bridget&password=notreal&project=EW  
&state=New:c_bugreport">  
http://www.example.com:8080/gui2/login.jsp?keyID=0  
&user=bridget&password=notreal&project=EW  
&state=New:c_bugreport</a>
```

Defining the Table in a Hotlink

The table value is found by going to Setup > Tables, choosing a table, and clicking Edit. On the first tab of the wizard after the labels, you'll see the logical table name to be used in a hotlink

*Table Name

sw_45165f_contact_table2

Logical Table Name

contacts

See Also:

[Hotlink API Reference](#)

Hotlink Integration

Introduction

This document explains the secure hotlinks facility by which Agiloft may be integrated with a third party portal.

Overview

Consider the case when a user has logged into the portal and now wants to perform some action in Agiloft such as creating a ticket, checking the status of their existing tickets or updating a ticket.

The user should not have to re-enter another login or password and if the account doesn't exist in Agiloft, it should be created automatically.

The possible invocation sequence may look like this:

1. The portal instructs Agiloft to create a user account if one does not already exist.
2. The portal instructs Agiloft to generate a hotlink for the user to create a ticket.
3. Agiloft generates and returns the hotlink.
4. The portal either displays the hotlink to the user to click or simply redirects the user's browser to that URL.
5. The standard New Ticket form opens and the user can fill in the data.
6. On clicking Finish the record is saved and the user is redirected back to the portal.

** The syntax by which the portal makes the requests in steps 1 and 2 is the standard hotlink syntax documented in the Administrator Guide / Access Methods / Hotlinks section of the user manual, with a few extra parameters such as the start and expiration date/time of the hotlink.*

** Using the expiration time parameter increases security since the hotlink after it has been disclosed to the user will stop working once it has expired. In the scenario when the user clicks on some link displayed in the portal and is immediately redirected to Create Ticket form the expiration time may be relatively short, e.g. 10 seconds from the current time.*

** The hotlink generated in step 3 is encrypted using a key unique to that knowledgebase so that the user's login and password are never exposed.*

** The steps 4,5 and 6 of the above mention redirecting to a New Ticket screen, but the Portal can also invoke the New Ticket screen in a popup browser window that could disappear when the user clicks Finish, or can log the user into the full Agiloft GUI.*

Example One

The example below uses plain HTML forms to illustrate how to perform the integration tasks. Please be aware that this is done **only** as a demonstration. The real integration should **never** use this method as it exposes the admin-level password to the users.

Request Agiloft to generate a hotlink to log a user in.

```

1  <html>
2  <body>
3  <form action="https://localhost/ewws/EWHotlinks" method="post">
4      <input type="hidden" name="$login" value="admin">
5      <input type="hidden" name="$password" value="1">
6      <input type="hidden" name="$lang" value="en">
7      <input type="hidden" name="$KB" value="Demo">
8      <input type="hidden" name="mode" value="debug">
9      <input type="text" name="template" size="200"
10         value="State=Main&user=guest&project=Demo
11             &expiration={to_number(now()+5minute)}
12             &start={to_number(now())}&exiturl={defaultExitURL()}">
13      <input type="submit" value="Go">
14  </form>
15 </body>
16 </html>

```

Line 3 contains the URL of hotlink generator entry point in Agiloft. To run this sample on a machine other than your default Agiloft server, replace "localhost" with the desired server's IP or host name. You may also need to specify the port if running on a port other than 80.

Lines 4-7 contain mandatory parameters for hotlink generation that should always be present. These are: the login/password for an admin-level user in Agiloft, your KnowledgeBase name, and the default language.

The parameter in line 8 is optional, and has two possible values: "debug" and "javascript". If left unspecified, the default is "debug". When "debug" is used, the hotlink generator returns the result as text/plain file in Javascript notation. In your real implementation you may find "javascript" more suitable as this returns a text/html snippet with a "script" tag that makes a client-redirect via Javascript to the URL generated. This snippet can be inserted directly into the stream being returned to the user's browser.

Lines 9-12 contains the hotlink template, which tells Agiloft what should be in the final hotlink to be generated and encrypted.

Template parameter

```
State=Main&user=guest&project=Demo&expiration={to_number(now()+5minute)}
```

```
&start={to_number(now())}&exiturl={defaultExitURL() }
```

This follows the general Agiloft hotlinks notation with some values being specified as formulas to be evaluated at the time of call.

Please note the absence of the user password. It is assumed that the call to generate the hotlink is coming from a trusted channel e.g. portal's backend and by the time the link will be displayed/disclosed to the user the portal has performed the authentication of the user.

Constant values

In the beginning of the template we can see simple constants known to the portal at the time of call:

State=Main - takes the user to the Main interface on login

user=guest - this is the name of the user profile we are using to log in to Agiloft. Can be derived from the portal user's login in the real integration scenario.

project=Demo - the name of the KnowledgeBase you're logging in to. Please note that this should match the *\$KB* parameter value above.

Formulas

The formulas (text, enclosed by `{}`) are used to define values to be calculated by Agiloft at the time of call.

To generate a hotlink, you need to convey the hotlink activation time via **start** parameter, and for security reasons, you should define the **expiration** time.

Using the function *now()* with a time offset of your choice is a preferred way to perform this. Additionally function *to_number* has to be applied to the results of *now()*.

The function now() calculates time with respect to the timezone configured for the Agiloft knowledgebase. In general this does not have to match the user or portal timezone.

In the example the hotlink is requested to be activated immediately (*start={to_number(now())}*) and to be valid for the next five minutes only (*expiration={to_number(now()+5minute)}*).

You can use the value of -1 for the expiration time if you don't want the hotlink to expire. Obviously, this is not recommended for security reasons.

The last parameter in the template is the **exiturl**. Two functions are provided by Agiloft - *defaultExitURL()* and *defaultCancelURL()*, that pull the global variable URLs

from the your installation of Agiloft. You can also hard-code any pre-set URL as a value there.

Encryption keys

Before testing your hotlinks, you should check if Agiloft contains valid hotlink encryption keys. Login as admin-level user, go to Access/Automatic login hotlinks. Check value "Keys generated at:", if they are not generated, press "Generate". Using this encryption key allows you to hide the variables which might be a security problem if sent in plain text.

The result of the call in "debug" mode would be similar to:

```
var redirectURL=
  'https://localhost/gui2/login.jsp?genhotlink=
  X7g7z+QhGmQAailWCwVumrj7gjF9jTD0MeyB1ZK/3DUZ5VYM8ejftln0r8x+zv6ZDoJR+PV7
  v33Bz4ehE4MkyuaOfaDZv0dZt38dJIJdzmX38dHwEz6WVg==&genproject=Demo '
```

The result of the call in "javascript" mode would be similar to:

```
<script>
  location.href=
  'https://support.enterprisewizard.com/gui2/login.jsp?genhotlink=
  X7g7z+QhGmQAailWCwVumrj7gjF9jTD0MeyB1ZK/3DUZ5VYM8ejftln0r8x+zv6ZDoJR+PV7
  v33Bz4ehE4MkyuaOfaDZv0dZt38dJIJdzmX38dHwEz6WVg==&genproject=Demo '
</script>
```

Example Two

Editing an existing record when the ID is known by the portal

*The example below uses plain HTML forms to illustrate how to perform the integration tasks. Please be aware that this is done **only** as a demonstration. The real integration should **never** use this method as it exposes the admin-level password to the users.*

This is a more advanced case, because it requires more data than just the login/password information from the portal side to generate the hotlink. For this case, the portal must pass to the generator:

- the ID of the record
- the Agiloft logical table name
- the subtable name
- .

The table and subtable names are explicitly defined since they cannot be changed by anyone but admin-level users.

One can look these names up in Agiloft (Setup/Tables/Edit/General tab) and store them in some configuration.

```
<html>
<body>
  <form action="http://192.168.1.4:8080/ewws/EWHotlinks" method="post">
    <input type="hidden" name="$login" value="admin">
    <input type="hidden" name="$password" value="qwerty">
    <input type="hidden" name="$lang" value="en">
    <input type="hidden" name="$KB" value="Demo">
    <input type="hidden" name="mode" value="javascript">
    <input type="hidden" name="tablename" value="case">
    <input type="hidden" name="subtablename" value="case">
    <input type="text" name="template" size="200"
      value="State=Edit:{$tablename}.{$subtablename}&record={$recordid}
      &user=guest&project=Demo&expiration={to_number(now()+5minute)}
      &start={to_number(now())}&exiturl={defaultExitURL()}
      &record_access=edit"><br>
    Record ID:<input type="text" name="recordid"><br>
    <input type="submit" value="Go">
  </form>
</body>
</html>
```

Note that the ID of the record, table and subtable names are passed as HTTP parameters, with corresponding formulas referencing them by name. These can also be hardcoded directly in the hotlink template instead.

The user "guest" used for demonstration usually doesn't have access to the Staff interface and no permissions to edit other user's records. Use a different user in your tests.

Example Three

Create a user in Agiloft

*The example below uses plain HTML forms to illustrate how to perform the integration tasks. Please be aware that this is done **only** as a demonstration. The real integration should **never** use this method as it exposes the admin-level password to the users.*

There is not much difference from the previous example:

```
<html>
```

```

<body>
  <form action="https://localhost/ewws/EWHotlinks" method="post">
    <input type="hidden" name="$login" value="admin">
    <input type="hidden" name="$password" value="qwerty">
    <input type="hidden" name="$lang" value="en">
    <input type="hidden" name="$KB" value="Demo">
    <input type="hidden" name="mode" value="javascript">
    <input type="hidden" name="tablename" value="contacts">
    <input type="hidden" name="subtablename" value="customer">
    <input type="text" name="template" size="200"
      value="State=New:{$tablename}.{$subtablename}&user=guest&project=Demo
        &expiration={to_number(now()+5minute)}&start={to_number(now())}
        &exiturl={defaultExitURL()}&gui=no"><br>
    <input type="submit" value="Go">
  </form>
</body>
</html>

```

The changes are limited to the table/subtable name, and some parts of the hotlink template (State=New instead of Edit, also added gui=no, to log the user out after the entry is created).

Example Four

Update the user information in Agiloft

*The example below uses plain HTML forms to illustrate how to perform the integration tasks. Please be aware that this is done **only** as a demonstration. The real integration should **never** use this method as it exposes the admin-level password to the users.*

This is actually a combination of the above:

```

<html>
<body>
  <form action="localhost/ewws/EWHotlinks" method="post">
    <input type="hidden" name="$login" value="admin">
    <input type="hidden" name="$password" value="qwerty">
    <input type="hidden" name="$lang" value="en">
    <input type="hidden" name="$KB" value="Demo">
    <input type="hidden" name="mode" value="javascript">
    <input type="hidden" name="tablename" value="contacts">
    <input type="hidden" name="subtablename" value="customer">
    <input type="text" name="template" size="200"
      value="State=Edit:{$tablename}.{$subtablename}&record={$recordid}
        &user=guest
        &project=Demo&expiration={to_number(now()+5minute)}
        &start={to_number(now())}
        &exiturl={defaultExitURL()}&record_access=edit"><br>
    Record ID:<input type="text" name="recordid"><br>
    <input type="submit" value="Go">
  </form>

```

```
</form>  
</body>  
</html>
```

To be able to edit the user information, the portal must know the user's ID. REST API may be used for that.

Please note that for EndUsers the permissions to edit records in the "Contacts" table are usually given for their own record only, so the ID of the record to update must match the login in "template" parameter (user=guest in example).

See Also:

[Hotlinks Overview](#)

[Hotlinks Parameters](#)

[REST API](#)

Hotlink API Reference

Introduction

In Agiloft it is possible to execute functions using hotlinks, such as login as user XXX, switch to the tab for table YYY and run a search on the records that meet criteria ZZZ.

For programmatic access, the product can also be driven using the Web Services API.

Examples

Live examples based on this specification can be found on your installation at: [URL] /gui2/hotlinks/test.jsp

For example, <http://server10.agiloft.com/gui2/hotlinks/test.jsp>

See Also:

[Syntax](#)

[Keywords](#)

[Examples](#)

Syntax

The hotlink consists of three main components.

\$URL_address_of_Agiloft?	For example, http://server10.agiloft.com/gui2/ login.jsp
KeyID=0	Strictly speaking, this is just another keyword/parameter pair, but it is always the same and is required in hotlinks
Keyword/Parameter pairs	See Keywords

See Also:

[Automatic Execution of Agiloft Functions](#)

[Keywords](#)

[Examples](#)

Keywords

Keywords are case insensitive, but their values (such as the table or field name) are case sensitive. To find the correct case for your table, go to Setup/Tables, select your table, click Edit, and look for the Logical Table Name. To find the correct case for a field, edit its parent table, and then edit the field. Remember to use the field name, and not the label.

Keyword	Possible values and example	Comments
kb, project, product	Any knowledgebase name	
	kb=Example	These keywords specify the name of the knowledgebase that the user is logging into. They all mean exactly the same thing and are interchangeable. There is more than one keyword with this meaning for historical reasons.
table	Any table name	This keyword specifies the name of the table that the user should view. In this example, it is as if the user logged on and then clicked the Cases tab.
	Table=Case	<p>The Table parameter is based on the table name rather than the label.</p> <p>If the user provides a table name that is not recognized, the system logs them into the default screen with an error message:</p> <p>"Table \$name not found. Please use the name of the table, rather than its label, for example Contact, rather than Contacts"</p> <p>Due to the current Agiloft GUI design, there is no difference between specifying a table or a subtable (i.e. Case.Bug takes the user to the same tab as Case)</p>
state	New	This keyword specifies what action should be performed automatically.
	New:\$table_name or New:\$table_name. \$subtable	"New:\$table_name" creates a new record of specified type, as in the example.
	<i>example:</i> State=New:contact.user	"Charts" takes the user to the Charts tab of the Staff interface in the table specified by Table parameter
	Charts	"Main" takes the user to the main mode of the interface and allows specification of a search that should be automatically executed
<i>example:</i> State=Charts		
Main	"FAQ" takes the user to the FAQ mode of the EndUser interface and allows to specify the <i>table</i> parameter to choose the table that contains FAQ records	
<i>example:</i> State=Main&search=MyAssigned		
FAQ	"FAQItem" takes the user to the specific FAQ record. <i>table</i> and <i>record</i> parameters are required	

Agiloft Reference Manual

	<i>example:</i> State=FAQ	"Convert:\$table_name" enables a specific record to be converted from a hotlink. The hotlink must include the name of a corresponding conversion action.
	FAQItem	
	<i>example:</i> State=FAQItem	This functionality offers a way to use template tickets to create new tickets with prepopulated fields. For instance, you could provide a user with a hyperlink to "create a password reset ticket" that converts a ticket that already has the category, assigned to, and other fields filled out for a password reset request.
	Convert	
	<i>example:</i> State=Convert:case	This parameter works not only in external login hotlinks, but also in the internal hotlinks available in the hotlink widget in the staff home page. So you can provide multiple hotlinks that each convert a different template ticket directly on the home page, such as: Create a Network Outage Ticket, Request a New Employee Setup, Request a Password Reset or Create a new server upgrade task
record	any valid record id	This parameter is mandatory for state=Edit. It identifies the record to be opened for editing.
expiration	expiration=05/05/2005	This keyword defines expiration date for anonymous links. Use it together with state=Edit
field	field= summary:test_summary	This keyword allows you to set field values for a record in state=Edit. A hotlink may contain several "field" parameters. Use the following format to modify fields: field=<\$field_name> :<\$value>
exiturl	Any url <i>example:</i> exiturl=http://www.sw.com	When the user logs out from Agiloft, take him to the specified URL
loginurl	Any url	If the user is unable to login to Agiloft because he supplied an invalid username/password combination, take him to this URL. If no loginurl is provided, but an exiturl is provided, the exiturl functions as the loginurl. If the user's login to Agiloft times out due to inactivity, take him to this URL.
timeouturl	Any url	If no timeouturl is provided but an loginurl is provided, the loginurl functions as the timeouturl If the user is manually logged out of Agiloft, take him to this URL
logouturl	Any url	If no logouturl is provided, but a timeouturl is provided, the timeouturl functions as the logouturl. If the user clicks the Cancel button when GUI is set to No, he is returned to this URL
cancelurl	Any url	If no cancelurl is provided, he is returned to the exiturl.
euiurl	Any url <i>example:</i>	When the user logs into Fully Custom Interface via Single-Sign On this url specifies where to redirect the

gui	euiurl=eui2template/x.php	user.
	No, no	"no" or "No" mean that only the new/edit record screen, together with any pre and post HTML is shown. The rest of the Agiloft interface is not be shown. When the user clicks the Finish button to create the new record (or Cancels), he is automatically returned to the exiturl
	EndUser, enduser	
	Staff, enduser	
	EndUser/No, Staff/No	"no/enduser" or "no/staff" are just like no, but explicitly specify whether the record should be shown using the end user layout or the staff layout.
	LowRes	
	<i>example: gui=no</i>	"staff" is the default and means that the full staff GUI is shown.
	no/enduser, no/staff	"LowRes" together with "state=Main" allows one to log into low resolution interface for mobile devices.
	<i>example: gui=no/enduser</i>	
	enduser	
<i>example: gui=enduser</i>		
staff		
<i>example: gui=staff</i>		
field=hide_btn:true	Prevents display of the Finish&New button	This keyword is used in conjunction with GUI=No when you want the user to provide a single record, such as an employment application
user	Any valid user login name	The keyword parameter provides the user login name
passwd, password	Any valid password for the specified user	"passwd or password" values provide the provide the password part of a user/password combination.
	<i>example: passwd=bob</i>	
search	Any valid search name	This keyword parameter provides the name of a saved search that is automatically executed when the user logs on.
customsearch		This parameter is always used together with the table parameter
	Any valid search	This keyword allows the user to specify the field names and values within the hotlink itself. The search uses the standard syntax ==, >=, >, <=, <, !=, (,), && and
record_access		However, many of these symbols have reserved meanings in HTML so they have to be expressed using % N equivalents. For example CustomSearch=State%3E=Closed%26%26Severity==Critical
	View, view	Multiple conditions can be joined together using or && symbols. It is possible to specify whether the data should be sorted in ascending or descending order by using sortasc=\$fieldname or sortdesc=\$fieldname
	Edit, edit	This keyword is only used together with a search or a customsearch that returns just one record and Edit state. It causes the specified record to be opened for

		viewing or editing.
	none	<p>Value "none" for this parameter indicates no GUI should be shown during hotlink's processing. Such behavior useful if a hotlink intended to change some data in Agiloft without editing screen. For example,</p> <pre>state=Edit:Cases& record=2& fields=priority:high& record_access=none</pre> <p>will set priority to "high" and redirect user to logout URL (login.jsp, if the URL is undefined)</p> <p>This keyword can be used together with gui=no to return a single record to the user without showing the rest of Agiloft.</p> <p>If the user specifies a chart/report to view, it shows it in the same fashion as if the user had clicked on the hotlink</p>
locale	de	This keyword is used to define the language in which the interface should be shown to the user.
	en	Accepted values correspond to supported languages:
	es	German - de
	fr	English - en
	ru	Spanish - es
	zh	French - fr
	pt_BR	Russian - ru
	SW_User_Selected_Lang	Chinese - zh
		Brazilian Portuguese - pt_BR
		For example:
	<pre>&locale=fr</pre> <p>A special value SW_User_Selected_Lang allows to log in with the same language as used for last user's session. This value will be used from now on in the email hotlinks as well.</p> <p>If no value is supplied English is assumed for backward compatibility.</p>	

Examples

Additional examples are provided at [your URL]/gui2/hotlinks/test.jsp

For example: <http://support.agiloft.com:8080/gui2/hotlinks/test.jsp>

Action	Hotlink
Go to the end user interface.	
When the user selects Exit, go to the URL: http://test.agiloft.com/gui2/hotlinks/test_exit_url.html	http://test.agiloft.com/gui2/login.jsp?keyID=0&kb=demo&user=admin&passwd=1&State=enduser &exiturl= http://test.agiloft.com/gui2/hotlinks/test_exit_url.html &loginurl=http://test.agiloft.com/gui2/hotlinks/test_login_url.html
If there is an error (for example the password is invalid) go to the URL: http://test.agiloft.com/gui2/hotlinks/test_login_url.html	
Go to the staff (Main) interface and select the Cases table	http://test.agiloft.com/gui2/login.jsp?keyID=0&kb=demo&user=admin&passwd=1&State=Main&table=cases
Try to go to a non-existent table (deliberate error)	http://test.agiloft.com/gui2/login.jsp?keyID=0&kb=demo&user=admin&passwd=1&State=Main&table=unexisting_contacts
Go to the contacts table	http://test.agiloft.com/gui2/login.jsp?keyID=0&kb=demo&user=admin&passwd=1&State=Main&table=contacts
Go to "contracts" table and perform the Saved Search named "company name"	http://test.agiloft.com/gui2/login.jsp?keyID=0&kb=demo&user=admin&passwd=1&State=Main&table=contracts_table&search=company%20name
Go to "contacts" table and search for the "test" keyword	http://test.agiloft.com/gui2/login.jsp?keyID=0&kb=demo&user=admin&passwd=1&State=Main&table=contacts &customsearch=test
Submit a new bug without showing the Agiloft GUI, logoff on finish	http://test.agiloft.com/gui2/login.jsp?keyID=0&kb=demo&user=admin&passwd=1&State=New:Cases.Bug_Report&table=cases&gui=no
Submit a new bug and show the GUI	http://test.agiloft.com/gui2/login.jsp?keyID=0&kb=demo&user=admin&passwd=1&State=New:Cases.Bug_Report&table=cases
Go to Reports for the contact table and show the GUI	http://test.agiloft.com/gui2/login.jsp?keyID=0&kb=demo&user=admin&passwd=1&State=Charts&table=contacts
Go to Reports for the contact table without showing the GUI	http://test.agiloft.com/gui2/login.jsp?keyID=0&kb=demo&user=admin&passwd=1&State=Charts&table=contacts&gui=no

Agiloft Reference Manual

- Submit a new bug without showing the GUI, set the "description" and "solution" fields http://test.agiloft.com/gui2/login.jsp?keyID=0&kb=demo&user=admin&passwd=1&State=New:Cases.Bug_Report&table=cases&gui=no &field= description:preset%20from%20hotlink &field= solution:--%20enter%20some%20--
- Submit a new bug without showing the GUI, set the " description "and" solution" fields and immediately exit to the specified URL without giving the user the opportunity to make any changes to the record http://test.agiloft.com/gui2/login.jsp?keyID=0&kb=demo&user=admin&passwd=1&State=FinishNew:Cases.Bug_Report&table=cases&gui=no &table=cases&gui=no &field= description:preset%20from%20hotlink &field= solution:--%20enter%20some%20--+&_mode=new &_unifier=50&type=30 &exiturl=http://test.agiloft.com/gui2/hotlinks/test_exit_url.html
- Open a bug (ID=2) in edit mode http://test.agiloft.com/gui2/login.jsp?keyID=0&kb=demo&user=admin&passwd=1&State=Edit:Cases.Bug_Report&record=2&record_access=edit
- Open a bug (ID=2) in view mode http://test.agiloft.com/gui2/login.jsp?keyID=0&kb=demo&user=admin&passwd=1&State=Edit:Cases.Bug_Report&record=2&record_access=view
- Open a bug (ID=2), set priority=low and exit http://test.agiloft.com/gui2/login.jsp?keyID=0&kb=demo&user=admin&passwd=1&State=Edit:Cases.Bug_Report&record=2&record_access=none&field=priority: low
- Open a bug (ID=3) in edit mode and change the summary and priority fields http://test.agiloft.com/gui2/login.jsp?keyID=0&kb=demo&user=admin&passwd=1&State=Edit:Cases.Bug_Report&record=3&record_access=edit &field=summary: from link Tue Mar 15 15:59:09 PST 2009 &field=priority: medium
- Open a bug (ID=1) in edit mode using a temporary password that expires on 10/25/09 http://test.agiloft.com/gui2/login.jsp?keyID=0&kb=demo&user=admin&record_access=edit &State=Edit:Cases.Bug_Report&record=1&expiration=10/25/09 &passwd=@@sKbNeWUkB5OEXEKRbsa9Ag% 3D% 3D

LOOK AND FEEL

The Look&Feel suite of wizards allow the customization of virtually every aspect of the End User and Staff user interface. Immediate visual feedback of the effect is provided for most customizations. Several different Look&Feel Schemes may be created and assigned to different teams The Look&Feel seen by a particular user is based on their Primary team, thus allowing custom branding for valued users.

The customizations include variation in font (face, size, style, color), mouse-over effects, scrollbars, borders, graphics, headers/footers and placement preferences.

The following table provides a guide to which wizard controls each type of customization on each element of the interface. These wizards are accessed from Setup/Look&Feel for the current scheme or by editing a different Scheme via Setup/Look&Feel/Manage End-user Schemes or Setup/Look&Feel/Manage Staff Schemes.

Customization	Appropriate Look and Feel Wizard
Required fields in user records and wizards <ul style="list-style-type: none"> • Location of* (Required field indicator) • Additional text inserts • Color preferences 	Edit Field Display
Select the base scheme - a new look and feel is created by editing an existing one and saving it under a new name.	Edit Current Scheme/Global
Tabs <ul style="list-style-type: none"> • Fonts, colors, border, etc • Mouseover, select effect • Order 	Edit Current Scheme/Toolbar Edit Tab Order
Page Layout <ul style="list-style-type: none"> • Text font, style, colors, background, etc • Scroll bars 	Edit Current Scheme/Body

Agiloft Reference Manual

- Selection and mouseover effects on links
- Titles

Table records

- Text font style colors, etc
- Odd/Even row background
- Table headers
- Hyperlinks (moused over and selected)

Data input or view record form

- Format for values, labels, buttons, help
- Alternate row background colors
- Button format

Search dialog

Website customization

- Header/footer
- Graphics

Title in the Browser

Scheme assignment to teams

Left Pane Customization (Staff I/F only)

Edit Current Scheme/Tables

Edit Current Scheme/Forms

Edit Current Scheme/Forms

Edit Current Scheme/Header/Footer

Go to Setup/System/Manage Global Variables.

Create a new global variable of type short text constant called `staff_title` and use whatever text you like as the value. Create another global variable of type short text constant called `enduser_title` and use your preferred text there. You will need to logout and login again to see your changes.

Edit Current Scheme/Apply

Edit Current Scheme/Left Pane

Icon shortcuts (Staff I/F only)

- Help, wrench, calendar, etc.

Edit Current Scheme/Icons

See Also:

[Teams](#)

[Staff Interface](#)

[End User Interface](#)

Left Hand Pane Setup

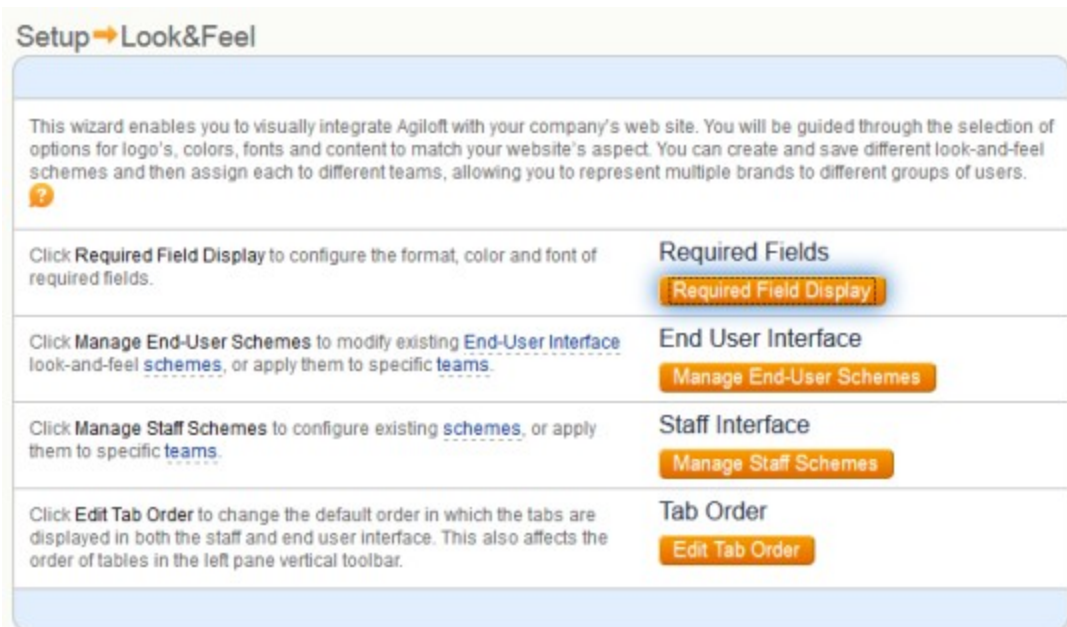
[Flash Tutorials](#)



LOOK AND FEEL WIZARDS

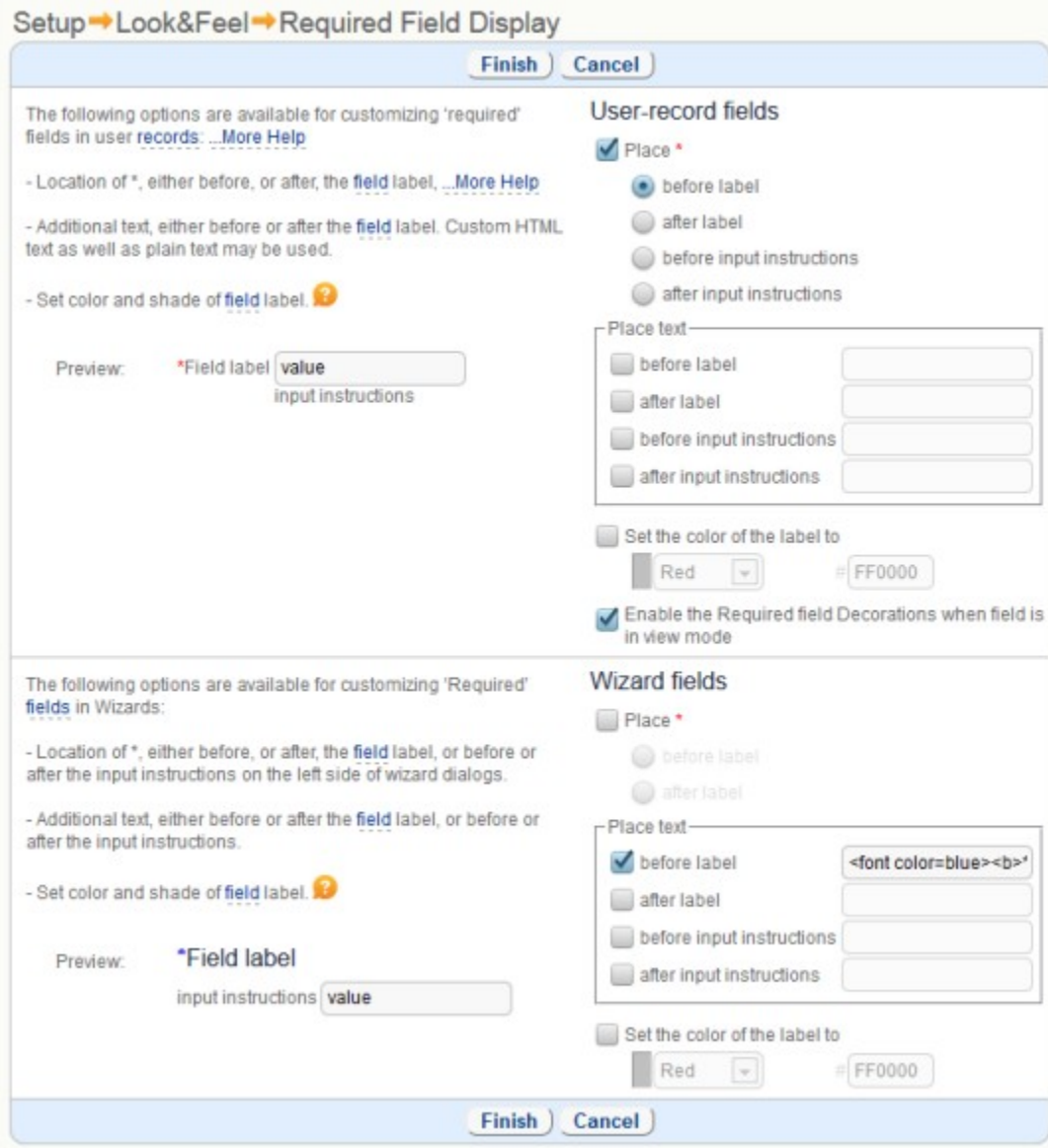
Setup/Look and Feel provides access to a wide range of options for changing the look and feel. From this tab you can add or remove header/footer HTML and company logos, control text font, size, and color, and change the appearance of buttons, tabs, and other user interface elements.

The Look and Feel tab provides two different Wizards with options tailored to their user group access permissions. The Edit End-User Schemes button, for instance, will not allow configuration of elements that are not visible to End-User groups. The Staff Schemes button allows configuration of the appearance of these extra internal-only options. On this tab you can add or remove header/footer HTML and company logos, control text font, size, and color, and change the appearance of buttons, tabs, and other user interface elements.



Field Display: Editing Required Fields markers

The display of required fields is controlled by Setup/Look and Feel, and clicking the Required Fields Display button. This button brings up the following wizard:



Customizing the Look and Feel

The look and feel of both the Staff interface and the End-User interface is governed by the user's Primary team

The Setup/Look & Feel page allows you to give each of your major customers, or even internal departments a separate look and feel

Click [Manage End-User Schemes](#) to modify existing [End-User Interface](#) look-and-feel [schemes](#), or apply them to specific [teams](#).

End User Interface

[Manage End-User Schemes](#)

Click [Manage Staff Schemes](#) to configure existing [schemes](#), or apply them to specific [teams](#).

Staff Interface

[Manage Staff Schemes](#)

Click New to create a Look and Feel, or click the edit icon to bring up a wizard to edit an existing definition.

Setup → Look&Feel → Manage Staff Schemes

Status: 10 record(s) found Interface: Staff

New | Delete | Set As Default

	View	Edit	Scheme name	Default for teams	Default for new teams
<input type="radio"/>			Light Blue with Orange accents	No Teams applied	<input type="checkbox"/>
<input type="radio"/>			Mellow Green	No Teams applied	<input type="checkbox"/>
<input type="radio"/>			Blue and Gold	No Teams applied	<input type="checkbox"/>
<input type="radio"/>			Blue Red Yellow	No Teams applied	<input type="checkbox"/>
<input type="radio"/>			Pastels	No Teams applied	<input type="checkbox"/>
<input type="radio"/>			Light blue	No Teams applied	<input type="checkbox"/>

Global Tab

From the Global tab you can name or rename your scheme, change the logo, and define your primary, secondary, and third default font faces. Once defined, your Primary Font will be available from the drop down font menu for all other font display selections. Selecting "Primary font" allows you to quickly change the font across many parts of your look by changing only the value of "Primary" in the Global tab.

These options defines the default fonts for the whole interface. If the Primary font is not available, the Second or Third will be used. A red star warns that the specified font is not available in the current browser.

A new L&F is not created from a blank slate, but by copy/editing the settings of an existing record. This define which record to start from and can also be used to replace all the settings in an existing L&F.

Please select a common Base Scheme before making further customizations.

Customize the font used by the base scheme.

Important: Changing the base style font could affect multiple text styles that inherit from the base style.

[...More Help](#)

Select Base Scheme

Light Orange

Choose the Base Fonts

Primary

Second

Third

The color used by each component of the GUI can be controlled individually on the other tabs of this wizard, or more typically, can be set to Color 1, ... Color 20.

Changing these colors may therefore change the color of many components of the GUI.

Color 1	<input type="text" value="User defined"/>	# 003575
Color 2	<input type="text" value="User defined"/>	# 003B88
Color 3	<input type="text" value="User defined"/>	# E1EEFD
Color 4	<input type="text" value="User defined"/>	# F8F8F8
Color 5	<input type="text" value="User defined"/>	# FBF8F8
Color 6	<input type="text" value="User defined"/>	# FFFBE8
Color 7	<input type="text" value="User defined"/>	# D7E8FD
Color 8	<input type="text" value="User defined"/>	# BDBDBC
Color 9	<input type="text" value="User defined"/>	# F7A000
Color 10	<input type="text" value="User defined"/>	# E66800

Preset the colors that you are going to use across your L&F here

You can change the default logo by uploading a new logo file, or you can disable the logo at the top of the page to save available space.

Change Logo

Use default Agiloft logo

Do not show a logo

Use custom logo

The icon visible at address bar

Shortcut Icon

[Select / manage...](#)

This option controls whether the main toolbar and tables are shown as a vertical toolbar on the left or a horizontal toolbar across the top

Select Layout

Horizontal Toolbar

Vertical Toolbar

Choose the design of the layout

ToolBar Tab

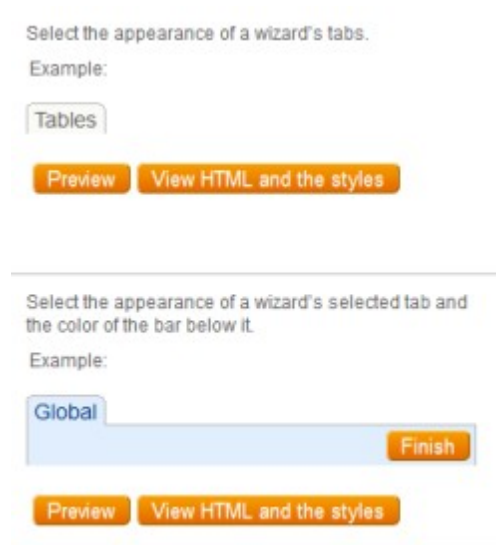
This page helps you set the appearance of the Toolbar elements, including tabs, menus, font faces and sizes, colors, and borders.

The Toolbar refers to the row of tabs visible in the Staff interface that appear at the top of any administration Set-Up Wizards. For instance, some of these tabs may be named "Fields", "Layouts", or "Rules".

This tab also controls the appearance of drop down menus in Tables.



The examples in the left hand column show what any new settings change will look like. The Preview button displays the changes in full context of the Agiloft look and feel scheme.



The "Other" or "Left Pane" tab

The third tab in the Look and Feel Wizard is different depending on whether you are editing End-user or Staff-user schemes and whether you are using Horizontal or Vertical Toolbar. For End Users, this tab is called "Other" and includes the Calendar and Go button settings. For Staff users, this tab is called "Left Pane" and additionally controls the Look and Feel of staff-only tools, such as the chat tool pane. Vertical toolbar, apart from standard Calendar, Chat etc, also contains Tables, Setup and other tools. Here, for example, you can configure the way Title Bar looks, including settings like gradients and rounded corners

Select the display of the tool titles, for example, Search, Calendar, or Snapshot

Example:

Setup

[Preview](#) [View HTML and the styles](#)

Tool Title

Border Bottom-Left Corner Radius: 0 px

Border Top-Right Corner Radius: 0 px

Border Top-Left Corner Radius: 0 px

Border Bottom-Right Corner Radius: 0 px

Font Face: Use Primary Font

Font Size: 12 px

Bold Underline

Font Color: Black #000000

Background Color: White #FFFFFF

Image:

- None
- Picture from file
- Gradient
 - From Top (Start Color) to Bottom (End Color)
 - From Left (Start Color) to Right (End Color)

Start Color: Color 4 #

End Color: Color 15 #

Padding Top: 3 px

Padding Bottom: 3 px

Also, from here you can control the appearance of the calendar in the left pane, as shown below.



You can also change the appearance of the Go button in the top pane.



Body Tab

The following controls help you customize various aspects of the Staff Interface layout, including background, fonts, scrollbars, hyperlinks, and status messages. It also controls the appearance of hyperlinks, titles and status messages.

The screenshot displays the Agiloft Support Home Page. The top navigation bar includes the Agiloft logo, a search bar, and user information (Welcome Christopher, English, Demo Master Copy, Logout, Help). The left sidebar contains navigation options: Home, My Assigned, Last Opened, Tables (Contacts, Companies, Support Cases, Helpdesk Cases, Change Requests, Assets, Software Apps, Repair Requests, Repair Items, Tasks, Contracts), Calendar, and Messaging.

The main content area is titled "Support Home Page" and "My Assigned Support Cases". It features a "Support Cases" section with a search bar (Search: -TEXT-, contains) and a table of cases:

Edit	ID	Summary
	424	Customer's keyboard not functioning correctly
	413	Hard Drive not accessible - BSOD
	412	Printed pages have streaks
	403	How do we login to the mobile interface?
	402	How do I request a license for Agiloft?

Below the table is a "Useful Links" section with various actions like "New Helpdesk Case", "View My Assigned Helpdesk Cases", etc. To the right is a "My Assigned Support Cases by Priority" section showing a funnel chart with three segments: Low (4), Medium (1), and High (1). A legend below the chart identifies the colors: High (red), Medium (yellow), and Low (green). The chart is generated on Apr 17 2013 07:29:07 and includes a Refresh button.

Table tab

This controls the appearance of tables:

Support Cases
 Status: 16 record(s) found, 1 pages [Click for details...](#)

New Mass Edit Actions

Views Search

Search: contains

Show All Refine String Context Active Only More Options

<input type="checkbox"/>	Edit View	ID ↓	Summary	Status	Priority	Assigned Person
<input type="checkbox"/>		424	Customer's keyboard not functioning correctly	Assigned	Low	Christopher Caldwell
<input type="checkbox"/>		415	problem with equipment	Closed	Low	Christopher Caldwell
<input type="checkbox"/>		414	Repair Ticket	Closed	High	
<input type="checkbox"/>		413	Hard Drive not accessible - BSOD	In Repair	Low	Christopher Caldwell
<input type="checkbox"/>		412	Printed pages have streaks	Assigned	Medium	Jack Repairguy
<input type="checkbox"/>		406	SQL Server query	Open	Critical	Ralph Knowles
<input type="checkbox"/>		405	Calculation with fields from the same record	Closed	Low	Ralph Knowles

Forms tab

This tab controls the appearance of the input forms.

Support Case

Save Cancel Details Working Related Records Contract Time Emails History

ID: 424 Status: Assigned

Assigned Team: Support Team Assigned Person: Christopher Caldwell

Contact Information

*Customer Name: John Suzuki User Company: IBM

Customer Phone: 333-555-1212 Ext. 122 Cell Phone: 333-124-4233

*Customer Email: example@example.com Customer Login: customer

Supported User: Yes

Problem Description

Type of Issue: Equipment Problem Priority: Low

*Summary: Customer's keyboard not functioning correctly

Problem Description: Customer's keyboard has function keys that do not work.

The Forms tab also controls the display of error messages, and the table tree:



Printing Tab

This tab controls the display of elements selected for printing. These elements include the background color used in viewing a record for printing, the label displays, and the field displays.

Details Related Info

Print Close

Support Case

Common Area

ID	415	Status	Closed	Assign to Me
Assigned Team	Support Team	Assigned Person	Christopher Caldwell	

Contact Information

*Customer Name	John Suzuki	User Company	IBM
Customer Phone	333-555-1212 122	Cell Phone	333-124-4233
*Customer Email	example@example.com	Customer Login	customer
Supported User	Yes		

Problem Description

Type of Issue	Equipment Problem	Priority	Low
*Summary	problem with equipment		
Problem Description	My server is not working		

Inbound Attachments

Investigation Notes and Time

Additional Notes

Time Spent

Time Description


Staff-Only Notes


Solution


Downloadable Files	I Would Like To Reopen My Ticket	No
Standard Solution	Published	No

Icons

Icons - This tab controls the icons, such as:

Edits the selected object or record. 

Opens the View Record page. 

Invokes the Calendar from within an input form. 

Content tabs

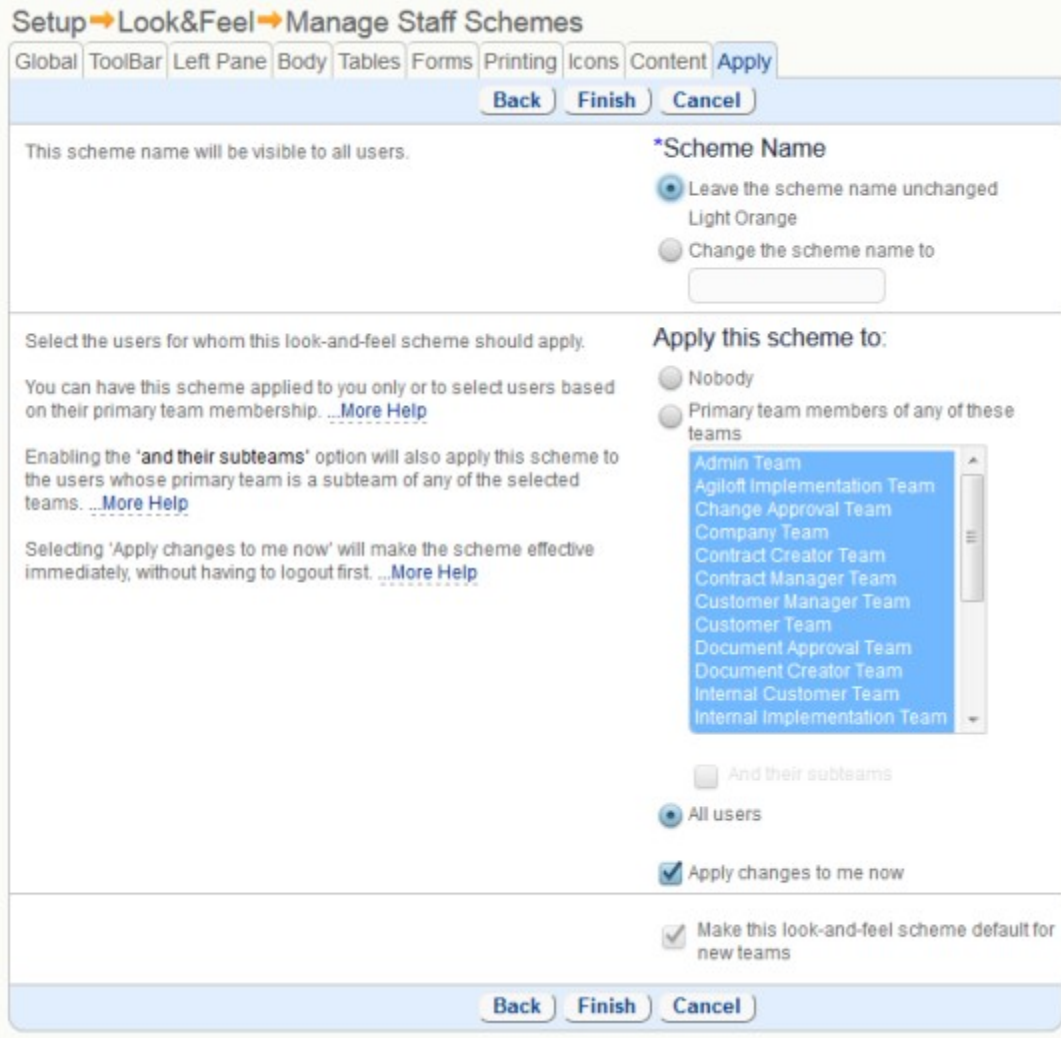
This tab lets you add Header & Footer HTML to embed the interface in a page of your website. For End-User Schemes this tab also controls the logo displayed in the interface.



Apply tab

This tab controls which teams the look and feel is applied to.

By defining a different look and feel for separate teams, you can allow each of your major customers to see their own logo in the end user interface, or provide visually impaired staff with a larger font:



Gradients

It is possible to specify either horizontal or vertical gradients for the main GUI elements such as:

Action bar (ie New, Edit, Mass Edit)

Left Pane tool tabs

Search block

Edit record background

Top table header row (ie ID, Summary...)

Gradients are specified as:

From Top (Start Color) to Bottom (End Color)

From Left (Start Color) to Right (End Color)

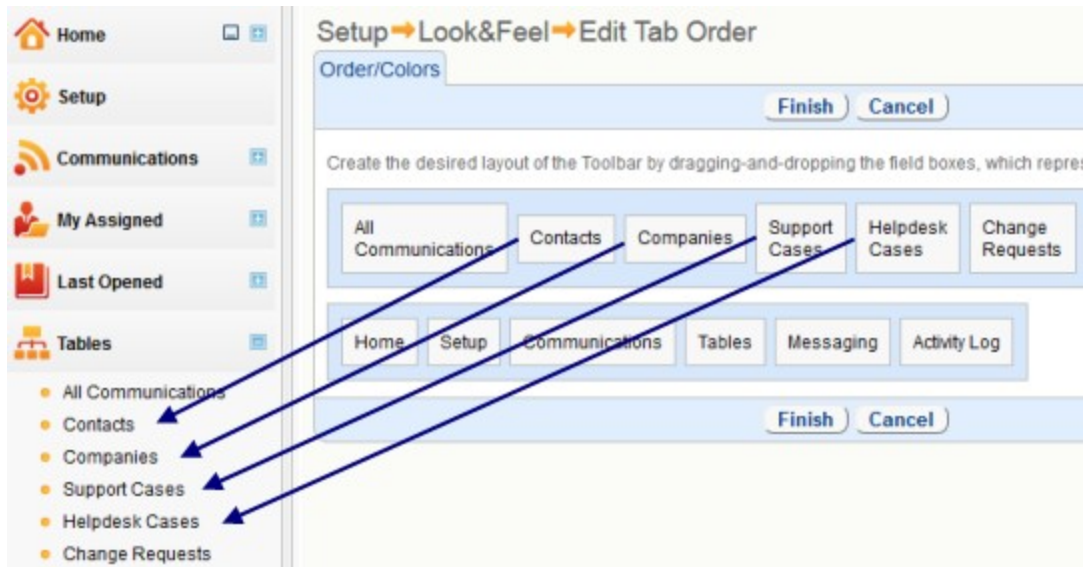
Start and End colors may be customized.

Edit Tab Order

This allows you to change the order of the table tabs at the top of the staff interface.



The result of changing the tab order applies to all users, and will be seen the next time you login or click on a tab.



If some tables have been removed from the toolbar (Tables/Edit Table/Permissions/Hide table from all users), then you will see them in the Edit Tab Order dialog, but not among the toolbar tabs.

END USER INTERFACE

Setting up End User Interface typically involves four steps:

- Set up My Items.
- Set up FAQs.
- Set up entry points to End User Interface.
- Set up the End User Interface appearance.

MY ITEMS SETUP

Configure the "My Items" search by going to Setup/End-User Interface/, and clicking the "Customize" button under "Setup My Items".

This gives you a menu of Tables with default "My Item" searches already defined. Click the Edit icon to change the default My Items search or view for any of these tables. Create a new default search by clicking "New", selecting a table, and selecting a saved search from the list of those saved to the table. You can also specify which view to use by default when viewing this table under "My Items".

Customize the content and field placement for records and their input forms via Setup/Tables, select the Table to edit, go to the "Layout" tab, and toggle to the End User Interface view.

SEARCH FAQ SETUP

Access the Search FAQ setup page by going to Setup/End User Interface, and clicking the "Setup FAQs" button.

Setup → End-User Interface → Setup FAQs

<input type="radio"/>		Table Name	Subtables included
<input type="radio"/>		Helpdesk Case	Yes
<input type="radio"/>		Document	Yes
<input type="radio"/>		Support Case	Yes

This page shows what custom FAQ groups are already defined. A custom FAQ group may have saved searches, specific sorting methods, subtopics, and field mechanisms.

- **Edit an existing FAQ table** by clicking the Edit icon
- **Create a New FAQ table** by clicking "New" in the toolbar

Tables Tab

You can define one FAQ collection for each different top level table, and select which collections are assigned to each group. Top level FAQ collections may include subtables, but subtables can not stand alone as a separate FAQ table.

Which records are considered FAQs is defined by a saved search. By default in the support tables there is a field called Published that is changed to Yes to create an FAQ and these records are found by the Published search

Display Tab

You can configure the FAQ page to display up to four topic selection drop-downs so users can easily narrow their search to certain subjects before entering keywords. You may select from all choice fields in the current table, and they will be displayed in the order selected.

Search results are initially displayed using the summary view selected here. Users can view record details by clicking on a field in the summary view. Select an existing summary view from the list or create a new one by clicking New.

Search Drop-down Fields:

1. Type of Issue
2. None
3. None
4. None

Show records using this view:

FAQ View [New](#)

Select which fields are displayed when the user opens an FAQ entry.

FAQ pages will show all of the selected fields to all viewers, overriding the viewer's group permissions.

Viewable Fields

Select All

Clear All

- | | |
|---|--|
| <input type="checkbox"/> Account Number | <input type="checkbox"/> Additional Notes |
| <input type="checkbox"/> All Cases for This Customer | <input type="checkbox"/> All Time Entries |
| <input type="checkbox"/> Assign to Me | <input type="checkbox"/> Assigned Person |
| <input type="checkbox"/> Assigned Team | <input type="checkbox"/> Assigned Team Leader |
| <input type="checkbox"/> Cell Phone | <input type="checkbox"/> Change Category |
| <input type="checkbox"/> Change Summary | <input type="checkbox"/> Closed By |
| <input type="checkbox"/> Communications | <input type="checkbox"/> Contract is For |
| <input type="checkbox"/> Conv ID | <input type="checkbox"/> Convert to Change Request |
| <input type="checkbox"/> Convert to Repair Request | <input type="checkbox"/> CR Conv ID |
| <input type="checkbox"/> CR Conversion Related To | <input type="checkbox"/> CR ID |
| <input type="checkbox"/> CR Spawned from SC ID | <input type="checkbox"/> CR Status |
| <input type="checkbox"/> Created By | <input type="checkbox"/> Creator Login |
| <input type="checkbox"/> Creator Team | <input type="checkbox"/> Customer Email |
| <input type="checkbox"/> Customer Login | <input type="checkbox"/> Customer Name |
| <input type="checkbox"/> Customer Phone | <input type="checkbox"/> Customer Team |
| <input type="checkbox"/> Date Assigned | <input type="checkbox"/> Date Closed |
| <input type="checkbox"/> Date Created | <input type="checkbox"/> Date Updated |
| <input type="checkbox"/> Deletable | <input type="checkbox"/> Demo Data |
| <input checked="" type="checkbox"/> Downloadable Files | <input type="checkbox"/> FAQ Topics |
| <input type="checkbox"/> I Would Like To Reopen My Ticket | <input checked="" type="checkbox"/> ID |
| <input type="checkbox"/> Inbound Attachments | <input type="checkbox"/> Latest Contract End Date |
| <input type="checkbox"/> Latest Contract Status | <input type="checkbox"/> Latest Contract Title |
| <input type="checkbox"/> Linked Company | <input type="checkbox"/> Priority |
| <input checked="" type="checkbox"/> Problem Description | <input type="checkbox"/> Published |
| <input type="checkbox"/> Rating Posts | <input type="checkbox"/> Repair Request Status |

Select the default search. This is executed when the Search FAQ tab is first displayed. You may want to display the most recently published FAQs, the highest rated, or the most often accessed using this search. The default search is applied as a filter over any other FAQ filters defined for the user.

For instance, if a user has access only to FAQs related to Table A and B, and the default search shows the Most Recent, the Search FAQs tab will initially display the most recently published FAQs for Tables A and B, but will not display recently published FAQs for any other tables.

Select the default sorting method to use when displaying FAQs. Most Useful shows those with the highest user ratings, Best Match shows those with the best text match to the user's search criteria, Newest will sort by Publication Date, and Most Viewed sorts by the number of hits.

Select the field that should be used for sorting the FAQ's in the 'Newest' order.

Create and save a search to define how long newly-published FAQs are marked with the "New" icon. You may wish to adjust this search later depending on how frequently new FAQs are published.

Don't display records until the user runs a search

Display the results of this search:

Published

Sort by:

Best Match

Natural Language

Make the following saved searches available:

Created by me
 Closed Cases
 Closed in past month
 Closed Updated by Customer
 Created in past month
 Date closed is in last month
 Demo Records
 Enhancement Requests

Use this field for Newest sort order:

Date Created

In reverse chronological order

None

All Cases, ordered by Priority

Access Tab

Permissions determine which FAQs a user can access: If a user is given access to View all FAQs, he will see all records found by the search that defines FAQs. If he is

given access to FAQs but filtered to some other search, he will only see those records that meet both searches' criteria.

Once a user can access an FAQ, field permissions are not applied. Admin-defined FAQ field permissions override individual group field permissions in the FAQ view.

Set group permissions for access to FAQ's

admin:	<input checked="" type="radio"/> All <input type="radio"/> None	<input type="radio"/>	Standard Solutions	<input type="button" value="New"/>
adminimport:	<input checked="" type="radio"/> All <input type="radio"/> None	<input type="radio"/>	Standard Solutions	<input type="button" value="New"/>
Anonymous:	<input checked="" type="radio"/> All <input type="radio"/> None	<input type="radio"/>	Standard Solutions	<input type="button" value="New"/>
Contract Creator:	<input type="radio"/> All <input checked="" type="radio"/> None	<input type="radio"/>	Standard Solutions	<input type="button" value="New"/>
Contract Manager:	<input type="radio"/> All <input checked="" type="radio"/> None	<input type="radio"/>	Standard Solutions	<input type="button" value="New"/>
Customer:	<input checked="" type="radio"/> All <input type="radio"/> None	<input type="radio"/>	Standard Solutions	<input type="button" value="New"/>
Customer Manager:	<input checked="" type="radio"/> All <input type="radio"/> None	<input type="radio"/>	Standard Solutions	<input type="button" value="New"/>
Document Creator:	<input type="radio"/> All <input checked="" type="radio"/> None	<input type="radio"/>	Standard Solutions	<input type="button" value="New"/>
Document Manager:	<input type="radio"/> All <input checked="" type="radio"/> None	<input type="radio"/>	Standard Solutions	<input type="button" value="New"/>
Guest:	<input type="radio"/> All <input checked="" type="radio"/> None	<input type="radio"/>	Standard Solutions	<input type="button" value="New"/>
Internal Customer:	<input type="radio"/> All <input checked="" type="radio"/> None	<input type="radio"/>	Standard Solutions	<input type="button" value="New"/>
Marketing:	<input checked="" type="radio"/> All <input type="radio"/> None	<input type="radio"/>	Standard Solutions	<input type="button" value="New"/>

Edit Group Permissions:

admin

Select whether a rating scale is displayed in the FAQ view, to help estimate how useful the FAQ information is. The rating mechanism is used for the Most Useful sorting, which puts the highest rated FAQs at the top. You can configure the scale, steps and the wording of each step.

Rating of FAQ's

- Don't display a rating scale
- 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
- 25%, 50%, 75%, 100%
- Not useful, Somewhat useful, Very Useful, Answered my question

Layout Tab

To select the fields displayed when the user views FAQ, click Setup/End-User Interface/Setup FAQ/Layout.



END USER INTERFACE APPEARANCE SETUP

The appearance of the End User Interface can be fully customized to match the appearance of your website.

To define a look-and-feel scheme, click Setup/Look&Feel/Manage End-User Schemes

To customize the text/graphics that appear above and below the End User interface, click Setup/Look&Feel/Manage End-User Schemes/Edit/Content Page Header.

The layout of the header in Agiloft can be customized.

You can use the built-in HTML Editor to type or copy/paste the text into the text area to the right.

You can also use your favorite HTML editor, save the file locally, browse and upload a file directly from your file system.

The HTML will be inserted at the top of the screen.

The HTML may reference images that you plan to use as ``. Note that the images will be uploaded later.

At any point if you want to revert to the state before the wizard was opened use Restore Default button.

HTML text of Header

```
<center>
<LINK id=css href="selectedSheet"
type=text/css rel=stylesheet>

</center>
```

Use HTML editor Restore Default

Browse... Upload

Select / Manage Images...

The layout of the footer in Agiloft can be customized.

You can use the built-in HTML Editor to type or copy/paste the text into the text area to the right.

You can also use your favorite HTML editor, save the file locally, browse and upload a file directly from your file system.

The HTML will be inserted at the bottom of the screen.

The HTML may reference images that you plan to use as ``. Note that the images will be uploaded later.

At any point if you want to revert to the state before the wizard was opened use Restore Default button.

HTML text of Bottom

