

Getting Started Guide

Alfresco Digital Business Platform - Contract Management

This guide show you how to set up a simple Contract Management system.

You can get going quickly and explore some of the key features and functionality of the Alfresco Digital Business Platform. You'll also learn how these different features can be easily connected to deliver solutions for specific business requirements.

In [45 mins] you'll have created a simple app that can be used to:

- Submit a contract for approval.
- Use information captured as part of that process to determine where the contract is stored, and ensure the required metadata is in place.
- Easily find contracts with customized search filtering.

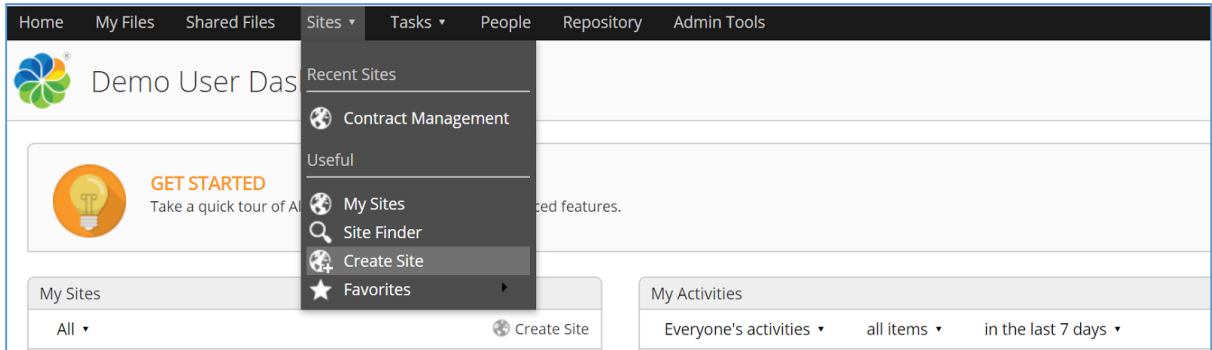
We've included a couple of zip files that you can use to set things up quickly. If you prefer to set up manually then instructions are included in the Appendix.

| | |
|--|----|
| 1. Creating sites and folders | 3 |
| 2. Creating content models | 5 |
| 3. Displaying properties | 6 |
| 4. Activating the model | 8 |
| 6. Configuring the Alfresco Repository | 11 |
| 7. Designing a Contract Approval App | 12 |
| 8. Using the Contract Approval App | 14 |
| Appendix | 16 |
| 1. Manually creating content models | 16 |
| 2. Manually designing a process model | 22 |
| 2.1 Designing a form to submit a contract | 25 |
| 2.2 Designing a form to review contracts | 28 |
| 2.3. Configuring the sequence flow conditions based on contract review | 32 |
| 2.4 Configuring the sequence flow conditions based on contract type | 33 |
| 2.5 Configuring the <i>Publish to Alfresco</i> tasks | 34 |

1. Creating sites and folders

To begin with we'll create a site to store contracts on, so that others can easily collaborate on them. We'll also set up two folders in a document library. Later, when we set up a process, it'll save contracts to the folders, based on what type of contract is uploaded.

1. Sign in to Alfresco Content Services at <http://192.168.10.10:8080/share/page/> using **demo** as the username and **alfresco** as the password.
2. On your dashboard select **Sites** then **Create Site**.

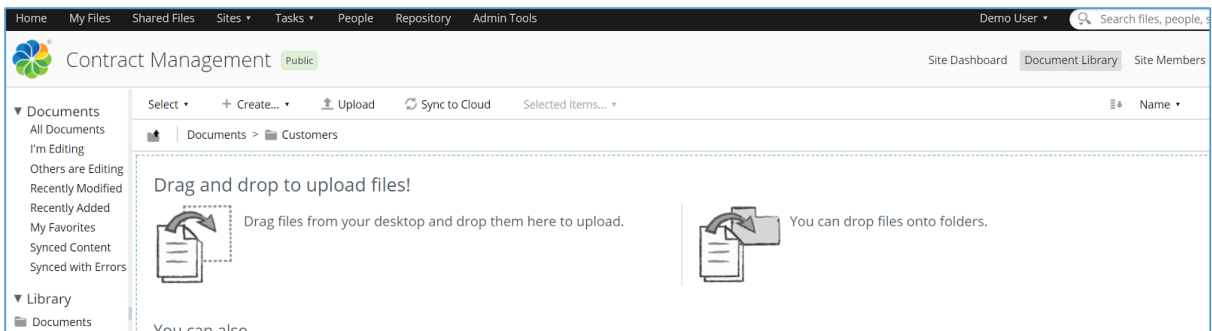


3. In the Create Site screen enter **Contract Management** as the name and leave all other settings as they are, then click **Create**.

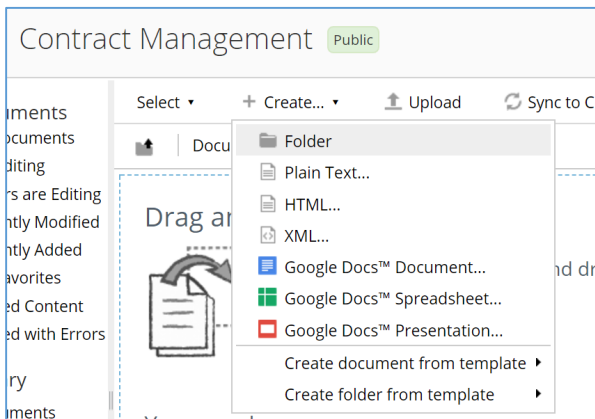
A screenshot of the 'Create Site' form. The form has a title bar 'Create Site' with a close button. The 'Type' dropdown is set to 'Collaboration Site'. The 'Name' field contains 'Contract Management'. The 'Site ID' field contains 'contract-management'. Below the Site ID field is a note: 'This is part of the site address. Use numbers and letters only.' The 'Description' field is empty. The 'Visibility' section has three radio buttons: 'Public' (selected), 'Moderated', and 'Private'. Below each radio button is a description of the visibility level. At the bottom of the form are 'Create' and 'Cancel' buttons.

The dashboard for the new Contract Management site opens.

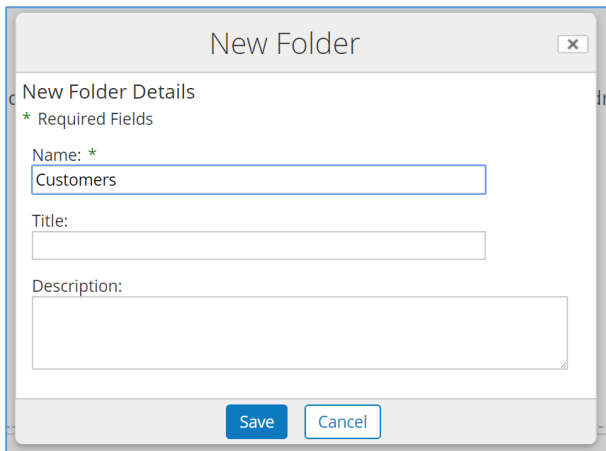
4. Click **Document Library** in the top right corner of the screen.



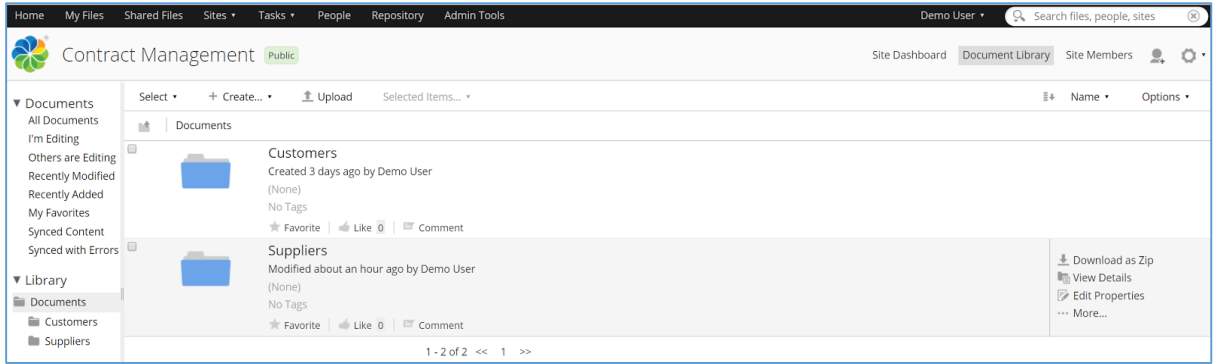
5. Click **Create** and select **Folder** from the list.



6. Type **Customers** as the name and select **Save**.



- Now create another folder with the name **Suppliers**.
The document library should now look like the screenshot below.



2. Creating content models

Next we're going to set up a content model. This is a metadata model that will include different types of contract, and the additional metadata to be stored depending on whether the contract is a 'supplier' or 'customer' contract.

These steps show you how to set it up using the downloaded zip file. If you prefer to set up the model manually then see [Appendix 1](#).

- Click **Admin tools** on the toolbar, then select **Model Manager** from the options on the left hand side.
- Click **Import Model** and, choose the **ImportContentModel.zip** file, and click **Import**.
The **Contract_Management** model is now displayed. If you click on **Contract_Management** you can see the Type and Aspects that it contains.

| Models | | | |
|---------------------------|--|----------|-----------|
| Name | Namespace | Status | Actions |
| Contract_Management | contract_management | Inactive | Actions ▾ |
| dynamicSecurityMarksModel | http://www.alfresco.org/model/dynsecuritymarks/1.0 | Active | Actions ▾ |

| Contract_Management | | | | |
|---|------------------|------------|--------|-----------------------|
| | | | | Inactive Parent Model |
| << Show Models Create Custom Type Create Aspect | | | | |
| Custom Types | | | | |
| Name | Display Label | Parent | Layout | Actions |
| cs:Contract | | cm:content | Yes | Actions ▾ |
| Aspects | | | | |
| Name | Display Label | Parent | Layout | Actions |
| cs:Customer_details | Customer details | | No | Actions ▾ |
| cs:Supplier_details | Supplier details | | No | Actions ▾ |

3. Displaying properties

Now all the model is created, you can decide how it's properties are displayed using the Layout Designer.

