Representing a List of Values

This topic discusses the options for representing a list of values within a field in a record. In Agiloft, a list of values can be implemented with a choice field or with a background table whose records are displayed as a list in a linked field in the main table. Often, both options might be possible solutions for holding a set of values, and we must weigh the pros and cons of each to make the best decision.

Typical Use Cases

There are many situations where users need to choose values from a list:

- Selecting a Contract Type within a contract record.
- Choosing the Department for which a contract is being created.
- Choosing an Attachment Type to specify what kind of document is attached.
- Selecting the Company Roles for a company.
- Choosing the Type of Problem to classify a helpdesk case.

Terminology

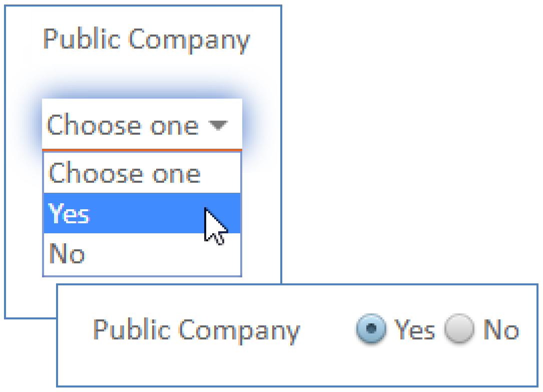
The following terms are used in this article:

Linked field	Linked field is an umbrella term for several data types that create a link from one table to another, most commonly the link to selected fields or the link to single field data types.
Background and process tables	A background table is used primarily to hold values that are selected or displayed in process tables. Process tables typically have workflows, and these are the tables users interact with on a regular basis. Only process tables are usually shown in the left pane for non-administrators.
Choice and multi-choice fields	Choice and multi-choice fields show a list of values to the user. When you create one of these fields, you can reuse an existing choice list or create a new one. Users can select one value in a choice field, while multi-choice fields allow multiple selections.

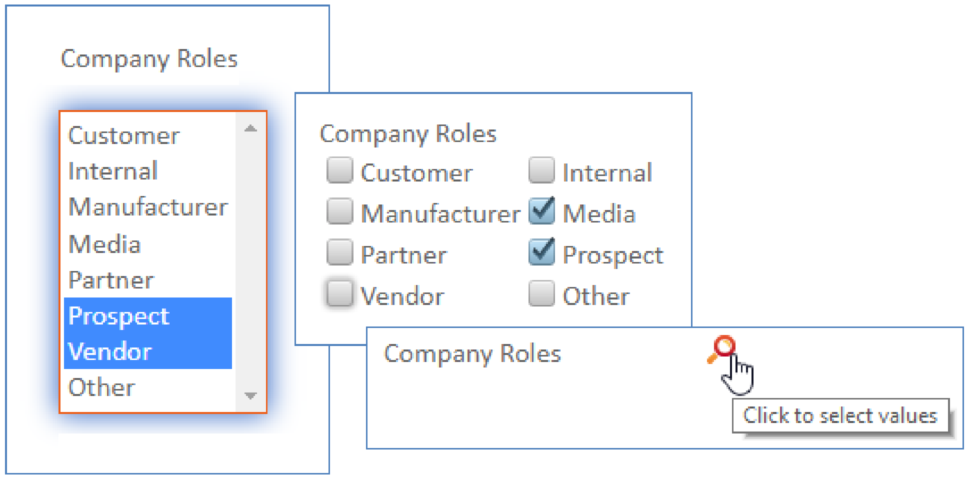
Choice and Multi-Choice Fields

The simplest way to create a list of values is to set up a new choice or multi-choice field. These are easy and quick to set up, and they offer several configuration options that allow you to build automation with other fields. Use a choice field when users need to choose just one value from the list; use a multi-choice field when multiple values can be chosen.

When you build a choice field, you can present the list as a drop-down menu or as radio buttons:

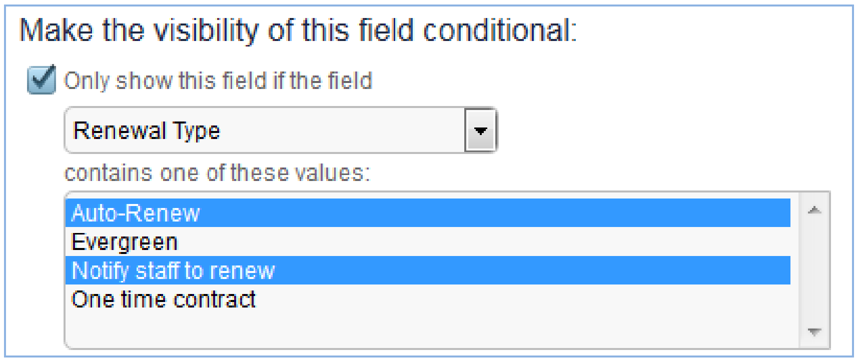


Multi-choice fields can be presented in a list, a group of checkboxes, or a lookup icon that opens a full list of options. The lookup icon also displays the selected values after a user selects something.



Hide other fields with visibility dependence

The visibility of other fields can depend on the selection in a choice field. For instance, a field such as Contract End Date can be made visible if a choice field called Renewal Type has a value of Auto-Renew or Notify staff to renew, but not if it has a value of Evergreen.