```
<center>
<LINK id=css href="selectedSheet"
type=text/css rel=stylesheet>

</center>
```

Use HTML editor Restore Default

Browse... Upload

Select / Manage Images...

See Also:

[Look and Feel](#)

[End User Interface](#)

[End User Interface Access and Permissions](#)

END USER INTERFACE ACCESS AND PERMISSIONS

A Guest user is only allowed to browse FAQs, but more privileges can be extended to registered end users.

Registered users usually belong to a 'Customer' group. There may be more than one customer group (e.g. Product A customers, Product B customers, European customers, etc) depending upon your specific business needs. The scope of the user's access within the End User Interface is controlled by the permissions granted to their group. Registered users must also be assigned to a team so they can access table views and EUI schemes defined for their team.

Access to "My Items"

The Toolbar displays only the tabs that the user has permission to view based on their group membership.

For example, the user may be authorized to view Case and Asset tables, but may create new records only in Case. In this scenario, the user will see New Case on the Toolbar, and "My Cases" and "My Assets" in the My Items menu. Table and record access to My Items is controlled via Setup/Access/Manage Groups/select Group to edit (See Permissions for further info).

The user's view of the table may be filtered by a saved search defined when group permissions are applied (e.g. only Cases where Company is the same as the user's company). An additional level of filtering, via a selectable predefined saved search, or a new saved search may be applied via Setup/End-User Interface/Setup My Items/Table.

Access to "Search FAQ"

The End User Interface allows users to find answers to their questions by browsing collections of frequently asked questions (FAQ) that have been published for them. The "Search FAQ" tab provides access to stored questions and answers on a particular subject. An administrator can define one or several FAQ collections and publish them to specific user groups.

Access to Search FAQ tab is based on group membership and can be configured from several places. In the Setup FAQ wizard (Setup/End User Interface/Setup FAQ/), Select FAQ to edit, then click the "Access" tab. This process is best if you're setting up a new FAQ collection for multiple user groups to view.

Finer group-level control is accessed from the Edit Group wizard (Setup/Access/Manage Groups/); select the Group to edit, select the Table containing FAQs, click the Permissions tab, and locate the checkbox "View FAQs for \$tablename" near the bottom right. If you have several tables containing FAQs, you will need to repeat this process for each, selecting the appropriate table in the second tab and proceeding from there.

Groups can be granted access to one or several FAQ collections. Users may access items in an FAQ collection based on the combination of the following criteria:

- Group permissions that define table level access (available at Setup/Access/Manage Groups/select Group to edit/Table),
- Saved searches that define record level access for groups (available at Setup/Access/Manage Groups/select Group to edit/Permissions),
- Default saved searches defined for each table that holds FAQ records (available at Setup/End-User Interface/Setup FAQ/TBD).

Important: FAQ record permissions override any established field-level group permissions. The user will see all fields defined in a FAQ layout, regardless of the field level permissions granted to their group.

Access to "My Profile"

The My Profile tab allows users to edit their user record. This tab is only shown to users who belong to a group with permission to edit their own Contacts. Only the fields the user is permitted to view or edit are displayed.

See Also:

[Groups](#)

[End User Interface](#)

[End User Interface Setup](#)

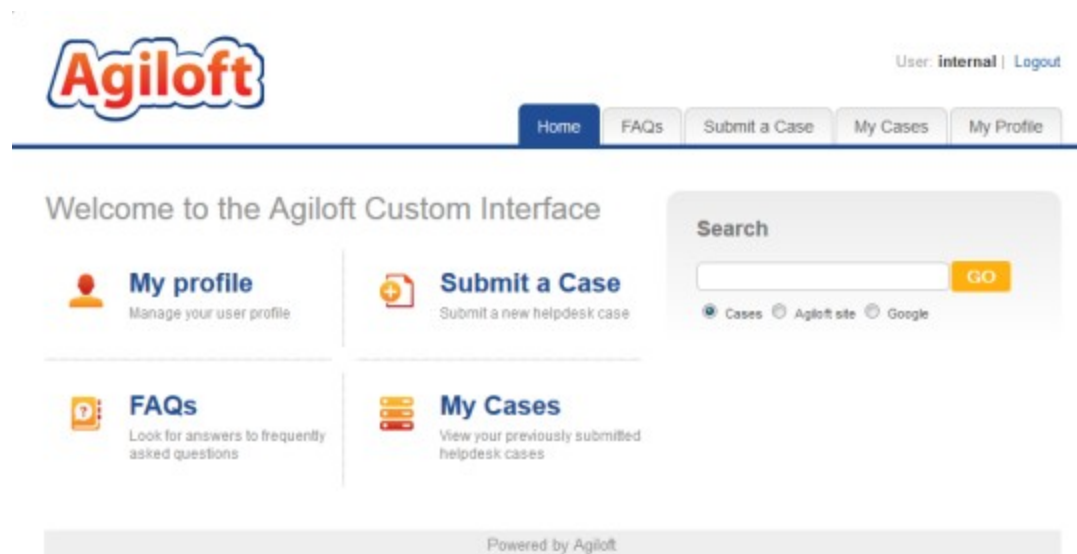
END USER INTERFACE

Overview of End User Interface

Members of end user groups access the system through the End User Interface. From there they can be permitted to:

- Search, view and print FAQs
- Submit new items in any table
- Edit items defined as their own in any table
- View items in any table and run and save simple and complex searches
- Edit their user profile

The tabs and menu items are dynamic based on the user's group permissions.



What Controls What?

- Which tabs the user sees depends on:
 - Group permissions table settings – View toolbar tab (must be on for New Record tabs), View FAQs on or off, Create Own Record on or off and View Records and View/Edit own contact record on or off
- Any saved searches the user sees in the FAQ screen are defined in the FAQ setup
- In the My Items screens she sees the Saved Searches that have been made both accessible and active for her group

- The default search run for the My Items screens are set up in the End User Interface setup screens
- Which tables have FAQs and how they appear is set in the End User Interface setup FAQ options

What are FAQs?

- FAQs are regular records of any table that someone decides should be available to end users
- The records can be created specifically as FAQs (or imported from some other system) or they can be records/questions received in the normal course of business that have answers that might be useful to others
- FAQ records can hold attached files that you want to make available to end users (software documentation, patches, HR documentation, Corporate Policy documentation, etc.)
- The records and attached files are fully searchable and the fields can be nicely formatted with HTML graphics etc.

To Setup FAQ's

- Go to Setup > End User Interface > Setup FAQs, then edit an existing entry or create a new one.
- Which records are considered FAQs is defined by a saved search. By default in the support tables there is a field called Published that is changed to Yes to create an FAQ and these records are found by the Published search

FAQs and Permissions

- Permissions are used to determine which FAQs a user can access: if a user is given access to View all FAQs, he will see all records found by the search that defines FAQs. If he is given access to FAQs but filtered to some other search, he will only see those records that meet both searches' criteria.
- Once a user can access an FAQ, field permissions are not applied – she sees the fields the admin selected to be displayed in the FAQ setup screen.

Setting up the My Items Search

- Go to Setup > Look and Feel > Setup My Items and click Customize
- Any table that you are making accessible to end users should have a search defined to be run when the user clicks the My Items tab and a view to use when displaying the results
- If you don't already have an appropriate search or view, create one from this screen

- Note that you can restrict the search to the user's own items by default, but may have additional saved searches he can select that show him his company's items, closed items, open items, etc. All search results are filtered by the user's permissions

Changing the Saved Searches that Appear in My Items

The Show drop-down contains saved searches that have been made accessible and active for the user's group

Agiloft User: Internal | Logout

Home FAQs Submit a Case My Cases My Profile

Saved Searches: Show all records Text: GO

Show all records
Mine
My Closed Helpdesk Cases
My Open Helpdesk Cases
Status is Not Closed

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

ID	Summary	Type of Problem	
61	New Employee	Service Request	New Employee Setup
43	My browser is hanging in our intranet	Internet Connection	Browser Problem
42	I can't connect to the company internet	Internet Connection	Can't connect
31	My laptop mouse is no longer working.	Laptop	Mouse Failure

Edit Search Accessibility

- Make a list of searches you want to add or eliminate.
- Click the main tab of the Table whose searches you need to change.
- Click Saved Search > Manage to view all saved searches.
- Edit a search, click Apply tab
- Note that end users can create their own saved searches that will also appear here – if you find a search in the EUI drop-down that isn't applied to end users, it means the end user created it and is seeing it because he is the creator (you can always see your own searches).

Adjust Permissions

- When testing a system setup, if you see tables you shouldn't have access to, or are missing tables you should be able to see, go to the group permissions for the user's group and check the table permissions.
- The New [Table] tabs will only be seen if the group has the toolbar turned on and has "create own permissions" for the table.
- Toolbar on or off permissions are now relevant to the EUI – the New [Table] tabs will only be seen if the group has the toolbar turned on and has created its own permissions for the table

Configuring Chat for End Users

- Agiloft contains a support chat interface that can be turned on or off. When on, it is typically displayed as a tab in the End User Interface. When the user requests a chat, a notification pops up to any staff users on the team assigned to chat requests asking them to accept the request.

Turn on Chat Interface

- Go to Setup > Chat and edit the default session. The important options to configure are:
- Select teams to be notified when a request is received
- Enable groups in both the Show Session and Allow Access drop-downs if you want them to see and use the chat tab. If no groups are enabled here, the chat tab is effectively disabled.
- Decide if a popup suggesting chat when a user logs in is desired

See Also:

[End User Interface Access and Permissions](#)

[End User Interface Setup](#)

[Staff Interface Overview](#)

MOBILE / ADA INTERFACE

The mobile interface is designed to give customers maximum accessibility to their data without relying on email or syncing. Using mobile devices such as iPhone, BlackBerry or Android, customers can quickly navigate to records of interest with a clean web based interface. The interface can also be used by customers that require an ADA compliant interface.

Most actions are accessible using hotkey and do not require the use of a mouse. Users may create their own preferences for how the mobile interface is displayed.

Features of the interface include:

- Access to all available tables and saved searches as well as lists of recently visited items.
- A table view of records with possibility of navigation (first / prev / next / last page)
- Table search, including saved searches and manual search (by field values) with possibility of refining the search as can be done in the Staff Interface
- Performing additional actions on a whole table or selected set of table records, such as mass delete, mass edit, sending mail or linking records
- Usual editing functions such as: create new record, edit existing record, remove existing record
- Read only view of single record

MOBILE INTERFACE ACCESS AND SETUP

Access to the mobile interface is made by adding a new parameter to the end of the standard login string: &GUI=LowRes. It can also be coded into an HTML form by adding the name=GUI and value=LowRes.

The Mobile Interface configuration wizard can be found in the Home/Preferences section on the tab labeled ADA/Mobile. Here the user can set:

- Maximum number of saved searches shown in Recent Searches window
- Maximum number of saved searches shown in Search box window
- Maximum amount of test to show in Record without paging
- Maximum width of an input on ticket editing
- Show empty fields in record view mode

The last section defines tables, saved searches and charts that should be shown in the Home window as favorites.

With the proper permissions, Staff users can apply the settings to selected teams.

A new default view named "Mobile view" is used for displaying a table's records in the Table view window. It is created automatically and by default includes just the "Created By" field. The view cannot be deleted, but it can be edited.

FULLY CUSTOM INTERFACE

There is now a way to provide your different groups of end users with their own completely distinct interface for Agiloft. The default End User Interface already supports distinct color schemes, home page text, logos, permission-based access to tabs, and so on. But with the new End User Portal, you can completely reconstruct the screen layouts, provide your own style of toolbar and different options on the main page, your own fonts, color scheme, logo, and buttons.

Each group can have its own interface, with a different home page/starting point and different navigational options. Once these interfaces have been created you simply supply each group with the URL to their particular interface.

New installations of the default KnowledgeBase include the following example of a default portal:



The cost of this flexibility is that you must effectively build your own webpages that will hold the Agiloft screens, and this customization is done using a programming language, but this is kept as easy as possible. The administrator creates pages in HTML, DHTML, Javascript etc. using his preferred editor, and inserts macros where Agiloft (using Apache's Velocity template engine) should insert dynamic elements like tables, status messages, and records to open for editing.

The following pages contain a tutorial aimed at quickly creating a portal from scratch. It introduces important concepts and some of the key macros that you'll need. It assumes familiarity with the Agiloft application and user interface. Some basic HTML knowledge is useful, but not absolutely necessary.

The portal you'll create in the tutorial is for illustrative purposes only. The HTML is very basic and designed for simplicity rather than representing 'state of the art' web

authoring techniques! It does work though, and should give you the practice you need to carry on with confidence.

TUTORIAL

Pre-requisites

You have access to a knowledgebase with the following setup:

- A user who is a member of the admin group

Contact Information			
First Name	EW	Last Name	Admin
Title		Department	Customer Support
Email		Direct Phone	
Cell Phone		Email Pager	
Company	Agiloft		

User Information			
*Login	admin	*Password	*****
Primary Team	Admin Team	Note: Please include the Primary Team value in the Teams file	
		Teams	Admin Team, Company Team, Internal Implementation Team, Support Team

Groups	admin
--------	-------

The user should be a member of the admin group

- A table with logical name "contacts", with a "Default Contact View" view

The internal [KnowledgeBase](#) name of the table. Staff and end users do not see this name and it cannot easily be changed once it has been set. [...More Help](#)

*Table Name
sw_34546f_contact_table2

Logical Table Name
contacts

The Logical Table Name is "contacts"

This is the Default View for the Contacts Table

Contacts
 Status: 35 record(s) found, 2 pages [Click for details...](#)
 Page: [Page 1] [Next](#) [Last](#) Go to page [Go](#)

New Mass Edit Delete Actions Views Search

Search: Refine String Co

<input type="checkbox"/> Edit	ID ↓	Full Name	Contact Table	Co
<input type="checkbox"/>	413	John Doe	Contact	Ag
<input type="checkbox"/>	412	Admin	Employee	
<input type="checkbox"/>	411	Guest End User	End User	
<input type="checkbox"/>	410	Jane Repairgal	Employee	
<input type="checkbox"/>	409	Jack Repairguy	Employee	Ag
<input type="checkbox"/>	407	Repair Customer	End User	IBI
<input type="checkbox"/>	406	Don Wilson	End User	St
<input type="checkbox"/>	405	Jerry Brown	Contact	State of California 222-333-4444
<input type="checkbox"/>	404	Sam Weston	Contact	Glaxo Smith Kline 351-222-3333

Views dropdown menu items:
 New ...
 Edit ...
 Manage ...
 Personal view
 • Default Contact View
 Demo Data
 Employees in other tables
 End user view
 External users in other Tables
 Lookup for Contact by Company
 Mobile view
 name company phone email lookup
 name login group teams
 name team phone email
 Summary view
 Help

- And a table with a logical name of "case". For example, Support Cases:

The Logical Table Name for Suppose Cases is "case"

The internal [KnowledgeBase](#) name of the table. Staff and end users do not see this name and it cannot easily be changed once it has been set. [...More Help](#)

*Table Name
sw_24508f_case

Logical Table Name
case

This option controls whether it is possible to delete this table. This option is used to protect tables that are critical to your business processes. [...More Help](#)

Deletable

Yes
 No

Select special search to use when subtype filtered in top tabs

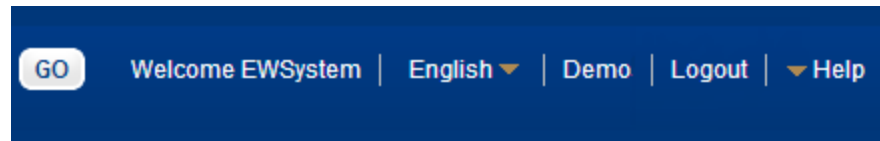
Use predefined filter
 Use custom search

[Create Search](#)

From this point forward we'll refer to your Agiloft server host as {host}, and your Knowledge base as {kbName}. For example - if the url in your browser after logging in looks something like

```
'http://ew.server.com/gui2/admin/main.jsp;jsessionid=5E..''
```

and the right top pane looks something like the attached screen shot:



then the {host} is

```
'http://ew.server.com/gui2'
```

and the {kbName} is 'Demo.'

CREATING A SIMPLE WEB PAGE

First we're going to create a user to act as the end user who'll be accessing the portal pages we'll create:

- Login as admin and add a new user with a login of 'portalUser' and a password of 'test01'
- Set the user to be a member of a group with permission to view ALL of the entries in the contacts table (e.g. the 'Support Staff' group).

Next we'll create a really simple home page for the new user to access the portal: Go to the EUI Templates tab and create a new record.

If you are using vertical Look&Feel, EUI Templates can be found under Tables in the Left Menu Pane.

EUI Templates
 Status: 40 record(s) found, 1 pages.
 Page: [Page 1] [Next](#) [Last](#) Go to page

New Mass Edit Delete Actions

Search:

<input type="checkbox"/>	Edit	ID ↓	Name	Description
<input type="checkbox"/>	<input type="button" value="Edit"/>	50	contact-info.htm	Custom contact page
<input type="checkbox"/>	<input type="button" value="Edit"/>	49	demopage.html	Custom home page
<input type="checkbox"/>	<input type="button" value="Edit"/>	48	meta.htm	Links to global Javasc
<input type="checkbox"/>	<input type="button" value="Edit"/>	46	style-ie.css	CSS for Internet Explo
<input type="checkbox"/>	<input type="button" value="Edit"/>	45	search-form.htm	Search block
<input type="checkbox"/>	<input type="button" value="Edit"/>	42	repair-search.htm	Search form for the s
<input type="checkbox"/>	<input type="button" value="Edit"/>	41	my_repairs.htm	shows my support ca
<input type="checkbox"/>	<input type="button" value="Edit"/>	40	newrepair.htm	Submit a new support
<input type="checkbox"/>	<input type="button" value="Edit"/>	39	newuser.htm	Submit a new user
<input type="checkbox"/>	<input type="button" value="Edit"/>	38	translation_zh.properties	中文资源包 - note: w only take effect if the
<input type="checkbox"/>	<input type="button" value="Edit"/>	37	translation_ru.properties	localization file for Ru changes will only take
<input type="checkbox"/>	<input type="button" value="Edit"/>	36	translation.properties	localization file for Eng changes will only take
<input type="checkbox"/>	<input type="button" value="Edit"/>	35	my_documents.htm	My Documents
<input type="checkbox"/>	<input type="button" value="Edit"/>	34	my_documents-search.htm	new documents sear

If for some reason, you still do not see the EUI Templates tab, then permission access to it has been turned off. You can turn it on by going to Setup/Access/Manage Groups, edit the admin group, and on the Table tab, select EUI Templates in the table tree, then scroll down and check the boxes to enable access to the table and to enable the toolbar tab for the table.

Then create a New EUI Template record with the following fields:

- In the 'Name' field type 'testContacts.htm'
- In the 'Description' field type 'Test contacts page'
- In the Body field, type in (or copy and paste) the following:


```
<html>
<head><title>Test portal</title></head>
<body>
<h1>Test portal</h1>
<div align="left">This is my new portal</div>
<p/>
</body>
</html>
```

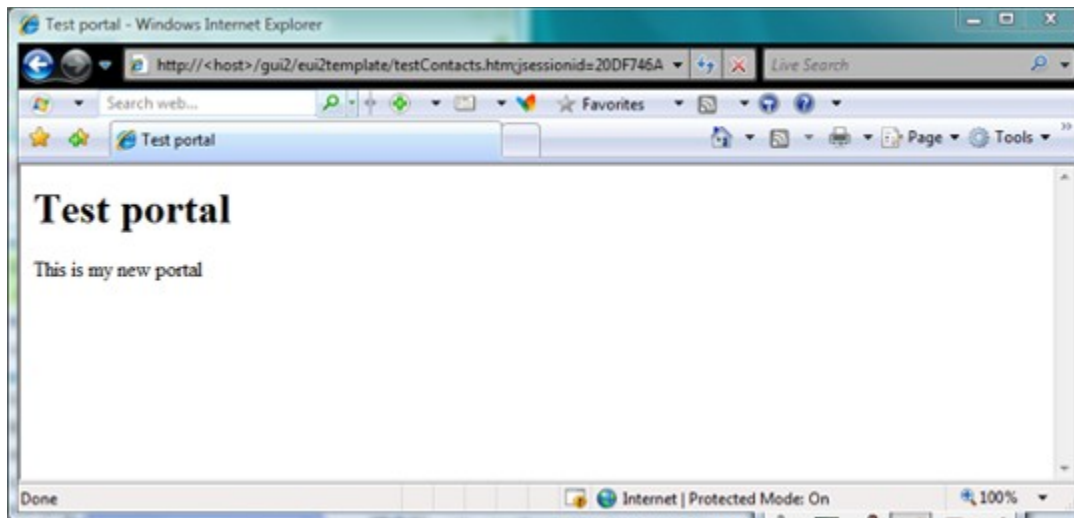
- Click 'Finish' to save the changes when you're done.

Next we'll test what we've done:

- Open a new browser window and type in the following url:

```
{host}/eui2template/testContacts.htm?kb={kbName}&user=portalUser&passwd=test01
```

You should see the following:



Congratulations! Even though this doesn't look like much, you've got the starting point we need to create a portal for end users!

MAKING IT A BIT MORE USEFUL

What we're going to do next is add a view of the contacts table to this page.

- Login as the admin user and go to the EUI Templates tab.
- Edit the 'testContacts.htm' template record that you created earlier.

```
<html>
<head><title>Test portal</title></head>
<body>
<h1>Test portal</h1>
<div align="left">This is my new portal</div>
<p/>

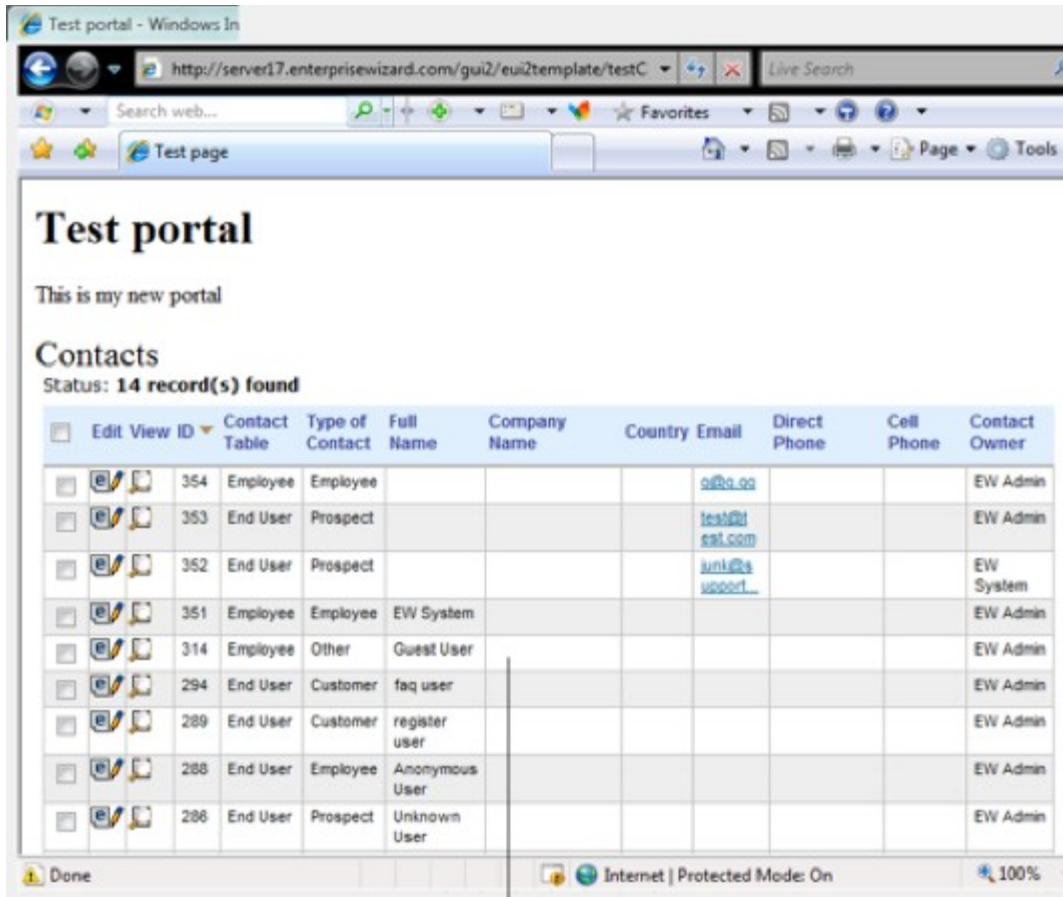
<div align="left"><font size="+2">Contacts</font></div>
#ew_table ("contacts" "Default Contact View" "" "" ""
"contacts_frame" "")

</body>
</html>
```

- Change the contents of the 'Body' field, adding a new div and the #ew_table macro so that it now looks like the following (changes highlighted in blue), and click finish
- Now let's test the page again: Open a new browser window and type in the following url:

```
{host}/gui2/eui2template/testContacts.htm?kb={kbName}&
user=portalUser&passwd=test01
```

This time you should see something that looks like the following:



Contacts Table records are included now

Try out the table; you can view and edit records - in fact it is fully functioning just as it is in Agiloft, as far as your user's group permissions allow.

HOW IT WORKS

Before going any further it is worth taking a step back and understanding how all this works. It will make what follows easier to learn.

What's in a URL?

A URL is an address. It gives all the information needed to locate something. In this case your web browser needs to locate the page we're constructing.

Take a look at the URL we used:

```
http://{host}/gui2/eui2template/testContacts.htm?  
kb={kbName}&user=portalUser&passwd=test01
```

The first part - `http://{host}/gui2/eui2template/testContacts.htm` - identifies the physical server machine on which the Agiloft software is installed, and specifies the name of the page template to retrieve from the EUI Templates table.

The last part - `?kb={kbName}&user=portalUser&passwd=test01` - tells the Agiloft software the knowledgebase that the page template belongs to, and gives the login credentials which identify an authorized user and validate that the page request comes from that user.

Please note, that when creating your EUI2 page you are NOT required to hardcode your user name and password right there in the template, but in the end all EUI2 requests should either have sessionid (see below) or pair user name/password. Normally, you would have a login block on your web site with user name and password fields, so when Login button is pressed a form action is created with credentials specified by your user. This form action would be EUI2 URL with user and passwd attributes being filled dynamically.

The only time that you would typically embed the user name/password is if you wanted to provide access to the FAQ interface, without requiring the user to login. In this case generic guest login is used (i.e. `eui2/eui2`) and it gives an access to the records that are usually public anyway.

You may see `;jsessionid=5E.....` in place of these details. This is because once the credentials have been passed to the Agiloft software and they have been verified as correct, a user session is created on the server. Once this session has been created your web browser only needs to refer to it by id when requesting a page from Agiloft. This allows the software to retrieve your session which contains all the details it needs.

Bear in mind this session is only valid for a certain length of time after you stop using Agiloft. If you try to reuse a url with a session id in it sometime later, it probably won't work. So you'll need to use the longer form of the url as above.

Putting it all together

When you put the url into the web browser, the following happens:

1. The Agiloft server looks up the page template in the database and retrieves it
2. The page template is passed through the Velocity template engine for macro expansion. In the example above, `#ew_table(...` is a macro that adds a table to a page. More on that later.
3. Once all the macros have been expanded, the finished HTML page is sent back to the web+ browser for display.

The HTML in the page may contain other links (URLs) for page elements, i.e., images. In this case the web browser will separately request all these links too - assembling all the results into the finished page.

A CLOSER LOOK

Let's take a closer look at the earlier example and see how all the theory works in practice.

Here's the page template "tesContacts.htm" from earlier, with line numbers added for ease of reference:

```
1. <html>
2. <head><title>Test portal</title></head>
3. <body>
4. <h1>Test portal</h1>
5. <div align="left">This is my new portal</div>
6. <p/>
7. <div align="left"><font size="+2">Contacts</font></div>

8. #ew_table("contacts" "Default Contact View"
   "" "" "" "contacts_frame" "")

9. </body>
10. </html>
```

Once this template has been retrieved from the database, it is passed through the Velocity template engine for macro expansion as described earlier. Line 8 is a velocity macro. Let's take a look at the finished HTML page that is sent back to the browser once macro expansion has taken place:

```
<html>
<head><title>Test portal</title></head>
<body>
<h1>Test portal</h1>
<div align="left">This is my new portal</div>
<p/>
<div align="left"><font size="+2">Contacts</font></div>

<iframe name="contacts_frame" width=100% height=100% border=0 class=""
src="/gui2/data/search.do;
jsessionid=FC9AC66893B5D7DA29DE3D07998D74050247?
q_id=63717&showToolBar=false&view=42744"></iframe>

</body>
</html>
```

(You can see this for yourself by right clicking in the browser window and choosing "View Source" in Internet Explorer or "View Page Source" in Firefox).

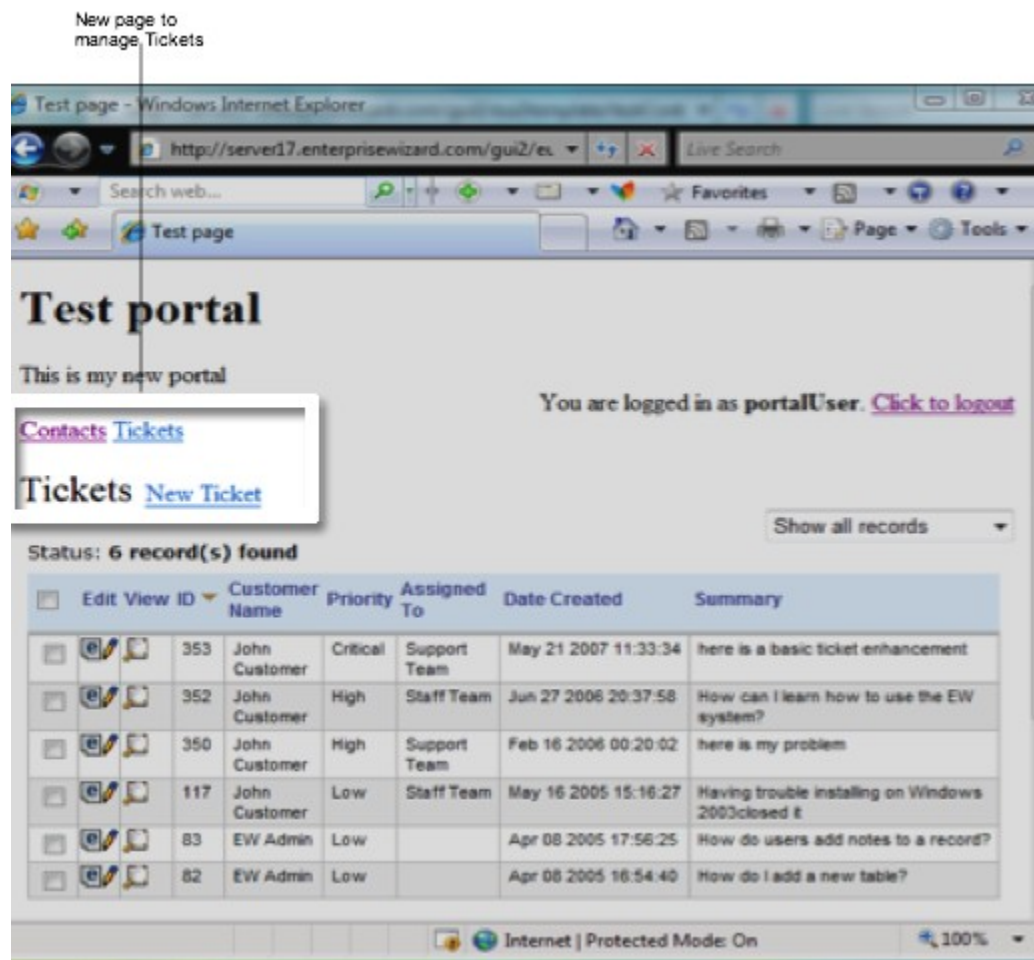
Line 8 contains the velocity macro `#ew_table` in the page template, which is expanded in the finished page with the `iframe` page element. The `iframe` has a `name` attribute with the value "contacts_frame" as specified in the macro. (Learn more about the `iframe` HTML element by going to <http://www.w3schools.com/> and following the *Learn HTML* link on the left hand side).

The `iframe` creates an area in the browser window and the content of the area is obtained by the browser requesting it separately from the server as described above. The value of the `src` attribute (`/gui2/data/search.do.....`) specifies the url that the browser requests to get the content. The HTML that is returned is complex; it references `.css` stylesheets and javascript functions. You can see what it looks like yourself by right-clicking somewhere on the table and choosing "View (Page) Source" as before.

ADDING MORE PAGES

What we've done works nicely, but a real portal will have multiple pages, working together to give users access to what they need.

In this section, our aim will be to expand the portal we're creating to have two pages that the user can switch between; one to manage contacts and the other to manage tickets. We'll add the ability to use saved searches to filter the records displayed, a convenient link for the user to logout, and some status information to display the current user's login.



ADDING RECORDS

First, we'll add a new link to the page to allow the user to add a new contact.

As Admin, edit the testContacts.htm page template, adding the lines highlighted in blue, so that it looks like the following:

```
<html>
<head><title>Test portal</title></head>
<body>
<h1>Test portal</h1>
<div align="left">This is my new portal</div>
<p/>
<div align="left">
<font size="+2">Contacts</font>

<a href="#ew_create_record("contacts"
    "/eui2template/testContacts.htm" "")">New Contact</a>

</div>
#ew_table ("contacts" "Default Contact View" "" "" "" "contacts_frame" "")
</body>
</html>
```

What we've done is add a link "New Contact". The url of the link is supplied by the velocity macro `#ew_create_record`.

Take a look at the parameters for the macro; the first specifies the name of the table that we want to create a new record for. The name of the table is found by going to Setup/Tables/, then choosing the table you want and clicking the Edit button. On the General tab of the table wizard, you will see the following values:

*Plural Table Label

Contacts

*Table Name

sw_34546f_contact_table2

Logical Table Name

contacts

The table name you want is the one displayed as the Logical Table Name.

The second parameter tells Agiloft which url to redirect to once the user has finished adding the new record and presses "Finish" or "Cancel". In this case we're specifying that the system should redisplay our contacts page once the user has finished adding the record.

Save the changes and re-test the portal interface. Add a new user and see how the contacts page re-displays when you've finished.

Next we'll add the saved search selection drop-down.

ADDING SAVED SEARCHES

Edit the testContacts.htm template again so that it looks like the following:

```
<html>
<head><title>Test portal</title></head>
<body>
<h1>Test portal</h1>
<div align="left">This is my new portal</div>
<p/>
<div align="left">
<font size="+2">Contacts</font>
<a href="#ew_create_record("contacts" "/eui2template/testContacts.htm" "")">
New Contact
</a>
</div>

<div align="right">
#ew_searches("contacts" "contacts_frame" "contacts_select" "")
</div>

#ew_table ("contacts" "Default Contact View" "" "" "" "contacts_frame" "")
</body>
</html>
```

The `#ew_searches` macro adds all the HTML and javascript needed to add a drop-down list of saved searches and hook it up to the displayed table so that when the user selects a search, the table re-displays with the results of the chosen search. Which searches are displayed will be dynamically based on the user's group permissions.

Take a look at the parameters of the macro. The first specifies the name of the table that we want to list the saved searches for. The second parameter tells Agiloft the name of the iFrame element which is used to display the results in. Notice how we've specified the name of the iFrame that we created earlier, with the `#ew_table` macro (the 5th parameter). That's because we want the table we've created to refresh with the new saved search results when an entry in the saved search drop-down is selected.

The third parameter gives a name for the HTML **select** element that is created by this macro. That's useful if you want to refer to the element for some reason such as changing some attributes of the element using javascript. We won't be taking advantage of this functionality in this tutorial.

Save the changes and re-test. See the new drop down list on the right hand side of the screen. Try choosing a saved search and see that the contacts table refreshes with the search results.

We're almost ready to add the Tickets page to the portal but before we do we're going to make an included template to contain the content that we want to be common across both pages.

MAKING AN INCLUDED TEMPLATE

Create a new page template. Call it "testHeader.htm", and type or copy-and-paste the following into the *Body* field:

```
<html>
<head><title>Test portal</title>
</head>
<body>
<h1>Test portal</h1>
<div align="left">This is my new portal</div>
<div align="right">
You are logged in as <b>#ew_user()</b>.
<a href="#ew_logout('http://www.example.com')">Click to logout</a>
</div>
<p/>
```

Notice the two velocity macros, `#ew_user` and `#ew_logout`. These expand to the login of the current user, and a link to logout of Agiloft, respectively (the logout macro takes a url parameter; this is the url the user is redirected to after they have logged out).

Save the new template, and then go back to the testContacts.htm template and open it for editing. We're going to remove some of the content from the contacts template and obtain it by including our new header template instead.

Here are the first few lines of the testContacts.htm template, with line numbers added for convenience:

```
1. <html>
2. <head><title>Test portal</title></head>
3. <body>
4. <h1>Test portal</h1>
5. <div align="left">This is my new portal</div>
6. <p/>

7. <div align="left">
8. <font size="+2">Contacts</font>
...
```

What we're going to do is delete lines 1 - 6 (highlighted in red above) and replace them with the `#ew_include` macro, so that the template looks like this:

```
1. #ew_include("testHeader.htm")
```

```
2. <div align="left">
3. <font size="+2">Contacts</font>
...
```

Save the changes and test out the testContacts.htm page.

We're nearly there. Next we're going to quickly create the Tickets page and then add a couple of navigation links to the included header we've just made.

Create a new EUI template. Name it "testTickets.htm", and type or copy-and-paste the following into the Body field:

```
#ew_include("testHeader.htm")
<div align="left">
<font size="+2">Tickets</font>
<a href="#ew_create_record("case" "/eui2template/testTickets.htm" "")">
New Ticket
</a>
</div>
<div align="right">#ew_searches("case" "tickets_frame" "case_select" "")
</div>
#ew_table ("case" "Staff default" "" "" "" "tickets_frame" "")
</body>
</html>
```

This is really just a straight copy of the testContacts.htm template with a few changes to reference tickets rather than contacts. It assumes that you have a table in your KB with a logical name of case.

Save the new template, and test it to make sure it works. You can do this by viewing the testContacts.htm page in your browser, and then editing the url, substituting testContacts.htm with testTickets.htm.

All we've got to do now is add links to the header template to allow the user to choose between the tickets and contacts pages.

Open the testHeader.htm template for editing, and add the two lines highlighted in blue so that it looks like the following:

```
<html>
<head><title>Test portal</title></head>
<body>
<h1>Test portal</h1>
<div align="left">This is my new portal</div>
<div align="right">
You are logged in as <b>#ew_user()</b>.
<a href="#ew_logout('http://www.example.com')">Click to logout</a>
</div>
```

```
<a href="#ew_forward("testContacts.htm")">Contacts</a>  
<a href="#ew_forward("testTickets.htm")">Tickets</a>  
  
<p/>
```

Notice the *#ew_forward* macro. This macro expands into a url to forward the user to the template named in the parameter.

Save the changes to the template and test out the portal. You should now have two pages, with links so the user can switch between them. The two pages should allow you to view, search and add new Contacts and Tickets. There should be a link to let you logout, and there should be some information text to display the login of the current user.

ADDING SOME STYLE

Finally - we're going to end this tutorial by demonstrating the use of the `#ew_url` macro to link in a `.css` stylesheet (See <http://www.w3schools.com> to find out more about stylesheets).

Create a new template and name it "testStyles.css". Type or copy-and-paste the following into the Body field: (Don't forget to click "Finish" when you're done)

```
body {font-family: arial;}
a:link {color: blue}
a:visited{color: blue}
a:hover {color: teal}
```

This simple stylesheet will change the font of our page to be arial, and sets the colors for both unvisited and visited links to blue, turning them to teal when you move the mouse over them.

Now open the testHeader.htm template for editing, and add the lines highlighted in blue so that it looks like the following:

```
<html>
<head><title>Test portal</title></head>

<link rel="stylesheet" type="text/css"
href="#ew_url("eui2template/testStyles.css")"/>

<body>
<h1>Test portal</h1>
<div align="left">This is my new portal</div>
<div align="right">
You are logged in as <b>#ew_user()</b>.
<a href="#ew_logout('http://www.example.com')">Click to logout</a>
</div>
<a href="#ew_forward("testContacts.htm")">Contacts</a>
<a href="#ew_forward("testTickets.htm")">Tickets</a>
<p/>
```

Click "Finish" to save, and see the changes in your portal's appearance.

Why did we use the `#ew_url` macro? Well the testStyles.css stylesheet that we created is stored as a record inside the EUI Templates database table. When the browser receives the HTML for the portal pages, it follows the url in the href attribute of the `<link>` element to request the contents of the stylesheet. So just like a url for a page template, it needs to contain all the information needed to

prove the request is coming from a valid Enterprise User, such as the *jsessionId* described earlier.

That's what the *#ew_url* macro does - it turns the relative url passed as the parameter, into a valid Agiloft url complete with user session information.

Here's an example of what the HTML looks like after macro expansion has been performed:

```
<link rel="stylesheet" type="text/css"  
href="/gui2/eui2template/testStyles.css;jsessionId=83B5B0A352942DAEAB0247"/>
```

WRAPPING IT UP

That's it - you should now have the basic knowledge you need to carry on and create a useful portal for your users. We've covered:

- How to create portal page templates in the EUI Templates table, and how to access them through a browser
- What's contained in the urls that your browser uses to access page templates
- Use of the *ew_table*, *ew_searches* and *ew_create_record* macros to add record management functionality
- Use of the *ew_user* and *ew_logout* macros to display login information and provide a logout link
- Use of the *ew_forward* macro to provide links that forward the user to different page templates
- Use of the *ew_include* macro to pull common content into multiple pages
- Use of the *ew_url* macro to create a valid Agiloft url including user session and authentication information

A good way to learn more is to go through the default portal template pages that are included with the default KnowledgeBase. With the knowledge you've gained already you should be able to understand how the templates work.

Because templates use the Velocity engine for Agiloft macros, any other Velocity constructions are also available for your use. Please see Velocity's documentation for details here: <http://velocity.apache.org/engine/releases/velocity-1.4/user-guide.html>

If no login details are included in the url when a user accesses Agiloft, the system will try to login with user=eui2, password=qwerty. If you would like to take advantage of this automatic login, ensure that a user created with those login credentials exists, and is granted the appropriate permissions.

If users belong to different groups, and their end-user interfaces allow them to access different areas of your Agiloft application, you will need to create a separate portal interface for each group. All users logging into one particular portal template will see the same HTML, regardless of the permissions granted to their individual groups. When users begin to interact with the Agiloft application through the portal controls (for example attempting to add a new record), detailed group and user permissions will come into play, and may restrict or permit access to tables, fields, and records. Unlike the default end user interface, however, the tabs and links shown in the portal will all be displayed irrespective of the users' permissions.

Agiloft Macro Reference

AGILOFT MACRO REFERENCE

#ew_session()

Prints Agiloft session ID to screen. This may be used to 'manually' construct a url that contains the session information.

```
<a href="/gui2/eui2template/testContacts.htm;#ew_session()">Contacts</a>
```

results in:

```
<a href="/gui2/eui2template/testContacts.htm;jsessionid=D7988DF0E8E173...">
```

#ew_login(\$kbName \$redirectTo)

Prints Agiloft login URL. This tag must be enclosed in <form> with input controls "login", "password" inside.

Parameters

\$kbName	name of KB to login
\$redirectTo	name of template to be

if no user and login is passed, then system will attempt to login under predefined user eui2/qwerty.

#ew_logout(\$url)

Logs out of Agiloft and redirects the user to the specified url.

Parameters

\$url	URL to be redirected aft
-------	--------------------------

Example usage:

```
<a href="#ew_logout("http://www.example.com")">Click  
to logout</a>
```

`#ew_user()`

Prints the current user login to screen.

`#ew_userid()`

Prints the current user ID to screen.

`#ew_userfield($fieldName)`

Prints current user's record field value. Note - it's rendered to be human-friendly (i.e. choice will be returned as Yes etc)

<code>\$fieldName</code>	A logical name of the field to return. Example: <code>#ew_userfield("_16_company_name0")</code>
--------------------------	---

`#ew_forward($templateName)`

Creates a URL to forward the user to the specified Agiloft template.

Parameters

`$templateName`

Template name. If such with enough info in the

Example usage:

```
<a href="#ew_forward("testContacts.htm")">Contacts</a>
```

`#ew_include($templateName)`

Parameters

`$templateName`

Includes \$templateName common pieces of code example of how this can

`#ew_scripts()`

Introduces a link to certain required JavaScript functions. Agiloft is a complex system, requiring some custom JavaScript code. Use it within a page's `<head>` tag if you are getting strange behavior and / or JavaScript errors about null and undefined objects in your portal templates.

`#ew_url($url)`

Parameters

\$url

Prints to screen \$url with encoded. If you want to portal users but not to a you can upload it to the a URL, including session

Example usage:

```
<img border = "0" src = "#ew_url("images/logo.jpg")"/>
```

#ew_status()

Prints to screen last operation status stored in the request (ActionStatus)

#ew_faq(\$subtypeName)

Prints to screen the URL of the standard FAQ page for the given subtype.

#ew_languages(\$controlName \$onChange \$controlStyle)

Renders a list of available languages for the current user with a particular HTML style. We need to pass the style here as a parameter, because Agiloft will render the HTML control automatically, making it impossible to customize its appearance any other way.

Parameters

\$controlName	Required name of the HTML select list.
\$onChange	JavaScript code to be run when a language is selected.
\$controlStyle	CSS class name for the select list control.

#ew_languages_all(\$controlName \$onChange \$controlStyle)

Renders a list of all languages available in the system with a particular HTML style. We need to pass the style here as a parameter, because Agiloft will render the HTML control automatically, making it impossible to customize its appearance any other way.

Parameters

\$controlName	Required name of the HTML select list.
\$onChange	JavaScript code to be run when a language is selected.
\$controlStyle	CSS class name for the select list control.

#ew_locale(\$localeName)

Sets the language (locale) for the current user session to the specified one if the language is available for the user.

Parameters

\$localeName	ISO-639 language code ("en", "de", "fr")
--------------	--

#ew_table(\$subtypeName \$viewName \$searchName \$queryString \$params \$frameName \$frameStyle)

Draws an HTML iFrame with the specified Agiloft table displayed inside it.

Parameters

\$subtypeName	The table to show.
\$viewName	The view to use for table. Note that it's localized for now, so different names should be used for different languages
\$searchName	The search name to apply to the table. Can be used simultaneously with \$queryString
\$queryString	The query to be applied to the table. When used, the result is the same as if the user had typed the contents of this string into the search box. See #ew_query macro below.
\$params	Visualization Parameters formed as a request query, e.g. "showstatus=false&showpages=false..." etc. Use it for precise tuning of the

table's display features. See the parameter list at the end of this section for more details.

\$frameName	The required name of the resulting iFrame. This is needed so that you can refer to it in other macros or pieces of JavaScript (for example to refresh the table view after a search has been applied - see the tutorial).
\$frameStyle	The CSS style name to be applied to the iFrame.

[#ew_create_record\(\\$subtypeName \\$returnURL \\$returnFrame\)](#)

Creates a URL that links to the standard Agiloft record creation page.

Parameters

\$subtypeName	Logical name of the table in which the new record should be created. (The logical name is shown in the tables wizard). Can contain subtable chains like case.question, contact.employee etc.
\$returnURL	URL to return to after the ticket is created or cancel is pressed.
\$returnFrame	Name of the iFrame that contains the view of the table that needs to be refreshed once the operation is completed.

[#ew_edit_record\(\\$subtypeName \\$recordId \\$returnURL \\$returnFrame\)](#)

Creates a URL that links to the standard Agiloft record editing page.

Parameters

\$subtypeName	Logical name of the table in which the new record should be created. (The logical name is shown in the tables wizard). Can contain subtable chains
---------------	--

like case.question, contact.employee etc.

`$recordId` The id of the record to edit

`$returnURL` URL to return to after the ticket is created or cancel is pressed

`$returnFrame` Name of the iFrame that contains the view of the table that needs to be refreshed once the operation is completed.

`#ew_searches_list($subtypeName $frameName $controlName $controlStyle $params $selectedSearch)`

Draws an HTML select list control with the Saved Searches available taking into account access permissions.

Parameters

`$subtypeName` Name of the table for which to display the appropriate searches.

`$frameName` Name of the iFrame that contains the view of the table that needs to be refreshed once a search is selected.

`$controlName` Required name of the HTML select list.

`$controlStyle` CSS class name to be applied to the select list.

`$params` String of the request Parameters that will be passed to the table in `$frameName` (see the `#ew_table` macro, and the following parameter list for details).

`$selectedSearch` Name of the search to be selected in list initially

`#ew_query($subtypeName $frameName $controlId $params)`

Provides a piece of JavaScript (mostly for use in `onClick/onChange` event handlers) that runs a search for a table in a iFrame

Parameters

\$subtypeName	The name of the table to show
\$frameName	The name of HTML iFrame in which to display the results.
\$controlId	The id of the HTML control which contains the query to use (for example a form field)
\$params	String of request Parameters that will be passed to the table in \$frameName (see the #ew_table macro, and the following parameter list for details).

#ew_new_search(\$subtypeName \$frameName \$allowSaving \$returnURL \$params)

Provides a piece of JavaScript (mostly for use in onClick event handlers) that runs the new Saved Search wizard

Parameters

\$subtypeName	Table for which the search should be created
\$frameName	The name of HTML iFrame in which to display the results after the Wizard has finished.
\$allowSaving	"true" if the user should be allowed to save the created search, otherwise "false".
\$returnURL	URL that should be applied to \$frameName after wizard has finished. If "", than the standard table search will be used.
\$params	Visualization Parameters formed as a request query, e.g. "showstatus=false&showpages=false..." etc. They will be applied to the table view for \$subtypeName. Use this parameter if you leave \$returnURL empty. Otherwise add required

Parameters to \$returnURL directly.

When returning to \$returnURL, an additional request parameter is appended for the created search (that was saved) - searchID. You can use it in your own code, for example to update a list of available Saved Searches.

#ew_repchart (\$repchartName \$rephartType \$tableName)

Forms the URL to show the repchart (Report/Chart) with the given name.

Parameters

\$repchartName	Name of repchart as shown in repcharts list.
\$rephartType	Allowed values are [html, image, excel, plain/text].
\$tableName	Name of the table the repchart belongs to.

#ew_image(\$imageName)

Forms the URL to the image with the given name. You can upload images to your KB via staff interface (logged as admin) in the Look&Feel section.

Visualization Parameters (\$params)

Applies to:

```
#ew_table  
#ew_searches_list  
#ew_query  
#ew_new_search
```

Parameters (default value in bold)

showSearchCriteria	true , false Whether used search criteria will be shown to user
workWithPopUp	true, false Determines whether edit and new operations should be performed within the current frame or the pop-up window

showToolBar	true , false Show toolbar on top of table (Edit, Delete etc)
showNavigation	true , false Show pages navigation above the table
showStatus	true , false Show status of operations on above the table
showfastsearch	true , false Show search dialog for the table

Additional tools.

Some helper macroses have been implemented to be used in velocity templates.

User related tools

In a template, one can use \$ewUser tool macros with several handy methods:

\$ewUser.isInGroup(gr_name)	<p>Returns true if current user belongs to any of pointed groups. False otherwise.</p> <p>gr_name - names of groups separated by comma</p> <p><i>Example: #if (\$ewUser.isInGroup("Staff,admin")) This is text for permitted users only #else This is text for all other users #end</i></p>
\$ewUser.isInTeam(tm_name)	<p>Returns true if current user belongs to any of specified teams. False otherwise.</p> <p>tm_name - names of teams separated by comma</p>
\$ewUser.getCurrentUserLogin()	Returns login name of the current user
\$ewUser.getCurrentUserField(field)	Returns string with the field value from the current user contact.
	field - name of the contact field
	<i>Example: Welcome \$ewUser.getCurrentUserField("full_</i>

name"!

EMAIL

EMAIL CONFIGURATION

To Configure Email follow these steps:

- Configure main email server and send test email
- Configure inbound email accounts for each table for which you will be sending email (even if you don't plan on giving customers an inbound address to use)
- Configure outbound email accounts for each table, defining how outbound emails are sent

The order is important. When you set up outbound email, you want to have inbound accounts set up already so you can use them as default outbound from address

CONFIGURING THE EMAIL SMTP SERVER

Go to Setup/Email/Configure Email Server

The default value of localhost will work if Sendmail is installed on the server. Otherwise, an SMTP server must be specified.

Setup → Email → Outbound Email Settings

SMTP Server

Finish **Cancel**

The SMTP server's role is to deliver outgoing email. Please specify the IP address or hostname of your SMTP server, for example, *smtp.yourhostname.com* or *localhost* if the SMTP server is on the same machine. If your SMTP server uses a non-default port, please enter it here. The default is Port 25.

SMTP servers may or may not require authentication. If your SMTP server does not require authentication, leave the login and password fields blank.

SMTP Server Info

*SMTP server :

SSL/TLS connection STARTTLS

SMTP login

SMTP password

Please enter your email address to have a test email sent to it. This message will confirm that the SMTP server is correctly configured to accept outgoing email from your installation of Agiloft.

Enter email address to send a test message

Each email generated manually or automatically in Agiloft may be configured to use its own outgoing email address or a default system account. This setting allows you to define the system default sender's (FROM:) address for outbound mail.

*Default outbound email address

Finish **Cancel**

See Also:

[Incoming Email Accounts](#)

[Outbound Email Settings](#)

[Communication Tables](#)

[Email Wizard](#)

Email Marketing and Tracking

INCOMING EMAIL ACCOUNTS

Configuring Inbound Account

Go to Setup/Email/Configure Inbound Email

Setup → Email → Configure Inbound Email

Table Server Account **Record Mapping** Email Updates Email Parsing Access Filters

Back Next Cancel

Store email subject in

Problem Description
Solution
Summary
Closed By

Select the field that will store email address of the sender.

Store sender's address in

Customer Email

The body of the email can be mapped to a field or entered into the Communications table. Typically you will want to populate a large-text field in the table of your choice with the email message body.

Store email message in

--Do not map--

Important: If the email body contains field labels and field values using the following format: *field_label: field_value*, (example: **Status:Closed**), and the user has the permission to edit the listed fields, the field content will be placed into the corresponding field rather than into the generic text field selected here.

From the main Setup/Communications tab, click the Configure Inbound Email button to display a table of configured incoming email accounts for all tables. Click New and select the table that will receive inbound email. Then configure inbound account: input a type of email server and its IP address or name. Input the account's name and password. Next, configure a record creation from inbound email. Define what fields Email Wizard will save the email data in. Possible information that can be saved includes the subject of the email, sender's email address, body of the email and attached files.

You can also define a rule on how to update or create records. To update a record you must indicate the ID of an existing record in the subject line. To define a record

update function, select what fields Agiloft will save the email to, and define if the data should overwrite existing data, or be written before or after it.

You can also define incoming email filters for spam protection, and to optimize handling of incoming email. Email Wizard can discard email containing certain strings in the subject line or sender address, disable auto-response to such emails, and stop responding to a particular email address if a lot of emails are received from it in a short time period.

Inbound Email Accounts

Agiloft checks inbound email accounts and downloads new messages every minute. New messages are copied into the Communications table for the associated email account's table. Depending on the contents of a particular email, the following processes occur:

- **If a Record ID is encoded in the subject or header** the system recognizes the email as an update to that record. Depending on your email settings, parts of the email can be copied to the Additional Notes field of the record, and change the Updated By field.

When a record ID is encoded in the email, Agiloft does not check the permissions of the sender. This allows non-Agiloft users to respond, for instance if they were CC'd on a ticket, and saves their update to the record.

- **If no Record ID is encoded, and the sender's email matches a user in the Contacts table** the system creates a new record under that user's name if they have the appropriate permissions.
- **If no Record ID is present, and the sender is not recognized** the system will either ignore or reject the email, or create a new record under a user account that you specify during setup (often a "guest" user account).
- If you are going to allow emails from unknown users, choose your guest account and its group permissions carefully. Be sure it has "create own" access to the table and the fields you have mapped.

Email updates do not check permissions of sender - they are applied to correct records without limitation - this is to enable people who were cc'd about the original ticket to be able to successfully reply to the system even if they are not users

Must we have an Inbound Account?

Tip: Leave email on the server for at least a day or two, since this allows troubleshooting of inbound email

Do we need an Inbound Account?

- Inbound accounts are not required for the system to function. However, without one you shut down all incoming email communication.
- Without an inbound account matching the reply-to address, users will not be able to respond or update their tickets via email.
- To discourage users from creating records by email, you can set a Reply-to Address in outbound emails that sends an auto-response saying "You have emailed a non-working account. Please submit your request online".

Tips for Inbound Email Mapping

- Specify different choice field values for different inbound accounts to the same table to automate assignment and enable better automatic processing. For instance a Problem Type of "Billing" can map to an email account called billing@xyz.com, and a Problem Type of "New Account" Setup can map to newaccount@xyz.com.
- Send "new email" notifications to techs by mapping inbound emails to a text field that overwrites with information from the new email. This way you can set up an automatic email notification that says "An email was just received from \$updated_by that says: \$latestemail." It is still possible to include all communications in the email, but not as easily as inserting a field variable.

Permission Checking for New Emails

- When a new email is received, the system looks at the sender 's address and checks its Contact tables for a match in any email field. If it finds one, it applies the user's group permissions to the request to create a record
- If it finds no match, it will either reject the email or apply permissions for the user account specified on the Access tab
- If you are going to allow emails from unknown users, choose this account and its group's permissions carefully to be sure it has create own access to the table and the fields you have mapped
- Tip: Leave email on the server for at least a day or two, since this allows troubleshooting of inbound email

Important Note for SW Customers

Agiloft no longer supports the field-update by email scanning feature that was available in SupportWizard. Instead, Agiloft offers a way to log users on in one click through a [specially created link](#), and an improved licensing structure for unlimited end-users that greatly improves the efficiency of data collection. Instead of relying on often-haphazard and error-prone user email to update fields, you can now allow a user to log on automatically and have Agiloft guide them in inputting or updating any required information. This allows the system to actively collect the correct data instead of passively waiting for the user to provide it. Due to implementation of unlimited end-user licensing, this option is now also more cost effective.

If email update scanning must be preserved, the functionality could be recreated in Agiloft by mapping an email body to a large text field and creating a custom script to scan it for field updates.

See Also:

[Global Email Settings](#)

[Outbound Email Settings](#)

[Communication Tables](#)

[Email Wizard](#)

Email Marketing

OUTBOUND EMAIL SETTINGS

What does Outbound Email Control?

- Outbound email configuration defines the default outbound addresses used for each table and some characteristics of emails sent by the system or manually by users from within the system
- If you do not specifically set it up for a table, the system will use the defaults shown in the wizard, so the wizard allows you to change from the system defaults to your own selections

Outbound Email Setup

Outbound Email Setup Wizard can be invoked by selecting Setup/Email/Configure Outbound Email tab. It contains options for how outgoing email messages are handled on a per-table basis. Those options can be inherited by sub-tables, so you may simply set up one set of options for each top level table and it will apply to all sub-tables.

You can define From and Reply-to addresses for system-generated and/or user-initiated emails. It is possible to make those addresses different. Users can input those addresses manually or select a person or default email address from users and emails fields. User list and email list fields display all fields of type email or email pager in the current table and also include fields that are linked to or copied from the Login or Full Name field of any sub-table of the Contacts table.

An administrator can give some groups permissions to see and change the default sender's and Reply-to addresses for user-initiated messages.

Additional features of Outbound Email Setting submenu are configurable format for outgoing user's email, configurable parameters for original (quoted) text, ability to configure the system to automatically send the CCs or BCCs to the selected addresses.

See Also:

[Global Email Settings](#)

[Unified Communications Setup](#)

[Incoming Email Accounts](#)

[Communication Tables](#)

[Email Wizard](#)

Email Marketing

EMAIL PARSING

Supported Use Cases

Direct Emails: A third-party system is capable of generating emails with appropriate formatting for parsing by Agiloft. These inbound emails are parsed and used to update specific fields in records. If an encrypted ID of an existing ticket is found in the subject line, the system updates that ticket. Otherwise it creates a new ticket. Whether it is a new ticket or an update to a ticket, the system will only update specific fields if field-specific updates have been enabled by the admin for that table.

Response Emails: A user may respond to an email generated by Agiloft by clicking Reply and editing some fields. Agiloft parses the email that is returned and updates those fields that have changed, provided that the user has authority to change them.

Supported email content

HTML emails

Agiloft receives a HTML formatted letter such as following example

```
Some HTML text that does not get mapped to specific fields
--Agiloft TICKET INFO; EN--
ID:                8077
My Issue:          [I would like to request an enhancement]
Priority:           Critical
Summary:           Add a parameter "SMTP-Assertion" to the Agiloft-audit email
header
Description:       Currently Agiloft creates an email. It is not necessary
Status:            Open
Date Created:      Apr 21 2008 07:27:48
--Agiloft TICKET INFO--
More text that does not get mapped to specific fields
```

The above example illustrates what the HTML text looks like in the user's email client. The user may have edited the field contents and these changes will be reflected in the ticket after he has emailed it back to Agiloft.

Note: Due to limitations of email clients, it is not currently possible to specify field specific changes to images or attached file fields.

How does it work?

The above example contains delimiters such as the following header and footer:

```
--Agiloft TICKET INFO; EN--
--Agiloft TICKET INFO--
```

The specific text for the header and footer can be configured in the Inbound Email Configuration Wizard.

The example specifies updates using the field labels: ID, Answer, My Issue, etc.

When Agiloft receives a message like this it parses the text between the specified header and footer that may look like the following:

```
<br>--Agiloft TICKET INFO; EN--<br>
<table border="1"
cellspacing="0" cellpadding="0"
style="mso-cellspacing:0in; border:solid white .25pt
;mso-padding-alt:.7pt .7pt .7pt .7pt">
<tr>
<td class="SIEmailLabel">ID:</td>
<td class="SIEmailData">8077</td></tr>
<tr>
<td class="SIEmailLabel">My Issue:</td>
<td class="SIEmailData">
I would like to request an enhancement
</td>
</tr>
<tr>
<td class="SIEmailLabel">Priority:</td>
<td class="SIEmailData">Critical</td>
</tr>
<tr>
<td class="SIEmailLabel">Summary:</td>
<td class="SIEmailData">
Add a parameter "SMTP-Assertion"
to the EW-audit email header
</td>
</tr>
<tr>
<td class="SIEmailLabel">Status:</td>
<td class="SIEmailData">Open</td>
</tr>
<tr>
<td class="SIEmailLabel">Answer:</td>
<td class="SIEmailData">Currently the system
creates an email. It is not necessary</td>
</tr>
<tr>
<td class="SIEmailLabel">Date Created:</td>
<td class="SIEmailData">Apr 21 2008 07:27:48</td>
</tr>
</table>
<br>--Agiloft TICKET INFO--<br>
```

A field is updated if the matching field label is found and the user has the necessary group permissions to update it.

Please note the following restrictions

1. The user **MUST** not delete the header or footer from the email, nor should he change the language ID (EN in example above) that allows Agiloft to parse the NLS name of labels
2. Significant changes **MUST** not be made to the table format.

In other words, the format:

```
<td class="SIEmailData">8077</td>
```

MUST be kept.

The first `<td>` is a field label, second one is a field value.

This has been tested and confirmed to be working with the Microsoft Outlook email client.

3. The date format from the original letter should be preserved. If the original letter contains Apr 21 2008 07:27:48, but a reply is sent containing a date such as 07:27:48 1 May 2008 or 07:27:48 06/01/2008 it will be treated as an error.

Plain Text Emails

Agiloft can also receive a plain text letter such as the following:

```
--Agiloft TICKET INFO; EN--  
ID: 8077  
My Issue: I would like to request an enhancement  
Priority: Critical  
Summary: Add a parameter "SMTP-Assertion" to  
the Agiloft-audit email header  
Status: Open  
Answer: Currently Agiloft creates an email. It is not necessary  
Date Created: Apr 21 2008 07:27:48  
--Agiloft TICKET INFO--
```

It is a plain text letter so there are no hyperlinks, text formatting (bold, italic fonts, table formatting).

For a plain text letter Agiloft will parse `<NEW LINE>^String:` as a field label and all that follows it, until the next such token or the footer, will be parsed as the field

value. (where ^ is ether the beginning of a string or Reply's marker like ED>, -->, >>>, ends always with > sign)

The ":" after the field label is obligatory.

Notes

To allow the text content to contain field labels without them being parsed Agiloft will convert occurrences of <NEW LINE><TAB> to <NEW LINE>.

For example:

```
Description: For reason that I explained, the value given
for the priority, currently
Priority: Low is no longer correct
Priority: High
```

This will result in the Description field being set to:

```
For reason that I explained, the value given for the priority, currently
Priority: Low is no longer correct
```

and the Priority field will be set to High

In brief, a program that generates emails to the system should place a tab at the start of each new line of text within the field value to guard against the possibility that the line might start with a field label and colon. (Because otherwise the field value would be terminated by the new field label)

Agiloft will only parse the first block of text between header and footer markers.

For example if the inbound email contains:

```
Hi, James
We received your letter:
--Agiloft TICKET INFO; en--
ID: 8077
My Issue: I would like to request an enhancement
Priority: Critical
Summary: Add a parameter "SMTP-Assertion" to the Agiloft-audit email
header
Status: Open
Answer: Currently Agiloft creates an email. It is not necessary
Date Created: Apr 21 2008 07:27:48
--Agiloft TICKET INFO--
But found old one from other ticket
```

Agiloft Reference Manual

```
--Agiloft TICKET INFO; EN--  
ID: 8022  
My Issue: it works  
Priority: Critical  
Answer: Currently Agiloft is down  
--Agiloft TICKET INFO --
```

In this case only the data about ticket 8077 (the first block) will be considered for parsing.

XML Emails

Agiloft can also receive a plain text letter where XML syntax is used to distinguish the fields from their values.

The format is `<Agiloft XML:field label>data</Agiloft XML field label>`.

For example:

```
--Agiloft TICKET INFO; EN; XML--  
<Agiloft XML:ID>8077</Agiloft XML:ID>  
<Agiloft XML:Description>The cup holder in my computer is broken</Agiloft  
XML:Description>  
--Agiloft TICKET INFO--
```

Note: The text "XML;" must be included in the header (e.g. `--Agiloft TICKET INFO; en; XML --`) so the system can distinguish XML formatted emails from HTML formatted emails.

As with the other formats, the field label, not the field name is used.

Group Permissions

Both field level and record level permissions are applied when updating a record using a formatted template.

When an email is received that is formatted, the system checks that the user has the right to update the record in question and that he has access to any fields that are included.

If specific field permissions are missing, those particular fields are not updated, while any fields to which he does have permission are updated.

The system determines the user by finding the first user in the Contact table with an email that matches the From: address using a case-insensitive search.

Configuration

Configuration for Inbound Email Parsing is done in Setup/Email/Configure Inbound Email/{inbound configuration}/Edit/Email Parsing.

INTEGRATION

IMPORT/EXPORT

Importing Data to Agiloft

- Data is easily imported into Agiloft from a backup, standard ASCII, Excel, XML, Salesforce, and SupportWizard format.
- When importing data, it should usually be cleaned up first to be sure that choice field values, team and group names are correct, etc.

Importing Users from Excel

- Go to Setup > Import to open the wizard and select the file and upload it.

Importing Tips

- If you want to find invalid data without importing, you can click the Test button first.
- You can now have the system create any missing choices, so this is a nice way to allow data to actually populate choice tables.
- Generally it is a good idea to use excel files that have only one worksheet, as the system has been known to import sheets you didn't want imported.

Importing Data from SupportWizard

- Once you have your system set up, you will want to import your data from SW by creating an export to the multi-file export option (you need a recent release of SW to do this).
- Then you will import this tar file in Agiloft and it will recognize it and allow you to map the fields for each table prior to importing them all.

See Also:

[KnowledgeBase Import/Export](#)

[Table Import/Export](#)

[Import/Export Using Backup Files](#)

Import/Export Using Backup Files

You can export a KnowledgeBase by using the Backup Wizard in the [Admin Console](#). Click Setup/Backup, select KnowledgeBase and press "Run Backup now". You can also set a schedule for automatic backups. Each backup will create a new export file, with old files being overwritten when they get older than the backup expiration date you set using the Backup Wizard. The export files created by the Backup wizard can be used just like regular KnowledgeBase export files.

See Also:

[Admin Console](#)

[KnowledgeBase Import/Export](#)

[Table Import/Export](#)

[KnowledgeBase](#)

KnowledgeBase Import/Export

An alternative to the simple import/export of a single table is offered via the Setup/Import and Setup/Export wizards. These wizards provide all the source location, format options and field mapping as the [Table Import/Export](#) but also facilitate the transfer of a complete KnowledgeBase, or several tables by means of a single file.

In addition to the standard formats provided by [Table Import/Export](#) (i.e. tab delimited, csv and Excel spreadsheet), you can also export into Agiloft's proprietary format. This has the added capability of transferring the table definitions and optionally exporting data, scripts, escalation rules, inbound email definitions and other customizations. All tables in the KnowledgeBase are transferred when utilizing the Agiloft format.

The standard transfer formats (.txt, .csv & .xls) may be used to transfer all KnowledgeBase tables, but can also transfer a selected subset of the tables in the KnowledgeBase.

Using the KnowledgeBase Import/Export, you can transfer entire Agiloft KnowledgeBases from one server to another. To make the transfer, export your KnowledgeBase in Agiloft format and then import it on the target server.

Only the users with project import/export permissions can export the entire KnowledgeBase in Agiloft format. To give a group import/export permissions for your KnowledgeBase, click Setup/Access/Manage Groups/{Select Group to edit}/Edit/General and enable the "Allow importing KnowledgeBase" or "Allow exporting KnowledgeBase" options.

You can only import a KnowledgeBase in Agiloft format using the [Admin Console](#). From there, click Setup/Import and enter the name for the new KnowledgeBase. It is no longer possible to import Agiloft KB using the Setup/Import wizard from within the Staff Interface.

See Also:

[Table Import/Export](#)

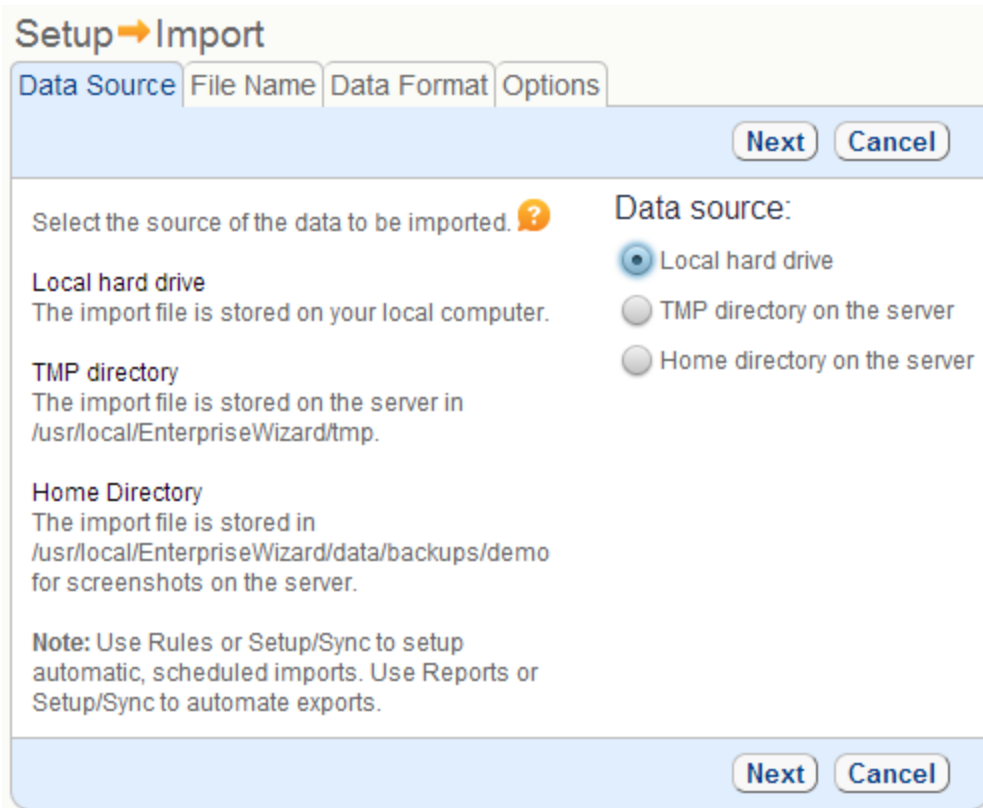
[Import/Export Using Backup Files](#)

[Admin Console](#)

[Groups](#)

Table Import/Export

Table Import/Export permits information transfer between Agiloft tables, text files and spreadsheets. The Import/Export wizard is invoked via Setup/Import. The wizard implicitly operates on the currently selected Table and provides control of the transfer format and field selection.



The source or target data location may be the local disk, the TMP directory or the server home directory. A table may be imported or exported directly to, or from, an Excel spreadsheet, or in tab-delimited (.txt) or comma-separated (.csv) format.

Setup → Import

Data Source | File Name | **Data Format** | Options

Back Next Cancel

Please select the data format of the import file.

If you are importing from a standard SupportWizard formatted export file, choose the SupportWizard option.

If you are restoring from a backup, choose the Agiloft option.

If you are importing SupportWizard data from a .tgz file created by a SupportWizard multi-file ASCII Export or a .tgz file that includes attached files, choose the option ".TGZ file with multiple text files/attachments". This .tgz file should contain not only the data files but also any attached files that you want to be imported, stored in the /files subdirectory.

If you are importing from a Salesforce database, choose the Salesforce Data option. Other formats should be automatically selected for you.

Data Format:

- Salesforce Data
- Tab-delimited text
- .TGZ file with multiple text files/attachments
- Other text file
- Excel spreadsheet
- XML file

Back Next Cancel

"Update imported fields only in matching records" option allows to update fields for matching records and append it if no existing records found. There is also another option for the Update and Replace modes: if no matching is found you can either import new record or skip it.

Data Source	File Name	Data Format	Options	Records	Fields
<input type="button" value="Back"/> <input type="button" value="Next"/> <input type="button" value="Cancel"/>					
Sheet name: Sheet1					
<p>There are five ways to import new records:</p> <ol style="list-style-type: none"> 1. You can append these records to the existing records in the table 2. You can look for matching records (using the ID or some other unique field to identify a match) and completely replace the existing record with the new record 3. You can skip matching records so that records in the import file that match existing records are ignored, i.e. the existing record is not modified at all 4. You can update the imported fields only in the matching records, leaving the other fields in the records untouched 5. You can delete all records first and then run the import (if the person running the import does not have delete permission, the records will be appended). Note that deleting all records first will greatly add to the time taken by the import 			<p>Act on Existing Records:</p> <ul style="list-style-type: none"> <input checked="" type="radio"/> Append new records <input type="radio"/> Replace matching records <input type="radio"/> Skip matching records <input type="radio"/> Update imported fields only in matching records <input type="radio"/> Delete all records before importing 		
<p>If you have chosen the second or third option above, choose the field that will identify a matching record. If you fail to include this field in your import file, the records will be appended.</p>			<p>Match records:</p> <p>By the <input type="text" value="ID"/> field</p>		
<p>Choose Create as new record mode to add records for the rows that don't have a matching record</p>			<p>If no matching record is found:</p> <ul style="list-style-type: none"> <input type="radio"/> Skip record <input checked="" type="radio"/> Create as new record 		
<p>In case there are two or more records in source file with the same value in "Match By" field, import can use all records or just the first one.</p>			<p>Duplicates:</p> <ul style="list-style-type: none"> <input type="radio"/> use all records <input checked="" type="radio"/> use first record 		
<p>Choose whether append only text fields must be overwritten with new data or not.</p>			<p>Append Only Text Fields</p> <ul style="list-style-type: none"> <input checked="" type="radio"/> Append new data <input type="radio"/> Overwrite with new data 		
<input type="button" value="Back"/> <input type="button" value="Next"/> <input type="button" value="Cancel"/>					

Field selection is accomplished via a field selection list. When Importing, Agiloft attempts to perform default mappings based on field type, but typically the user needs to manually match all import fields to existing table fields prior to import.

Setup → Import

Data Source | File Name | Data Format | Options | Records | **Fields**

Back Finish Test Cancel

Sheet name: Sheet1

Id	Sample Data	Column
1	Description	Asset Name
2	Very Low Factors	Asset Subtype
3	Hours - Very Low	Asset Tag
4	Low Factors	Asset Type
5	Hours - Low	CPU
6	Medium Factors	Current Contract End Date
7	Hours - Medium	Current Contract ID
8	High Factors	Current Contract Start Date
9	Hours - High	Current Contract Status

Back Finish Test Cancel

These tabs are only shown for users in groups with permission to Import/Export multiple records to a file. This permission setting is defined on the Permissions tab for the group. Users also need permission to 'create record' to import and 'view record' permission for export.

System displays a progress message about what sheet and row is processing during record importing. Now the user can leave the import page and view some records, then return back to the import and if the import is still in progress he will be shown a progress screen.

See Also:

[KnowledgeBase Import/Export](#)

[Import/Export Using Backup Files](#)

Command Line Utilities

Introduction

ewimpex is a command line utility that allows you to list, backup and restore any (or all) KB's from the database. It is not necessary for the Agiloft application server (eg JBoss) to be running when you execute this command, so you can shut off access to the system during exports and imports.

Usage

```
ewimpex <kblist | backup | restore | help> [options]
```

The available commands: kblist, backup, restore and help are described below, followed by examples.

kblist

The kblist command outputs a list of all available knowledge bases in the database, one knowledge base per line. Each line contains a tab-separated (\t) knowledge base ID and name, for example:

```
1 Demo
  2 My test kb
  3 Working kb
```

backup

The backup command exports selected knowledge bases to a specified folder, or a new "backup folder".

This command requires one of more of the following options:

```
-k (--kbases) - list of comma-separated knowledge bases
names to backup.
For example: -k Demo,Tickets,Support or --kbases Demo,Tickets,Support
-a (--all) - ignores -k option and tells ewimpex to backup all
available
user knowledge bases, including admin one.
For example: -a or --all
-o (--output) - folder to save knowledge bases to.
If not set, ewimpex will create a "backup" folder inside the current
working directory and place the backups in it.
-b - backup damaged knowledge bases as well.
Will backup kbases that are flagged as damaged
(ie with a projecttype > 1), this flag is OFF by
default.
```

The KnowledgeBases are saved to the specified folder using their original names, ending with .xml.zip extensions. All non-standard characters in file names are automatically replaced by the "_" (underscore) character.

restore

The restore command restores one or many knowledge bases, previously backed up by the ewimpex utility or the Agiloft GUI.

This command requires one or more of the following options:

```
-f (--file ) - knowledge base file name to restore.
For example -f MyKB.xml.zip or --file "My Best base.xml.zip"
-a (--all ) - tells to restore all knowledge bases, found in the
specified folder.
Option -f will be ignored, if present.
-s (--skip-admin) - skips admin KB from restore list
This option is only applicable with the -a option and is useful
to avoid importing the admin KB
-r (--replace) - instructs ewimpex to overwrite system tables during
import.
See the documentation for more details.
-x - disables knowledge bases patching.
This option is useful when importing a knowledge base without having
the
the application server (eg JBoss) running. If this is used,
the imported knowledge bases will be patched when the application
server
next starts up.
```

If you are restoring a KB that already exists, ewimpex will add a prefix to the restored knowledge base in order to avoid naming conflicts. For example, if you have "Demo" knowledge base installed and restoring another "Demo" knowledge base from the file, the result will be KB's named "Demo" and "0_Demo". After restoration, you may login to Agiloft as administrator and rename or delete unneeded copies of the databases.

Common Options

In addition to the command-specific options listed above, there are several options that are common to all commands that specify the database parameters. If none of these options are set, ewimpex will try to find the Agiloft installation and use its data, so these parameters are optional.