1. Click on the `Contract_Management` model.
You'll see the properties you defined in the Layout Designer.
2. Click **Actions** on the `cs:Supplier_details` line and select **Layout Designer**.

Contract_Management Inactive Parent Model

[<< Show Models](#) [Create Custom Type](#) [Create Aspect](#)

Custom Types

| Name | Display Label | Parent | Layout | Actions |
|-------------|---------------|------------|--------|-----------|
| cs:Contract | Contract | cm:content | Yes | Actions ▾ |

Aspects

| Name | Display Label | Parent | Layout | Actions |
|---------------------|------------------|--------|--------|-----------------------------------|
| cs:Supplier_details | Supplier details | | Yes | Actions ▾ |
| cs:Customer_details | Customer details | | No | Layout Designer Edit Delete |

The Layout Designer opens with the aspect's properties displayed.

Contract_Management - Supplier_details Inactive Parent Model

[<< Show Types and Aspects](#)

Use the Layout Designer to define how properties you create are displayed.

Properties

- Aa Supplier name [Supplier_name]
- # Supplier number [Supplier_number]
- Aa Supplier status [Supplier_status]
- + Create Property

Layout Area

Step 1: Drag one or more layout elements onto the Layout Area.

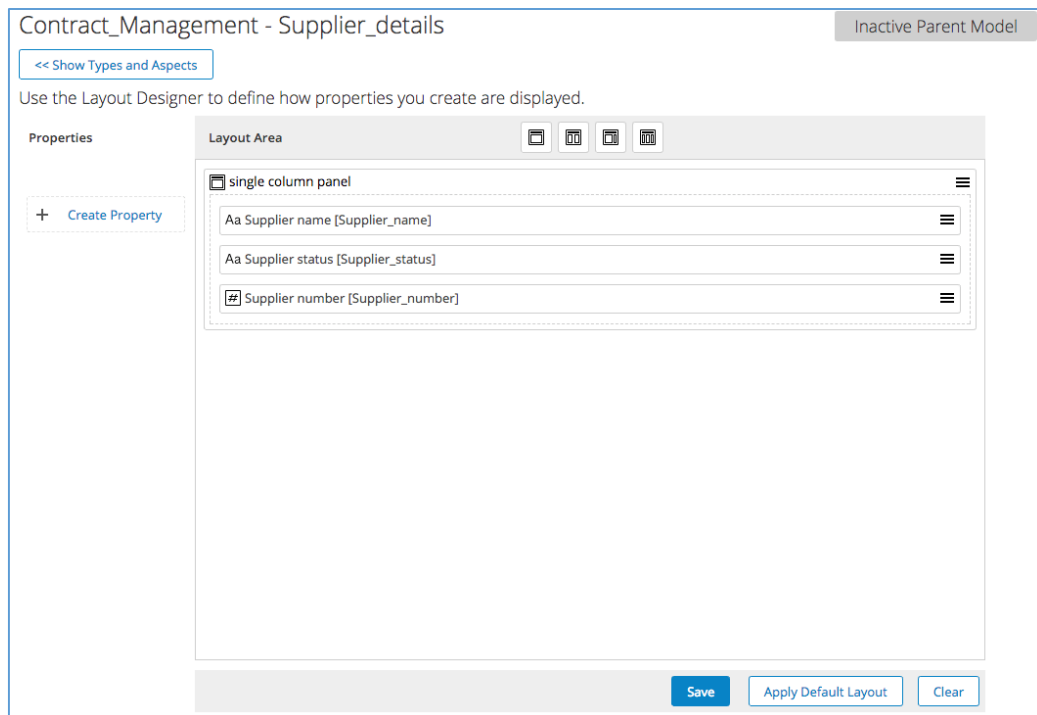
Step 2: Drag one or more properties onto each layout element. Hover over an element or property icon to reorder or delete it. Click an element or property header to edit it.

Step 3: Save your layout! You can always edit it later. Apply Default Layout adds all properties to a single column.

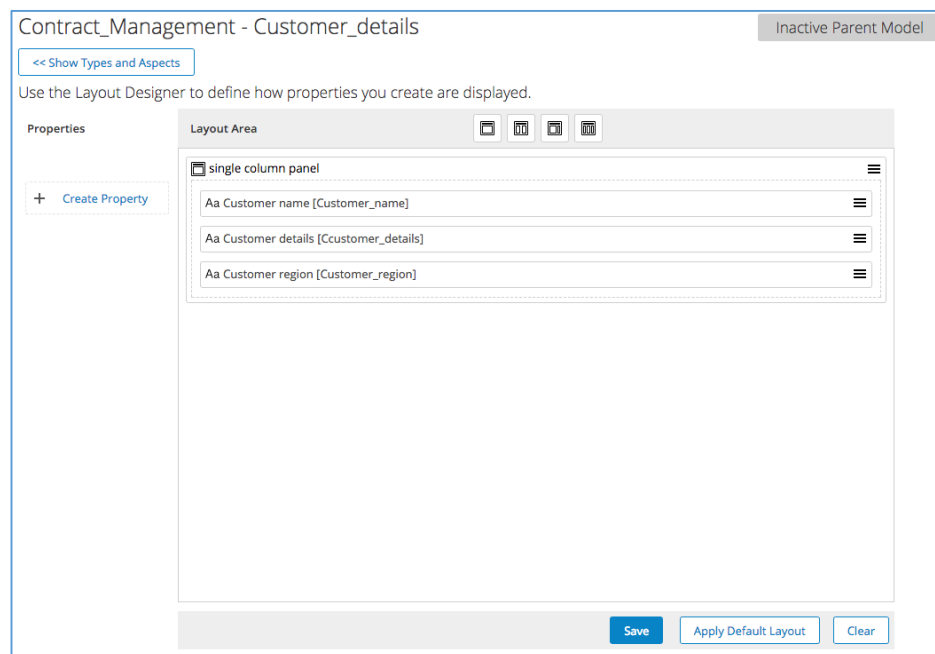
[Save](#) [Apply Default Layout](#) [Clear](#)

3. Drag the  first (**single column panel**) of the layout options into the area below.

4. Drag and drop the 3 properties onto the layout area, as shown in the screenshot below.



5. Now click **Save**.
6. Now repeat these steps for the 'Customer' aspect.



7. The 'Contract' type includes standard attributes and the ones you have just created. We have included a layout for this with the zip file.

Contract_Management - Contract Inactive Parent Model

[<< Show Types and Aspects](#)

Use the Layout Designer to define how properties you create are displayed.

Properties

- Aa Author [cm:author]
- Taggable [cm:taggable]
- Categories [cm:categories]
- + Create Property

Layout Area

- double column panel
 - Aa Name [cm:name]
 - Aa Title [cm:title]
- single column panel
 - Aa Description [cm:description]
- double column panel
 - Aa Creator [cm:creator]
 - Created Date [cm:created]
 - Aa Modifier [cm:modifier]
 - Size [size]
 - Modified Date [cm:modified]
 - Mimetype [mimetype]
- Contract Details
 - Aa Contract type [Contract_type]
 - Contract number [Contract_number]
 - Contract date [Contract_date]
 - Aa Contract comments [Contract_comments]
 - Contract value [Contract_value]

[Save](#) [Apply Default Layout](#) [Clear](#)

This sets how properties are displayed on forms. You can group properties and create a cleaner layout, so feel free to experiment.

4. Activating the model

The model is complete, so now you need to activate it so it can be used.

1. Go back to the Models screen.
2. Click **Actions** next to Contract_Management and select **Activate**.

Model Manager

Tools

- javascript Console
- Application
- Category Manager
- Cloud Sync Manager
- Module Browser
- Node Browser
- Search Manager
- Tag Manager
- Model Manager
- Sites Manager
- Mail Client
- Email Metadata Settings
- Email Integration Settings
- Email Access Tokens
- Email Licenses

Models are used to manage custom types and aspects.
You can create a new model, or select an existing model name, and create and view associated custom types and aspects.

[Create Model](#) [Import Model](#)

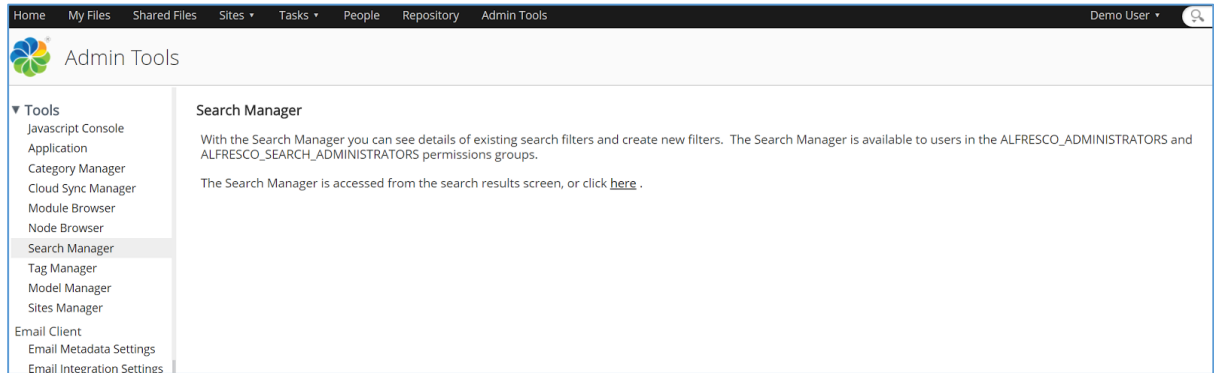
Models

| Name | Namespace | Status | Actions |
|---------------------------|--|----------|--------------------------------------|
| Contract_Management | contract_management | Inactive | Actions |
| dynamicSecurityMarksModel | http://www.alfresco.org/model/dynsecuritymarks/1.0 | Active | Activate Edit Delete Export |

5. Creating custom search filters

Before starting to set up the process, the last thing you want to do is set up search filter options for contract type, supplier type and customer region.

