d.velop

d.velop connect for Salesforce
CRM: Administrator

Table of Contents

1. d.velop connect for Salesforce CRM: Administrator

1.1. Basic information on the application

In this chapter, you can find notes on the product and general information.

1.1.1. About d.velop connect for Salesforce CRM

d.velop connect for Salesforce CRM is an innovative integrating application that seamlessly adds Salesforce to your CRM system and enriches it with information from your ECM system (SharePoint or d.velop documents). It displays documents from the ECM system in standard or custom Salesforce objects based on the specific context.

1.2. Installing and uninstalling the solution

This section provides you with information about installing d.velop connect for Salesforce CRM.

1.2.1. System requirements

The following requirements apply for Salesforce:

- To experience the full functionality and optimum visual display, you require Salesforce Lightning Experience. If you want to use Salesforce Classic, get in touch with your d.velop contact person for more information and an estimate of cost.
- We recommend using Google Chrome as the browser for Salesforce.
- **My Domain** must be activated for the Salesforce instance.
- All users of the d.velop Lightning component require a license.

The following requirements apply for the d.velop documents infrastructure (on-premises):

To connect a d.velop documents system (on-premises), you must ensure that the Salesforce app is given access to specific components in your infrastructure. There are also minimum requirements with regard to the versions of your components.

Minimum requirements for software and versions:

- Internet Information Services (IIS)
- d.3 server: version 8.1 or higher
- d.3 logview: the appropriate version for the version of d.3 server
- d.3 search: version 3.0.0, hotfix 6 or higher
- d.3 presentation server: compatible with the version of d.3 server but at least version 1.2.4
- d.ecs pdf extension: version 1.4.3 or higher
- d.3 webservice: version 2.3 or higher
- d.ecs jstore: version 1.1 or higher
- d.ecs http gateway: version 1.1 or higher
- d.ecs identity provider: version 2.0 or higher
- d.ecs shell: version 7.1 or higher
- d.3one: version 1.7.0 or higher
- Microsoft .NET Framework: Version 4.5.2
- Microsoft URL Rewrite module: version 2.0

The following requirements apply for the d.velop documents infrastructure (cloud):

We provide all the infrastructure components that you require to use d.velop cloud.

1.2.2. Installing d.velop connect for Salesforce CRM for d.velop documents on-premises

You wish to install d.velop connect for Salesforce CRM with the connection to d.velop documents on-premises. In the case of on-premises systems, Salesforce and the Salesforce connector must be connected to the d.3one application server. Prepare your infrastructure accordingly. You can obtain more information from Salesforce.

In order to use d.velop connect for Salesforce CRM, you must register the Salesforce app in d.ecs http gateway.

In some cases you may need to register apps manually, as some apps cannot be registered automatically. These include, for example, older apps, third-party apps or apps that were not registered properly.

In this case or similar ones, you will be able to register the app yourself.

This is how it works

1. Open the d.ecs http gateway configuration screen and select **Apps**.
2. Click **Add new app registration** and enter the information required to register an app instance.
3. Click **Register**.

Specify the following properties when manually registering an app:

- **App**: Required. Under **App** enter the name of the app to which the new app instance will be added.
- **Destination URL**: Required. Under **Target URL** enter the URL with which the new app instance can be accessed.
- **Instance ID**: Optional. You can enter an ID under **Instance ID**. If you do not enter an ID, one will be generated by d.ecs http gateway.
- **Version**: Optional. You can enter a version number for your app under **Version**.
- **Non-sticky session**: Standard. With **Non-sticky session** you can indicate that queries are being routed to any given instances of an app.
- **Sticky session**: With **Sticky session**, you can indicate that a client PC will only be forwarded to the same app instance during a session. This enables you to prevent the app communication from being spread over various cluster nodes. Only use this option for apps that need this option.

Once you have completed the installation, continue with the configuration in Salesforce.

1.2.3. Installing d.velop connect for Salesforce CRM for d.velop documents

You wish to install d.velop connect for Salesforce CRM with the connection to d.velop documents.

Open the d.velop store and add the **d.velop connect for Salesforce** app to your instance.

Once you have completed the installation, continue with the configuration in Salesforce.

1.2.4. Installing d.velop documents for Salesforce CRM

The following steps are required for the installation:

- Enabling Email-to-Case in Salesforce
- Installing the package for d.velop documents
- Assigning licenses
- Setting up remote sites
- Assigning permission sets
- Entering the user data of d.velop documents
- Setting up DMS mapping
- Placing the Lightning component

Enabling “Email-to-Case” in Salesforce

To use d.velop connect for Salesforce CRM, you require the Email-to-Case function (entering customer cases via e-mail). Enable the **Email-to-Case** function in the settings in Setup under **Feature Settings > Service > Email-to-Case**.

Installing the package for d.velop documents

You can install the **d.velop documents** package through a quick link or in Salesforce AppExchange. You receive the quick link from d.velop.

During the installation, you choose whether to install the package only for administrators, for specific profiles or for every user.

The installation may take several minutes.

Assigning licenses

You want to assign the licenses for d.velop connect for Salesforce CRM in Salesforce. You do not need to assign any licenses for installation in the sandbox.

This is how it works

1. Ensure that a sufficient number of licenses are available for your Salesforce users.
2. Navigate to **Setup > Apps > Packaging > Installed Packages** and assign the available licenses to the users.

Setting up remote sites

Salesforce prevents communication with third-party systems by default. You therefore have to create a remote site in order to connect Salesforce and d.velop documents.

You need to create one additional remote site each for the d.velop configuration object and for the associated web service.

This is how it works

1. Open **Setup > Security > Remote Site Settings**.
2. Click **New Remote Site**.
3. Create a new remote site with the public base URL of your d.velop documents instance.
4. Save your changes.

Assigning permission sets

Three tabs for Salesforce are provided with d.velop connect for Salesforce CRM. The tabs are the d.3one web application, the user settings and the configuration area. You assign the rights for these tabs using predefined permission sets.

The **d.velop documents standard user** permission set includes the right to access the d.velop documents system and the user settings. The **d.velop documents administration** permission set also includes the right to access the configuration area. Assign this permission set only to administrative users.

You assign the permission sets under **Setup > Users > Permission Sets**.Alternatively, you can also distribute the permission sets via the user profiles in the **Profiles** menu item.

Entering d.velop documents user data

You enter the user data in the **d.velop documents configuration** object. We recommend using a service user.

This is how it works

1. Open the system configuration.
2. Open the **Technical Information** menu item.
3. Enter the user data.

Setting up DMS mapping

The values entered in the Salesforce fields are automatically transferred to the attribute fields of the document types in d.velop documents by DMS mapping. The values are transferred when the documents are uploaded to Salesforce. You have to perform DMS mapping for every Salesforce object that you use.

First, carry out the mapping in d.velop documents. Then assign the document types in Salesforce and perform the object mapping.

This is how it works

1. To carry out the mapping in d.velop documents, open the d.velop documents system or the d.velop cloud instance as a user with administrative rights.
2. Click **Mappings**.
3. For the repository, create a new mapping for the source **Salesforce** or open an existing mapping with the name **Salesforce**.
4. Assign the transfer categories to the corresponding DMS categories.
5. Open Salesforce.
6. Assign the document types.
7. Perform the object mapping.

See also:

Assigning document types

Defining the object mapping

Placing the Lightning component

You want the documents in d.velop documents to be displayed in the Salesforce data records. Place the **D3Xocor** Lightning component on the Lightning pages for the object accounts and opportunities. You can obtain the component from d.velop.

This is how it works

1. Select the relevant Salesforce object.
2. Select the data record of the object.
3. Open **Setup > Edit Page**.
4. Select the **d.velop archive** component.
5. Place the component on the page.
6. Save your changes.

1.2.5. Installing d.velop connect for Salesforce CRM for Microsoft SharePoint

Before you can install Microsoft SharePoint, you must book and install the **d.velop connect for Microsoft SharePoint Online** adapter. You can book the adapter via the d.velop store.

The following steps are required for the installation:

- Configuring an OpenID Connect provider for productive instances or Configuring an OpenID Connect provider for sandbox instances
- Setting up the connected application in Salesforce
- Configuring external access in Salesforce

Configuring an OpenID Connect provider for productive instances

You can use OpenID Connect if you wish to use an external authentication server.

Contact the authentication server provider to obtain the data required to configure the OpenID Connect provider.

This is how it works

1. Open the d.ecs identity provider configuration.
2. Open the settings for the OpenID Connect provider using the gearwheel icon (**OpenID Connect settings**).
3. Add a user provider of the type **OpenID Connect** by clicking the plus sign (**Add OpenID Connect provider**).
4. Enter the required data.
5. Under **Description**, enter a brief description to enable your users to recognize the external service used when they log in.

Enter the following data to set up a productive instance in the OpenID Connect provider for Salesforce:

- **Description**: title shown during login (recommendation: **Salesforce**).
- **Enabled**: activated
- **Authorize endpoint**: https://login.salesforce.com/services/oauth2/authorize
- **Token endpoint**: https://login.salesforce.com/services/oauth2/token
- **UserInfo endpoint**: https://login.salesforce.com/services/oauth2/userinfo
- **Client ID**: taken from the connected application in Salesforce.
- **Client secret**: taken from the connected application in Salesforce.
- **Base URI**: https://openidgateway.service.d-velop.cloud/openidcallback
- **Redirect URI**: generated automatically and required for the connected application in Salesforce.

See also: Configuring an OpenID Connect provider for sandbox instances

Configuring an OpenID Connect provider for sandbox instances

You can use OpenID Connect if you wish to use an external authentication server.

Contact the authentication server provider to obtain the data required to configure the OpenID Connect provider.

This is how it works

1. Open the configuration of d.ecs identity provider.
2. Open the settings for the OpenID Connect provider using the gearwheel icon (**OpenID Connect settings**).
3. Add a user provider of the type **OpenID Connect** by clicking the plus sign (**Add OpenID Connect provider**).
4. Enter the required data.
5. Under **Description**, enter a brief description to enable your users to recognize the external service used when they log in.

