

Object Image File Administrator Guide - v1b

The Object Image File package provides the ability to:

- **Display a Product image or Contact photo** on record detail pages, lightning record pages, experience cloud object pages, flows, list views, reports and the mobile app.
- **Bulk Import Image Files** using standard data loading tools that support ContentVersion file uploads and are matched against their related Product/Contact records using a (unique) reference field such as a Product Code or Contact Email field.

Note: The term “Object Image File”, “image (content) file” or “image file” refers specifically to a ContentVersion file (and associated record) that has been loaded using the Object Image File import/upload process (described below). Existing files already in the org will have other image files including other file types which are not managed or queried by the features or functionality described in this guide.

Note 2: References to configuration metadata are written in `Courier New` font and can be found (and in some cases directly linked to) under the list of managed package components in:

Installed Package > Package Details - Object Image File (Managed) > Metadata Components Included in Package

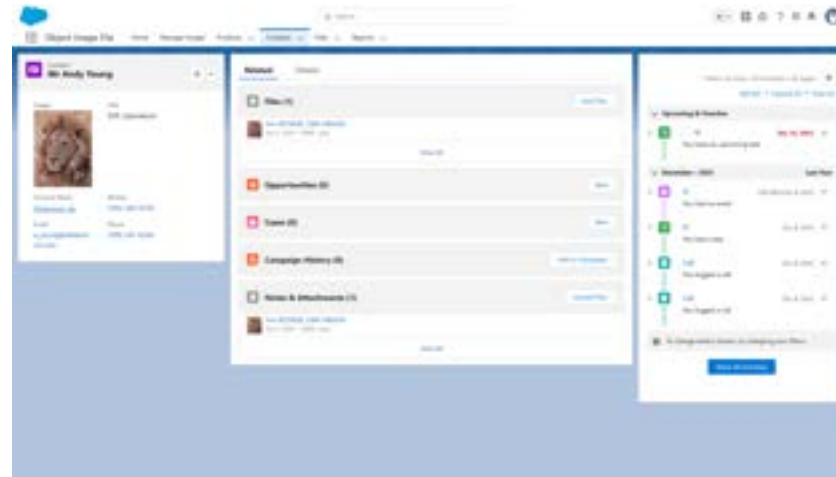
Screenshots

Examples where Product images and Contact photos can appear include:

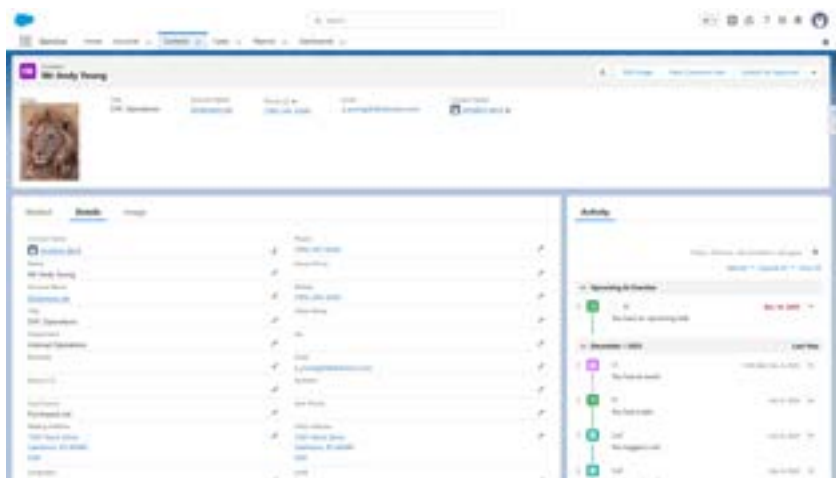
- **Page Layouts** - by using the Product/Contact image field In the Highlights Panel and/or Field Sections.
- **Lightning Experience Record Pages** - by using the Product/Contact image field in the Record Detail component, Dynamic Forms and/or Dynamic Highlights Panel.