1. In the **Admin tools** select **Search Manager**, then click the **click here** link.



You'll see the following screen:

The screenshot shows the Alfresco Search Manager interface. The top navigation bar includes Home, My Files, Shared Files, Sites, Tasks, People, Repository, and Admin Tools. The user is logged in as Demo User. The main content area is titled 'Search Manager' and features a 'Create New Filter' button. Below the button is a table with the following data:

| | Filter ID | Filter Name | Filter Property | Filter Type | Show with Search Results | Default Filter | Filter Availability |
|----|-----------------|--|---------------------------------|---------------|--------------------------|----------------|---------------------|
| ▼ | filter_creator | faceted-search.facet-menu.facet.creator | cm:creator (Creator) | Simple Filter | Yes | Yes | Everywhere |
| ▲▲ | filter_mimetype | faceted-search.facet-menu.facet.formats | cm:content.mimetype (MIME type) | Simple Filter | Yes | Yes | Everywhere |
| ▲▲ | filter_modifier | faceted-search.facet-menu.facet.modifier | cm:modifier (Modifier) | Simple Filter | Yes | Yes | Everywhere |

2. Click **Create New Filter**, and add the following information, leaving other default settings as they are.

- Filter ID – **contract_type**
- Filter Name – **Contract Type**
- Filter Property – **cs:Contract_type (Contract Type)**
(click in the Filter Property field and type **cs** and your model properties should appear)

Tip: All the properties you added in the previous step start with the prefix 'cs'- so you can type 'cs' into the **Filter Property** field and the properties you added will be displayed.

Create New Filter ✕

Filter ID: *
The unique identifier for this search filter.

Filter Name: *
The displayed filter name.

Show with Search
Results: You can switch off this option to hide the filter from search results.

Default Filter:
Default filters are predefined and can't be deleted. You can hide them by switching off Show with Search Results.

Filter Property: *
Select a property to filter by.

Filter Type:
Select how the filter is shown.

Sort By:
Select the order the filter results are shown in.

Number of Filters:
Select the maximum number of filters shown for search results (users can select to show more filters).

Minimum Filter Length:
Select to show only filter results with a minimum number of characters so you can exclude short words such as "and" and "to" from filter results.

Minimum Required Results:
Select the minimum number of matches a filter result must have to be shown.

3. Click **Save**.

Now repeat this and add new search filters for 'Customer_region' and 'Supplier_status'. Make sure you change the **Filter Property** field to select the relevant property.

The Search Manager screen should now look like this:

| Search Manager | | | | | | | | |
|--|---------------------|--|--------------------------------------|---------------|--------------------------|----------------|---------------------|---|
| <input type="button" value="Create New Filter"/> | | | | | | | | |
| | Filter ID | Filter Name | Filter Property | Filter Type | Show with Search Results | Default Filter | Filter Availability | |
| ▼ | filter_creator | faceted-search.facet-menu.facet.creator | cm:creator (Creator) | Simple Filter | Yes | Yes | Everywhere | |
| ▲▼ | filter_mimetype | faceted-search.facet-menu.facet.formats | cm:content.mimetype (MIME type) | Simple Filter | Yes | Yes | Everywhere | |
| ▲▼ | filter_created | faceted-search.facet-menu.facet.created | cm:created (Created Date) | Simple Filter | Yes | Yes | Everywhere | |
| ▲▼ | filter_content_size | faceted-search.facet-menu.facet.size | cm:content.size (Size) | Simple Filter | Yes | Yes | Everywhere | |
| ▲▼ | filter_modifier | faceted-search.facet-menu.facet.modifier | cm:modifier (Modifier) | Simple Filter | Yes | Yes | Everywhere | |
| ▲▼ | filter_modified | faceted-search.facet-menu.facet.modified | cm:modified (Modified Date) | Simple Filter | Yes | Yes | Everywhere | |
| ▲▼ | Contract_Type | Contract Type | CS:Contract_type (Contract Type) | Simple Filter | Yes | No | Everywhere | ⊗ |
| ▲▼ | Customer_region | Customer Region | CS:Customer_region (Customer Region) | Simple Filter | Yes | No | Everywhere | ⊗ |
| ▲ | Supplier_status | Supplier Status | CS:Supplier_status (Supplier Status) | Simple Filter | Yes | No | Everywhere | ⊗ |

You can click the arrows to move your new properties up to the top of the list (optional).

6. Configuring the Alfresco Repository

The next steps take place in Alfresco Process Services.

1. Sign in to Alfresco Process Services at <http://192.168.10.10:9090/activiti-app> using **demo** as the username and **alfresco** as the password.
Note: You will see a red banner saying your license is about to expire, this is because you are within the 30 day trial period, it will not affect your trial.
2. Click **Identity Management**, then select **Tenants** then **Alfresco Repositories**.
3. Click the **+** Add button to create an Alfresco Repository connection with the following settings:
 - Name – **Alfresco Content Services**
 - Repository base url - <http://192.168.10.10:8080/alfresco>
 - Share base url - <http://192.168.10.10:8080/share>
 - Alfresco verions – **4.2 (or higher)**
 - Alfresco Share Connector – **selected**
 - Repository secret - **60c7404940bae16607a31ae0cb41a459**

Create Alfresco repository ×

Name

Alfresco tenant

Tenant name to use in Alfresco. When omitted, the default tenant (-default-) is assumed

Repository base url

Base URL for Alfresco repository install, for example: https://my-domain.com/alfresco

Share base url

Base URL for Alfresco Share install, for example: https://my-domain.com/share. Used to generate links to view content in an Alfresco repository.

Alfresco version

Select the version of your Alfresco repository.

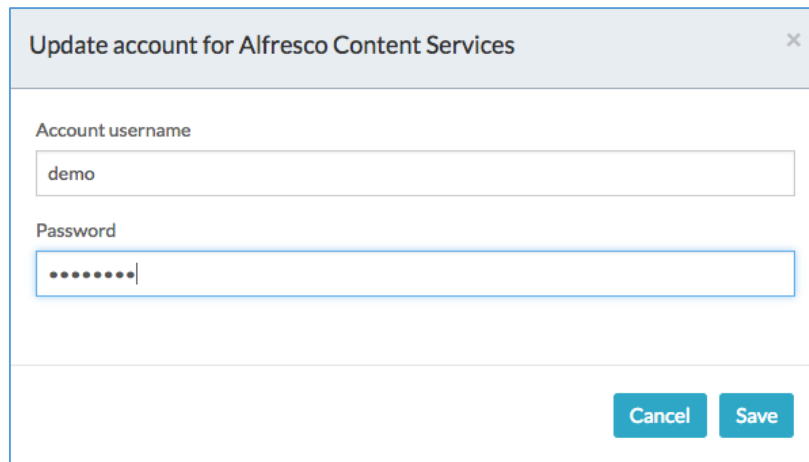
Alfresco Share Connector
Select if you are using the Alfresco Share connector

Repository secret

Specify if using the Alfresco Share connector. Will allow Alfresco Process Services to communicate with Alfresco without creating Alfresco user accounts in Alfresco Process Services for each user.

4. Click **Save** then select **Personal**.
5. Click **Alfresco Content Services** under Alfresco Repositories.

6. Enter **demo** as the username and **alfresco** as the password then click **Save**.



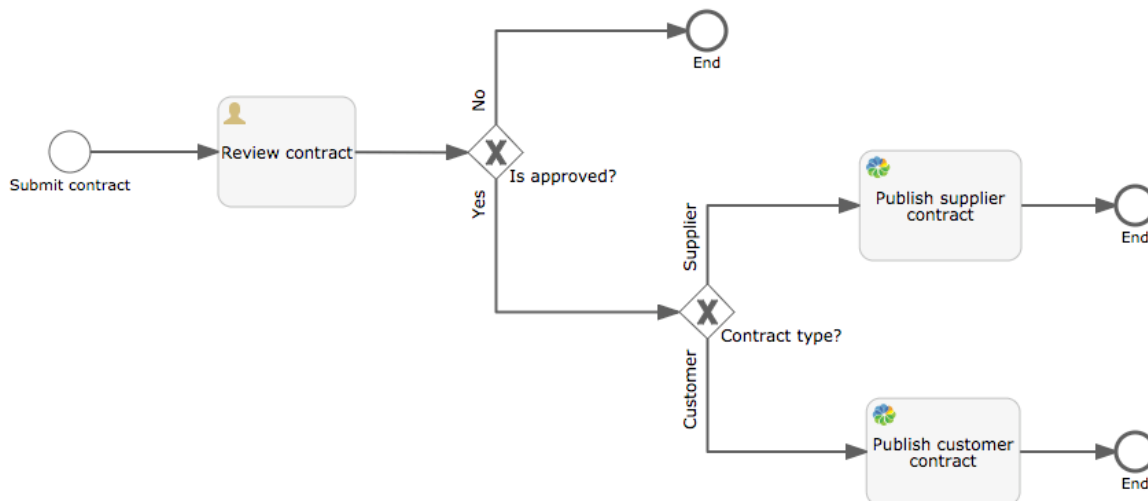
7. Designing a Contract Approval App


Now you can design a simple BPMN model and create an app to manage a contract approval process.