Enter the following data to set up a sandbox instance in the OpenID Connect provider for Salesforce:

- **Description**: title shown during logon (recommendation: **Salesforce**).
- **Enabled**: activated
- **Authorize endpoint**: https://test.salesforce.com/services/oauth2/authorize
- **Token endpoint**: https://test.salesforce.com/services/oauth2/token
- **UserInfo endpoint**: https://test.salesforce.com/services/oauth2/userinfo
- **Client ID**: taken from the connected application in Salesforce.
- **Client secret**: taken from the connected application in Salesforce.
- **Base URI**: https://openidgateway.service.d-velop.cloud/openidcallback

- **Redirect URI**: generated automatically and required for the connected application in Salesforce.

See also: Configuring an OpenID Connect provider for productive instances

Setting up the connected application in Salesforce

You want to set up a new connected application in Salesforce.

This is how it works

1. Open **Setup > Apps > App Manager**.
2. Create a new connected application. You can find detailed instructions in the Salesforce documentation.

Configuring external access

You want to configure external access.

This is how it works

1. Open **System configuration > Add-On> External Access**.
2. Enter all the necessary data for the connected application.

1.2.6. Uninstalling d.velop documents for Salesforce CRM

You wish to remove d.velop connect for Salesforce CRM from Salesforce. To do so, you first have to remove any dependencies, such as components, triggers, processes, invoice readers, and so on. You can then uninstall the app.

This is how it works

1. Remove all the dependencies for the **d.velop documents for salesforce** package. The package has the type **Managed - Published** (Managed Package).
2. Open the Salesforce setup by clicking the gearwheel icon.
3. Open **Apps > Packaging > Installed Packages**.
4. Click **Uninstall**.

1.3. Configuring d.velop connect for Salesforce CRM

This section provides you with information about configuring d.velop connect for Salesforce CRM.

1.3.1. Updating d.velop documents for Salesforce

You want to update the d.velop documents for Salesforce application.

This is how it works

1. Open the d.velop documents configuration.
2. Open **Update**.
3. Click the link under **Installation URL** to install the latest version.

1.3.2. Defining the technical information

You wish to define the technical information in the system configuration.

This is how it works

1. Open the d.velop documents configuration.
2. Open **Technical Information**.
3. Enter the following data in the **d.velop cloud information** area:
   - Login name and password (API key) of the technical user
   - Target system (e.g. d.velop documents)

- Base address (base URL)
- ID of the target repository

1.3.3. Managing document types

This section shows you how to manage document types in Salesforce.

You can create, edit and delete your own document types. You can also translate and sort document types as well as assign document types to different Salesforce objects for search or storage purposes.

Creating, editing and deleting document types

Creating document types

You wish to create document types to restrict the search for documents. You can select the document type when importing documents.

This is how it works

1. Open the d.velop documents configuration.
2. Open **Document Types > Custom Document Types**.
3. Enter a name for your document type under **Add a custom document type**.
4. You can choose to activate **Is folder** if the new document type is to be flagged as a folder.
5. Click **Add**.

Editing document types

You wish to edit document types.

This is how it works

1. Open the d.velop documents configuration.
2. Open **Document Types > Custom Document Types**.
3. Click the pen icon in the table field that you wish to edit. Then enter a new value or activate/deactivate the displayed option.
4. Confirm the changes by pressing **ENTER**. The field is highlighted in color when the change has been successfully applied.
5. Click **Save**.

Deleting document types

You wish to delete document types.

This is how it works

1. Open the d.velop documents configuration.
2. Open **Document Types > Custom Document Types**.
3. Click **Delete** in the row for the document type that you wish to delete.
4. Confirm the operation with **Delete.**

Synchronizing document types from the document management system (DMS)

You can synchronize your categories from the d.velop documents DMS with Salesforce as document types and therefore use them without having to configure them.

> **Note**
> Synchronization of DMS categories is currently only available for d.velop documents cloud environments.

This is how it works

1. Open the d.velop documents configuration.
2. Open **Document Types > Custom Document Types**.
3. Click **Synchronize**.

Translating document types

You wish to add translations in other languages for your document types.

This is how it works

1. Open the d.velop documents configuration.
2. Open **Document Types > Naming**.
3. Select a language.
4. Click the pen icon in the table field that you wish to edit.
5. Enter a new value.
6. Confirm the changes by pressing **ENTER**. The field is highlighted in color when the change has been successfully applied.
7. Click **Save**.

Sorting document types

You wish to define the sorting of your document types for importing documents.

1. Open the d.velop documents configuration.
2. Open **Document Types > Sorting**.
3. Under **Select an object**, select an object.
4. Under **Select data record types**, select a data record type.
5. Select a document type by clicking the name. The selected entry is highlighted.
6. Change the sort order using the arrow buttons.
7. Click **Save**.

> **Note**
> You can then sort the document types that are assigned to the selected object and data record types.

Assigning document types to objects

You wish to assign document types to objects and data record types in order to restrict the search for documents. You can select the document type when importing documents.

This is how it works

1. Open the d.velop documents configuration.
2. Open **Document Types > Assignment**.
3. Under **Select an object**, select an object.
4. Under **Select data record types**, select a data record type.
5. Select the document types that you wish to assign. **Selected for search** activates the search for documents with the corresponding document type; **Selected for upload/folder creation** activates the import of documents for the document type.
6. Click the pen icon in the table field that you wish to edit. Activate or deactivate the displayed option.
7. Confirm the changes by pressing **ENTER**. The field is highlighted in color when the change has been successfully applied.
8. Click **Save**.

Customizing the settings for document types

You wish to define the settings for the document types.

You can specify whether only activated document types or also unknown document types are to be displayed in the document list. You can also specify for which objects it is possible to create folders.

This is how it works

1. In d.velop documents, navigate to **Configuration > Document Types > General Settings**.
2. Make the settings you require.

You have successfully defined the settings for the document types.

1.3.4. Managing properties

This section shows you how to manage properties in Salesforce.

You can create, edit and delete your own properties. You can also translate and sort properties as well as assign properties to different Salesforce objects for search or storage purposes.

Creating, editing and deleting properties

Creating properties

You wish to create properties. Properties are used to expand the available table columns in the d.velop archive Lightning component and to define the modifiable properties of a document when importing or editing.

This is how it works

1. Open the d.velop documents configuration.
2. Open **Properties > Custom Properties**.
3. Enter a name for your document type under **Add a custom property**.
4. Click **Add**.

Editing properties

You wish to edit properties.

This is how it works

1. Open the d.velop documents configuration.
2. Open **Properties > Custom Properties**.
3. Click the pen icon in the table field that you wish to edit. Then enter a new value.
4. Confirm the changes by pressing **ENTER**. The field is highlighted in color when the change has been successfully applied.
5. Click **Save**.

Deleting properties

You wish to delete properties.

This is how it works

1. Open the d.velop documents configuration.
2. Open **Properties > Custom Properties**.
3. Click **Delete** in the row for the property that you wish to delete.
4. Confirm the operation with **Delete**.

Synchronizing properties from the DMS

You can synchronize your properties from the **d.velop documents DMS** with Salesforce as properties and therefore use them without having to configure them.

> **Note**
> Synchronization of DMS properties is currently only available for d.velop documents cloud environments.

This is how it works

1. Open the d.velop documents configuration.
2. Open **Properties > Custom Properties**.
3. Click **Synchronize**.

Translating properties

You wish to add translations in other languages for your properties.

This is how it works

1. Open the d.velop documents configuration.
2. Open **Properties > Custom Naming**.
3. Select a language.
4. Click the pen icon in the table field that you wish to edit.
5. Enter a new value.
6. Confirm the changes by pressing **ENTER**.
7. Click **Save**.

Sorting properties

You wish to define the sorting of your properties for importing documents.

1. Open the d.velop documents configuration.
2. Open **Properties > Sorting**.
3. Under **Select an object**, select an object.
4. Under **Select data record types**, select a data record type.
5. Under **Select the document type**, select a document type.
6. Select a property by clicking the name. The selected entry is highlighted.
7. Change the sort order using the arrow buttons.
8. Click **Save**.

> **Note**
> You can then sort the properties that are assigned to the selected object, data record types and the selected document type.

Assigning properties to objects

You wish to assign properties to objects and data record types in order to restrict the search for documents. You can select the document type when importing documents.

This is how it works

1. Open the d.velop documents configuration.
2. Open **Properties > Mapping**.
3. Under **Select an object**, select an object.
4. Under **Select data record types**, select a data record type.
5. Under **Select the document type**, select a document type.
6. Select the properties that you wish to assign. **Display in Salesforce** makes the property available as a table column in the d.velop archive Lightning component and allows you to edit it when importing a

document in the selected document type. **Is mandatory** and **Read only** only affect write operations, e.g. importing or editing.
7. Click the pen icon in the table field that you wish to edit. Activate or deactivate the displayed option.
8. Confirm the changes by pressing **ENTER**.
9. Click **Save**.

1.3.5. Managing mappings

This section shows you how to manage mappings in Salesforce.

Using mappings, you can link Salesforce source fields to properties of your DMS documents. With a mapping, you can use the properties as search criteria to find documents with matching values. You can also predefine the values of the properties of a new document during import.

Creating mappings

You wish to create a mapping.

This is how it works

1. Open the d.velop documents configuration.
2. Open **Mapping**.
3. Under **Select an object**, select an object.
4. Under **Select data record types**, select a data record type.
5. Click **Add mapping**.
6. Define the source field to specify the calculated value of the mapping for the search and import. The source field can originate from various sources in Salesforce. Examples:
   - Data record on which the search or import is based
   - User that is logged in
   - Name of the file to be uploaded
   - Your own predefined values
7. Determine the target field. With the target field, you specify the property of the documents in which the value from the source field is searched for, or the property into which the source value is transferred during import.
8. Under **Select property**, select the relevant property of your DMS documents.
9. You can define properties for the mapping if you wish.
10. Click **Add mapping** or **Save & New** to close the dialog and create the mapping.
11. Activate **Search field** if the mapping is to be used for the search.
12. Activate **Allow Empty Search Value** if you want to allow a search despite an empty source value.
13. Activate **Hidden** if you want the property to be used during import but not displayed in the interface. The source value is then written to the document property but is not displayed visibly.
14. Click **Add mapping** or **Save & New** to close the dialog and create the mapping.