Make the visibility of this field conditional:

[✓] Only show this field if the field

Renewal Type

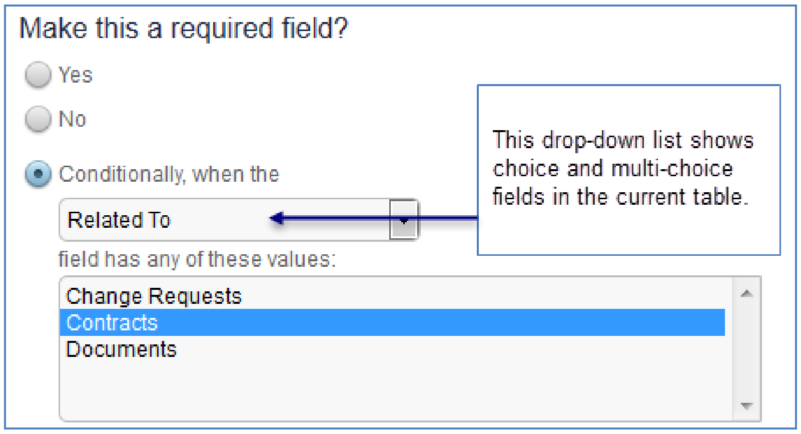
contains one of these values:

- Auto-Renew [✓]
- Evergreen
- Notify staff to renew [✓]
- One time contract

On the Options tab of the field wizard, fields can be made visibility dependent based on the selected values in a choice or multi-choice field

Make other fields required

Other fields in the table can be conditionally required based on a value held in a choice or multi-choice field in the current table. For example, when a task's Related To field is set to Contract, you might make the Contract linked field required.


Make this a required field?

Yes [✓ Conditionally, when the] No

Related To

field has any of these values:

- Change Requests
- Contracts [✓]
- Documents

This drop-down list shows choice and multi-choice fields in the current table.

On the Options tab of the field wizard you can make a field conditionally required based on a choice field

Set up hierarchical dependence

You can configure hierarchical relationships between choice or multi-choice fields so that the values that appear in the child field depend on the selection in the parent field. For example, when you select a Project Category, you can use that selection to determine which options are available in the Project Type list.

You can also use the hierarchical dependence options to set a different default value in a child field depending on the selection in the parent field.

Import new list values

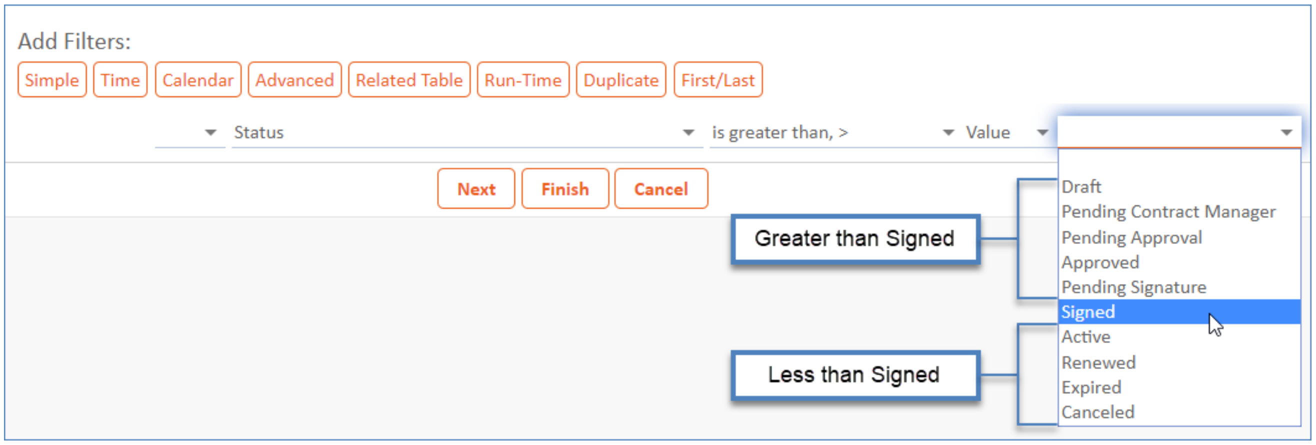
During data imports, you can automatically add missing values to choice or multi-choice fields. Adding these values automatically can save a lot of time, especially when data is automatically imported from external systems.

Localization

Choice and multi-choice values are stored as integers in the database, so the text that appears is determined by the language of the session. If your users speak more than one language, you can use the same internal database value in searches and formulas while showing appropriate localization for your users.

Searching considerations

When searching, choice values are shown as a drop-down list, making it easy to search. They are indexed in descending order, and users can search for values less than or greater than a particular choice. For instance, a search for more than "Signed" finds all items above Signed on the choice list.



In advanced and saved searches, you can use the "is contained in" operator to search for records with any of the selected values. For example, to search for Status values that aren't consecutive, you could use "is contained in" and select three values. Saved searches also store the integer value of the choice rather than the text, so any future changes to the text don't affect existing searches.