```
-w (--ew-home ) - path to the installation
folder to use
installation data. (if auto-detection fails for some reason)
-d (--database-type ) - type of the database. Currently, mysql, db2 and
```

```
mssql are supported.  
-h (--database-address ) - IP address of the database server to connect  
with; 127.0.0.1 is used by default  
-P (--database-port ) - port number for the database.  
Database-specific default port value is used if this option is missed  
-H - combines options -h and -P and allows setting the database  
address and port as one option.  
For example: -H localhost:3306  
-u (--database-username ) - username to use for database login  
-p (--database-password ) - password to use for database login  
-n (--database-name ) - database name to connect to.  
sw2_std is used by default  
-v (--verbose) - turn on debug output.  
Print a lot of debugging information to the console.  
Use only to investigate problems or to generate a bug report for  
developers.
```

Error codes:

On success, ewimpex returns a zero value. On error, the error message is printed and negative error code is returned:

```
-1 - database connection problems  
-2 - invalid parameters set  
-3 - invalid database structure  
-4 - internal error during import/export  
-5 - specified knowledge base file is invalid or damaged  
-6 - specified knowledge base is not an admin knowledge base  
-7 ... -999 - unexpected error
```

Examples

Get a list of all installed knowledge bases:

On Windows from INSTALL_DIR\bin\ do:

```
ewimpex kblast
```

On Linux/Unix from INSTALL_DIR/bin/ do:

```
./ewimpex kblast
```

Export all knowledge bases into the folder

On Windows from INSTALL_DIR\bin\ do:

```
ewimpex backup --all --output C:\backups
```

On Linux/Unix from INSTALL_DIR/bin/ do:

```
./ewimpex backup --all --output ~/backups
```

Export one knowledge base

On Windows from INSTALL_DIR\bin\ do:

```
ewimpex backup --kbases Demo --output C:\demos
```

On Linux/Unix from INSTALL_DIR/bin/ do:

```
./ewimpex backup --kbases Demo --output ~/demos
```

Export list of knowledge bases

On Windows from INSTALL_DIR\bin\ do:

```
ewimpex backup --kbases Demo,MyCRM --output C:\MyKB
```

On Linux/Unix from INSTALL_DIR/bin/ do:

```
./ewimpex backup --kbases Demo,MyCRM --output ~/MyKB
```

Import all knowledge bases from a folder to Agiloft system

Existing knowledgebases will NOT be overwritten

On Windows from INSTALL_DIR\bin\ do:

```
ewimpex restore --all C:\backups
```

On Linux/Unix from INSTALL_DIR/bin/ do:

```
./ewimpex restore --all ~/backups
```

Import one knowledge base

On Windows from INSTALL_DIR\bin\ do:

```
ewimpex restore --file c:\MyCRM.xml.zip
```

On Linux/Unix from INSTALL_DIR/bin/ do:

```
./ewimpex restore --file ~/MyKB/MyCRM.xml.zip
```

SOAP INTERFACE

Agiloft SOAP Web Services Quick Start

Enable the SOAP Web Services Interface

In Agiloft the SOAP Web Services interface is enabled on a per knowledgebase basis. Once enabled a WSDL is generated and a web services package is deployed automatically.

This package will persist through application server restarts and will be regenerated by Agiloft installer on update of the server.

WSDL defines a set of generic operations common to all knowledgebases, a set of data types specific to the knowledgebase in question and a set of table-specific flavors of the generic operations.

Each table in the knowledgebase known at the moment of WSDL generation is represented in WSDL as a complex type definition.

Table types extend a common **EWWSBaseUserObject** type.

The generic methods use this common ancestor as arguments and result type for the methods where relevant. Generic methods that would be invoked for specific tables are designed to accept the name of the table as a string parameter and they return the corresponding specific descendent type as the result in runtime. The client code is supposed to perform the necessary type casting if required.

Table specific methods use specific descendent types as arguments and result type where relevant. The name of the table is not used as it can be deduced from the method being invoked.

Web services can be enabled by admin-level user via *Setup/System* mode. Once enabled the WSDL is available for download.

Two flavors of SOAP Web-Services are offered:

- v2 - the current version using a token obtained on Login to identify the session in subsequent calls
- v1 - the legacy version using HTTP cookies to maintain the session after the initial login.

The legacy version is not described in this manual anymore. However all calls available in v2 are available in v1 with one difference - the sessionid parameter is not present, as the session is maintained via cookies. Though being supported, this

is considered non-standard and use of v1 Web Services is not recommended anymore.

Obtain the WSDL document for your knowledgebase

Once enabled, the endpoint for the SOAP Web Service for knowledgebase ABC will become available at URL: `http://host:port/ewws/ABC/EWSimpleAPIv2`.

One can access the WSDL document for the purpose of generation of the client code at the following address: `http://host:port/ewws/ABC/EWSimpleAPIv2?wsdl`.

Please note that by default for security reasons the WSDL document specifies localhost for the endpoint. One can either alter this immediately after obtaining the WSDL or override the endpoint location in the client code.

To reflect the latest changes made to your KB structure, re-enable Web Services for this KB.

Import the WSDL File into Your Development Environment

This section provides sample instructions for Apache Axis. For instructions about other development platforms, see your platform's product documentation.

Agiloft SOAP Web Services follow industry standards and are known to work with Java, .NET, PHP, Perl and Python client applications.

Once you have the WSDL file, you need to import it into your development environment to generate the necessary objects for use in building client Web service applications.

Instructions for Java Environments (Apache Axis)

Java environments access the API through Java objects that serve as proxies for their server-side counterparts. Before using the API, you must first generate these objects from your organization's WSDL file.

Each SOAP client has its own tool for this process. For Apache Axis, use the WSDL2Java utility.

Before you run WSDL2Java, you must have Axis installed on your system and all of its component JAR files must be referenced in your classpath.

The basic syntax for WSDL2Java is:

```
java -classpath pathToJAR/fileName org.apache.axis.wsdl.WSDL2Java
    -a pathToWsdldoc
```

The `-a` switch generates code for all elements, referenced or not, which may be necessary depending on your WSDL. For more information, see the WSDL2Java documentation.

If you have JAR files in more than one location, list them with a semicolon separating the files. For example, if the Axis JAR files are installed in `C:\axis-1_3`, and the WSDL is named `MyKB.wsdl` and is stored in `C:\myKB`:

```
java -classpath c:\axis-1_3\lib\axis.jar;c:\axis-1_3\lib\axis-ant.jar;  
c:\axis-1_3\lib\axis-schema.jar;c:\axis-1_3\lib\commons-discovery-0.2.jar;  
c:\axis-1_3\lib\commons-logging-1.0.4.jar;?c:\axis-1_3\lib\jaxrpc.jar;  
c:\axis-1_3\lib\log4j-1.2.8.jar;c:\axis-1_3\lib\saaj.jar;  
c:\axis-1_3\lib\wsdl4j-1.5.2.jar; org.apache.axis.wsdl.WSDL2Java  
-a C:\myKB\MyKB.wsdl
```

This command will generate a set of folders and Java source code files in the same directory in which it was run. After these files are compiled, they can be included in your Java programs for use in creating client applications.

For most Java development environments, you can use wizard-based tools for this process instead of the command line. For more information about using WSDL2Java, see <http://ws.apache.org/axis/java/reference.html>.

SOAP API Call Basics

SOAP API calls implement specific knowledgebase operations that your client applications can invoke at runtime to perform tasks, such as:

- Search and read data in your knowledgebase.
- Add, update, and delete data in your knowledgebase.

Please note that SOAP interface is most suited for online integration between systems, rather than processes that involve bulk synchronization. agiloft.com has other more suitable ways of doing this - for example, export/import, two-way email exchange, or the Sync API developed for this purpose.

Using your development environment, you can construct Web service client applications that use the standard Web service protocols to: programmatically:

- Log in to the knowledgebase and receive the session token to be used for subsequent calls.
- Perform SQL-based searches across the knowledgebase data.
- Create, read, update, and delete data.
- Trigger workflow transitions and business rules processing.

Characteristics of API Calls

All SOAP API calls are:

- Service Requests and Responses - your client application prepares and submits a service request to the server via the API, the agiloft.com Web Service processes the request and returns a response, and the client application handles the response.
- Synchronous - once the API call is invoked, your client application waits until it receives a response from the service. Asynchronous calls are not supported.
- Committed Automatically - every operation that writes to an agiloft.com object is committed automatically. This is analogous to the AUTOCOMMIT setting in SQL. Each API call is done in one transaction.
- Followed by a delay inserted after the operation has completed. The delay is set to one second by default and is configurable via the global variable *WSDelay* in the Admin Console.

An operation on a record may invoke rules and other functions, which need to be allowed enough time and resources to complete. Additionally, the delay is needed because client applications have mistakenly used web services, resulting in a flood of requests.

Factors that Determine Data Access

When using the API, the following factors determine access to your organization's data:

- Your Knowledgebase has to have SOAP API access enabled.
- The generated WSDL lists all data structures and fields that existed in the knowledgebase at the time of generation. The generated WSDL file contains all of the objects that are available in your knowledgebase. Via the API, a client application can access objects that are defined in your WSDL file.
- The generated WSDL is user-agnostic, but when invoking a method in the SOAP Interface with credentials of a certain user - the runtime permissions of that user come into play. This may result in some fields appearing empty when the user doesn't have read permission, for example.

Run-time access via the SOAP API is governed by the same access permissions as the GUI. Certain operations on the tables and fields can be performed only if the combined permissions in the logged-in user's group list permit such access.

For example, values for the fields that are not visible to a given user are not returned in the results of the query. Another often reported issue is when one is not able to write a structure, retrieved in the very previous call, back to the server simply because the user has the necessary read permissions, but not the write ones.

Please refer to the Groups section in your User Manual or Online Help for full details regarding permissions.

- As such, the API respects table-level permissions configured for the given user group in the knowledgebase.
- Further, the API fully respects the ownership concept and related table-level and field-level permissions configured in the knowledgebase.
- Ownership for a record is determined in runtime at the time of the call and if modified in the call itself, comes into effect immediately upon the completion of the API call.
- Ownership changes to one object instance do not normally automatically cascade to other object instances unless specifically configured so via a Linked Field with changes propagation. For example, if ownership changes for a given Account, ownership does not then automatically change for any Contract associated with that Account - each ownership change must be made separately and explicitly by the client application.
- The configuration of the workflow for a given knowledgebase may influence the ability to create or delete a record in a certain state or to change the state of a record.

- Business rules (especially validation ones) may affect the success of create and modification operations.
- Success also depends on whether a particular change would compromise the referential integrity of your knowledgebase. For example, the data used in Linked Fields sets is validated in create and update calls. In a similar fashion the delete calls are handled in a way to ensure the integrity of the data is maintained and will fail otherwise.
- Certain features that affect the agiloft.com user interface are not accessible or implicitly enforced via the API. For example:
 - Layouts define whether a given field is shown or hidden, but the API does not enforce such layout-specific field restrictions or validations in create and update calls. It is up to the client application to enforce any such constraints, if applicable.
 - Record types can control which layouts users with different profiles can see. However, via API all fields available for the table are presented.
 - Dependent choices and conditional visibility are not enforced by the API.

If any such constraints are required, it is up to the logic in the client application to enforce them explicitly.

Error Handling

In the event of an error the API calls return SOAP fault messages with additional information.

There are currently six types of faults distinguished by the SOAP API:

- `EWIntegrityException`

Signals that the knowledgebase setup has become incompatible with the client application logic/expectations.

- `EWOperationException`

Signals that the operation may not be performed because of some dependencies between agiloft.com functions. This message should be considered in the context of the operation. It may signal a permanent condition (e.g. workflow forbidding transition from one state to another) or a temporary one (e.g. the record is locked by another user from another GUI or API session).

- `EWPermissionException`

Signals that the username used to trigger the API call lacks sufficient privileges to perform certain operations.

- EWSessionException

Signals that client session has expired or has been removed.

- EWUnexpectedException

Signals that an unknown, not handled and unexpected exception has happened on the server side. These exceptions should be reported to the vendor for further investigation. The message contains a token that helps to trace the root cause of the problem.

- EWWrongDataException

Signals that data passed by client is wrong in the context of the operation.

SOAP API Security

Authentication

The client application must log in using valid credentials for a knowledgebase. Agiloft verifies these credentials and if valid creates an HTTP session that the client can use in subsequent calls.

Statefulness

Though stateless in appearance Agiloft Web Services are stateful by nature. A certain set of information can be shared as a "session" between multiple calls and there is usually no good reason to re-create it on every call.

A typical integration scenario obeys the pattern of "login, do multiple calls, logout".

The statefulness is supported via a widely used pattern of obtaining a session token first via the EWLogin method and explicitly passing it with each subsequent call performed.

Once the session is finished it is advisable to destroy the session with a call to EWLogout to ensure proper freeing of server resources.

Failed Logins

The SOAP API handles a series of failed logins in the same fashion as the GUI i.e. it will produce an error if the account becomes blocked due to multiple unsuccessful login attempts.

SSL Support

Agiloft SOAP API relies on the underlying application server and front-end web server (if present) for SSL support.

EWLogin

Syntax

```
String sessionId = ew.EWLogin(String KB, String user, String password, String language);
```

Usage

Use the EWLogin call to log in to Agiloft and start a client session. A client application must log in and obtain the session token (sessionId) before making any other API calls.

When the client application invokes the EWLogin call, it passes the knowledgebase name (note: not the label but the name), the username and the password as user credentials. The language code is used in localization of error messages (specify "en" for English if not sure).

After logging in the client has to pass the obtained session token with every subsequent call as a parameter.

Example Task

Log into MyKB knowledgebase as user A with password "password" and English as the local language.

The task is completed by performing the following steps:

1. Obtain the Service SOAP stub.
2. Call EWLogin.
3. Handle results.

Sample Code - Java

```
public String login() {  
    EWServiceAPI binding = new EWServiceAPIServiceLocator().getMyKB();  
    String sessionId = binding.EWLogin("MyKB", "A", "password", "en");  
    return sessionId;  
}
```

You can generate a sample Web Services code for any table by selecting Setup/Table/<Edit Table>/API/Download Sample.

Arguments

Name	
KBName	String
user	String
password	String
language	String

Response

The session token to be used in all subsequent calls.

Faults

EWPermissionException - user lacks the sufficient privileges to log in.

EWWrongDataException - the parameters supplied are not valid to log in.

EWUnexpectedException - an unexpected exception has happened; user should report this for investigation.

EWLogout

Syntax

```
void ew.EWLogout(String sessionId);
```

Usage

Use EWLogout call to explicitly indicate the end of the session to facilitate freeing up of the server resources.

Generally, client applications are not required to logout. Sessions will expire automatically after a predetermined length of inactivity, which can be configured via the global variable "disconnect_timeout". However, the explicit logout ensures that the server resources are freed up faster which aids scalability.

It is advisable to enclose the EWLogout call in the Java finally block (or equivalent in other languages).

Sample Code - Java

```
public String login() {
    EWServiceAPI binding = new EWServiceAPIServiceLocator().getMyKB();
    String sessionId = binding.EWLogin("MyKB", "A", "password", "en");
    try {
        doSomeOperations(binding, sessionId);
    } finally {
        binding.EWLogout(sessionId);
    }
}
```

You can generate a sample Web Services code for any table by selecting Setup/Table/<Edit Table>/API/Download Sample.

Arguments

Name	
sessionId	String

Faults

EWUnexpectedException - an unexpected exception has happened; user should report this for investigation.

EWRead

Retrieves the record data from your knowledgebase.

Syntax

```
EWWSBaseUserObject o = ew.EWRead(String sessionId, String
    tableName, long id);
```

or for the table-specific call:

```
WSCase wsCase = ew.EWRead_WSCase(String sessionId, long id);
```

Usage

Use the EWRead call to retrieve individual records from the table. The client application passes the tableName and an identifier of the record to be read in the generic call; and just the record identifier in the table-specific call.

Rules and Guidelines

When creating records, consider the following rules and guidelines:

- The username that was used to obtain the specified session token must have sufficient access rights to read individual records within the specified table. Please verify specific permissions via Setup/Access/Manage Groups/Edit/Table/Select Table/Permissions.
- Agiloft allows specifying fine-grained access permissions on the field level. The username that was used to obtain the specified session token must have sufficient access rights to be able to read field content. Please verify specific permissions via Setup/Access/Manage Groups/Edit/Table/Select Table/Field Permissions.
- This call does not return records that have been deleted.
- This call never returns null when the record is not found (throws a fault instead).
- In general, you use EWRead when you know in advance the identifiers of the records to retrieve. The client application may use the EWSelectFromTable call to obtain record identifiers beforehand or take the identifiers from the id field of the data structures and the linking classes.
- Client applications can use EWRead to perform a client-side join. For example, a client application can run a query to obtain a set of opportunity records, iterate through the returned opportunity records, obtain the accountId for each opportunity, and then call EWRead to obtain account information for those accountIds.

Linked Fields

Values for the fields that are imported into the target table from the donor table as a part of a Linked Fields set are available via special linking classes.

In WSDL a Linked Fields set takes form of a `DAO_Dao3_Link<N>` field in the complex data structure that corresponds to the target table, where N is a sequential number assigned automatically at the time of the set creation (e.g. `WSCase.DAO_Dao3_Link3`).

As a value, such fields can take one or more `WS<Table1><Table2>_Dao3_Link<N>` data structures; linking classes, where Table1 is the target table and Table2 is the donor table of the Linked Fields relationship (e.g. `WSCaseTeams_Dao3_Link3`).

Unfortunately, at this moment one has to rely on investigating the actual sets of fields inside the classes to trace the fields (visible in the Field Wizard in the GUI) back to the main object property (not visible in the GUI).

Non-source values for Linked Field sets that allow them are present directly in the table itself additionally to those in the Linking Classes (if the link was in fact forged).

Choice Fields

The values for choice columns are returned as instance(s) of the enumerated types described in the WSDL.

The original text values have undergone the following transformations:

1. Spaces replaced by "_"
2. Dashes replaced by "_MINUS_"
3. Pluses replaced by "_PLUS_"
4. Prefixed with "OPTION_"
5. Converted to upper case

One has to perform the reverse transformation to get to the text value.

Unsupported Types of Fields

Related tables and embedded search results are not supported by the SOAP interface.

Basic Steps for Reading Records

Reading records involves the following basic steps:

1. Determine the id of the record you want to update. You may want to use the EWSelectFromTable call to get the identifiers of the records based on some search condition or get the identifier from a previous EWCreate call, or get the id of a linked record from the linking class after performing another EWRead or EWUpdate.
2. Call EWRead.
3. Process the results.

Example Task

In MyKB knowledgebase as user A, read case #456 and return the value from the "date_created" field.

This task is completed by performing the following steps:

1. Login to MyKB with "A" and "password" and English as the local language.
2. Read record 456 from table Case.
3. Get date_created value.
4. Logout.

Sample Code - Java

```
public Date read() throws Exception {
    EWServiceAPI binding = new EWServiceAPIServiceLocator().getDemo();
    try {
        String sessionId = binding.EWLogin("MyKB", "A", "password", "en");
        WSCase wsCase = (WSCase) binding.EWRead(sessionId, "case", 456);
        return wsCase.getDate_Created();
    } finally {
        binding.EWLogout(sessionId);
    }
}
```

You can generate a sample Web Services code for any table by selecting Setup/Table/<Edit Table>/API/Download Sample.

Arguments

Name	
sessionId	String
tableName	String

id	long
----	------

Response

The record data as a descendant of the EWWSBaseUserObject - a complex structure described in WSDL.

Faults

EWSessionException - client not logged in or the session has expired; client should re-login.

EWPermissionException - user used to create the session lacks the sufficient privileges to read the record.

EWWrongDataException - client has supplied the wrong data (id cannot be found).

EWOperationException - the operation has been blocked by one of Agiloft functionalities (e.g. table-level lock).

EWIntegrityException - the specified table cannot be found or its primary key cannot be identified.

EWUnexpectedException - an unexpected exception has happened; user should report this for investigation.

EWSelectFromTable

Executes a query against the specified table and returns record identifiers that match the specified criteria.

Syntax

```
long ids[] = ew.EWSelectFromTable(String sessionId, String
    tableName, String whereClause);
```

Usage

Use the EWSelectFromTable call to search for records in the specified table.

The SQL query supplied should only include the "where" clause from a construct similar to:

*select primary-key-field-name from table-name **where** ...*

Rules and Guidelines

When querying records, consider the following rules and guidelines:

- The queries operate on the database level and should be written in terms of names visible in the Agiloft GUI in the table wizard in the dbname column. Also see EWGetChoiceLineId for special handling of choice values.
- This method never returns null. In the case where no records are found an empty array is returned.
- *Special note on memory management:* As WS integration implies pass-by-value semantics, the memory allocated for the resulting array will be released once the data is sent to the client and the server-side JVM's garbage collector considers it eligible for discarding.
- If the client-side environment is one with a garbage collector, the memory used by client-side array will be cleared once the client-side garbage collector considers it eligible.
- When the client environment uses explicit memory management, the client is responsible for freeing up the used memory explicitly.
- It is possible to limit the number of records returned through means available to the underlying database used by your instance of Agiloft e.g. "limit 0,200" for MySQL will limit the number of the returned records to 200).
- It is not possible currently to influence the sequence in which records are returned. They are returned in the effectively arbitrary default order returned by the underlying database.

Basic Steps for Selecting Records

1. Compose the whereClause. Note: "prepared" statements are not supported, all parameters have to be passed inside the whereClause string.
2. Perform the call.
3. Handle results, specifically the situations where there are no elements, one element, or more than one element in the returned array.

Example Task

In MyKB knowledgebase, as user A, find all cases assigned to John Doe.

The task is completed by performing the following steps:

1. Login to MyKB with "A" and "password", English as the local language.
2. Search for cases assigned to John Doe.
3. Logout.

Sample Code - Java

```
public long[] select() throws Exception {
    EWServiceAPI binding = new EWServiceAPIServiceLocator().getDemo();
    try {
        String sessionId = binding.EWLogin("MyKB", "A", "password", "en");
        long[] ids = binding.EWSelectFromTable(sessionId, "case",
            "assignee='John Doe'");
        return ids;
    } finally {
        binding.EWLogout(sessionId);
    }
}
```

You can generate a sample Web Services code for any table by selecting Setup/Table/<Edit Table>/API/Download Sample.

Arguments

Name	
sessionId	String
tableName	String
where	string

Response

The identifier(s) of the record(s) that match the specified criteria.

Faults

EWSessionException - client not logged in or the session has expired; client should re-login.

EWPermissionException - user used to create the session lacks the sufficient privileges to run the query.

EWWrongDataException - client has supplied the wrong data.

EWOperationException - the operation has been blocked by one of Agiloft functionalities (e.g. table-level lock).

EWIntegrityException - the specified table cannot be found or its primary key cannot be identified.

EWUnexpectedException - an unexpected exception has happened; user should report this for investigation.

EWCreate

Adds a record to your knowledgebase data.

Syntax

```
long id = ew.EWCreate(String sessionId, String table,  
    EWWSBaseUserObject object)
```

or for the table-specific call:

```
long id = ew.EWCreate_WSCase(String sessionId, WSCase object)
```

Usage

Use EWCreate to add records to the tables, such as Case or Contact, in your knowledgebase.

The EWCreate call is analogous to the INSERT statement in SQL.

Rules and Guidelines

When creating records, consider the following rules and guidelines:

- The username that was used to obtain the specified session token must have sufficient access rights to create individual records within the specified table. Please verify specific permissions via Setup/Access/Manage Groups/Edit/Table/Select Table/Permissions.
- Agiloft allows specifying fine-grained access permissions on the table field level. The username that was used to obtain the specified session token must have sufficient access rights to set every field being specified in the data object supplied in the call when creating the record. Consider this if you are using record data obtained from or just used in another call like EWRead or EWUpdate - read and update permissions of a particular user may not match the created ones. Please verify specific permissions via Setup/Access/Manage Groups/Edit/Table/Select Table/Field Permissions.
- The identifier of the new record is calculated automatically unless the field-level permissions allow the client to supply the pre-calculated one and the client supplied one.
- Certain fields can be defined to have default values. If permissions allow, these may be overwritten by data supplied in the call.
- In the context of overwriting the default values the API and WSDL distinguish between an empty (null) value set explicitly (i.e. the one that should overwrite the default) and a value not set at all in the EWCreate call (i.e. where default should remain as is).

- For required fields that do not have a pre-configured default value, you must supply a value. For all other fields in the record that do not have pre-configured default value, if you do not explicitly specify a value, then its value is empty (null).
- Agiloft allows one to establish relationships between the records in different tables and ensures the data integrity once the links are forged (see Linked Fields below).
- When creating the records in the table it is required to explicitly specify the type of the record. This is especially important in the case when the record is created for the true subtype, such as Contacts.Customer. In this case the table specified in the call will be "Contacts" and the type of the record will be "Customer". For the top-level subtypes the type of the record is equal to the table name. Please consult the Table Wizard for the specific names of the tables/subtypes in your KB, available via Setup/Tables/Your Table/Edit.
- Fields that are present in the table directly are filled in as simple values.
- Some environments (like .NET) require a special property set for simple-type fields to properly handle empty values (.<name>Specified = true for .NET).
- Fields from the Linked Fields relationships that allow values not present in the donor tables are present both in the table (for the case when the value is non-source) and in the linking classes (for the case when the value is truly imported and the link is forged).

Linked Fields

Fields that are imported into the table from another table as a part of a Linked Fields set have to be filled via special linking classes.

To forge a link the client has to create one or more instances of the linking class and select one of 3 options:

1. Specify the identifier previously obtained through the call to EWSelectFromTable, EWCreate or EWUpdate in the "id" field of the linking class. Please note: some environments (like .NET) require a special property set for simple-type fields to properly handle empty values (.<name>Specified = true for .NET).
2. Supply example data in the fields of the linking class that will be used to run Query-by-Example.
3. Supply the SQL statement similar in form to the one used for EWSelectFromTable in the special field "searchSQL". Such SQL is evaluated prior to the creation of the record to identify the actual value for the linked record identifier(s). This evaluation is done in exactly the same manner as when selecting from the table. Normally this method would be used if the query is indirect, such as "where full_name like 'John%'".

See examples of all of the above in the sample code below.

If any of the query mechanisms are used to identify the related records one has to take into account if the relationship allows multiple records to be linked. If the query

returns multiple results and the relationship doesn't allow multiple records to be linked an EWOperationException is thrown.

Choice Fields

The values for choice columns should be supplied as instance(s) of the enumerated types described in the WSDL.

Basic Steps for Creating Records

Creating records involves the following basic steps:

1. Instantiate the concrete subclass of EWWSBaseUserObject such as WSCase or WSContact. Populate its fields with the data that you want to add.
2. Call create, passing in the table name and the data map.
3. Process the results.

Example Task

In MyKB knowledgebase as user A, create a case describing a lost present, with High priority, linked to contact #125, owned by a Support Person and assigned to a Support Team.

The task is completed by performing the following steps:

1. Login to MyKB with "A" and "password", English as the locale language.
2. Create a new data structure, fill in the fields.
3. Specify a customer link to the Contacts record 124.
4. Specify the assignee Support Team from the Team table.
5. Specify an ownership link to the Contact record identified by the full name "Support Person".
6. Create the record.
7. Logout.

Sample Code - Java

```
public long create() {
    EWServiceAPI binding = new EWServiceAPIServiceLocator().getMyKB();
    String sessionId = binding.EWLogin("MyKB", "A", "password", "en");
    WSCase wsCase = new WSCase();
    wsCase.setSummary("A case of a lost present");
    wsCase.setProblem_description("People keep losing things"+
    +" bought for their dearest.");
}
```

```

WSCaseContacts_Dao3_Link5 contact = new WSCaseContacts_Dao3_Link5();
contact.setId(125L); // forge link by specifying the identifier
wsCase.setDAO_Dao3_Link5(contact);
WSCaseTeams_Dao3_Link3 assignedTo = new WSCaseTeams_Dao3_Link3();
assignedTo.setAssigned_To("Support Team"); // forge link by QBE
wsCase.setDAO_Dao3_Link3(assignedTo);
wsCase.setPriority(WSCasePriority.OPTION_HIGH);
WSCaseContacts_Dao3_Link7 ownedBy = new WSCaseContacts_Dao3_Link7();
ownedBy.setSearchSQL("full_name='Support Person'"); // forge link by SQL
wsCase.setDAO_Dao3_Link7(ownedBy);
wsCase.setType("case");
final long id = binding.EWCreate(sessionId, "case", wsCase);
binding.EWLogout(sessionId);
return id;
}

```

You can generate sample Web Services codes for any table by selecting *Setup/Table/<Edit Table>/API/Download Sample*.

Arguments

Name	Type	Description
sessionId	String	Session token
tableName	String	The name of the table where record is to be created (or generic methods).
obj	EWWSBaseUserObject or a specific descendent like WSCase or WSContact for table-specific calls.	The descendant of EWWSBaseUserObject - or complex types described in that correspond to the table Agiloft side

Response

The identifier of the record created.

Faults

EWSessionException - client not logged in or session has expired; client should re-login.

EWPermissionException - username used to create the session lacks sufficient privileges to perform record modification.

EWWrongDataException - client has supplied the wrong data.

EWOperationException - creation operation has been blocked by one of Agiloft functionalities.

EWIntegrityException - specified table cannot be found or its primary key cannot be identified.

EWUnexpectedException - an unexpected exception has happened; user should report this for investigation.

EWUpdate

Updates a record in your Agiloft knowledgebase.

Syntax

```
EWWSBaseUserObject res = ew.EWUpdate(String sessionId, String
    table,
    EWWSBaseUserObject obj);
```

or for the table-specific call:

```
WSCase wsCase = ew.EWUpdate_WSCase(String sessionId, WSCase case);
```

Usage

Use EWUpdate to update records in a table, such as Case or Contact, in your knowledgebase. The client application passes the tableName and the map with data to be applied to the record in the generic call and just the map in the table-specific call.

The EWUpdate call is analogous to the UPDATE statement in SQL.

Rules and Guidelines

When updating records, consider the following rules and guidelines:

- The username that was used to obtain the specified session token must have sufficient access rights to update individual records within the specified table. Please verify specific permissions via Setup/Access/Manage Groups/Edit Group/Table/Select Table/Permissions.
- Agiloft allows specifying fine-grained access permissions on the table field level. The username that was used to obtain the specified session token must have sufficient access rights to set every field being specified in the data object supplied in the call on update of the record. Consider this if you are using the record data obtained from or just used in another call like EWRead or EWCreate - read and create permissions of a particular user may not match the updated ones. Please verify specific permissions via Setup/Access/Manage Groups/Edit Group/Table/Select Table/Field Permissions.
- The data object must contain the identifier of the record being modified.
- When updating the records in the table it is required to specify explicitly the type of the record. This is especially important in the case when the record is created for the true subtype, such as Contacts.Customer. In this case the table specified in the call will be "Contacts" and the type of the record will be "Customer". For the top-level subtypes the type of the record is equal to the table name. Please consult the Table

Wizard for the specific names of the tables/subtypes in your knowledgebase, available via Setup/Tables/Select Table/Edit.

- Certain fields can be defined to have default values. If permissions allow these may be overwritten by data supplied in the call.
- A record created by another user (via API or via GUI) may have some required fields not filled as the user didn't have adequate privileges. Any required fields which do not have a value at the time of the call or a pre-configured default value must have a value supplied if they fall within the access privileges of the logged in user that is triggering the EWUpdate call.
- The API and WSDL distinguish between an empty (null) value set explicitly and a value not set at all in the EWUpdate call (i.e. the one that should remain unchanged).
- Fields that are present in the table directly are filled in as simple values.
- Some environments (like .NET) require a special property set for simple-type fields to properly handle empty values (.<name>Specified = true for .NET).
- Agiloft allows one to establish relationships between the records in different tables via Linked Fields and ensures the data integrity once the links are forged.
- Fields from the Linked Fields relationships that allow values not present in the donor tables are present both in the table (for the case when the value is non-source) and the linking classes (for the case when the value is truly imported and the link is forged).

Unsupported Types of Fields

Related tables and embedded search results are not supported by the SOAP interface.

Basic Steps for Updating Records

Updating records involves the following basic steps:

1. Determine the id of the record you want to update. You may want to use the EWSelectFromTable call to get the identifiers of the records based on some search condition or get the identifier from a previous EWCreate call, or get the id of a linked record from the linking class after performing an EWRead.
2. If you do not have it already from one of the previous calls, construct an instance of the complex type that corresponds to the table you are trying to update. Fill in the identifier and those fields that need updating.
3. Call EWUpdate.
4. Process the results.

Example Task

In MyKB knowledgebase, as user A, update a given case, setting the field Escalate to Support Staff to Yes (in this particular KB this triggers a round-robin assignment to the members of the Support Team). Return the name of the employee who is assigned to the case as a result.

This task is completed by performing the following steps:

1. Login to MyKB with "A" and "password" and English as the local language.
2. Set the Escalate to Support Staff field to Yes.
3. Update the record.
4. Read the name of the assignee.
5. Logout.

Sample Code - Java

```
public String update(WSCase wsCase) {
    EWServiceAPI binding = new EWServiceAPIServiceLocator().getMyKB();
    String sessionId = binding.EWLogin("MyKB", "A", "password", "en");
    wsCase.setEscalate_To_Support_Staff(WSCase.WSChoice_Yes_No.OPTION_YES);
    WSCase result = (WSCase) binding.EWUpdate(sessionId, "case", wsCase);
    WSCaseTeams_Dao3_Link3 assigneeLink = result.getDAO_Dao3_Link3();
    String assignee;
    if (assigneeLink!=null) {
        assignee = assigneeLink.getAssigned_To();
    } else {
        assignee = null;
    }
    binding.EWLogout(sessionId);
    return assignee;
}
```

You can generate a sample Web Services code for any table by selecting Setup/Table/<Edit Table>/API/Download Sample.

Arguments

Name	
sessionId	String
tableName	String

obj	EWWSBaseUserObject
-----	--------------------

Response

Updated record data as a descendant of EWWSBaseUserObject - a complex structures of the same type as the object argument.

Faults

EWSessionException - client not logged in or the session has expired; client should re-login.

EWPermissionException - username used to create the session lacks the sufficient privileges to perform record modification.

EWWrongDataException - data causes an index or constraint violation; key, name of problematic column and the value causing the problem as parameters.

EWOperationException - modification operation has been blocked by one of Agiloft functionalities.

EWIntegrityException - the specified table cannot be found or its primary key cannot be identified.

EWUnexpectedException - an unexpected exception has happened; user should report this for investigation.

EWDelete

Deletes one or more individual records from your knowledgebase data.

Syntax

```
ew.EWDelete(String sessionId, String tableName, long[] keys,  
            DeleteRule deleteRule, long[] replacementKeys);
```

Usage

Use EWDelete to delete one or more existing records from any given table.

The operation accepts a list of identifiers for the records to be deleted in one transaction. The transaction will either succeed if each individual record has been successfully deleted or a rollback is performed otherwise.

The EWDelete call is analogous to the DELETE statement in SQL.

Rules and Guidelines

When deleting records, consider the following rules and guidelines:

- The username that was used to obtain the specified session token must have sufficient access rights to delete individual records within the specified table.
- Agiloft allows one to establish relationships between the records in different tables and ensures the data integrity once the links are forged.

In the presence of such relationships when the delete operation is performed via GUI, the user who triggers the operation would be prompted with a series of dialogs to resolve the conflicts in each case.

When the same operation is triggered programmatically it is necessary to specify the policy to be applied for conflict resolution beforehand via a deleteRule argument.

This is achieved via a deleteRule parameter that can take one of the four values:

- ERROR_IF_DEPENDANTS - fails when there are any dependents,
- APPLY_DELETE_WHERE_POSSIBLE - tries to delete all dependent records, when delete cannot be done an attempt to unlink the record is made,
- DELETE_WHERE_POSSIBLE_OTHERWISE_UNLINK - same as above,
- APPLY_UNLINK - tries to unlink dependent records,

- UNLINK_WHERE_POSSIBLE_OTHERWISE_DELETE - tries to unlink all dependant records, when unlink cannot be done an attempt to delete the record is made,
- REPLACE_WITH_ANOTHER - tries to link dependent records to the substitute one specified in the replacementKeys.

Normally the decision which policy to use is taken at the "design" time and depends on the structure of the knowledgebase.

Note: If the specific strategy fails the error message returned will suggest alternatives to be used. One may use REST interface to find the best strategy.

Note: If APPLY_DELETE_WHERE_POSSIBLE or DELETE_WHERE_POSSIBLE_OTHERWISE_UNLINK strategies are used and the configuration of the knowledgebase allows it, a special "Fast Delete" algorithm is used, the same as via the GUI.

- Certain objects cannot be deleted via the API.

Basic Steps for Deleting Records

Deleting records involves the following basic steps:

1. Determine the id of each record that you want to delete. For example, you might call EWSelectFromTable to retrieve the identifiers for the set of records that you want to delete based on a specific criteria.
2. Construct the keys[] array and populate it with the identifiers of each record that you want to delete.
3. If the REPLACE_WITH_ANOTHER delete policy is used construct the substitute replacementKeys[] array.
4. Call EWDelete, passing in the table name, the keys[] array, the policy identifier and the substitute keys array (if necessary).
5. Process the results.

Example Task

In MyKB knowledgebase as user A delete case records #1234, #556, #123456 performing a "cascade" delete for all dependent records.

The task is completed by performing the following steps:

1. Login to MyKB with "A" and "password", English as the locale language.
2. Construct the keys array.

3. Call EWDelete with for case table with these keys and APPLY_DELETE_WHERE_POSSIBLE as delete policy.
4. Logout.

Sample Code - Java

```
public void delete() {
    long[] keys = new long[] { 1234L, 556L, 12346L };
    try {
        EWServiceAPI binding = new EWServiceAPIServiceLocator().getMyKB();
        String sessionId = binding.EWLogin("MyKB", "A", "password", "en");
        binding.EWDelete(sessionId, "case", keys,
            DeleteRule.APPLY_DELETE_WHERE_POSSIBLE, null);
    } catch ( EWWrongDataException e) {
        System.out.println(" EWWrongDataException encountered:\n\n" +
            e.getMessage());
    } catch ( EWOperationException e) {
        System.out.println(" EWOperationException encountered:\n\n" +
            e.getMessage());
    }
}
```

You can generate a sample Web Services code for any table by selecting Setup/Table/<Edit Table>/API/Download Sample.

Arguments

Name	
sessionId	String
tableName	String
keys	long[]
deleteRule	DeleteRule

replacementKeys	long[]
-----------------	--------

Response

The call does not return any value. A transaction is either completed as a whole or a rollback is performed.

Faults

EWSessionException - client not logged in or the session has expired; client should re-login.

EWPermissionException - user used to create the session lacks sufficient privileges to perform a record deletion.

EWWrongDataException - supplied data is wrong; caused by either the record to delete cannot be found (cause, array index and key of the problematic record returned as parameters) or replacement failed (cause, array index, key of the problematic record and key of the replacement one as parameters).

EWOperationException - delete operation cannot be done (cause, array index and key of the problematic record returned as parameters); either the record has dependents (number of dependents as parameter); or the delete and the unlink are both not possible (relationship description as parameter); or the unlink is not possible (relationship description as parameter); or the replacement is not possible (relationship description as parameter); or the delete has failed (cause, array index, key of problematic record as parameters).

EWIntegrityException - specified table cannot be found or its primary key cannot be identified.

EWUnexpectedException - an unexpected exception has happened; user should report this for investigation.

EWGetChoiceLineId

Obtains the internal identifier that corresponds to a choice value for use in SQL-based expressions.

Syntax

```
long id = ew.EWGetChoiceLineId(String sessionId, String
    tableName,
    String fieldName, String value);
```

Usage

Use EWGetChoiceLineId to obtain the internal identifier that corresponds to the choice text value for further use in SQL-based expressions such as the where parameter of EWSelectFromTable or the value of searchSQL property when forging links to other tables via Linked Field sets.

SQL-based expressions are evaluated on the database level where choice values are stored as identifiers. These identifiers change if the knowledgebase is copied or re-imported, or the values in the choice list are re-created. It is therefore advisable to use the EWGetChoiceLineId call to obtain the id value at runtime immediately before the SQL call.

Rules and Guidelines

When querying for choice value identifiers, consider the following rules and guidelines:

- The value parameter is intentionally specified as String rather than an enumerated WSDL type. This is to give client applications more flexibility.

If the client needs to lookup the identifier of the choice value enumerated in WSDL, the client should convert the enumerated value into String. Since enumerated WSDL types that correspond to the choice values are based on Strings, this should always be possible although specific details depend on the client Web Services environment used.

When converting one of the WSChoice_ values back to String, please note that the original text values have undergone the following transformations:

1. Spaces replaced by "_"
2. Dashes replaced by "_MINUS_"
3. Pluses replaced by "_PLUS_"
4. Prefixed with "OPTION_"

5. Converted to upper case

One has to perform the reverse transformation to get to the text value.

- If the factors that trigger the change of the internal choice identifiers (as listed above: when knowledgebase is copied, re-imported or the values in the choice list are re-created) are somehow detected on the client, then the values returned by this call can be safely cached on a per KB basis between multiple calls.

Example Task

In MyKB knowledgebase as user A, find all cases with High priority.

The task is completed by performing the following steps:

1. Login to MyKB with "A" and "password", English as the locale language.
2. Obtain the identifier for the choice value High for the field Priority in the table Case.
3. Run the SQL query to obtain the identifiers.
4. Logout.

Sample Code - Java

```
public long[] selectCasesPriorityHigh() throws Exception {
    EWServiceAPI binding = new
EWServiceAPIServiceLocator().getMyKB();
    String sessionId = binding.EWLogin("MyKB", "A", "password", "en");
    try {
        long high = binding.EWGetChoiceLineId(sessionId, "case", "priority",
"High");
        long ids[] = binding.EWSelectFromTable(sessionId, "case",
"priority="+high);
        return ids;
    } catch (EWSessionException e) {
        // Normally the client should just re-login and re-try the call.
        throw e;
    } catch (EWWrongDataException e) {
        // Since the field name and the choice value being looked up are most
likely
        // hard-coded this is really a show-stopper that should be reported to
the
        // developer/implementer
        throw e;
    } catch (EWIntegrityException e) {
        // Since the table name is hard-coded this is a show-stopper that
should be
        // reported to the developer/implementer
        throw e;
    } catch (EWUnexpectedException e) {
        // This exception should be reported to the vendor.
        // The message contains a token that will help to trace the root cause

```



```

of
// the problem.
throw e;
} finally {
binding.EWLogout(sessionId);
}
}

```

You can generate a sample Web Services code for any table by selecting Setup/Table/<Edit Table>/API/Download Sample.

Arguments

Name	
sessionId	String
tableName	String
fieldName	String
value	String

Response

The identifier that corresponds to the choice text value.

Faults

EWSessionException - client not logged in or the session has expired; client should re-login.

EWWrongDataException - the specified field does not exist in the table or the choice value passed cannot be found.

EWIntegrityException - the specified table cannot be found or its primary key cannot be identified.

EWUnexpectedException - an unexpected exception has happened; user should report this for investigation.

EWAttachFromSOAPAttachment

Attaches a file into a field of the table record in your knowledgebase.

Syntax

```
int n = ew.EWAttachFromSOAPAttachment(String sessionId, String  
tableName, long id, String fieldName, String fileName);
```

Usage

Use the EWAttachFromSOAPAttachment call to attach a file passed as a SOAP attachment into a file or image field in the table using the specified file name.

Rules and Guidelines

When attaching files, consider the following rules and guidelines:

- The username that was used to obtain the specified session token must have sufficient access rights to modify individual records within the specified table. Please verify specific permissions via Setup/Access/Manage Groups/Edit/Table/Select Table/Permissions.
- Agiloft allows specifying fine-grained access permissions on the field level. The username that was used to obtain the specified session token must have sufficient access rights to be able to update the field content. Please verify specific permissions via Setup/Access/Manage Groups/Edit/Table/Select Table/Field Permissions.
- This call returns the current number of the attached files in the specified field.
- One can use EWRead method to obtain an array of file names for the attached fields in the specified field.
- Only one file per call is attached.
- The file name does NOT have to be unique across multiple calls against the same record and field.
- In general, you use EWAttachFromSOAPAttachment when you know in advance the identifiers of the records to retrieve. The client application may use the likes of EWSelectFromTable call to obtain record identifiers beforehand or to take the identifiers from the id field of the data structures and linking classes.

Basic Steps for Attaching Files

Attaching files to records involves the following basic steps:

1. Determine the id of the record you want to update. You may want to use the EWSelectFromTable or EWSearchTable calls to get the identifiers of the records based on

some search condition or get the identifier from a previous EWCreate call, or get the id of a linked record from the linking class after performing EWRead or EWUpdate.

2. Wrap your file on the client-side in the matter specific to your client-side SOAP environment.
3. Call EWAttachFromSOAPAttachment.
4. Process the results.

Example Task

In MyKB knowledgebase as user A, attach from the client-side filesystem file B.txt located in the current directory to the field Additional Files of case #456 and return the total number of files currently attached.

The task is completed by performing the following steps:

1. Login to MyKB with "A" and "password" and English as the local language.
2. Prepare the file for transmission.
3. Invoke attach call.
4. Get total number of files currently attached.
5. Logout.

Sample Code - Java

```
public int attach() throws Exception {
    EWServiceAPI binding = new
EWServiceAPIServiceLocator().getDemo();
    try {
        String sessionId = binding.EWLogin("MyKB", "A", "password",
"en");

        DataHandler dataHandler =
            new DataHandler(new FileDataSource("B.txt"));
        binding.addAttachment(dataHandler);
        int n = binding.EWAttachFromSOAPAttachment(sessionId, "case",
456,
            "additional_files", "B.txt");
        return n;
    } finally {
        binding.EWLogout(sessionId);
    }
}
```

You can generate sample Web Services code for any table by selecting Setup/Table/<Edit Table>/API/Download Sample

Arguments

Name	
sessionId	String
tableName	String
id	long
fieldName	String
fileName	String

The file to be attached is passed as a SOAP attachment.

Response

The current number of files attached in the specified field.

Faults

EWSessionException - client not logged in or session has expired; client should re-login.

EWPermissionException - user used to create the session lacks sufficient privileges to read the record.

EWWrongDataException - client has supplied wrong data (id cannot be found).

EWOperationException - the operation has been blocked by one of Agiloft functionalities (e.g. table-level lock).

EWIntegrityException - specified table cannot be found or its primary key cannot be identified.

EWUnexpectedException - an unexpected exception has happened; user should report this for investigation.

EWCreateAndRead

Adds a record to your knowledgebase data and reads the result.

Syntax

```
EWWSBaseUserObject o = ew.EWCreateAndRead(String sessionId, String table,
      EWWSBaseUserObject object)
```

or for the table-specific call:

```
WSCase case = ew.EWCreateAndRead_WSCase(String sessionId, WSCase object)
```

Usage

Use EWCreateAndRead to add records to the tables, such as Case or Contact, in your knowledgebase and read the result in one call, potentially retrieving values for the fields set by business rules and other backend logic.

The EWCreateAndRead call is analogous to an INSERT statement in SQL immediately followed by SELECT.

Rules and Guidelines

When creating records, consider the following rules and guidelines:

- The username that was used to obtain the specified session token must have sufficient access rights to create individual records within the specified table. Please verify specific permissions via Setup/Access/Manage Groups/Edit/Table/Select Table/Permissions.
- Agiloft allows specifying fine-grained access permissions on the table field level. The username that was used to obtain the specified session token must have sufficient access rights to set every field being specified in the data object supplied in the call when creating the record. Consider this if you are using a record data obtained from or just used in another call like EWRead or EWUpdate - read and update permissions of a particular user may not match the created ones. Please verify specific permissions via Setup/Access/Manage Groups/Edit/Table/Select Table/Field Permissions.
- The identifier of the new record is calculated automatically unless the field-level permissions allow the client to supply the pre-calculated one and the client supplied one.
- Certain fields can be defined to have default values. If permissions allow, these may be overwritten by the data supplied in the call.
- In the context of overwriting the default values the API and WSDL distinguish between an empty (null) value set explicitly (i.e. the one that should overwrite the

default) and a value not set at all in the EWCreate call (i.e. where default should remain as is).

- For required fields that do not have a pre-configured default value, you must supply a value. For all other fields in the record that do not have a pre-configured default value, if you do not explicitly specify a value, then its value is empty (null).
- Agiloft allows one to establish relationships between the records in different tables and ensures the data integrity once the links are forged (see Linked Fields below).
- When creating the records in the table it is required to explicitly specify the type of the record. This is especially important in the case when the record is created for the true subtype, such as Contacts.Customer. In this case the table specified in the call will be "Contacts" and the type of the record will be "Customer". For the top-level subtypes the type of the record is equal to the table name. Please consult the Table Wizard for the specific names of the tables/subtypes in your KB, available via Setup/Tables/Your Table/Edit.
- Fields that are present in the table directly are filled in as simple values.
- Some environments (like .NET) require a special property set for simple-type fields to properly handle empty values (.<name>Specified = true for .NET)
- Fields from the Linked Fields relationships that allow values not present in the donor tables are present both in the table (for the case when the value is non-source) and in the linking classes (for the case when the value is truly imported and the link is forged).

Linked Fields

Fields that are imported into the table from another table as a part of a Linked Fields set have to be filled via special linking classes.

In WSDL a Linked Fields set takes form of DAO_Dao3_Link<N> field in the complex data structure that corresponds to the target table, where N is a sequential number assigned automatically at the time of the set creation (e.g. WSCase.DAO_Dao3_Link3).

As a value such fields can take one or more WS<Table1><Table2>_Dao3_Link<N> data structures - linking classes, where Table1 is the target table and Table2 is the donor table of the Linked Fields relationship (e.g. WSCaseTeams_Dao3_Link3).

Unfortunately, at this moment one has to rely on investigating the actual sets of fields inside the classes to trace the fields (visible in the Field Wizard in the GUI) back to the main object property (not visible in the GUI).

Non-source values for Linked Field sets that allow them are present directly in the table itself additionally to those in the Linking Classes (if the link was in fact forged).

To forge a link the client has to create one or more instances of the linking class and select one of 3 options:

1. Specify the identifier previously obtained through the call to EWSelectFromTable, EWCreate or EWUpdate in the "id" field of the linking class. Please note: some environments (like .NET) require a special property set for simple-type fields to properly handle empty values (.<name>Specified = true for .NET).
2. Supply example data in the fields of the linking class that will be used to run a Query-by-Example.
3. Supply the SQL statement similar in form to the one used for EWSelectFromTable in the special field "searchSQL". Such SQL is evaluated prior to the creation of the record to identify the actual value for the linked record identifier(s). This evaluation is done in exactly the same manner as when selecting from the table. Normally this method would be used if the query is indirect, such as "where full_name like 'John%'".

See examples of all of the above in the sample code below.

If any of the query mechanisms are used to identify the related records, one has to take into account if the relationship allows multiple records to be linked. If the query returns multiple results and the relationship doesn't allow multiple records to be linked an EWOperationException is thrown.

Choice Fields

The values for choice columns should be supplied as instance(s) of the enumerated types described in the WSDL.

The original text values have undergone the following transformations:

1. Spaces replaced by "_"
2. Dashes replaced by "_MINUS_"
3. Pluses replaced by "_PLUS_"
4. Prefixed with "OPTION_"
5. Converted to upper case

One has to perform the reverse transformation to get to the text value.

Unsupported Types of Fields

Related tables and embedded search results are not supported by the SOAP interface.

Basic Steps for Creating and Reading Records in One Call

Creating records involves the following basic steps:

1. Instantiate the concrete subclass of EWWSBaseUserObject such as WSCase or WSContact. Populate its fields with the data that you want to add.
2. Call EWCreateAndRead, passing in the table name and the data map or the table-specific variant with the data map only.
3. Process the results.

Example Task

In MyKB knowledgebase as user A create a case describing a lost present, with High priority, linked to contact #125, owned by a Support Person and assigned to the Support Team. Return the value from "date_created" field.

The task is completed by performing the following steps:

1. Login to MyKB with "A" and "password" and English as the local language.
2. Create a new data structure, fill in the fields.
3. Specify a customer link to the Contacts record 124.
4. Specify the assignee Support Team from the Team table.
5. Specify an ownership link to the Contact record identified by the full name "Support Person".
6. Create the record.
7. Get date_created value from the returned object.
8. Logout.

Sample Code - Java

```
public Date create() {
    EWServiceAPI binding = new
EWServiceAPIServiceLocator().getMyKB();
    try {
        String sessionId = binding.EWLogin("MyKB", "A", "password",
"en");
        WSCase wsCase = new WSCase();
        wsCase.setSummary("A case of a lost present");
        wsCase.setProblem_description("People keep losing things"+
+" bought for their dearest.");
        WSCaseContacts_Dao3_Link5 contact = new
WSCaseContacts_Dao3_Link5();
        contact.setId(125L); // forge link by specifying the identifier
        wsCase.setDAO_Dao3_Link5(contact);
        WSCaseTeams_Dao3_Link3 assignedTo = new
WSCaseTeams_Dao3_Link3();
        assignedTo.setAssigned_To("Support Team"); //forge link by QBE
        wsCase.setDAO_Dao3_Link3(assignedTo);
```



```

        wsCase.setPriority(WSCheck_Priority.OPTION_HIGH);
        WSCaseContacts_Dao3_Link7 ownedBy = new
WSCaseContacts_Dao3_Link7();
        ownedBy.setSearchSQL("full_name='Support Person'"); //link by
SQL
        wsCase.setDAO_Dao3_Link7(ownedBy);
        wsCase.setType("case");
        WSCase wsCase = (WSCase) binding.EWCreateAndRead(sessionId,
"case",
                                wsCase);
        return wsCase.getDate_Created();
    } finally {
        binding.EWLogout(sessionId);
    }
}

```

You can generate sample a Web Services code for any table by selecting Setup/Table/<Edit Table>/API/Download Sample.

Arguments

Name	
sessionId	String
tableName	String
obj	EWWSBaseUserObject or WSContact for table-specific

Response

The record data as a descendant of EWWSBaseUserObject - a complex structure described in WSDL.

Faults

EWSessionException - client not logged in or the session has expired; client should re-login.

EWPermissionException - username used to create the session lacks sufficient privileges to perform the record modification.

EWWrongDataException - client has supplied the wrong data.

EWOperationException - creation operation has been blocked by one of Agiloft functionalities.

EWIntegrityException - specified table cannot be found or its primary key cannot be identified.

EWUnexpectedException - an unexpected exception has happened; user should report this for investigation.

EWRemoveAttached

Removes the attached file specified by a position in the named field of the specified table record.

Syntax