Editing mappings

You wish to edit the properties for a mapping.

This is how it works

1. Open d.velop documents configuration.
2. Open **Mapping**.
3. Click **Edit**.
4. Change the values for the properties that you want to update.
5. Click **Save**.

Deleting mappings

You wish to delete mappings.

This is how it works

1. Open the d.velop documents configuration.
2. Open **Mapping**.
3. Navigate to the row for the mapping.
4. Click **Delete**.
5. Click **Delete** to confirm the operation.

1.3.6. Configuring the search

You wish to configure the search for documents in the context of different objects and document record types.

The parameters for a search within d.velop components in Salesforce are determined from the respective object context. You adapt the settings at object level. The changes are then available in real time and you can search for the corresponding documents immediately.

The following steps are required to configure the search for a specific Salesforce object:

1. Determine which document types you wish to search for.
2. Configure the search attributes and create the appropriate search values using mappings.

Determining the document types for the search

You can define which document types you want to search for in the context of certain objects. You can use the document types to further narrow or expand the search results. Assign the corresponding document types to the appropriate context using the following steps.

This is how it works

1. Open the d.velop documents configuration.
2. Open **Mapping**.
3. Under **Select an object**, select an object.
4. Under **Select data record types**, select a data record type.
5. Select the document types that are to be mapped. **Selected for search** activates the search for documents with the corresponding document type; **Selected for upload/folder creation** activates the import of documents for the document type.
6. Click the pen icon in the table field that you wish to edit. Activate or deactivate the corresponding option.
7. Confirm the changes by pressing **ENTER**.
8. Click **Save**.

Setting up and assigning search properties and values

To set up a criterion for the search and match the criterion contextually with the correct field of a Salesforce record, create a mapping for the property and mark the property as a search field.

This is how it works

1. Open the d.velop documents configuration.
2. Open **Mapping**.
3. Under **Select an object**, select an object.
4. Under **Select data record types**, select a data record type.
5. Click **Add mapping**.
6. Determine the source field.
7. Define the source field to specify the calculated value of the mapping for the search and import. The source field can originate from various sources in Salesforce. Examples:
   - Data record on which the search or import is based

- User that is logged in
- Name of the file to be uploaded
- Your own predefined values

8. Under **Select source types**, select a source type. Then, depending on the source type you selected, you can either select a source field of a Salesforce object or enter a static value.
9. Determine the target field. With the target field, you specify the property of the documents in which the value from the source field is searched for, or the property into which the source value is transferred during import.
10. Under **Select property**, select the relevant property of your DMS documents.
11. You can define properties for the mapping if you wish.
12. Click **Add mapping** or **Save & New** to close the dialog and create the mapping.
13. Activate **Search field** if the mapping is to be used for the search.
14. Activate **Allow Empty Search Value** if you want to allow a search despite an empty source value.
15. Activate **Hidden** if you want the property to be used during import but not displayed in the interface. The source value is then written to the document property but is not displayed visibly.
16. Click **Add mapping** or **Save & New** to close the dialog and create the mapping.

Using a wizard for the search

To set up document searches quickly and easily with just a few clicks, you can use a wizard to guide you through the individual setup steps.

> **Note**
> The wizard for guided search setup is currently only available for d.velop documents cloud environments.

This is how it works

1. Open the d.velop documents configuration.
2. Click **Home**.
3. Click **Configure Search**.

The wizard then guides you through the setup process. Click **Finish** to complete the setup.

1.3.7. Configuring storage

You wish to configure the import of documents in the context of different objects and document record types.

The parameters for an import within d.velop components in Salesforce are determined from the respective object context. All settings are configured at object level so that the settings can then be retrieved in real time and the correct document types, properties and predefined values of the properties can be displayed and used.

The following steps are required to configure the import to a specific Salesforce item:

1. Determine which document types are to be imported.
2. Determine which properties of the documents are to be filled.
3. Prepopulate the selected properties with values from the data set context using mappings.

Determining the document types for the import

You can define which document types are to be uploaded in the context of certain objects and make these document types available to your users in the import dialog. To do so, assign the required document types to the appropriate context using the following steps.

This is how it works

1. Open the d.velop documents configuration.
2. Open **Document Types > Assignment**.
3. Under **Select an object**, select an object.
4. Under **Select data record types**, select a data record type.
5. Select the document types that are to be mapped. The **Selected for search** field activates the search for documents with the corresponding document type; the **Selected for upload/folder creation** field activates the import of documents for the document type.
6. Click the pen icon in the table field that you wish to edit. Then activate or deactivate the option.
7. Confirm the changes by pressing **ENTER**.
8. Click **Save**.

Determining the properties for the import

Just like document types, you can define the properties for importing documents. These properties are then displayed in the import dialog and filled in when the document is transferred to the DMS. To define them, assign the required properties to the appropriate context using the following steps.

This is how it works

1. Open the d.velop documents configuration.
2. Open **Properties > Mapping**.
3. Under **Select an object**, select an object.
4. Under **Select data record types**, select a data record type.
5. Under **Select the document type**, select a document type.
6. Select the properties that are to be mapped. The **Display in Salesforce** field makes the property available as a table column in the d.velop archive Lightning component and allows you to edit it when importing and editing a document in the selected document type. The **Is mandatory** and **Read only** fields only affect write operations, e.g. importing or editing.
7. Click the pen icon in the table field that you wish to edit. Then activate or deactivate the option.
8. Confirm the changes by pressing **ENTER**.
9. Click **Save**.

Prefilling properties using mappings

To fill documents with data from the current Salesforce data record during import, you can define assignments (mappings) that allow you to link the values from fields in the data record or other sources with the document properties. The properties are pre-filled accordingly in the import dialog and your users have the option to edit them before the document is transferred to the DMS. Perform the following steps to create a mapping.

This is how it works

1. Open the d.velop documents configuration.
2. Open **Mapping**.
3. Under **Select an object**, select an object.
4. Under **Select data record types**, select a data record type.
5. Click **Add mapping**.
6. Define the source field to specify the calculated value of the mapping for the search and import. The source field can originate from various sources in Salesforce. Examples:
   - Data record on which the search or import is based
   - User that is logged in
   - Name of the file to be uploaded
   - Your own predefined values
7. Under **Select source types**, select a source type.
8. Then, depending on the source type you selected, you can either select a source field of a Salesforce object or enter a static value.

9. Determine the target field. The target field determines the property of the documents in which the value from the source field is searched for, or the property to which the source value is written during import.
10. Under **Select property**, select the relevant property of your DMS documents.
11. You can define properties for the mapping if you wish:
    1. Activate **Search field** if the mapping is to be used for the search.
    2. Activate **Allow Empty Search Value** if you want to allow a search despite an empty source value.
    3. Activate **Hidden** if you want the property to be used during import but not displayed in the interface. The source value is then written to the document property but is not displayed visibly.
12. Click **Add mapping** or **Save & New** to close the dialog and create the mapping.

Using a wizard for the import

To set up document imports quickly and easily with just a few clicks, you can use a wizard to guide you through the individual setup steps.

> **Note**
> The wizard for guided import setup is currently only available for d.velop documents cloud environments.

This is how it works

1. Open the d.velop documents configuration.
2. Click **Home**.
3. Click **Configure import**.

The wizard then guides you through the setup process. Click **Finish** to complete the setup.

1.3.8. Customizing the document list

The following section contains information about the customization options for document lists, e.g. setting the sort order or configuring table columns and context options.

Customizing the settings for the document list

You wish to define the general settings for the document list. The settings for the document list apply to Salesforce for the whole organization as standard. Users can change these settings again in their own context.

This is how it works

1. In d.velop documents, navigate to **Configuration > Document list > General settings**.
2. Make the required changes.

You have successfully defined the settings for the document list.

Defining the sort order for search results

You wish to configure the sort order in which the search results of the document list are to be displayed.

This is how it works

1. Open the d.velop documents configuration.
2. Open **Document list > Sorting**.
3. Select the following properties:
   - Object
   - Record type

- Sort attribute
- Sort order

4. Click **Check**.
5. Following successful validation, click **Add**.

Configuring the table columns

You wish to define the attributes to be used as columns in the document list. You can make the settings either at object level or at user level.

This is how it works

1. Open the d.velop documents configuration.
2. Open **Document list > Column Assignment**.
3. Select the following properties:
   - Object
   - User
   - Available attribute
4. Use the arrows to add the attribute to **Selected attributes**.
5. Click **Save**.

Customizing the available context options

You wish to edit the available context options in the document list.

This is how it works

1. Open the d.velop documents configuration.
2. Open **Document list > Context Action Assignment**.
3. Select an object from the **Select an object** selection box.
4. Select a user from the **Select a user** selection box.
5. Under **Available options**, select an option that you would like to assign.
6. Select the following properties:
   - Object
   - User
   - Available option
7. Use the arrows to add the option to **Selected options**.
8. If necessary, adjust the display order using the arrows.
9. Click **Save**.

1.3.9. Exporting and importing configurations

This chapter explains how to back up (export) and restore (import) your configured settings.

Exporting

As an administrator, you wish to export and back up your configurations from Salesforce.

This is how it works

1. Open the system configuration.
2. Select **Import/Export**.
3. Select the settings that you wish to export. If you want to export all settings, click **Select all**.
4. Click **Download export file**.

A JSON file is saved.

Importing

You wish to restore saved configurations in Salesforce.

This is how it works

1. Open the system configuration.
2. Select **Import/Export**.
3. Click **Select file** and select the relevant file.
4. Click **Start import**.

The individual points are assigned automatically. At the end of the import process, you will receive a summary of which options were successfully restored.