Advantages of Choice and Multi-Choice Fields

- Choice fields are easy to implement and simple to use
- Easily make other fields hidden or required based on user selections
- Use one choice field to filter another with the built-in hierarchy wizard

- Permissions are simple and easy to maintain
- Searching is simple and saved searches don't need to be updated for text changes
- Easily add missing choice values during imports
- For multilingual systems, easily maintain value text in multiple languages

Disadvantages of Choice or Multi-Choice Fields

- Only admins can update choice lists
- Managing hierarchical dependencies can be tricky and requires admin permissions
- It can be difficult to build customized field dependencies beyond the built-in hierarchy, visibility, and required field options
- Automation for specific choices must be handled by individual rules and actions per choice value
- Long choice lists are difficult to maintain
- Permissions are all-or-nothing; users can see the field and every choice in the list, or users can't see the field at all

Background Tables with Linked Fields

Another way to present a list of values is with a linked field, pointing to a background table that holds the actual values. Background tables can support more sophisticated relationships and automation, but they don't offer some of the built-in options of choice fields. The background table holds individual records for each value, and they are displayed in a linked field as a drop-down list in the main process table.

One example of this arrangement is the field Contract Type in the default Contracts table. The field links to the Contract Types background table. For more details on how to set this up, refer to the Implementing a List of Values with a Background Table section, which walks through this specific example.

Power users can edit lists

Since each list value is actually a record, power users can be given permission to create, edit, and delete the records, and therefore the list values. For example, in the Contracts/Contract Types scenario, contract managers or members of the legal team can change the existing contract types without asking an administrator to do this for them.

Filtered lists

This setup makes it possible to filter the choices each user sees based on group permissions. For instance, you can show different Contract Type choices to users in different departments by linking contract types to specific departments and showing users only contract types linked to their own department. This can't be done with a regular choice or multi-choice field.

Linked relationships enable sophisticated automation

Using a background table creates a link between the process record and the background record. Establishing a linked relationship allows the system to run automation rules and actions based on a chain of relationships. Having a linked relationship also provides the ability to create a related table in the background table, such as a related table for Contract records linked to a specific contract type.

In addition to the primary field used to make a selection, it is common to add other fields to the background table that are pulled into the main record along with the "selection" field to control automation or other processing. For instance, in the Contract Type example, we can add fields to the background table to set the default workflow, the print template used to generate the contract, the default Internal Signer, and to bring in any other data associated with a particular contract type. The additional fields can be pulled in as part of the linked set when the user chooses the contract type or accessed by rules through the linked relationship, and the linked information can then be used to simplify rules and automation.

Example

For example, imagine you need to assign different internal signers depending on the contract type. With a choice field, you would need a rule with several conditions:

```
if contract type = x
   set internal signer to Graham Smith
else if contract type = y
   set internal signer to Maria Cho
else
   set internal signer to Alex Gonzalez
```

With a background table, you can create an Internal Signer field that can be set in each contract type. You can pull in the Internal Signer field and create a simple rule that copies that field's value to a local Internal Signer field in the table, or you could use an Update Fields action to copy the value into the contract using the linked relationship between the tables.

The ability to bring in other information and thereby reduce automation complexity is a major advantage of using a background table.

Setting visibility dependence

When other fields depend on the selection field's value, we usually handle this by adding a multi-choice field in the background table and bringing it into the main table as part of the linked set. The multi-choice linked field can then be used as the parent of any visibility, hierarchical or requirement dependencies. You can find more information about this and other methods in the Setting Up Visibility Dependence With a Background Table section.

Unless created as a choice field in the background table, the linked field, e.g., Contract Type, cannot be used like a choice field to control the visibility, conditional requirement, or hierarchical choices of other fields in the main table. If the field must control the visibility of several other fields, we occasionally use a choice field in addition to a short text field in the background table, but this is not usually the best solution. For more information about that approach, see The Background Table Choices Solution.

Parent-child list values

When you need to build a hierarchical parent-child relationship using the linked field, you can use one of two main approaches:

- Create the parent field as a choice field in the process table, and create the child values in a background table
- Create the parent and child values in two separate background tables

In both cases, the child background table includes a field that holds its parent values. When one of these values is selected in the process table, child values that aren't assigned to that parent value are filtered out of the list of values.

These approaches are more complex to set up than two choice fields, but they're easier to maintain in the future, and they can be managed by power users as well as admins.

Localization

Localizing list values with a background table is challenging because record data can't be translated or localized. If you need to support localized values with a background table, you need a record for each value and then another record for each translation of that value; you need to add a field to the background table to identify the language; and you need a filter in the process table to show the records in the appropriate language without showing the other translations.

This can also complicate automation because users are selecting different records based on their language, so you need automation to include each translation of each value. If you need to search or report on records that have a certain value selected, you also need to add a field to the background table so you can group each value together with its translations.

Advantages of a Background Table with Linked Field

- It is easy to add any number of records; each record in the background table is a possible value in the choice list.
- Non-admin users can be granted permission to add new records or modify existing records, thereby customizing and/or maintaining the list of available choices. Power users can manage workflows and automation directly.
- You can set more granular permissions. Different sets of users can be shown different records, and therefore list values, based on permission filters.
- Hierarchical lists, where the values in a parent list determine the available values in a child list, can easily be maintained and viewed by non-admin users.
- Additional fields in the background table can hold other important information or data related to that choice value. That information is pulled into a process table along with the record chosen, simplifying automation.
- Using a background table creates a link between the process record and the background record. Establishing a linked relationship allows automation to run based on a chain of relationships through records in other tables.