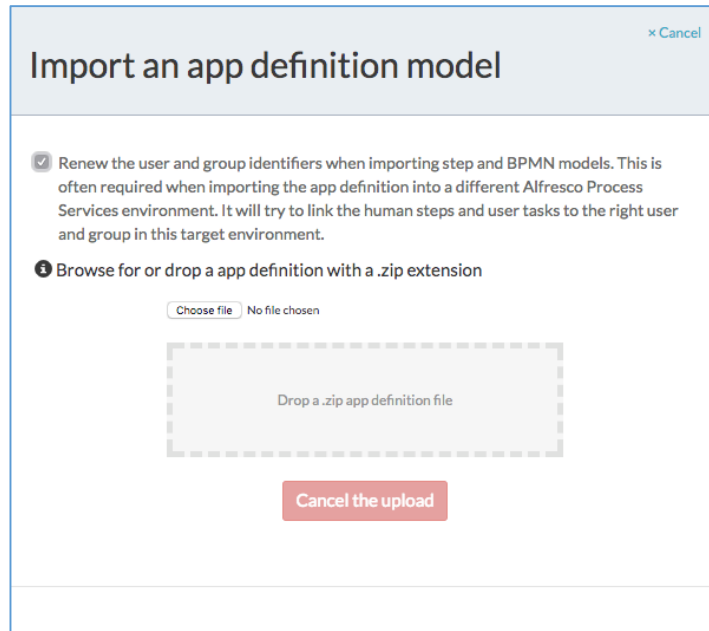
You can do this by uploading a zip, or by setting up the model yourself, see [Appendix 2](#).

The model includes a process initiator submitting a contract for approval, a reviewer deciding if the contract is valid or not, and an automatic task to store the contract in Alfresco Content Services. The final BPMN diagram will include the following BPMN elements:

- **Start event** to trigger the process by submitting a new contract for approval
- **User task** to review the submitted contract
- **Exclusive gateway** to split the workflow depending on the reviewer approval
- **Exclusive gateway** to split the workflow depending on the contract type
- Two **Publish to Alfresco** tasks to publish the approved contract in Alfresco Content Services and update the contract custom properties as well

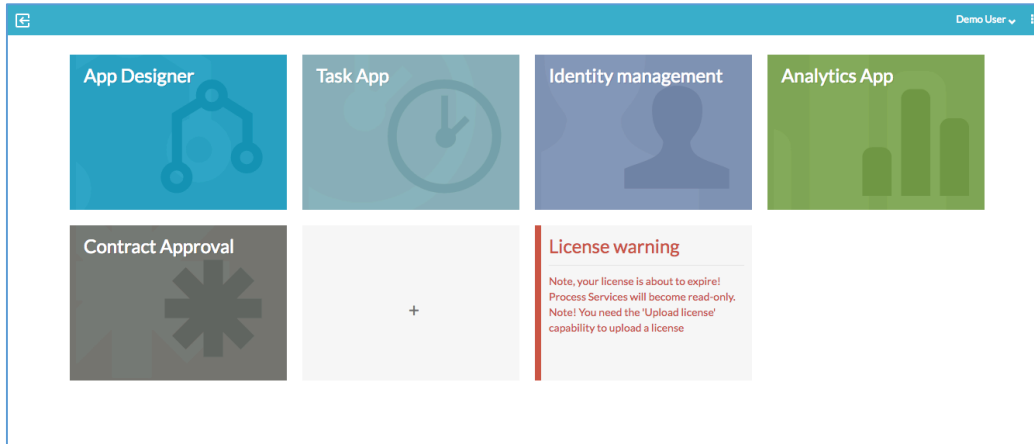


1. If you're not currently on the dashboard then click  Back.
2. Click **App Designer** select **Apps** then **Import App**.
3. Select the **Renew the user and group identifiers** check box



4. Choose the **ImportContractApprovalApp** zip file.
5. When it's uploaded select **Publish**. This will make the APP available on the dashboard.
6. Go back to your dashboard and click on the + tile and click **Deploy**. The app is added to your dashboard.

Note: Apps are a way of grouping related processes and making them available to the users.



8. Using the Contract Approval App

Once you've published the Contract Approval app, it can be used to submit a new contract for approval.

1. Click the **Contract Approval** app on your dashboard then click **Start**.
2. Complete the fields with suitable selections, selecting **Supplier** as the contract type.
3. Attach a MS Word or PDF document as the contract
4. Click **Start Process** to submit a new contract for approval.

Contract approval - July 10th 2017

Contract details

Contract number: 235235
Contract date (D-M-YYYY): 10-7-2017
Upload contract: SDO-250-2017-1001415447.pdf
Contract value: € 50000.00
Contract type: Supplier

Supplier details

Supplier name: ACME
Supplier reference number: 43334
Supplier status: Approved

START PROCESS

5. Click **Review Contract** task under the **TASKS** section.
The contract is displayed for you to review.

Review contract

Assignee: Demo User | Due: No due date | Part of process: Contract approval - August 16th 2017

No people involved | No groups involved | No content items | No comments | No checklist | SHOW DETAILS

SAVE | REJECT | APPROVE

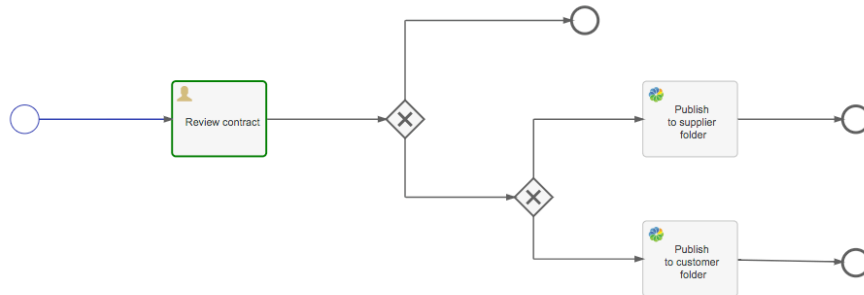
Contract details

Contract Number: 1
Contract Date (D-M-YYYY): 16-8-2017
Contract Value: \$ 1000.00
Contract: DBP Sandbox Using Guide - 1
Contract Type: Supplier

Supplier details

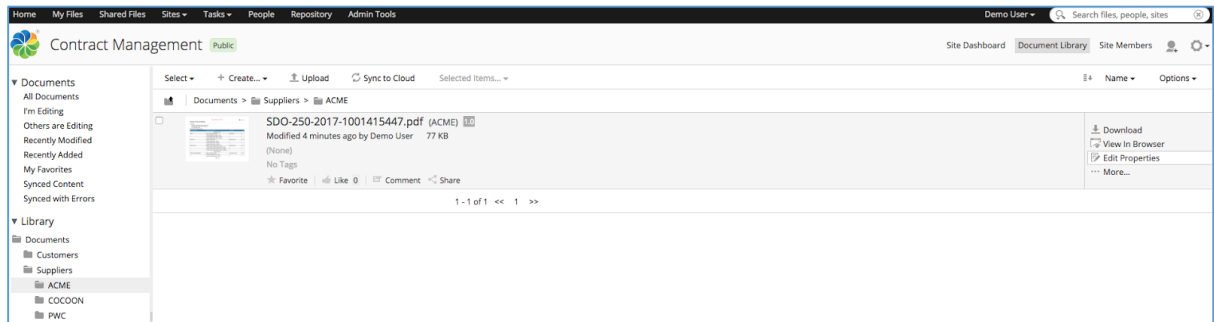
Supplier Name: Acme
Supplier Reference Number: 1
Supplier Status: Approved

6. Click **Processes**.
The **Contract Approval** process is shown in the Running processes section.
7. Click **Show Diagram** to see the current status of the contract approval request. The active task is highlighted in green to indicate it's at review stage.



CLOSE

8. Click **Close** to go back to the task.
9. Click **Review contract** then click **Approve**.
10. Now sign in to **Alfresco Content Services**.
11. Select Sites then **Contract Management**.
12. In the Document Library open the Supplier folder and you'll see that the supplier contract you just approved has been added.
13. Click on the contract to see a preview, and in the Properties section all the custom metadata from the custom model created earlier is displayed.



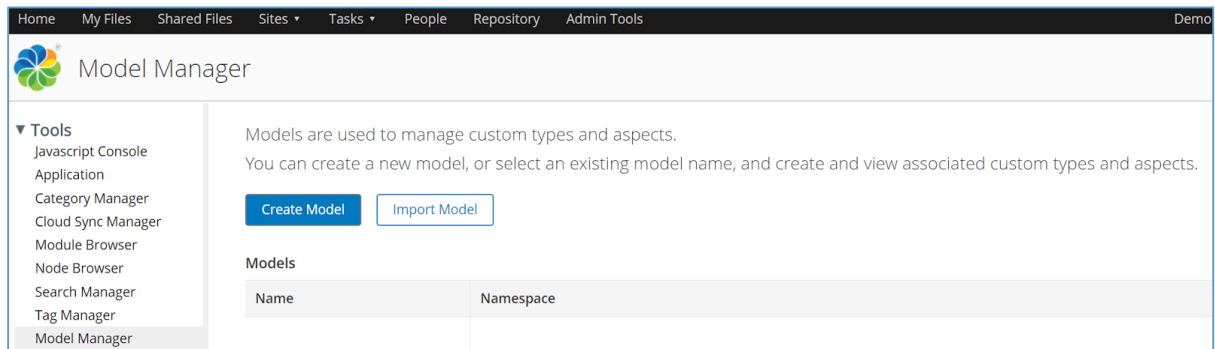
Tip: If you type the file name into the search box and press enter, you'll see that the search filters you set up earlier are available.

Appendix

If you prefer to set up your content model and process model manually, then follow the sections below.