1.3.10. Activating single sign-on

You want users to be able to log in with single sign-on.

This is how it works

1. Open the system configuration.
2. Under **Security settings,** enable the **Activate single sign-on** function.
3. Click **Save**.

Within the **User Cockpit**, you can check which Salesforce users are currently assigned to a corresponding d.velop documents user. You can also create d.velop documents users for the Salesforce users.

Requirements for single sign-on

For single sign-on to work, it is necessary for the following requirements to be fulfilled:

- Within your d.velop documents instance, Salesforce is listed as a **Trusted app**.
- In your d.velop documents instance, d.ecs identity provider version 2.4 or higher is installed.
- Upon activation of single sign-on, you have administrative permissions in d.ecs identity provider.
- The e-mail addresses of Salesforce users for whom single sign-on is to be set up must also be available as the e-mail addresses of the respective SCIM users in d.velop documents.

1.4. Configuration and administration of additional functions

Learn how to handle DMS documents in different areas of your Salesforce organization including: Lightning Experience, Flow, Salesforce Mobile App and Digital Experience.

1.4.1. Lightning App Builder

Use the d.velop components in the Lightning App Builder to connect your Lightning data record pages seamlessly with your DMS.

d.velop archive

d.velop archive is your central port of call for contextualized searching, filing and working with documents on Salesforce data records.

The document list shows you organized information. You can customize the document list to meet different requirements for each object, data record type, page layout and Lightning record page.

You can make the following entries in the Lightning application generator:

- **Maximum number of results:**
  The maximum number of hits in the table. Please note that a large number of documents may result in a correspondingly longer loading time.
- **Component Title:**
  A customized title that is displayed above the table.
- **Maximum number of files to upload:**
  The maximum number of files that can be uploaded to the connected DMS per operation.

  - Min.: 1
  - Max.: 50
- **Open documents in preview mode:**
  Activate this option if you want documents to be opened only in preview mode.
  This affects both the modal preview (click on a document; context action **View**) and the preview in a new tab (context action **New tab**).
- **Only show documents from within the last [x] days:**
  The maximum age of the displayed documents. Use this entry if you only want to show the most recent documents and want to hide older documents from the table.
- **Hide Title:**
  Activate this option if you wish to hide the title above the table.
- **Deactivate Upload:**
  Activate this option if you wish to deactivate the upload of documents in the table.
  This affects both the storage of new files (**Upload files**) and the storage of new versions (context action **New version**; context action **New version and properties**).

d.velop community archive

d.velop community archive is the counterpart to d.velop archive in **Digital Experience** (formerly **Communities**). To conform to the different environments, d.velop community archive has more input options than d.velop archive.

In d.velop community archive, you can enter the following:

- **Default Source Object:**
  The object type that is to be used as the context/source for all object-related operations.
  If you set this parameter, the current data record context is ignored and the object-wide context is used instead. Note that for this object you must define search parameters that do not depend on a record.
- **Source Object for Mappings:**
  The object type to be used to determine the object mappings.
  This entry overwrites **Default Source Object** and the data record context.
- **Source Object for Document Type Assignments:**
  The object type to be used to determine the document type assignments.
  This entry overwrites **Default Source Object** and the data record context.

For all other entries, refer to the entries in d.velop archive.

1.4.2. Flows

Salesforce Flow enables you to automate complex business processes. Use the DMS options in Flow Builder in order to work directly with your documents in your flows.

You can carry out simple document actions as well as more complex processes with the help of ready-made modules. You can also put together your own processes from the individual modules. You can use the following modules, among others:

- Apex actions
- Apex-defined data types
- Subflows
- Flow templates

Retrieving a document

You can retrieve the properties of a DMS document in a separate flow.

You can use the Apex action **Get DMS Document** to determine the properties of a document within a flow based on the document ID.

Creating the input variables

The Apex action expects a required parameter: The ID of the DMS document. We recommend creating a corresponding variable for this parameter so that you can configure the variable as an input for your flow, for example.

This is how it works

1. Navigate to **Flow Builder > Toolbox.**
2. Click **New Resource** to create a new variable for the document ID **(DMS Document ID)**.
3. Select **Variable** as the **resource type**.
4. Enter a unique name for your variable (e.g. **dmsDocumentId**).
5. Select **Text** as the **data type** for your variable. Leave the check box for **Allow multiple values (collection)** deselected.
6. Optional: Give your variable a default value or make the variable available for input (e.g. to include the flow on a Lightning record page). Neither is necessary for the Apex action.

Calling the Apex action

Once you have created the variable resource for the input, call the Apex action **Get DMS Document**.

This is how it works

1. Add the **Action** element to the flow.
2. Select the filter **Dvelop** or **Apex action** as the **Category** or **Type**.
3. Select the **Get DMS Document** action in the selection list.
4. Under **Label**, enter a label for your action.
5. Under **API name**, enter a unique name for your action.
6. Use the variables you have already created for the input values.
7. Optional: Define the input value **Run as Service-User**. Use the value **True** if the access details of the general system user are to be used. If you leave the input value blank or use **False**, the access details of the executing user are used.

Handling the output

When the Apex action has been executed, the Apex action returns the DMS document. If you want to continue using the DMS document, we recommend creating a variable.

This is how it works

1. Create a new variable for the DMS document. To do so, follow the instructions under Creating the input variables.
2. Select your Apex action.
3. Click **Edit element**.
4. Under **Advanced options**, activate the option **Manually assign variables**.
5. Under **Output Value**, select your new variable for the DMS document.

Updating a document

You can update the properties of a DMS document in a separate flow.

You can use the Apex action **Update DMS Document** to update the properties of a document within a flow.

Creating the input variables

The Apex action expects two required parameters: The ID of the DMS document that is to be updated and a list of properties that are to be adapted. We recommend creating corresponding variables for these parameters so that you can configure them as inputs for your flow, for example.

This is how it works

1. Navigate to **Flow Builder > Toolbox.**
2. Click **New Resource** to create a new variable for the document ID **(DMS Document ID)**.
3. Select **Variable** as the **resource type**.
4. Enter a unique name for your variable (e.g. **dmsDocumentId**).
5. Select **Text** as the **data type** for your variable. Leave the check box for **Allow multiple values (collection)** deselected.
6. Optional: Give your variable a default value or make the variable available for input (e.g. to include the flow on a Lightning record page). Neither is necessary for the Apex action.
7. Click **New Resource** to create a new variable for the list of properties to be updated **(Updated Attributes)**.
8. Select **Variable** as the **resource type**.
9. Enter a unique name for your variable (e.g. **updatedAttributes**).
10. Select **Defined by Apex** as the **data type** for your variable.
11. Select the Apex class **dvelop_docs_dev__DocumentAttribute**.
12. Activate the check box for **Allow multiple values (collection)**.
13. Click **New Resource** to create an auxiliary variable. The auxiliary variable defines an updating property that is added to the list created previously.
14. Select **Variable** as the **resource type**.
15. Enter a unique name for your variable (e.g. **attributeContainer**).
16. Select **Defined by Apex** as the **data type** for your variable. Select the Apex class **dvelop_docs_dev__DocumentAttribute**. Leave the check box for **Allow multiple values (collection)** deselected.

You can add additional optional parameters to the Apex action, e.g.:

- **Updated Category**
  - Type: Text
  - Collection: No

Repeat steps 2 to 6 for the other parameters that you wish to use.

Filling the variables

You can use the Apex class **dvelop_docs_dev__DocumentAttribute** to define a list variable. An auxiliary variable is required for the list variable. The auxiliary variable temporarily isolates the required data for an attribute. You then add the auxiliary variable to the list variable.

This is how it works

1. Add the **Assignment** element to the flow.
2. Select **dvelop_docs_dev__DocumentAttribute** and **key** under **Variable**.
3. Select the operator **Equal**.
4. Enter the attribute key (e.g. **objecttitle**) for the value of the assignment.
5. Click **Add assignment**.
6. Define the value of your attribute. To do this, repeat the procedure and select the property **value** of your auxiliary variable under **Variable**.
7. Optional: Add more values for the updated attribute, e.g. for an attribute with a lot of values. To do so, proceed as follows:
   a. Click **Add assignment**.
   b. Under **Variable**, select your auxiliary variable and the **values** property of the auxiliary variable (note the plural). Please note that the **value** property is always required for the first value of the attribute; otherwise, incorrect attribute values can occur.
   c. Select the operator **Add**.
   d. Enter the next desired value for the attribute.
8. When you have assigned all values to the property, click **Add assignment**.

9. Select your list of attributes as the **variable**.
10. Select the operator **Add**.
11. Select your auxiliary variable as the **value**.
12. Repeat all of the preceding steps for the remaining attributes.

Calling the Apex action

Once you have created the variable resources for the input, you can call the Apex action **Update DMS Document**.

This is how it works

1. Add the **Action** item to the flow.
2. Select the filter **Dvelop** or **Apex action** for **Category** or **Type**.
3. Select **Update DMS Document** in the selection list.
4. Under **Label**, enter a label for your action.
5. Under **API name**, enter a unique name for your action.
6. Use the variables you have already created for the input values.
7. Optional: Define the input value **Run as Service-User**. Use the value **True** if the access details of the general service user are to be used. If you leave the input value blank or use **False**, the access details of the executing user are used.

Handling the output

After executing the Apex action, you will receive the ID of the updated document for further processing. If you want to continue using the ID, we recommend creating a variable for the ID.

This is how it works

1. Follow the instructions under Creating the input variables and create a new variable for the document ID.
2. Select your Apex action and click **Edit element**.
3. Under **Advanced options**, activate the switch **Manually assign variables**.
4. Select your new variable as the output value for the document ID.

Creating a folder

You can use the Apex action **Create DMS Folder** to create a folder in the DMS within a flow.

Creating the input variables

The Apex action expects two required parameters: The ID of the DMS document that is to be updated and a list of properties that are to be adapted. We recommend creating corresponding variables for these parameters so that you can configure the variables as inputs for your flow, for example.