Disadvantages of a Background Table with Linked Field

- It takes longer to implement a table and the corresponding linked field than a simple choice or multi-choice field.
- A background table adds a layer of complexity to group permissions. In order to see and interact with the values in the process table, users must be granted appropriate access to the background table, its records, and its fields in addition to the linked field displayed in the process table.
- Linked text fields cannot be the parent of a visibility dependence or conditional requirement without some additional setup to implement these dependencies.
- Searching is less intuitive – no value list is displayed and the user has to type in the search criteria. Also, if a saved search is based on a hard-coded value of a linked text field, such as Contract Type=NDA, and someone edits the underlying contract type record and changes the wording of that field to "Mutual NDA," the search will not be updated and will no longer find the records it found previously.

Key Questions

To determine which method is better for your implementation, think about the detailed requirements. This set of questions guides you toward choosing the best field type to implement.

1. **Who maintains the data?** If power or end users maintain the data, a table is the only option. Choice lists cannot be modified by power users. If only administrators modify the choice list values, then either option may work.
2. **Is there any other data and/or automation based on this choice selection?** For example, should selecting the item result in a different set of default field values, a different assignee, different print templates or workflows applied? If so, a background table is definitely the best option, as it allows cleaner and simpler automation that can be differentiated and maintained by non-admin users.
3. **Do all users see the same value list? Or, should the list be filtered in any way based on who is viewing it?** If you need to filter the list for different user groups, a linked field with a background table is the best option.
4. **Is the visibility of several other fields dependent on the value selected?** If the visibility dependence is very simple and not likely to change, a choice field is easiest. If the dependent relationships are dynamic, you may instead want to use a background table. This can then be maintained by non-admin users, which is an advantage over choice fields.
5. **Will you be syncing the list values with a back-end system through an automated import?** Choice fields have the advantage of an import option to add missing choices automatically. If the data may change in the back-end system and you are using a background table, you need to do a separate import/update of the background table records first before importing into the process table.
6. **Does your system use multiple languages?** If so, a choice field is simpler to use. Choice fields can be translated while still maintaining their unique underlying values in the database. Record data can't be translated or localized, so using a background table in a multilingual system requires a lot of additional setup.

Sample Scenarios

After deciding which option makes more sense for your implementation, keep in mind the pros and cons of that method. No matter what decision you make, it impacts the rest of the design and build-out process. For a summary of the factors to consider when determining the benefits and drawbacks of each option, see Criteria for Evaluating Design.

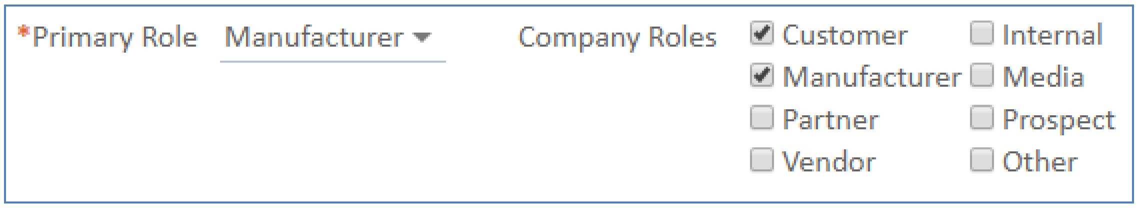
Choice Field Examples

Let's look at some examples where a choice field was implemented:

Company Roles in Companies

In the Companies table, the field Company Roles is a multi-choice field. In this situation we use a multi-choice field because:

- There is no additional data or automation associated with a company's Company Roles;
- The number of values for Company Roles is not very large;
- The Company Roles values can be easily made hierarchically dependent on Primary Role as a choice field using built-in settings in the field wizard.
- It is easier to search for companies with specific roles using a choice field.



*Primary Role: Manufacturer

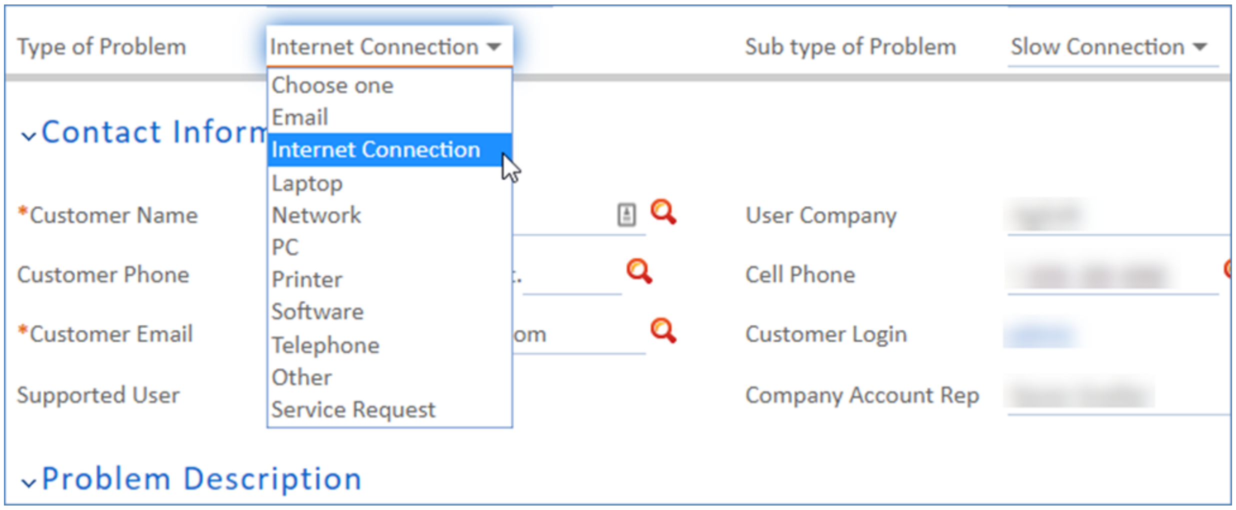
Company Roles: [✓ Customer] [✓ Manufacturer] Partner Vendor Internal Media Prospect Other