```
int n = ew.EWRemoveAttached(String sessionId, String
                             tableName, long id, String fieldName, int position);
```

Usage

Use the EWRemoveAttached call to remove an attached file from a File or Image field in a record in the table.

Rules and Guidelines

When removing attached files, consider the following rules and guidelines:

- The username that was used to obtain the specified session token must have sufficient access rights to modify individual records within the specified table. Please verify specific permissions via Setup/Access/Manage Groups/Edit Group/Table/Select Table/Permissions.
- Agiloft allows specifying fine-grained access permissions on the field level. The username that was used to obtain the specified session token must have sufficient access rights to be able to update field content. Please verify specific permissions via Setup/Access/Manage Groups /Edit Group/Table/Select Table/Field Permissions.
- This call requires the position of the attached file in the specified field. Position numbering starts from 0.
- One can use the EWRead method to obtain an array of file names for the attached field in the specified field. The sequence of file names in the array will correspond to the positions of the files.
- Only one file per call is removed.
- In general, you use EWRemoveAttached when you know in advance the IDs of the records to retrieve. The client application may use the likes of EWSelectFromTable call to obtain record identifiers beforehand or take the identifiers from the id field of the data structures and linking classes.

Basic Steps for Removing Attached Files

Removing files from records involves the following basic steps:

1. Determine the id of the record you want to update. You may want to use the EWSelectFromTable or EWSearchTable calls to get the identifier of the record based on

some search condition or get the identifier from a previous EWCreate call, or get the id of a linked record from the linking class after performing EWRead or EWUpdate.

2. Determine the position of the file in the field.
3. Call EWRemoveAttached.
4. Process the results.

Example Task

In MyKB knowledgebase, as user A, remove all files named B.txt currently attached in the field Additional Files of case #456 and return the total number of files currently attached.

The task is completed by performing the following steps:

1. Login to MyKB with "A" and "password" and English as the local language.
2. Read the record to get the list of the files.
3. Find the position of the file.
4. Invoke the remove file call.
5. Get the total number of files currently attached.
6. Logout

Sample Code - Java

```
public int remove() throws Exception {
    EWServiceAPI binding = new
EWServiceAPIServiceLocator().getDemo();
    try {
        String sessionId = binding.EWLogin("MyKB", "A", "password",
"en");
        WSCase wsCase = (WSCase) binding.EWRead(sessionId, "case",
456);
        String[] fileNames = wsCase.getAdditional_Files();
        int n = 0;
        if (fileNames!=null) {
            for (int i = 0; i < fileNames.length; i++) {
                if ("B.txt".equals(fileNames[i])) {
                    n = binding.EWRemoveAttached("case", 456,
"additional_files", i);
                }
            }
        }
        return n;
    } finally {
        binding.EWLogout(sessionId);
    }
}
```

You can generate sample Web Services code for any table by selecting Setup/Table/<Edit Table>/API/Download Sample.

Arguments

Name	
sessionId	String
tableName	String
id	long
fieldName	String
position	int

Response

The current number of files attached in the specified field.

Faults

EWSessionException - client not logged in or session has expired; client should re-login.

EWPermissionException - user used to create the session lacks sufficient privileges to read the record.

EWWrongDataException - client has supplied wrong data (id cannot be found).

EWOperationException - the operation has been blocked by one of Agiloft functionalities (e.g. table-level lock).

EWIntegrityException - specified table cannot be found or its primary key cannot be identified.

EWUnexpectedException - an unexpected exception has occurred; the admin user should report this for investigation.

EWRetrieveAttachedAsSOAPAttachment

Retrieves an attached file from the specified field of the table record in your knowledgebase.

Syntax

```
ew.EWRetrieveAttachedAsSOAPAttachment(String sessionId, String  
tableName, long id, String fieldName, int position);
```

Usage

Use the EWRetrieveAttachedAsSOAPAttachment call to retrieve an attached file from a File or Image field in the table record.

Rules and Guidelines

When retrieving attached files, consider the following rules and guidelines:

- The username that was used to obtain the specified session token must have sufficient access rights to read individual records within the specified table. Please verify specific permissions via Setup/Access/Manage Groups/Edit/Table/Select Table/Permissions.
- Agiloft allows specifying fine-grained access permissions on the field level. The username that was used to obtain the specified session token must have sufficient access rights to be able to read the field content. Please verify specific permissions via Setup/Access/Manage Groups/Edit/Table/Select Table/Field Permissions.
- This call requires the position of the attached file in the specified field. Position numbering starts from zero.
- One can use the EWRead method to obtain an array of file names for the attached fields in the specified field. The sequence of file names in the array will correspond to the positions of the files.
- Only one file per call is retrieved.
- In general, you use EWRetrieveAttachedAsSOAPAttachment when you know in advance the identifiers of the records to retrieve. The client application may use the likes of the EWSelectFromTable call to obtain record identifiers beforehand or take the identifiers from the id field of the data structures and the linking classes.

Basic Steps for Retrieving Attached Files

Retrieving files from records involves the following basic steps:

1. Determine the id of the record you want to update. You may want to use the EWSelectFromTable or EWSearchTable calls to get the identifiers of the records based on some search condition; or get the identifier from a previous EWCreate call; or get the id of a linked record from the linking class after performing EWRead or EWUpdate.
2. Determine the position of the file in the field.
3. Call EWRetrieveAttachedAsSOAPAttachment.
4. Process the results.

Example Task

In MyKB knowledgebase as user A, retrieve the first file named B.txt currently attached in the field Additional Files of case #456 in the current directory.

The task is completed by performing the following steps:

1. Login to MyKB with "A" and "password" and English as the local language.
2. Read the record to get the list of the files.
3. Find the position of the file.
4. Invoke the retrieve file call.
5. Logout.

Sample Code - Java

```
public int retrieve() throws Exception {
    EWServiceAPI binding = new
EWServiceAPIServiceLocator().getDemo();
    try {
        String sessionId = binding.EWLogin("MyKB", "A", "password",
"en");
        WSCase wsCase = (WSCase) binding.EWRead(sessionId, "case",
456);
        String[] fileNames = wsCase.getAdditional_Files();
        int n = 0;
        if (fileNames!=null) {
            for (int i = 0; i < fileNames.length; i++) {
                if ("B.txt".equals(fileNames[i])) {
                    binding.EWRetrieveAttachedAsSOAPAttachment("case", 456,
"additional_files", i);
                    Object[] attachments = binding.getAttachments();
                    AttachmentPart attachmentPart = (AttachmentPart)
attachments[0];
                    final InputStream inputStream =
                        attachmentPart.getDataHandler().getInputStream();
                    BufferedOutputStream bos =
                        new BufferedOutputStream(new FileOutputStream("B.txt"));
                    int b;
                    while ((b = inputStream.read()) != -1) bos.write(b);
                }
            }
        }
    }
}
```

```

        bos.flush();
        bos.close();
        break;
    }
}
}
return;
} finally {
binding.EWLogout(sessionId);
}
}

```

You can generate a sample Web Services code for any table by selecting Setup/Table/<Edit Table>/API/Download Sample.

Arguments

Name	
sessionId	String
tableName	String
id	long
fieldName	String
position	int

Response

The file as a SOAP attachment.

Faults

EWSessionException - client not logged in or the session has expired; client should re-login.

EWPermissionException - user used to create the session lacks the sufficient privileges to read the record.

EWWrongDataException - client has supplied the wrong data (id cannot be found).

EWOperationException - the operation has been blocked by one of Agiloft functionalities (e.g. table-level lock).

EWIntegrityException - the specified table cannot be found or its primary key cannot be identified.

EWUnexpectedException - an unexpected exception has happened; user should report this for investigation.

EWSearchTable

Executes a pre-configured named Saved Search against the specified table and returns an array of record data for the records that match the search.

Syntax

```
EWWSBaseUserObject[] os = ew.EWSearchTable(String sessionId, String  
tableName, String[] fieldNames, String searchName);
```

Usage

Use the EWSearchTable call to search for records in the specified table based on a Saved Search pre-configured in the GUI.

Rules and Guidelines

When querying records, consider the following rules and guidelines:

- The username that was used to obtain the specified session token must have sufficient access rights to read individual records within the specified table. Please verify specific permissions via Setup/Access/Manage Groups/Edit Group/Table/Select Table/Permissions.
- Agiloft allows specifying fine-grained access permissions on the field level. The username that was used to obtain the specified session token must have sufficient access rights to be able to read field content. Please verify specific permissions via Setup/Access/Manage Groups/Edit Group/Table/Select Table/Field Permissions.
- This call does not return records that have been deleted.
- This method never returns null. In the case when no records are found an empty array is returned.
- *Special note on memory management:* As WS integration implies pass-by-value semantics, the memory allocated for the resulting array will be released once the data is sent to the client and server-side JVM's garbage collector considers it eligible for discarding.
- If the client-side environment is the one with a garbage collector, the memory used by client-side array will be cleared once the client-side garbage collector considers it eligible.
- When the client environment uses explicit memory management, the client is responsible for freeing up the used memory explicitly.
- When using the EWSearchTable method only the fields explicitly listed in the call are read and passed to the client.

Linked Fields

Values for the fields that are imported into the target table from the donor table as a part of a Linked Fields set are available via special linking classes.

In WSDL a Linked Fields set takes form of a `DAO_Dao3_Link<N>` field in the complex data structure that corresponds to the target table, where N is a sequential number assigned automatically at the time of the set creation (e.g. `WSCase.DAO_Dao3_Link3`).

As a value such fields can take one or more `WS<Table1><Table2>_Dao3_Link<N>` data structures - linking classes, where Table1 is the target table and Table2 is the donor table of the Linked Fields relationship (e.g. `WSCaseTeams_Dao3_Link3`).

Unfortunately, at this moment one has to rely on investigating the actual sets of fields inside the classes to trace the fields (visible in the Field Wizard in the GUI) back to the main object property (not visible in the GUI).

Non-source values for Linked Field sets that allow them are present directly in the table itself and additionally to those in the Linking Classes (if the link was in fact forged).

Choice Fields

The values for choice columns are returned as instance(s) of the enumerated types described in the WSDL.

The original text values have undergone the following transformations:

1. Spaces replaced by "_"
2. Dashes replaced by "_MINUS_"
3. Pluses replaced by "_PLUS_"
4. Prefixed with "OPTION_"
5. Converted to Upper Case

One has to perform the reverse transformation to get to the text value.

Unsupported Types of Fields

Related tables and embedded search results are not supported by the SOAP interface.

Basic Steps for Searching Records with a Pre-configured Saved Search

1. Create a Saved Search in the GUI.

2. Perform the call using the name of the search as the last parameter.
3. Handle the results, specifically the situations where there are no elements, one element, or more than one element in the returned array.

Example Task

In MyKB knowledgebase, as user A, find all cases assigned to the user used to login. Return summaries as a String array.

The task is completed by performing the following steps:

1. Login to MyKB with "A" and "password" and English as the local language.
2. Search for cases using My Assigned search.
3. Logout.

Sample Code - Java

```
public String[] search() throws Exception {
    EWServiceAPI binding = new
EWServiceAPIServiceLocator().getDemo();
    try {
        String sessionId = binding.EWLogin("MyKB", "A", "password",
"en");
        EWWSBaseUserObject[] records = binding.EWSearchTable(sessionId,
"case",
        new String[] {"summary"}, "My Assigned");
        String[] result = new String[records.length];
        for (int i=0; i<records.length; i++) {
            result[i] = records[i].getSummary();
        }
        return result;
    } finally {
        binding.EWLogout(sessionId);
    }
}
```

You can generate a sample Web Services code for any table by selecting Setup/Table/<Edit Table>/API/Download Sample.

Arguments

Name	
sessionId	String
tableName	String

fields	String array
savedsearch	String

Response

An array of the records as descendants of EWWSBaseUserObject - a complex structure described in WSDL.

Faults

EWSessionException - client not logged in or the session has expired; client should re-login.

EWPermissionException - user used to create the session lacks the sufficient privileges to run the query.

EWWrongDataException - client has supplied the wrong data.

EWOperationException - the operation has been blocked by one of Agiloft functionalities (e.g. table-level lock).

EWIntegrityException - specified table cannot be found or its primary key cannot be identified.

EWUnexpectedException - an unexpected exception has occurred; the admin user should report this for investigation.

EWSearchTableWithQuery

Executes a pre-configured named Saved Search against the specified table and/or an ad-hoc query and returns an array of record data for the records that match the criteria.

Syntax

```
EWWSBaseUserObject[] os = ew.EWSearchTableWithQuery(String sessionId, String  
tableName, String[] fieldNames, String searchName, String  
query);
```

Usage

Use EWSearchTableWithQuery call to search for records in the specified table based on a Saved Search pre-configured in the GUI and/or to filter the records with an ad-hoc query.

Rules and Guidelines

When querying records, consider the following rules and guidelines:

- The username that was used to obtain the specified session token must have sufficient access rights to read individual records within the specified table. Please verify specific permissions via Setup/Access/Manage Groups/Edit Group/Table/Select Table/Permissions.
- Agiloft allows specifying fine-grained access permissions on the field level. The username that was used to obtain the specified session token must have sufficient access rights to be able to read field content. Please verify specific permissions via Setup/Access/Manage Groups/Edit Group/Table/Select Table/Field Permissions.
- This call does not return records that have been deleted.
- This method never returns null. In the case when no records are found an empty array is returned.
- *Special note on memory management:* As WS integration implies pass-by-value semantics, the memory allocated for the resulting array will be released once the data is sent to the client and the server-side JVM's garbage collector considers it eligible for discarding.
- If the client-side environment is the one with a garbage collector, the memory used by client-side array will be cleared once the client-side garbage collector considers it eligible,
- When the client environment uses explicit memory management, the client is responsible for freeing up the used memory explicitly.

- When using `EWSearchTableWithQuery` method only the fields explicitly listed in the call are read and passed to the client.
- The main difference from using the `EWSelectAndRead` method, which uses an SQL where clause, is that ad-hoc queries operate on a higher level, can use logical field names, are capable of recognizing choice values and high-level relationships between table fields, and can use advanced and time-based criteria.
- If the query doesn't parse according to the grammar, an attempt is made to parse the parameter value as a sequence of identifiers (using a different grammar). If both fail, the parameter value is treated as a Full-Text Search query.
- The ad-hoc query grammar is described at the end of this section.

Steps for Searching Records with a Saved Search and/or Ad-hoc Query

1. Optionally create a Saved Search in the GUI.
2. Perform the call using the name of the search and additionally filter the results with an ad-hoc query or use the ad-hoc query without the search.
3. Handle the results, specifically the situations where there are no elements, one element, or more than one element in the returned array.

Example Task

In MyKB knowledgebase, as user A, find all cases assigned to the user used to login with low priority. Return summaries as a String array.

Completion of the task is performed by the following steps:

1. Login to MyKB with "A" and "password" and English as the local language.
2. Search for cases using My Assigned search, additionally filtering by low priority.
3. Logout

Sample Code - Java

```
public String[] search() throws Exception {
    EWServiceAPI binding = new EWServiceAPIServiceLocator().getDemo();
    try {
        String sessionId = binding.EWLogin("MyKB", "A", "password", "en");
        EWWSBaseUserObject[] records = binding.EWSearchTableWithQuery(sessionId,
"case",
        new String[] {"summary"}, "My Assigned", "Priority=Low");
        String[] result = new String[records.length];
        for (int i=0; i<records.length; i++) {
            result[i] = records[i].getSummary();
        }
        return result;
    } finally {
```

```
binding.EWLogout(sessionId);
}
}
```

You can generate sample Web Services code for any table by selecting Setup/Table/<Edit Table>/API/Download Sample

Arguments

Name	
sessionId	String
tableName	String
fields	String array
savedsearch	String
query	String

Response

An array of the records as descendants of EWWSBaseUserObject - a complex structure described in WSDL.

Faults

EWSessionException - client not logged in or session has expired; client should re-login.

EWPermissionException - user used to create the session lacks sufficient privileges to run the query.

EWWrongDataException - client has supplied wrong data.

EWOperationException - the operation has been blocked by one of Agiloft functionalities (e.g. table-level lock).

EWIntegrityException - specified table cannot be found or its primary key cannot be identified.

EWUnexpectedException - an unexpected exception has happened; the admin user should report this for investigation.

Informal Grammar Description for Ad-hoc Queries

Field names are usually column labels as seen in the UI. However, DB and User column names are accepted too. Both field names and values may be surrounded by single quotes ('). If they contain spaces or some weird characters then quoting is mandatory. For example:

Priority=Low	OK
'Priority'='Low'	OK
Bug Priority=Low	Invalid
'Bug Priority'='Low'	OK
'Bug Priority'='Very Low'	Invalid
'Bug Priority'='Very Low'	OK

Simple criteria

Simple criteria has the form of `<field name><operator><value>`, where operator is one of:

=	equals
!=	not equals
~=	contains
!~=	doesn't contain
>=	greater or equals
<=	lesser or equals
>	greater
<	lesser
<<	included by
!<<	not included by

The `(not) included by` operator expects a comma-separated (without spaces!) list of values at the right-hand side and checks if field value is included (or not-included) into this list. In other words

```
Priority<<High,Low
```

Is a short-hand for

```
Priority=High || Priority=Low
```

(where `||` is OR, as described below).

Logical criteria

Allows to combine other criterias using AND and OR operators. '&&' is AND, '||' is OR.

Operator precedence

Expression is evaluated from left to right, braces may be used for grouping. For example 'A && B || C' means '(A && B) || C'.

Time-based criteria

Allows to set relative date constraints. The form is `<field name><operator><mode><value>`, where operator is one of `=, !=, <, >, <=, >=`, mode is either `'-'`, which means 'old', `'+'`, which means 'in the future' or `'#'`, which means 'absolute'. 'value' is an integer followed by a single character:

m	minute
h	hour
w	week
M	month
y	year

Examples:

Date<-1y	'Date' is less than one year old
Date>=+10m	'Date' is greater or equal than 10 minutes in the future
Duration=#2h	'Duration' is exactly two hours

Currently, more complex expressions like 'two years, one month and three hours' are not supported.

Advanced criteria

Advanced criteria has the form of `<field name>:<from>-><to>` and means 'field `field name` has changed from *from* to *to*' Either *from* or *to* (but not both at the same time) may hold '?' meaning 'any value'.

This criteria searches through record history, and thus the history column must exist and must track changes to the specified field. All simple and time-based criterias have implicit 'now' flag set, which means that they will match the current record state, not the state when advanced criteria has been satisfied. In the other words, if we have a record with the following modification history, with the bottom state being the most recent:

```
State=Open, Priority=Low
State=Closed, Priority=Low
State=Closed, Priority=High
```

Then 'State:Open->Closed && Priority=Low' will not find it, but 'State:Open->Closed && Priority=High' will.

More examples

- `Status=Open && ('Assigned To' = john || 'Assigned To' = jane)`
- `(Priority>High || Summary ~= Urgent) && State:Closed->Reopened`
- `OS=Windows,Linux && 'Modification Date' < -1y`

EWSelectAndRead

Executes a query against the specified table and returns an array of record data for the records that match the specified criteria.

Syntax

```
EWWSBaseUserObject[] os = ew.EWSelectFromTable(String sessionId, String  
        tableName, String whereClause);
```

or for the table-specific call:

```
WSCase[] cases = ew.EWCreateAndRead_WSCase(String sessionId, String  
        whereClause)
```

Usage

Use EWSelectFromTable call to search for records in the specified table.

The SQL query supplied should only include the "where" clause from a construct similar to:

*select primary-key-field-name from table-name **where** ...*

Rules and Guidelines

When querying records, consider the following rules and guidelines:

- The username that was used to obtain the specified session token must have sufficient access rights to read individual records within the specified table. Please verify specific permissions via Setup/Access/Manage Groups/Edit Group/Table/Select Table/Permissions.
- Agiloft allows specifying fine-grained access permissions on the field level. The username that was used to obtain the specified session token must have sufficient access rights to be able to read field content. Please verify specific permissions via Setup/Access/Manage Groups/Edit Group/Table/Select Table/Field Permissions.
- This call does not return records that have been deleted.
- The queries operate on the database level and should be written in terms of names visible in the Agiloft GUI in the table wizard in the dbname column. Also see EWGetChoiceLineId for special handling of choice values.
- This method never returns null. In the case when no records are found an empty array is returned.
- *Special note on memory management:* As WS integration implies pass-by-value semantics, the memory allocated for the resulting array will be released once the

data is sent to the client and the server-side JVM's garbage collector considers it eligible for discarding.

- If the client-side environment is one with a garbage collector, the memory used by client-side array will be cleared once the client-side garbage collector considers it eligible,
- When the client environment uses explicit memory management, the client is responsible for freeing up the used memory explicitly.
- It is possible to limit the number of records returned through means available to the underlying database used by your instance of Agiloft e.g. "limit 0,200" for MySQL will limit the number of the returned records to 200.
- It is not possible currently to influence the sequence in which records are returned. They are returned in the effectively arbitrary default order returned by the underlying database.
- When using EWSselectAndRead method all fields of the resulting records are read and passed to the client.

Linked Fields

Values for the fields that are imported into the target table from the donor table as a part of a Linked Fields set are available via special linking classes.

In WSDL a Linked Fields set takes form of DAO_Dao3_Link<N> field in the complex data structure that corresponds to the target table, where N is a sequential number assigned automatically at the time of the set creation (e.g. WSCase.DAO_Dao3_Link3).

As a value such fields can take one or more WS<Table1><Table2>_Dao3_Link<N> data structures - linking classes, where Table1 is the target table and Table2 is the donor table of the Linked Fields relationship (e.g. WSCaseTeams_Dao3_Link3).

Unfortunately, at this moment one has to rely on investigating the actual sets of fields inside the classes to trace the fields (visible in the Field Wizard in the GUI) back to the main object property (not visible in the GUI).

Non-source values for Linked Field sets that allow them are present directly in the table itself and additionally to those in the Linking Classes (if the link was in fact forged).

Choice Fields

The values for choice columns are returned as instance(s) of the enumerated types described in the WSDL.

The original text values have undergone the following transformations:

1. Spaces replaced by "_"
2. Dashes replaced by "_MINUS_"
3. Pluses replaced by "_PLUS_"
4. Prefixed with "OPTION_"
5. Converted to upper case

One has to perform the reverse transformation to get to the text value.

Unsupported types of fields

Related tables and embedded search results are not supported by the SOAP interface.

Basic Steps for Selecting and Reading Records in One Call

1. Compose the whereClause. Note: "prepared" statements are not supported, all parameters have to be passed inside of the whereClause string.
2. Perform the call.
3. Handle results, specifically the situations where there are no elements, one element, or more than one element in the returned array.

Example Task

In MyKB knowledgebase, as user A, find all cases assigned to John Doe. Return summaries as a String array.

The task is completed by performing the following steps:

1. Login to MyKB with "A" and "password" and English as the local language.
2. Search for cases assigned to John Doe.
3. Logout

Sample Code - Java

```
public String[] select() throws Exception {
    EWServiceAPI binding = new
EWServiceAPIServiceLocator().getDemo();
    try {
        String sessionId = binding.EWLogin("MyKB", "A", "password",
"en");
        EWWSBaseUserObject[] records =
binding.EWSelectAndRead(sessionId,
                                                                    "case", "assignee='John
Doe'");
        String[] result = new String[records.length];
```

```

for (int i=0; i<records.length; i++) {
    result[i] = records[i].getSummary();
}
return result;
} finally {
binding.EWLogout(sessionId);
}
}

```

You can generate sample Web Services code for any table by selecting *Setup/Table/<Edit Table>/API/Download Sample*.

Arguments

Name	
sessionId	String
tableName	String
where	string

Response

An array of the records as descendants of EWWSBaseUserObject - a complex structure described in WSDL.

Faults

EWSessionException - client not logged in or session has expired; client should re-login.

EWPermissionException - user used to create the session lacks sufficient privileges to run the query.

EWWrongDataException - client has supplied wrong data.

EWOperationException - the operation has been blocked by one of Agiloft functionalities (e.g. table-level lock).

EWIntegrityException - specified table cannot be found or its primary key cannot be identified.

EWUnexpectedException - an unexpected exception has happened; the admin user should report this for investigation.

Useful Example and Hints

This section contains some general hints and examples for various client-side platforms.

PHP SOAPClient

```
$soapSession = new SoapClient("http://host/ewws/KBName/EWSserviceAPIv2?wsdl",
    array("exceptions" => false,
        "location"=>"http://host/ewws/KBName/EWSserviceAPIv2"));
$sess_id = $soapSession-
>EWlogin("KBName", "user", "password", "en");
echo "$sess_id\n";
$ids = $soapSession->EWselectFromTable($sess_id, "case", "1=1");
print_r($ids);
$map = $soapSession->EWRead($sess_id, "case", 521);
print_r($map);
```

SoapUI

When using generic methods a specific descendant of the EWWSBaseUserObject structure needs to be passed. This is achieved by adding the **xsi:type** specialization to the tags defining the structure in the method call parameters.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:tsc="http://KBname.api.ws.agiloft.com"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soapenv:Header/>
    <soapenv:Body>
        <tsc:EWScreate>
            <String_1>d57d</String_1>
            <String_2>case</String_2>
            <EWWSBaseUserObject_3 xsi:type="tsc:WSCase">
                <summary>Example 1</summary>
            </EWWSBaseUserObject_3>
        </tsc:EWScreate>
    </soapenv:Body>
</soapenv:Envelope>
```


REST INTERFACE

REST Interface Overview

Agiloft supports REST-style invocations that correspond to CRUD (Create, Read, Update, Delete) operations. Additionally limited SQL select functionality is supported via EWSelect, Saved Searches and ad-hoc query via EWSearch, and attachment related operations via EWAttach, EWRemoveAttached, and EWRetrieveAttached.

This allows automation of certain actions from external agents (browser forms, Javascript/AJAX). Further details are provided in the REST API Reference section of this manual.

For CRUD operations two invocation styles are supported: pure REST one, where CRUD operations map directly to HTTP methods, and a fallback GET/POST one that can be used instead as not all user-agents may support all HTTP methods. EWSelect is only available via GET/POST.

Operation	/ews/. ..	GET/POST	Returns
Create	POST	/ews/create	id of the newly created record

Read	GET	/ews/ /EWRead	encoded record information
Update	PUT	/ews/ /EWUpdate	encoded record information after update
Delete	DELETE	/ews/ /EWDelete	does not return anything
Select		/ew	a list of record identifiers and a length of that list

e	c	t	w	s	/	E	W	S	e	r	e	c	t
---	---	---	---	---	---	---	---	---	---	---	---	---	---

URL conventions

Note that both KnowledgeBase names and table names are case-sensitive when using the REST interface.

The following general conventions apply to how the URL is constructed:

For pure REST:

```
/ewws/REST/
{kbName}/{table}[/ {id}]?$login={login}&password={password}&lang={lang}&...
```

where the/{id}part is omitted for Create (POST), but used for all others

For fallback Get/Post interface:

```
/ewws/{operation}?
$KB={kbName}&$table={table}&$login={login}&
password={password}&lang={lang}&...
```

The body (for POST and PUT) or the rest of the URL string should contain the parameter name/value pairs, as per operation specification.

For example:

```
GET /ewws/REST/Demo/Company/123?$login=user&$password=123&$lang=en&id=123
```

OR

```
GET /ewws/EWRead?$KB=Demo&$table=Company&$id=123&$login=user&
$password=123&$lang=en&id=123
```

When using the REST interface assume that all table names are case-sensitive. To find the correct case for your table, go to Setup/Tables, select your table, click Edit, and look for the Logical Table Name. This is the name and capitalization you should use when accessing that particular table.

Delays

Each call via the REST interface has a delay inserted after the operation has completed. The delay is set to one second by default and is configurable via the global variable *WSDelay* in the Admin Console.

This has been done due to the frequent mistake of using a web services client to bombard agiloft.com with a constant barrage of requests. Normally an operation on the record invokes rules and other functionalities, so they should be allowed enough time and resources to complete.

Data encoding

The following conventions are in place to encode the following aspects of a typical agiloft.com knowledgebase:

1. Simple fields

can be filled directly by setting the value for them. ...
&first_name=John&last_name=Doe&...

2. Choice fields

are encoded directly with their text values as seen in the GUI.

...&country=USA&...

3. Multi-choice fields

are encoded as multiple key/value pairs.

... &contactMethod=phone&contactMethod=email&...

4. Date, date-time and time fields

can be encoded with any of 3275 formats currently supported.

The system evaluates the possible formats sequentially and stops when parsing in one of the formats succeeds.

Please refer to the following document to see the list of supported date-time formats: <http://support.enterprisewizard.com/ewws/datetime.txt>

5. Elapsed time fields

can be encoded as "days:hours:minutes:seconds" e.g "0:1:35:15"6

6. Linked fields relationships

4a) If the linked field allows independent values these can be simply assigned to the columns in the main table:

```
...&company_name=Agiloft&...
```

If the value contains the colon ':' or the question mark '?' symbols, they have to be escaped with backward-slash "\" in the following way:

```
...&company_name=Who\?When\?&...
```

6b) To forge a link based on the values of imported columns one can use Query By Example, expressed with a colon ':' qualifier.

Example values have to be provided for one or more of the imported columns in the following way:

```
...&company_name=:Agiloft&...
```

or

```
...&company_name=Company:Agiloft&...
```

where donor table name is required if the link type is "single field from multiple tables", but may be omitted in single donor table case.

If the value contains the colon ':' symbol or the question mark '?' symbols, they have to be escaped with backward-slash "\" in the following way:

```
...&company_url=Company:http\://www.example.com&...
```

6c) For more complex queries, possibly including sub-selects, aggregative functions and columns from the donor table that are not imported into the target one can use the SQL query, expressed with a question mark '?' qualifier. What follows the qualifier is the "where" clause for the query that will be run against the donor table. At this moment the columns in the query have to be referred with their database names rather than logical ones.

```
...&company_name=?name like '.com' &...
```

or

```
...&company_name=Company?name like '.com' &...
```

where donor table name is required if the link type is "single field from multiple tables", but may be omitted in the single donor table case.

If the query itself contains a question mark '?' symbol or a colon ':' symbol, they have to be escaped with backward-slash "\" in the following way:

`company_name = Company?name like 'what?'`

7. Multiple values for the linked field

These are encoded as multiple key/value pairs:

`...&company_name=Company:Agiloft& company_name=Company:SaaSWizard&...`

8. File and Image fields (attached files)

REST interface accepts files in POST requests when used with **enctype="multipart/form-data"** encoding.

The name of the form field should match the name of the file or image field. Additionally a field `fieldName$overwrite` can be specified with any value to instruct REST interface to override the current data in the file or image field, rather than add.

Application clients can use [REST - Attach](#) operation instead with PUT HTTP method.

Return values:

Return values are returned in an encoded form suitable for Javascript eval() operation to be applied. After this the values can be accessed from local variables.

To avoid interfering with variables that may already exist in the client script or document, all table column names in the variables that result from the "eval()" call are prefixed with "EWREST_".

```
EWREST_company_name= 'Agiloft';
EWREST_1794_full_name=' agiloft.com Admin';
EWREST_website_url=' http://www.agiloft.com'
EWREST_date_updated=' 21 8 06 15:18:43 PM';
EWREST_id=' 21';
```

HTTP status codes used:

200: successful operation
400: wrong data specified in the request
403: operation not permitted with specified credentials
408: request timeout, user may attempt to retry the request
409: operation cannot be performed, usually means one of the functionalities has blocked it
500: any other problem

Decorators

REST calls allow use of two decorators:

- Asynchronous decorator

Can be applied to EWCreate, EWUpdate and EWDelete calls to do "fire-and-forget" type of call. Should be used if the results normally returned by an operation are not important to the caller e.g. a scenario when a lot of records have to be created in the backend.

Use:

```
/ewws/async/EWCreate?...  
/ewws/EWCreate/.async?...
```

- Redirect decorator

Can be applied to all calls to have a HTTP redirect issued depending on the success of the operation. Useful the scenarios of integration with web sites. Please note the page to which redirect is performed will NOT receive any return data.

Calls require two parameters to be specified: *\$exiturl* and *\$errorurl* - for redirecting in case of successful operation and in case of error respectively. Both parameters should be absolute URLs and URL encoded if necessary.

Use:

```
/ewws/redirect/EWCreate?...&$exiturl=http%3A%2F  
%2Fwww.google.com&$errorurl=http%3A%2F%2Fwww.google.com%2F404  
/ewws/EWCreate/.redirect?...&$exiturl=http%3A%2F  
%2Fwww.google.com&$errorurl=http%3A%2F%2Fwww.google.com%2F404
```

Please note that decorators can be chained. In the case of chaining they are applied from left to right.

Use:

```
/ewws/redirect/EWCreate/.async?...&$exiturl=http%3A%2F  
%2Fwww.google.com&$errorurl=http%3A%2F%2Fwww.google.com%2F404  
/ewws/async/EWCreate/.redirect?...&$exiturl=http%3A%2F  
%2Fwww.google.com&$errorurl=http%3A%2F%2Fwww.google.com%2F404  
/ewws/redirect/async/EWCreate?...&$exiturl=http%3A%2F  
%2Fwww.google.com&$errorurl=http%3A%2F%2Fwww.google.com%2F404  
/ewws/async/redirect/EWCreate?...&$exiturl=http%3A%2F  
%2Fwww.google.com&$errorurl=http%3A%2F%2Fwww.google.com%2F404
```

See Also:

[REST - Create](#)

[REST - Read](#)

[REST - Update](#)

[REST - Delete](#)

[REST - Select](#)

REST Interface - Create

Creates a record in the specified kb and table with values for the specified fields.

Accepts the URL with parameters as per general URL conventions and record data encoded as per Data Encoding, both of which can be viewed in the [REST Interface Overview](#). All parameters MUST be properly URL-encoded.

Returns the identifier of the newly created record.

Example:

Assume an instance of Agiloft is available on localhost, port 8080 and is called "Demo". Create a record for an employee named John Doe, that is assigned to the Staff group and Staff team, with login 'jdoe' and password 'password'.

The following needs to be considered:

- The Employee table is a sub-table of Contacts.
- To create the user we need to fill in the first_name, last_name, _login and password fields.
- We also need to forge links with the Groups and Teams tables via Linked Fields that import multi-value fields Groups and Teams and a single-value field Primary Team.
- We will use the Query By Example capability and supply the names of the Staff group and Staff team.

The following request is issued:

```
http://localhost:8080/ewws/EWCreate?$KB=Demo
&$table=Contacts.Employees&$login=admin&$password=qwerty
&$lang=en&first_name=John&last_name=Doe&_login=jdoe
&password=password&f_group=:Support Staff
&teams=:Support Team&_106_sw_description=:Support Team
```

The following result will be returned in the case of successful creation of the record:

```
EWREST_id='353';
```

The following is an example for Javascript-based client that invokes REST interface via AJAX:

```
function xmlhttpGet (strURL) {
    var xmlhttpReq=false;
    var self=this;
```

```
// Mozilla/Safari
if (window.XMLHttpRequest) {
    try {
        netscape.security.PrivilegeManager.
            enablePrivilege("UniversalBrowserRead");
    } catch (e) {
        alert("Permission UniversalBrowserRead denied.");
    }
    self.xmlHttpRequest=new XMLHttpRequest();
} // IE
else if (window.ActiveXObject) {
    self.xmlHttpRequest=new ActiveXObject("Microsoft.xmlHTTP");
}
self.xmlHttpRequest.open('GET', strURL, true);
self.xmlHttpRequest.onreadystatechange=requestComplete;
self.xmlHttpRequest.send(null);
}
function requestComplete() {
    if (xmlHttpRequest.readyState==4||xmlHttpRequest.readyState=="complete") {
        eval (self.xmlHttpRequest.responseText);
        alert ("Id of new ticket"+EWREST_id);
    }
}
function main() {
    xmlhttpGet('http://localhost:8080/ewws/EWCreate?$KB=Demo
    &$table=Contacts.Employees&$login=admin&$password=qwerty
    &$lang=en&first_name=John&last_name=Doe&login=jdoe
    &password=password&f_group=:Support Staff
    &teams=:Support Team
    &_106_sw_description=:Support Team');
}
```

See Also:

[REST Interface Overview](#)

[REST - Read](#)

[REST - Update](#)

[REST - Delete](#)

[REST - Select](#)

REST Interface - Read

Implements the Read operation of the REST interface.

Accepts the URL with parameters as per general URL conventions which can be viewed in the [REST Interface Overview](#). Additionally, the URL MUST contain the identifier of the record. All parameters MUST be properly URL-encoded.

Returns the encoded data of the record.

Example:

Assume an instance of Agiloft is available on localhost, port 8080 and is called "Demo". Read the record updated in the [update example](#).

The following request is issued:

```
http://localhost:8080/ewws/EWRead?$KB=Demo&$table=Contacts.
  Employees&$login=admin&$password=qwerty&$lang=en&id=358
```

The following result will be returned in the case of successful modification of the record:

```
EWREST_teams_0= 'Support Team';
EWREST__1890_full_name='Admin';
EWREST_id='358';
EWREST__106_sw_description='Support Team ';
EWREST__login='jdoe';
EWREST_company_name_0='Example';
EWREST__ldap='No';
EWREST_type='employees';
EWREST_f_group_0='Support Staff ';
EWREST__groupid= '1772';
EWREST_password='*password is hidden*';
EWREST_first_name='John';
EWREST_opportunity_name_0='new oppt for at7t';
EWREST_last_name='Doe';
EWREST_date_updated='31 8 06 16:34:46 PM';
```

The following is be an example for Javascript-based client that invokes REST interface via AJAX:

```
function xmlhttpGet (strURL) {
  var xmlhttpReq=false;
  var self=this;
  // Mozilla/Safari
  if (window.XMLHttpRequest) {
    try {
```

```
        netscape.security.PrivilegeManager.  
        enablePrivilege("UniversalBrowserRead");  
    } catch (e) {  
        alert("Permission UniversalBrowserRead denied.");  
    }  
    self.xmlHttpRequest=new XMLHttpRequest();  
} // IE  
else if (window.ActiveXObject) {  
    self.xmlHttpRequest=new ActiveXObject("Microsoft.xmlHTTP");  
}  
self.xmlHttpRequest.open('GET', strURL, true);  
self.xmlHttpRequest.onreadystatechange=requestComplete;  
self.xmlHttpRequest.send(null);  
}  
function requestComplete() {  
    if (xmlHttpRequest.readyState==4||xmlHttpRequest.readyState=="complete") {  
        eval (self.xmlHttpRequest.responseText);  
        alert ("Opportunity"+EWREST_opportunity_name_0);    }  
}  
function main() {  
    xmlhttpGet(' http://localhost:8080/ewws/EWRead?$KB=Demo  
&$table=Contacts.Employees&$login=admin&$password=qwerty  
&$lang=en&id=358');  
}
```

See Also:

[REST - Create](#)

[REST - Update](#)

[REST - Delete](#)

[REST - Select](#)

REST Interface - Update

Updates the specified record, implementing the Update operation of the REST interface.

Accepts the URL with parameters as per general URL conventions and record data encoded as per Data Encoding, both of which can be viewed in the [REST Interface Overview](#). All parameters MUST be properly URL-encoded.

Returns the encoded data of the updated record.

Example:

Assume an instance of Agiloft is available on localhost, port 8080 and is called "Demo". Update the record created in the [REST Create example](#) section by linking that employee to an opportunity record for company Agiloft. This operation was not included as part of the create example for illustrative purposes. The same QBE criteria could have been supplied in the create request.

The following needs to be considered:

- The Employee table is a sub-table of Contacts.
- The link to Opportunities is forged via a Linked Field that imports multi-value fields company_name and opportunity_name.
- We will use the Query By Example capability and supply the name of the company as an example.

The following request is issued:

```
http://localhost:8080/ewws/EWUpdate?$KB=Demo
    &$table=Contacts.Employees&$login=admin&$password=qwerty
    &$lang=en&id=358&company_name=:Example
```

The following result will be returned in the case of successful modification of the record:

```
EWREST_teams_0='support Team ' ;
EWREST__1890_full_name='Admin' ;
EWREST_id=' 358' ;
EWREST__106_sw_description='Support Team ' ;
EWREST__login='jdoe' ;
EWREST_company_name_0='Example';
EWREST__ldap=' No ' ;
EWREST_type=' employees';
EWREST_f_group_0='Support Staff ' ;
EWREST__groupid= '1772' ;
EWREST_password='*password is hidden*';
```

```
EWREST_first_name= 'John';
EWREST_opportunity_name_0= 'new oppt for at7t';
EWREST_last_name= 'Doe';
EWREST_date_updated='31 8 06 16:34:46 PM';
```

The following is an example for Javascript-based client that invokes REST interface via AJAX:

```
function xmlhttpGet (strURL) {
    var xmlhttpReq=false;
    var self=this;
    // Mozilla/Safari
    if (window.XMLHttpRequest) {
        try {
            netscape.security.PrivilegeManager.
enablePrivilege("UniversalBrowserRead");
        } catch (e) {
            alert("Permission UniversalBrowserRead denied.");
        }
        self.xmlhttpReq=new XMLHttpRequest();
    } // IE
    else if (window.ActiveXObject) {
        self.xmlhttpReq=new ActiveXObject("Microsoft.xmlHTTP");
    }
    self.xmlhttpReq.open('GET', strURL, true);
    self.xmlhttpReq.onreadystatechange=requestComplete;
    self.xmlhttpReq.send(null);
}
function requestComplete() {
    if (xmlhttpReq.readyState==4||xmlhttpReq.readyState=="complete") {
        alert ("Update completed");
    }
}
function main() {
    xmlhttpGet('http://localhost:8080/ewws/EWUpdate?$KB=Demo
    &$stable=Contacts.Employees&$login=admin&$password=qwerty
    &$lang=en&id=358&company_name=:Example');
}
```

See Also:

[REST Interface Overview](#)

[REST - Read](#)

[REST - Create](#)

[REST - Delete](#)

[REST - Select](#)

REST Interface - Delete

Deletes the specified record or records.

Accepts the URL with parameters as per general URL conventions that can be viewed in the [REST Interface Overview](#). Additionally, the URL must contain one or more record identifiers and the delete rule name, and optionally, the substitute record identifiers.

All parameters **MUST** be properly URL-encoded.

This call does not return anything when the operation is successful. The transaction completes only if all records are successfully deleted, otherwise it is rolled back.

The **deleteRule** parameter defines one of the following strategies to be applied for dependent records:

- **ERROR_IF_DEPENDANTS** - operation fails when there are any dependent records,
- **APPLY_DELETE_WHERE_POSSIBLE** - tries to delete all dependent records. When delete cannot be done, an attempt to unlink the record is made,
- **DELETE_WHERE_POSSIBLE_OTHERWISE_UNLINK** - same as above,
- **APPLY_UNLINK** - tries to unlink dependent records,
- **UNLINK_WHERE_POSSIBLE_OTHERWISE_DELETE** - tries to unlink all dependent records, when unlink cannot be done an attempt to delete the record is made,
- **REPLACE_WITH_ANOTHER** - tries to link dependent records to the substitute one specified in replacementKeys

The **subs** parameter is taken in consideration only if the deleteRule **REPLACE_WITH_ANOTHER** is specified and should contain identifiers of records from the same table to be used as substitutes. For each record dependent on the record being deleted, a record with a corresponding replacement key will become the parent one.

If the specific strategy fails the error message returned will suggest alternatives to be used.

*If **APPLY_DELETE_WHERE_POSSIBLE** or **DELETE_WHERE_POSSIBLE_OTHERWISE_UNLINK** strategies are used and the configuration of the knowledgebase allows it, a special "Fast Delete" algorithm is used just as via the GUI.*

Example:

Assume an instance of Agiloft is available on localhost, port 8080 and is called "Demo". Delete the record updated in the [REST Update Example](#).

The following request is issued:

```
http://localhost:8080/ewws/EWDelete?$KB=Demo
&$table=Contacts.Employees&$login=admin&$password=qwerty
&$lang=en&id=358&deleteRule=APPLY_DELETE_WHERE_POSSIBLE
```

The following is an example for Javascript-based client that invokes REST interface via AJAX:

```
function xmlhttpGet (strURL) {
    var xmlhttpReq=false;
    var self=this;
    // Mozilla/Safari
    if (window.XMLHttpRequest) {
        try {
            netscape.security.PrivilegeManager.
enablePrivilege("UniversalBrowserRead");
        } catch (e) {
            alert("Permission UniversalBrowserRead denied.");
        }
        self.xmlHttpRequest=new XMLHttpRequest();
    } // IE
    else if (window.ActiveXObject) {
        self.xmlHttpRequest=new ActiveXObject("Microsoft.xmlHTTP");
    }
    self.xmlHttpRequest.open('GET', strURL, true);
    self.xmlHttpRequest.onreadystatechange=requestComplete;
    self.xmlHttpRequest.send(null);
}
function requestComplete() {
    if (xmlHttpRequest.readyState==4||xmlHttpRequest.readyState=="complete") {
        alert ("Delete completed");
    }
}
function main() {
    xmlhttpGet('http://localhost:8080/ewws/EWDelete?$KB=Demo
&$table=Contacts.Employees&$login=admin&$password=qwerty
&$lang=en&id=358
&deleteRule=APPLY_DELETE_WHERE_POSSIBLE');
```

See Also:

[REST Interface Overview](#)

[REST - Read](#)

[REST - Update](#)

[REST - Create](#)

[REST - Select](#)

REST Interface - Select

Performs a select on the specified table, implementing the Select operation of the REST interface.

Accepts the URL with parameters as per general URL conventions which can be viewed in the [REST Interface Overview](#). Additionally, the URL MUST contain the where clause of the SELECT query to be performed. All parameters MUST be properly URL-encoded.

This returns the number of records found, and an encoded data list of record IDs (if any were found).

Only one SQL statement is accepted. Core functionality automatically and transparently applies security filters to prevent unauthorized access to information.

These SQL queries operate on the database level and should be written in terms of dbname column names as visible in the in the Table Wizard in the GUI. Some values for high-level complex field types may have very different representation from what is seen in the GUI. For example, see [REST Interface - GetChoiceLineId](#) for the special handling of choice values.

The SQL query supplied should only include the 'where' clause from a construct similar to:

```
select primary-key-field-name from table-name where ...
```

The query should be properly URL-encoded

This returns the number of records found, and an encoded data list of record IDs (if any were found).

It is possible to limit the number of records returned through means available to the underlying database used by your instance of Agiloft e.g. "limit 0,200" for MySQL will limit the number of the returned records to 200.

It is not possible currently to influence the sequence in which records are returned. They are returned in the effectively arbitrary default order returned by the underlying database. Please consider using a EWSearch call instead if possible. This will allow the specification of the sort order in the Saved Search to be used.

Example 1:

Assume an instance of Agiloft is available on localhost, port 8080 and is called "Demo". List the records assigned to 'John Doe'.

The following request is issued:

```
http://localhost:8080/ewws/EWSelect?
$KB=Demo&$login=admin&$password=qwerty&
$table=case&$lang=en&where=_1905_full_name='John%20Doe'
```

If there are no records found, the following result will be returned:

```
EWREST_id_length = '0';
```

The following result will be returned in the case of four records being found:

```
EWREST_id_length = '4';
EWREST_id_0 = '82';
EWREST_id_1 = '83';
EWREST_id_2 = '352';
EWREST_id_3 = '355';
```

Example 2:

Assume an instance of Agiloft is available on localhost, port 8080 and Demo KB. List the records with summaries starting from word 'How'

The following request is issued:

```
http://localhost:8080/ewws/EWSelect?
$KB=Demo&$login=admin&$password=qwerty&
$table=case&$lang=en&where=summary%20like%20'How%25'
```

If there are no records found, the following result will be returned:

```
EWREST_id_length = '0';
```

The following result will be returned in the case of four records being found:

```
EWREST_id_length = '4';
EWREST_id_0 = '82';
EWREST_id_1 = '83';
EWREST_id_2 = '352';
EWREST_id_3 = '355';
```

The following is an example for the Javascript-based client that invokes the REST interface via AJAX:

```
function xmlhttpGet (strURL) {
```

```
var xmlHttpRequest=false;
var self=this;
// Mozilla/Safari
if (window.XMLHttpRequest) {
    try {
        netscape.security.PrivilegeManager.
enablePrivilege("UniversalBrowserRead");
    } catch (e) {
        alert("Permission UniversalBrowserRead denied.");
    }
    self.xmlHttpRequest=new XMLHttpRequest();
} // IE
else if (window.ActiveXObject) {
    self.xmlHttpRequest=new ActiveXObject("Microsoft.xmlHTTP");
}
self.xmlHttpRequest.open('GET', strURL, true);
self.xmlHttpRequest.onreadystatechange=requestComplete;
self.xmlHttpRequest.send(null);
}
function requestComplete() {
    if (xmlHttpRequest.readyState==4||xmlHttpRequest.readyState=="complete") {
        eval (self.xmlHttpRequest.responseText);
        alert ("Id of new ticket"+EWREST_id);
    }
}
function main() {
    xmlhttpGet('http://localhost:8080/ewws/EWSelect?$KB=Demo
&$stable=case&$login=admin&$password=qwerty
&$lang=en&where=summary%20like%20'How%25''');
}
```

See Also:

[REST Interface Overview](#)

[REST - Read](#)

[REST - Create](#)

[REST - Update](#)

[REST - Delete](#)

REST Interface - Search

Runs a Saved Search or an ad-hoc query on the specified table, implementing a Search-like operation of the REST interface.

Accepts the URL with parameters as per general URL conventions which can be viewed in the [REST Interface Overview](#). The name of the Saved Search or an ad-hoc query MAY be included in the call. All parameters MUST be properly URL-encoded.

The operation returns the number of records found, and encoded sets of field values - one for each record found. Each set will always include the id and the fields used by the table and its subtables to define record ownership, such as customer_login in the examples below. Additional fields can be added to the list by specifying a series of 'field' parameters with logical field names.

The name should be properly URL-encoded

Example 1:

Assume an instance of Agiloft is available on localhost, port 8080 and the knowledgebase is called 'Demo'. List the records that correspond to the Saved Search 'Closed Cases'. Additionally, only return those that have Low priority.

The following request is issued:

```
http://localhost:8080/ewws/EWSearch?
$KB=Demo&$login=admin&$password=qwerty&
$table=case&$lang=en&search=Closed%20Cases&
query=Priority%3DLow
```

If there are no records found, the following result will be returned:

```
EWREST_id_length = '0';
```

The following result will be returned in the case of four records being found. Please note id, customer_login (based on the record ownership defined for the Support Case table), and type (based on the permissions definition for the Support Case table):

```
EWREST_length = '4';
EWREST_id_0='355';
EWREST_customer_login_0='admin';
EWREST_type_0='Case';
EWREST_id_1='117';
EWREST_customer_login_1='john1';
EWREST_type_1='Case';
EWREST_id_2='83';
```

```
EWREST_customer_login_2='admin';
EWREST_type_2='Case';
EWREST_id_3='82';
EWREST_customer_login_3='admin';
EWREST_type_3='Case';
EWREST_id_4='83';
EWREST_customer_login_4='admin';
EWREST_type_4='Case';
EWREST_id_5='82';
EWREST_customer_login_5='admin';
EWREST_type_5='Case';
```

Example 2:

Assume an instance of Agiloft is available on localhost, port 8080 and Demo KB. Now we want to retrieve extra fields in addition to the default ones.

The following request is issued:

```
http://localhost:8080/ewws/EWSearch?
$KB=Demo&$login=admin&$password=qwerty&
$stable=case&$lang=en&search=Closed%20Cases&field=summary&field=priority
```

If there are no records found, the following result will be returned:

```
EWREST_id_length = '0';
```

The following result will be returned in the case of six records being found:

```
EWREST_length = '6';
EWREST_type_0='Case';
EWREST_summary_0='How do users add notes to a record?';
EWREST_priority_0='Low';
EWREST_id_0='355';
EWREST_customer_login_0='admin';
EWREST_type_1='Case';
EWREST_summary_1='How can I learn how to use the system?';
EWREST_priority_1='High';
EWREST_id_1='352';
EWREST_customer_login_1='customer';
EWREST_type_2='Case';
EWREST_summary_2='here is my problem';
EWREST_priority_2='High';
EWREST_id_2='350';
EWREST_customer_login_2='customer';
EWREST_type_3='Case';
EWREST_summary_3='Having trouble installing on Windows 2003closed it';
EWREST_priority_3='Low';
EWREST_id_3='117';
EWREST_customer_login_3='john1';
EWREST_type_4='Case';
```

```

EWREST_summary_4='How do users add notes to a record?';
EWREST_priority_4='Low';
EWREST_id_4='83';
EWREST_customer_login_4='admin';
EWREST_type_5='Case';
EWREST_summary_5='How do I add a new table?';
EWREST_priority_5='Low';
EWREST_id_5='82';
EWREST_customer_login_5='admin';

```

The following is an example for the Javascript-based client that invokes the REST interface via AJAX:

```

function xmlhttpGet (strURL) {
    var xmlhttpReq=false;
    var self=this;
    // Mozilla/Safari
    if (window.XMLHttpRequest) {
        try {
            netscape.security.PrivilegeManager.
enablePrivilege("UniversalBrowserRead");
        } catch (e) {
            alert("Permission UniversalBrowserRead denied.");
        }
        self.xmlhttpReq=new XMLHttpRequest();
    } // IE
    else if (window.ActiveXObject) {
        self.xmlhttpReq=new ActiveXObject("Microsoft.xmlHTTP");
    }
    self.xmlhttpReq.open('GET', strURL, true);
    self.xmlhttpReq.onreadystatechange=requestComplete;
    self.xmlhttpReq.send(null);
}
function requestComplete() {
    if (xmlhttpReq.readyState==4||xmlhttpReq.readyState=="complete") {
        eval (self.xmlhttpReq.responseText);
        alert ("Id of new ticket"+EWREST_id);
    }
}
function main() {
    xmlhttpGet ('http://localhost:8080/ewws/EWSearch?
$KB=Demo&$login=admin&$password=qwerty&
$table=case&$lang=en&search=Closed%20Cases&field=summary&field=priority');
}

```

See Also:

[REST Interface Overview](#)

[REST - Read](#)

[REST - Create](#)

[REST - Update](#)

[REST - Delete](#)

[REST - Select](#)

REST Interface - GetChoiceLineId

Obtains the internal identifier that corresponds to a choice value for use in SQL-based expressions.

Accepts the URL with parameters as per general URL conventions which can be viewed in the [REST Interface Overview](#). Additionally, the URL MUST contain the name of the choice field and the choice list element value. All parameters need to be properly URL-encoded.

This returns the identifier of the choice list element suitable to use in EWSelect.

Use EWGetChoiceLineId to obtain the internal identifier that corresponds to the choice text value for further use in SQL-based expressions. This is needed for the where parameter of EWSelectFromTable or the query text when forging links to other tables via Linked Field sets.

SQL-based expressions are evaluated on the database level where choice values are stored as identifiers. These identifiers change if the knowledgebase is copied or re-imported, or the values in the choice list are re-created. It is therefore advisable to use the EWGetChoiceLineId call to obtain the id value at runtime immediately before the SQL call.

If the factors that trigger the change of the internal choice identifiers (knowledgebase copy or re-import, or choice list values are re-created) are detected on the client, then the values returned by this call can be safely cached on a per KB basis between multiple calls.

The choice text value should be properly URL-encoded

Example 1:

Assume an instance of Agiloft is available on localhost, port 8080 and is called "Demo". Obtain the identifier for choice value 'High' for the field 'Priority' in the table 'Case'.

The following request is issued:

```
http://localhost:8080/ewws/EWGetChoiceLineId?
$KB=Demo&$login=admin&$password=qwerty&
$table=case&$lang=en&field=priority&value=High
```

If there were no matching choice list values found, the HTTP code 400 will be returned with a brief explanation of the problem.

The following result will be returned in the case of the text value being successfully translated into a numeric identifier:

```
EWREST_choiceLineId = '1';
```

The following is an example for the Javascript-based client that invokes the REST interface via AJAX:

```
function xmlhttpGet (strURL) {
    var xmlhttpReq=false;
    var self=this;
    // Mozilla/Safari
    if (window.XMLHttpRequest) {
        try {
            netscape.security.PrivilegeManager.
            enablePrivilege("UniversalBrowserRead");
        } catch (e) {
            alert("Permission UniversalBrowserRead denied.");
        }
        self.xmlHttpRequest=new XMLHttpRequest();
    } // IE
    else if (window.ActiveXObject) {
        self.xmlHttpRequest=new ActiveXObject("Microsoft.xmlHTTP");
    }
    self.xmlHttpRequest.open('GET', strURL, true);
    self.xmlHttpRequest.onreadystatechange=requestComplete;
    self.xmlHttpRequest.send(null);
}
function requestComplete() {
    if (xmlHttpRequest.readyState==4||xmlHttpRequest.readyState=="complete") {
        eval (self.xmlHttpRequest.responseText);
        alert ("Id of new ticket"+EWREST_id);
    }
}
function main() {
    xmlhttpGet('http://localhost:8080/ewws/EWGetChoiceLineId?$KB=Demo
    &$table=case&$login=admin&$password=qwerty
    &$lang=en&field=priority&value=High');
}
```

See Also:

[REST Interface Overview](#)

[REST - Read](#)

[REST - Create](#)

[REST - Update](#)

[REST - Delete](#)

REST Interface - Attach

Attaches the file passed in the body of the request to the specified field of the record identified by the supplied primary key.

Accepts the URL with parameters as per general URL conventions which can be viewed in the [REST Interface Overview](#). Additionally, the URL MUST contain the identifier of the record, the name of the field to attach the file to and the file name to use. The file content is passed in the body of the request. All parameters MUST be properly URL-encoded.

Returns the current number of files attached to the specified field.

Example:

Assume an instance of Agiloft is available on localhost, port 8080 and is called "Demo". Attach a file to the field "someField" in record 1234 of table "someTable".

The request has to be issued via the PUT HTTP method where the body contains the content of the file to be attached.

```
http://localhost:8080/ewws/EWAttach?$KB=Demo
$table=someTable&$login=admin&$password=qwerty
&id=1234&field=someField
```

The following result will be returned when the file is attached.

```
EWREST_someField.length='1';
```

*At the moment browsers do not support sending PUT requests. This REST operation is intended for the use of application clients. If you need to attach a file from a browser, use the regular POST requests with EWCreate or EWUpdate operations, and specify **enctype="multipart/form-data"** in the browser HTML form.*

See Also:

[REST Interface Overview](#)

[REST - Read](#)

[REST - Update](#)

[REST - Delete](#)

[REST - Select](#)

[REST - Attach](#)

[REST - Remove Attached](#)

[REST - Retrieve Attached](#)

REST Interface - Remove Attached

Removes the attached file specified by the position in the named field of the record identified by the supplied key.

Accepts the URL with parameters as per general URL conventions which can be viewed in the [REST Interface Overview](#). Additionally, the URL MUST contain the identifier of the record, the name of the field to remove the file from and the position of the file in the field. All parameters MUST be properly URL-encoded.

Returns the number of files remaining attached in the field.

Example:

Assume an instance of Agiloft is available on localhost, port 8080 and is called "Demo". Remove the second file attached to the field "someField" in the record 1234 of the table "someTable". Assume the field has four files attached. Note that position count starts from zero.

```
http://localhost:8080/ewws/EWRemoveAttached?$KB=Demo
    &$table=someTable&$login=admin&$password=qwerty
    &id=1234&field=someField&filePosition=1
```

The following result will be returned in the case of successful removal of the file:

```
EWREST_someField.length='3';
```

The following can be an example for Javascript-based client that invokes REST interface via AJAX:

```
function xmlhttpGet (strURL) {
    var xmlhttpReq=false;
    var self=this;
    // Mozilla/Safari
    if (window.XMLHttpRequest) {
        try {
            netscape.security.PrivilegeManager.
                enablePrivilege("UniversalBrowserRead");
        } catch (e) {
            alert("Permission UniversalBrowserRead denied.");
        }
    }
    self.xmlhttpReq=new XMLHttpRequest();
} // IE
else if (window.ActiveXObject) {
    self.xmlhttpReq=new ActiveXObject("Microsoft.xmlHTTP");
}
self.xmlhttpReq.open('GET', strURL, true);
self.xmlhttpReq.onreadystatechange=requestComplete;
```

```
        self.xmlHttpRequest.send(null);
    }
    function requestComplete() {
        if (xmlHttpRequest.readyState==4||xmlHttpRequest.readyState=="complete") {
            eval (self.xmlHttpRequest.responseText);
            alert ("Id of new ticket"+EWREST_id);
        }
    }
    function main() {
        xmlhttpGet('http://localhost:8080/ewws/EWRemoveAttached?$KB=Demo
        &$table=someTable&$login=admin&$password=qwerty
        &id=1234&field=someField&filePosition=1#39;');
    }
}
```

See Also:

[REST Interface Overview](#)

[REST - Read](#)

[REST - Update](#)

[REST - Delete](#)

[REST - Select](#)

[REST - Attach](#)

[REST - Remove Attached](#)

[REST - Retrieve Attached](#)

REST Interface - Retrieve Attached

Retrieves the attached file specified by position from the named field in the record identified by the supplied key.

Accepts the URL with parameters as per general URL conventions which can be viewed in the [REST Interface Overview](#). Additionally, the URL MUST contain the identifier of the record, the name of the field to retrieve the file from and the position of the file in the field. All parameters MUST be properly URL-encoded.

File will be returned as binary content in the body of the response (GET or POST HTTP method),

Example:

Assume an instance of Agiloft is available on localhost, port 8080 and is called "Demo". Retrieve the second file attached to field "someField" in the record 1234 of the table "someTable". Note that position count starts from zero.

The following request is issued:

```
http://localhost:8080/ewws/EWRetrieveAttached?$KB=Demo
&$table=someTable&$login=admin&$password=qwerty
&id=1234&field=someField&filePosition=1
```

The file will be returned in the body of the response.

The following can be an example for Javascript-based client that invokes REST interface via AJAX:

```
function xmlhttpGet (strURL) {
    var xmlhttpReq=false;
    var self=this;
    // Mozilla/Safari
    if (window.XMLHttpRequest) {
        try {
            netscape.security.PrivilegeManager.
                enablePrivilege("UniversalBrowserRead");
        } catch (e) {
            alert("Permission UniversalBrowserRead denied.");
        }
        self.xmlhttpReq=new XMLHttpRequest();
    } // IE
    else if (window.ActiveXObject) {
        self.xmlhttpReq=new ActiveXObject("Microsoft.xmlHTTP");
    }
    self.xmlhttpReq.open('GET', strURL, true);
    self.xmlhttpReq.onreadystatechange=requestComplete;
    self.xmlhttpReq.send(null);
}
```

```
}  
function requestComplete() {  
    if (xmlHttpRequest.readyState==4||xmlHttpRequest.readyState=="complete") {  
        alert ("A file has been returned");  
    }  
}  
function main() {  
    xmlhttpGet('http://localhost:8080/ewws/EWRetrieveAttached?$KB=Demo  
&$table=someTable&$login=admin&$password=qwerty  
&id=1234&field=someField&filePosition=1#39;);  
}
```

See Also:

[REST Interface Overview](#)

[REST - Read](#)

[REST - Update](#)

[REST - Delete](#)

[REST - Select](#)

[REST - Attach](#)

[REST - Remove Attached](#)

[REST - Retrieve Attached](#)

SYNCHRONIZATION GUIDE

Introduction

The Agiloft synchronization sub-system (Sync) provides facilities for automatic synchronization between Agiloft tables and corresponding records in some external system (XS). Synchronization can be bidirectional or unidirectional in either direction.

Sync manages all the functionality that is common between external systems. For example, it allows the user to define the table and field mapping, it handles the communication of the data and determines which records need to be updated in each system.

Each external system type also needs an External System Adaptor (ESA) to handle its unique characteristics. The ESA is responsible for implementing functions such as "List the available tables" "List the fields in a given table", "Create, Replace, Update, Delete a record".

Agiloft comes with prebuilt ESA's for common systems and new ones can be created by customers in Java or their choice of language. If the customer develops the ESA in Java, it can be linked with the Remote Proxy that actually communicates with the Sync sub-system. If the customer develops the ESA in some other language, it communicates with the Remote Proxy using standard I/O streams.

To be synchronizable with Agiloft, an external system must have these properties:

- It must be possible to obtain some kind of modification timestamp for each record. Typically this timestamp is held in the record itself, but this is not absolutely necessary.
- Record ID's should not be changed or at least there must be a way to know any record ID as it was on the last sync. Record ID's can be "remapped" on a sync, but the old ID must be known to the ESA. This is used to track record matches.
- Records from the external system, as presented to Sync by the ESA, must be grouped by "structures", which are mapped 1-1 to Agiloft types or Tables (e.g. Contact/Employee). A structure can be a table, a folder or anything else that logically groups records in the external system. Within a given structure, external records must use the same set of field mappings.

A given Agiloft table can be synced with multiple external systems and it is not necessary for all fields to be mapped. Existing ESA's provide support for synchronization with: File Directory (files), Microsoft Exchange, QuickBooks, Excel

Terminology

Agiloft	Agiloft Inc/Agiloft Installation
Sync	Agiloft Synchronization Sub-system
syncn	Synchronize/synchronization
XML	Extensible Markup Language. See http://www.w3.org/TR/REC-xml/
XS	External System (Separate from Agiloft)
STDIO	Standard Input/Output streams of console
HTTPs	Hypertext Transfer Protocol (secure)
URL	Universal Resource Locator
URL Encoding	Encoding as prescribed in http://www.w3schools.com/TAGS/ref_urlencode.asp
NAT	Network Address Translation. see http://www.webopedia.com/TERM/N/NAT.html

Part I - How to Synchronize

How to Synchronize

The provisions to simplify the interface between any ESA and Sync are these:

- **Sync Configuration Wizard**
This creates a record within Agiloft, specifying synchronization of a unique XS, and all aspects of the syncn. The record is called a "Sync Configuration".
- **ESA Remote Proxy**
This is an Agiloft program, with parameters set by the Sync Configuration Wizard, which runs a "command-line" ESA in the XS host and communicates between that ESA and Sync. A command-line ESA is one communicating with the XS by STDIO messages, or running in the Agiloft host and communicating with it by STDIO messages.
- **ESA Setting Tab**
This enables the ESA to establish types of parameter to be held by the Sync Configuration.
- **Table-Structure Mapping Tab**
This maps Agiloft tables to structures in the XS, or derived from it by the ESA
- **Field Mapping Wizard**
This is entered for each structure selected in the Table-Structure Mapping Tab, pairs fields in the table with matching fields in the structure, and controls updating both in the XS and in Sync.
- **Manual Sync Menu**
This appears on the Table toolbar of Agiloft when you browse a table involved in any Sync Configuration, and provides manual triggering of the appropriate sync. As a rule Sync performs automatically.
- **Agiloft Helper API**
This interface of Agiloft accepts calls from the ESA, of two kinds:
 - Within a syncn cycle
 - To initiate a syncn cycle, from a non command-line ESA (see ESA Development - Outline).