This is how it works

1. Navigate to **Flow Builder > Toolbox.**
2. Click **New Resource** to create a new variable for the DMS category of the folder **(Category)**.
3. Select **Variable** as the **resource type** and enter a unique name for your variable (e.g. **folderCategory**).
4. Select **Text** as the **data type** for your variable. Leave the check box for **Allow multiple values (collection)** deselected.
5. Optional: Give your variable a default value or make the variable available for input (e.g. to include the flow on a Lightning record page). Neither is necessary for the Apex action.
6. Click **New Resource** to create a new variable for the list of properties for the folder **(Attributes)**.
7. Select **Variable** as the **resource type** and enter a unique name for your variable (e.g. **folderAttributes**).
8. Select **Defined by Apex** as the **data type** for your variable. Select **dvelop_docs_dev__DocumentAttribute** as the Apex class. Activate the check box for **Allow multiple values (collection)**.

9. Click **New Resource** to create an auxiliary variable.
10. Select **Variable** as the **resource type** and enter a unique name for your variable (e.g. **attributeContainer**). The auxiliary variable defines a property for the folder that is then added to the list created previously.
11. Select **Defined by Apex** as the **data type** for your variable. Select **dvelop_docs_dev__DocumentAttribute** as the Apex class. Leave the check box for **Allow multiple values (collection)** deselected.

Filling the variables

You can use the Apex class **dvelop_docs_dev__DocumentAttribute** to define a list of document attributes. To fill such a list variable, an auxiliary variable is required that temporarily encapsulates the necessary data for an attribute and is then added to the collection of attributes.

This is how it works

1. Add the **Assignment** element to the flow.
2. Under **Variable**, select your auxiliary variable of the Apex-defined data types **dvelop_docs_dev__DocumentAttribute** and the **key** property.
3. Select the operator **Equal**.
4. Enter the attribute key (e.g. **objecttitle**) for the value of the assignment.
5. Click **Add assignment**.
6. Define the **value** of your attribute. Repeat the process for this and under **Variable**, select the **value** property of your auxiliary variable.
7. Optional: Add more values for the updated attribute, e.g. for an attribute with a lot of values. To do so, proceed as follows:
    a. Click **Add assignment**.
    b. Under **Variable**, select your auxiliary variable and then the **values** property (note the plural). Please note that the **value** property is always required for the first value of the attribute; otherwise, incorrect attribute values can occur.
    c. Select the operator **Add**.
    d. Enter the next desired value for the attribute.
8. When you have assigned all values of the property, click **Add assignment**.
9. Select your list of attributes as the **variable**.
10. Use the operator **Add**.
11. Select your auxiliary variable as the **value**.
12. Repeat all of the preceding steps for the remaining attributes.

Calling the Apex action

Once you have created the variable resources for the input, you can call the Apex action **Create DMS Folder**.

This is how it works

1. Add the **Action** element to the flow.
2. Select the filter **Dvelop** or **Apex action** for **Category** or **Type**.
3. Select the **Create DMS Folder** action in the selection list.
4. Under **Label**, enter a label for your action.
5. Under **API name**, enter a unique name for your action.
6. Use the variables you have already created for the input values.
7. Optional: Define the input value **Run as Service-User**. Use the value **True** if the access details of the general service user are to be used. If you leave the input value blank or use **False**, the access details of the executing user are used.

Handling the output

After executing the Apex action, you will receive the ID of the created folder for further processing. If you want to continue using the ID, we recommend creating a variable for the ID.

This is how it works

1. Follow the instructions under Creating an input variable and create a new variable for the folder ID.
2. Select your Apex action and click **Edit element**.
3. Under **Advanced options**, activate **Manually assign variables**.
4. Select your new variable as the output value for the folder ID.

Downloading a document

You can download a single document in the DMS in a separate flow and attach it to a data record.

You can use the Apex action **Download DMS Document to Record** to download a document within a flow based on the document ID and attach it to a Salesforce data record.

Creating the input variables

The Apex action expects two required parameters: The ID of the data record and the ID of the DMS document. We recommend creating corresponding variables for these parameters so that you can configure them as inputs for your flow, for example.

This is how it works

1. Navigate to **Flow Builder > Toolbox.**
2. Click **New Resource** to create a new variable for the ID of the data record **(Record ID)**.
3. Select **Variable** as the **resource type** and enter a unique name for your variable (e.g. **recordId**).
4. Select **Text** as the **data type** for your variable. Leave the check box for **Allow multiple values (collection)** deselected.
5. Optional: Give your variable a default value or make the variable available for input (e.g. to include the flow on a Lightning record page). Neither is necessary for the Apex action.
6. Click **New Resource** to create a new variable for the ID of the document **(DMS Document ID)**.
7. Select **Variable** as the **resource type** and enter a unique name for your variable (e.g. **documentId**).
8. Select **Text** as the **data type** for your variable. Leave the check box for **Allow multiple values (collection)** deselected.
9. Optional: Give your variable a default value or make the variable available for input (e.g. to include the flow on a Lightning record page). Neither is necessary for the Apex action.

Calling the Apex action

Once you have created the variable resources for the input, you can call the Apex action **Download DMS Document to Record**.

This is how it works

1. Add the **Action** element to the flow.
2. Select the filter **Dvelop** or **Apex action** for **Category** or **Type**.
3. Select the **Download DMS Document to Record** action in the selection list.
4. Under **Label**, enter a label for your action.
5. Under **API name**, enter a unique name for your action.
6. Use the variables you have already created for the input values.
7. Optional: Define the input value **Run as Service-User**. Use the value **True** if the access details of the general service user are to be used. If you leave the input value blank or use **False**, the access details of the executing user are used.

Handling the output

After executing the Apex action, you will receive the ID of the created document (**ContentDocument**) for further processing. If you want to continue using the ID, we recommend creating a variable for the ID.

This is how it works

1. Follow the instructions under Creating an input variable and create a new variable for **ContentDocument**.
2. Select your Apex action and click **Edit element**.
3. Under **Advanced options**, activate **Manually assign variables**.
4. Select your new variable as the output value for **ContentDocument**.

Downloading several documents

You can download several DMS documents in a separate flow and attach them to a data record.

You can use the **Download Documents** subflow to download several documents within a flow based on the document IDs and attach them to a Salesforce data record.

Creating the input variables

The subflow expects two required parameters: The ID of the data record and a list of IDs of the DMS documents. We recommend creating corresponding variables for these parameters so that you can configure them as inputs for your flow, for example.

This is how it works

1. Navigate to **Flow Builder > Toolbox**.
2. Click **New Resource** to create a new variable for the ID of the data record **(recordId)**.
3. Select **Variable** as the **resource type** and enter a unique name for your variable (e.g. **recordId**).
4. Select **Text** as the **data type** for your variable. Leave the check box for **Allow multiple values (collection)** deselected.
5. Optional: Give your variable a default value or make the variable available for input (e.g. to include the flow on a Lightning record page). Neither is necessary for the Apex action.
6. Click **New Resource** to create a new variable for the list of IDs for the DMS documents **(documentIdsToDownload)**.
7. Select **Variable** as the **resource type** and enter a unique name for your variable (e.g. **documentIds**).
8. Select **Text** as the **data type** for your variable. Activate the check box for **Allow multiple values (collection)**.

Filling the variables

You can fill a list of document IDs in different ways. For example, you can fill the entire list by writing the output of a previous action directly to the variable. The simplest way, however, is to add individual IDs using an assignment element.

This is how it works

1. Add the **Assignment** element to the flow.
2. Select the list variable you created previously as the **variable**.
3. Select **Equal** as the **operator**.
4. Enter the ID of a DMS document for the value of the assignment.
5. Repeat all of the preceding steps for more document IDs.

Calling the subflow

Once you have created the variable resources for the input, you can call the **Download Documents** subflow.

This is how it works

1. Add the **Subflow** element to the flow.
2. Select the flow **d.velop: Download Documents**.
3. Under **Label**, enter a label.
4. Under **API name**, enter a unique name.

5. Use the variables you have already created for the input values.
6. Optional: Define the input value **Run as Service-User**. Use the value **True** if the access details of the general service user are to be used. If you leave the input value blank or use **False**, the access details of the executing user are used.

Handling the output

After executing the subflow, you will receive a list of all IDs of the created documents (**ContentDocument**) for further processing. If you want to continue using the IDs, we recommend creating a variable for the IDs.

This is how it works

1. Follow the instructions under Creating an input variable and create a new variable for **ContentDocument**.
2. Tag the variable as a collection.
3. Select your subflow and click **Edit element**.
4. Under **Advanced options**, activate **Manually assign variables**.
5. Select your new variable as the output value for **ContentDocument**.

Uploading a document

You can upload a document to the DMS in a separate flow.

You can use the Apex action **Upload ContentDocument** to upload a document within a flow based on its **ContentDocument** ID and thereby adapt the properties of the DMS document to be created.

Creating the input variables

Once you have created the variable resources for the input, you can call the Apex action **Upload ContentDocument**.

This is how it works

1. Add a new **Action** element to the flow.
2. Select the filter **Dvelop** or **Apex action** for **Category** or **Type**.
3. Select the action **Upload ContentDocument** in the selection list.
4. Under **Label**, enter a label for your action.
5. Under **API name**, enter a unique name for your action.
6. Use the variables you have already created for the input values.
7. Optional: Define the input value **Preserve file after upload**. Use the value **True** if the document is to be retained after it has been successfully uploaded to Salesforce. If you leave the input value blank or use **False**, the document will be deleted once it has been uploaded.

Filling the variables

You can use the Apex class **dvelop_docs_dev__DocumentAttribute** to define a list of document attributes. To fill such a list variable, you require an auxiliary variable that temporarily encapsulates the necessary data for an attribute and is then added to the collection of attributes.

This is how it works

1. Add the **Assignment** element to the flow.
2. Under **Variable**, select your auxiliary variable of the Apex-defined data types **dvelop_docs_dev__DocumentAttribute** and the **key** property.
3. Select **Equal** as the **operator**.
4. Enter the attribute key (e.g. **objecttitle**) for the **value** of the assignment.
5. Click **Add assignment**.
6. Define the value of your attribute. Repeat the process for this and under **Variable**, select the **value** property of your auxiliary variable.