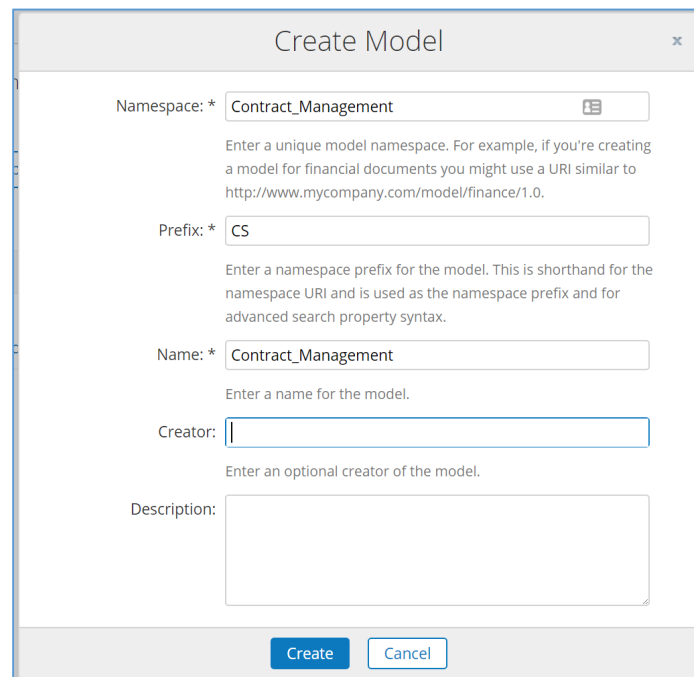
1. Manually creating content models

1. Click **Admin tools** on the toolbar, then select **Model Manager** from the options.



2. Next click **Create Model**.
3. Fill out the form as shown below and click **Create**.

- Namespace: **Contract_Management**
- Prefix: **CS**
- Name: **Contract_Management**



Namespace: * ⓘ

Enter a unique model namespace. For example, if you're creating a model for financial documents you might use a URI similar to <http://www.mycompany.com/model/finance/1.0>.

Prefix: *

Enter a namespace prefix for the model. This is shorthand for the namespace URI and is used as the namespace prefix and for advanced search property syntax.

Name: *

Enter a name for the model.

Creator:

Enter an optional creator of the model.

Description:

Having created a model you can now add *types* and *aspects* to it.

- **Types** - A custom type enumerates the properties and relationships that a file of that type can support

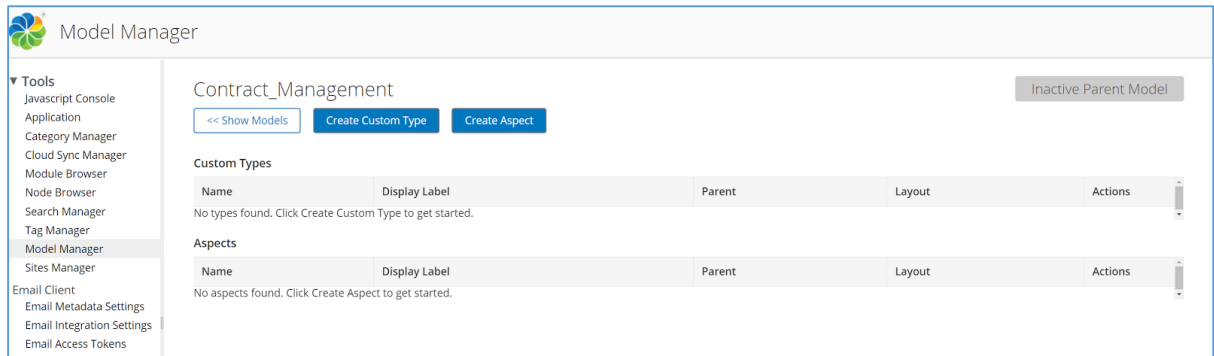
- **Aspects** - An aspect is a collection of properties that can encapsulate both data and behaviour

You can read more about this in [Content modeling with Model Manager](#).

We'll create:

- One type - This will define the basic contract information to be stored as metadata with every contract document.
- Two aspects - These will define the metadata that will be added to files depending on the contract type ("supplier" or "customer" in this example)

4. Click on the Contract_Management model name then click **Create Custom Type**.



5. Enter the details as shown below and click **Create**.

- Name: **Contract**
- Parent Type: **CM:content (Content)**
- Display Label: **Contract**

The 'Create Custom Type' dialog box is shown with the following fields and values:

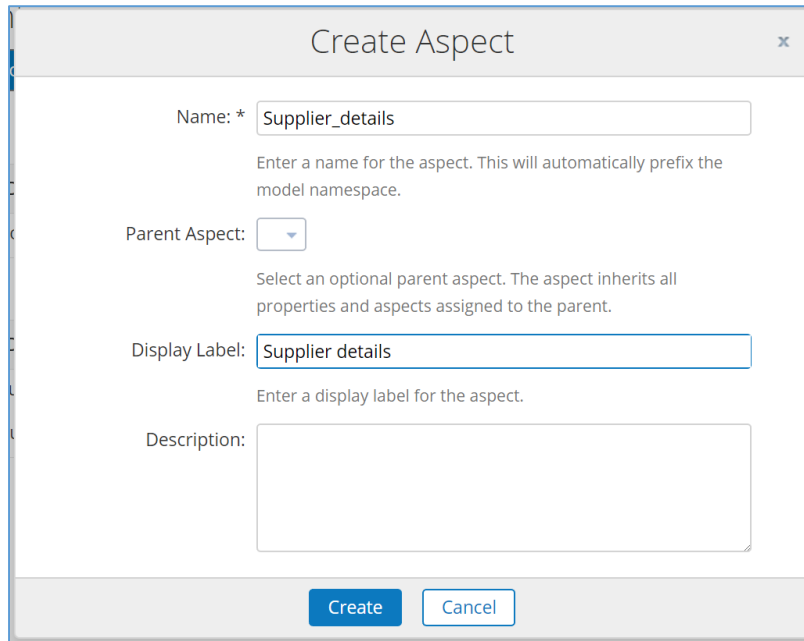
- Name:** * Contract
- Parent Type:** cm:content (Content)
- Display Label:** Contract
- Description:** (Empty text area)

Buttons for 'Create' and 'Cancel' are located at the bottom of the dialog.

The next step is to create two aspects.

6. Click **Create Aspect**, enter the details shown below then click **Create**:

- Name: **Supplier_details**
- Display Label: **Supplier details**



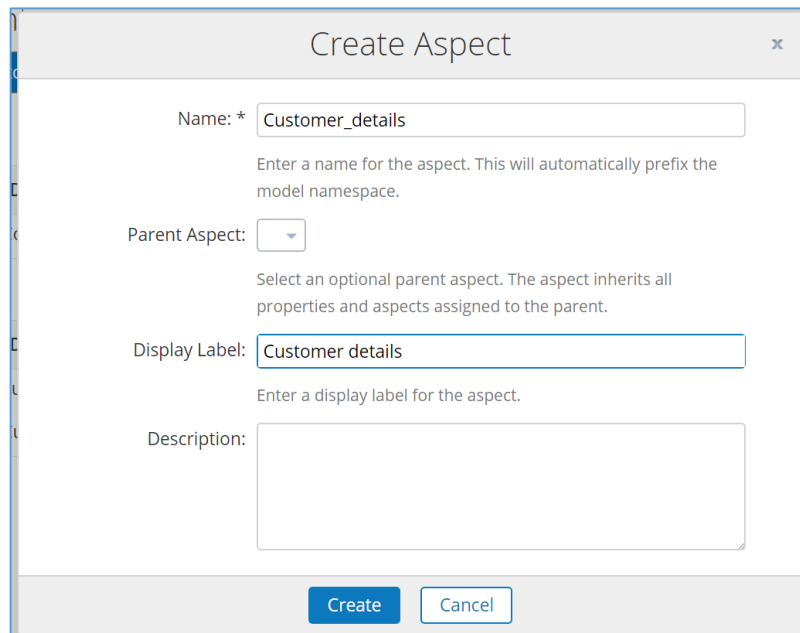
The screenshot shows a 'Create Aspect' dialog box with the following fields and values:

- Name:** * Supplier_details
- Parent Aspect:** (empty dropdown)
- Display Label:** Supplier details
- Description:** (empty text area)

Buttons at the bottom: **Create** and **Cancel**.

7. Now create another aspect, using the following details:

- Name: **Customer_details**
- Display Label: **Customer details**



The screenshot shows a 'Create Aspect' dialog box with the following fields and values:

- Name:** * Customer_details
- Parent Aspect:** (empty dropdown)
- Display Label:** Customer details
- Description:** (empty text area)

Buttons at the bottom: **Create** and **Cancel**.

When you've created these two aspects the contracts model will look like this:

Contract_Management Inactive Parent Model

<< Show Models
Create Custom Type
Create Aspect

Custom Types

| Name | Display Label | Parent | Layout | Actions |
|-------------|---------------|------------|--------|-----------|
| CS:Contract | Contract | cm:content | No | Actions ▾ |

Aspects

| Name | Display Label | Parent | Layout | Actions |
|---------------------|------------------|--------|--------|-----------|
| CS:Supplier_details | Supplier details | | No | Actions ▾ |
| CS:Customer_details | Customer details | | No | Actions ▾ |

Now it's time to add some properties, starting with properties within the type CS:Contract.

8. Click **cs:Contract**.
Now you can add multiple properties to the Type.
9. Click **Create Property** and create a property using the details in the first row of the table below, leaving other fields at the default setting.
When you're done, click **Create and Start Another**, and repeat until you've created properties for each table row. When you've entered details of the final row, click **Create**.

| Name | Display Label | Data type | Constraint | Indexing |
|-------------------|-------------------|-----------|--|--------------------------------|
| Contract_value | Contract Value | d:long | None | Basic |
| Contract_number | Contract Number | d:int | None | Basic |
| Contract_comments | Contract Comments | d:mltext | None | Free Text |
| Contract_date | Contract Date | d:date | None | Basic |
| Contract_type | Contract Type | d:text | List of Values -Supplier -Customer | List of Values – Partial Match |

When you're done, the properties table should look as shown below (the property order might be different).