In addition, Java classes for message handling are available for any ESA that is to be developed in Java. The section "ESA Development" describes ESA development requirements, the extensive re-usable Java classes available, and provides a complete ESA in Java.

Sync Outline

XS Requirements

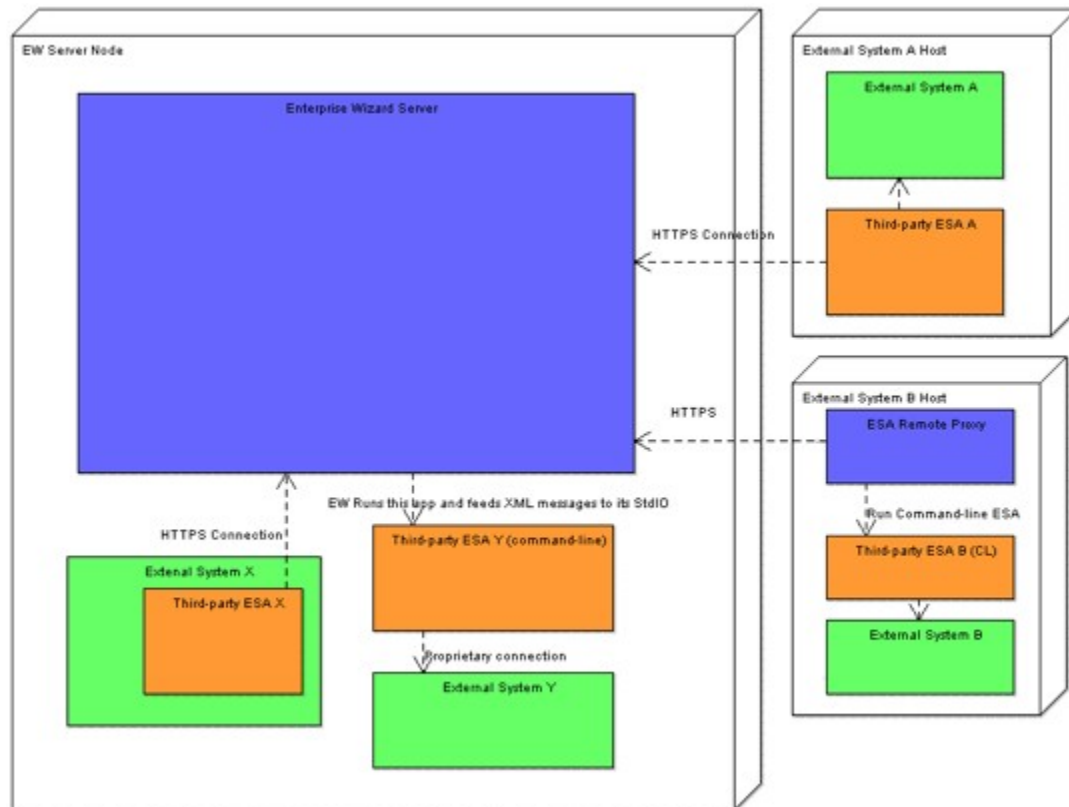
The synchronization process is based on comparing record timestamps in Agiloft and the XS. If a synchronization was run on Wednesday and is next run on Friday, it will look for all records, changed in one or both systems since Wednesday and will propagate those updates to the other system. At the end of a synchronization process, both systems will have matching data. Creation of a new record or deletions may also be carried over to the other system.

To be able to synchronize with Agiloft, an external system must have these properties:

- It must be possible to obtain some kind of modification timestamp for each record. Typically this timestamp is held in the record itself, but this is not absolutely necessary.
- Record ID's should not be changed or at least there must be a way to know any record ID as it was on the last syncn. Record ID's can be "remapped" on a syncn, but the old ID must be known to the ESA. This is used to track record matches.
- Records from the external system, as presented to Sync by the ESA, must be grouped by "structures", which are mappable 1-1 to Agiloft types or Tables (e.g. Contact/Employee). A structure can be a table, a folder or anything else that logically groups records in the external system. Within a given structure, external records must have the same set of fields, and the corresponding Agiloft Table must have a corresponding set.

ESA Deployment

Possible arrangements are shown below:



Blue components are Agiloft components: the Agiloft server application itself and the ESA Remote Proxy application. Green rectangles are External Systems (XSEs).

Orange rectangles are possible ESA deployments:

- ESA A sits on the external system host, talking with XS over some possibly proprietary protocol and talking with Agiloft over HTTPs, as if over the Internet
- ESA B is a command-line application, which also sits on the external system host but is managed (launched) by an ESA Remote Proxy which, in turn, talks with Agiloft over HTTPs and uses standard I/O streams for message exchange with the ESA.
- X and Y are similar to A and B respectively, but are working on the Agiloft host. Y is run and communicates directly with Agiloft through STDIO, X is run within its XS, and communicates with Agiloft over HTTPs, as if over the Internet.

There are 4 "rules of thumb" when choosing the deployment type:

- If your "External system" is a server- or daemon-like continually running software, implementing the ESA as a module thereof might be more natural, because you will

control the lifecycle of your ESA. In this case, the ESA should connect with Agiloft over HTTPs.

- For an application that is not always running, use the command-line form.
- If you need to trigger synchronization upon External System events, use an HTTPs possibly daemon-like ESA. A command-line ESA cannot connect to Agiloft on its own. It is only started when Agiloft begins a syncn.
- If you don't ever need the XS to trigger a syncn, develop your ESA as a command-line application, using standard I/O communication with the XS. If the ESA is not on the Agiloft host, an ESA Remote Proxy will be needed to interface it with Agiloft, but in any case the ESA will be simpler to develop.




The Agiloft ESA for QuickBooks must run on the Agiloft host, but otherwise ESA's can be fitted in any of these contexts. The disposition of Firewalls will often determine the most convenient context. The context you choose will dictate certain Sync Configuration Wizard entries.

Sync Configuration Wizard

Sync Configuration Wizard

When you select Setup/Sync you are presented with a list of any existing Sync Configurations. You may create a new configuration, or edit or delete existing ones.

Setup → Sync

New ▾ Delete Export Import				
Edit	Name	Synchronization Type	EW Tables	Run Mode
<input type="radio"/> 	Google Sync	GOOGLE	Calendar, Contact	Manual
<input type="radio"/> 	4-1-13 Look and Feel	Entity Set		
<input type="radio"/> 	Google Sync	Entity Set	Calendar, Contact	

To create a new configuration, press "New". To delete a configuration, select it in the list and press "Delete". To edit an existing configuration, press edit icon next to the configuration. Upon creating new or editing an existing configuration, the General tab of GUI is displayed:

General	
<input type="button" value="Next"/> <input type="button" value="Cancel"/>	
<p>Please enter a name for this Sync Configuration that is based on the system with which it synchronizes.</p>	<p>Configuration Name <input type="text" value="MySync"/></p>
<p>Please select what External System Adapter (ESA) you want to use to communicate with the other system. You may choose an existing ESA or use your own.</p>	<p>External System Type <input type="button" value="QuickBooks"/> <ul style="list-style-type: none"> <input checked="" type="radio"/> Third-party ESA (HTTP) <input type="radio"/> Third-party ESA (Command Line) Command Line <input type="text"/> <p>You can download third-party ESA development example here</p> </p>
<p>You can temporary Enable or Disable Synchronization Configurations</p>	<p>Status <input checked="" type="checkbox"/> Enabled</p>
<p>You may specify one-way transfer in either direction or full bi-directional synchronization.</p>	<p>Directions <input checked="" type="radio"/> Two-way sync <input type="radio"/> Update EW only <input type="radio"/> Update External System only</p>
<p>A conflict can arise in two way synchronization if a record is edited on both systems before the changes from one are synced with the other. For example if the systems are synced at 1:00pm and 1:05pm and the records are edited at 1:01pm and 1:02pm</p> <p>This option allows you to specify how such conflicts are resolved.</p>	<p>Conflicts <input type="radio"/> This system should take precedence <input checked="" type="radio"/> External System should take precedence <input type="radio"/> Take the most recent record <input type="radio"/> Duplicate records</p>
<p>An ESA can be run remotely and connect to the the Agiloft server through a firewall. To setup a remote connection, download the ESA Remote Proxy package and run it on the target machine using java -jar esa.jar.</p> <p>Notes:</p> <p>You will also need latest Java Runtime environment, obtainable from www.java.com.</p> <p>The downloaded package includes all the current settings and parameters such as the External System ID. If you change these settings, you will need to re-download the package.</p> <p>You may also specify a specific IP address to only allow connections from that IP. If you are using NAT, you should specify the external IP address as "seen" by EW, rather than the internal address.</p>	<p>Remoting <input type="checkbox"/> External System Adapter runs remotely <input type="button" value="Download ESA Remote Proxy"/> Polling Period (secs) <input type="text" value="30"/> IP Filter (optional) <input type="text"/></p>
<p>If you set this option, synchronization data are stored on a per-user basis. This is useful when synchronizing with a PIM (Personal Information Manager) software, like MS Outlook. As an admin, you can set mappings, options, etc at once, and your users will be able to synchronize with their own PIM installations, sharing the data.</p>	<p>Personal synchronization <input type="checkbox"/> Personal data</p>
<p>The system has automatically generated a globally unique "External System ID" for this configuration. This name is used internally to manage communications between external systems and Agiloft KB's. You may need this name if you are debugging a third party ESA.</p>	<p>*External System ID Prefix <input type="text" value="1387935569839"/> Full External System ID: <input type="text" value="1387935569839@Demo for Soree"/></p>
<input type="button" value="Next"/> <input type="button" value="Cancel"/>	

The entries are:

Configuration Name

This will identify the Sync Configuration name, as seen in the Table menu for manual syncn.

External System Type

The choices available include the XS's for which Agiloft already supplies an ESA. For other XS's, check one of the two checkboxes below, Thiiird-Party ESA (HTTP) or Third-Party ESA(command-line). These denote two kinds of ESA - XS communication, described in the next two sections.

Command Line

For a command-line ESA, enter in this field a command line in which the path and any arguments will be valid for the ESA host machine, whether this is the Agiloft host, or an external host. This command will be used to start the ESA from an ESA Remote Proxy if the ESA is externally hosted, or from Agiloft itself if the ESA is on the Agiloft host.

Remoting

Under this heading are 3 entries and a switch button which control much of the Sync operation.

- External System Adapter runs remotely: check this if the ESA is to run outside the Agiloft host.
- Download ESA Remote Proxy: this button is active if the previous checkbox is ticked, but should only be used for a remote Command-Line ESA. The Proxy is described in its own section later. It takes parameters from the entries in this GUI, so the Download button will not be effective till these entries are completed.
- Polling Period: When an ESA is run remotely, it is presumed that it is behind the firewall. So, ESA should eventually come to Agiloft and "ask" whether there are synchronizations pending. The "Polling period" setting (seconds) determines how long the ESA will wait between such requests.
- IP Filter (optional): If an IP Address is entered here, ESA or Remote Proxy connections will be accepted by the Sync Configuration concerned only from that IP. In the case of NAT, this must be the outermost IP of the ESA machine, as seen by Agiloft.

Directions

Here you choose one of 3 checkboxes, to select for updating records in Agiloft, the XS or in both data sets.

External System ID

An ID which uniquely identifies the Sync Configuration being produced is generated by the Sync system and placed in this field. At this point (before the configuration is Created) you can edit it to make it more easily memorable - taking care that it remains unique.

A remote HTTPs ESA needs this ID as a parameter, and its developer must ensure that this parameter is received by the ESA.

Next Button

Click this to create the Sync Configuration which will handle all syncns on the selected ESA. After clicking "Create", that button is replaced by a "Connect ESA" button, which brings up the ESA Settings GUI. Upon pressing Next, Agiloft will wait for incoming connection from the ESA. If an ESA is to run remotely, it or the ESA Remote Proxy should be started now.

ESA Remote Proxy

A command-line ESA separated from Agiloft by a firewall can not use STDIO streams to communicate with Agiloft. The ESA Remote Proxy is a Java application provided by Agiloft to interface with an ESA by STDIO, and with Agiloft by HTTPs. All messages in either direction are passed by the Remote Proxy, and the ESA executable is launched by the Proxy.

To run the remote proxy on an external system host, you should:

1. Download and install the latest Java version, e.g. JRE 1.6, from www.java.com to the XS host, if it isn't already installed.
2. Fill in the command line field in the Sync configuration GUI.
3. Download "esa.jar" by clicking the "Download ESA Remote Proxy" button in the Sync Configuration GUI.
4. Start the ESA Remote Proxy application by running the "java -jar esa.jar" command on the XS host.

If you are running a third-party command-line ESA remotely, please make sure the ESA command-line executable can be launched by the ESA Remote Proxy application. In particular, check "execute" permissions and PATH. The ESA Remote Proxy should then launch the ESA application via the command, specified in the Sync Configuration.

ESA Remote Proxy doesn't change the current directory. If you need the current directory to be different from the one containing the esa.jar file, use a path to it within the start up command:

```
java -jar <path>\esa.jar
```

For example:

```
java -jar c:\ewsync\esa.jar
```

When you download esa.jar from the Sync Configuration GUI, it will have the necessary configuration parameters such as ESA type, External System ID and polling period predefined within the jar file. Those parameters can also be changed using command-line arguments in the java -jar command:

Command-line argument	Description
-ewhost	Host (and port) where Agiloft is run. This parameter is useful if a server

Agiloft Reference Manual

Command-line argument	Description
	"outer" name differs from its "intranet" name
-id	External System ID, see below
-pollprd	Polling period, seconds
-esatype	ESA Name. Valid names are "EXCHANGE" "FILE" "THIRD_PARTY_COMMAND_LINE"

ESA Settings GUI

This appears on clicking the Connect button which appears in the Sync Configuration GUI after you click Create:

The tab is completely ESA-specific. For a File-Directory ESA it looks like this:

General **ESA Settings**

Local path to the file directory to sync. Multiple directories can be given, separated by local system path separator (";" on Windows, ":" on Unix).

Format of the data files. Can be "Properties" for "plain" Java Properties format (property=value lines) or "XML" for XML "`<data><field-name>field-value</field-name>...</data>`" Java Properties format

Comma-separated list of fields. A field is a single mappable data element. Depending on the format, it may designate a property name or an XML element name.

*Directory path
e:\temp

*File Format
 Properties
 XML

*Field list
AgiloftID, Summary

For the Exchange ESA it looks like this:

Label	Value
Microsoft Active Directory host	*Host localhost
Microsoft Active Directory port	*Port 389
MASD user to perform CRUD action over contacts	*User Administrator
MASD user password	*Password
MASD domain	*Domain domain.local
MASD base DN	*dn dc=domain,dc=local
Organizational Unit which contains the Contacts	OU Public Contacts
Global Address List which contains the Contacts	gal

The parameter set is ESA-specific, but usually, you will be at least prompted to enter the external system connection attributes here.

After you press Next, the ESA will usually have enough information to connect to the external system. The connection will be made for some but not all ESA's. In cases when it does connect to the XS, it may open DB connections, make some queries, verify, and /or provide additional parameters to manage, based on what it finds.

For example, a "basic" set of parameters may have host and login/pass and "advanced" set may include selected tables to synchronize. If names of tables are extracted from the remote system, ESA will have to connect to it first.

Clicking "Next" takes you to the next, Table-Structure Mapping, GUI. The buttons will change once again and the Table-Structure mapping tab will be shown.

Table-Structure Mapping GUI

This option lets you map to External System structures.

Subtype mapping

Subtype Table	External Structure
Acct Period	Not Mapped <input type="button" value="Map"/>
Agiloft Request	Not Mapped <input type="button" value="Map"/>
Asset	Not Mapped <input type="button" value="Map"/>
Billing	Not Mapped <input type="button" value="Map"/>
Calendar	Not Mapped <input type="button" value="Map"/>
Campaign	Not Mapped <input type="button" value="Map"/>
Change Request	Not Mapped <input type="button" value="Map"/>
Chat Log	Not Mapped <input type="button" value="Map"/>
Company	Not Mapped <input type="button" value="Map"/>
Contact	Not Mapped <input type="button" value="Map"/>
Employee	Not Mapped <input type="button" value="Map"/>
End User	Not Mapped <input type="button" value="Map"/>

Some ESA's may ask for additional parameters after connecting to XS. To set them, press "Back" button, so returning to ESA Settings.

In this GUI, you select an Agiloft table to be synchronized, from the list, enter the corresponding external structure name (which you must know) and click the Map button at the extreme right against that table name. The Field Mapping Wizard (see below) will then be shown and used. If the named structure in the XS can be mapped successfully against the selected Agiloft table, the Map button is replaced by Edit Mapping and Unmap buttons, enabling you either to alter or cancel the mapping.

This process is done for each table you select.

Field Mapping Wizard

Field Mapping
Filters

Next Finish Cancel

This option lets you specify a field, holding record modification time in your EW subtype.

EW Modification Time Field

This option lets you to control whether table records are synchronized based on record timestamps or just replaced in any system

Table update type

Synchronize
 Export
 Import

You can restrict sync operations for these subtype and structure.

Allowed operations

EW Create External Create
 EW Update External Update
 EW Delete External Delete

When a new External Record is to be created during Sync, it should get some new External ID. If you want External System (or ESA) to auto-assign this ID, leave "External System generates ID" checkbox checked. Alternatively, you may specify an optional prefix and a subtype column (must hold unique values) to be used as a new ID.

ID Generation

External System generates ID

External ID Prefix:

External ID Value column:

This option lets you map EW fields and External System record fields.

Field Mapping

Use strict match for identification

Some fields might be marked as "identifying". They are used to establish initial relation of records, which are not in sync yet. This is important for the first time synchronization and if information is eventually added into both systems. For example, "Email" is a good "identifying" field if synchronizing contacts. You may set

External Field	EW Field	Update in EW	Update in External	Identifying
*Comment ID	<input type="text" value="Not Mapped"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
From	<input type="text" value="Not Mapped"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Message	<input type="text" value="Not Mapped"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

The entries are explained in the left hand column, they are summarized here:

Agiloft Modification Time

Usually an Agiloft table has a "Date Updated" Datetime field, but not invariably. It may have several DateTime fields, and in this GUI the user can select one of them to hold the modification timestamp.

Allowed Operations

For the Agiloft Table or the corresponding XS structure, the named operations can be prevented by unchecking the checkboxes shown.

Field Mapping

Field names in the external structure, provided by the ESA, are to be matched with the corresponding field-names in the Agiloft Table by the user.

The checkbox "Use strict match for identification" decides how fields marked as "Identifying" will be used to determine the match between records in the Agiloft Table and the External Structure records. See "Pairing" below, and "Sync Operations - Record Peering".

This wizard will look slightly different based on the fields in the Agiloft table and External Structure, but in any case, you will have to select an Agiloft "counter-part" for each External field.

The two checkboxes following the field list specify updating of the Agiloft Table values and of the XS structure values for that field in the selected Table-Structure pair.

Pairing

The "Identifying" checkbox should be ticked for one or more fields which are expected to have values unique to each record. At least one of the fields marked as Identifying must have a value unique for each record in the table or structure concerned. These fields determine the 1-1 mapping of the Agiloft table records to the XS structure records, explained under "Sync Operations - Record Peering" below.

The Identifying field is not necessarily an ID field. Moreover, if there is an auto-incrementing / GUID ID field as in Agiloft, it is generally a bad idea to have this field marked as identifying, With 99% probability, there will be no match for this ID value in another system, or even worse, a wrong match.

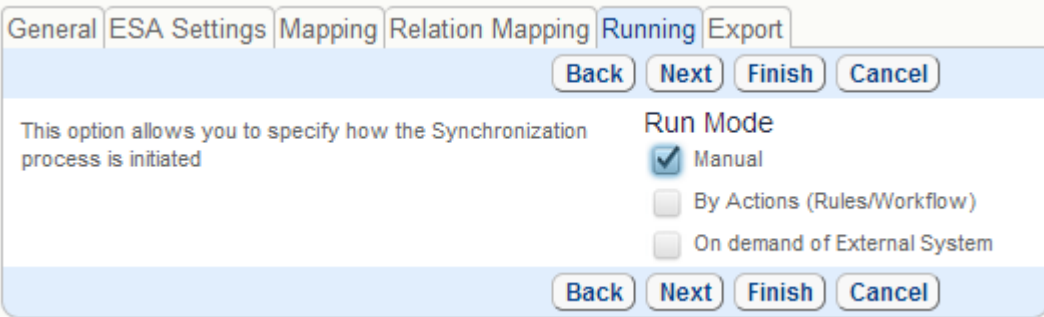
Good examples of Identifying fields are Email and Full Name when synchronizing contacts and "Summary" when synchronizing cases. Those fields are likely to identify a logical match to Sync records in both systems.

Completion

Clicking "Finish" either in this Wizard or in the Table-Structure Mapping GUI completes a Sync Configuration.

Run Mode

The "Running" tab on the Sync Configuration GUI brings up this GUI:



The screenshot shows a GUI window with tabs: General, ESA Settings, Mapping, Relation Mapping, Running (selected), and Export. Below the tabs are buttons for Back, Next, Finish, and Cancel. The main content area has a text box on the left: "This option allows you to specify how the Synchronization process is initiated". To the right is the "Run Mode" section with three radio buttons: "Manual" (checked), "By Actions (Rules/Workflow)", and "On demand of External System". At the bottom are another set of Back, Next, Finish, and Cancel buttons.

Run Mode

This specifies the ways a synchronization can be initiated:

- Manually, through the Sync option in a Table view.
- Actions (among background Agiloft activities, including running syncn upon a record change or by a schedule).
- On demand, of the ESA.

By default, all three "triggers" are allowed.

Conflicts

The choice here (1 out of 4) specifies what to do if a record has been modified (to different values) in both systems. The effect is described under "Conflict Resolution" below.

Sync Operations

Sync Operations

Scope

The "structure" with which an Agiloft table can be synchronized may be a table, a folder or anything else which logically groups records, and can be presented or derived by the ESA from the external system. All records within a single "Structure" must have the same set of fields.

As well as records, it is also possible to synchronize relations between external records. Those are used to update Linked Fields in Agiloft.

Sync also supports "Collection fields", which are something in between normal data field, holding single value and a relation to other table. Collection fields hold multiple values, which might be complex objects, with multiple fields. These objects can be mapped to a separate Agiloft table, just as related records for relations. But, unlike a relation, which treats all records independently, these records are treated as a part of their "master" record. If any of them is modified, a "master" record is considered to be modified.

Sync Configuration

In summary, the Sync Configuration Wizard creates a record in Agiloft holding:

- The external system to syncn with
- The tables to be synchronized and their corresponding "Structures"
- The fields to be synchronized within a Table-Structure pair
- The relations to be synchronized
- The directions (Agiloft to XS, XS to Agiloft or both) of updating
- The command that launches the ESA
- The "identifier" fields Sync will use to identify matching records

If a Sync Configuration is modified after it has been run, all previous peering of Agiloft records with XS records is lost, and the next run will be an entirely new synchronization.

Record Peering

For every record in an Agiloft table or XS structure subject to a new syncn, and for every such record created since the last syncn, a corresponding record in the other system is determined, if it exists, by matching the Identifying fields (checked in the

Field Mapping Wizard) from the two systems. If there's no such record, one is created. The two records are held by Sync in an Agiloft ID - XS ID pair, thus "peering" the Agiloft record with the external record. Logically, Sync treats the pair as a single data record, having Agiloft and external views.

If a new record is created manually in both systems, the Identification process will prevent Sync from duplicating them.

Matching

There are two algorithms for matching:

- Best match: If there are 3 identifying fields, Sync will first try matching all 3. If there is no match, it will try matching by any 2 of 3. If there is still no match, it will match by any 1 of the 3 fields.
- Exact match: If there are 3 identifying fields, Sync will first try matching all 3. If there is no match it will create a new record.

The algorithm is selected in the Field Mapping GUI - "Use strict match for identification" checkbox. The choice of exact matching may prevent matching if any identifying field has non-unique values.

After the peering is established, it will not be cleared until either one of these conditions is met:

- The records are deleted in both systems
- The peering information is reset by using "Reset Records Peering" button on "General" tab of Sync Configuration edit GUI.

Conflict Resolution

If both records of a pair have been changed since the last syncn, to different values, one of them should be updated. The "conflict" can be resolved either:

- By choosing the value in one of the systems (Agiloft/XS) to preserve.
- Or by preserving the most recent value, according to the timestamps

This choice is made in the Run Mode GUI - see above.

There are no "merges" for a record; all fields in a record are always updated from a particular source record, but you can choose which fields to synchronize, so it is not necessary for the whole record to be overwritten.

Dual Synchronization Problem

If two or more external systems are kept "in syncn" with a given Agiloft Table - which means using two or more different Sync Configurations - and if conflicts are resolved by preserving the most recent value, the results are affected by the order in which the synchronizations are run.

Suppose an updatable field in the 3 systems XS A, XS B and Agiloft has the values and timestamps shown, and we synchronize A with Agiloft, and then B with Agiloft at 4p.m:

XS A p (3p.m) => p (4p.m)

XS B q (2p.m) => p (4p.m)

Agiloft r (1p.m) => p (4p.m)

The first update changes Agiloft because 3p.m > 1p.m, and the second changes XS B because 4p.m > 2p.m

But if we synchronize B first, and then A, the effect is:

XS A p (3p.m) => q (4p.m)

XS B q (2p.m) => q (4p.m)

Agiloft r (1p.m) => q (4p.m)

The first update changes Agiloft because 2p.m > 1p.m, and the second changes XS A because 4p.m > 3p.m

This is because there is only one timestamp on each value in a pair, whether it results from syncn or user updating.

Part 2 - ESA Development

ESA Development Outline

An ESA receives and parses XML messages from Agiloft, either directly or from the remote proxy, processes them, usually accessing XS data, frames an XML response and passes that back to Agiloft.

An addition is necessary for HTTPs ESA's, which can initiate calls beginning a syncn to the Agiloft Helper API. A command-line ESA can't do that, because it is only run by Sync when Agiloft decides that it is a time to synchronize. Such an ESA can't pass a message to Helper API outside of the syncn cycle.

Pseudocode

Pseudocode for an ESA is:

```
Boolean quit = false;
Do
{
String xmlMsg = readXmlMessageFromEw();
Message parsedMsg = parseXmlMessage(xmlMsg);
// Dispatch message
Message result;
Switch (parsedMsg.type)
{
Case getModified:
{
// Find all modified records ...
result = new RecordListResultMessage( modified records);
break;
}...
Case update:
{
// Update a record
result = new RecordResultMessage( updated record);
break;
}
...Case release:
{
// Release all resources...
quit = true;
}
Default:
{
result = new ExceptionResult("Unkno wn operation");
}
}
String xmlResponse = xmlizeResponse(result);
sendResponseToEw(xmlResponse);
} While (Not quit)
```


XML Use

All messages from or to Agiloft must be well-formed XML documents, starting with an XML declaration and a root <sync> tag. A complete, formal XML Schema for XML messages can be found in "XML Messages Schema". This ensures that Agiloft and the ESA will "understand" each other.

Calls to ESA - Schematic

All messages are framed by <sync> and <esa-call> elements and are pre-pended by an XML declaration. Actual messages look like this:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<sync xsi:noNamespaceSchemaLocation="sync.xsd"
      :xsi="http://www.w3.org/2001/XMLSchema-instance">
  <esa-call call-id="12345">
    <startSync>
      <external-system-id>My-External-ID</external-system-id>
    </startSync>
  </esa-call>
</sync>
```

Results from ESA - Schematic

All result messages should also include the XML declaration and be wrapped into <result> tags, like this:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<sync xsi:noNamespaceSchemaLocation="sync.xsd"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <result response-to="12345">
    <value>1.0</value>
  ...</result>
</sync>
```

Calls to Helper API - Schematic

All these messages must be framed by <sync> and <api-call> elements and are prepended by the XML declaration. Actual messages look like this:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<sync xsi:noNamespaceSchemaLocation="sync.xsd"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```



```
<api-call call-id="12345">  
  <startSync>  
    <external-system-id>My-External-ID</external-system-id>  
  </startSync>  
</api-call>  
</sync>
```

Results from API

These messages are identical in form to Agiloft - ESA call results.

Text in ***bold italics*** above is to be replaced in your own code.

Calls to and from an ESA

All communications between Agiloft and the ESA are in the form of XML message exchange, of two kinds:

- Calls (Method Calls)
- Results (including exceptions)

Most calls come from Sync to the ESA, but the ESA can respond by a call to the Agiloft Helper API.

A remote Command-line ESA can never initiate a syncn cycle, but an HTTPs one can do so, by a call on the Agiloft Helper API. In this case the initial call is *startSync(HelperAPI)*. Agiloft then responds with a *startSync(ESA)* and other calls to the ESA, finishing communication by a void result of *startSync(Helper API)*.

Call ID

Every call has a mandatory "call-id" attribute. This attribute should be set by the caller to a positive integer value. It is recommended (but not obligatory) that this value would be incremented on each call. A result of a call has a "response-to" attribute, filled by the call-id value of the call, to which this result belongs. This simple mechanism allows monitoring of message flow and detection of communication (or application) failures.

STDIO

This type of third-party ESA uses standard input and output streams, stdio. In this case, the ESA developer provides his ESA as an executable, command-line application. It need not be a compiled native application; it could even be a script. The two requirements are:

- It must be runnable always using the same command-line
- It will not print anything except XML Sync messages to the standard output

Line breaks

Although the XML standard says little about line breaks, standard console I/O is line-based. To handle this difference, every `<sync>` and `</sync>` tag which delimit all XML syncn messages must be placed on a line of its own.

Cycle

The typical ESA cycle using STDIO transport is

1. Read message, using standard input:
 - a. Read line. This must always be the XML declaration ("`<?xml...?>`")
 - b. Read line. This must always be "`<sync>`"
 - c. Read and accumulate all lines before getting "`</sync>`" line.
 - d. Parse what was read as XML message
2. Process message.
3. Write response using standard output
 - a. Write XML declaration
 - b. Write "`<sync>` " line.
 - c. Write XML message body (a "`<call>`" or "`<result>`" element).
 - d. Write "`</sync>`" line.
4. If the message was NOT release, goto 1.
5. Terminate.

For ESA's using STDIO, Sync is responsible for starting the ESA application when necessary.

HTTP(S)

This type of message transport does not require the ESA to be an application executable by command, and is adapted for Internet communication.

URLs

There are 3 URLs, used for HTTP transport. If the Agiloft GUI runs at `https://www.example.com/gui2`, these URLs will be:

`https://www.example.com/gui2/sync/connect`
`https://www.example.com/gui2/sync/disconnect`
`https://www.example.com/gui2/sync/message`

Every request to any URL must contain an `extsysid=<externalSystemID>` parameter, where `<externalSystemID>` matches the External System ID field in the Sync Configuration.

Important: A third-party ESA, using HTTP transport must be aware of its externalSystemID.

HTTP response codes should always be examined to check whether transportation was successful. All parameters (GET and POST) should always be URL-encoded. Character encoding is UTF-8.

`connect` and `disconnect` URLs should be accessed when the ESA is about to start/stop receiving messages. These requests don't have any other parameters. The HTTP response to `connect` will be an `externalSystemID` value passed in the HTTP request. The ESA may check this in addition to the HTTP response code to ensure it has connected successfully.

The message URL can be accessed via a GET HTTP request to get the next message or via a POST HTTPs request to send a message. When GETting message, the XML message is returned as a HTTPs response. When POSTing, the message should be passed as the message HTTPs "form parameter".

Cycle

The typical ESA cycle for using HTTPs transport is:

1. GET `/sync/connect`
2. GET `/sync/message`
3. Process the message.
4. POST `/sync/message`

5. If the message was NOT "release", goto 2
6. GET /sync/disconnect
7. Wait for some period of time. The Sync Configuration setting "Polling period" is accessible as the "config.pollperiod" parameter which you may request from within sync cycle 1-6. Goto 1

HTTPs transport does not limit the way the ESA is actually implemented. It can be a command-line application, a service, a component within an application server, etc.

Starting and stopping this application and the exact mechanism by which the ESA obtains the External System ID value are outside the scope of this spec, but typically the application is installed just like any other piece of software and reads the ID from a command line parameter or configuration file.

Data Mapped to Snippets

To assist in the coding of messages, many kinds of data are mapped to XML snippets associated with the "XML Messages Schema" as follows:

External Records

ExternalRecords are mapped to XML snippets of the form:

```
<record id="id-value"  
old-id="old-id-value"  
modified="modificationtimestamp">  
<field name="field-name"> field value </field>  
...  
<field name="field-name">...</field>  
<relation id="relation-id">  
<related>related-object-id</related>  
...  
<related>related-object-id</related>  
</relation>  
...  
<relation id="relation-id">  
...  
</relation>  
<collection>  
<record> tags, describing collection members.  
</collection>  
</record>
```

External Structures

ExternalStructures are mapped to XML snippets of the form:

```
<structure name="name-value" screen-name="screen-name"/>
```

External Collections

ExternalCollections are mapped to XML snippets of the form:

```
<collection id="collection-id"  
screen-name="screen-name"  
required="boolean-value"  
updatable="boolean-value"  
updatableOnCreate="boolean-value"/>
```

External Fields

ExternalFields are mapped to XML snippets of the form:

```
<field name="name-value"
screen-name="screen-name"
type="xsd-type-name"
required="true/false"
identifying="true/false"
updatable="true/false"
updatableOnCreate=" true/false"
maxLength="maximum length of string representation">
<!-- If a field is of enumeration type, it should have type "ew:enum"
and contain one or more nested <enum-value> elements: -->
<enum-value>
<value>value</value>
<screen-name>
name to be used in mapping editor
</screen-name>
</enum-value>
</field>
```

External Relations

ExternalRelations are mapped to XML snippets of the form:

```
<relation id="relation-id"
related="related-structure-name"
screen-name="screen-name"
multiple="Boolean-value"
required="Boolean-value"/>
```

ESA parameters meta

the ESA parameters metadata are mapped to XML snippets of the form:

```
<esa-parameter
name="string-name"
type="single | radio | multi | expandable | xml"
required="true | false">
<label>Label to show above input</label>
<hint>Hint to show on the left</hint>
```

```
<item-type>
xsd:string | xsd:integer | xsd:double | xsd:Boolean
</item-type>
<default-value>
String-value | Integer-value | Double-value | Boolean-value
</default-value>
<!-- For parameters of type "radio", "multi" and
"expandable", a number of nested <enum-value>
elements should follow: -->
<enum-value>
<value>value</value>
<screen-name>
name to be used in mapping editor
</screen-name>
</enum-value>
</esa-parameter>
```

Cursors

Cursors are mapped to XML snippets of a form:

```
<cursor id="cursor-id"
page-size="records-per-page"
number-of-pages="number-of-data-pages"/>
```


Other data types

Exceptions

Exceptions are reported using a `<result>` tag with a nested `<exception>` element.

```
<exception type="general | record | configuration|alreadyconfigured
|optlockfailed | concurrentdelete">
<message>Exception message to be shown to user</message>
<trace>Exception message to be put to logs</trace>
</exception>
```

Exceptions of type "alreadyconfigured" should also have nested

```
<configured-to>External ID, to which ESA is configured to</configured-to>
```

tag.

Exceptions of type "optlockfailed" and "concurrentdelete" should also have nested

```
<external-id>record ID</external-id>
<modified-at>record ID</modified-at>
```

tags.

`<trace>` should contain exception stack trace or some other diagnostic information (FILE/LINE, etc) which would allow ESA developer to investigate the problem better.

Locale

Some messages have a "Locale" argument, which is a locale and country code, as described in the ISO-639 and ISO-3166 (`<language-code>`- `<country-code>`-`<variant>`). Examples are "en_US", "ru_RU" and "pt_BR". If the ESA doesn't support the required locale or doesn't support localization at all, it should return an American English label and hint.

Problem Areas

A few message or data types need special discussion.

Deletions

For many systems, tracking deleted records is problematic. The Agiloft HelperAPI can simplify this task. If you can't easily track deletions in the external system you can either:

- Enumerate all record IDs known by Agiloft and check whether those records actually exist at the moment.
- Or report "deletion events" to Agiloft when deletion happens in the external system. The latter is usable if you can "hook in" into the external system to get deletion notifications somehow.

A more sophisticated approach to ID enumeration, with better performance is possible if your system uses never repeating, always increasing auto-incremented IDs. If this is the case, you may process the countRange message and send a detectDeleted message to the Agiloft HelperApi. It will then apply a dichotomy method (binary division) to find deleted IDs. This will be considerably faster than comparing known and currently existing IDs one-by-one, especially if you can have a fast implement count($A < id < B$) operation and the number of deleted records is not large.

Transactions

Sync is transaction-less, because not all external systems support transactions or expose them over integration APIs. If an external system requires transaction usage, the ESA should behave as it would with "auto-commit on" in the SQL meaning, i.e. it should execute every operation in a separate transaction.

Collections

Collection fields may hold complex objects, which can be mapped to Agiloft tables. For this reason, getField, getRelations and getCollections calls should accept both structure names and collection ids.

An ESA must ensure that structure names and any collection ids use different name spaces.

Locking (Clashing Modifications)

To deal with concurrent data modifications, Sync employs an "optimistic locking" strategy. When the ESA is asked to modify or delete an external record, it is given a "last seen" timestamp. If the ESA detects that the record was modified after that time, the ESA should respond with an OptimisticLockingFailure exception and not modify or delete the record. If an external system uses the "pessimistic locking"

approach internally, the ESA should wait until the lock is granted first, or else report a record update failure if waiting for the lock is not reasonable.

Currently, "delayed" updates due to locks are not supported, but this feature is likely to be added in the upcoming releases because it may speed up syncn appreciably. It is a good idea to make your ESA ready to throw some exception when a pessimistic lock is detected - this might be employed in the next Sync release.

When a record update/create/delete error occurs, the ESA should respond with `EsaRecordException`. Such exceptions indicate a single record related error, probably recoverable in the future and do not stop the syncn cycle. On the next cycle, Sync will try to re-synchronize the erroneous records.

Cursors

In some setups, `getModified/getDeleted` may return a huge amount of data. It may be inefficient or even unfeasible to transfer and process such traffic in one call.

To overcome this, Data Cursors can be employed. A Data Cursor is a paginated stream of data. When requested, cursor "core" information (ID, page number) is transferred from the ESA to Agiloft. Agiloft requests actual pages data from the cursor passing "readDataPage" messages supplying cursor ID and page index.

Cursors are a more advanced technique than the simple return of all data. Their support is optional in the ESA. For the `getModified` function, Agiloft first sends `getModified`, which has a chance to return all data at once. If and only if the `getModified` response is "USE_PAGES" string (not an empty result), Agiloft sends `getModifiedPaged`, which returns a cursor. The ESA may choose the transferring approach in runtime, depending on the amount of the data.

When a data cursor is fully read, the cursor is closed. However, due to possible communication failures, this may not happen. The ESA should retire cursors in 15 minutes since the last "readDataPage" or "leaseCursor" message.

ESA Parameters

The ESA is likely to have configuration parameters, such as "external system host", "login", "password", etc. While the ESA may maintain them by its own configuration mechanisms, it is often desirable to have them in one place, to simplify administration and maintenance.

For this purpose, Sync may store some "ESA parameters" within the Agiloft database, providing a simple GUI for changing them. To do this, Sync "asks" ESA for the names and types of parameters to manage. The ESA provides parameter meta-data (type, default value, required / not required) and Sync shows them in the configuration GUI, "ESA Settings" page.

ESA may later request its parameter values during the syncn cycle, using parameter names. See `getParameter` message. Though the number of parameter types is limited (strings, booleans, etc), they are likely to cover most of the ESA configuration needs and simplify administration greatly. If, for example, your ESA will require some configuration file, it is a good idea to put the local path to it in an ESA parameter.

Parameter names, starting with "config." are reserved for getting Sync Configuration settings. Currently, the following are available:

- `config.pollperiod` - polling period
- `config.commandline` - command line to invoke a third-party command line in the ESA.
- `config.esatype` - ESA type name.

It is likely that only `config.pollperiod` is of interest for ESA developers, and this only for HTTPs ESA's.

Timestamps

The time resolution for Sync is 1 second; all timestamps are truncated to the closest second. All timestamps passed to or received from the ESA are in UTC (GMT, Greenwich) time. It is the responsibility of the ESA to translate external system timestamps into UTC, taking time zones and daylight savings into effect.

Sync may compute the system clock difference between the external system (ESA) machine and the Agiloft host machine. To do so, Sync passes a `getCurrentTime` message to the ESA. If the ESA wants the clock difference to be taken into account, it should respond with the current external system UTC time. Alternatively, the ESA may respond with an empty result, setting clock difference to zero.

If you are sure that the Agiloft host and external system host have their system clocks synchronized, for example via Network Time Protocol, NTP, it is a good idea to turn clock difference computation off by responding with an empty message. Since the Sync time resolution is less precise than NTP, it will never get better results than it could if the system clocks are really synchronized.

Due to limited time difference precision, it is possible that some records may stay unsynchronized. Though the chance of this is very low, it is always better to have machine clocks synchronized and time difference detection off.

If you are unsure that clocks are synchronized, for example, your ESA can be deployed on a number of machines, you should run `getCurrentTime`. The time difference calculation works best if the ESA actually measures time in the middle of a "message delivery + message processing + message delivery back" loop, but in practice, it is fine to just measure and respond as fast as you can.

The Time difference calculation is performed on each Sync cycle, so if system clocks are corrected between synchronizations, Sync will work correctly. It is however, presumed that system clocks are not changed during synchronization.

"Last seen" timestamps are always passed just as they were received from the ESA last time and thus are not affected by system clock changes.

Calls to ESA

Tables below list and describe all calls made by Sync to an ESA, with the expected results - not including exceptions.

String startSync(String externalSystemID)

Starts a synchronization. This is the very first method, called during Sync. Return value is the Sync protocol version, supported by the ESA. Currently it must be "1.0"

Input XML Example

```
<startSync>
  <external-system-id>
    String-value
  </external-system-id>
</startSync>
```

Output XML Example

```
<value>1.0</value>
```

void leaseSession()

Resets session timeout counter (if the ESA has any).

Input XML Example

```
<leaseSession>
</leaseSession>
```

**Set <ExternalRecord>
getModified(String structure,
Timestamp after)**

For some systems, this may result in a great amount of data to be transferred, especially for the initial sync.

If the ESA predicts such traffic, it should return NULL from this method (NOT an empty set!). In this case, Agiloft will use the "getModifiedPaged()" method.

The ESA may decide how to transfer data at runtime depending on the amount. In any case, Agiloft will call "getModified()" first and only call "getModifiedPaged()" if "getModified()" returns NULL

Input XML Example

```
<getModified>
<structure>
  structure-name
</structure>
<after>
  timestamp-value
</after>
</getModified>
```

Output XML Example

A number of ExternalRecord snippets (<record> tags)

**Cursor<ExternalRecord>
getModifiedPaged(String structure,
Timestamp after)**

Gets modified records in a paged manner. See "getModified()" description. This method is only called if "getModified()" call returned NULL. The Cursor object returned contains information about the number of data pages, data page size and so on.

Input XML Example

```
<getModifiedPaged>
<structure>
  structure-name
</structure>
<after>
  timestamp-value
</after>
</getModifiedPaged>
```

Output XML Example

A number of ExternalRecord snippets (<record> tags)

Set<String> getDeleted(String structure, Timestamp after)

Gets IDs of records, deleted in external system after given timestamp. For some systems, this may result in a large amount of data to be transferred. If the ESA predicts such traffic, it should return NULL from this method (NOT an empty set!). In this case, Agiloft will make use of "getDeletedPaged()" method. The ESA may decide how to transfer data at runtime depending on the amount. In any case, Agiloft will call "getDeleted()" first and only call "getDeletedPaged()" if "getDeleted()" returns NULL.

Input XML Example

```

<getDeleted>
  <structure>
    structure-name
  </structure>
  <after>
    timestamp-value
  </after>
</getDeleted>

```

Output XML Example

```

<id>id-value</id>
...
<id>id-value</id>

```

Cursor<String>
getDeletedPaged(String structure,
Timestamp after)

Gets deleted records in a paged manner. See "getDeleted()" description. This method is only called if "getDeleted()" call returned NULL. The Cursor object returned contains information about the number of data pages, data page size and so on.

Input XML Example

```

<getDeletedPaged>
  <structure>
    structure-name
  </structure>
  <after>
    timestamp-value
  </after>
</getDeletedPaged>

```

Output XML Example

```
<cursor> snippet
```

Set<T> readDataPage(Cursor<T> cursor, int pageIndex);

Gets data from a cursor. This method is used to actually access the data in the cursor.

Input XML Example

```
<readDataPage>  
<cursor-id>  
  cursor ID  
</cursor-id>  
<page-index>  
  data page index  
</page-index>  
</readDataPage>
```

Output XML Example

A number of ExternalRecord snippets (<record> tags)
OR <id>id-value</id>, depending on the cursor type

void leaseCursor(Cursor <T> cursor)

Resets cursor expiration timer.

Input XML Example

```

<leaseCursor>
<parameters>
<cursor-id>
  cursor ID
</cursor-id>
</leaseCursor>

```

void closeCursor(Cursor<T> cursor)

Closes the cursor. This call indicates that Agiloft doesn't need the cursor anymore and guarantees that "readDataPage" will not be called again for this cursor.

Input XML Example

```

<closeCursor>
<cursor-id>
  cursor ID
</cursor-id>
</closeCursor>

```

ExternalRecord read(String structure, String id)

Reads a single record.

Input XML Example

```

<read>
<structure>
  structure-name
</structure>
<id>id-value</id>
</read>

```

Output XML Example

```
ExternalRecord snippet
```

ExternalRecord create(String structure, ExternalRecord values)

Creates new record in external system, returns creation timestamp

Input XML Example

```
<create>
  <structure>
    structure-name
  </structure>
  An ExternalRecord snippet
</record>
</create>
```

Output XML Example

```
ExternalRecord snippet
```

Timestamp update(String structure, Timestamp lastSeen, ExternalRecord values)

Updates the record in external system, returns update timestamp. Only does the update if the record is not modified since "lastSeen" timestamp. If the record is modified, throws exception (see below)

Input XML Example

```

<update>
<structure>
  structure-name
</structure>
<last-seen>
  timestamp-value
</last-seen>
An ExternalRecord snippet
(<record>)
</update>

```

Output XML Example

```

      <value>timestamp-
value</value>

```

void delete(String structure, Timestamp lastSeen, String pk)

Deletes the record in external system, returns true. Only does the deletion if the record is not modified since "lastSeen" timestamp. If the record is modified, throws exception (see below).

Input XML Example

```

<delete>
<structure>
  structure-name
</structure>
<last-seen>
  timestamp-value
</last-seen>
<id>id-value</id>
</delete>

```

**Set<ExternalStructure>
getStructureList()**

Gets a list of external data structure names. "Structure" is a logical data unit, mappable to an Agiloft table; a table, a folder, a class (in OODB) or any other type of data grouping.

Input XML Example

```
<getStructureList>  
</getStructureList>
```

Output XML Example

External Structure snippet

**Set<ExternalRelation>
getRelations(String
structureOrCollection, String
locale)**

Gets a list of relations (links) from the given structure. This should only include relations, "navigable" from this object. A "Locale" argument can be used to localize relation screen names.

Input XML Example

```
<getRelations>  
<structure>  
  structure-name  
</structure>  
<locale>  
  lang-value  
</locale>  
</getRelations>
```

Output XML Example

A number of <relation> snippets.

**Set<ExternalCollection>
getCollections(String
structureOrCollection, String
locale)**

Gets a list of collection fields for the given structure. A "Locale" argument can be used to localize relation screen names.

Input XML Example

```
<getCollections>
<structure>
  structure-name
</structure>
<locale>
  lang-value
</locale>
</getCollections>
```

Output XML Example

A number of <collection> snippets.

**Set<ExternalField>
getFieldList(String
structureOrCollection, String
locale)**

Gets a list of fields, for a given structure. ID and "ModifiedAt" fields should not be included.

Input XML Example

```
<getFieldList>
<structure>
structure-name
</structure>
<locale>
  lang-value
</locale>
</getFieldList>
```

Output XML Example

A number of ExternalFields snippets (<field> tags)

List<ESAParameter> getParametersMeta(String locale)

Gets localized ESA Parameters meta-data.

Input XML Example

```
<getParametersMeta>
<locale>String-value</locale>
</getParametersMeta>
```

Output XML Example

A number of <esa-parameter> tags.

void endSync()

Returns Bit-ORed constants, restricting synchronization run modes:

- 1 - Run Manually allowed
- 2 - Run by Sync Actions allowed
- 4 - Run by external system request allowed.

For example, 6 (2 bit-OR 4) would designate that the ESA supports (non-interactive) synchronizations by both Agiloft and external system requests

Input XML Example

```
<getAllowedRunModes>
</getAllowedRunModes>
```

Output XML Example

```
<value>
  integer value
</value>
```

String configure(String externalSystemID, boolean force)

Configures the ESA to use with an Agiloft instance. This method is called on the ESA when a Sync configuration is created (this also allows a check on ESA availability). The implementation of this method depends on the ESA. Simple ESAs, which don't need to communicate (call) Agiloft back (i.e. neither use Helper API outside of syncn cycle nor initiate Sync by external system requests) may simply ignore this call and return the passed

externalSystemID. More complex ESAs should retain these values somewhere and use them to call Agiloft and to authenticate themselves (by externalSystemID). If such an ESA is already configured (already has an associated ExternalSystemID) it must return the stored value and only replace it by the new parameters if parameter "force" is true.

Input XML Example

```
<configure>
  <external-system-id>
    String-value
  </external-system-id>
  <force>
    Boolean-value
  </force>
</configure>
```

Output XML Example

```
<value>
  String-value
</value>
```

int countRange(String idMin, String idMax)

Returns a number of external system records, whose (external) ID is in the given range numerically or alphabetically, whichever is applicable. You should only implement this method if your ESA calls "detectDeleted" HelperApi method.

Input XML Example

```
<countRange>
  <min-id>
    String-value
  </min-id>
  <max-id>
    String-value
  </max-id>
</countRange>
```

Output XML Example

```
<value>integer value
</value>
```

boolean needSyncAgain()

Sometimes, the ESA may decide there is a necessity to re-run synchronization immediately after sync finish. An example is if update actually updates more than 1 record, so data are changed in a broader scope than the Sync subsystem expects. In this case, the ESA should return true from this method.

Input XML Example

```
<needSyncAgain>
</needSyncAgain>
```

Output XML Example

```
<value>boolean value
</value>
```

Date getCurrentTime()

Gets current UTC time of a remote system, if available. Sync uses this method to compute the time shift between Agiloft server and the external system. The shift is taken into account for all datetime "control" values, passed to the ESA (getModified(), record.modifiedAt, lastSeen timestamps, etc). Note: This shift is not applied when converting record date. Datetime or time fields.

Input XML Example

```
<getCurrentTime>  
</getCurrentTime>
```

Output XML Example

```
<value>timestamp value  
</value>
```

Calls from ESA to Helper API

Tables below list and describe all calls made to Sync by an ESA, with the expected results - not including exceptions.

void startSync(String externalSystemID)

This method triggers Sync with an external system, identified by externalSystemID. The method can be used (by ESA or any other application, if using HTTPs transport to send a message to Helper Api) to start synchronization, based on some event in the external system.

Input XML Example

```
<startSync>
<external-system-id>
  String-value
</external-system-id>
</startSync>
```

Void trackRecordDeletion(String externalSystemID, String externalStructure, Timestamp time, String id)

Places a record deletion entry in the internal log associated with this externalSystemID. A timestamp argument can be useful if the ESA chooses to batch reports of record deletions. However, if some deletion reports are delayed, the ESA should flush them as soon as it receives any call on the ESA interface, prior to actually processing the call

Input XML Example

```
<trackRecordDeletion>
  <external-system-id>
    String-value
  </external-system-id>
  <time>
    Timestamp-value
  </time>
  <structure>
    structure-name
  </structure>
</trackRecordDeletion>
```

Set<String> enumerateKnownIDs(String externalSystemID, String externalStructure, Timestamp knownBefore)

Asks Agiloft to enumerate (external system) IDs, known by Agiloft at some moment. the ESA can then check whether records with such IDs exist and compose a list of deleted records.

Input XML Example

```
<enumerateKnownIDs>
  <external-system-id>
    String-value
  </external-system-id>
  <known-before>
    Timestamp-value
  </known-before>
  <structure>
    structure-name
  </structure>
</enumerateKnownIDs>
```

Output XML Example

```
<id>String-value</id>
```

```
...
<id>String-value</id>
```

Set<String> detectDeleted(String externalSystemID, String externalStructure, Timestamp after)

This is a helper method for deletion detection over auto-incremented IDs. The ESA interface must support an optional "countRange" method in this case.

Input XML Example

```
<detectDeleted>
  <external-system-id>
    String-value
  </external-system-id>
  <after>
    Timestamp-value
  </after>
  <structure>
    structure-name
  </structure>
</detectDeleted>
```

Output XML Example

```
<id>String-value</id>
...
<id>String-value</id>
```

boolean isKnownID(String externalSystemID, String externalStructure, String id)

Asks Agiloft whether it has a known peer for this external ID.

Input XML Example

```
<isKnownID>
<external-system-id>
  String-value
</external-system-id>
<structure>
  structure-name
</structure>
<id>String-value</id>
</isKnownID>
```

Output XML Example

```
Boolean-value
```

Object `getParameter(String externalSystemID, String name)`

Gets an ESA parameter value, stored on the Agiloft Server.

Input XML Example

```
<getParameter>
<external-system-id>
  String-value
</external-system-id>
<name>
  String-value
</name>
</getParameter>
```

Output XML Example

```
<value>
```


Value. The type depends on the
ESA parameter type.
</value>

...
There can be several <value> tags
within result

XML Messages Schema

[Download here](#)

Development Using Java

Agiloft, inc. will provide a number of support libraries to simplify third-party ESA development. Currently, the only libraries available are for the Java programming language.

To develop an ESA in Java, all you need do is to develop a class implementing a single ExternalSystemAdapter interface and write about 10 lines to construct your ESA using either STDIO or HTTP transport.

Reusable Classes

The Java ESA Example following contains a number of classes, which can be reused by any Java ESA. These classes are:

- ExternalSystemAdapter interface. Your ESA implementation must include a class, implementing this interface. In fact, this is the only piece of code you must write to get a fully working ESA, based on the example code.
- Message classes, residing in com.supportwizard.sync.interfaces.transport package. These classes encapsulate XML messages to be exchanged.
- Classes of com.supportwizard.sync.interfaces.transport.xml and com.supportwizard.sync.interfaces.transport.xml.jaxb packages (notably JaxbParser and JaxbSerializer), parsing and generating XML.
- StreamTransport class, implementing STDIO Transport
- HttpXmlEsaTransport class, implementing HTTP(s) Transport
- A set of message dispatching classes (JavaEsaInvoker, MessageDispatcher, etc), which are responsible for calling ExternalSystemAdapter when a message arrives, and marshalling the result back to the message exchange stream.

The only things needed to build an ESA using STDIO transport are:

- Write your implementation of ExternalSystemAdapter interface
- Replace the line "FileDirectoryEsa esa = new FileDirectoryEsa();" in the "main" class with a construction of your class.
- If you wish to compile and jar using the script "make.bat", you will have to add your package(s) to the javac (java compiler) call arguments.

You may reuse any classes of the example and handle messages in some other manner, if you choose, but we recommend the method above to facilitate incorporating future changes and fixes.

The Example

The example given is a command-line ESA, with an alternative transport class to make an HTTPs ESA, synchronizable with a set of files in some file system directory.

For the sake of clarity, some details of XML parsing, logging and message-to-method dispatching are omitted here. Full sources are available as a separate file (sync-example.zip).

File Directory External System

The XS that this example fits holds files in the Java properties file format, either "plain" or "XML properties". Data samples:

```
"Plain" properties:
#comment
fieldA=value A
fieldB=value B

"XML" properties:
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
  <comment>comment</comment>
  <entry key=" fieldA"> value A</entry>
  <entry key=" fieldB"> value B </entry>
</properties>
```

The particular type is to be defined by the ESA Parameter "format". The set of fields (property names) is defined by the ESA Parameter "fields".

To test ESA Compilation

- Install latest JDK 1.6 on your machine
- Unzip sync-example.zip to some directory (\$ESA_DIR\$)
- Run make.bat in that directory. This will compile the sources to \$ESA_DIR\$/out directory and build "fileesa.jar" JAR file.
- Run "run.bat < in.txt" to test the ESA. This will run the example command-line ESA and feed "in.txt" file to its standard input.

You should get an output such as:

Agiloft Reference Manual

```
E:\sync-example>run.bat < in.txt
<sync>
<result>2009-06-16T19:58:49.062+00:00</result>
</sync>

<sync>
<result></result>
</sync>
```

If you see this, you have compiled the example successfully.

Principal Classes

Two classes are central to this example:

First is `FileDirectoryEsa`, which is an ESA implementation class itself. Its methods are called on receiving corresponding messages and it may call the `HelperApi` to make a call back to the Agiloft server.

Second is `JavaExecutableEsa`, which is the "main" class of the command-line application. You may modify line 28 (`new StreamTransport(true, System.in, System.out)`), replacing it with (`new HttpXmlEsaTransport(externalSystemID, ewHost)`) to transform the example ESA into an independent ESA, connecting over HTTP. You will have to supply (or hardcode for a start) `ExternalSystemID` and Agiloft `hostname:port`.

Other classes in the example can be treated as a Java support library. You don't have to modify them and may use them as is in your own ESA.

To Test in Sync

To use this ESA in Sync configuration, do this:

- Uncomment and correct the first line of `run.bat` file. It should read "`@cd ESA_DIR`", where `ESA_DIR` is a directory where the example is unzipped.
- Select "Third-party (command-line)" in the Sync Configuration Wizard.
- In the same GUI, put the command "`cmd /c ESA_DIR\run.bat`" in the input box labelled "Command Line"..
- Set up the rest of a configuration.

Please note that the "`ESA_DIR`" in this guide is to be substituted with the real full path of the directory where the example is unzipped.

Java Example Code

External System Adapter interface

This is an interface which every Java ESA should implement.

[ExternalSystemAdapter.java](#)

ESA Implementation base class

This is a helper class. It is not mandatory for a Java ESA to extend it. Instead, a Java ESA may implement the ExternalSystemAdapter interface directly. However, most simple ESAs will benefit from subclassing that class.

[ExternalSystemAdapterBase.java](#)

ESA class

This is a File Directory ESA class itself.

[FileDirectoryEsa.java](#)

FileDataGate interface referenced in FileDirectoryEsa is a pure implementation class

[FileDataGate.java](#)

The example ESA uses FileDataGate interface implementation to access (plain and XML) property files. This is a FileDirectoryEsa implementation specific interface, not a Sync interface.

"Main" class for command-line app

"Main" class is quite small:

```

1  package com.supportwizard.sync.remoteesa;
2
3  import com.supportwizard.sync.interfaces.transport.*;
4  import com.supportwizard.sync.interfaces.transport.EsaReceiver;
5  import com.supportwizard.sync.interfaces.
        transport.invokers.JavaEsaInvoker;
6  import com.supportwizard.sync.file.FileDirectoryEsa;
7
8  import org.apache.log4j.Logger;
9
10 import javax.xml.bind.JAXBException;
11 import javax.xml.bind.JAXBContext;
12

```

Agiloft Reference Manual

```
13  /**
14   * @author Anatoly Kochubey
15   * @version $Id: $
16   */
17  public class JavaExecutableEsa {
18
19      private static final Logger log =
20          Logger.getLogger(JavaExecutableEsa.class);
21
22      public static void main(String[] args) {
23          try {
24              log.debug("Start JavaExecutableEsa");
25
26              FileDirectoryEsa esa = new FileDirectoryEsa();
27              JavaEsaInvoker invoker = new JavaEsaInvoker(esa);
28              EsaReceiver receiver = new EsaReceiver(invoker,
29                  new StreamTransport(System.in, System.out));
30              esa.setHelperApi(invoker.getHelperApi());
31              receiver.run();
32          } finally {
33              log.debug("Exit JavaExecutableEsa");
34          }
35      }
```

There are several essential things it does:

- ESA class instance is constructed(line 25)
- StreamTransport class instance, handling STDIO transportation (standard IO streams System.in, System.out are passed) is constructed (line 28).
- EsaReceiver is constructed and run (lines 27, 30). EsaReceiver "glues" together ESA implementation class and transport.

As you may see, to convert an example into full-featured ESA, you only have to provide implementation of the ExternalSystemAdapter interface and modify the line 25 (and probably 29) of JavaExecutableEsa class above.

Other sources, including XML parsing/generating, message dispatching, etc, are available as a reference within example.jar.

ACTIVITY LOGS

ACTIVITY LOGS SETUP

Click Setup/System/Configure Activity Log to define what statistics are kept of system usage and for how long they are retained

Setup → System







System

<p>Click Set KnowledgeBase Time to adjust the local time for the current <u>KnowledgeBase</u> relative to the default time provided by the server. ?</p>	<input type="button" value="Set KnowledgeBase Time"/>
<p>Click Configure Activity Log to define what statistics are kept of system usage and for how long they are retained</p>	<input type="button" value="Configure Activity Log"/>
<p>Click Manage Global Variables to launch the Global Variables wizard. Global variables store system values that can be used throughout the current KnowledgeBase. ?</p>	<input type="button" value="Manage Global Variables"/>
<p>Click Manage Table Variables to launch the Table Variables wizard. Table variables store formulas that can be used throughout the current KnowledgeBase.</p>	<input type="button" value="Manage Table Variables"/>
<p>Click Manage Choice Lists to create new <u>choice lists</u>, or edit and delete existing ones. ?</p>	<input type="button" value="Manage Choice Lists"/>
<p>Click Edit Page Headers to further integrate the Staff Interface with the look and feel of your company web site. ?</p> <p>The page header is available (and ability to edit it) when a Global variable called "Header Text" is available</p>	<input type="button" value="Edit Page Headers"/>
<p>Use this button to turn SOAP Web Services on or off Please check the log file for messages: EW_DIR/jboss/server/sw/logs/server.log</p> <p>Note: Any change in WS-Security parameters requires the Web Services to be [re-]enabled. The WS-Security parameters have to be set to the required values when [re-]enabling Web Services.</p> <p>Note: Web Services will not work unless you have an "Enterprise" or "Web Services" license. You can view your licenses using Setup/License/Manage Licenses</p>	<p>SOAP Web Services are currently Off</p> <p><input type="button" value="Enable WS"/></p> <p>version 2 ▾</p> <p>WS-Security enabled <input type="checkbox"/></p> <p>Timestamping TTL</p> <p>10</p>

You can click New, or Edit the one that's been already created.

Audit Rules

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

New Delete		Views  ▼   ▼		
<input type="checkbox"/>	Edit	ID ↓	Comment	Project
<input type="checkbox"/>		279	All logins in past month	97
<input type="checkbox"/>		278	All admin actions last month	97
<input type="checkbox"/>		277	All deletions last month	97

For example:

Finish Cancel

Enter a name that briefly describes the purpose of this log

***Comment:**

Choose the language in which the report should be generated

English ▼

Define or select a saved search for which user's actions should be logged

All ▼ Create Saved Search

Select which actions should be logged for the set of users defined above. An action is only logged if it is selected from this list and is executed by a user who meets the search criteria defined above

Select All Clear All

<input checked="" type="checkbox"/> Login	<input checked="" type="checkbox"/> Logout by User	<input checked="" type="checkbox"/> Logout by Timeout
<input checked="" 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	

Define for how long entries created by this rule should be retained.

1 Month ▼

If a rule is deleted, any records created by the rule are also deleted

Finish Cancel

In this wizard you can:

- Choose the language in which the report should be generated

Agiloft Reference Manual


- Define or select a saved search for which user's actions should be logged
- Select which actions should be logged for the set of users defined above. An action is only logged if it selected from this list and is executed by a user who meets the search criteria defined above
- Define for how long entries created by this rule should be retained. If a rule is deleted, any records created by the rule are also deleted
- You can run activity reports showing you how often specific reports on other tables have been run. An activity log entry will be created whenever the report is run manually, run according to a schedule, or run when the staff home page on which it sits is opened.