7. Optional: Add more values for the updated attribute, e.g. for an attribute with a lot of values. To do so, proceed as follows:
   a. Click **Add assignment**.
   b. Under **Variable**, select your auxiliary variable and its **values** property (note the plural). Please note that the **value** property is always required for the first value of the attribute; otherwise, incorrect attribute values can occur.
   c. Select the operator **Add**.
   d. Enter the next desired value for the attribute.
8. When you have assigned all values of the property, click **Add assignment**.
9. Select your list of attributes as the **variable**.
10. Use the operator **Add**.
11. Select your auxiliary variable as the **value**.
12. Repeat all of the preceding steps for the remaining attributes.

Calling the Apex action

Once you have created the variable resources for the input, you can call the Apex action **Upload ContentDocument**.

This is how it works

1. Add the **Action** element to the flow.
2. Select the filter **Dvelop** or **Apex action** for **Category** or **Type**.
3. Select the action **Upload ContentDocument** in the selection list.
4. Under **Label**, enter a label for your action.
5. Under **API name**, enter a unique name for your action.
6. Use the variables you have already created for the input values.
7. Optional: Define the input value **Preserve file after upload**. Use the value **True** if the document is to be retained after it has been successfully uploaded to Salesforce. If you leave the input value blank or use **False**, the document will be deleted once it has been uploaded.

Searching for documents

You can search for DMS documents in a separate flow.

You can use the Apex action **Search DMS Documents** to search for DMS documents within a flow.

Creating the input variables

The Apex action expects a required parameter: The ID of the data record that is to be used as context for the search for the DMS documents. Depending on this ID, search categories and search attributes are determined from the configured mappings and assignments. We recommend creating a corresponding variable for this parameter so that you can configure the variable as an input for your flow, for example.

This is how it works

1. Navigate to **Flow Builder > Toolbox**.
2. Click **New Resource** to create a new variable for the ID of the data record (**Record ID**).
3. Select **Variable** as the **resource type** and enter a unique name for your variable (e.g. **recordId**).
4. Select **Text** as the **data type** for your variable. Leave the check box for **Allow multiple values (collection)** deselected.
5. Optional: Define a default value for your variable or make the variable available for input (e.g. to include the flow on a Lightning record page). Neither is necessary for the Apex action.

You can add additional optional parameters to an Apex action to specify and format the search results. Below is a list of the parameters with their respective data types. Repeat steps 2 to 4 for the other parameters that you wish to use.

- **Search Categories**
  - Type: Text
  - Collection: Yes
- **Search Attributes**
  - Type: Apex-defined (**dvelop_docs_dev__DocumentAttribute**)
  - Collection: Yes
- **Hide Folders?**
  - Type: Boolean
  - Collection: No
- **Sort-Property**
  - Type: Text
  - Collection: No
- **Sort-Direction**
  - Type: Text (ascending/descending)
  - Collection: No
- **Search Text**
  - Type: Text
  - Collection: No
- **Maximum Age of Results in Days**
  - Type: Number
  - Collection: No

Defining additional search attributes

To search for additional attributes, you must fill your previously created list variable with the values of the attributes.

You can use the Apex class **dvelop_docs_dev__DocumentAttribute** to define a list of document attributes. To fill such a list variable, an auxiliary variable is required that temporarily encapsulates the necessary data for an attribute and is then added to the collection of attributes.

This is how it works

1. Add a new **Assignment** element to the flow.
2. Under **Variable**, select your auxiliary variable of the Apex-defined data types **dvelop_docs_dev__DocumentAttribute** and the **key** property.
3. Select **Equal** as the operator.
4. Enter the property key (e.g. **objecttitle**) for the value of the assignment.
5. Click **Add assignment**.
6. Define the value of your attribute. To do this, repeat the procedure and select the property **value** of your auxiliary variable under **Variable**.
7. Optional: Add more values for the updated attribute, e.g. for an attribute with a lot of values. To do so, proceed as follows:
   a. Click **Add assignment**.
   b. Under **Variable**, select your auxiliary variable and its **values** property (note the plural). Please note that the **value** property is always required for the first value of the attribute, otherwise incorrect attribute values can occur.
   c. Select the operator **Add**.
   d. Enter the next desired value for the attribute.
8. When you have assigned all values of the property, click **Add assignment**.
9. Select your list of attributes as the **variable**.
10. Use the operator **Add**.
11. Select your auxiliary variable as the **value**.
12. Repeat all of the preceding steps for the remaining attributes.

Calling the Apex action

Once you have created the variable resources for the input, you can call the Apex action **Search DMS Documents**.

This is how it works

1. Add a new **Action** element to the flow.
2. Select the filter **Dvelop** or **Apex action** for **Category** or **Type** .
3. Select the **Search DMS Documents** action in the selection list.
4. Under **Label**, enter a label for your action.
5. Under **API name**, enter a unique name for your action.
6. Use the variables you have already created for the input values.
7. Optional: Define the input value **Run as Service-User**. Use the value **True** if the access details of the general service user are to be used. If you leave the input value blank or use **False**, the access details of the executing user are used.

Handling the output

After executing the Apex action, you will receive the ID of the DMS documents found for further processing. If you want to continue using the ID, we recommend creating a variable for the ID.

This is how it works

1. Follow the instructions under Creating an input variable and create a new variable for the DMS document IDs.
2. Select your Apex action and click **Edit element**.
3. Under **Advanced options**, activate **Manually assign variables**.
4. Select your new variable as the output value for the DMS document IDs.

Searching for documents based on IDs

You can search for DMS documents in a separate flow based on IDs.

You can use the Apex action **Search DMS Documents With IDs** to search for DMS documents within a flow based on IDs.

Creating the input variables

The Apex action expects two required parameters: A list with IDs of the DMS documents and the key of the property that is assigned to the document ID in d.velop documents. We recommend creating a corresponding variable for this parameter so that you can configure the variables as input for your flow, for example.

This is how it works

1. Navigate to **Flow Builder > Toolbox**.
2. Click **New Resource** to create a new variable for the list of document IDs (**DMS Document IDs**).
3. Select **Variable** as the **resource type** and enter a unique name for your variable (e.g. **documentIds**).
4. Select **Text** as the **data type** for your variable. Activate the check box for **Allow multiple values (collection)**.
5. Click **New Resource** to create a new variable for the key of the ID property **(Searchattribute Key)**.
6. Select **Variable** as the **resource type** and enter a unique name for your variable (e.g. **documentId**).
7. Select **Text** as the **data type** for your variable. Leave the check box for **Allow multiple values (collection)** deselected.
8. Optional: Give your variable a default value or make the variable available for input (e.g. to include the flow on a Lightning record page). Neither is necessary for the Apex action.

Filling the variables

You can fill a list of document IDs in different ways. For example, you can fill the entire list by writing the output of a previous action directly to the variable. The simplest way, however, is to add individual IDs using an assignment element.

This is how it works

1. Add a new **Assignment** element to the flow.
2. Select the list variable you created previously as the **variable**.
3. Select **Equal** as the operator.
4. Enter the ID of a DMS document for the **value** of the assignment.
5. Repeat all of the preceding steps for more document IDs.

Calling the Apex action

Once you have created the variable resources for the input, you can call the Apex action **Search DMS Documents With IDs**.

This is how it works

1. Add a new **Action** element to the flow.
2. Select the filter **Dvelop** or **Apex action** for **Category** or **Type** .
3. Select the **Search DMS Documents With IDs** action in the selection list.
4. Under **Label**, enter a label for your action.
5. Under **API name**, enter a unique name for your action.
6. Use the variables you have already created for the input values.
7. Optional: Define the input value **Run as Service-User**. Use the value **True** if the access details of the general service user are to be used. If you leave the input value blank or use **False**, the access details of the executing user are used.

Handling the output

After executing the Apex action, you will receive a list of the DMS documents found for further processing. If you want to continue using this list, we recommend creating a variable for it.

This is how it works

1. Follow the instructions under Creating an input variable and create a new variable for the DMS documents.
2. Select your Apex action and click **Edit element**.
3. Under **Advanced options**, activate **Manually assign variables**.
4. Select your new variable as the output value for the DMS documents.

Sending e-mails with documents

In a separate flow, you can send e-mails with DMS and Salesforce documents attached.

You can use the Apex action **Send Email with Documents** to attach several DMS and Salesforce documents to an e-mail and send it within a flow.

Creating the input variables

The Apex action expects several required parameters: The e-mail subject, the body of the e-mail and a list of recipients. We recommend creating corresponding variables for these parameters so that you can configure them as inputs for your flow, for example.

This is how it works

1. Navigate to **Flow Builder > Toolbox**.
2. Click **New Resource** to create a new variable for the subject of the e-mail (**Subject**).

3. Select **Variable** as the **resource type** and enter a unique name for your variable (e.g. **subject**).
4. Select **Text** as the **data type** for your variable. Leave the check box for **Allow multiple values (collection)** deselected.
5. Optional: Define a default value for your variable or make the variable available for input (e.g. to include the flow on a Lightning record page). Neither is necessary for the Apex action.
6. Click **New Resource** to create a new variable for the content of the e-mail (**Body**).
7. Select **Variable** as the **resource type** and enter a unique name for your variable (e.g. **body**).
8. Select **Text** as the **data type** for your variable. Leave the check box for **Allow multiple values (collection)** deselected.
9. Optional: Define a default value for your variable or make the variable available for input (e.g. to include the flow on a Lightning record page). Neither is necessary for the Apex action.
10. Click **New Resource** to create a new variable for list of recipients (**Recipients**).
11. Select **Variable** as the **resource type** and enter a unique name for your variable (e.g. **recipients**).
12. Select **Text** as the **data type** for your variable. Activate the check box for **Allow multiple values (collection)**.