Contract_Management - Contract Inactive Parent Model

[<< Show Types and Aspects](#) [Create Property](#)

Properties

| Name | Display Label | Data Type | Requirement | Default Value | Multivalued | Actions |
|----------------------|-------------------|-----------|-------------|---------------|-------------|-----------|
| cs:Contract_value | Contract value | d:long | Optional | | No | Actions ▾ |
| cs:Contract_number | Contract number | d:int | Optional | | No | Actions ▾ |
| cs:Contract_comments | Contract comments | d:mtext | Optional | | No | Actions ▾ |
| cs:Contract_date | Contract date | d:date | Optional | | No | Actions ▾ |
| cs:Contract_type | Contract type | d:text | Optional | | No | Actions ▾ |

Now that the type's properties are set up, you can do the same for the aspects.

- Click **Show Types and Aspects** then click on **CS:Supplier_detail**.
- Click **Create Property** and create a property using the details in the first row of the table below, leaving other fields at the default setting.
When you're done, click **Create and Start Another**, and repeat until you've created properties for each table row. When you've entered details of the final row, click **Create**.

| Name | Display Label | Data type | Constraint | Indexing |
|-----------------|-----------------|-----------|-------------------------------------|--------------------------------|
| Supplier_name | Supplier Name | d:text | None | Free Text |
| Supplier_number | Supplier Number | d:int | None | Basic |
| Supplier_status | Supplier Status | d:text | List of Values -Approved -New | List of Values – Partial Match |

When you're done, the properties table should look as shown below (the property order might be different).

Contract_Management - Supplier_details Inactive Parent Model

[<< Show Types and Aspects](#) [Create Property](#)

Properties

| Name | Display Label | Data Type | Requirement | Default Value | Multivalued | Actions |
|--------------------|-----------------|-----------|-------------|---------------|-------------|-----------|
| CS:Supplier_number | Supplier Number | d:int | Optional | | No | Actions ▾ |
| CS:Supplier_name | Supplier Name | d:text | Optional | | No | Actions ▾ |
| CS:Supplier_status | Supplier Status | d:text | Optional | | No | Actions ▾ |

- When you're done click **Show Types and Aspects** then click the **CS:Customer_detail** aspect.

13. Click **Create Property** and create a property using the details in the first row of the table below, leaving other fields at the default setting. When you're done, click **Create and Start Another**, and repeat until you've created properties for each table row. When you've entered details of the final row, click **Create**.

| Name | Display Label | Data type | Constraint | Indexing |
|------------------|------------------|-----------|--|--------------------------------|
| Customer_name | Customer Name | d:text | None | Free Text |
| Customer_details | Customer Details | d:mltext | None | Free Text |
| Customer_region | Customer Region | d:text | List of Values - Europe - South America - North America - Africa - Asia | List of Values – Partial Match |

When you're done, the properties table should look as shown below (the property order might be different).

Contract_Management - Customer_details Inactive Parent Model

[<< Show Types and Aspects](#) [Create Property](#)

Properties

| Name | Display Label | Data Type | Requirement | Default Value | Multivalued | Actions |
|---------------------|------------------|-----------|-------------|---------------|-------------|-----------|
| CS:Customer_name | Customer Name | d:text | Optional | | No | Actions ▾ |
| CS:Customer_region | Customer Region | d:text | Optional | | No | Actions ▾ |
| CS:Customer_details | Customer Details | d:mltext | Optional | | No | Actions ▾ |

2. Manually designing a process model

1. Select **App Designer** on the Alfresco Process Services dashboard and click **Create Process**.
2. Enter the following details then click **Create new model**:
 - Model Name – **Contract Approval**
 - Editor Type – **BPMN editor**
 - Stencil – **Default BPMN**

Create a new business process model

You need to give a name for the new model and you may want to add a description at the same time.

Model name

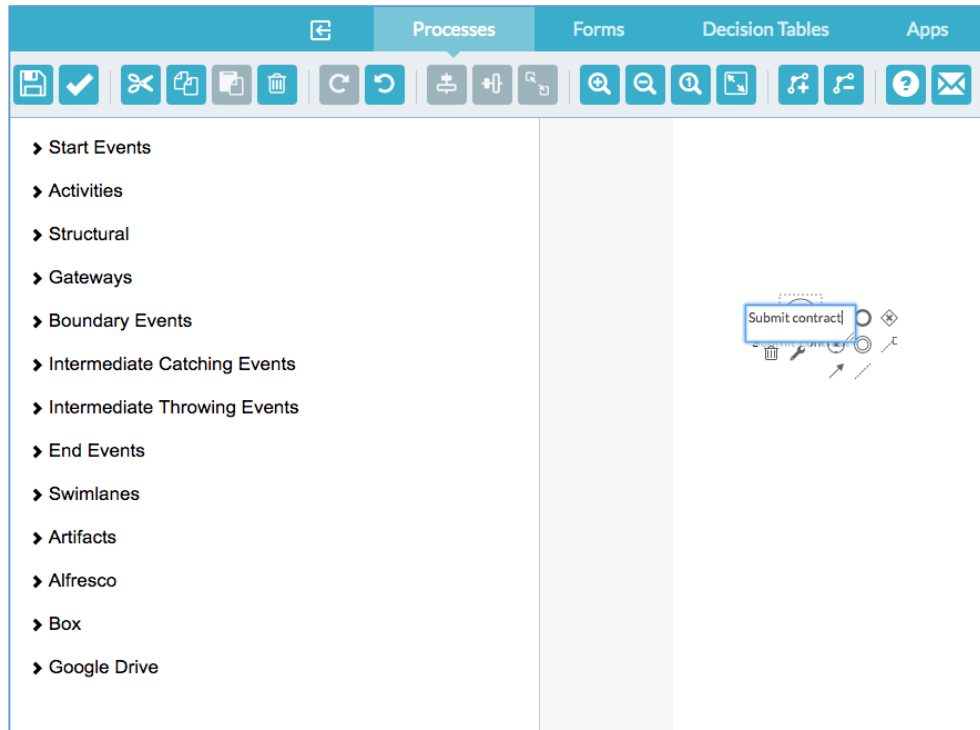
Description


Editor type

Stencil

Tip: If this is the first time you have used the BPMN modeler, a tutorial will be displayed. You can watch this or close it.

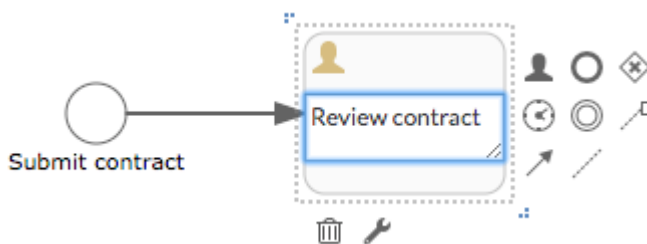
3. Double-click on the **Start event** (displayed as a circle) and type **Submit contract**.





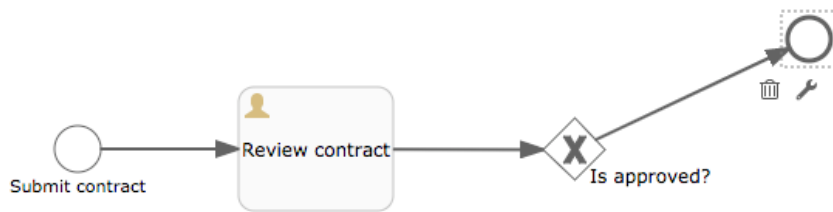
7. Click the **Start event** element and select the  User task icon and drag and drop it to the right. It will automatically add a sequence flow with the user task connected to it.



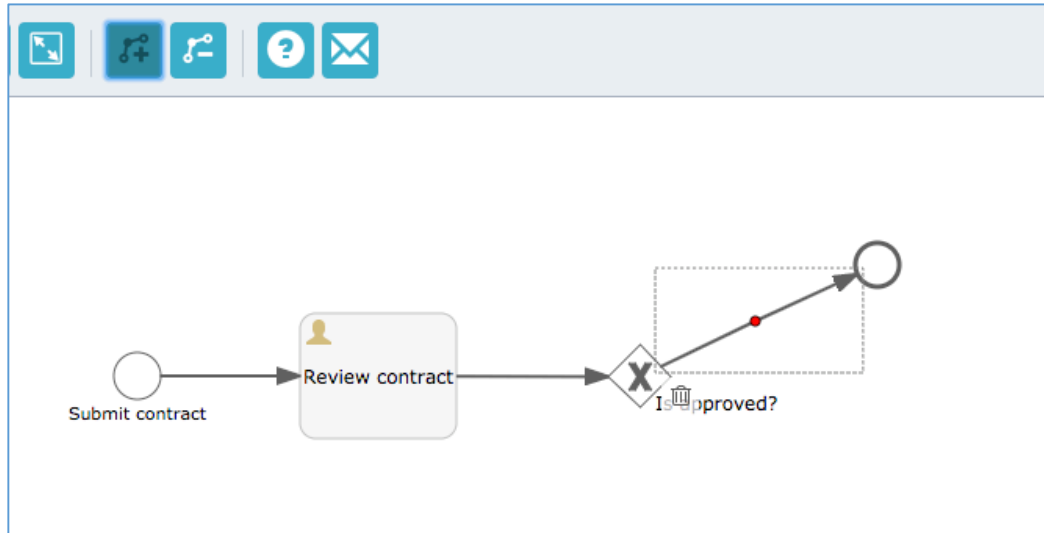
8. Double-click the task element and type **Review contract**.