See Also:

[View Activity Logs](#)





VIEW ACTIVITY LOGS













To view your logs go to main screen/Activity Log tab.

 Activity Log


Status: More than 100 record(s) found. Click [here](#) to count records..

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

Delete  Actions Views  Search  

<input type="checkbox"/>	View	Id ↓	Date	User Name	Action	Description
<input type="checkbox"/>		5948	Apr 25 2013 07:34	EWSsystem	Login	ewsystem
<input type="checkbox"/>		5947	Apr 25 2013 07:34	EWSsystem	Login	ewsystem
<input type="checkbox"/>		5946	Apr 25 2013 07:34	EWSsystem	Logout by Admin	ewsystem
<input type="checkbox"/>		5945	Apr 24 2013 10:35	EWSsystem	Logout by Timeout	ewsystem
<input type="checkbox"/>		5944	Apr 24 2013 08:31	EWSsystem	Logout by Admin	ewsystem
<input type="checkbox"/>		5943	Apr 24 2013 08:31	EWSsystem	Login	ewsystem
<input type="checkbox"/>		5942	Apr 24 2013 08:31	EWSsystem	Login	ewsystem
<input type="checkbox"/>		5941	Apr 22 2013 16:07	EWSsystem	Logout by Timeout	ewsystem
<input type="checkbox"/>		5940	Apr 22 2013 11:06	EWSsystem	Login	ewsystem
<input type="checkbox"/>		5939	Apr 10 2013 18:17	Christopher Caldwell	Logout by Timeout	support
<input type="checkbox"/>		5938	Apr 10 2013 18:17	EWSsystem	Logout by Timeout	ewsystem
<input type="checkbox"/>		5937	Apr 10 2013 14:26	Christopher Caldwell	Login	support

You can view and edit each of them by pressing the View icon.

 Activity Log

[Edit](#) [Close](#) [General](#) << >>

Id	5946	Rule Id	279
User Information			
User Login	ewsystem	User Name	EWSsystem
Company	Agiloft	User Primary Team	Admin Team
Groups	admin, adminimport	IP	75.37.250.193
Action Information			
Date	Apr 25 2013 07:34	Action	Logout by Admin
Description	ewsystem	Interface	Staff
Table Label		Record Id	
Session Id	1463452	Session duration	0 days 0 hours 0 minutes 0 seconds
Action Duration		Cookie	1221799382

[Edit](#) [Close](#) << >>

Rules based on Activity Logs table

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

See Also:

[Activity Logs Setup](#)

CUSTOM SCRIPTS

EXECUTION

Scripts are automatically executed by the correct interpreter based on their file extension.

Files with an extension of .pl are executed automatically by the Perl interpreter.

Files with an extension of .class are automatically executed by the Java interpreter.

Files with an extension of .jar are automatically executed by the Java interpreter. Main class is taken from the Script-Class Manifest entry.

Files with an extension of .exe are automatically executed as stand-alone programs.

See Also:

[Perl API](#)

[EWget Module](#)

[EWset Module](#)

[Operations Flow](#)

[Sample Scripts](#)

OPERATIONS FLOW

Agiloft runs scripts in response to some events as defined by Rules specification. Events are always associated with a record from a table and a user action that actually triggered this event. Here is a list of possible events and the order of the resulting actions:

Record created/removed/modified through the GUI

user record: a record that has been created/removed/modified and matched our filter

user action: user record creation/removal/modification

The order of operations is:

- 1) Validation is performed on the record at the browser level, for example to confirm that the user has provided value for all required fields.
- 2) The validity of changes made by the user is checked based on group permissions.

This is an important security feature. Although Agiloft only sends to the browser the fields that the user is permitted to see and only displays the fields for which the user has edit permissions, a hacker might construct their own http stream by hand in an attempt to circumvent this protection. If any of the changes are invalid, the record update is cancelled and an error message is displayed.

- 3) If the user has changed the workflow state, any associated scripts are called. These scripts may modify fields in the record and act with absolute authority. In other words, they may set field values that the user would not be allowed to set based on his group permissions.

- 4) The record may trigger business rules that execute automatic actions and any associated scripts. The order in which the business rules are checked is based upon the Priority field in each rule, so if a business rule with high Priority modifies the record, a rule with lower priority will look at the modified record before deciding whether or not it should execute

Summary based: number of user records matching given filter [operator] [threshold], where operator is one of <, >, <=, >=, = and threshold is a number.

user record: all records meeting the filter. User can tell Agiloft to run actions either for all matching records, or only for one of them (see rule settings).

user action: user record creation/removal/modification after which number of matching records exceeded the threshold.

Timer based: periodically executed summary, number of records matching the filter>0.

user record: the same as in summary based
user action: undefined

See Also:

[Perl API](#)

[EWget Module](#)

[EWset Module](#)

[Execution](#)

[Sample Scripts](#)

LINKED FIELDS IN SCRIPTS

Since Java scripts have full access to the whole Agiloft functionality, the script author may retrieve whatever linked records he needs. This gives the script developer more flexibility (no constraints on multiple linked fields or depth of followed links) and causes minimal performance impact as long as only the necessary data is retrieved.

However, it is not necessary to fill the linkholders - they will be reconstructed automatically from the values supplied to one or more of the Linked Field data columns. It is important here NOT TO COPY a set of values from the source record and change just some of them, as it may lead to links not being restored correctly.

In perl/exe scripts Linked Fields are exported like any other fields together with the linked record from the donor table. Please note, that the linked record is exported only for a single Linked Field for performance reasons.

To create a link it is enough to set a value to one of the Linked Field data columns, which is done in the same way as for any other field. For example, if you want to change the Assigned To for a record and the Linked Field data column is named 118_f_login (you can find this column name in Table Wizard on Fields tab in column "Field name") then put the following to the output file:

```
<field name= "_118_f_user_login"/>  
  <value>Umbra</value>  
</field>
```

or if you are using Agiloft Perl Script API call

```
EWset::setRecordField('_118_f_user_login','Umbra');
```

The script running engine will try to find the best matching record in the donor table and set the Linked Field link to that record. It works the same way for "multiple fields from other table" LF, but you are free to fill only a subset of LF columns. For example, if there are fields User Login and User Phone exported to your table, you may provide a value just for the User Login if you are sure it will result in an exact match.

This will always work for user login, since they are unique across each KB, but other fields could contain repeating values. In this case one has to fill additional fields to achieve a unique match, otherwise the first suitable match will be used.

If no match at all is found, the link will be left empty.

See Also:

[Script Input/Output](#)

[Java-based Scripts](#)

[Perl Scripts and External Programs](#)

[Stringifiers Description](#)

Input File Example

The following information is provided as a guide to the internal implementation of scripts, for use mainly in adding support for additional languages.

Scripts receive several input parameters:

- a set of global variables including current user's record from Contacts, same as used in saved searches and formulas.

- script event source that could be on of the following:

WEB_WORKFLOW - script is triggered by workflow while record was edited in GUI

WEB_BUSINESS_RULE - script is triggered by rule while record was edited in GUI

EMAIL - script is triggered when record was created from inbound email

API - script is triggered when record was created or modified with Web Services

TIMER_BUSINESS_RULE - when script is triggered by rule based on schedule

- **current record values.** Permissions won't be applied either on reading or on saving. Scripts can also block user action or force user log-out. If a user record contains linked fields, then corresponding records from the linked tables are also provided. For example, if we have a record from Cases table, that contains assigned_to field= "John" which references Contacts table, then John's record from Contacts table will be also included. Only single linked records are exported, i.e. communications tables and any other multiple value LF records will not be exported. This constraint is applied for scripts on perl, .exe scripts and all other languages that interact with Agiloft via .xml files.

- **[optional] old record values,** available when script run in response to "record modified" event. This information allows a script to understand what fields were actually changed by the user (or by preceding rules-triggered actions).

Script should output:

- exit code - either in XML or as standard exit code. All other parts of output are optional. See DTD for output XML file.

- [optional] new values for record. Please note, that it is not necessary and even not recommended for some cases (like Linked Fields with multiple columns) to "copy" input records data to output. It is enough to output only the fields your script changes.

- [optional] debug message (will go to Agiloft log)

- [optional] user message - will be shown to the user in the case of non-redirecting exit code - as error message if the script blocks the change or as status message in table view if change accepted by script.
- [optional] redirect-URL to redirect user browser in the case of redirecting exit code
- [optional] report-message to store in record history to describe script actions upon the record

More detailed description of input and output formats is provided in the next chapters.

See Also:

[Java-based Scripts](#)

[Perl Scripts and External Programs](#)

[Linked Fields in Scripts](#)

[Stringifiers Description](#)

[Input File Example](#)

STRINGIFIERS DESCRIPTION

Date, Time and Datetime fields

Conversion to string:

Date field type is formatted using following pattern: dd/MM/yyyy

Datetime is formatted using the following pattern: dd/MM/yyyy HH:mm:ss

Time is formatted using the following pattern: HH:mm:ss

Conversion is done using current KB timezone and locale.

Parsing from string is as follows:

1. For each type (Date/Datetime/Time) there are list of patterns, which are used to parse.

- Date "dd/MM/yyyy", "MM/dd/yyyy", "MM/dd/yy", "dd/MM/yy", "MMM dd yyyy", "MMM dd yy", "dd MMM yyyy", "dd MMM yy"

- Datetime "dd/MM/yyyy HH:mm:ss", "MM/dd/yyyy HH:mm:ss", "MM/dd/yy HH:mm:ss", "dd/MM/yy HH:mm:ss", "MMM dd yyyy HH:mm:ss", "MMM dd yy HH:mm:ss", "dd MMM yyyy HH:mm:ss", "dd MMM yy HH:mm:ss"

- Time "HH:mm:ss"

2. If the string can't be parsed using any of these patterns then the converter tries to parse its value as a number, which represents time in milliseconds

3. If finally string can't be parsed - then exception is thrown

4. Parsed date is converted to KB time zone and current locale

Elapsed time fields

All values in DB are stored as time in milliseconds since January 1, 1970 00:00:00.000 GMT (Gregorian). Conversion is dependant on elapsed time column properties. They are:

showDays - use days in date format

showHours - use hours in date format

showMinutes - use minutes in date format

showSeconds - use seconds in date format

decimalHours - means show hours in decimal format, like 3.5

singleInputBox - means show single input box. No input boxes for days, hours, etc.

round - means round to a certain number of digits (for example, setting "round" to 2 would round 5.66666 to 5.67)

padding - means show a certain number of digits to the right of the decimal point (for instance, setting "padding" to 3 would display 5.7 as 5.700)

All the above properties could be tuned from Setup - > Tables - > Edit - > Fields screen.

Conversion to string is quite simple, the algorithm is as follows:

1. Convert from milliseconds to seconds
2. if (showDays) - calculate number of days (value/86400). if it's not single input box - append word "XXX days" to output, otherwise only XXX
3. if (showHours) -
 - a. - if singleinputbox and we already wrote something in output - then append delimiter ":"
 - b. - calculate number of hours (remain_value/3600)
 - c. - if decimal hours enabled - then append formatted value of hours using padding/round, then print out delimiter (if single box is enabled) or word "hours". Go to step 6.
 - d. - otherwise - go to step 4.
4. if show minutes is enabled - print out floor value of hours, otherwise append rounded value of hours
5. if singlebox is disabled - print out word "hours"
6. if showMinutes is enabled and decimal hours disabled
 - a. - if singlebox is enabled and we already printed out something - print out delimiter ":"
 - b. - calculate and print out minutes.
 - c. - if singlebox disabled - print out word "minutes"
7. if showseconds is enabled and decimal hours disabled

Agiloft Reference Manual

- a. - if singlebox is enabled and we already printed out something - print out delimiter ":"
- b. - calculate and print out seconds
- c. - if singlebox disabled - print out word "seconds"

So, possible values for 1 day 2 hours and 23 min 15 seconds could be:

1: 2:23:15 - when singlebox is enabled

1: 2 - when singlebox is enabled, decimal hours are enabled and show minutes with seconds is disabled.

1 day 2 hours 23 minutes 15 seconds - when singlebox is disabled and we are showing all time units.

1 day 2.0 hours when singlebox is disabled and decimal hours is enabled with round/padding equal to 0 etc. depending on current settings of the column.

Conversion from string to Agiloft internal representation is as follows:

1. if singlebox is enabled then use ":" as delimiter, otherwise space ""
2. Parse whole string and extract string values for each time unit.
3. Iterate through time units and convert/parse string values to numbers.
4. Calculate final value of elapsed time. Note, time must be in milliseconds passed from January 1, 1970 00:00:00.000 GMT (Gregorian)

Object Type fields

Converted to string as `${RootSubtypeLabel}. {...}{RecordSubtypeLabel}.please`, note that subtype labels are used here so you will get localized label value for locale of rule creator.

Record links fields

The external form is (example) "[ID1][ID2][ID3]" where ID1, ID2, ID3 are IDs of linked records.

You can't modify record links by passing new set of linked IDs on save. An empty string represents an empty link set.

Choice fields

Choices are shown as simple text strings (note that choice lines cannot contain commas inside their labels).

values "\$none", "\$choose_one" and "\$choice_item" are treated as "nothing selected".

Multichoice fields

Multichoices are shown as a text strings separated by commas (hence the choice line itself cannot contain comma inside its label).

Multiple values linked fields

Multiple values are exported in format:

```
<dao_line $recordId1> Some value</dao_line> <dao_line $recordId2> Another value</dao_line>
```

where \$recordId1 and \$recordId2 are Ids of the donor records these values are from.

Multiple Linked Field values cannot be modified by external scripts.

See Also:

[Script Input/Output](#)

[Java-based Scripts](#)

[Perl Scripts and External Programs](#)

[Linked Fields in Scripts](#)

Input File Example

PERL BASED SCRIPTS

Using Custom Scripts

Introduction

Agiloft provides a custom scripting capability that is a powerful tool for validating data, automating business processes and integrating with other back-end programs.

Scripts may be created in any standard programming language such as Perl, Java, C++, or Basic and run by Business Rules (select Setup/Rules) or Workflows (select Setup/Workflow). These wizards tell you in which directory on the server the script must be placed. For security reasons, it is not possible to upload the script to the server using the GUI; the script must be installed directly on the Unix or Windows file system.

Scripts can:

- Set field values in the record. For example, a script can apply sophisticated logic to automatically assign new Cases to the correct member of staff.
- Logout the user and redirect his or her browser to a specific URL.
- Verify the data being entered by comparing it against data in an external file or database and, if necessary, block the requested change to the record.
- Write to an external program or file or generate email.
- Return personalized messages based on field values, such as "Hi Bob, thank you for the input"
- Accept or reject changes to the record made by the user.

Scripts are provided with the following information:

- Field values of the record that is being modified. For example, Severity.
- Field values that record held prior to the current modification. For example, the old value of Severity.
- Field values (from the Contacts table) of the user who is modifying the record. For example, Full Name.
- Field values of any records that are linked to the record that is being modified. For example, the Full Name of the person that the record is currently Assigned To.
- Global variables, such as the name of the KnowledgeBase.
- Modification event source. The possible values are: WEB_WORKFLOW, WEB_BUSINESS_RULE, EMAIL, API, TIMER_BUSINESS_RULE.

See Also:

[Perl API](#)

[EWget Module](#)

[EWset Module](#)

[Execution](#)

[Operations Flow](#)

[Sample Scripts](#)

Perl API

The Perl API makes it easy to write custom scripts. It consists of two modules- EWGet.pm and EWSet.pm-for reading input and writing output files.

There are many ready-made functions available, but many scripts only use 6 functions: EWget: load, EWget::getValue, EWset::setRecordField, EWset::setMessage, EWSet::setExitAction and EWset::save, and two of these ('load' and 'save') are identical for all scripts. If you just master the functions EWget::getValue, EWset::setRecordField, EWset::setMessage and EWSet::setExitAction, you will be able to write very powerful scripts.

Many of the functions take a parameter \$recordName as an input variable. There is always a record named "current_state", which is the record that is currently being edited. In addition, the script may have available to it "old_state", which provides the old field values of the record and other \$recordNames, which provide the values of records that are linked to the input record via linked field relationships.

See Also:

[Introduction](#)

[EWget Module](#)

[EWset Module](#)

[Execution](#)

[Operations Flow](#)

[Sample Scripts](#)

EWget Module

EWget::load(\$fileName)

Loads the input file into internal data structures, you will be calling it like this:

Example

```
my ($input_fname, $output_fname) =@ARGV;
EWget::load($input_fname);
```

Comments

This is the first thing you must do before any further calls to EWget functions. This call is the same for all scripts.

EWget::getValue(\$recordName, \$fieldname)

Returns the value for the field of the record.

Example

To retrieve the current value of the 'status' field:

```
EWget::getValue('current_state', 'status')
EWget::getValue('old_state', 'status')
```

Retrieves the old value of the 'status' field, i.e its value prior to any changes made by the user:

Comments

This is the core method for accessing field values, but see also `EWget::record($path)`, it is a more elegant method for retrieving the fields of the 'current_state' record.

EWget::getGlobalVariable(\$variableName)

Returns the value of the specified variable. For example:

Example

The following function retrieves the value of the "my_first_name" field in the contact entry of the user who caused the script to run.

```
printf "my_name= '% s' \n", EWget::getGlobalVariable('my_first_name');
```

Comments

Global variables contain KB-wide settings along with user-specific settings, they are also used in saved searches and formulas.

EWget::getLinkedValue(\$recordName, ...)

Returns value of a field from a linked record.

Parameters:

- name of base record
- name of field from base record
- index of linked record (always=0 at present, because as noted in the getLinkedRecordsNamesList description, only "single" value linked fields are currently exported to scripts)
- name of field of linked record

Example

To find the value of the "email" field of the employee responsible for the given case:

```
my $responsible_person_email=  
EWget::getLinkedValue("current_state", "assigned_to", 0,  
"email");
```

Comments

This is the core method for accessing the data in linked records.

EWget::record(\$path)

Returns the value of a field in a record.

This function provides a "compact" navigation through the record data for the "current_state" record.

Path format: fieldname [.reffieldname[#index]]

Example

"email" - returns the value of current_state.email

"email.username" - returns the value of the "username" field for the first referenced record from current_state.email

So:

```
EWget::record(email.username)
```

is equal to

```
EWget::getLinkedValue("current_state",
"email", 0, "username");
```

Comments

As the above example shows, EWget::record() is not really necessary, but it is certainly convenient.

EWget::getCallInitiator()

Returns a string that indicates the action that caused the script to be called.

The value returned is one of:

WEB_WORKFLOW-A workflow action, typically initiated by a user changing the Workflow State in the GUI.

WEB_BUSINESS_RULE-A business rule that is triggered by a change to a record

TIMER_BUSINESS_RULE-A business rule that runs automatically at specified intervals

EMAIL-Inbound email

API-An action triggered by the Web Services API

Comments

This function is useful when you have a single script that responds differently, depending upon the cause of the event. For example, if a ticket is updated via email and meets some special criteria, you might send a confirmation email if the update was performed through email, and provide a message in the GUI if the update was performed using the web interface.

EWget debug functions:

The following functions are mostly used for debugging scripts.

EWget::getGlobalVariablesNamesList()

Returns a sorted array of global variables names.

Example (names of all available variables)

```
printf "all vars: %s\n", join(',', EWget::getGlobalVariablesNamesList());
```

Comments

You will usually know what global variables you want to use in your scripts, so this function is mostly for debugging purposes and to find out what variables are available.

EWget::getFieldsNamesList(\$recordName)

Returns list of field names for record by record's name.

Example

Obtain dump of all the fields' names for "current_state" record

```
printf "fields names: '%s' \n",  
join ('', EWget::getFieldsNamesList("current_state"));
```

Comments

This function is mostly used for debugging since you will usually know what field names you are interested in before writing the script.

EWget::getRecordsNamesList()

Returns an array with the names of all the records provided to the script.

Example

The following code dumps all the records names from the input file passed to the script.

```
printf "record names: '%s' \n",  
join(',', EWget::getRecordsNamesList());
```

Comments

This function is mostly used for debugging purposes. There always will be a record named "current_state" and possibly one named "old_state" (if the current record is

being modified, rather than created). There may also be records that were linked to the current record. These are named "recXXX", where XXX is an incremental number starting from 1.

EWget::getLinkedRecordsNamesList(\$recordName, \$fieldName)

Returns sorted list of names of records linked via a "linked" field.

Comments

This function is mostly used for debugging because you will probably know the names of the records that you want to access before writing the script.

Only the linked records where the link does not allow multiple values are passed to scripts, so this function will return an array of 0 or 1 entry in a form of "recXXX" string.

EWget::getTable_name_for_record(\$recordName)

Returns the name of the table that a record belongs to. If \$recordName is omitted, it returns the value of the "current_record". For example:

```
printf "table for current record: '%s' \n", EWget::getTable_name_for_record();

printf "table for rec3: '%s' \n", EWget::getTable_name_for_record('rec3');
```

In most cases you will already know what tables your records are from, but in case of the linked field type "single field from multiple tables" this function will allow you to determine what table the associated record is from and handle it appropriately.

See Also:

[Perl API](#)

[EWset Module](#)

[Execution](#)

[Operations Flow](#)

[Sample Scripts](#)

EWset Module

EWset::save(\$outfileName)

Processes changes requested by the user and stores them in an xml file.

You should call this method at the end of your script to make your changes available to Agiloft. For example, the script may have changed the value of a field, this change will be ignored, unless it calls "save" prior to exiting.

Example

If you used

```
my ($input_fname, $output_fname) =@ARGV;
```

at the start of your script to extract commandline params, then use

```
EWset::save($output_fname);
```

to save your changes.

EWset::setRecordField(\$fieldName, \$fieldValue)

Assigns a value to the specified field in the record or sets the field to NULL if the value is undefined.

Example

To set the value of the "email" field in the current record

```
EWset::setRecordField("email",  
    "somemail@somesite.com");
```

Comment

This is the core function for change data in a your script.

This function manipulates both common fields and linked fields.

See [Stringifiers Description](#) for full description of acceptable formats for each type of data.

See also [Linked Fields in Scripts](#) chapter for more information.

EWSet::setExitAction(\$exitAction)

Determines what happens when the script exits.

Allowed options are (please note double quotation marks):

- **"AcceptChanges"** - the record is updated in the normal fashion. You can use `setMessage` to inform user about the details of update.
- **"RejectChanges"** - the modification is rejected. User will be shown the record editing window (if action was initiated via web interface).
- **"RejectChangesAndExit"** - the modification is rejected and the user is automatically logged out. After logout, user will be redirected to URL specified by `setRedirect` function call.
- **"AcceptChangesAndExit"** - the record is updated in the normal fashion and the user is automatically logged out. After logout, user will be redirected to URL specified by `setRedirect` function call.

Alternative to setExitAction

As an alternative to `setExitAction`, provided for compatibility with languages that do not allow strings to be returned, the script can provide an exit (return) code.

- If the exit code is 0, the record is updated in the normal fashion. You can use `setMessage` to inform user about the details of update.
- If the exit code is 1, the modification is rejected. User will be shown the record editing window (if action was initiated via web interface).
- If the exit code is 2, the modification is rejected and the user is automatically logged out. After logout, user will be redirected to URL specified by `setRedirect` function call.
- If the exit code is 3, the record is updated in the normal fashion and the user is automatically logged out. After logout, user will be redirected to URL specified by `setRedirect` function call.

Formally speaking this function sets the value for the tag 'exitAction'.

EWset::setMessage(\$userMessage)

Send a message to the user.

This message will be shown to the user if the script is executed because the user has changed a record in GUI.

Example

```
EWset::setMessage("Please, select yes to accept
end-user agreement and create new account")
```

When a script blocks a record edit, it is a good practice to use `setMessage` to explain why the change was blocked and what the user should do. Of course, it can also be used to confirm success or provide some other message.

Formally speaking, this method sets the value for the tag 'message'.

EWset::setRedirect(\$redirectUrl)

Sets the value of the URL that the user should be directed to if the script logs them out of Agiloft

The redirect URL should be absolute and include "http". It will be opened in the same browser window.

Example

```
EWset::setRedirect("http://www.enterprizewizard.com")
```

It is useless to call `setMessage()` with `setRedirect` since there will be no Agiloft page shown to display the message.

Formally speaking, this method sets the value of the tag 'redirect'.

EWset Debug Functions

The following functions are mostly used for debugging scripts.

```
EWset::setDebug($debugMessage)
```

Write to the log file (formally speaking, sets value for tag 'debug').

This message is saved to the Agiloft log. It can be useful for logging the result of a script or to debug the script.

EWset::setReport(\$reportMessage)

Sets a value in the record history

This message is stored in the record history.

Comment

It is sometimes helpful to write a description of what the script did to the data and why.

Formally speaking, this method sets the value for the tag 'report'.

EWSet::exit_ew (\$exitAction, \$outfileName)

Combination of functions EWSet::setExitAction(\$exitAction) and EWset::save(\$outfileName) with the same arguments and meanings.

Comment

The function is able to get an exitCode directly, as \$exitAction argument, thus the following pairs of values are equivalent:

```
"AcceptChanges" or 0 or "0"  
"RejectChanges" or 1 or "1"  
"RejectChangesAndExit" or 2 or "2"  
"AcceptChangesAndExit" or 3 or "3"
```

See Also:

[Perl API](#)

[EWget Module](#)

[Execution](#)

[Operations Flow](#)

[Sample Scripts](#)

Perl Scripts and External Programs

Perl scripts and external programs have much in common. External programs (.exe on Windows, runnable [+x] on Unix or Linux) will be executed by Agiloft directly, using `java.lang.Runtime.exec()`. Perl scripts are distinguished by ".pl" extension and will be run using "perl scriptname.pl" command. Agiloft includes a Perl interpreter with the standard installation.

Scripts take 2 command line arguments: names of the input and output files. Input and output files are in XML format. Note that the output file is precreated by Agiloft but is empty.

Input file conforms to the following DTD:

```
<!DOCTYPE input[
  <!ELEMENT input project_name, old_state?,
  linked_records, global_vars)>
  <!ELEMENT project_name (#PCDATA) >
  <!ELEMENT old_state (record) >
  <!ELEMENT current_state (record) >
  <!ELEMENT linked_records (record) *>
  <!ELEMENT global_vars (variable) *>
  <!ELEMENT record (field) +>
  <!ELEMENT variable (value) >
  <!ATTLIST record
  table CDATA #REQUIRED
  xml:id ID #REQUIRED>
  <!ELEMENT field (value) >
  <!ATTLIST field
  name CDATA #REQUIRED
  linked_refid IDREFS #IMPLIED>
  <!ELEMENT value (#PCDATA) >
  <!ATTLIST variable
  name CDATA #REQUIRED>
]>
```

See also full-sized input file example taken from a real KnowledgeBase.

For Perl scripts there are interface modules for interaction with Agiloft provided; see full description below.

Output file example

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE output[
  <!ELEMENT output (exitAction?, debug?, message?, redirect?, report?,
  record?) >
  <!ELEMENT exitAction (#PCDATA) >
```

```

<!ELEMENT debug (#PCDATA) >
<!ELEMENT message (#PCDATA) >
<!ELEMENT redirect (#PCDATA) >
<!ELEMENT report (#PCDATA) >
<!ELEMENT record (field) +>
<!ELEMENT field (value) >
<!ATTLIST field
name CDATA #REQUIRED>
<!ELEMENT value (#PCDATA) >
]>
<output>
<exitAction>AcceptChanges</exitAction>
<debug> Something to be sent to the server logs</debug>
<message> Script accepts your data</message>
<report> Some report for the history fuctionality</report>
<record>
<field name= "assigned_to" >
<value> Umbra</value>
</field>
<field name= "summary" >
<value> some summary here</value>
</field>
</record>
</output>

```

This file describes the script result, with `assigned_to` field value set to "Umbra", summary set to "some summary here", changes accepted to be put to database, and "Script accepts your data" message is shown to user.

Both files always use UTF-8 encoding. Input file uses xml: id <http://www.w3.org/TR/2004/WD-xml-id-20041109/> for referring to the linked records. Things like project and table names and values from user record are not translated into the user's language. When converting user record into XML format Agiloft uses so-called Stringifiers from `com.supportwizard.util.stringify` package.

In short, for every column type there exists a stringifier that can translate the internal data representation into String and vice-versa. For instance, if there is a Choice 'priority' column that contains 'Medium' value, it's internal representation is Medium's ID and choices stringifier will convert this ID into 'Medium' string for scripts, - then some script can change 'Medium' into 'High' and stringifier will convert it back into 'High's ID. This implies that values produced by scripts must be understandable by stringifiers. See the appendix for full stringifiers info.

All of the output file fields are optional, except for `<redirect>` that must be present if exit code is 2 or 3 (see below). If there 's no `<record>`, then it is considered to be unmodified. `<debug>` only sends it's content to system log. `<report>` provides data for a report that will appear in the history tab. `<message>` and `<redirect>` are covered below.

The main thing that Agiloft expects from a script is its return (exit) code. There are 4 possible values:

- **0 means success.** If <message> is presented, it is shown to the user. <redirect> is ignored, all modifications from <record> take their place.
- **1 leads to user action block.** If <message> is presented, then it will be shown to the user. <record> modifications will be ignored.
- **2 will cause action block, user log-off and redirection.** <message> will be logged, and user will be redirected to <redirect>. <record> modifications will be ignored.
- **3 leads to user log off and redirection, but keeps changes.** This is similar to 2, but action won't be blocked and changes made by script will take place.

It is also possible to use <exitAction> to override the exit codes. Recognized values are:

- AcceptChanges equivalent to exit code of 0
- RejectChanges equivalent to exit code of 1
- RejectChangesAndExit equivalent to exit code of 2
- AcceptChangesAndExit equivalent to exit code of 3

If exitAction is present, it overrides the "Raw" exit code. For instance if <exitAction> is set to AcceptChanges and exit code is 1, changes will be accepted.

Scripts must reside in the scripts directory. By default it is Agiloft.installation.dir}/data/{kb.name}/scripts. When creating script action, user should specify the script filename as action name, like "myScript.pl", for example.

See Also:

[Script Input/Output](#)

[Java-based Scripts](#)

[Linked Fields in Scripts](#)

[Stringifiers Description](#)

Input File Example

Sample Scripts

The following scripts provide examples of using custom scripting capabilities in a typical Agiloft installation (See Appendix B).

The Agiloft distribution includes Perl and will automatically use it to execute scripts with a .pl subscript. If you want to use your own version of perl, you may do so by specifying it on the first line of the script, eg #!/usr/local/perl.

Sample Script 1 redirects the user after he/she submits a record.

Sample Script 2 sends new users a confirmation email with their login and password.

Sample Script 3 checks if the user belongs to QA group and blocks closure of the ticket if not, and sets the status field to its old value.

Sample Script 4 illustrates some common tasks you would perform in a typical script.

JAVA BASED SCRIPTS

Scripts can also be written in Java, this allows very tight integration with Agiloft and direct access to all user data.

Java scripts are the most flexible and performance-effective way to implement your custom logic. You can retrieve only the necessary data and also skip export to xml, running external program and the import results steps. The result is a significant performance boost in the case of complex tables structure with a lot of linked fields.

The script class must implement

`com.supportwizard.actions2.interfaces.ExternalScript`

This interface has only one method:

`ScriptOutput runScript (final ScriptInput input) throws ActionException;`

Please consult Javadoc for more details.

ftp://ftp.agiloft.com/pub/Java_Based_Scripts_Javadoc.zip

Input Data

A `ScriptInput` instance provides input data: project and user table ID 's, user's Seance and the record data.

The record data is passed in a map form - an instance of `com.supportwizard.dml.SWDataMap` class. Specific field values can be accessed by name (as visible in GUI in `Setup/Table/<yourtable>/Edit/Fields`). The script gets one or two instances of the data map - the "old" one and the "new" one. At least one will be present, depending on when the script is invoked (create - only "new", delete - only "old", modify - both).

This is the mapping of specific data types returned for different field types:

- Auto-Increment => class `java.lang.Long`
- Elapsed Time => class `java.lang.Long`
- DAO3 link field => class `com.supportwizard.functionalities.dao3.util.SWDao3LinkHolder`
- Long integer field => class `java.lang.Long`
- Billing field => class `com.supportwizard.functionalities.dao3.util.SWDao3LinkHolder`
- Integer => class `java.lang.Integer`

- Choice => class com.supportwizard.dictionary.SWChoiceLine
- EMail => class java.lang.String
- EMail Pager => class java.lang.String
- Telephone/Fax => class java.lang.String
- Multi-Choice => class com.supportwizard.dictionary.MultichoiceLines
- Short Text => class java.lang.String
- Password => class java.lang.String (however it comes in ***** form for security reasons and is generally useless to a script writer)
- Text => class java.lang.String
- URL => class java.lang.String
- File => class com.supportwizard.functionalities.blob.SWBlobRefHolder
- Image => class com.supportwizard.functionalities.blob.SWBlobRefHolder
- History => class java.lang.String
- DAO3 multiple field => class com.supportwizard.functionalities.dao3.util.SWDao3MultiValue
- WMI Field => class java.lang.String
- Append Only Text => class com.supportwizard.dictionary.appendtext.AppendOnlyTextContainer
- Floating Point => class java.lang.Double
- Percentage => class java.lang.Double
- Currency => class java.lang.Double
- Date/Time => class java.sql.Timestamp
- Date => class java.sql.Timestamp
- Time => class java.sql.Time
- Compound => class java.lang.String
- Calculation on Multiple Linked Records => class java.lang.Double

A linked field that imports N field from the source table into the target one will be represented by N+1 entries in the "input" Map on the data level.

If a single record is imported each of the N entries corresponding to N fields will have a single value of corresponding type. If a multiple records are imported, each of the N entries corresponding to the N fields will have an instance of SWDao3MultiValue, a list of String values, one for each imported record, with "__{NULL}__" value marking the null one. The sequence of values in different multi-value fields is the same for the same set of imported records.

The extra entry mentioned above will be under the name that can be seen in the GUI in Setup/Table/<yourtable>/Edit/Fields/<yourLF>/Edit/Fields at the bottom "Linked Column Name". It will contain an instance of the SWDao3LinkHolder - essentially an array of identifiers for the records in the source table(s).

To access the n-th identifier:

```
List<SWDao3LinkHolder.Link> links = linkHolder.getLinks();
SWDao2LinkHolder.Link link = links.get(n);
Long pk = link.getLinkPK();
```

A script call always starts with the input object and is being called for a single record only. A script however can use the CRUD+S (select) API underlying WebServices and REST access - SimpleAPI.

```
Context jndiContext = new InitialContext();
EWSimpleAPILocalHome localHome = (EWSimpleAPILocalHome)
    jndiContext.lookup("ew/EWSimpleAPI");
EWSimpleAPI EWSimpleAPI = localHome.create();
long[] ids = EWSimpleAPI.EWSelectFromTable("allocation",
    "general_issue=" + general_issue +
    " and specific_issue=" + specific_issue +
    " and default_team = " + line.getId(), seance);
if (ids.length == 0) {
    return blockedScriptOutput(output,
        "There is no default team defined for this combination of Issue
Types.");
}
```

For more details on SimpleAPI please consult the following Javadoc:

<http://support.agiloft.com/ewws/com/enterprisewizard/ws/api/ejb/EWSimpleAPI.html>

Output data

A ScriptOutput instance describes script output, an instance of this class is intended to be created with

public ScriptOutput (final ScriptInput input)

constructor. ScriptOutput can optionally contain modified user record data (don't change ScriptInput.getRecord() instance).

DO NOT copy any values your script leaves unchanged from the input record to output. Doing so is unnecessary and could have unexpected results if you are filling linked fields with multiple columns.

Modifications will take place if ScriptOutput.getExitCode() is ExternalScript.SUCCESS_CODE. If exit code is equal to BLOCKED_CODE, then the user action will be blocked and the user will get ScriptOutput.getMessage() as an error message (this message will also be shown in case of ACCEPT_CODE). In cases where the user action is undefined (see timer based rules above) nothing will be blocked and error message will only be logged.

BLOCK_REDIRECT_CODE will also block user action and will force the current user to log-off. The user will be redirected to ScriptOutput.getRedirect(). ACCEPT_REDIRECT_CODE will logoff the current user, but won't block an action and script changes will take place.

Unhandled exceptions propagate to Agiloft where they are caught, logged and displayed to the user depending on the way the script was run. This display is often too verbose and is not really suited for ordinary users.

Thus the recommended way of handling the exceptions is either suppress them in the script itself and instead return a ScriptOutput instance with a blocking exit code and an optional message or wrap in action-related exceptions.

```
private ScriptOutput blockedScriptOutput(ScriptOutput output, String s,
SWDataMap newRecord, Seance seance) {
    output.setExitCode(ExternalScript.BLOCK);
    output.setMessage(s + " ID:" +
newRecord.getSWRecordPK(seance).getID());
    return output;
}
```

When wrapped exceptions can be used to change control flow. A plain ActionException means some system-level error and leads to transaction rollback. Throwing ActionBlockedException (String message) amounts to returning BLOCKED_CODE with ScriptOutput.getMessage() being 'message'.

Throwing ActionBlockedException (String message, String redirectURL) will do the same thing as returning BLOCK_REDIRECT_CODE.

General considerations

The call to the script is done in a synchronous manner. The calling code (Agiloft) waits. There is a potential to the encompassing business transaction to time out and be rolled back by Agiloft's application server if it takes too long. If this is a frequent case, consider implementing asynchronous mode of operation on the caller side.

A new classloader is created on each script run as well as the script class instance. It is not isolated as far as it delegates to parent classloader in java 2 compliant manner. The script doesn't have to be thread safe since each call is being served by a different instance.

At this moment if a script relies on some 3rd party libraries not available in the Agiloft environment, these have to be repackaged into the script archive.

The Agiloft jar files you may need while developing are in
{Agiloft.installation.dir}/jboss/server/sw/lib/sw and
{Agiloft.installation.dir}/jboss/server/sw/deploy

Normally, the following classes are enough: SWFunctionalities.jar, SWSeance-ejb.jar

The jars that make the JBoss environment would be under
{Agiloft.installation.dir}/jboss/lib, {Agiloft.installation.dir}/jboss/server/lib

Deployment considerations

Scripts must reside in the scripts directory. By default it is

{Agiloft.installation.dir}/data/{kb.name}/scripts

If your script consists of a single class, then you can simply drop the .class file in the scripts directory. This .class must not belong to any package. If you have several classes, then you must put them in a jar file with a special key in it's Manifest file:

Script-Class: name of the class implementing ExternalScript

For example: com.mycompany.test.TestExternalScript

Now your script name will be the name of the script (.jar or .class) file.

A Java custom script is run by Agiloft within the same JVM. All libraries present in Agiloft application server instance classpath
({Agiloft.installation.dir}/jboss/server/sw/lib/,
{Agiloft.installation.dir}/jboss/server/sw/lib/sw/, {Agiloft.installation.dir}/jboss/lib)
are available for the script to use.

Agiloft runs on Java 6. As for JAXWS - the libraries available to the script will be those distributed with the application server rather than those available from the runtime environment.

See Also:

[Java Sample Scripts](#)

Sample Java Scripts

The following scripts provide examples of using custom Java scripting capabilities in a typical Agiloft installation (See Appendix B).

ReplicateTicket Sample Script

The purpose of the "ReplicateTicket" is to automatically replicate Tickets from one server to another via REST call

CreateInEWSupportScript Sample Script

The purpose of the "CreateInEWSupportScript" is to automatically replicate Tickets from one server to another.

QueryAsset Sample Script

The purpose of the "QueryAsset" is to demonstrate the current way of working with Agiloft SOAP Web Services. In particular the interface is generic - exposing and accepting parent object to allow one to use the same web service with different tables. Methods are low level - one has to combine several calls to achieve required. This is also a stateful version of the interface that relies on the underlying protocol support for stateful communication via cookies. This script is run from the commandline, accepts host, port, kbName, user, password and searchValue parameters. The script performs search by SQL on Asset_Table object of the knowledgebase, parses the returned object tree and prints results to the standard output.

QueryAsset New Sample Script

The purpose of the "QueryAssetNew" is to demonstrate the upcoming changes in EnterpriseWizard Web Services interfaces, in particular "Single Method Operations", "Immediate Read", "Web Service Per Table" concepts.

CreateCaseCurrent Sample Script

The purpose of the "CreateCaseCurrent" is to demonstrate the current way of working with Agiloft SOAP Web Services. In particular the interface is generic - exposing and accepting parent object to allow one to use the same web service with different tables. This is also a stateful version of the interface that relies on the underlying protocol support for stateful communication via cookies. This script is run from the commandline, accepts host, port, kbName, user, password and several specific field parameters. The script invokes EWCreate from SOAP API to create a Case in Agiloft.

CloneTicketNew Sample Script

The purpose of the "CloneTicketNew" is to demonstrate the way to pass results of one WS method to another and challenges associated with this. The script uses "Single Method Operations", "Immediate Read" calls that will be available in upcoming release of Agiloft Web Services. This script is run from the commandline, accepts host, port, kbName, user, password and searchValue parameters. The script performs search by SQL on Case object of the knowledgebase and submits the returned object right back in to create a clone.

POSTPush Sample Script

The purpose of the "POSTPush" is to submit the result of modification of a record in Agiloft to an external host via HTTP POST. This script is run in a script action from the rule invoked on update. The script re-uses parts of WS/REST internal code to convert internal datatypes into POST parameters.

NetcoolTicketUpdateScript Sample Script

The purpose of the "NetcoolTicketUpdateScript" is to invoke a policy in IBM Tivoli Netcool when a ticket is closed in EnterpriseWizard. A policy can be invoked by calling a "Listener" Web Service provided with each instance of Netcool. This script is run in a script action from the rule invoked on update and also periodically every 5 minutes. The purpose of the periodic run is to pick up those tickets for which for some reason the information was not transmitted successfully i.e. retry if some failure has occurred. The script uses an integer field "submitted_to_netcool" to indicate that the invocation was successful. To avoid cluttering history by recording the success on the very first run the suggested pattern is to have the field set to 1 (success) by default and set value to 0 only when it fails. The script will only update the field if a change is required. To attempt a resend such records can be isolated later by a rule run periodically with an IF/THEN condition based on Advanced Filter - the field changed value from 1 to 0 in the last X minutes.

SYSTEM SETUP

SETUP HOME PAGE

The Setup Home Page presents administrators with the setup options in the main body area with icons and links to main setup options and brief descriptions. This eliminates the need for a submenu for Setup and provides further details and one-click access. The Setup menu is only visible to users with the appropriate permissions.

Setup Home Page |



Tables

Create, edit and delete:
Tables and relationships, fields
and action buttons, custom forms
and layouts.
Generate sample WS code.



Rules

Rules automate your business:
Send automatic emails, update
fields, run scripts, auto-create new
records, schedule data imports,
validate user entries and set up
escalation rules.



Workflow

Map your process flow and
automate work assignments,
notifications.
Ensure full auditability and control.



Access

Manage permission groups.
Manage working teams.
Configure LDAP/MS AD.
Setup single sign-on.
Set Hotlink security/encryption.



Email

Configure the email server.
Setup inbound email accounts.
Setup outbound email behavior.
Create/edit email templates.



System

Set the KnowledgeBase Time.
Configure activity log for auditing.
Manage global variables for
timeouts, locking, and login page
URL.
Edit HTML header text.



Look&Feel

Manage look and feel schemes to
control fonts, colors and graphics.
Set the order of tabs.
Define required field appearance.



End-User Interface

Setup FAQs for end users and the
display of My Items records.
To create a fully custom GUI, see
online help



Chat

Configure the appearance and
behavior of chat sessions between
staff and end-users.



License

Request or install a license.
View current license usage and
assign or terminate licenses.



Import

Import data from ASCII, Excel,
XML, or Agiloft formats.



Export

Export data to ASCII, Excel, XML
or Agiloft format.



Localization

Translate choice lists and all
program text into other languages.



Sync

Sync with Exchange, Quickbooks,
file systems and 3rd party
software.



Twitter

Create, edit and delete settings for
integration with Twitter



Skype

Enable, disable and set integration
with Skype



LogMeIn Rescue

Create, edit and delete settings for
integration with LogMeIn Rescue

See Also:

[Flash Tutorials](#)



INSTALLATION

The Agiloft installation package includes everything necessary for installation on Windows, Linux, Solaris, AIX or Mac OS X, including a web-server and a database if you do not have one installed already.

Installation, upgrades and de-installation are all managed through the installer wizard.

Installer manual is available here: [Agiloft Installation Documentation](#)

The Agiloft (XML based) export format is compatible across all operating systems and databases so you can easily move the entire installation from one hardware platform/OS/Database to another as follows:

Install the software on the new machine, export the KnowledgeBase by selecting Setup/Export/Destination=Local hard drive, Data Format=Agiloft format, Options=All, - and save the file on your local hard drive.

On the new server, login to the admin interface and select Setup/Import/Local hard drive and enter the file name to upload.

To save time for large export files export to the TMP directory and copy the KB across directly.

The installation bundle includes a "servlet container". This is a specialized web server for providing the dynamic web pages that make up the Agiloft interface. This servlet container is usually installed on port 8080 and will not interfere with the regular web server. The installer can automatically configure web servers such as Apache to redirect Agiloft traffic to another port on the container. This allows Agiloft to be accessed by users on port 80, just like the rest of the web site.

It is possible to install Agiloft under Windows with IIS7 (for example on Windows 2008 Server). The installer will find the working IIS7 and automatically integrate with it.

See Also:

[KnowledgeBase](#)

[Administrator Console](#)

[Licensing](#)

[Flash Tutorials \(Upgrading on Windows and Unix\)](#)



SECURING AGILOFT

To ensure the security of Agiloft and the server it is run on, you should take the following steps:

Use complex passwords.

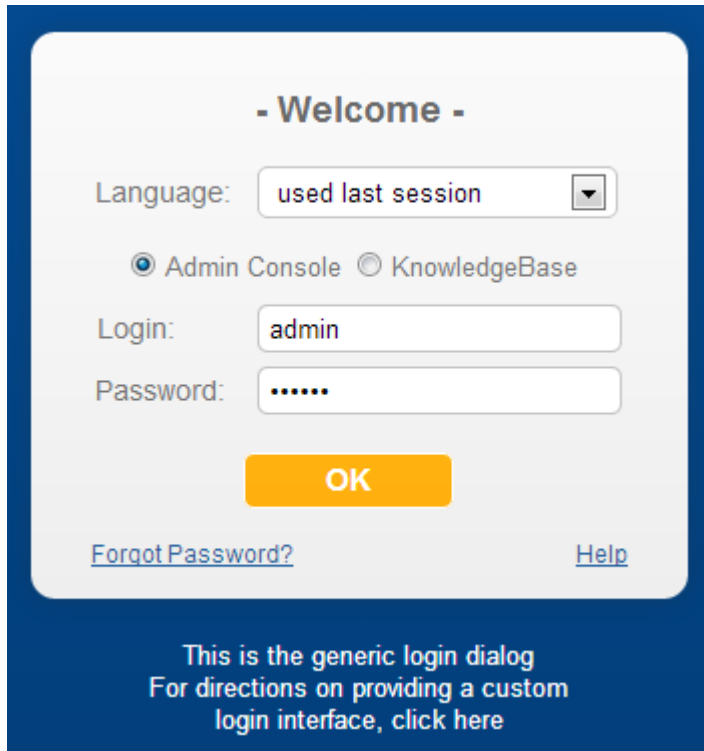
Passwords that are resistant to attack should be at least 8 characters in length, contain a mixture of upper and lower-case characters, contain one or more numbers or other non-alphabetic characters, and not be derived in any obvious way from the username. All staff accounts should be secured with such passwords, especially those in the **Admin** group. If it is desired to give end-user accounts simple passwords for user convenience, then these accounts should be severely restricted in what they may do (for example only filling out a single form). End-user accounts with the ability to modify existing records or view sensitive data should also be given attack-resistant passwords.

Change the password

Change the password of the Admin Console and default KnowledgeBase users. These default passwords are well-known, and are an extremely easy method of attack.

To change the Admin Console password, do the following:

1. Log in to the Admin Console:



- Welcome -

Language: ▼

Admin Console KnowledgeBase

Login:

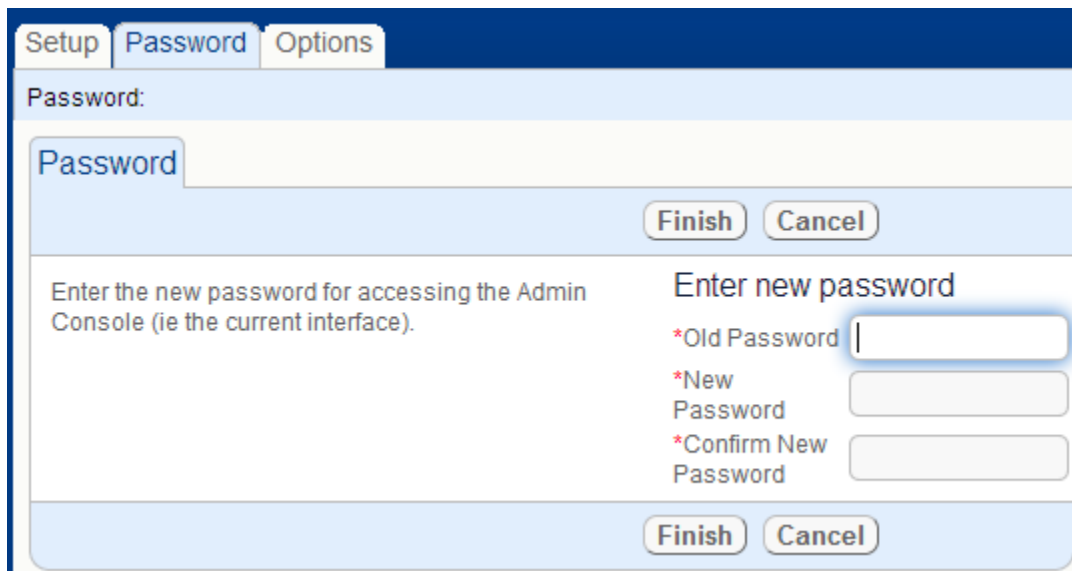
Password:

OK

[Forgot Password?](#) [Help](#)

This is the generic login dialog
For directions on providing a custom
login interface, click here

2. Select the **Password** tab:



Setup Password Options

Password:

Finish Cancel

Enter the new password for accessing the Admin Console (ie the current interface).

Enter new password

*Old Password

*New Password

*Confirm New Password

Finish Cancel

3. Enter the existing and new passwords and click **Finish**

Sample Users

Each KnowledgeBase you create is automatically populated with a number of sample users:

Contacts

Status: 35 record(s) found, 2 pages. Click [here](#) to re-count records.

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

Search:


Refine String Context Active Only [More Options](#)

<input type="checkbox"/>	Edit	ID	Full Name ↑	Contact Table	Company	Direct Phone
<input type="checkbox"/>		352	System	Employee	Agiloft	415-333-4444
<input type="checkbox"/>		412	Admin	Employee		
<input type="checkbox"/>		288	Anonymous User	Employee		
<input type="checkbox"/>		384	Chris Connoly	End User	NewsWire	123-444-5555:213
<input type="checkbox"/>		282	Christopher Caldwell	Employee	Agiloft	444-478-4444
<input type="checkbox"/>		406	Don Wilson	End User	State of California	
<input type="checkbox"/>		213	EW Admin	Employee	Agiloft	
<input type="checkbox"/>		351	EWSystem	Employee	Agiloft	415-232-4444
<input type="checkbox"/>		411	Guest End User	End User		

The users: **Anonymous**, **faquser**, **register** are essential to certain functionalities. Users like **admin** and **ewsystem** should have their passwords changed to be more secure. Remaining users should be deleted, as they are probably not relevant to your organization.

The **ewsystem** user is used by Agiloft customer support staff to assist customers. The default password, although not publicly known, is well known internally to the company. You should change this password to be secure, but please be sure that Agiloft either has the password on file, or that the person requesting support can supply the password if necessary.

To change the password:

1. Log into a KnowledgeBase and select Contacts/View Contacts, as shown above.
2. Click the Edit icon  to edit the user" alt="penicon.gif" class="inline"> to edit the user"s information, and select the **Account Info** tab:

End User

Save Cancel Contact Information Account Info Activities Support Items His

ID 390 Export to Outlook

Contact Information

First Name Happy Last Name Customer
 Title Director of Customer Support Email example25@example
 Opt Out Do not email me
 Direct Phone 234-222-3333 Ext. Cell Phone 234-555-6666
 Company NewsWire Company Role Customer
 Manager Name Manager Email

Login Information

*Login customer2 *Password *****
 *Confirm Password *****

Groups Customer Primary Team Customer Team

Supported User Yes
 Contact Owner Jack Richards
 Comments
 User time zone Choose one

Save Cancel <>

3. Enter the new password in the fields and click **Finish**.

Assign users to Agiloft groups carefully.

Users should not be assigned privileges they do not need or do not have the skills to use safely. For example, a user with the ability to delete all records in a table in one operation can do considerable damage accidentally if they are not familiar enough with Agiloft. Users in the **admin** group should only be those trusted and skilled enough to make structural changes in a KnowledgeBase.

Use SSL

Use SSL (via HTTPS) to secure Web browser connections to the Agiloft server. Using standard HTTP to connect to the Agiloft server exposes passwords and potentially sensitive information to anyone able to monitor network traffic, and opens up additional methods of attack to those able to intercept network traffic.

To connect to your Web server using SSL, you will need to configure it, as it is not the default configuration. You will need to purchase or generate a server certificate that authenticates your server to the clients. This configuration differs depending on the host operating system type and release, and the Web server software in use. The following resources may help:

[Securing Your Apache 2 Server with SSL](#)

[Van's Apache SSL/TLS mini-HOWTO](#)

[How to implement SSL in IIS](#)

Even if you must allow access to some accounts through standard HTTP, ensure that HTTPS is used to access more sensitive accounts such as those in the **admin** group of KnowledgeBases and the Admin Console.

Restrict login access to the Agiloft server machine.

A **root** user on Unix/Linux or a user in the **Administrators** group on Windows can circumvent Agiloft internal security by modifying program and data files or directly changing data in the database, including passwords. But even a regular unprivileged user can circumvent security by using local Web access to use special debugging features of Agiloft (such as the JMX console, as shown below) that are not accessible to connections from outside the server.



Restrict services accessible on the Agiloft server.

Treat the Agiloft server as you would any other sensitive server by only allowing connections essential for Agiloft operation, such as HTTP and HTTPS, and administration, such as ssh (Unix/Linux) or Terminal Services (Windows). Other services or applications run on the same server machine, including other Web

applications, could potentially contain security holes which could lead to the compromise of Agiloft data.

The default services installed with most recent Linux distributions are generally minimal. You should use the [nmap](#) tool to verify which ports are exposed on your server. For example:

```
linux# nmap -sS wizard.example.com
Starting Nmap 4.00 ( http://www.insecure.org/nmap/ ) at 2006-12-14 18:12 PST
Interesting ports on wizard.example.com (10.0.0.1):
(The 1667 ports scanned but not shown below are in state: filtered)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
113/tcp   closed auth
443/tcp   open  https
8080/tcp  open  http-proxy
MAC Address: 00:E0:81:00:00:12 (Tyan Computer)
Nmap finished: 1 IP address (1 host up) scanned in 64.320 seconds
linux#
```

These are the TCP ports normally used by Agiloft:

80

The standard HTTP port that connects to the Apache or IIS Web server. The **/gui2/** URL is forwarded to the Tomcat server and is the normal unsecured access to the Agiloft application.

8080

The native connection port to the Tomcat server that is part of the Java framework behind Agiloft.

443

The standard HTTPS port for Web service over SSL. This is either forwarded to the Tomcat server by the native Web server or forwarded directly to port 8443 by Linux kernel using the internal firewall module.

8443

The native HTTPS port that Tomcat may be configured to listen to. It is often better to use the SSL engine in Tomcat (with requests forwarded from port 443) than to configure the native Web server for SSL and request forwarding.

3306

600

The standard server port for MySQL, the default Linux back-end database, This port is not exposed externally (i.e., it is bound only to **localhost**).

SYSTEM TIME

All time values displayed throughout Agiloft are based upon the KnowledgeBase time, as opposed to a users local time or Server time. The KnowledgeBase time is controlled through the Setup/System wizard and can be set manually, as a relative displacement from the server time, or by selecting the desired time zone.

Setup → System → Set KnowledgeBase Time

System Finish Cancel

This is the time zone set on the physical server (server that hosts your installation of Agiloft). To change the time zone, you need to configure time zone settings using the server's OS-specific time tools.

Server Time Zone
Pacific Standard Time (corresponds to GMT -8).

This is the time that is set on the physical server. You can adjust time on the server using server's OS-specific time tools.

Server Time
Monday, April 29, 2013 8:31:54 AM PDT

You may edit timezone abbreviations (GMT, PST, etc) and see detailed timezone info for every timezone by pressing "Edit Timezone Abbreviations" button.

Time Zone List Details
Edit Timezone Abbreviations

To configure the time zone for the current KnowledgeBase, please select a time zone from the list of available time zones, or set the time difference manually. If you select time difference manually, the KnowledgeBase time will be set to **Server Time plus the difference** you specify.

KnowledgeBase Time Zone

Select Time Zone
(GMT-08:00) Pacific Time (US & Canada); Tijuana - Pacific Standard Time

Set Time Difference Manually

Difference:

Daylight Savings Time:

Same as Server

Starts at midnight on

Ends at midnight on

Click **Refresh** to see the current KnowledgeBase time, based on your selections above.

Current KnowledgeBase Time
8:31 AM Refresh

Finish Cancel

- Server Time Zone is the time zone set on the physical server (server that hosts your installation of Agiloft). To change the time zone, you need to configure time zone settings using the server's OS-specific time tools.
- Server Time is the time that is set on the physical server. You can adjust time on the server using the server's OS-specific time tools.
- To configure the time zone for the current KnowledgeBase, please select a time zone from the list of available time zones, or set the time difference manually. If you select the time difference manually, the KnowledgeBase time will be set to Server Time plus the difference you specify.
- Click Refresh to see the current KnowledgeBase time, based on your selections above.

See Also:

[KnowledgeBase](#)

[Concepts](#)

VARIABLES

Global Variables

A global variable stores some value in a predefined format that is accessible throughout the KnowledgeBase. Global variables can be created or edited using Setup/System/Manage Global Variables. For more information on the available data types, see [Field Types](#).

Some global variables can be set in the Admin console so as to be available to all the KnowledgeBases on a particular server. However, if a global variable already exists in the KB then it takes precedence over the variables set in the Admin console. If the referenced variable is unavailable in a particular KB then the Admin console variables take over.

Setup → System → Manage Global Variables

Customized Variables Variables with Default Values

Status: 22 record(s) found

New | Delete | Edit

	Edit	Label	Name
<input type="radio"/>		Display Database Update Warning	warn_on_record_changed
<input type="radio"/>		Displaying Time Zone Label next to field	timezone_label_display
<input type="radio"/>		Dummy Communications	create_dummy_communication_records
<input type="radio"/>		Enable Hotkeys	hotkeys_enabled
<input type="radio"/>		Enduser Title	enduser_title
<input type="radio"/>		FTS Response Time	averageftsresponsetimelimit
<input type="radio"/>		Login URL	loginurl
<input type="radio"/>		Max Email Attach Size	max_size_of_email_attachments_in_kbytes

The global variables set in the Admin console are not visible in the KB setup.

Below is the list of available Global Variables:

Global Variable	Description
bug_report_destination	The URL which is used to automatically submit a bug report to the vendor's support system. By

	default:
	https://support.agiloft.com/gui2/login.jsp
report_frequency	Represents the minimum number of minutes that can be used in the "Every ___ minutes" option in the Report Wizard. If the user tries to save a schedule with a lower number of minutes in the Report Wizard, the system provides an error message that says: "The minimum interval for which reports can be scheduled is \$report_frequency minutes". Has a default value of 20 minutes.
manual_license_request_url	The URL that the user will be forwarded to once a licence is requested.
lock_expiration_time	Sets the time interval after which a edit lock on a record expires. When a record is opened by a user, it is locked to prevent simultaneous or conflicting edits. This variable accepts an integer value and specifies the expiration time in minutes. This variable works on per-project basis.
RevindexThreadsLimit	This variable limits the number of threads indexing KB contents at any one time. Agiloft starts one dedicated revindex execution thread per KB, so this is useful for performance tuning on servers that host multiple KnowledgeBases. The default value is set as "CPU number + 1".
show_stack_trace_button_on_sod	Choice [Yes/No], whether to show "Show StackTrace" button on the automatic bug report submission screen (SoD). This can be set as a default for all of a server's KBs via the Admin console, and overridden in individual KnowledgeBases.
include_jboss_log_file_with_sod	Choice [Yes/No], whether to include

	the tail of the server log in the data submitted from the automatic bug report screen (SoD). Defined in the Admin console for all KBs at once.
collect_system_information_on_sod	Choice [Yes/No], whether to collect and include system info in the data submitted from the automatic bug report screen (SoD). Defined in Admin console for all KBs at once.
require_matching_cookie_and_seance	String [No, Admin_Users, Staff_Users, All_Users] whether to check for the presence of the security cookie on KB access for given users. Defined on per-KB basis.
casserverloginurl	Used to integrate with a Centralized Authentication Server (CAS). Defines the URL to which the system redirects a user, to authenticate against CAS before proceeding with login to Agiloft. For example: https://{yourhost}/cas/login. Defined on per-KB basis.
hotlinktype	Currently used in integration with Centralized Authentication Server (CAS). Defines what type of hotlink Agiloft sends in the outgoing emails. Accepted string values: STANDARD or CAS. Set on per-KB basis.
swheader_text	HTML to be displayed next to the Saved Search Pane.
product_name	Used for branding to replace "Agiloft" with your product name everywhere where it is supported. Set on per-KB basis.
disconnect_timeout and warning_timeout	By default a user is disconnected after one hour of inactivity. This timeout can be changed by adding a global integer variable "disconnect_timeout" and specifying a non-zero value in

minutes.

Display a warning in a popup by adding a global integer variable "warning_timeout" specifying the value in minutes from the start of the inactivity period. If it contains "0", no warning will be shown.

max_report_size

This variable was introduced to gracefully handle reports with big amounts of data. While preparing a report the system approximates the size of the resulting file for each 100 rows, based on the size of the first 100 rows rendered as HTML report. When the estimated number approaches 90% of the specified value, the number of rows shown for all further groupings will be reduced to 25, although the summary values will be calculated over all rows being grouped. When the estimated number reaches 100% of the specified value, all further rows will be omitted and the report will only have the summary lines for each further grouping. The default value is 10 Megabytes.

staff_title and enduser_title

These strings replace the title in the browser that says "Success.... Guaranteed". These variables should be set on per-KB basis.

ExitURL and LoginURL

When users log out explicitly by clicking the Log Out link, they are taken to the location defined by the ExitURL. This URL can also be overridden by parameters in the custom login block or hyperlink.

When users are logged out automatically for any reason (wrong login or password, timeout, etc.) they are given the LoginURL so they can log back into the

system.

Define these global variables to ensure that users never see the default system login screen.

Many components of the system require to know the host name of the system. For example the outbound email can include a hyperlink to log users into the system. To generate this hyperlink the system has to know its own host name.

hotlinkServerRoot

This variable is also used in company branding from a hosted server, and to address non-standard installation scenarios. See more details on what effect changing this value may have in the FAQ record 13989 available for view [here](#).

Set this global variable on per-KB basis.

Defines a delay inserted after each REST/SOAP web services operation has completed. Configurable via global variable in Admin Console only i.e. system-wide. Assumed to be set to 1 second by default if absent.

WSDelay

This has been done due to the frequent mistake of using a web services client to bombard Agiloft with a constant barrage of requests. Normally an operation on the record invokes rules and other functionalities, they should be allowed enough time and resources to complete.

max_excel_export_size

Default value = 200 Mbyte.
Maximum size of the export Excel file. Once system reaches the limit,

it stops writing into the file.

Main reason for introducing this variable is to prevent high memory usage during the export to Excel.

Please note that there is no need to set up a limit for Excel 2007 (.xlsx format), as it allows more memory efficient export

A new global variable is added to the admin console called:

`max_size_of_email_attachments_in_kbytes`

with a default value of 10000 (the equivalent of 9.7 MB). It prevents a performance problem that can arise on sending emails with huge attachments.

If one or several files are excluded because they are greater than the limitation, an error message is provided:

`max_size_of_email_attachments_in_kbytes`

The file `$file_name_1` was not attached because it is `$kbytes` kbytes in size and the default or specified value of the global variable

`max_size_of_email_attachments_in_kbytes`

in the admin console specifies a maximum size for attached files of `$max_size_in_kbytes` kbytes.

To add or edit the variable login to the admin console and click on the variables tab. Create a new variable of type integer or edit the variable if it has been added.

`global.user_agent`

A new variable, `$global.user_agent`, will display standard information about the current user's browser and operating system. This variable

can be used as the default for any text field just like other formula variables.

This variable can be used to give support staff detailed version information automatically and with increased accuracy, especially if the staff had to collect it manually or request it from the user.

Example of variable contents from a user using Windows and Firefox to access Agiloft: Mozilla/5.0 (Windows; U; Windows NT 5.2; en-US; rv:1.9.0.6) Gecko/2009011913 Firefox/3.0.6

Using the variable, this type of content can be displayed in any text field.

Per-KB global variable. Default value of 22-04. The value refers to the KB time when the optimization task should run, for example 22-04 refers to 10PM to 4AM.

AllowableOptimizeTablePeriod

This will allow scheduling the optimization during nights or weekends and will prevent the production servers from slowing down during business hours.

System Constants

Unlike Global Variables that can be edited or deleted, System Constants are used in read-only mode. They are defined for the entire KnowledgeBase. For example, the "\$global.daylightServer" constant returns *true* or *false* depending on whether the server is configured to account for daylight saving time.

System constants have the prefix "\$global". A full list of system constants is available in the Global Variables tab of the Formula Wizard.

Label	Variable name
averagetimetoclosetlastmonth	\$global.averagetimetoclosetlastmonth
casesclosedlast3months	\$global.casesclosedlast3months
casesopenedlast3months	\$global.casesopenedlast3months
contracts_renewed_last3months	\$global.contracts_renewed_last3months
criticalcaseslast3months	\$global.criticalcaseslast3months
daylightServer	\$global.daylightServer
ip_address_of_user	\$global.ip_address_of_user
isTimezoneManual	\$global.isTimezoneManual
largeobjects.repository_path	\$global.largeobjects.repository_path
leads_created_last3months	\$global.leads_created_last3months
LEXICON_MAX_DIGITS_INSIDE_WORD	\$global.LEXICON_MAX_DIGITS_INSIDE_WORD
LEXICON_SKIP_HEX	\$global.LEXICON_SKIP_HEX
LEXICON_SKIP_HEX_LENGTH	\$global.LEXICON_SKIP_HEX_LENGTH
monthly_total_quotes_to_close	\$global.monthly_total_quotes_to_close

User Variables

Saved Searches and Reports can use variables based on information obtained from the user table. If a saved search reads "Company=\$global.my_company" and there is a field named Company in the User table, the value of this field for the current user would be substituted into the search when it is run. For a user who works for Acme, the resulting search condition would thus be "Company=Acme".

When you are running a report, these values are substituted at run time for each user, so a given report can be sent to multiple users and contain the appropriate information – James may get a report on all bugs that he submitted, while Mary gets a report on all the bugs she submitted.

You can also refer to values in another field from the current table by prefixing the field name with "\$", so, \$amount would return the value in the Amount field for the current record.

To use a variable in a formula expression, wizard field, search or report, you need to prefix it with a "\$", for instance, \$conversion_rate. Whether you are using a global variable, system constant, or user variable, you should refer to it by the variable name, not the label.

See Also:

[Formula](#)

[Field Types](#)

[Choice Lists](#)

[KnowledgeBase](#)

LICENSING

Agiloft licensing is based on the concept of active named users. An active named user is someone who is registered to use the software and has actually done so in the past 30 days. There may be 10,000 users in the user table or LDAP repository, but if only 10 of them have used the system in that past 30 days, you only need a license for 10 users. This reduces costs by ensuring that companies only pay for licenses they actually need and eliminates the burden on administrators of having to free up licenses used by ex-employees.

There are two classes of user, staff and end users. Staff users have the authority to login to the staff interface and, depending on their group membership, can have the authority to edit records created by other users. End users are restricted to the easier, but less powerful end user interface and are not able to edit other user's records. End users are usually not licensed individually, but as a percentage (25%) of staff licensing for unlimited access. For example, purchase of a license for 10 staff with unlimited end users costs 25% more than a license for 10 staff and zero end users.

Licensing is based on the hostname and IP address of the server, so if you move the installation from one server to another, you must generate a new license. This service is covered at no additional cost under the maintenance agreement.

See Also:

[KnowledgeBase](#)

[Administrator Console](#)

[Staff Interface](#)

[End User Interface](#)

LOCALIZATION WIZARD

To translate KnowledgeBase specific information such as choice tables, saved searches, table and field labels, charts and dashboard names into other languages you can use Localization Wizard (Setup/Localization). You can translate the program itself, such as status messages, wizard help (eg this text) and program labels into other languages.

In the wizard, choose a target language for your installation of Agiloft.

Choosing a language from the list will not affect the user interface immediately; it will generate a file that you can then translate to the desired language. After installing this file you can switch to the new language.

When new fields or choice lists are created, they are assumed to be in English. However it is perfectly okay to create fields in your local language first without translating into English. If you later decide that you want them to display in English, edit the English to English Translation table and create entries converting from your local language to English.

Agiloft has a feature that shows localization keys in the GUI together with their values. It is controlled by a "ew.nls.debugging" system property. This property can be toggled either in the run.sh file or using a beanshell console: login into admin interface, go to Debugging tab, click Beanshell button. Put the following in the appeared window and click "Execute"

```
System.setProperty("ew.nls.debugging", "true");
```

To disable debugging run the same code again, just replace "true" with "false".

PERFORMANCE TUNING

As detailed in <http://www.agiloft.com/enterprisewizard-scalability-and-redundancy.pdf>, Agiloft can support thousands of users and generate over 200,000 records per hour on an inexpensive server. The following tips will enable you to maximize system performance.

Reduce the amount of data that is transferred

1. Click Set View/Edit and minimize the amount of data that has to be transmitted to display a table view as follows:
 - a. On the Fields tab, only show those fields that you really need to see.
 - b. On the General tab, display 5-20 records per page (rather than the maximum of 150).
2. Minimize the frequency with which view data is refreshed by selecting Setup/Access/Manage Teams/Edit a Team and set the Table View Refresh Rate to Never. The table will still refresh whenever users return to it from somewhere else or from editing a record. Note that this is also the only way to time people out for inactivity - if refresh is on, they are never timed out.
3. Reduce the amount of data transferred during record editing.
 - a. Move any related tables or embedded search result tables off of the first layout tab so they can load in the background.
 - b. If you have linked fields displaying as a value list with more than 50 values, change the display type to a box with lookup so the list doesn't have to be loaded.

Use a fast browser

Different browsers provide vastly different performance. Firefox 3.5 is the recommended browser and is available on all platforms.

Safari and Chrome are as fast as FireFox and are fully supported, except for the admin-only graphical workflow editor, which is prevented from working by some limitations of these browsers. The Google Chrome Frame add-on for IE seems to accelerate IE8 to Chrome speed while still enabling the workflow editor (download it [here](#)).

IE8 is significantly faster than IE7 and fully supported, but still slower than Firefox, Safari or Chrome.

Avoid using email fields to define ownership

The problem with email fields is that they can contain all the data from the header in the incoming email, for example the field might contain "John Doe john.doe@gmail.com", yet only the john.doe@gmail.com portion is used to define

ownership and although we automatically add an index for fields that define ownership, MySQL cannot use an index in this case. We are working on an enhancement to address this limitation.

Do not pass unnecessary records to an if-then-else action in a time-based rule.

Passing a large number of records to an if-then-else action which then ignores 99% of them can cripple performance. The reason is that each record must be read from the database, with all Agiloft functionality applied to it, before it is passed to the if-then-else action.

Happily, there is a simple solution - create a saved search on the Condition tab of the rule that will only find records that some action within the if-then-else will actually operate on.

For example, the following rule is very inefficient if there are a lot of open records, but only a few that are Urgent or Critical.

Condition: Find all open records

Action: If (Priority == Critical)

Then Send Escalation Email to Team Leader

Else If (Priority == Urgent)

Then Send Escalation Email to Team

Else do not do anything (all these records have been opened, parsed and then ignored)

It is much more efficient if the rule is constructed with:

Condition: Find all open records where Priority == Critical or Priority == Urgent

Tune the database

If you are comfortable with databases and have a good feel for the makeup of the data and what searches are most frequently exercised, it can be very helpful to add database indexes. If you are unsure or need the changes to be made outside of working hours, it may be worth using our consulting services to take care of this task.

If you often search on a field such as Assigned To or Status and the number of matching records is much lower than the number of records in the knowledgebase, you should be able to improve performance by adding a database index for that field.

For example, adding an index to the Assigned To field will speed performance if there are 100 records assigned to a particular team or individual in a database of

100,000 records, but if there are 50,000 records assigned to that individual, it will not help at all and may actually hurt performance.

Although database indexes can be very effective, you have to be careful because:

1. Database locking makes the table unavailable while you are adding the index. Depending upon the number of records in the table, this may take between a couple of minutes and several hours for tables with hundreds of thousands of records.
2. Inappropriate or excessive indexes will actually hurt performance. In general, you should not have more than 7 or 8 indexes per table.
In some cases, such as searches for "Assigned to == \$global.my_Full_Name and Status == Open", it may be most efficient to add a compound index.

SQL Statements taking a long time to execute

To find out what SQL statements are taking a long time to execute, login to the mysql console as follows:

1. Using the host/port/password from /etc/EnterpriseWizardConfig.xml file:

```
<databaseServerHome>/usr/local/EnterpriseWizard/mysql</
databaseServerHome>

<databaseAdminLogin>root</databaseAdminLogin>

<databaseAdminPassword>123456</databaseAdminPassword>

<databaseAddress>127.0.0.1</databaseAddress>

<databasePort>3333</databasePort>
```

2. Run the command:

```
/usr/local/EnterpriseWizard/
mysql/bin/mysql -u root --host
IP_ADDRESS --port=PORT -p sw2_std
-A
```

3. This will typically be:

4.