You can add additional optional parameters to an Apex action in order to add attachments to the e-mail or add additional recipients (on Bcc/Cc). Below is a list of the parameters with their respective data types. Repeat steps 2–4 for the other parameters that you wish to use.

- **CC Recipients**
  - Type: Text
  - Collection: Yes
- **BCC Recipients**
  - Type: Text
  - Collection: Yes
- **DMS Document IDs**
  - Type: Text
  - Collection: Yes
- **ContentDocument IDs**
  - Type: Text
  - Collection: Yes

Filling the variables

You can fill a list of document IDs in different ways. For example, you can fill the entire list by writing the output of a previous action directly to the variable. The simplest way, however, is to add individual IDs using an assignment element.

This is how it works

1. Add a new **Assignment** element to the flow.
2. Select the list variable you created previously as the **variable**.
3. Select **Equal** as the operator.
4. Enter the ID of a DMS document for the value of the assignment.
5. Repeat all of the preceding steps for more document IDs and **ContentDocument** IDs.

Calling the Apex action

Once you have created the variable resources for the input, you can call the Apex action **Send Email with Documents**.

This is how it works

1. Add a new **Action** element to the flow.
2. Select the filter **Dvelop** or **Apex action** for **Category** or **Type** .

3. Select the **Send Email with Documents** action in the selection list.
4. Under **Label**, enter a label for your action.
5. Under **API name**, enter a unique name for your action.
6. Use the variables you have already created for the input values.
7. Optional: Define the input value **Run as Service-User**. Use the value **True** if the access details of the general service user are to be used. If you leave the input value blank or use **False**, the access details of the executing user are used.

Using the document list as a screen flow component

You can integrate the document list into a screen flow.

Creating the input variable

The screen flow component expects a required parameter: The ID of the required record. We recommend creating a corresponding variable for this parameter so that you can configure it as input for your flow, for example.

In addition, you have the option to create a variable to deactivate the upload.

This is how it works

1. Navigate to **Flow Builder > Toolbox**.
2. Click **New Resource** to create a new variable for the data record ID.
3. Select **Variable** as the **resource type** and enter a unique name for your variable (e.g. **recordId**).
4. Select **Text** as the **data type** for your variable. Leave the check box for **Allow multiple values (collection)** deselected.
5. Optional: Give your variable a default value or make the variable available for input (e.g. to include the flow on a Lightning record page).

Integrating the screen flow component

Once you have created the variable resources for the input, you can integrate the components in a screen.

This is how it works

1. Add the **Screen** element to the flow.
2. Select the custom component **d.velop archive** in the selection list.
3. Enter a label and a unique API name for the component.
4. Use the variables you have already created for **Record Id**.
5. Optional: Define the input value **Run as Service-User**. Use the value **True** if you want to prevent document uploads in this screen flow. If you leave the input value blank or use **False**, the upload area of the document list is displayed and works in the context of **Record Id**.

You can now use and expand the flow.

Setting up follow-up flows

After specific functions have been performed, you can run a custom flow. This allows you to link your own logics to relevant events.

The following mechanisms (Apex classes) support the calling of a follow-up flow:

- **DvelopAttachmentUpload**
- **DocumentUploader**
- **ContentDocumentUploader**

In addition to these Apex classes, you can also start follow-up flows from other flows. The following flows support the calling of a follow-up flow:

- **Upload ContentDocument** (**dvelop_docs_dev__ContentDocumentUpload**)
- **Upload ContentDocuments** (**dvelop_docs_dev__ContentDocumentUploader**)
- **Upload Files from Record** (**dvelop_docs_dev__D3ProcessBuilderAttachmentUploader**)

When you start the document upload process, you can specify the API name of the flow which is to be called following the upload. Please note that the logic is considered completed even if the upload was not successful.

To use your flow for this functionality, you must make certain that this flow accepts the import parameter **uploadResults** of the type **dvelop_docs_dev.DocumentUploadResult**.

Structure of the "DocumentUploader" class

Data type	Name	Description
String	relatedEntityId:	The ID of the Salesforce object associated with the uploaded file (e.g., a user account ID with an attachment).
String	entityId	The ID of the processed file. This can be a ContentDocument, an e-mail or an attachment.
String	dmsDocId	The ID assigned to the file by the d.velop documents system after uploading.
String	processUUID	An ID that was automatically assigned during the upload process and can be used for analysis in the process log. This ID can be used in SOQL in the table **dvelop_docs_dev__ProcessLog__c**, for example.
Boolean	uploadSuccessfull	Indicates that the upload was a success.
Boolean	wasStarted	Indicates whether the upload for this file has already been started.

The follow-up flow is typically called when the upload has been completed, regardless of the success. In some cases, however, multiple instances of the follow-up flow may be running. This occurs when the Apex Heap Size for the running context is reached. This only takes place in the case of a large number of upload processes, however. It may be the case that the **wasStarted** attribute of the value is set as **false**, which means there are corresponding entries in the upload process chain that still need to be edited.

Use the **DocumentUploadResult** class to run your corresponding logics.

Common use cases for logics include cleaning up referenced entities or setting values for referenced entities.

Make sure that the API name for the follow-up flow matches the API name of the corresponding flow. If available, the name must include the namespace of your instance. Please note that the specified name is case-sensitive, and upper and lower case letters must be taken into account.

If the specified flow is not found, the documents cannot be uploaded. If no follow-up flow is specified, the upload will occur without a follow-up flow after each upload process.

> **Note**
> If you define a follow-up flow for the **Upload ContentDocuments** flow, the follow-up flow will be run for all **ContentDocumentUploadRequests** contained.

1.4.3. Quick action dialogs

Using **quick action** dialogs, you can give your users access to d.velop components at many points in your workflow. In order to set up **quick action** dialogs, you can configure Lightning components as **quick actions** and integrate them in your tasks and processes.

Displaying documents on a record as a quick action dialog

You want to set up a dialog to display documents on a data record as a **quick action**.

This is how it works

1. Navigate to **Setup > Object Manager**.
2. Select the object on which you want the display dialog to be displayed.
3. Open **Buttons, Links and Actions**.
4. Click **New Action.**
5. In the **Action Type** selection box, select the option **Lightning Web Component** and then **dvelop_docs_dev:documentViewerQuickAction**.
6. Enter a label and a unique name.
7. Click **Save**.
8. Open **Page Layouts**.
9. Select **Mobile & Lightning Actions**.
10. Drag the action you created previously into the **Salesforce Mobile and Lightning Experience Actions** area. You may need to select the option to override the actions first.

Sending e-mails with documents in a quick action dialog

You want to set up a dialog for sending emails with documents as a **quick action** dialog.

This is how it works

1. Navigate to **Setup > Object Manager**.
2. Select the object on which you want the e-mail dialog to be displayed
3. Open **Buttons, Links and Actions**.
4. Click **New Action.**
5. In the **Action Type** selection box, select the option **Lightning Web Component** and then **dvelop_docs_dev:documentViewerQuickAction**.
6. Enter a label and a unique name.
7. Click **Save**.
8. Open **Page Layouts**.
9. Select **Mobile & Lightning Actions**.
10. Drag the action you created previously into the **Salesforce Mobile and Lightning Experience Actions** area. You may need to select the option to override the actions first.

Using the document list in a quick action dialog

You want to set up the document list in a **quick action** dialog.

This is how it works

1. Navigate to **Setup > Object Manager**.
2. Select the object on which you want the document list to be displayed.
3. Open **Buttons, Links and Actions**.
4. Click **New Action**.
5. In the **Action Type** selection box, select the option **Lightning Component** and then **dvelop_docs_dev:Xocor**.
6. Enter a label and a unique name.
7. Click **Save**.
8. Open **Page Layouts**.
9. Select **Mobile & Lightning Actions**.
10. Drag the action you created previously into the **Salesforce Mobile and Lightning Experience Actions** area. You may need to select the option to override the actions first.

1.4.4. Salesforce Connect

With Salesforce Connect, you can synchronize your DMS documents as an external object in your Salesforce environment.

You need to perform the following steps for the synchronization:

1. Set up the external data source
2. Create and synchronize the external object
3. Allow reports from the external object (optional)

Setting up an external data source

In order to synchronize an external object for your DMS documents, you must first set up an external data source.

This is how it works

1. Open **Setup > Integrations > External Data Sources**.
2. Click **New External Data Source**.
3. Enter a label and a name for your external data source.
4. Select the type **Salesforce Connect: Custom - dvelop_docs_dev.DataSourceProvider**.
5. Click **Save**.

Creating and synchronizing an external object

In order to make your DMS documents available in Salesforce, use an external data source to create an external object that represents the structure of your documents.

This is how it works

1. Open **Setup > Integrations > External Data Sources**.
2. Open your external data source.
3. Click **Add**.
4. Click **Validate and Sync.**
5. Select the **Document** table for synchronization.
6. Assign a name and the label. Under **Name Field**, you can select **filecaption** by default.
7. Click **Synchronization**.

Linking an external object with other Salesforce objects

External objects can be linked with other Salesforce objects using lookup relationships. You can then use your external object as normal in page layouts, Lightning data record pages and other areas.

This is how it works

1. Open **Setup > Integrations > External Objects**.
2. Open your external object.
3. Select a synchronized field that you want to convert to a lookup relationship. You must enter the ID of a Salesforce data record in the field value or property value of the document. Otherwise, a relationship cannot be established between an external data record and other data records.
4. Click **Edit > Change Field Type**.
5. In the **Reference to** selection box, select the Salesforce object that is to be linked with a data record via the field value. If the field value contains, for example, the ID of account data records, select **Account** as the object.
6. Click **Next**.
7. Optional: Enter a name in **Child Relationship Name**.
8. Click **Save**.

1.4.5. Platform events

Platform events allow you to connect unrelated processes within and outside of Salesforce.

Use platform events, for example, to react dynamically to the upload of a new **ContentDocument** document or DMS document and to process the documents further with your own flows.

Reacting to the upload of a Salesforce file

When a file is uploaded in Salesforce to **Files** or to the record under **Files & attachments**, a **ContentDocument** data record is created. The **ContentDocument** data record is linked to the corresponding data record. Use **ContentDocument** data records to react to the upload using an Apex trigger and then to trigger your own processes or execute your own Apex code.