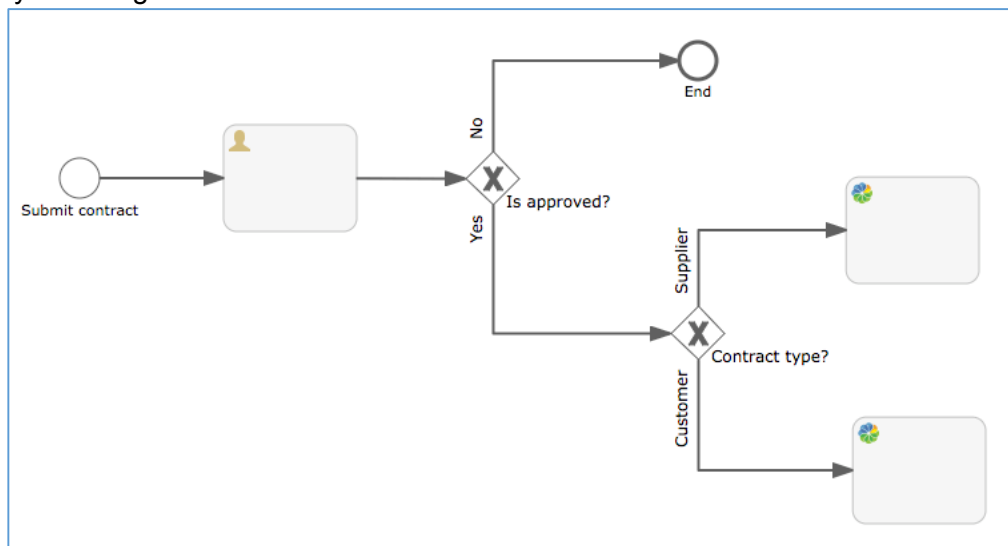
9. Click the task element and select the  **Gateway** icon and drag and drop it to the right.
10. Double-click it and type **Is approved?**.
11. Click on the element and drag and drop an  End event diagonally up and right.





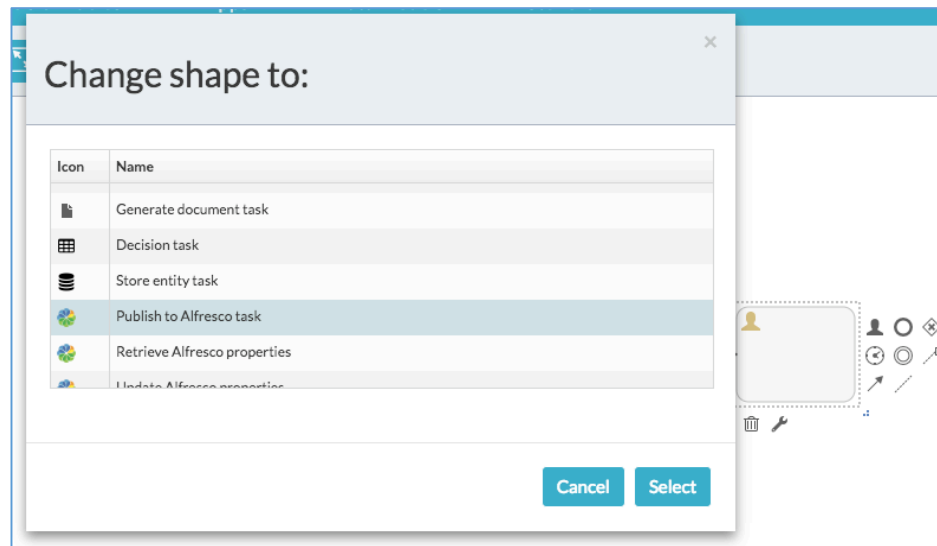
Tip: To bend a sequence flow, just select it and click on the appropriate bending function as shown here.



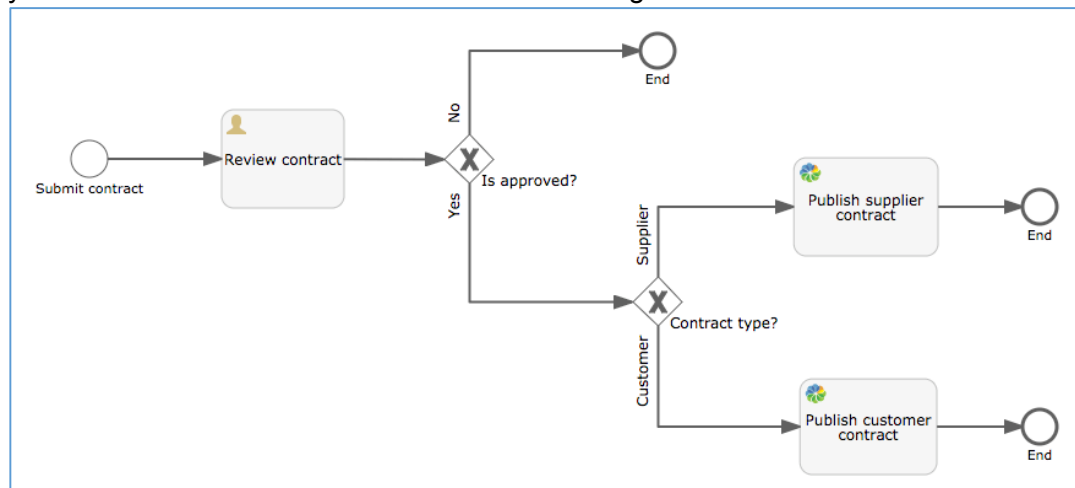
12. Continue your design as shown below.



Tip: To add a **Publish to Alfresco** task, just drag and drop a  User task and click on the  tool icon to change its type.





When you're done the model should look as in the image below.



2.1 Designing a form to submit a contract

Next you need to design the form for the process initiator to submit a contract for approval.

1. Click the **Start event** of your process model (the one called Submit Contract).
2. On the properties panel at the bottom, click on the **Referenced form** property.
3. Click **New form** and name it **Contract submission**, then click **Create form**.
4. From the form controls on the left side drag and drop a **Header** onto the design area.
5. Hover over the **Label** and click the Edit icon.
6. Type **Contract details** as the Label and click **Close**.
7. Complete the form by adding more fields by dragging and dropping the following stencils onto the form:
 - **Number** named **Contract number**
 - **Date** named **Contract date**
 - **Amount** named **Contract value** (you also need to click the **Amount** tab after you've entered the name and enter your preferred currency, for example \$ or £)
 - **Attach File** named **Upload contract** (you also need to click the **Options** tab after you've entered the name and add two options, Supplier and Customer)
 - **Dropdown** named **Contract type**

8. Add another **Header** and click the  Edit icon.
9. Type **Supplier details** as the Label.
10. Click the **Visibility** tab then click  Add.
11. Select **Contract type – label**, in the first field, **equal** in the second, and type **Supplier** in the third, then click **Close**.

| Depends on | If it's | Value | Next condition operator |
|-------------------|---------|----------|-------------------------|
| contracttype_L... | equal | Supplier | |

Depends on

Form field | **Variable**

Contract type - label - contracttype_LABEL - dropdown

If it's

equal

Value | **Form field** | **Variable**

Supplier

Next condition operator

Close

12. Add a **Text**, a **Number** and a **Dropdown** control inside the **Supplier details** header as shown below.

13. Edit the Dropdown list as shown below.

- 14. Add another **Header** and name it **Customer details**.
- 15. Edit the Header properties to make it only visible when “Customer” is the selected option of the “Contract details” **Dropdown** list.

| Depends on | If it's | Value | Next condition operator |
|-------------------|---------|----------|-------------------------|
| contracttype_L... | equal | Customer | |

16. Add a **Text** and a **Dropdown** control as shown below.

17. Edit the **Dropdown** with the following regions:

- Europe
- South America
- North America
- Africa
- Asia

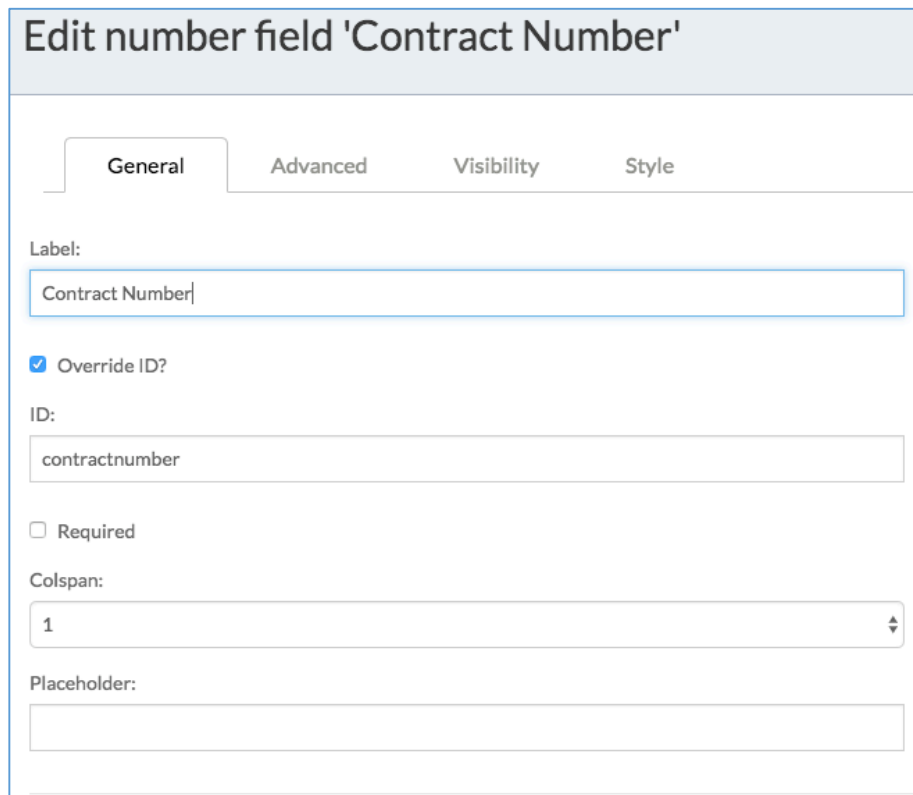
18. When you done click  Save to save the form.

2.2 Designing a form to review contracts

Next you can design the form for the reviewer to review the contract submitted by the process initiator.

This form is very similar to the previous one. You need to drag and drop three **Headers** corresponding to the 3 sections: “Contract details”, “Supplier details” and “Customer details”.