```
/usr/local/EnterpriseWizard/
mysql/bin/mysql -u root --host
127.0.0.1 --port=3333 -p sw2_std
```

```
-A
(MySQL will prompt you to enter
the password that you found in
/etc/EnterpriseWizardConfig.xml)
```

5. Run the following command to list all sql statements running for more than 15 seconds:

```
select * from
information_schema.processlist
where command != 'Sleep' and
      time>15;
```

Large number of tmp tables being created

The reason that a large number of tmp tables can be created is because there may be no covering index for a query in a table. The Agiloft optimizer tries to find a covering index for the SQL query turned on for the table, and informs the MySQL optimizer about such index, or logs a warning to the server.log about the absence of such an index to the server.log. Detailed activities of the optimizer can be obtained by turning on OptimizationLog debug output. This will cause information to be written to ewoptimizer.log.

To do it you need change INFO level to DEBUG level at .../jboss/server/sw/conf/log4j.xml at next sections (no jboss restart is required):

```
<appender name="EWOptimizer">
  <param name="File" value="$ {jboss.server.home.dir}/log/ewoptimizer.log"/>
  <param name="DatePattern" value="'.yyyy-MM-dd'"/><!-- Rollover at midnight each day -->
  <param name="Append" value="true"/>
  <param name="Threshold" value="INFO"/>
  <layout>
    <param name="ConversionPattern" value="%d %p %x %.25c:\n%m%n\n"/>
  </layout>
</appender>
```

and

```
<category name="OptimizationLog">
  <priority value="INFO"/>
  <appender-ref ref="EWOptimizer" />
</category>
```

After 1 minute the ewoptimizer.log will start being populated. The most interesting parts are successes and failures.

Successes

```
=====USER:akbaheti=====
```

```

(((linked_id=852)&&(linked_table=company))&&(-TEXT~=_E_W_%))&&((type<<261)||
((((type<<262,263)&&(f_from!=abc)||
(f_from=com.supportwizard.filters.parameters.NullParameter@38018f91)))&&((linked_table
!=contacts.employees)||
((MultiValueLFSelectionCriteria {condition=1, swSavedSearchID=1757,
internalSelectionCriteria=null, fieldName='f_to'}))||
(f_from=$global.my__login))))||((type<<259,260)&&(f_from=abc)))||
((type<<259,260)&&((linked_table!=contacts.employees)||
((MultiValueLFSelectionCriteria {condition=1, swSavedSearchID=1757,
internalSelectionCriteria=null, fieldName='f_to'}))||((f_from=$global.my__login))))))

```

```

-----
OptimizationInformation {optimizationPossible=true, joinsNumber=2,
possibleJoins={f_from=4}, impossibilityWarning='null',
columnDBNames=[id, linked_table, f_from, type, linked_id],
selectedIndexDBNames2JoinColumns={cvr_2=[f_from]}, asIS=true}

```

Here we see a final query internal representation (subject to optimization) and the result of the Agiloft optimizer. The Query is some combination of used saved searches, and various groups permission filtration. In different situations, more filters may be added automatically:

- optimizationPossible=true, means success
- joinsNumber=2, number of joins at the final query
- possibleJoins={f_from=4}, possible joins
- impossibilityWarning='null', null for success.
- columnDBNames=[id, linked_table, f_from, type, linked_id], columns used at the query
- selectedIndexDBNames2JoinColumns={cvr_2=[f_from]}, selected covering index with fat field to join optimizations
- asIS=true, internal flag.

Failures:

```

=====EWOptimizationFail=====USER:rmansfor=====
((((linked_table=1545)&&(linked_id=328059))&&(((type=260)&&((email_type=1)||
(outgoing_email_type=1))))||((type=261)))&&(-TEXT~=_E_W_%))&&
((type<<260,261)||((((type<<262,263)&&(f_from!=abc)||
(f_from=com.supportwizard.filters.parameters.NullParameter@38018f91)))&&
(linked_table!=contacts.employees)||((MultiValueLFSelectionCriteria {condition=1,
swSavedSearchID=1757, internalSelectionCriteria=null, fieldName='f_to'}))||
(f_from=$global.my__login))))||((type=259)&&(f_from=abc)))||((type=259)&&((linked_table!
=contacts.employees)||
((MultiValueLFSelectionCriteria {condition=1, swSavedSearchID=1757,

```

```
internalSelectionCriteria=null, fieldName='f_to')|(f_from=$global.my__login))))))
```

OptimizationInformation {optimizationPossible=false, joinsNumber=0, possibleJoins={},
impossibilityWarning='There is not a covering index for the query.

Sorting (first, in order) fields: ([]),

Required fields: ([id, email_type, linked_table, f_from, outgoing_email_type, type, linked_id]),

Optional fields: ({}).',columnDBNames=[], selectedIndexDBNames2JoinColumns={},
asIS=true}

impossibilityWarning is the only informative part.

Sorting (first, in order) fields: ([]), covering index should start with these columns

Required fields: ([id, email_type, linked_table, f_from, outgoing_email_type, type,
linked_id]), fields from the query

Optional fields: ({}). set of fat fields where join optimization can be applied.

How to Construct Optimal Indexes

As described above, the optimum index for queries reported in the ewoptimizer.log, or obtained from the MySQL admin report on long running queries is generally one that includes the fields from the query, in the same order that they were specified in the query. Please note that creating the best index does require some experience and judgment; if it did not, the standard would be for the database engines to create indexes automatically.

Load Testing and Performance Optimization

We provide consulting services to load test and optimize your application. These range from reviewing the log files and creating/modifying indexes, to simulating user load and fine-tuning performance based upon this load.

The time taken for this optimization can range from one day to several weeks, depending upon whether you want us to simply review log files and create/modifying indexes based upon this information; or create test programs that actually simulate production use and precisely measure the number of concurrent users that your server can support.

IMPLEMENTATION TIPS

ORDER OF IMPLEMENTATION

The ideal order of implementation is generally as follows:

1. Configure the Activity log to track all admin activities for an audit trail in case the customer comes in and messes with the implementation.
2. Create any new teams and set their working hours.
3. Work through all of the tables, one at a time, and create all their fields and linked relationships. This is generally an iterative process because some data types like Related Table cannot be created until you have created linked fields in the opposite direction. So you may have to return to each table after creating the others to do these data types.
4. Create the workflow states for each table and then go back and add the workflow field to the layouts, change its label if necessary, etc.
5. Set up inbound and outbound email addresses and email behavior.
6. Create all the rule and workflow automation.
7. Create login page(s) and access and define the global variables for the login page locations and any other settings.
8. Import users and data (users without their email addresses until ready to go live, so they will not be emailed during testing of rules). Create test users in each group and team with email accounts, gmail or yahoo.
9. Set up the end user interface: FAQs, My Item default searches, available saved searches.
10. Create custom look and feel schemes for staff and end user interface.
11. Perform any integration and further data import necessary.
12. Test the system.
13. When ready to go live, reimport users' email addresses as an update import to populate the system.

See Also:

[Initial Implementation](#)

TEAM'S SETUP

If working hours will be used, provide them for the company team and then use that team's hours for the other teams (unless they differ, of course).

Put a word in the Team Description that you will use in a search filter on any assigned team fields to find just the teams you want to appear on the list.

Always include the word Team in the team name so it is obvious you are dealing with a team if you see the value in the field.

If you are going to use team leaders, they have to be added after the users exist, so you will need to cycle back to the teams after importing the users.

Always use a team like the company team as the parent team unless you are dealing with lower level teams. This should allow the subteam to inherit the default view permission from the parent team.

Where practical, rename rather than deleting unused default teams to avoid being asked to change the team field for existing demo users.

See Also:

[Initial Implementation](#)

[Order of Implementation](#)

SAVED SEARCH TIPS

When creating saved searches for different purposes, prefix the search name with the appropriate letter shown below to show at a glance the purpose of the search.

R: search used in a rule

P: permission search (filter used in group permissions)

F: search used to filter a related table or the available records in a linked field set

C: search used in a chart/report

Also, any new search you create for one of these purposes should be made available to the admin group and should usually also be made active for them (unless there is a table that is going to have more than 30 search rules, in which case don't make each one active, just make them available). Otherwise other admins will not see these searches.

It is a pain to rename saved searches, as it is a bit buggy and doesn't allow you to rename without creating a new search if you are doing it from within a rule or certain other places, due to refresh problems. If you do need to rename a search, do it from the main Searches/Manage menu in the upper right of the screen. This method gives the search the new name in any rules that are using it.

See Also:

[Initial Implementation](#)

[Order of Implementation](#)

[Team's Setup](#)

TABLE CONFIGURATION

If a customer is using Agiloft for a function that is not included in the default KB and is not using a default table, consider whether you can rename one of the default tables to benefit from the preconfigured reports and charts.

Be careful which item in the table tree is selected when you create a new table - select the top level of the hierarchy for a new top level table, or an existing table to create a subtable of it.

- Completing work on a table involves the following tasks:
- Create the table and all the fields
- Create a good layout that uses the fields in a reasonable order.
- Set the owner permissions on the table wizard permissions tab according to the spec (this detail is usually located above the group permissions section for each table).
- Create a default view (sometimes the default views are specified in the spec for what they should contain, otherwise use judgment about what fields ought to be included) and apply it to all teams and their subteams both as being available and as the default.

FIELD TASKS

For a new table some general field cleanup is likely to be necessary:

- Correct the History field that is automatically created - it must have the following changes made to it: General tab: change the field label to upper case: History. This way we know it has been touched. Permissions tab - give admin all permissions. Display tab - check all the boxes for the columns to display. Be sure to put the history field on a tab on the layout, usually on its own tab displays the best.
- Change the tab name for Communications in the layout to Emails to make it take less space.

Specific Fields Tips

Here are some general tips about specific fields types and things that almost always need to be changed from their default behavior:

- For a Related table data type, be sure to edit the label on the first screen and on the display tab, check the box to left justify it. This makes a better display.
- Always when creating a linked set that includes ID field, edit the display characteristics for the ID field itself on the Display tab to make it 10 characters or less in width, otherwise it has a bad default and takes up about 70 characters, which looks bad.

- Never use the data type for a "link to single field in other table" EVER, EXCEPT when multiple values must be enabled. It is inevitable that when you do this, the customer decides they want to include a second or 3rd field in the set, and it has to be completely redone. Always use link to selected fields, whether or not it is specified.
- When creating a linked field set, on the Mapping tab, never choose Do Not Update while leaving the box that enables non-source values empty. This will cause errors. "Always Update" should be selected in 99% of cases. If you enable non-source values you can use update matching fields instead.
- For a linked field with a default value, on the Options tab, always choose one of the two radio buttons that define whether to update when the underlying condition changes,

Set default value for the imported fields based on:

The values from the record(s) in which the
 Accounting Period
 field matches the value in the
 Account Number
 field in the current Support Case
 Choose the record in which the value of the
 --Select one--
 field is less than all other matching records

run the Change Status To Reopened action when the
 or edited

The values for the user in the Choose one
 field of the user who changes the Choose one

None

Please select the desired behavior:

Once the field has a value, never overwrite it

Update the field whenever the underlying condition changes

Only update when the field in current table changes

- When setting up date/time fields that will never be entered by user, don't use calendar and popup display, use simple text box 22 wide.
- When doing choice fields with only 2 choices, typically radio buttons look better, on the display tab explicitly set it to 2 per row.

- When creating a text or short text field, don't skip the Display tab - you will almost always need to set the size of the display box away from the 70 default value.
- Wide text fields should all have the same display width to look neat (unless you use the "expand all fields to the same width" option on the alignment tab of the layout.) Typically 90 or 95 is a good width that works on most screens.

LAYOUT TIPS

Typically, it's suggested to change the default alignment for all layouts to use the second alignment option (not the first default one.) Depending on the table, I turn off the expand all fields to the same length option if it makes choice fields too wide and funny looking

For new tables, don't forget to create an end user layout (unless you know no end users will be accessing this table at all), typically by copying the staff layout to the end user layout to start with, so at least something will show up during testing.

As a general rule, a 2 column layout looks best - don't put a single field on each row (unless it is a wide field).

Use the Preview button to see how the layout looks (you can't preview the end user layout). Know your fields and their widths to make the best layout.

If you are using visibility dependent fields (fields that only show up if another field has a certain value), always put them to the right or on their own row - never use them in the left column before a field that will always appear.

Likewise, if you are putting a field on the form that only some users can see, don't put it in the lefthand column followed by a field that always appears. The missing field's space will not be compressed, and it will show up as an empty space in front of the field to the right.

Be sure to hit finish after editing a layout before leaving the table wizard or your changes will not be saved!

DEFAULT VIEW TIPS

On a new table, always create a default view and name it default. Views generally look clean if they are limited to 2 lines in height. Be sure to include an edit icon, and if you are going to use the view in a report, be sure to show the ID field as a view hyperlink (check the box in right hand column) because this will allow a user looking at the report to click it to open up a view of the underlying record.

Typically about 100 characters wide is a good width, and choose how many records to display on a page - the more records, the longer to load, but the easier to navigate.

See Also:

[Initial Implementation](#)

[Order of Implementation](#)

[Team's Setup](#)

[Saved Search Tips](#)

GROUP PERMISSIONS

Always wait to create new groups until the last possible minute (before doing workflow but after all tables have become stable in terms of their fields.)

Before creating new groups, try to finalize the permissions for the groups that will be closest to the new groups in terms of their permissions. Then you can copy those groups to create the new groups and will have less work to do on the new groups.

In general, if a group can see the toolbar tab and edit other people's records, they should have the middle options selected for the view, search, and report tab permissions.

You need to know which field defines ownership for each table when setting the record and field level permissions. For instance, if ownership is based on a match of the Company field, but a group should only have permission to edit their own personal records, you will have to filter the Edit Own set on the record level permission screen.

The ownership criterion is not defined in the group screens, but rather on the Permission tab of the Table Wizard. It will also be indicated in the spec.

FIELD LEVEL PERMISSIONS

Generally there will be nothing in the specification about the details of the field level permissions that should be set for each group. There may be specific fields that are indicated as being editable only by certain groups (for instance the publication fields for FAQs), but otherwise, you have 2 choices when setting field level permissions:

- Make your best guess which groups should see which fields based on what you know about the use cases
- Set the permissions with the customer on the phone in a gotomeeting session and ask them to make every decision themselves.

If you have to guess, use the following general guideline - anyone who can edit a record as a staff user should probably have only view own and view other access to all date and auto-entered fields, and create, view and edit access to all the other user-entered fields. The ID field is an exception - it is auto-generated, but we always give create access to this one so people see it when creating.

Keep field level permissions clean, even when it is more work. If a group does not have record level create and edit permissions, technically it doesn't matter if all field permissions are turned on for the create own and edit own columns, because they won't be able to edit the record at all. But rather than leaving it sloppy so that the field permissions imply an ability to edit while the record level permissions contradict it, make them consistent - de-select all field permissions for the columns for create own, edit own, edit others. Otherwise, someone may accidentally turn on

record edit permissions and suddenly those users will see fields never intended for their editing.

Always deselect the create own box for the communications "field", since no one should be creating an email when creating a record. Also deselect history for that column.

See Also:

[Initial Implementation](#)

[Order of Implementation](#)

[Team's Setup](#)

[Saved Search Tips](#)

[Table Configuration](#)

RULES CONFIGURATION

Because rules cannot be sorted by table yet, we suggest naming the rule with the table name as the first word, so we can sort by the comments field to find all rules for a particular table. It's also handy to be able to group the Time based rules vs. the other rules. So you can use 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. So 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:

- all new record actions
- all new validations in an if-then action
- Rule 1: 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 staff users update.
- Rule 4: Edit actions when end user updates.
- Rule 5: Parent-child handling (if complicated, 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: 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. 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 retrigger that rule. That is why we usually suggest separating 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.

Rule 7 - infinity:

Time based rules - We suggest 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 that are a pain to sort through to figure out how they interact, and they are much more difficult to troubleshoot, control priority, and document.

See Also:

[Initial Implementation](#)

[Order of Implementation](#)

[Team's Setup](#)

[Saved Search Tips](#)

[Table Configuration](#)

[Groups Permissions](#)

BROADCASTING

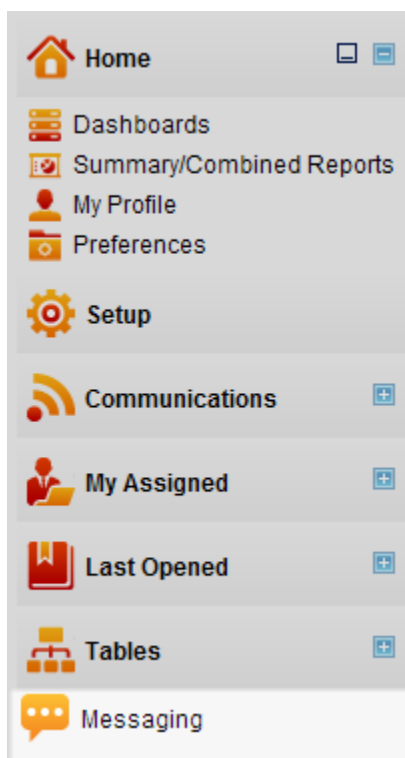
DIFFERENT WAYS OF BROADCASTING YOUR MESSAGE

Being an Admin you may find that you need to send round a notification about urgent update, server shutdown time or maybe something that is only relevant to a certain group of people.

There are different ways to do that and you can choose the one that is appropriate or suits you the best.

Using Messaging dialog in Staff Interface to broadcast within one Knowledgebase

To send a broadcast within one KB go to the Messaging, it's on the left pane of the Staff Interface.



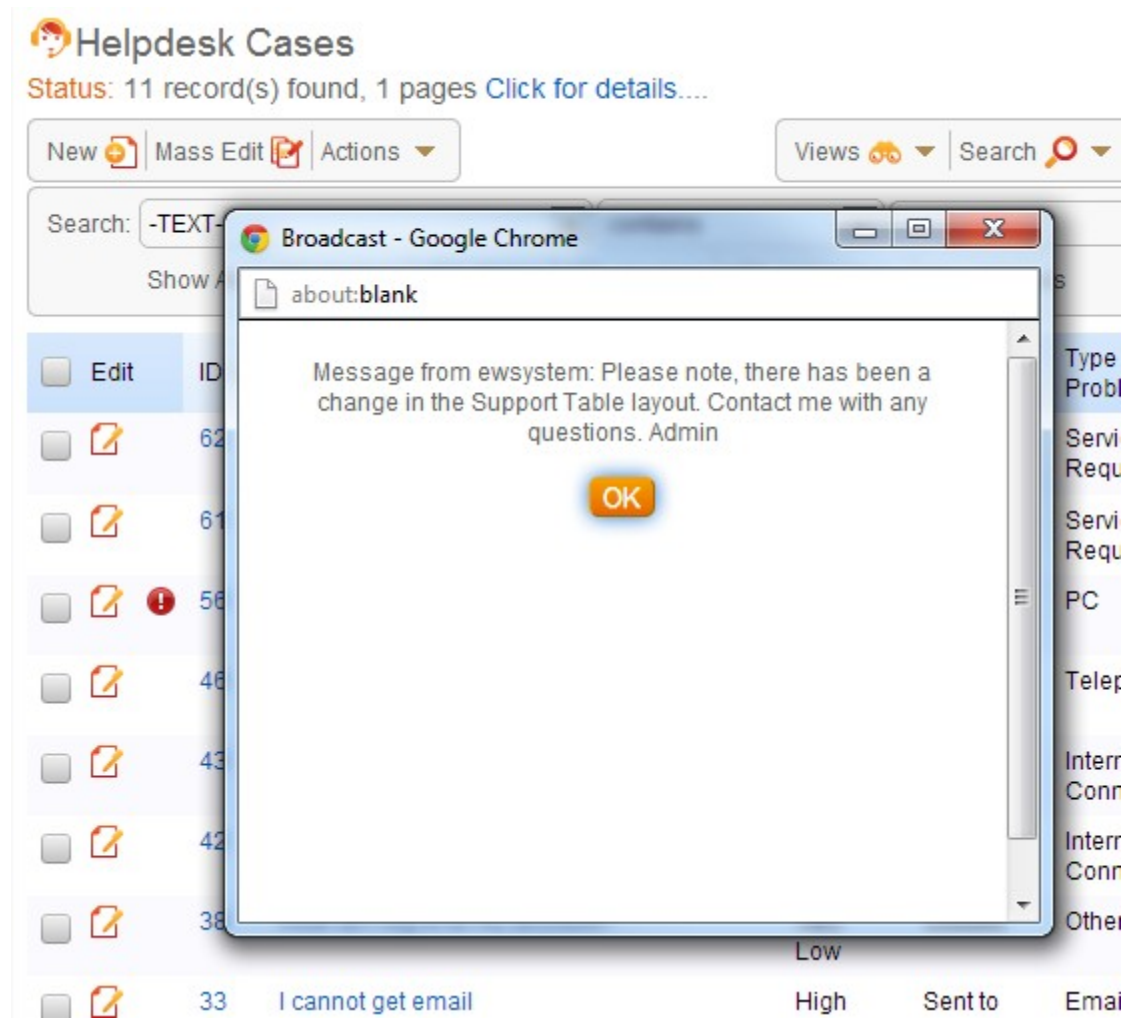
You can send your message to a number of groups, one user, chat users or all of them.

The screenshot shows a dialog box for sending a message. At the top, there are 'Send' and 'Cancel' buttons. The dialog is divided into two main sections. The top section is for entering the message. On the left, it says 'Enter the message that you would like to broadcast.' On the right, there is a text area labeled '*Message:' containing the text: 'Please note, there has been a change in the Support Table layout. Contact me with any questions. Admin.' The bottom section is for selecting recipients. On the left, it says 'Select the recipients for this message. If you want to send a message to all KnowledgeBases, you need to log in to the ADMIN interface.' On the right, there is a 'Recipients:' section with three radio buttons: 'Everyone' (unselected), 'Only these people:' (selected), and 'Do not show message to author' (checked). Under 'Only these people:', there are two checked options: 'Teams' and 'Users'. The 'Teams' option has a dropdown menu open showing a list of teams: Marketing Team, Support Team, Sales Team, Admin Team, Professional Services Team, and System Admin Team. The 'Users' option has a search box containing 'Ralph Knowles' and a magnifying glass icon. At the bottom of the dialog, there are 'Send' and 'Cancel' buttons.

or only for End Users

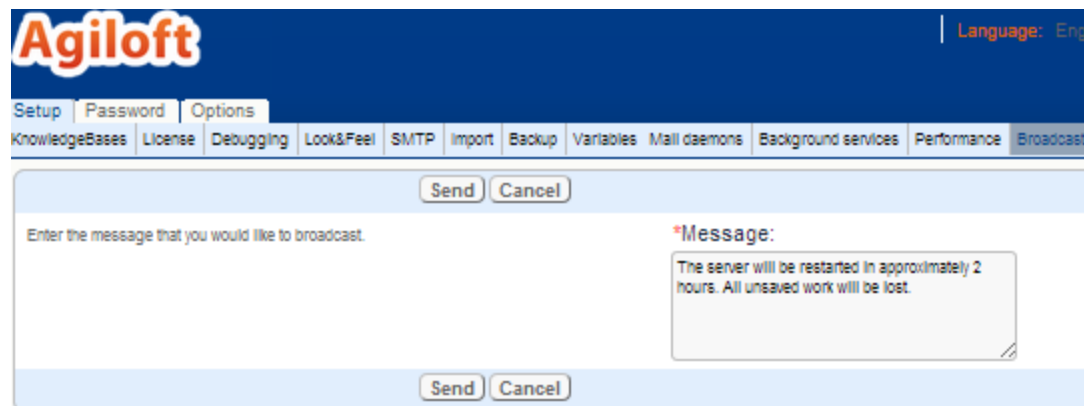
The screenshot shows a dialog box for sending a message. At the top, there are 'Send' and 'Cancel' buttons. The main area is divided into two sections. The first section, titled '*Message:', contains a text area for entering the message. The second section, titled 'Recipients:', contains several options: 'Everyone' (unselected), 'Only these people:' (selected), 'Teams' (checked), 'Users' (unchecked), 'Chat users' (unchecked), and 'Do not show message to author' (unchecked). A dropdown menu is open under 'Teams', listing 'Company Team', 'Internal Customer Team', 'Customer Team' (highlighted), 'Marketing Team', 'Support Team', and 'Sales Team'. At the bottom, there are 'Send' and 'Cancel' buttons.

The message arrives shortly as a popup in the Staff Interface or End-User interface, according to which teams received the message.



Using Admin console to notify users of all KB

To send a message to EVERYONE, to every user logged in to any of Knowledge Bases installed on this server, use Admin's console Broadcast feature (Setup/Broadcast).



Using REST request to send a message

By constructing an http request with specific attributes and parameters admin can send a message without logging in to the system.

Default message

You can send this default message when you about to restart the server:

```
Please save your work and logoff, the server is going to be
  restarted for the system checks in 5 minutes. It will be available in
  10 minutes.
```

To do that, use this request:

```
http://myewserver:8080/ewws/EWBroadcast?$KB=Demo&
  $table=chatmessage&$login=admin&
  $password=qwerty&$lang=en
```

Here KB attribute expects the name of your KnowledgeBase. The login and password pair should identify the user that has enough privileges to send a broadcast message.

The same request can be used to send the default restart message to ALL KB's installed on your server. For that, you need to use "admin" as KB name and valid login to the admin console.

```
http://myewserver:8080/ewws/EWBroadcast?$KB=admin&
  $table=chatmessage&$login=admin&
  $password=bigsecret&$lang=en
```

Your own arbitrary message

To send your own message you will need to use two attributes:

```
messagekey=clear_message and messageparam=your message text
```

So to send a message "Please save your work!", the request URL should look like this:

```
http://myewserver:8080/ewws/EWBroadcast?$KB=Demo&
  $table=chatmessage&$login=admin&
  $password=qwerty&$lang=en&messagekey=clear_message&
  messageparam=Please save your work!
```

Message from GUI2.properties

As part of localization solution the system uses a so-called properties file. This file contains all Agiloft system messages and texts. For example, the text you're reading in any wizard of the system is contained in this file. You should not edit this file unless you are adding support for a new language. This file is shared by all KnowledgeBases, so for security reasons you can only apply your changes if you have login access to the server itself.

The file contains pairs of keys and values. If you know the key name, you can access the value.

To download the file, go to Setup/Localization/Translate:

Setup → Localization

Language Table Options **Translate**

Back Finish Cancel

The translation file that contains all user-defined customizations has the following format:

User Translation File
 Dutch.txt

```
$String=$English_Text
```

Strings for which no translation is defined are blank to the right of the "=" sign.
 A typical string in the user translation file is "tables.cases.description=Sample Description Here". To translate this string into a target language, you would need to replace "Sample Description Here" with its equivalent in that language. Lines that start with # are treated as comment lines.

Note: You should update this file whenever you edit a table, e.g. by adding a new field.

When you're finished translating the language file, you can upload it here so that the translations are visible on the user interface.

Upload Translation File
 No file chosen

Note that the file has to contain ASCII characters only, but the translation should be in Unicode format. Non-ASCII characters e.g. symbols or glyphs of local alphabets should be encoded using a \uXXXX escape sequence where X is a hexadecimal digit. For example, the file might have the following entries:

- *s1=hello there*
- *s2=\u3053\u3093\u306b\u3061\u306f*

Note that empty messages in the file mean untranslated and are skipped

This file contains all Agiloft system messages and texts. For example, the text you're reading is contained in this file. You should not edit this file unless you are adding support for a new language. This file is shared by all KnowledgeBases, so for security reasons you can only apply your changes if you have login access to the server itself. [...More Help](#)

Program Translation File
 GUI2_nl.properties

Back Finish Cancel

For broadcasting using REST, we employ this concept to access predefined, often parameterized, text messages, that also can be translated in different languages.

It is recommended to consult our support in case you require to add a new key/value pair, otherwise existing ones can be used with any parameters.

See the example below, where we use the GUI2.properties key "server.isdown.soon" as a messagekey and value 15 as a parameter.:

```
http://myewserver:8080/ewws/EWBroadcast?$KB=Demo&
$stable=chatmessage&$login=admin&$password=qwerty&$lang=fr
&messagekey=server.isdown.soon&messageparam=15
```

This results in the following message being sent:

```
Please save your work and logoff, the server is going to be
restarted for the system checks in 15 minutes. It will be available in
10 minutes.
```

Using ssh connection to the server

If you have a login access to the server itself, you can send a broadcast message from the command line (Unix/Linux)

Navigate to the Agiloft installation directory and run this:

```
echo The server will be restarted for a system check. >
./tmp/message2broadcast
```

This message will be send to all KB's on that server.

See Also:

[Staff Interface Overview](#)

HOW TO GUIDE

TWITTER INTEGRATION

Once integrated, a new Twitter feed can be sent from within an Agiloft knowledgebase in real time. Responses can also be stored in the knowledgebase. Options provide flexibility for storing responses and publishing tweets.

AGILOFT SET-UP

To integrate Twitter, an admin simply selects the table from which they wish to publish tweets. Once the table has been selected, the user then designates a text field to be the source from which the tweets will be published. There are also optional fields that may be selected: a text field that may be designated for responses, and a yes/no choice field that may be used to allow for publishing on a case by case basis.

Any text field may be designated as the source field for tweeting. Any time a record is created, if the designated field has text, it will be posted to Twitter immediately when the record is saved (assuming it is to be published). It is important to note that if a record is edited and the text in the designated field changes, the tweet (as posted on Twitter) will be unchanged, and no new tweets will be added to the Twitter feed. In other words, any edits made to the source field after the creation of the record will not be reflected on Twitter in any way.

Additionally, a second text field (or append-only text field) may be designated as the «response» field. The knowledgebase will periodically poll Twitter (the frequency of how often Twitter is polled is configurable) so that whenever another Twitter user responds to a tweet for which a record exists, those responses will be captured during the poll and will either prepend, append, or overwrite the text in the response field based on the admin's selection during setup.

Lastly, any table designated for Twitter integration may include a simple yes/no choice field which will determine whether the contents of the Twitter designated field in the record are to be published as a tweet. The user simply designates a choice field using the yes/no choice list to determine if a record should be tweeted. If no choice field is selected in the wizard, the source field will be tweeted automatically.

Important notes:

- If the choice field option is turned on in the wizard, it is recommended that you set a default value for the field (or make it required), because if the field has not been set

by the user at record creation to either 'Yes' or 'No', the knowledgebase will interpret the value to be 'Yes' and tweet the text in the source field.

- If a new tweet is created outside of the knowledgebase (from within a Twitter client for example), the knowledgebase will not create a new record for the tweet, nor will it store it in any existing records.

CONFIGURING TWITTER

- In your browser, go to <http://dev.twitter.com>, and sign in to your Twitter account or create one if you do not already have one.
- Click the «Create an app» link.
- Fill out the required fields and click «Create your Twitter application».
- Click the «Settings» tab, and under the heading «Application Type», choose the «Read and Write» option, and then click «Update this Twitter application's settings».
- Click back to the «Details» tab, and click «Create my access token».
- Save/write down the «Consumer key», «Consumer secret», «Access Token», and «Access token secret» (See the following image).

OAuth settings

Your application's OAuth settings. Keep the "Consumer secret" a secret. This key should never be human-readable.

Access level	Read and write About the application permission model
Consumer key	f6WOxLJA052fIDHaD6YV9g
Consumer secret	102aY34Op53fusH1csGVyfmZmZXCATCsZtCbnrTfY
Request token URL	https://api.twitter.com/oauth/request_token
Authorize URL	https://api.twitter.com/oauth/authorize
Access token URL	https://api.twitter.com/oauth/access_token
Callback URL	None

Your access token

Use the access token string as your "oauth_token" and the access token secret as your "oauth_token_secret". Do not share your "oauth_token_secret" with anyone.

Access token	465867210-HMKoX5lAsGDsxP7m68UEcrPgkYrr7Chc3teNnSFG
Access token secret	caOS98e0WAUWX4oBioERZqtwekl6MY7IFNQDwUBbY
Access level	Read and write

[Recreate my access token](#)

- Once Twitter has been configured, login to your knowledgebase to continue set-up of Twitter Integration.

CONFIGURING THE KNOWLEDGBASE

- Once you've logged into your knowledgebase, click Setup->Twitter->New.
- Enter a description, then fill in the «Consumer key», «Consumer secret», «Access Token», and «Access token secret» from the Twitter configuration. Do not worry about the «Screen Name» field.
- Click the button "Test login and fetch screen name" – if everything has been entered correctly, the "Screen Name" will automatically be filled in for you and then click "Next".
- Select the table you wish to integrate with Twitter and click «Next».
- The "Store tweet in" option allows you to select a text field to be the source field from which tweets will be published.

Agiloft Reference Manual

- The "Store Tweet text in" option allows you to select a text field (or append-only text field) to store any responses to your tweets from other users. Additionally, you may select whether you want these responses to prepend, append, or to overwrite the text in the response field. See the images below to see the difference in how replies appear if a text field is chosen or an append-only text field is chosen.

Regular text field response appearance:

```
@ewtest1 some test reply
@ewtest1 second reply test
@ewtest1 third reply test
```

Append-only text field response appearance:

```
[ewtest1 Jan 16 2012 15:07]
@ewtest1 test append reply 1
[ewtest1 Jan 16 2012 15:07]
@ewtest1 test append reply 2
[ewtest1 Jan 16 2012 15:07]
@ewtest1 test append reply 3
```

- The «List of fields of type yes/no choices» allows you to select a choice field that uses the «yes/no» choice list to determine whether to tweet a particular new record on a case by case basis (only if it has a Yes or blank value will the tweet be sent).
- The «Polling Frequency» option allows you to set how often you wish the Twitter account to be polled for responses to your tweets.
- An example follows which shows a portion of the Twitter configuration:

*Description

test

*Screen Name

ewtest1

*Twitter Account Credentials

*Consumer key

N94fJKA2P6M0LTRHKL4QSw

*Consumer secret

LI01OtzS6rzUxU74SKTIN1e5V3DPZjZQaiutfnIac

*Access Token

465867210-uKgTN4JLwrIKvDb3IYjbNjPMfTGHUfLLnvJmSb12

*Access Token Secret

OUTZiW9eS5wG3uEnOlvBOA3U4pMpiXGIgjAuNfdh2Y

Test login and fetch Screen Name

*Account is enabled

- Yes
- No

POPULATING A LINKED FIELD WITH A SAVED SEARCH

This topic describes the new functionality of using a saved search to update values in fields linked from other tables. The primary reason this was implemented was to allow the use of saved searches to update linked fields instead of using complex formulas to achieve the same result. The ability to populate a linked field based on the results of a saved search gives the admin a more flexible way of populating a linked field beyond a simple match on a single field.

Previously, the only ways to populate a linked field were with a fixed value or based on a match on a single field. When the desired match actually involved checking values in several fields, complex compound fields needed to be created to provide enough data for the match. Even then there was still a limitation that it was very difficult to include information about the user causing the update to be made.

This functionality allows the use of any saved search to populate a linked field. One condition on the search is that it can't use any Run-Time filters since these require input from the user at the time the search is executed. The search results will be returned and either the first result or all results will be used, depending on the options selected. Note that using all results only makes sense for links that accept multiple values.

You can access the new functionality by editing the linked field(s) in your table and selection the Options tab. Under the Default Value section you will see the new option as seen in the example below.



run the action when the field is first set or edited

To summarize, the benefits of this enhancement include:

- The ability to populate a linked field based on the results of a saved search
- More flexibility in specifying values in linked fields

CREATING A NEW ACTION

Before you are able to set the action to run in the default value of a linked field, you will first need to set up the search and the action that will be used. In the example below we have a Companies table that contains linked fields to contract information. When the company is created or updated, and its Date Updated field changes, we repopulate this linked set with the linked contract record for that company that has a status of valid and has the latest end date of all matching contracts.

The first step is to create a new action to be used. To do this you would go to Companies/Setup Companies, then Rules. You can either edit an existing rule to get to the actions or create a new one.

Company

General | Graphics | Fields | Layout | Action Bar | Workflow | **Rules** | Permissions | Indexes | Convers

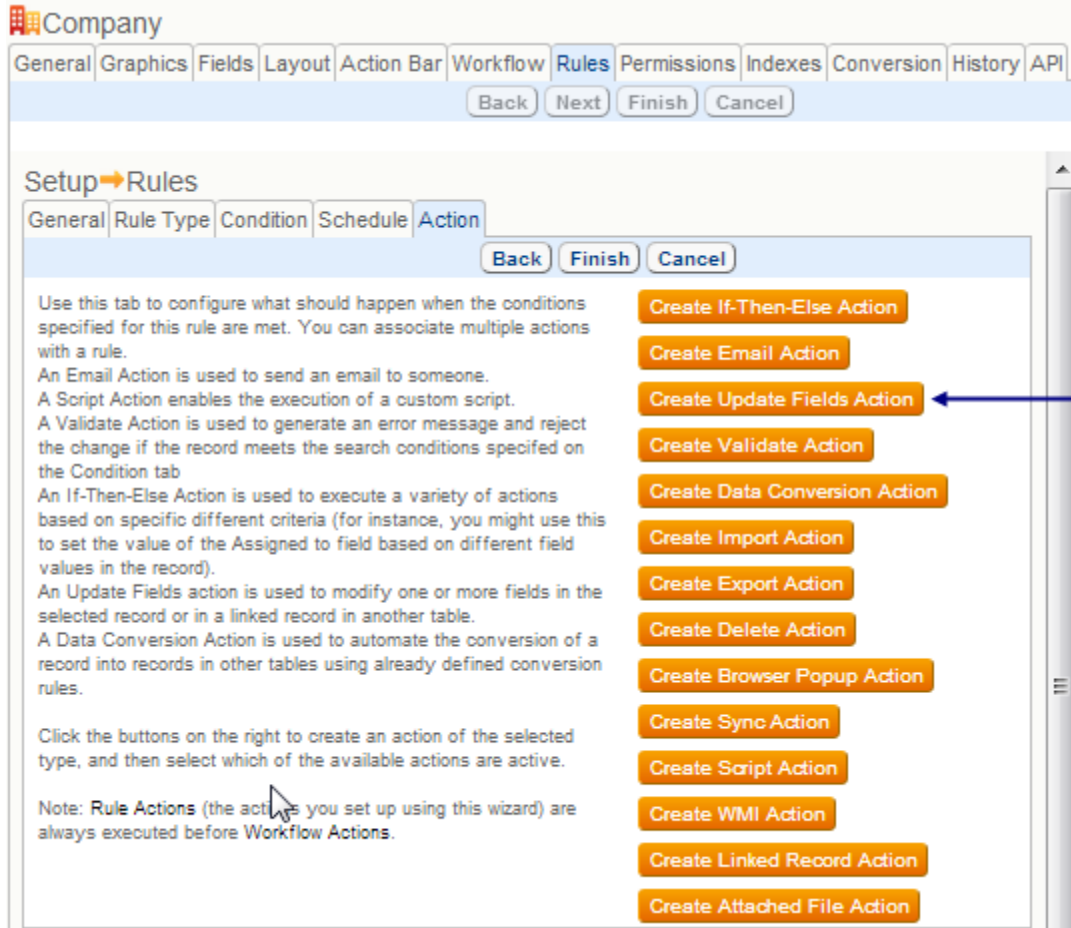
Back Next Finish Cancel

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

New | Delete | Enable | Disable

<input type="checkbox"/> Edit	Rule ID	Comment	Rule Type	Conditions	Name of Action(s)
<input type="checkbox"/>	6526	TB Demo Data Update: Update date fields by one month each month so reports have data	At selected time intervals.	Saved Search:Demo Records	Update Demo Dates
<input type="checkbox"/>	6492	Create/Edit: Copy Address to Billing	When a Company is created/edited/deleted.	Apply Rule: When a Company is Created/Edited	Update Billing from Address

After you are in the rule, go to the Action tab and click on the button for Create Update Fields Action.




You will need to create a name for your new action and then click Next. We are updating the linked fields for the latest contract information, so in our example we selected the Latest Contract ID as the field to update.

General **Fields** Values Errors

Back Next Cancel

Company Fields

To edit linked records select the linked field:

Please select the fields to be mass edited. 

Select All **Clear All**

The fields to modify

<input type="checkbox"/> Account Login	<input type="checkbox"/> Account Mgmt URL	<input type="checkbox"/> Account Number
<input type="checkbox"/> Account Password	<input type="checkbox"/> Account Rep	<input type="checkbox"/> Annual Revenue
<input type="checkbox"/> Attached Files(delete)	<input type="checkbox"/> Billing Address 1	<input type="checkbox"/> Billing Address 2
<input type="checkbox"/> Billing City	<input type="checkbox"/> Billing Country	<input type="checkbox"/> Billing same as shipping
<input type="checkbox"/> Billing US State	<input type="checkbox"/> Billing US State Long	<input type="checkbox"/> Billing Zip or Postal Code
<input type="checkbox"/> BUGGED: Related Product Names	<input type="checkbox"/> City	<input type="checkbox"/> Company Name
<input type="checkbox"/> Company Roles	<input type="checkbox"/> Company Table	<input type="checkbox"/> Company Website
<input type="checkbox"/> Contract Description	<input type="checkbox"/> Contract Info	<input type="checkbox"/> Contract is For
<input type="checkbox"/> Country	<input type="checkbox"/> Date Created	<input type="checkbox"/> Date Updated
<input type="checkbox"/> Deletable	<input type="checkbox"/> Demo Data	<input type="checkbox"/> Fax
<input type="checkbox"/> Industry	<input type="checkbox"/> Intl State or Province	<input type="checkbox"/> Latest Contract
<input type="checkbox"/> Latest Contract End Date	<input checked="" type="checkbox"/> Latest Contract ID	<input type="checkbox"/> Latest Contract Status
<input type="checkbox"/> Main Contact	<input type="checkbox"/> Ongoing Notes	<input type="checkbox"/> Original Start Date
<input type="checkbox"/> Parent Company	<input type="checkbox"/> Primary Role	<input type="checkbox"/> Primary Team
<input type="checkbox"/> Public Company	<input type="checkbox"/> Reference as of	<input type="checkbox"/> Region
<input type="checkbox"/> Rep Email	<input type="checkbox"/> Rep Phone	<input type="checkbox"/> State or Province

Click Next to go to the values tab. In the options for the values, you will find a radio button to enable the use of a search to define the value that is used. Clicking on that radio button will enable the drop down menu that contains your saved searches. At this point you can also create a new search which is what we did in our example.

General Fields **Values** Errors

Back Next Finish Cancel

You can enter new field values directly, or use the Formula radio button to set field values from existing fields

Please enter new values for these fields

Latest Contract ID

Treat the above as Standard text A formula Reimport source record matching current value and recreate links

Use the search to find more info

Edit Create New Search

If the field allows multiple values, then we will show the following two options indented beneath this:

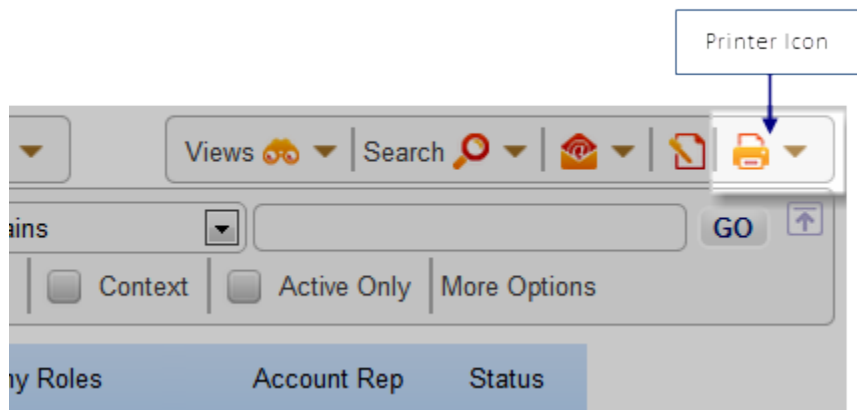
- Choose the first record returned by the search in sort order
- Choose all records that match the search criteria

Back Next Finish Cancel

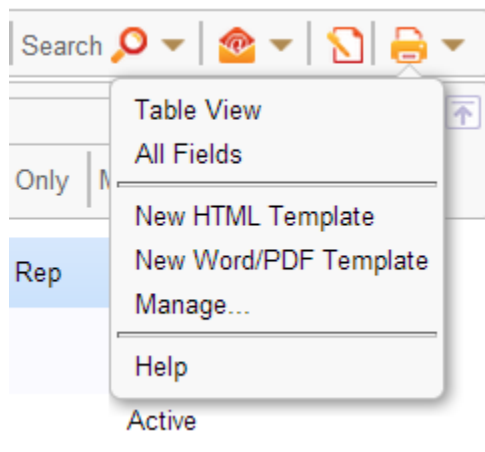
PRINTING FROM HTML/WORD/PDF TEMPLATES

Templates can be created in Word, PDF or HTML formats. Browser will either open the document or ask the user to save it, depending on browser settings. All standard operations are available: templates may be created, modified, and deleted. A Word template may be downloaded, corrected and uploaded.

Any staff user may select the preconfigured format by name to print a record.



Select a record or multiple records in a table view, mouse over the print icon and select the template.



CREATING NEW TEMPLATE

Mousing over the printer icon from the table view of records, you can select New to create a print format which may contain text with specific fields inserted wherever you like.

For HTML based print format:

The name of this format	*Name <input type="text"/>
Select the type of printing.	<input checked="" type="radio"/> Print whole page <input checked="" type="radio"/> Start each record on a new page <input type="radio"/> Put line separator only between records <input type="radio"/> Print labels Across <input type="text" value="2"/> <input type="text" value="Down"/> <input type="text" value="11"/>
Enter the size of the labels you will be printing or select a standard paper size and let the system calculate the label size.	<input checked="" type="radio"/> US Paper Size: <input type="text" value="Letter"/> <input type="radio"/> ISO <input type="text" value="A0"/> <input type="radio"/> Label size: Width <input type="text" value="2"/> <input type="text" value="inches"/> Height <input type="text" value="1"/> <input type="text" value="inches"/> <input checked="" type="checkbox"/> Add borders around labels
Adjust vertical and horizontal margins	Vertical Margin <input type="text" value="25"/> <input type="text" value="millimeters"/> Horizontal Margin <input type="text" value="45"/> <input type="text" value="millimeters"/>
If this option is selected, this print template will be accessible to all users who are allowed to view the table. Otherwise, it will only be visible to its creator.	Published <input checked="" type="radio"/> Yes <input type="radio"/> No
Add variable fields, such as Address, to the label.	<input type="button" value="Add Variables"/>
Add variable fields, displayed as barcode, to the label.	<input type="button" value="Add Barcode"/>

It is also possible to print records to named Word/PDF templates.

Finish
Cancel

Provide a name for the Word/PDF template.
This does not need to be the file name

***Title**

Choose what kind of output you would like to be created for this saved format.

Create format in:

MS Word 2007+
 PDF

Override header height inches ▼

Override footer height inches ▼

Click the button to get a list of all field variables that can be used in your Word template. Each variable should be inserted into your template with the format: \$fieldname. When printing in a MS Word format, use \$merge(\$fieldname) to merge the content of attached .html/.docx files from the field \$fieldname.

Show Field Variables
Download Template

Define whether the template is for printing a single record or for printing labels.

Printout will be:

one page per record
 label format

Across Down ▼

Add borders around labels

If the printout will include any embedded tables, the printout will use the view that is already defined for the display of the field in the record view. However, you can design your own table format and style in Word and choose whether or not to include the column headings from the EW View or just the data.

If you choose to include column headings, insert the table field variable into the upper left cell of the table in your template. If you choose to include just the data, insert the field variable into the first body cell of your table (the row below the heading row).

Table Column Headings

Include column headings in table
 Include only table data without headings
 Expand table to fit page width

Upload the Word document here when it is complete. Note that EW supports printing of files with the following extensions only: .docx

File

Choose File No file chosen Upload

No file uploaded

If this option is selected, this print template will be accessible to all users who are allowed to view the table. Otherwise, it will only be visible to its creator.

Published

Yes
 No

Finish
Cancel

Fields that you might want to include in your print format may be something complex like embedded related tables as well as ordinary fields.

INCLUDING EMBEDDED TABLES

There are two ways to include embedded tables:

1. Simple & Fast. Automatically generates table for user. Just place the formula with embedded field in a separate line:

```
$formula($related1231231231)
```

In this case we can generate table according to a defined view for this embedded table with automatically calculated cell width(like in a current view) - there's no need for user to draw table/define columns at all

2. Custom. Lets user to define table format, labels, columns to be displayed and cell width.

For example:

\$related1231231231

\$id	\$item_name	\$unit	\$price	\$quantity	\$total_price
------	-------------	--------	---------	------------	---------------

Table will be generated with specified columns, given cell width, rows of the same color and with/without headings (depending on table labels settings).

If format is set to include only table data without headings, format of the first row is used to generate odd data rows and second row is used to generate even data rows.

\$related1231231231

\$id	\$item_name	\$unit	\$price	\$quantity	\$total_price
\$id	\$item_name	\$unit	\$price	\$quantity	\$total_price

If format is set to include column headings in the table then format of the first row is used to generate table headers and format of the second row will define data rows.

\$related1231231231

\$id	\$item_name	\$unit	\$price	\$quantity	\$total_price
\$id	\$item_name	\$unit	\$price	\$quantity	\$total_price

Below is the example of the print format with the static header where settings to exclude/include table headings are completely ignored. Second row is used to define template for the odd data rows, third row defines even data rows.

\$related1231231231

ID	CUSTOM LABEL		PRICE	N	SUM
\$id	\$item_name	\$unit	\$price	\$quantity	\$total_price
\$id	\$item_name	\$unit	\$price	\$quantity	\$total_price

SURVEYS AND QUESTIONNAIRES

With the creation of the Survey and Questionnaire functionality, Customers can now create and manage complex surveys or questionnaires without needing admin level training or admin level privileges. Questions, question sets, answers, and surveys can be easily specified in an efficient manner. Instances of question sets and answers can be saved for future reference.

- Regular users can manage sets of an unlimited number of questions without creating fields for each question.
- Individual questions may be given an identifier so their answers can be searched and reported on and they may be displayed in table views.
- The same questions may be used in multiple question sets and may behave differently in each set (required in some, ordered differently, and so on).
- A set of questions and their answers may be stored in one table for reuse and pulled into another table based on some matching criteria.
- Sets of questions may be made visibility dependent on specific answers so you can build a decision tree structure if desired.
- Different kinds of surveys can easily be combined into a single survey table with a “survey type” field defining the question set to display.

DATA TYPES

The Survey and Questionnaire functionality requires three new data types that work in conjunction with each other:


- Question Description- allows users to define questions and answer types in a normal table record
- Survey Definition - allows users to select the questions for a particular questionnaire, define their order, visibility dependencies, and requirements
- Survey Presentation - stores an instance of a questionnaire or survey for a given record. It is used to define which survey definition applies, to display a set of questions defined by the selected survey definitions, and to store the combined questions and answers.

What follows is an explanation of each of these new data types and how they are used in their simplest manner.






See Also:

QUESTION TABLE AND DESCRIPTION FIELD

The first step in setting up Surveys and Questionnaires is to set up a table to hold all of the questions that will be defined. This new table will include the new "Question Description" field. You may also want to add a field for Question Name or some other identifying fields to help you find questions that are related to specific concepts. You may also want to have a field that describes the type of answer or the usage of the question, since the details of the question cannot be seen from the table view - in the table view you see the question itself, but not its potential answers. But such fields are not absolutely necessary. Below is an example of a table view of a Survey Questions table in which a "Name" field is defined to summarize the point of the question.

 **Survey Questions**

Status: 33 record(s) found, 1 pages. [Click here](#) to re-count records.

<input type="checkbox"/>	Edit	ID ↓	Name	Question Description	Su
<input type="checkbox"/>		38	Name Impression: AppLab	Using any criteria you choose, please give us your impression of...	F
<input type="checkbox"/>		37	Name Impression: Workflow Express	Using any criteria you choose, please give us your impression of...	F
<input type="checkbox"/>		36	Name Impression: AppSuite	Using any criteria you choose, please give us your impression of...	F
<input type="checkbox"/>		35	Name Impression: Agiloft	Using any criteria you choose, please give us your impression of...	F
<input type="checkbox"/>		34	Comfort Level: AppSuite	Would you feel more or less comfortable purchasing software...	F

There are no special options in the field wizard for this data type. It should be given a name like Question and assign the permissions for the field to those groups who should be able to create/edit questions, and then give those same people the right to create and edit records in the Questions table.

Creating a Question Record

Each question will be defined in a separate record in this table. The question description field will appear as an "Edit Question" button where this field is placed:

Clicking the 'Edit Question' button brings up the question wizard. This dialogue is similar to a field wizard, in that it allows the user to select the data type of the answer, the choices for the answer, and so on.

General Tab

On the General tab there are options for Question Text, Answer Type, and Name.

The screenshot shows a software interface for configuring a survey question. At the top, there are three tabs: 'General' (selected), 'Options', and 'Display'. Below the tabs are three buttons: 'Next', 'Finish', and 'Cancel'. The main area is divided into four sections:

- Enter question text:** A text input field containing the placeholder text '*Question Text'.
- Internal Notes:** A text area with a light gray background. To the left of the text area is a descriptive paragraph: "Internal notes allow you to specify that a question is used by more than one Survey or to provide other useful information to staff users when they are defining Surveys. They are not seen by the end users of the Survey."
- Select answer type:** A dropdown menu labeled 'Answer Type' with 'Text' selected.
- Name:** A text input field. To the left is a descriptive paragraph: "If you want to search among answers to this question, give it unique name that will be used in search dialog; otherwise leave this field empty. Note: Making too many questions searchable may slowdown the system."

At the bottom of the form, there are three buttons: 'Next', 'Finish', and 'Cancel'.

Question Text

The question text is the text that will actually be displayed when a user is presented with the survey.

Answer Type

The answer type allows you to select the type of answer from any of the standard data types: Choice, Currency, Date, Date/Time, Email, Floating Point, Integer, Multi-choice, Percentage, Short Text, Telephone/Fax, Text, and Time.

General Options Display

Next Finish Cancel

Enter question text *Question Text

Internal notes allow you to specify that a question is used by more than one Survey or to provide other useful information to staff users when they are defining Surveys. They are not seen by the end users of the Survey.
 Internal Notes

Select answer type Answer Type
 Short Text
 Choice
 Currency
 Date
 Date/Time
 EMail
 Floating Point
Integer
 Multi-Choice
 Percentage
 Short Text
 Telephone/Fax
 Text
 Time

If you want to search among answers to this question, give it unique name that will be used in search dialog; otherwise leave this field empty. Note: Making too many questions searchable may slowdown the system.

Next Finish Cancel

An answer type of Choice or Multi-Choice allows you to select/edit an existing choice list or create a new choice list.

Select answer type Answer Type
 Choice

You can populate this question with values from an existing choice list. If none of the available choice lists meet your needs, create a new set of values by clicking Create Choice List.
 Select choice list
 Excellent to Poor
 Create Choice List Edit Choice List

Note: The standard choice lists are not available for Question Description choice and multi-choice answer types. You are only allowed to use those lists that have been created through the Question Description wizard. Like standard choice fields, once a choice list has been created for a question it can be used in other questions. Once a question has been saved the answer type cannot be changed.

Note: Any user given permission to create and edit questions will have the ability to create and edit the choice lists for **any** questions. These should be trusted users, though they do not need to be administrators.

Name

The Name field is optional, but is necessary if you want to be able to reference/search/report on the values in this question's answers as a separate field wherever the surveys are presented. Once a question has been named, that name cannot be changed. If the Name is left blank initially, it can still be filled in later.

It is only advisable to name a question if you will want to 1) display its answer in a table view, 2) search on its answer, or 3) include its value in a report. The named questions create virtual fields that will appear on search dropdowns and elsewhere, which can be confusing if you do not actually plan to use them.

Options Tab

Depending on the answer type selected, different options will be displayed on the Options tab, similar to the options in a normal field wizard. Below are the options for a Choice Answer Type.

General Options Display

Back Next Finish Cancel

Define the type of input aid for this field. You can display each of the available values next to a radio button or arrange the values in a drop-down list. You can sort the values by their order in the choice list or alphabetically.

Display values as:

Radio buttons

Drop-down list

in as in choice list order

Back Next Finish Cancel

Note: The ability to make a question required or visibility conditional is not handled within the Question Description wizard - it is defined when creating a Survey Definition.

Display Tab

The Display tab has the standard options depending on the Answer Type. Individual questions may have input instructions, popup instructions and so on.

Once a question record has been created, it is possible to edit the question or view its options by clicking the Edit Question button again.

SURVEY DEFINITION TABLE AND FIELD

Once the questions table has been defined and some questions created, you are ready to combine them into question sets, which will be based on a Survey Definition field.

Typically it will make sense to store the question set definitions in another table called Survey Types or Question Sets or whatever makes sense for your usage. This table will include a record for each Survey Definition or Question Set that defines how the questions behave for that set.

The table should generally include one or more fields that can be matched against one or more fields in the table where you want the questions to appear, to auto-display the correct question set.

For instance, suppose you have named this table Question Sets and you want to ask a different set of questions in a support table depending on which product a user selects. You would have a field in the Question Sets table linked to your products (it may support one or more values, so a particular question set could be linked to multiple products). When a user creates a support case, they will select a product, and the Survey Presentation field in the support case table will be defined to match the product they select to this multivalue linked product field in the Question Set table to pull in the right survey definition value and therefore the correct question set record.

Or suppose you are using this to present a variety of surveys to customers. You would create a Survey Type table with a record for each survey/set of questions. You might have a choice field for survey type in this table as well.

To present a customer with a particular survey, you could simply give them a hyperlink (in an email or online) to click that sets the Survey type field to the desired value, thus pulling in the matching set of questions.

Setup of Survey Definitions Field

First create the table that will hold the survey definition records. Then create any fields you want to use to identify the survey or question set in another table and that can be used to pull in the right set of questions.

Then create a new field using the Survey Definition data type. On the General tab of the field wizard, choose the table where the Question Description field can be found and the specific Question Description field that the questions will be drawn from. The only options presented will be fields of the Question Description type.

The screenshot shows a configuration window with four tabs: 'General', 'Options', 'Permissions', and 'Display'. The 'General' tab is active. At the top right are buttons for 'Next', 'Finish', and 'Cancel'. The main area contains three sections:

- *Field label:** A text input field containing 'Survey Definition'. Below it is an orange button labeled 'Use HTML editor'.
- *Field name:** A text input field containing 'survey_definition'. To its left is explanatory text: 'This is the actual name of the field in the database. It may not contain spaces or non-alphanumeric characters. Users generally do not see this name.' Below this is another line of text: 'By default, the field name is constructed by replacing spaces in the field label with " _ " characters.'
- *Questions source:** A dropdown menu showing 'Survey Questions : Question Description'.

The other tabs have the standard field options.

Creating a Survey Definition Record

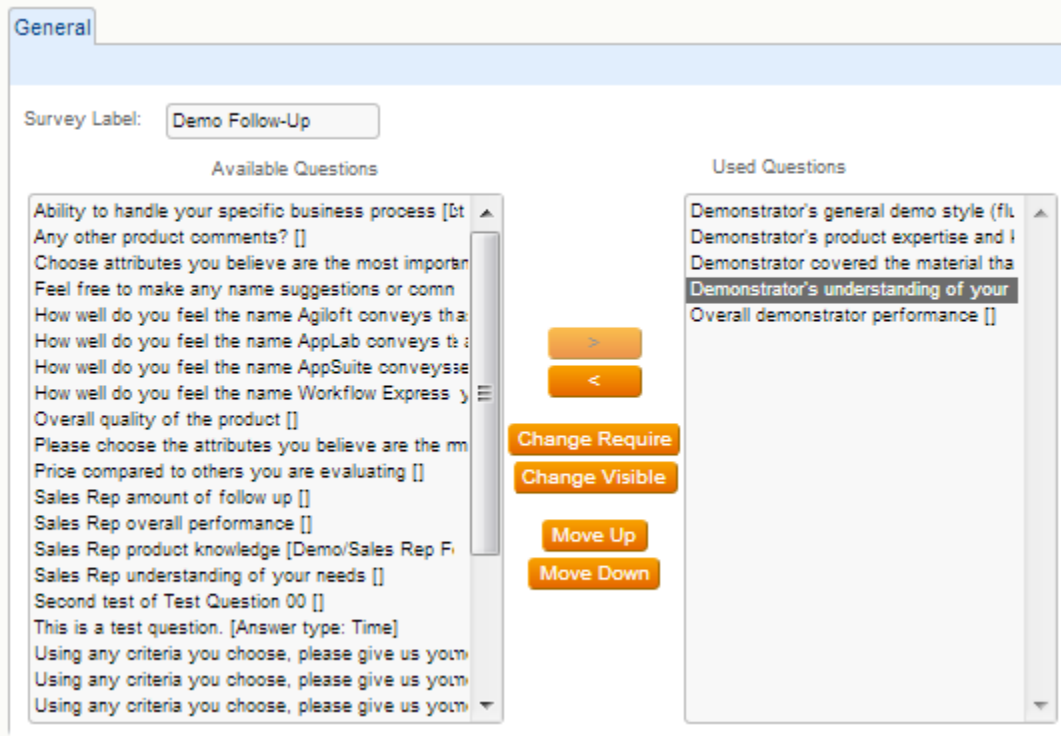
When creating a record in a table with this data type, the user sees a button 'Edit Survey' where this field is placed.

The screenshot shows a record form for 'Survey Definition'. At the top left is a table icon and the title 'Survey Definition'. To the right is a copy icon. Below the title are 'Save' and 'Cancel' buttons with dropdown arrows. A breadcrumb trail shows 'Survey Definition' and 'Communications' with navigation arrows. A print icon is on the far right. The main form area has a light grey background and contains:

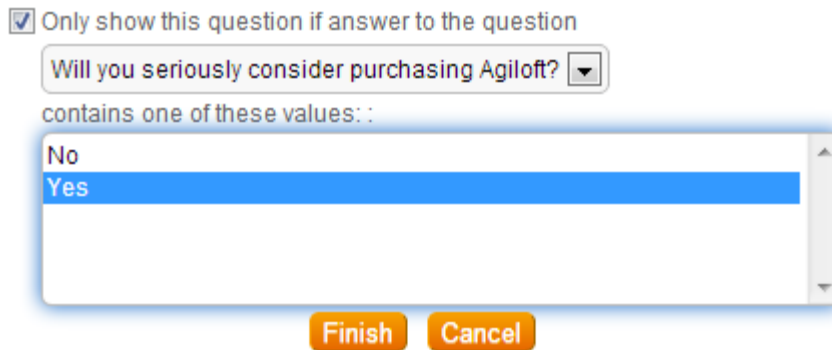
- ID:** 7
- Survey Type:** A dropdown menu with 'Demo Follow-Up' selected.
- Survey Definition:** An orange button labeled 'Edit Survey' is positioned to the right of the field label.

At the bottom are 'Save' and 'Cancel' buttons with dropdown arrows, and navigation arrows.

Clicking the button brings up the survey wizard. There is only one tab; it resembles the Available Actions / Selected Actions in rules and workflow. All questions defined by the table and field selected in the field wizard will be listed under Available Questions.



Note the options for Change Require and Change Visible between the sections. This is where requirement and conditional visibility are defined for any of the questions in the Used Question section.



Agiloft Reference Manual

Make the question required:

Yes

No

Conditionally, when the

Will you seriously consider purchasing Agiloft? ▼

contains one of these values: :

No

Yes

Finish

Cancel

Note that Conditional Requirements or Visibility can only be based on other used questions with a type of Choice or Multi-choice.

Once you have defined your question sets, you can see the questions listed above the Edit Survey button:

Survey Type ▼

Survey	Demo Follow-Up: Will you seriously consider purchasing Agiloft? Any other product comments? Demonstrator's general demo style (fluency, speed, clarity) Demonstrator covered the material that was most important to you Demonstrator's product expertise and knowledge Demonstrator's understanding of your needs Overall demonstrator performance What could the demonstrator have done to make the demo more effective for you?
--------	--

[Edit Survey](#)

When the survey questions are presented to a user, the questions will be displayed in a single column from top to bottom in the order shown here. It is not possible to

arrange the questions in multiple columns on the layout. However, it is possible to use two question sets and put each one in a separate column to give a two column display. Here is the result of the setup shown above:

Survey Type Demo Demo Follow-Up ▾

Reset

Will you seriously consider purchasing Agiloft? Yes No

Any other product comments?

Demonstrator's general demo style (fluency, speed, clarity) %

Demonstrator covered the material that was most important to you %

Demonstrator's product expertise and knowledge %

Demonstrator's understanding of your needs %

Overall demonstrator performance %

What could the demonstrator have done to make the demo more effective for you?

SURVEY PRESENTATION

Once the questions and survey definitions/question sets have been created, you are ready to start using the questionnaires. This will involve adding a field of the data type Survey Presentation to the table in which you want the questions to appear.

Field Wizard for Survey Presentation Field

General Tab

After specifying the field label, you must provide an Answer Column Prefix. This is used as a prefix for the names of the virtual, searchable fields for any questions that were given a name when they were defined.

General	Options	Permissions	Display
<input type="button" value="Next"/> <input type="button" value="Finish"/> <input type="button" value="Cancel"/>			
<p>This label is associated with the field and is seen by most users.</p>	<p>*Field label</p> <input type="text" value="SPR Questions"/> <p><input type="button" value="Use HTML editor"/></p>		
<p>This is the actual name of the field in the database. It may not contain spaces or non-alphanumeric characters. Users generally do not see this name.</p> <p>By default, the field name is constructed by replacing spaces in the field label with " _ " characters.</p>	<p>*Field name</p> <input type="text" value="SPR_Questions"/>		
<p>Enter up to 10 characters of prefix text that will appear in the field names of answers that can be searched.</p>	<p>*Answers Fields Prefix</p> <input type="text" value="Spr"/>		
<p>Use the Admin Notes field to put your comments about the purpose and details of the field's configuration and usage.</p> <p>These notes can then be included in the automated Field Documentation printout.</p>	<p>Admin notes</p> <div style="border: 1px solid #ccc; height: 100px; width: 100%;"></div>		
<p>If this checkbox is selected the Admin Notes shown above will be included in the automated Field Documentation printout.</p>	<p><input checked="" type="checkbox"/> Include in Field Documentation</p>		
<input type="button" value="Next"/> <input type="button" value="Finish"/> <input type="button" value="Cancel"/>			

For example: if the Answer Column Prefix is "Spr" and there is a question with the name "Satisfaction" a virtual, searchable field will be created called Spr_Satisfaction.

Options Tab

The Options tab is more detailed for this field type than for the other new types. Here you define a default survey to be used, conditional visibility, and whether a search pop-up should be added to the field.

Default Survey

This option controls which Survey will be used to form the set of questions.

Generally only one Survey should be matched, but if several are found, the first in the list will be used.

Default Survey

Select Table

Survey defined in

at field

which matches SS

Survey defined in of current record.

You have three choices for the default survey.

The first option is a fixed field reference. It will allow you to choose a specific survey definition. After selecting the survey definition table and the survey definition field in that table, you may search and select the survey you wish to display. This is the least common option.

Let's suppose you want to pull in a different set of survey questions based on the value in a Survey Type field in the current record. You can do this using either of the next two options.

With the second option, you find the correct set of questions using a matching saved search against the survey definition table. You have a choice field for Survey Type in both the current Survey table and the table holding the survey definition field. Then you create a Saved Search on the survey definition table. This search uses a variable to match the Survey Type value in the survey definition table against the variable Survey Type field in the current Survey table. It pulls in the matching set of questions for that Survey Type.

With the third option, you don't create a Survey Type field in the current Survey table, you just create a linked set of fields from the table containing the survey definition field that includes both the survey definition field and the Survey Type field from that table. The user selects the Survey type in the same way, which is really a link to the survey definition table, and that pulls in the definition field, and the survey presentation field then populates the questions matching that definition.

From an end user point of view, these two options are interchangeable and provide the same result.

Note: A default behavior must be defined for the field and once the field is saved this is not changeable.

Visibility

This is the standard visibility option, allowing you to determine the visibility of a field based on a choice or multi-choice value.

Pop-up Search

The purpose of the pop-up search option is to enable a user to pull in an already saved set of answers to a specific set of questions, effectively to reuse the answers. This was developed for the original use case where customers were providing environment details in a service request and wanted to store these answers to a saved Environment record to be able to reuse them.

For this kind of situation, the initial set of answers need to be saved into a new record, which can easily be done with an action button running a conversion action. The complete set of survey answers for a question set can be saved in a new record using a standard conversion action – the single Survey Presentation field can be mapped to a Survey Presentation field in another table and all question/answers will populate there.

Given this kind of context, adding the search pop-up lets you choose another table that has a survey presentation field in it and choose the particular survey presentation field, if there is more than one in that table. When used, it will only display records in that table that include the same survey definition as the current record from which the lookup is run. You can also set a view to be used for the lookup of the available records.

Use this option to add a search pop-up to the field. This allows users with *create/edit* privileges to search through a filtered set of existing Surveys for content to be imported into this field. For instance, you might use this pop-up with the Solution field so that staff members can find existing answers and populate the Solution field of a new Case with an existing solution. This is an alternative to the Standard Solutions functionality.

Add search pop-up to this field

Do not add search pop-up

Add search pop-up to this field

Search table

where field matches current one.

Use to display items for import

Execute Actions

No items to drag

Add Action

Remove

Display mismatch warning to user

Selecting the popup option adds two buttons above the question set called “Load from [table name]” and “Reset”. Clicking the Load.. button brings up a normal linked field lookup to find the matching records (those that use the same survey definition).

Survey Type Demo Demo Follow-Up ▾

Load from Surveys...
Reset

Ability to handle your specific business process %

Price compared to others you are evaluating %

Overall quality of the product %

A record can be selected for import:

Import/Replace
Cancel

Surveys

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

	Edit View	ID ↓	Survey Type	Submitter Name	Submitter Company	Timeliness of Support	Overall Quality Value	Overall Satisfaction Value
<input type="radio"/>		13	Support Case	Fern Wood	Agiloft	Very Good	8.00	10.00
<input type="radio"/>		10	Support Case	John Suzuki	IBM	Excellent	5.00	5.00
<input type="radio"/>		4	Support Case	Fern Wood	Agiloft	Excellent	5.00	8.00

And when imported, the question set is refreshed with any answers. Note that if the question set itself has changed in the meantime (suppose new answers were added), the latest questions will be shown.

Under the search pop-up options you can also specify another field to be populated with other information from the imported record. This allows you to pull in the ID of the record in the other table, for instance, along with any other linked fields that you might want from that table.

You may also specify that the user receive a warning if they import a record with a different list of questions from the current definition.

Creating a Survey Record

The survey presentation field displays the questions used in its definition with the appropriate answer fields.

Survey Type Demo Follow-Up

Load from Surveys... **Reset**

Ability to handle your specific business process %

Price compared to others you are evaluating %

Overall quality of the product %

Will you seriously consider purchasing Agiloft? Yes No

Any other product comments?

Demonstrator's general demo style (fluency, speed, clarity) %

Demonstrator covered the material that was most important to you %

Demonstrator's product expertise and knowledge %

Demonstrator's understanding of your needs %

Overall demonstrator performance %

The Reset button will restore the original set of questions (from before the saved record was imported) but leave the answers for any matching questions.

If any questions in a particular survey presentation are named (see discussion of the Name field in section Creating a Question) the virtual fields will be filled in with the appropriate values so that they may be used in a search.

Search: Timeliness of Support equals Excellent **GO**

Show All Refine String Context

Edit View ID ↓ Submitter Name Submitter

- Excellent
- Very Good**
- Good
- Fair
- Poor

Named questions not in a particular survey will have blank values for that record. Note that the survey presentation field itself is not searchable but the entire survey (questions and answers) can be copied into a text field to allow text searching on it.

SURVEY AND QUESTIONNAIRE EXAMPLE SETUP

Provided below is the field setup and layouts for three tables using the new data types for a very straightforward Helpdesk and Support case survey.

Survey Questions

This table has a field of Question Description type. It also has a text field for Question Name. This text field is just for ease of reference elsewhere in the system. It is not actually necessary.

Field Label	Data Type
Answer Details	Text
Created By Creator Login, Creator Team	Link to selected fields from Allow multiple values:No Allow entries not in source
Date Created	Date/Time
Date Updated	Date/Time
History	History
ID	Auto-Increment
Question	Question Description Field
Question Name	Short Text
Survey Types	Related Table : Survey Ty
Updated By, Updater Login, Updater Team	Link to selected fields from Allow multiple values:No Allow entries not in source

<input type="checkbox"/> Edit	ID ↓	Name	Question Description	Answer Details
<input type="checkbox"/>	42	Timeliness	Timeliness of Support	Choices: Excellent, Good, Fair, Poor, Very Poor
<input type="checkbox"/>	41	Rate Quality	Rate the overall...	Choices: Excellent, Good, Fair, Poor, Very Poor
<input type="checkbox"/>	40	Rate Satisfaction	Rate your overall...	Choices: Satisfied Neutral, Dissatisfied
<input type="checkbox"/>	39	Suggestions	Please provide anv...	Text: Max size: 5000; Display size: 90x10

Survey Questions



ID 40

Question Description

Rate your overall satisfaction with the support experience

Answer Type

Choice

Answer Details

Choices: Satisfied Neutral, Dissatisfied

Name

Rate Satisfaction



Survey Types

This table has a field of Survey Definition type. It also has a text field for Survey Name. This is so that the surveys can be referenced by name elsewhere in the system.

Field Label	Data Type
Created By Creator Login Creator Team	Link to selected fields from Allow multiple values:No Allow entries not in source
Date Created	Date/Time
Date Updated	Date/Time
History	History
ID	Auto-Increment

Show Extra Fields	Choice
Survey Definition	Survey Definition Field
Survey Name	Short Text
Updated By Updater Login Updater Team	Link to selected fields from Allow multiple values:No Allow entries not in source

Survey Definitions

Status: Survey Definition #5 successfully created. 4 record(s) found, 1 pages. [Click here](#) records.

<input type="button" value="New"/> <input type="button" value="Delete"/> <input type="button" value="Actions"/>		<input type="button" value="Views"/> <input type="button" value="Search"/> <input type="button" value=""/> <input type="button" value=""/>		
<input type="checkbox"/>	Edit View	ID ↓	Survey Type	Survey Definition
<input type="checkbox"/>		5	Helpdesk Case	Helpdesk Case: Rate your overall satisfaction with the support experience Rate the overall quality of the support you received Please provide any suggestions ...
<input type="checkbox"/>		4	Support Case	Support Case: Feel free to make any name suggestions or comments. Rate the overall quality of the support you received Rate your overall satisfaction...

Survey Definition

ID 5

Survey Type

Survey Definition

Helpdesk Case:
 Rate your overall satisfaction with the support experience
 Rate the overall quality of the support you received
 Please provide any suggestions
 Timeliness of Support
 Ability to handle your specific business process

Surveys

This is the main table where the surveys for both the Helpdesk and Support Case tables are collected. A survey record is created via a hotlink in an email template in the Helpdesk and Support Case tables. Because the survey is created using a Guest account, a rule will run on the creation of the survey to update the Submitter Name based on information in the linked Helpdesk or Support Case record.

Field Label	Data Type
Communications	Communications Search F
Created By Creator Login Creator Team	Link to selected fields from Allow multiple values:No Allow entries not in sourc
Date Created	Date/Time
Date Updated	Date/Time
Definition ID Show Extra Fields Survey Definition Survey Type	Link to selected fields from Allow multiple values:No Allow entries not in sourc
HD Assigned Team/Person,HD Assigned Person,HD Submitter Phone,Helpdesk Case ID,HD Submitter Name, HD Priority,HD Summary,HD Type of Problem	Link to selected fields from Allow multiple values:No Allow entries not in sourc
History	History
ID	Auto-Increment
Linked Company	Link to selected fields from Allow multiple values:No Allow entries not in sourc
SC Assigned Person,SC Assigned Team,SC Customer Name,SC Customer Phone,SC Date Closed,Support Case ID,SC Priority,SC Summary,SC User Company,SC Assigned Team/Person	Link to selected fields from Allow multiple values:No Allow entries not in sourc
Submitter Company,Submitter Phone,Submitter Email,Submitter Name,Submitter Team	Link to selected fields from Allow multiple values:No Allow entries not in sourc
Survey Questions	Survey Presentation Field

Updated By Updater Login Updater Team	Link to selected fields from Allow multiple values:No Allow entries not in source
---	---

Survey Tab in Staff Interface

Common Area *	New Survey	Questions	Numbers	Related Records	Emails
Survey Type					
Survey Definition					

Note: EUI Interface will consist solely of this tab

Details Tab in Staff Interface

Common Area *	New Survey	Questions	Numbers	Details	Emails
Submitter Information					
Submitter Email		Submitter Phone			
Submitter Team		Linked Company			
Support Case Information					
Support Case ID		SC Priority			
SC Customer Name		SC Date Closed			
SC Assigned Team		SC Assigned Person			
SC Summary					
Helpdesk Case Information					
Helpdesk Case ID		HD Assigned Person			
HD Submitter Name		HD Priority			
HD Type of Problem					
HD Summary					

Email Template

A survey email template was created in both the Helpdesk and Support Case tables. This email could be sent manually or based on a rule.

Message Heading

Next Finish Cancel

Insert Format

*From address Inbound email account for Support Case example@example.com

*Reply to Inbound email account for Support Case example@example.com

*To

Users: Teams: User Fields: Email Fields: Customer Email: Address in Field: Link to Team:

Addresses:

Subject: We welcome your feedback on our support service **Formula Help**

Include encrypted ID information in subject line to enable replies to update correct record

Enable field parsing for any reply to this email

Include Hidden Code for Truncating replies

Marketing Options Tracking on for Email Marketing Campaign

Message to import

HTML text

Context #0

Fonts A B I U

Dear \$formula(\$end_user_name),

Thanks for using our support system. We would appreciate hearing your feedback so that we may improve our service.

[Click here to submit a survey](#)

[Note, this link will need to be edited to replace the KB name, the URL, and any other information that changes when you create your own KB]

HTML Source HTML

Attachments: No Files Attached Fields Included: None

Next Finish Cancel

A hotlink is created using the following elements:

- Server Name: <https://sc5.saas.enterprisewizard.com/gui2/login.jsp?KeyID=0>
- Project/KB: Project=Demo Master Copy

Agiloft Reference Manual

- Guest User Account: user=guest
- Guest User Password: password=Guest22x
- Table and State: State=New:survey (uses logical table name)
- GUI Used: GUI=No/EUI (will display the EUI form page only, no other menu items)
- Survey Type field: Field=survey_type0: Support Case (determines which Survey to use)
- Support/Helpdesk Case ID field: Field=support_case_id:\$id (links the Support or Helpdesk Case to the Survey record)
- Exit URL: ExitURL= <https://www.agiloft.com/surveythanks.htm>

Source HTML

```
<p>Dear $formula($Send_user_name),</p><p>Thanks for using our support system.&nbsp; We would appreciate hearing your feedback so that we may improve our service.</p><p><a href="http://server7.enterprisewizard.com/gui2/login.jsp?KeyID=0&Project=New%20Demo&user=guest&password=guestuser&State=New:survey&GUI=No/EUI&Field=survey_type:Support%20Case&Field=support_case_id:$id&ExitURL=https://www.enterprisewizard.com/surveythanks.htm" target="_blank">Click here to submit a survey</a></p><p>[Note, this link will need to be edited to replace the KB name, the URL, and any other information that changes when you create your own KB]<br></p>
```

FAQ

GENERAL

How can I reset the password for the admin user?

It is now possible to reset the admin password for a KB from the Admin console. If you can login to the admin console, then go to Setup/Knowledgebases and click the Edit icon next to the knowledgebase for which you need to reset the password. Then select the button to Reset Admin Password and enter a new password. If you have lost the password for the admin console, you will need to contact our technical support for further instructions.

How to display support calls in different colours? Can you tell me how we would set things up so that support calls from high priority customers of ours would be displayed in a different colour? We also want to display calls that are late in different colours.

You can do it via the view settings. Select your table tab, select view to change and press 'Set View/Edit' on the table menu. In the edit view dialog select the 'Order' tab and press the 'Set row Coloring' button. Select a 'choice' field whose values will serve as a trigger to set the display attributes of any particular record in the table. (Priority field for example). Next Select desired font and color for the priorities. Finish

How do I configure logs to save space? We would like to either limit the size of logs or notify the administrator if it goes beyond the specified size.

It is possible to influence the behavior related to wrapper.log via \$EW_DIR/jboss/conf/wrapper.conf

```
# Maximum size that the log file will be allowed to grow to before
# the log is rolled. Size is specified in bytes. The default value
# of 0, disables log rolling. May abbreviate with the 'k' (kb) or
# 'm' (mb) suffix. For example: 10m = 10 megabytes.
wrapper.logfile.maxsize=1m
# Maximum number of rolled log files which will be allowed before old
# files are deleted. The default value of 0 implies no limit.
wrapper.logfile.maxfiles=5
```

The default policy for logs in \$EW_DIR/jboss/server/sw/log/* is to zip files older than 2 days, delete mail logs older than 10 days and other logs older than 7 days.

We would like to set the time that should pass before user sessions are automatically timed out, and we would like to warn the users before timing them out. Is this possible?

By default a user is disconnected after 1 hour of inactivity.

You can set the timeout interval by using a global variable and you can also set the time period for how far in advance a user should be warned about the timeout.

To set the timeout period:

1. Go to Setup/System/Manage Global Variables
2. Click New variable to create one of type Integer constant
3. Give it a label and use the name: disconnect_timeout
4. Supply the number of minutes of inactivity after which you want users to be logged out
5. Click Finish to save.

To set the time when users should be warned about an upcoming timeout:

1. Go to Setup/System/Manage Global Variables
2. Click New variable to create one of type Integer constant
3. Give it a label and use the name: warning_timeout
4. Supply the number of minutes of inactivity after which you want users to be warned (typically this might be 5 or 10 less than the disconnect period). If it contains "0", no warning is shown.
5. Click Finish to save.

When the warning period arrives, users will see a popup warning telling them they will be logged out in x minutes and to hit OK to stay logged in. This will reset the clock for them.

We'd like to use port 80 instead of 8080. We have some clients and partners who have limited outbound firewall access and they cannot get to our Helpdesk on port 8080. Is there a way to set up the app server to use port 80 or to alternatively use Apache as a proxy on port 80 for the app server traffic. Is this possible?

Both options are possible.

You can do the following:

- re-install the Agiloft , during installation it is possible to either agree to integration with the existing front-end web server or decline the integration and specify a port where Agiloft should listen itself. OR
- Stop Agiloft process, do a manual change in several files and start it again.

1. Change the port from 8080 to 80 in the following files:

```
/etc/EnterpriseWizardConf.xml
/etc/init.d/ew-server
$EW_DIR/jboss/server/sw/deploy/jbossweb-tomcat55.sar/server.xml
$EW_DIR/resources/java/build/local.properties
```

2. Disable integration in /etc/EnterpriseWizardConf.xml

```
<enableIntegration>false</enableIntegration>
```

3. Disable integration in \$EW_DIR/resources/java/build/local.properties

```
webserver.integration=false
```

4. Remove all lines that reference apache2 from
\$EW_DIR/resources/java/build/local.properties

These changes should survive further updates of Agiloft

Where do we put the customised login pages?

You have to place them in your Apache home directory.

For example:

Web pages are served from the directory as configured by the DocumentRoot directive. The default directory location is:

- Red Hat 7.x-9, Fedora Core, Red Hat Enterprise 4, CentOS 4: /var/www/html/
- Red Hat 6.x and older: /home/httpd/html/
- Suse 9.x: /srv/www/htdocs/

How do I change the title in the browser? Our KB is currently showing the KB Label, followed by "Success.... Guaranteed" for the browser title of every page. How do I change the global title?

Go to Setup/System/Manage Global Variables. Create a new global variable of type short text constant called staff_title and use whatever text you like as the value. Create another global variable of type short text constant called enduser_title and use your preferred text there. You will need to logout and login again to see your changes.

Can we use our own html template? Is there a way to apply our own html template, specifically for wrapping Agiloft with our own look. I'm not just talking about colors and fonts, but rather a header, footer, and navigation menu that appears on all of our support pages.

The appearance of the End User and Staff interfaces can be customized to match the outlook of your website. To customize the End-User interface, click Setup/Look&Feel/Manage End-User Schemes. Choose the Schema you want to change and press Edit. To customize the Staff interface, click Setup/Look&Feel/Manage Staff Schemes Choose the Schema you want to change and press Edit. In particular, the page header, footer and logo can be customized on Content tab of the wizard.

Can I run Agiloft on VMWare? If it can, is it supported through maintenance? Is there documentation for installation? Would there be a recommended configuration?

We have tested Agiloft installation on a VMWare for Windows 32 and 64 bit. It works fine. There is no special Agiloft configuration required. Common installation documentation applies.

How do I disable the chat functionality?

To disable the chat functionality do the following: go to Setup / Chat / Delete chat session settings.

To disable chat only for some users you can change the chat settings: Setup / Chat / Edit settings record / Access tab / Define the end user groups that can create chat requests for this session type

Can I link to a view/page in our KB without login? I have public information in our KB that I would like to link to and display from our corporate site without requiring to log in. What would be the best way to do that?

You can always create an End-User guest account with minimal permissions and hard-code the guest credentials into a hotlink that links to the data you need. I suppose State=Edit, record_access=view, gui=no is what you are after. Permissions for the guest user group will define the content of the record being shown.

Please review Online Help on how to create hotlinks:

[Hyperlink Reference](#)

Another option would be to use the REST API to retrieve data in JavaScript and display only the relevant data. Please review Online Help on REST operations, and specifically on Read operation:

[REST Interface Overview](#) [REST Read](#)

Starting point - what can we delete?

1. When creating a new KB from the ground up, may I create a full set of new tables, or shall I keep at least the original CONTACTS table?

You must keep the Contacts table tree, the default admin user, his group and his primary team or you will not be able to login to the new KB.

2. When customizing the CONTACTS table, what may not be changed? (Sub-tables, fields, ...)

We recommend never deleting the Employee sub-table as it is hard coded for use with LDAP and MS Active Directory authentication - if LDAP is set up, users are mapped into the Employee sub-table and that cannot be changed. The admin user is also held in the employee sub-table.

Before deleting any sub-table of contacts, it is wise to review any linked fields in other tables that you are keeping to see what contact table they may point to. For instance, the Assigned to field points to the Employee sub-table and if you delete the table, you will be deleting or breaking the linked fields.

Never delete the following fields: login, password, primary team, teams, groups. If you do not want to use some of them hide them by removing from the layout. If you delete any one of those fields you will likely completely break all access to the KB.

Keep in mind when deleting fields from any table that you may break searches that depend on those fields, reports that depend on those searches, and linked fields in other tables that use those as the source fields.

Generally when deleting a field, the system's integrity manager will warn you about dependencies on that field, with the important exception that it does not warn about the use of that field in if-then-else rule searches and in saved searches.

So suppose that you have a group permission filter to show end users other people's records, but only records submitted by people whose "Company" field matches the company field in the ticket - if you delete the Company field from the ticket or the contact table, you will have broken that group's record level permissions and you may not be warned about this. Eventually saved searches will be included in the integrity manager as well.

3. Besides the CONTACTS table, are there any other components of the standard KB that should not be changed when customizing a new KB?

Agiloft Reference Manual

Never change the name of the admin group. It is special and changing its name could break it.

Do not delete the primary team of the admin user. Note that if you delete a team that is the primary team for any user, you prevent them from accessing the system until you give them another value in the primary team field.

Do not delete the admin user or the anonymous user - they are both essential to the functioning of the system. The anonymous user is used in email hyperlinks when sent to an unknown email address and should be put in an appropriate group to control the permissions in that situation.

Be careful about deleting and modifying the default saved searches if you are using any of the default automation, as they may be used as the basis for a rule action and you may break that action by changing or deleting the search.

Do not delete the Calendar table if you plan to use any of the built-in calendar functionality.

Do not delete the Billing table if you plan to use any of the built-in billable timekeeping functionality.

Do not delete the EUI Templates table if you plan to use the alternative end user portal/interface.

We recommend starting from our default KB with many tables already set up and simply hiding all tables you do not wish to use. This is easily done by going to Setup/Tables/choose the table/Edit/Permissions and checking the box to hide the table from all users. If you later decide to use the table you can easily turn it on by editing the admin group's permissions for the table.

How can I disable or hide teams? I don't want to delete them. In the Assigned Field I don't want to see all of the teams -- in other places as well.

There is no disabling functionality for teams.

If you wish to remove the team from the linked field list you can edit that field and enter a restriction for it to hide the team from list based on the saved search.

For example:

Setup / Tables / Edit Support Case table / Fields tab / Edit Assigned To field / Options tab / Filter by a saved search / Filter Team by saved search: TeamName != 'Staff Team'

System Requirements for Agiloft Installation.

1. What is the minimum RAM for running ew application? 2 GB? 4 GB?

Minimum RAM is 4G. Agiloft can be installed on 2G but it is not enough for a production server.

2. How many resources of a server are to be used for each user? I need this information to estimate my server (RAM and Storage)

Resources depend on the number of concurrent users / tables / fields / rules / documents per day etc. For example Agiloft with 10000 users goes well on a server with 16G RAM and 500G HDD space. Usual OS used is Linux x64.

3. What is the typical server used for ew application for some partner or customer? If you know the model and brand that would be good to know as well.

Currently we are using servers equivalent to <http://www.supermicro.com/products/system/2U/6025/SYS-6025B-TR+V.cfm> with 16GB RAM, an 8 port 3ware 9560SX RAID card, and 6 500GB drives in a RAID10 configuration. with 2 Intel 5355 Xeons.

Need to pass data from our internal system to Agiloft

Customers are looking to be able to import a file and update a field value in a particular record automatically every day. The field value they need is stored in another system in their organization. Is there a way to accomplish this using a rule with a script action, an action button with a script, a scheduled import action, or some combination?

For example, say the Agiloft Contact table has a "Current Value in the Other System" field. They want that field to be automatically updated 1:00 AM every day based on the value the "Other System" shows for each corresponding contact record.

A scheduled import may be the way to accomplish this. Particular fields in particular records can be updated in this way by setting up a rule to run periodically importing an ASCII file with the appropriate fields and values from an accessible URL. The ASCII file would contain the ID or login (if a contact record) of the record you want to update in one column and the field value to be updated in another column. It can be done for one record or multiple records.

Script based approach mentioned above is also possible in the pull scenario where Agiloft initiates the update, though it would require programming. The action button approach would require a user to press the button at 1:00 AM. Another way to push data to Agiloft is via REST or SOAP interfaces.

Which release of JBoss can I use with Agiloft ?

I have two questions about JBoss Application Server:

1. Is it possible to use JBoss already installed on a server?

Yes, you can use it.

2. Which release of JBoss can I Use? 4.2.1.GA? 4.0.5.GA?

We support 4.2.2.GA at the moment.

Can we extend the session time out parameter for the end-user?

We have two system variables: 'disconnect timeout' and 'warning timeout'. The first variable is user auto-logout interval in minutes, the second variable is warning interval to confirm user presence.

You can check them in your KB: Setup/System/Manage Global Variables

To increase EUI session timeout you have to increase the default tomcat session timeout.

Configuration file is placed on \$EW_DIR/jboss/server/sw/deploy/jbossweb-tomcat55.sar/conf/web.xml

```
<!-- ===== Default Session Configuration ===== -->
<session-config>
<session-timeout>30</session-timeout> <!-- 30 minutes -->
</session-config>
```

You have to restart JBoss daemon (/etc/ew-server) after it.

Is there a way to read Agiloft data from another application?

Agiloft has a set of different interfaces to communicate with outside applications. There are REST interface, SOAP interface, Custom script functionality and export/import. Read more about it here:

[REST_interface_overview.jspx.html](#)
[SOAP_Interface.jspx.html](#)
[Using_Custom_Scripts.jspx.html](#)
[Importing_Data.jspx.html](#)

Is there a way to hide the User and Language data in the End User Interface? I'd like to hide the Logout link if possible as well.

To remove User name and Language drop down, go to Setup/Look&Feel/Manage End-User Themes/Your Theme/Edit/Content edit the HTML text (where we insert the logo) and add(!) the following to the HTML text:

```
<style type="text/css">
.SIStatusLabel {
visibility: hidden
}
```

```
.SIStatusValue {
visibility: hidden
}
.SIStatusLabel a {
visibility: visible;
position: relative;
}
</style>
```

If you want to remove the default logout as well remove the last line i.e.:

```
<style type="text/css">
.SIStatusLabel {
visibility: hidden
}
.SIStatusValue {
visibility: hidden
}
</style>
```

You can insert the logout block anywhere in Header, Footer or Home page content as:

```
<a href="javascript:logout()" title="alt+X">Logout</a>
```

with other formatting around if required.

Is it possible to print my records directly as Word or PDF document?

We now support the creation of Word templates (using MS Word 2007 and above) that can be used to print out a record in a nicely formatted word document or pdf file. The printout can include embedded tables, and there is even a new rule action type to automatically create the printout and put it in an attached file field in the record, from which it can be emailed to someone!

To get started, navigate to the table whose records you want to print. Suppose you want to print out price quotes, for instance, and you have a table for quotes. Once the table is on the screen, mouse over the printer icon in the upper right of the action bar and choose New Word/PDF Template. On the dialogue, choose whether you want the output to be to Word or PDF, then click the button to either show the field variables or to download the template, which will open Word with a template file with all field variables populated for you to move around and select. You can add graphics and your own formatting, and lay out the fields and their labels wherever you want them. Once you have the layout the way you want it, just upload the file back into the dialogue and save it. Then test it by selecting a record's radio

button and mousing over the printer to print it out to Word. You may need to make some adjustments to the template file once you see actual data and how it fits. Just modify and upload again.

To automatically create a PDF or Word document from fields in a record and upload it to a file field, first be sure you have a file field to use and that you have created the print template as described above. Then create a rule that will be triggered by an appropriate condition (certain fields populated, certain status, or whatever makes sense). Choose the new action type called "Create Attached File Action" and fill out the values. Voila! The file is created and attached.

How to report bugs properly?

Some users are delighted by the speed with which we resolve any bugs, while others are frustrated. The difference between the two groups is often the manner in which they submit tickets.

By far, the most effective technique for reducing the response time is to describe exactly how to reproduce the bug, with one step per line. For example:

Login as User/Password to KB_Name using Browser_Name

Step 1

Step 2

...

Result:

For example:

Login as user:john password:xd93Ks to KB:SrtNow using FF 3.5 Select Opportunities/Charts Run report - Opportunitites by Stage Excel Result: Error message: Invalid parameter x

The key points are that the bug report says what login/password/ browser to use in order to reproduce the bug and provides exact steps, with one step per line.

The last point of specifying one step per line may seem pedantic, but it turns out to be critical because it forces clarity. Bug reports that use paragraphs usually omit critical details about how to reproduce the problem. Another common error is to describe what has gone wrong or what you are seeing, rather than how to reproduce it.

Remember: Before a bug is assigned to Engineering to fix, support staff usually have to reproduce it and if you make this easy for them, the problem will get addressed a lot sooner. If a bug cannot be reproduced, then not only will the fix be delayed, but it cannot be properly tested.

Sometimes you may not be able reproduce the bug and in this case, all you can do is to describe all the symptoms, hopefully with an error report from the system, and

let our support people try to reproduce it. Naturally, we do our best in these circumstances, but it does delay things.

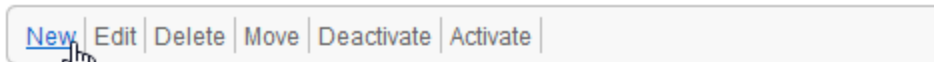
TABLES

How do I add a new table? I would like to divide my support cases into different types and show different fields for each type. Should I add a new sub-table for this?


You can accomplish this task in two different ways. If the only difference between these types of support case is that you want users to fill out different fields, you can use a choice field for the type of case and then dependent fields that only appear based on the particular type chosen. This is how the demo KB is set up, because it is simpler to maintain. The type of problem field has some dependent fields, like "Steps to Reproduce" if the type is a bug, and so on.

If you want to have different layouts for each type of case (a different data entry form in which the fields are arranged in different order for each sub-type), and you want different users to only be able to see certain sub-types at all, then you may want to create sub-tables of the Account table. To do this, go to Setup/Tables, select the Account table, and click New as in the screenshot:

Setup → Tables



To create a new table, select the top of the Table Tree if it is not already selected and click New.

Agiloft tables are arranged in a hierarchy so that sub-tables created from an existing table inherit the properties of their parent. 

To create a sub-table, select the parent whose properties will be used as a template for the new table and click **New**. For example, you can select the Contact/User/Employee table and click New to add a Manager table. Managers will automatically inherit the Employee fields in addition to any custom fields that you specify. To edit an existing table, select it and click **Edit**. To move a sub-table to a different position in the hierarchy, click **Move**.

To hide an existing table from all users, select it and click **Deactivate**.

To turn the toolbar tab on for the admin group, select it and click **Activate**.



The new table will inherit all of the fields from the parent table, and you can then add your additional fields, modify the layout, and so on.

Can I hide the table from all users?

You can hide a table from all user groups with a single click. As an administrator, go to Setup > Tables, select the table you wish to hide, and click the Permissions tab. To hide the table from all groups, select the first option "Hide the table from all users".

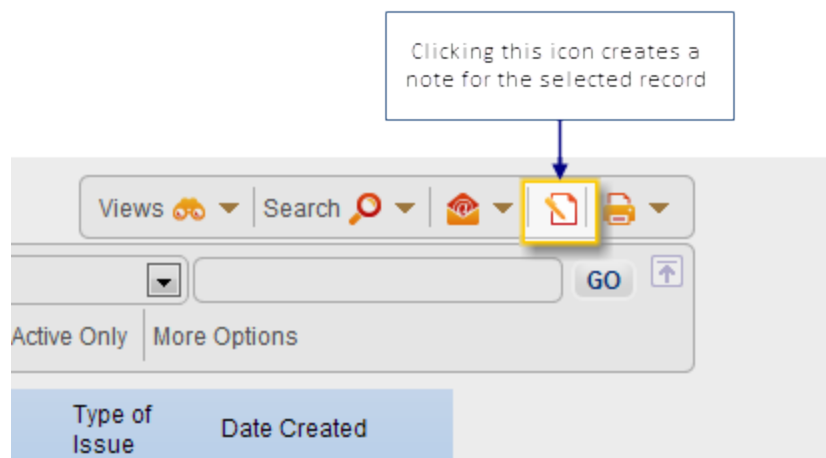
If you wish to display the table to some users at a later date, go to Setup > Access > Manage Groups and turn it on for those groups of users

How do users add notes to a record? I want users to be able to add notes and have a thread of communications in my support case table. What is the best way to accomplish this?

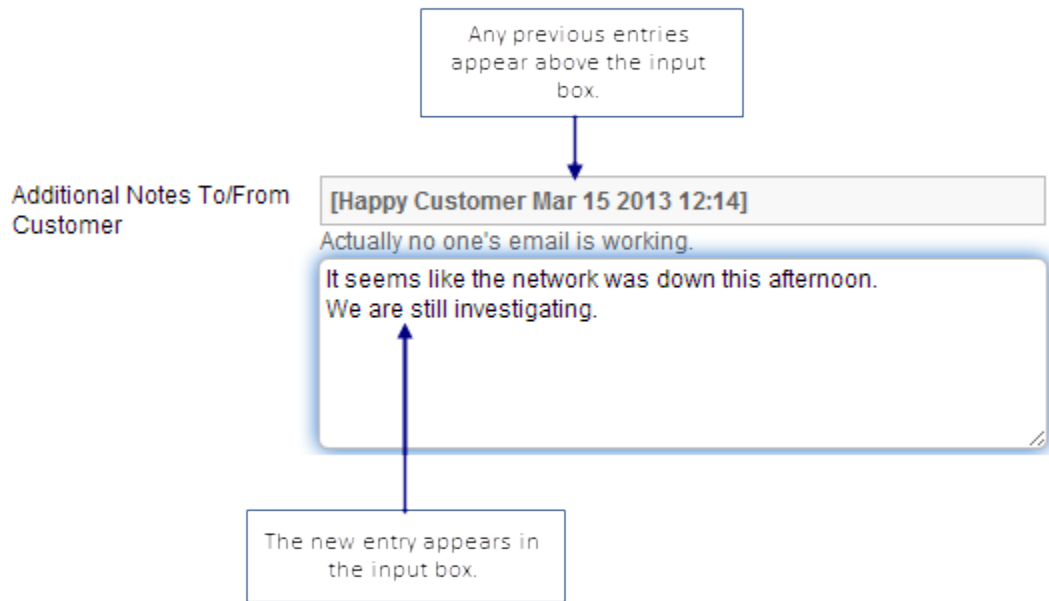
There are two ways to enable users to add notes and display them as a thread of communications in a record.

One method is to put them in the communications sub-table, which is automatically created when you create a new table.

Both notes and emails can be stored in this sub-table and displayed within a record. A user adds a note or email by clicking on the appropriate icon to the right of the tabs when the record is open:



The other method is to create a large text field of type Append Only Text, with this field you will get a thread of communications each of which is date/time/user stamped displayed on the normal layout form.



One difference between the two methods is that items in the communications table are permission controlled—that is, you can set up permissions so that only staff can see all notes, while customers can only see those that are marked Public (by the person creating the note). All items in an append-only text field are visible to anyone who can see the field. So you might use the append-only text field to solicit comments from your end users while using the communications notes to store internal notes by your staff about work they have done.

One advantage of using an Append-only notes field is that the data is stored in the actual table (Support Cases) rather than in a sub-table (Support Case Communications), which means it is easier to trigger notifications based on changes to the field and to include this field in any outbound emails.

For instance, if you have your staff enter comments for the end user in the Additional Notes field, you can have a rule that automatically emails the customer whenever new notes are added. You can have a second "Staff Only Notes" field for internal comments only. The default setup provides these two fields set up with the correct permissions.

There is a new option in the Table edit wizard on the Indexes tab called 'Create FTS index for this table'. What does this option do?

FTS index stands for Full-Text Search index.

A full-text search is the one that you normally get when you just use the Search Pane with some keywords (for example: FTS Index)

Once run it is depicted above the table data as the following, meaning 'Show me the records where these words occur anywhere in the text': (Search: -TEXT-~='fts' and -TEXT-~='index')

Without an FTS index the full-text search won't work for the table.

We have added this option to the Setup/Tables/Edit Table wizard on the Indexes tab to turn full text searching off (by deselecting the checkbox titled Create FTS index for this table), because in some cases, FTS may not be desirable for a table.

Questions about Communications tables

- 1. When I change the label of a table, the label of the related communications table does not change. Is there any way to change the label of the communications table to keep it matching the name of its master table?**

Communication Tables are very much usual tables. The label can be changed on the first tab of the Table wizard: Setup/Tables/YourTable/Edit

- 2. If I delete a table, shouldn't its communications table be automatically deleted? (this is not happening in my KB)**

Yes the Communications table is deleted if both of them are marked as Deletable. The Communications table inherits this property from the main one on create, but from this moment on it is not being synchronized. If you try to delete the main one when it is marked as non-deletable there will be a warning. It is natural to forget to change the flag on the Communication one as well when this happens.

- 3. What are the limitations related to editing / adding / deleting fields in the Communications Tables?**

As mentioned above Communication tables are very much usual tables. However it is not recommended to remove any fields that are created by default. You may change labels, the way they are depicted, but do not remove them. If you have a field you want to hide - use Hidden tab in Layout editor

Putting a new table on Toolbar.

I have created a table and tried to give access to my group. I don't see the new table tab when logging in as a user of that group even though I gave the user all the privileges.

Looks like the tab did not appear because your group had the option "Show table on Toolbar Tab?" set to "No" in Setup/Access/Manage Groups/YourGroup/Table/YourTable. Set it to "Yes" and your table will be visible.

Permissions influence user's ability to work with the records in the table, not only via the Toolbar Tab, but also Linked Field relationships, embedded tables, etc. However this specific option only influences visibility of the table on the tab - one

may want the users to be able to work with the records but not show them the table directly.

Related table - Need to hide some buttons.

I am using a Related Table in my layout and in a specific case I do not want buttons NEW and DELETE to show up on the toolbar. Is it possible to hide them?

The visibility of these buttons depends on user's group permissions. If one has no permissions to Create/Delete records in the linked table the buttons are not visible.

EMAILS

How does the system know which record to update when an email update is received?

The outbound email header contains a whole lot of encrypted information about the table and record ID from which it was sent. Because MS Exchange strips out headers when sending internal emails, Agiloft is also capable of adding this encrypted information to the subject line of outbound emails as a backup.

The outbound email headers also contains full details about how and why the email was sent, including the server URL, KB, table, rule or workflow ID that sent it, and so on. This information is displayed in the communications record header field and can be viewed in your email client e.g. in Outlook by opening the email and clicking View/Options then scrolling through the Internet Header display. This is useful when testing or troubleshooting surprise emails of the "why did I get this email?" variety.

It seems if an email template includes variables in the subject (like \$summary) and the email is sent as collated, these variables do not get replaced by field content and are sent as is.

This is as designed. Since the email contains several tickets collated with different subjects, the subject line will show only the field variable as it cannot resolve multiple fields at once.

If you are creating an email to notify people about multiple tickets, it may be best to leave field variables out of the subject line and body of the email if you wish to avoid this.

How do I disable incoming and outgoing email for testing purposes? There are times when you want to have a backup of your main production system in order to test new development. In these cases, it may be desirable to turn off outbound email notification and inbound email popping.

1 - To stop all email activities for all projects (requires JBoss restart)

- find JBoss launcher script (please make a backup copy)
- edit it, add 2 properties stop.mail.receiver=true and stop.mail.sender=true In result the script should contain a line similar to `JAVA_OPTS="$JAVA_OPTS -Dstop.mail.receiver=true -Dstop.mail.sender=true -Djava.awt.headless=true -Dprogram.name=$PROGNAME"`
- restart JBoss

If you want to start all mail activities again, overwrite JBoss launcher script with backup copy and restart JBoss again.

2 - To stop selected email activity for selected projects (does not require JBoss restart)

- log in admin interface
- go to the last tab named "Mail daemons"
- you will see 2 sections
- with daemon name=[com.supportwizard.swmail.daemon.Receiver] - it manages inbound activities
- with daemon name=[com.supportwizard.swmail.daemon.QueueManager] - it manages outgoing activities
- press "stop" button against selected projects for one or both daemons

If you want to start email activities back - press "start" against selected projects on the same page.

Can we update specific fields by email like we did in SupportWizard?

Yes. In the Autumn 2008 release, functionality was added for email parsing. This allows the user to provide values for fields within a designated formatted section of an email.

How to change display order of ticket fields in the auto-generated emails? Is it possible to control the order in which fields from the ticket form appear in the email that is auto-generated and sent out by the workflow? The order in which they appear doesn't appear to match the order in which they're listed in the workflow setup.

If you are sending an email about a single ticket, you can replace the automatically listed fields (whose order cannot be changed - this order is determined by their order in the field table) with variables for the fields you want to include. The field variable is its name preceded by "\$". So you can put the fields in the body of the email instead. For instance, in the email body you might type:

```
Here is the solution: $Solution
Details:
Description: $Description
Summary: $Summary
Email: $Email
```

and so on, selecting whatever order and layout you like. Of course you would then delete the fields from the list of fields being sent (or choose to send no fields).

Where did the email come from? I received an email from Agiloft, but I do not know what KB or rule generated it. How can I found out

Look in the email "Long Headers". The SMTP headers will contain Agiloft Audit information that includes the name of the KB and the rule or Workflow transition that generated the email. To see these headers in Outlook, click to view the message, then click Options to view the Internet headers.

How to enable automatic case creation via email submission. E.g. user sends an email to ops@example.com, Agiloft creates a case number, auto-replies to the user that case has been created and sends an email to Ops staff for handling.

The procedure to accomplish this is as follows:

1. Set up an inbound email in Agiloft for the Case table and map the email parameters like address, subject, body to the required fields of the table. (Setup/Email/Configure Inbound Email/New/Table/{choose Case}/Server/{enter example.com}/Account/{your account credentials}/Record Mapping/{map the fields here}/ Email Updates/{map further updates to the ticket}/Access/{decide what to do with unknown senders}/Filter/{apply filters to inbound emails if required})
2. Test (1) by sending an email to ops@example.com and confirming a case record is created.
3. Decide whether you would like to use a Rule or a Workflow action to handle auto-email-response to Case record creation.
4. If you opt for Workflow, edit the workflow for the Case table and add an Email Action to the transition. (Setup/Workflow/Table/{choose Case}/Edit/{double-click the state transition}/{select an existing email template or create a new one and add it to the selected items}/Finish. You could add two email templates over here - one to acknowledge the sender, the other to notify the Ops staff.

The Workflow editing interface requires Java Plug-In to be installed in the browser.

5. If you opt for a Rule, then create a rule for the Case table and add an Email Action on case creation (Setup/Rules/New/General/{select Case table}/{give the rule a name}/Rule type/{select When Case is created}/Condition/{select When a support case is created}/Schedule/{select Always}/Action/{Create or use existing email actions and add them to the Selected Actions}/Finish)
6. Test (4) or (5) by sending a test email to ops@example.com and confirming the receipt of the acknowledgment and notification emails.

Our Demo project already has a rule set up to acknowledge the sender of a new case creation. If you are using such a project, then you may just need to modify it to add another email template that notifies the Ops staff.

ACTIONS

Is it possible to control the order in which fields from the ticket form appear in the email that is auto-generated and sent out by the workflow? The order in which they appear doesn't appear to match the order in which they're listed in the dialog setup.

If you are sending an email about a single ticket, you can replace the automatically listed fields (whose order cannot be changed-this order is determined by their order in the field table) with variables for the fields you want to include. The field variable is its name preceded by \$. So you can put the fields in the body of the email instead. For instance, in the email body you might type: Here is the solution: \$Solution Details: Description: \$Description Summary: \$Summary Email: \$Email and so on, selecting whatever order and layout you like. Of course you would then delete the fields from the list of fields being sent (or choose to send no fields).

KNOWN ISSUES

When using IE 7 to access either my Demo KB (hosted at Server 14) I'm getting the IE "Library Loading Error".

When using IE 7 to access this Agiloft Support KB, I do not get this error, but upon trying to create a ticket, I get a "cannot run the code" IE error.

HOWEVER, with Firefox, I can access both KBs without any problem.

Please clean the browser cache, seems like a file got stuck and references a wrong library.

Internet Explorer 7

1. From the Tools menu, select Internet Options... .
2. Choose the General tab.
3. Under Browsing history, click Delete... .
4. Next to ""Temporary Internet Files"", click Delete files... .
5. Click Close, and then click OK to exit.

Firefox 2.0

1. From the Tools menu, select Clear Private Data, and then choose Cache.

Sometimes, a certain page will show a JavaScript error in the lower left of the browser window. This can prevent a screen from being saved or prevent a field variable from being automatically pasted into an email or edit screen

Usually this kind of problem is caused by Internet Explorer having cached some invalid data or old code. IE sometimes fails to delete cached data that has expired, which can cause conflicts and this particular type of error.

After clearing your browser cache, try the same action again to see if this has fixed the problem.

Another alternative is to temporarily switch browsers. Sometimes an error will occur only in IE and Mozilla will work without any problem.

Table size warning!

I created a new field in the ticket table called customer reference. After saving the field, there is an error message at the top of the ticket table.

Warning: Table size: 9672 maximum, 8090 possible.

This is a common message and can generally be ignored. The maximum record size for an individual record in MySQL is 8090 bytes. As soon as the fields defined for a table can theoretically hold more than that, you will get this message whenever you add a field. All production tables are over the theoretical "limit" in size, and this has never caused a problem, so the message is more informational than actionable.

Linked fields are included in the calculation and depending on how you set them up, they can reserve a lot of space. Also any text or append only field reserves 512 bytes (the remainder of the field is stored separately), so if you have a lot of long text fields, the limit can be reached pretty easily. The likelihood of a record containing the maximum value for all included fields is pretty low, and in real use we have never had the limit prevent a record from being saved.

Since large fields tend to reserve more space than they actually use, it is more likely that the record size limit would be reached by a record containing a very large number of small fields rather than by a record with several large fields.

Note that this limitation is only for MySQL. MS SQL does not have this limit.

HOTLINKS

How can I change the URL for the hyperlink sent in emails? When setting up Email actions, you can insert a hyperlink into the email so the user can click to open his record for editing. Sometimes because of system configuration issues, the default hyperlink URL may not be correct.

The solution is simple. There is a global variable that can be created to identify the correct hostname. Follow these steps:

1. Go to Setup/System
2. Click the Manage Global Variables button
3. Check to see if there is already a variable named hotlinkServerRoot. If there is, edit it according to step 6.
4. If the variable does not already exist, click on the New button to create a new variable of type URL Constant.
5. Name the variable hotlinkServerRoot (with exactly this capitalization).
6. Type the start of the URL that corresponds to your machine, i.e. http://server24.agiloft.com
7. Click Finish.

This URL will now be used in all outbound hyperlinks that are included in emails.

Do not add ""/" at the end of the URL

How to redirect users entering via hotlink if the ticket cannot be edited

Sometimes when our users click on a link to edit a ticket from inside their email message, they get a login screen. Is it possible to redirect to another page if ticket cannot be open for edit for any reason? Users are confused by the login screen because most of them are using auto-login.

Please note "cancelurl" and "exiturl" parameters in the hotlink provided.

```
<a href='http://{host}:8080/gui2/login.jsp?keyID=0&user={user}
&kb={kbname}
&State=Edit:helpdesk_case.helpdesk_case
&record=4946&expiration=10/12/06
&passwd={password}
&cancelurl=&exiturl=&record_access=edit&gui=no'
arget=_blank >Click here to edit the case</a>
```

Supplying an URL to some page will lead to the desired effect. More details can be found in Agiloft Help under topics "Hotlinks" and "Keywords" in the index.

When creating a new ticket, how do we make a hyperlink to show up in the middle of the ticket? We want "Click here to submit your issue to AppX" to be a hyperlink, after Issue Subtype = "AppX" is selected.

From the point of desired functionality - to open an external web site - the Action Button field is what you need. At this time the condition "only show this field if these values are in that other field" only affects the label and input instructions for Action Button - the button itself or the image, if the image is used, always stays on the screen. This shall be addressed soon.

A result closest in appearance to the one presented is achieved when the hyperlink is simply hardcoded into input instructions of some field that appears or disappears based on that condition. Most of the field types, when visible, produce an input control of one sort or another (input box, radio button). However that same Action Button if used with an 1 pixel transparent image produces exactly the required result.

Obviously either of these will only allow you to map one hyper-linked text value to one or many drop-down box values. In the pure Action Button case though the text displayed will be the same, the actual hyperlink can be different as it can contain values from the fields of the record itself (a cascade choice field off screen can be used to provide different URL based on selection in the master choice), in the entry instructions case it will all be static. So if you want to have links to multiple external systems with different texts being displayed - you will have to create multiple fields each visible only for one of the values being selected in the master choice.

To hardcode a link in the input instructions one may use an HTML fragment similar to the following:

```
<h3><a href="http://www.example.com" target="_blank">  
Go to example.com  
</a></h3>
```

To associate this Action Button field with a transparent image, obtain one - any Agiloft installation has it at <http://yourKBhost:8080/gui2/1.gif> Save it to disk, open the Setup/Tables dialog, select the table in which you want this hyperlink field to appear, click Edit. On the second tab - Graphics, upload the image. It should show up as an empty radio button. Don't select it for the table. Proceed to edit Fields. Add the Action Button field. Select Action button appearance as image, select the uploaded transparent image. Choose "Do nothing option" below. Proceed to Options tab. Select "Only show this field if the field X has value X Y Z". Choose your master choice field. Choose the choice (-s) on which you want the hyperlink to appear. Proceed to Permissions Tab. It should be sufficient to make a copy of permissions from the master choice. Proceed to Display Tab. Enter the hardcoded link as above into Input instructions field. Choose "Input instructions only" for "When Editing the

field" and "Neither..." for "When Viewing the field" Choose "Above the input box" for "Place input instruction" choice. Finish adding the field.

Place the field in the Layout and verify in Preview mode that the linked text appears where intended - it will be in the area "above the field" as seen on Layout screen.

Single Login.

Our customers login to our web site with their assigned IDs and passwords. We want to give them the ability to click on a link and enter a new case without logging into Agiloft. Basically, one ID. We easily import the IDs into Agiloft. Please feel free to reference a tutorial. How do we set this up?

Try to use hotlinks for this.

<http://support.agiloft.com/gui2/help/en/Hyperlinks.htm>

<http://support.agiloft.com/gui2/help/en/Syntax.htm>

<http://support.agiloft.com/gui2/help/en/Examples.htm>

See Also:

See also links here (optional)

RULES AND WORKFLOWS

How do I create a rule that does not let a customer open a closed ticket

We are having problems with customers reopening closed tickets. We do not have any rule notifications to tell us a closed (Final/Hard NOT Soft Closed) ticket is reopened. How can I make it so a customer can NOT reopen a closed (Not soft closed) ticket?

You can set it by the workflow. Setup/Workflow/Select the Ticket table/Edit Select the Final Closed state and remove the outgoing to Reopened state link.

I need to decrease a value when a case is created for customer. For example, JaneDoe company has 10 incidents per their customer plan. Each time a support case is created, the value is decreased. Once the field is < 3, I need an alert. When the field is 0, this company cannot submit new cases. I am aware I will use a rule for the field but what logic should I apply?

A rule applied on creation of a case can have an update action, that traverses the linked field (link to company?) back to the company record and updates the field there. The logic will be exactly as you put it: \$fieldname - 1

Additionally that counter field may be imported into the ticket so another - verification - action can be added to stop the ticket from submission when the counter is zero, or better a routing action (update the workflow field) can be implemented, so as to allow customers to submit but immediately put the ticket into special state where it would remain until the contract issues are cleared and the counter is increased again.

To send an alert one can use a rule running every 5 minutes on the company's table and being triggered by a counter falling below 3.

REPORTS AND CHARTS

I need to run a daily billing report

I need a report for just the last 24 hours. I want to use the "relative Dates" functionality. What should be chosen to give a report for just the last 24 hours?

To achieve the desired result - choose all dates, from the earliest to "Today". Records can be further constrained via Filter with Time Filter specifying last 24 hours.

FIELDS

I created a new field and I can see it when I create a record, but when user X creates a ticket, it is not visible. Why not?

First, find out what Groups, user X is a member of.

Verify how record ownership is defined Setup/Table/YourTable/Permissions.

Check Setup/Table/YourTable/Fields/YourField/Permissions. Maybe user X does not belong to a group that can see this field.

Check Setup/Table/Layout. Maybe this field has not been placed on the layout.

Check Setup/Table/Layout and click the EndUser button. Maybe X is a member of an end-user group and this field has not been placed on the End User layout.

Linked fields display in the search result. How do I make a particular field appear in the results of the search I'm creating?

When creating a linked field you can specify what linked fields you want to display – up to 4. These fields should help your technicians or users identify the correct record.

Go to

Setup / Tables / Support Ticket

Edit

Switch to the Fields tab

Edit 'For Company' field / Display tab

Select the fields to be displayed: {desired fields}

How do I change the default search behavior for linked fields? When we want to add a contact to our ticket, when we hit the magnifying glass, the default search is set to the "this field" radio button. Is there a way to make the radio button "All linked fields" the default?

Yes, simply edit the set of linked user fields (go to Setup/Tables, choose the Ticket table, Edit/Fields, locate the set of fields that includes the user name, click Edit) and on the Display tab, scroll down to where you can select the default search behavior. Choose the option you want there and save the field set and that default will be applied next time someone uses the lookup.

Can the color of the field label be changed or the size of the font to draw attention to these critical fields?

If you need to change the field label font/color you can do the following:

Setup/Tables/Choose the table and click Edit/Fields tab

Edit the field

General tab/Field label

Click 'Use HTML Editor' and make any color or font changes for the label text.

Finish

SEARCH

I have a table with a text field that is filled with codes such as MN-FIN-009, PJ-DIS-027.

Performing a search with arguments such as "FIN", "PJ", ... I get no response. How do I search for such strings in this case?

Using the "More Options" available at the "Search Box", select "Search for Strings".

Is there a practical limit to the number of Saved Searches?

Customer N reporting requirements are driving a large number of Saved Searches, since there's no report level querying. Is the practical limit of Saved Searches the number that are visible in the Saved Search drop-down for a given screen resolution, or is there a hard number limit?

There is no limit for the number of searches in the system. The number of searches visible in the drop-down menu is limited by the screen size and resolution only. To decrease the list of visible searches you can uncheck the 'active' state for some searches in the Saved Search / Manage dialog.

CONCLUSION

Thank you for taking the time to review this manual. If you have feedback or suggestions, please write to doctask@agiloft.com or visit our web site at <http://www.agiloft.com>