Company Roles in Companies

Type of Problem in Helpdesk Case

In a Helpdesk Case table, the default field Type of Problem is a choice field. This is no longer part of the out-of-the-box configuration, but the example can still be useful in analogous situations. Here, we implemented a choice field because:

- There is no default automation associated with the Type of Problem.
- The list of problem types is relatively short.
- The available Subtype of Problem values are hierarchically dependent on the Type of Problem chosen, a simple use of the built-in dependence wizard.
- These values are easy to search, and people often want to search by this field.
- Any changes to the values or dependencies are only made by an admin user.



Type of Problem	Internet Connection	Sub type of Problem	Slow Connection

Contact Inform

*Customer Name		User Company	[redacted]
Customer Phone	.	Cell Phone	[redacted]
*Customer Email	om	Customer Login	[redacted]
Supported User		Company Account Rep	[redacted]

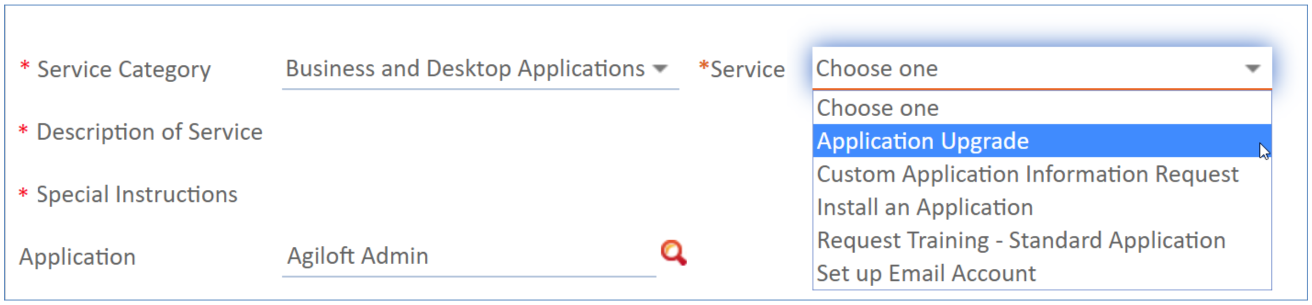
Problem Description

Type of Problem in Helpdesk Cases

While this is a simple setup that is easy to manage, it is not adequate for a more sophisticated implementation. In a sample ITIL KB, the more robust table-based solution for the Service Requests table replaces the Helpdesk Cases table from the previous example.

Service Category in Service Requests

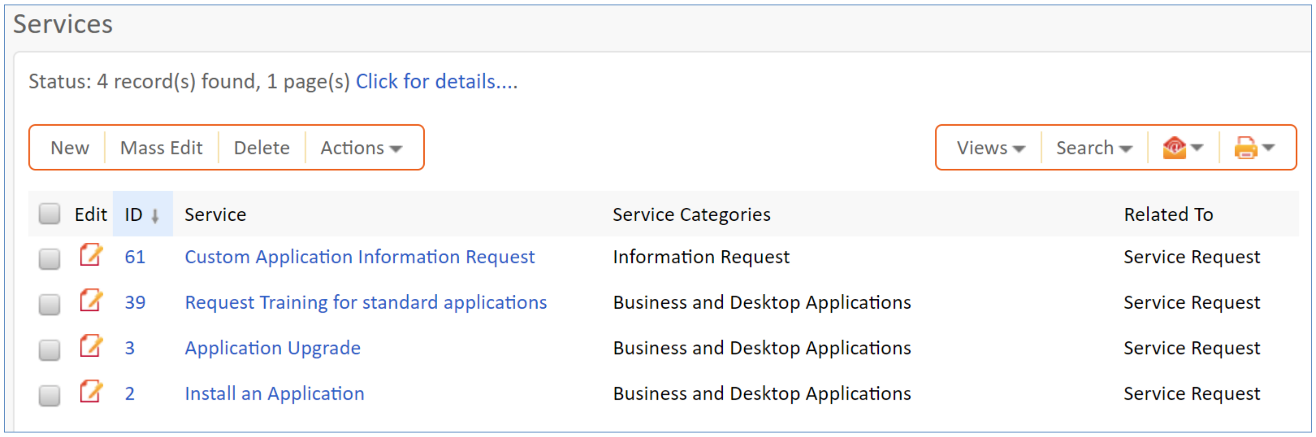
The Service Requests table uses a choice field for the Service Category, which is the parent field, and then a background table called Services that defines the child choices for particular services. The selected Service defines a description of service for each new service request as well as its default assigned team, service-level agreement (SLA) additional fields to show, and special instructions. When a user selects a Service, they also see any other fields that are visibility dependent on that Service.



* Service Category	Business and Desktop Applications	*Service	Choose one
* Description of Service			
* Special Instructions			
Application	Agiloft Admin		

Choose one
Application Upgrade
Custom Application Information Request
Install an Application
Request Training - Standard Application
Set up Email Account

Service Category in Service Requests



Services

Status: 4 record(s) found, 1 page(s) Click for details....