So the reviewer can visualize and possibly modify the previously submitting information, you need to apply the same ID that you defined for each of the fields. For example, here is the configuration to display the “Contract number” and make it editable for the reviewer.



The screenshot shows a configuration window titled "Edit number field 'Contract Number'". It has four tabs: "General", "Advanced", "Visibility", and "Style". The "General" tab is active. The configuration includes the following fields:

- Label:** A text input field containing "Contract Number".
- Override ID?** A checked checkbox.
- ID:** A text input field containing "contractnumber".
- Required:** An unchecked checkbox.
- Colspan:** A dropdown menu set to "1".
- Placeholder:** An empty text input field.

Here is the “Contract details” header for the reviewer.

Contract details

Contract Number

Contract Value

Contract Type

Contract Date

Show value of "Upload contract"

Use the details below.

| Label | ID | Type |
|---------------------------|-------------------------|---|
| Contract Number | contractnumber | Number |
| Contract Date | contractdate | Date |
| Contract Value | contractvalue | Amount |
| Upload Contract | uploadcontract | Display Value |
| Contract Type | contracttype | Dropdown (Supplier/Customer) |
| Supplier Name | suppliername | Text |
| Supplier Reference Number | supplierreferencenumber | Number |
| Supplier Status | supplierstatus | Dropdown (New/Approved) |
| Customer Name | customername | Text |
| Region | region | Dropdown (Europe/South America/North America/Africa/Asia) |

To display the contract document previously attached, you need to use the **Display value** stencil and configure it as described here.

Edit display value field 'Contract'

General Visibility Style

Label:

Form field Variable

Upload contract - uploadcontract - upload

Allow editing documents in Office

Show document content embedded in form

Colspan:

Continue the design by adding the supplier and customer sections. Then add an additional section (header called Comments) and add a Multiline text for the reviewer to leave comment.

Supplier Details

| | |
|---|---|
| Supplier Name <input type="text" value="Text"/> | Supplier Reference Number <input type="text" value="123"/> |
| Supplier Status <input type="text" value="Select..."/> | |

Customer Details

| | |
|--|--|
| Customer Name <input type="text" value="Text"/> | Region <input type="text" value="Select..."/> |
|--|--|

Comments

Reviewers Comments

Configure the visibility of the supplier and customer section as we did for the previous form. Select the **Outcome** tab and add two custom outcomes: "Reject" and "Approve".

Design Tabs **Outcomes** Style Javascript Properties

You can define multiple outcomes for this task. When completing a task, the user selects one of the available outcomes that can be used further on in the process, such as in a condition.

Don't use custom outcomes, only show a 'Complete' button.

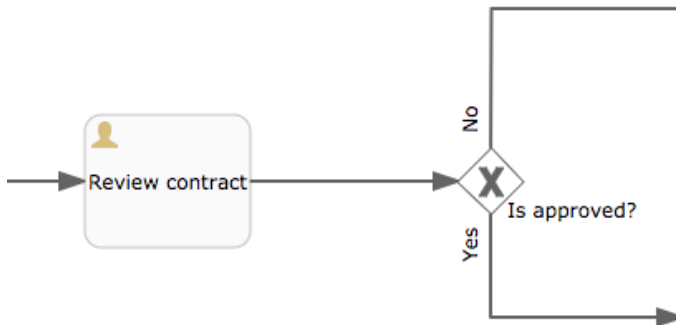
Use custom outcomes for this form.

Possible outcomes

| | |
|---------|--|
| Reject | |
| Approve | Remove Add outcome |

2.3. Configuring the sequence flow conditions based on contract review

Now that you've designed the form for the contract reviewer, you can define the conditions for the sequence flows exiting the exclusive gateway.



1. Click on the sequence flow for the rejected path (“No”).
2. On the properties panel at the bottom, click on the **Flow Condition** property.
3. Click **Simple**, then **Form Outcome** and enter the following details:
 - Customer contract review
 - equal
 - Reject

Sequence flow condition

Condition type: No condition Simple Advanced Expression

Depends on: Form field Form outcome Variable

4. Select the sequence flow for the approved path (“Yes”) and click the **Flow Condition** property.

5. Click **Simple**, then **Form Outcome** and enter the following details:

- Customer contract review
- equal
- Approve

Sequence flow condition

Condition type: No condition, **Simple**, Advanced, Expression

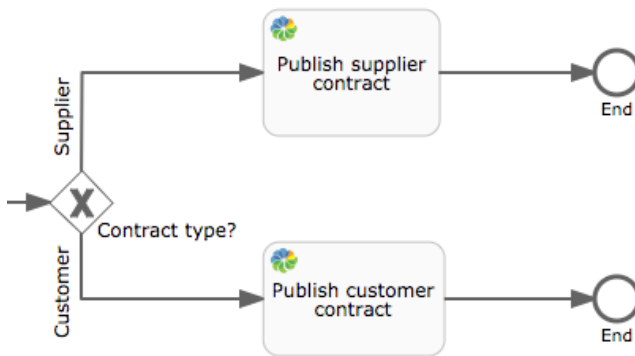
Depends on: Form field, **Form outcome**, Variable

Customer contract review equal Approve

Cancel Save

2.4 Configuring the sequence flow conditions based on contract type

To be able to store the approved contract in the correct folder according to its type, you need to configure the two sequence flows exiting the “Contract type?” gateway, based on the selected contract type.



1. Select the sequence flow for the Supplier path and click the **Flow Condition** property.
2. Click **Simple**, then **Form Outcome** and enter the following details:
 - Contracttype_LABEL
 - equal
 - Supplier

Sequence flow condition

Condition type: No condition, **Simple**, Advanced, Expression

Depends on: Form field, **Form outcome**, Variable

contracttype_LABEL equal Supplier

3. Repeat this for the Customer path, but using **Customer** as the equal to value.

2.5 Configuring the *Publish to Alfresco* tasks

1. Select the sequence flow for the Publish customer contract task and click the **Alfresco Destination** property.
2. Define the settings as shown below.

The screenshot shows a dialog box titled "Change value for 'Alfresco destination'". It contains the following fields and options:

- Account:** A dropdown menu with "alfresco-1" selected.
- Destination:** A field showing "Site: Contract management" and "Path: documentLibrary > Contracts > Customers". A "Select folder..." button is to the right.
- Publish as:** Two buttons: "Process initiator" (selected) and "Specific user".
- Subfolder:** A checkbox labeled "Create or reuse subfolder" which is checked.
- Based on field:** A text input field containing "Customer name - customername - text".

At the bottom right, there are "Cancel" and "Save" buttons.

3. Now select the **Alfresco content** property and select Publish content uploaded in field.
4. Complete the details as shown below. Copy and paste the name of the content model ("cs:Contract") you defined in Alfresco Content Services and add the "Customer details" aspect.

The screenshot shows a dialog box titled "Change value for 'Alfresco content'". It contains the following fields and options:

- Publish content uploaded in field:** A text input field.
- Value type:** Two buttons: "Form field" (selected) and "Variable".
- Form field:** A text input field containing "Upload contract - uploadcontract - upload".
- Document definition:** A tabbed interface with "Document definition" selected and "Properties mapping" as an alternative tab.
- Type:** A text input field containing "cs:Contract".
- Aspects:** A list box containing "cs:Customer_details". To the right, an "Aspect Name" text input field also contains "cs:Customer_details".

At the bottom right, there are "Cancel" and "Save" buttons.

5. When you're done click the **Properties mapping** tab.
You can now map the Customer details aspect properties and Contract details properties with

the corresponding form fields so that these properties values are stored in Alfresco Content Services.

Change value for "Alfresco content" ✕

Publish content uploaded in field

Value type:

Form field
Variable

Form field:

Document definition
Properties mapping

Properties:

| File Property | Form Field | Variable |
|--------------------|-------------------------|----------|
| cs:Customer_region | Customer region - label | |
| cs:Customer_name | Customer name | |
| cs:Contract_value | Contract value | |
| cs:Contract_number | Contract number | |
| cs:Contract_date | Contract date | |
| cs:Contract_type | Contract type - label | |

File Property

Property Type

Mapping Type:

Form field
Variable

+ -

Cancel
Save

Use the details below.

| File property | Form field |
|----------------------|-------------------------|
| cs:Customer_region | Customer region - label |
| cs:Customer_name | Customer name |
| cs:Contract_value | Customer value |
| cs:Contract_number | Contract number |
| cs:Contract_date | Contract date |
| cs:Contract_type | Contract type - label |
| cs:Contract_comments | Reviewer Comments |
| cm:title | Customer name |

- Now select the "Publish supplier contract" task and configure the destination folder to be the Supplier folder, and the subfolder name based on **Supplier name** form field.

Change value for "Alfresco destination"

Account:

Destination: Site: Contract Management
Path: documentLibrary > Suppliers Select folder...

Publish as: Process initiator Specific user

Subfolder: Create or reuse subfolder

Based on field:

7. Next, select the **Alfresco content** property and use the **Document definition** configuration shown below.

Publish content uploaded in field

Value type:

Form field Variable

Form field:

Document definition Properties mapping

Type:

Aspects:

| Aspect Name | |
|---------------------|--|
| cs:Supplier_details | |

No property selected

+ -

8. Finally, select the **Properties mapping** tab and configure the mapping as shown in the table below.

| File property | Form field |
|--------------------|---------------------------|
| cs:Supplier_status | Supplier status - label |
| cs:Supplier_number | Supplier Reference Number |
| cs:Supplier_name | Supplier name |
| cs:Supplier_number | Supplier reference number |

| | |
|----------------------|-----------------------|
| cs:Contract_value | Contract value |
| cs:Contract_number | Contract number |
| cs:Contract_date | Contract date |
| cs:Contract_type | Contract type - label |
| cs:Contract_comments | Reviewers Comments |
| cm:title | Supplier name |

Alfresco Copyright

©2017 Alfresco Software, Inc. All Rights Reserved.