> **Note**
> The following types of data records are not supported by data record flows:
>
> - **ContentDocument**
> - **ContentDocumentLink**
> - **ContentVersion**
>
> If these unsupported data records are created, the reaction with an Apex trigger is required. In order to still use flows – and thus also all flow modules for work with DMS documents – follow the sections below.

Creating a platform event

Create a platform event in order to be able to react to the upload of a Salesforce file in a flow.

This is how it works

1. Navigate to **Setup**.
2. Open **Setup > Integrations > Platform Events**.
3. Select **New Platform Event**.
4. Enter a unique label, plural label and a unique object name.
5. Select **Publish After Commit**. This ensures that the new **ContentVersion** data record is saved correctly in the system before an upload attempt is started.
6. Under **Deployment Status**, select the status **Deployed**. This enables you to use the event straight away.
7. Save your changes.
8. Under **Custom Fields & Relationships**, click **New** to add new fields.
9. Select **Text** as the **data type**.
10. Enter a unique field label and a field name and enter the length as follows:
    - **ContentDocumentId** - Length: 18
    - **RelatedEntityId** - Length: 18
    - **RelatedEntityType** - Length: 255
11. Click **Save**.

Publishing the event when uploading a Salesforce file

If a file is uploaded in Salesforce, you can react to it with an Apex trigger with the **ContentDocumentLink** object. In the object, you can, for example, publish your own platform event with the necessary information in order to then react to it in a flow.

This is how it works

1. Open **Developer Console**.
2. Select **File > New > Apex Trigger**.
3. Enter a name for the trigger.
4. Under **Object**, select the option **ContentDocumentLink**.
5. Replace the content of the trigger as shown in the following example:

```
trigger ContentDocumentLinkTrigger on ContentDocumentLink(after insert) {
    List<ContentDocumentUpload__e> eventsToPublish = new
```

```
List<ContentDocumentUpload__e>();

    // Collect necessary fields from each created ContentDocumentLink
    for (ContentDocumentLink createdLink : Trigger.NEW) {
        ContentDocumentUpload__e eventForLink = new
ContentDocumentUpload__e(
            ContentDocumentId__c = createdLink.ContentDocumentId,
            RelatedEntityId__c = createdLink.LinkedEntityId,
            RelatedEntityType__c =
String.valueOf(createdLink.LinkedEntityId.getSObjectType())
        );

        eventsToPublish.add(eventForLink);
    }

    EventBus.publish(eventsToPublish);
}
```

Reacting to the upload of a DMS document

You can use the platform event **UploadFinishedEvent** (**dvelop_docs_dev__UploadFinishedEvent__e**) to react dynamically to the upload of a new DMS document. The event can be used by triggers, Process-Builder and flows.

Every successful upload of a DMS document triggers a new event. Failed upload attempts cannot be used with the event.

Overview of the available fields in the event

The platform event **dvelop_docs_dev__UploadFinishedEvent__e** has the following fields that you can use in your flow or Apex trigger:

- **RecordId__c**: The unique Salesforce ID of the object for which the upload was performed (Account, Opportunity, etc.).
- **RelatedDocumentId__c**: The unique ID of the uploaded document in the corresponding DMS if the ID is provided in response to the upload.
- **RuntimeContext__c**: The execution context in which the upload was performed. You can find possible values here: Salesforce Apex Reference Guide > Quiddity Enum.
- **UserId__c**: The unique Salesforce ID of the user who performed the upload or on whose behalf the upload was performed.
- **Category__c**: The unique ID of the category of the uploaded document in the corresponding DMS.
- **Properties__c**: The properties of the uploaded document. The properties are described as follows:
  - **values**: Values entered for the property
  - **key**: Unique ID for the property in the corresponding DMS

Using the event in a flow

You can react to the upload of a DMS document with a **platform event** and a separate flow.

This is how it works

1. Navigate to **Setup**.
2. Open **Process Automation > Flows**.
3. Select **New Flow**.
4. Define the type of the flow.
5. Select **Platform Event-Triggered Flow**.
6. Under **Start**, you will find the start conditions for your flow. As the flow is triggered by an event, select the event **UploadFinishedEvent** under **Platform Event**.

7. Click **Save**.
8. Define a **flow label** and a **flow API name**.

Using the event in an Apex trigger

You can react to the upload of a DMS document with a platform event and your own Apex trigger.

This is how it works

1. Open **Developer Console**.
2. Select **File > New > Apex Trigger**.
3. Enter a name for the trigger.
4. Under **Object**, select the option **dvelop_docs_dev__UploadFinishedEvent__e**.
5. Use the event in your trigger as shown in the following example:

```
trigger UploadFinished on dvelop_docs_dev__UploadFinishedEvent__e (after
insert) {
    for (dvelop_docs_dev__UploadFinishedEvent__e event : Trigger.NEW) {
        System.debug('Event RecordID: ' +
event.dvelop_docs_dev__RecordId__c);
        System.debug('Event RelatedDocumentID: ' +
event.dvelop_docs_dev__RelatedDocumentId__c);
        System.debug('Event RuntimeContext: ' +
event.dvelop_docs_dev__RuntimeContext__c);
        System.debug('Event UserId: ' +
event.dvelop_docs_dev__UserId__c);
        System.debug('Event CategoryKey: ' +
event.dvelop_docs_dev__Category__C);
        System.debug('Event Properties: ' +
event.dvelop_docs_dev__Properties__C);
    }
}
```

Transferring a new Salesforce file to the DMS

You can use your own platform events that will be started after uploading Salesforce files. This enables you to transfer the files directly to the connected DMS.

The following sections show some example use cases.

Creating a flow

The simplest way to react to the upload of a Salesforce file is with a platform event and a separate flow.

This is how it works

1. Navigate to **Setup**.
2. Open **Process Automation > Flows**.
3. Select **New Flow**.
4. Define the type of the flow.
5. Select **Platform Event-Triggered Flow**.
6. Under **Start**, you will find the start conditions for your flow. As the flow is triggered by an event, select a previously created event under **Platform Event**.
7. Click **Save**.
8. Enter a **flow label** and **flow API name**.

Execute action – Transfer new file to the DMS

Once you have created your flow with the basic modules, you can execute actions and access the data/fields of the platform event. For example, you can read the object type from the event and, based on this, store the uploaded **ContentDocument** data record in different DMS categories.

This is how it works

1. Add the **Action** element to the flow.
2. Select the filter **Dvelop** or **Apex action** for **Category** or **Type** .
3. Select the **Upload ContentDocument** action.
4. Enter a **label** and an **API name**.
5. For the input values, use the fields from the platform event, which you can access via the global variable **$Record**.
6. Optional: Define the input value **Preserve file after upload**. Use the value **True** if you do not want the Salesforce file to be automatically deleted after it has been successfully uploaded. If you leave the input value blank or use **False**, the file will be deleted automatically once it has been uploaded.
7. Optional: Define the input value **Run as Service-User**. Use the value **True** if the access details of the general service user are to be used. If you leave the input value blank or use **False**, the access details of the executing user are used.

1.5. Frequently asked questions

In this section, you can find answers to frequently asked questions.

1.5.1. Why is the type "Salesforce Connect Custom" not displayed?

You require an additional license to use Salesforce Connect. If you do not yet have a license for Salesforce Connect, contact your Salesforce contact person.

1.5.2. How do I define the column width for the document list?

Open **System Configuration > Documentlist > Column Width**. Then define the column width for the document list for each Salesforce object and user.

1.5.3. How do I ensure the connection between d.velop documents (on-premises) and the Salesforce and AWS servers is secure?

As the network structure is unique to each organization, we cannot provide you with one-size-fits-all instructions. Contact your internal IT department to ensure your server connection is secure.

1.5.4. How do I delete all active user sessions?

Active user sessions are cached using the Salesforce Platform Cache to prevent repeated logins to the DMS and improve application performance. You can delete all active user sessions.

This is how it works

1. Navigate to **Setup > Custom Cache > Platform Cache**.
2. Open the **UserSessions** partition.
3. Click **Delete Cache**.

1.5.5. How do I open the log?

In the log, you have access to all error messages that occur during the use of d.velop connect for Salesforce. Navigate to **System configuration > Log** to display the log.

1.5.6. How do I open the system configuration?

You wish to adjust the system configuration for d.velop documents for Salesforce.

This is how it works

1. Open the app launcher  in Salesforce.
2. Search for **d.velop documents configuration** to open the general settings.
3. Adapt the system configuration.

If you receive an error message when opening the system configuration, you do not have the necessary permissions to change the system configuration.

1.5.7. How does Salesforce connect with d.velop documents in terms of technology?

The Salesforce app connects to the d.velop documents API using HTTP/S. The d.velop documents API is based entirely on REST technology. To establish the connection, enter the API endpoint (URI) in the Salesforce app.

1.5.8. Where do I enter my e-mail address and other login information?

In many organizations, the login information for each user is applied centrally. In some cases, you may also be prompted to enter your own login information (e-mail, user name, password).

This is how it works

1. Open the App Launcher in Salesforce.
2. Click **View All > d.velop documents user settings**.
3. Enter your user name or e-mail address. The data that is required depends on the settings in your ECM system.
4. Enter your password or the API key. The data that is required depends on the settings in your ECM system.
5. Save your settings.

1.6. Additional information sources and imprint

If you want to deepen your knowledge of d.velop software, visit the d.velop academy digital learning platform at https://dvelopacademy.keelearning.de/.

Our E-learning modules let you develop a more in-depth knowledge and specialist expertise at your own speed. A huge number of E-learning modules are free for you to access without registering beforehand.

Visit our Knowledge Base on the d.velop service portal. In the Knowledge Base, you can find all our latest solutions, answers to frequently asked questions and how-to topics for specific tasks. You can find the Knowledge Base at the following address: https://kb.d-velop.de/

Find the central imprint at https://www.d-velop.com/imprint.