Edit	ID	Service	Service Categories	Related To
	61	Custom Application Information Request	Information Request	Service Request
	39	Request Training for standard applications	Business and Desktop Applications	Service Request
	3	Application Upgrade	Business and Desktop Applications	Service Request
	2	Install an Application	Business and Desktop Applications	Service Request

Services

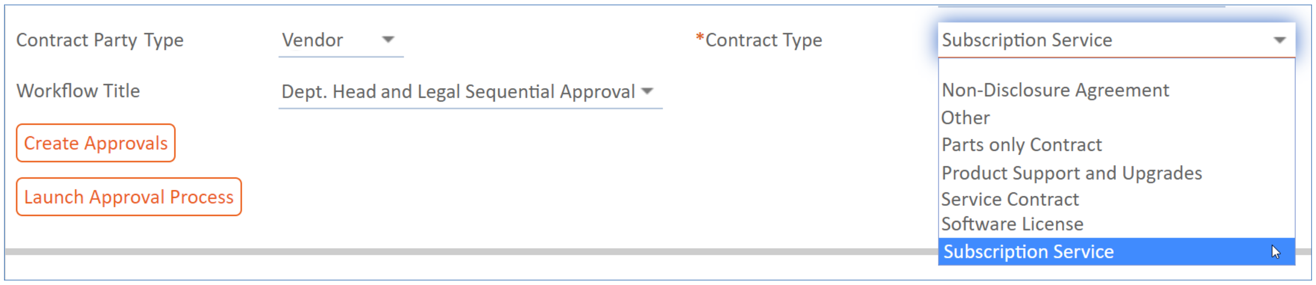
Linked Field from Background Table Examples

Let's look at some additional examples where a background table and linked field were implemented:

Contract Type in Contracts

The field to specify Contract Type within a contract record is a link to the Contract Types background table. A background table is the better option because:

- Information specific to each contract type can be pulled in when the Contract Type is selected, such as a default approval workflow and print template – using automation to generate customized approvals and a specific document.
- The list of contract types can be filtered for a user based on the user's department, permissions, or any other criteria. Often you want different users to see different values.
- Power users such as contract managers or the legal team need to be able to create or edit existing contract types. If Contract Type were a choice field, contract managers would not be able to edit or add to the list of options unless they are given admin privileges.

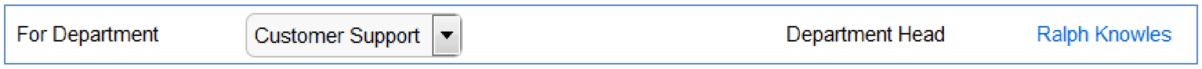


Contract Party Type	Vendor	*Contract Type	Subscription Service
Workflow Title	Dept. Head and Legal Sequential Approval		

Department in Contracts

Within a contract, the For Department field is a link to the Departments table. This is the best option because:

- Thanks to the linked relationship, we can create a related table of Contracts in the Departments table to show all contracts for a particular department. We can also show the total value of their buy-side contracts and compare it to a departmental budget.
- We can also track other information about the department, such as a default account code for billing, specific contract types available for that department, and so on.
- By making Departments a background table we can link users to departments and identify a department head who is pulled into the contract automatically, making it easy to generate an approval for the relevant department head.



Based on the selection in the For Department field, the Department Head is shown in the contract record

Attachment Type in Attachments

An Attachment Type is selected from a list in an Attachment record (attachments are shown in a table within a contract record). In this situation:

- There are a relatively small number of values, and they are relatively static.
- Selecting different attachment types results in different fields being shown in the Attachment record. For instance, some attachment types need to display an Expiration Date field while others do not.

This decision could have gone either way. A choice field would have handled the visibility dependencies more easily, but we wanted non-admin users to be able to modify the attachment types, which is only possible with a table. We therefore chose to implement Attachment Types as a background table, with our standard "Extra Fields to Show" setup used to handle visibility dependence—see The "Extra Fields to Show" Solution below.

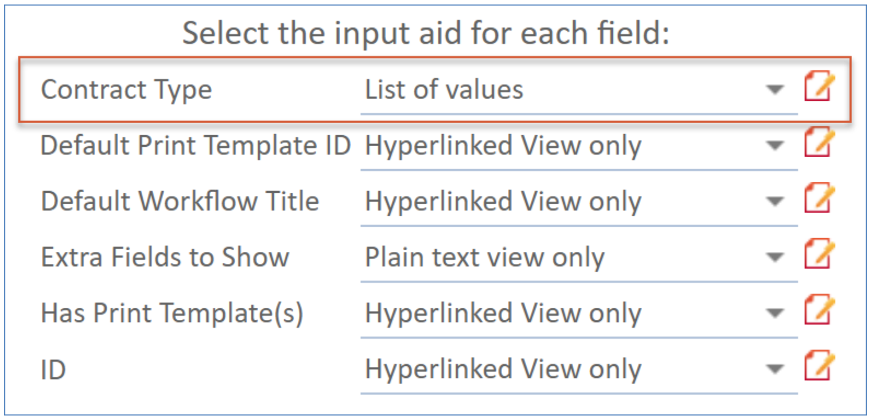
Use Cases

Below are some examples of implementation solutions for creating a list of values.

Implementing a List of Values with a Background Table

This example implementation describes the method used to create a list of Contract Types to choose from in the Contracts table. You can follow along and generalize this process to any pair of process and background tables to suit your needs.

1. First create a table called Contract Types. This is now the background table.
2. Add a short text field called Contract Type (typically you will choose the options to make this field the Summary Field, required, and unique on the Options tab).
3. Add any other fields you want to associate with a contract type, such as default workflow, print template, etc.
4. Once the background table is saved, create a record for each Contract Type that will appear in the list. For example, create five records with these values in the Contract Type field: Service Contract, Non-Disclosure Agreement, Software License, Subscription Service, Product Support and Upgrade.
5. Create a new field in the original table (the Contracts table) that links to the Contract Type field in the Contract Types table.
    a. For a single value choice list (users can only select one contract type), use a Link to selected fields from other table field type. Include the Contract Type field and any other fields from the background table that you want. To display the values as a drop-down list, choose List of values for the primary field on the Display tab.



Select the input aid for each field:

Contract Type	List of values
Default Print Template ID	Hyperlinked View only
Default Workflow Title	Hyperlinked View only
Extra Fields to Show	Plain text view only
Has Print Template(s)	Hyperlinked View only
ID	Hyperlinked View only

    b. For a multi-value list, use a Link to single field from other table and select either Yes, fast search or Yes, save space on the Mapping tab of the field wizard. This parallels the basic function of a Multi-choice field and allows you to display the contract types as checkboxes or as a multi-value lookup.

Setting Up Visibility Dependence With a Background Table

You are implementing a value list using a linked background table, like the Contract Types example above. In this case, you have an Attachment Types background table used as a linked field in the Attachments table. There are several fields in the process table you want to make visibility-dependent on the value selected in the selection field, but you can't set up visibility dependencies based on a linked short text field.

Below are two solutions for visibility dependence with background tables.

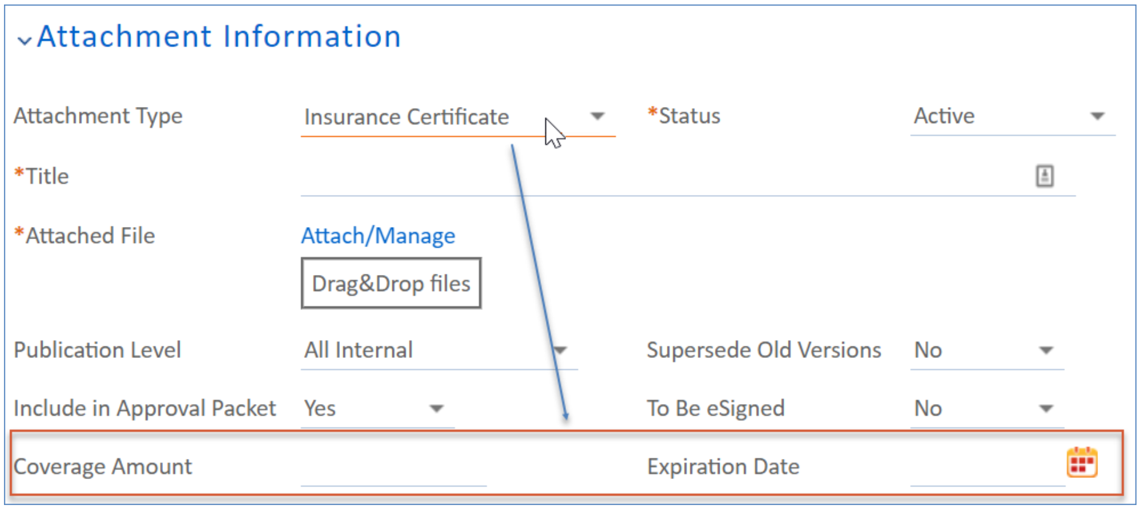
The Extra Fields to Show Solution

With this solution, you would create a multi-choice field in the Attachment Types table whose only purpose is to control the visibility of other fields. We usually name this type of field Extra Fields to Show, or simply Fields to Show, and each item in the choice list is the name of one of the fields in the Attachments table that only appears conditionally.

Once you have the field created, update some attachment types to give them values in the field as appropriate.

In the Attachments table, update the linked set to the attachment type and add the multi-choice field to the linked set as view only. Now, you can edit other fields in the attachment table and make them dependent on the Attachment Type new field values. For example, Coverage Amount and Expiration Date only appear in the form for specific attachment types.

Multiple Attachment Type records can include the same selections in the Extra Fields to Show field, and this makes it easier to maintain the other Attachment fields, since you can use just one condition to determine whether to show the field. This system is also easy to explain to non-admin users who need to maintain the records, and it offers transparency on which fields appear for each attachment type.



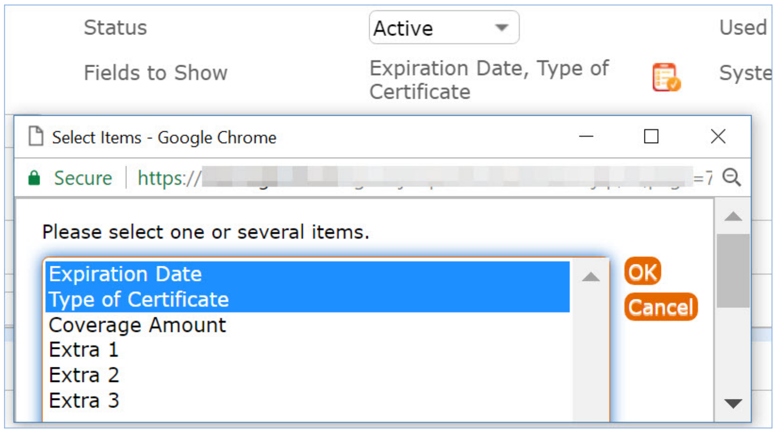
Attachment Information

Attachment Type	Insurance Certificate	*Status	Active
*Title			
*Attached File	Attach/Manage Drag&Drop files		
Publication Level	All Internal	Supersede Old Versions	No
Include in Approval Packet	Yes	To Be eSigned	No
Coverage Amount		Expiration Date	

Setting up Multi-Choice Values in the Extra Fields to Show

The simplest method to define necessary choice values in Extra Fields to Show is to add a choice value for every field in the process table that depends on the Attachment Type. In other words, set up the choice list so it contains the names of all visibility dependent fields.

Multiple records in the Attachment Types table can include the same field in Extra Fields to Show. For example, the Coverage Amount field appears for both "Performance Bond" and "Insurance Certificate" attachment types. When setting the visibility dependence of the Coverage Amount field in the Attachments table, we make it visible if the value contains "Coverage Amount" and the field will appear for all the relevant attachment types—those in which Coverage Amount is selected in the Fields to Show.


Status: Active
Fields to Show: Expiration Date, Type of Certificate

Please select one or several items.

Expiration Date
Type of Certificate
Coverage Amount
Extra 1
Extra 2
Extra 3

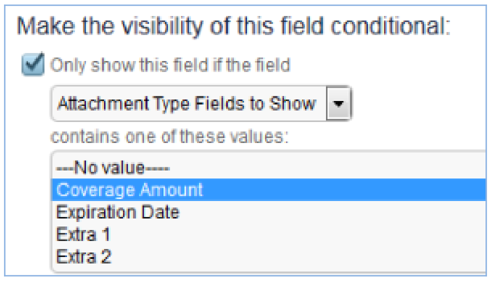
The advantage of adding a choice value for each visibility dependent field in the process table is system transparency for admins and users. It makes it clear to someone who manages Attachment Type records exactly which fields will appear if they select a particular value.

One choice value per dependent field also simplifies setting up visibility dependence for those fields—the admin just selects the field's own name as the value to match.

Process Summary

These steps summarize how to implement an "Extra Fields to Show" setup:

1. Create a background table and short text field to act as the value list in the process table.
2. In the background table, create a multi-choice field called Extra Fields to Show.
3. Create the choice list for Extra fields to Show and include a value for each visibility-controlled field in the process table.
4. Edit each relevant record in the background table. Select the dependent field names from the Extra Fields to Show list that are appropriate for that record.
5. In the process table, create the linked field set to include the short text field and Extra Fields to Show.
6. Still on the process table, edit each dependent field and set the visibility dependence: when Extra Fields to Show contains the value of "this field's" name. For instance, Coverage Amount is visibility dependent on the value "Coverage Amount" in the Attachment Type Fields to Show.


Make the visibility of this field conditional:

☑ Only show this field if the field

Attachment Type Fields to Show

contains one of these values:

---No value----
Coverage Amount
Expiration Date
Extra 1
Extra 2

Adding new dependent fields

If a new dependent field is added to the process table, complete the following steps:

1. In the background table, edit the "Extra Fields to Show" choice list and add the new field's name to the list of values.
2. Edit any applicable records in the background table and add the new field to the selections in Extra Fields to Show.
3. Create the new field in the process table.
4. In the process table, edit the field you just created and set the visibility dependence based on a match in Extra Fields to Show.

The Background Table Choices Solution

Sometimes, instead of creating a Multi-Choice field, the best solution is creating a Choice field in the background table. For example, instead of creating an Extra Fields to Show field, you might create an Attachment Type Choices field in the Attachment Types table.

This eliminates one of the big advantages of the background table because an admin still needs to maintain a Choice field. This approach is only preferable under certain circumstances:

- A background table is needed to pull in other field values and to control automation. Otherwise, you could use a Choice field alone without a background field.
- Table column size limits are a concern in the process table. Multi-Choice fields take up a lot more space (767 bytes vs. 4 bytes) than Choice fields, so in tables such as contracts, that may have hundreds of columns and may hit the limit on column size within the main table, this option can be better.
- There are a limited number of choices that rarely change, but new dependent fields are added more frequently to the process table. With the Extra Fields to Show solution, an admin would frequently need to update the Extra Fields to Show field to accommodate the new dependent fields.
- Only admin users need to add new values. Because the Choice list needs to be updated each time a new value is added, admins must be involved each time.

  - There are several visibility or conditional requirement dependencies in the process table. These can be easier to manage based on a single Choice field, rather than creating intermediary fields in the background table.

Here's how to set this up:

1. Create the background table.
2. Create the short text field as the selection field and a matching choice field with the list of values you want to display.
3. Add any other fields necessary to the background table and save it.
4. With the background table in place, add a new record for each value in the choice list.
    a. In record 1, choose the first value from the choice list. Then, set the value of the short text field to match that value.
    b. In record 2, choose the second value from the choice list. Again, set the value of the short text field to match that value.
    c. Repeat the step above until all records are added.
5. In the process table, create a linked set which includes the short text and choice fields. Add the text field to the layout for users to select a value; the choice field is used in the background for automation.
6. When setting up the visibility dependent fields, use the choice field to define the visibility.

To add new choices to an existing system with this setup, an admin first edits the choice list, then adds a new record and sets the field values to match.