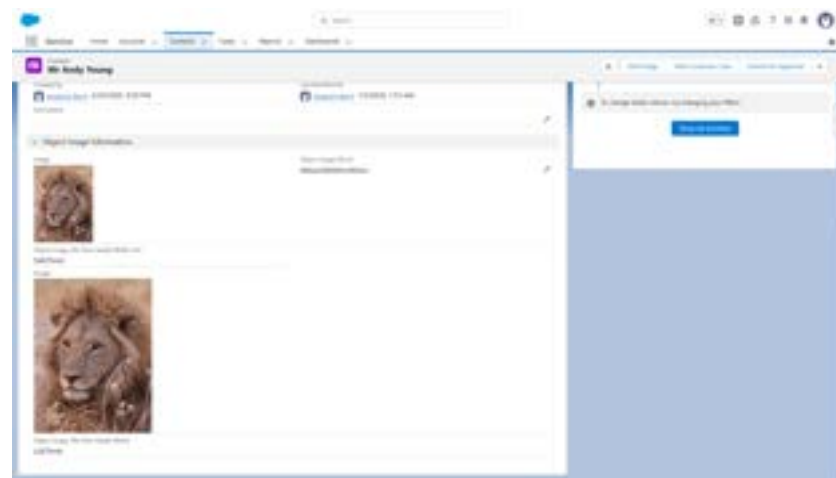
Product Record Page - Image



Contact Record Page - Image2

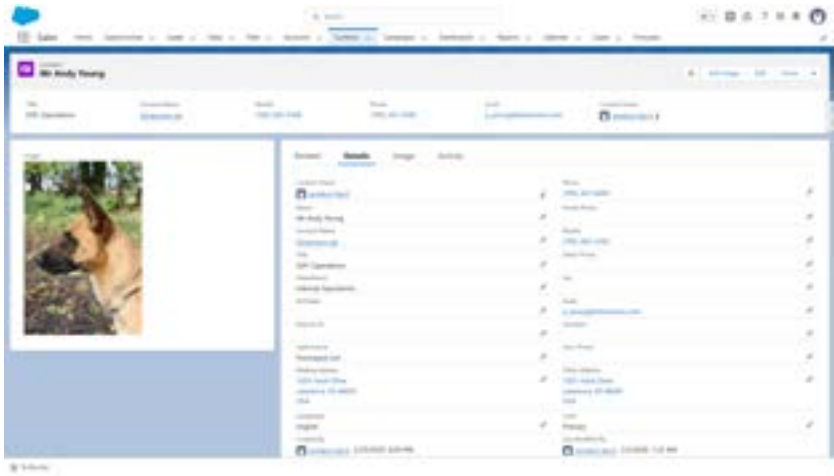


Contact Record Page - Image



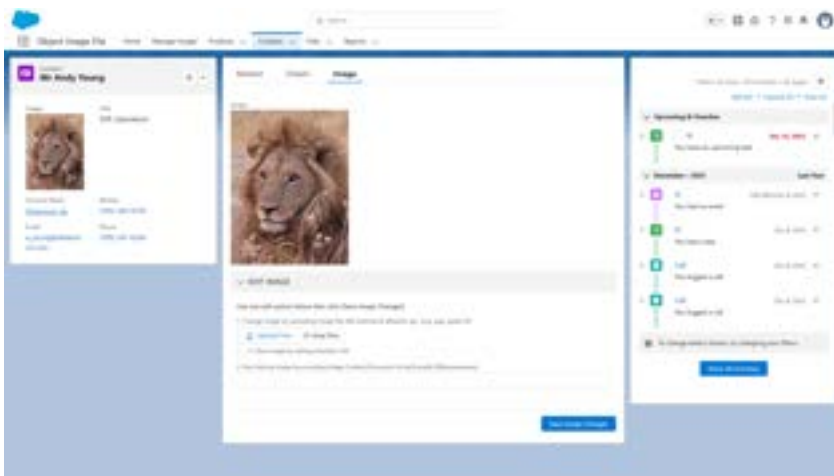
Contact Layout - Image

- **Lightning Web Component (LWC)** - by using the Lightning Web Component (`objectImageDisplay`) to display images where LWCs can be placed (and the record ID is available) such as lightning pages or screen flows.



Contact Record Page - Image3

- **Screen Flow** (as a flow on a page) - by using the screen flow (`Object Image File View Flow`) images can be displayed wherever flows can be placed (and the record ID is available). The flow also allows the image to be added/updated (with a new file), replaced (with an existing file) or deleted. The flow can also be used as an action on the detail page to provide an edit image action button.

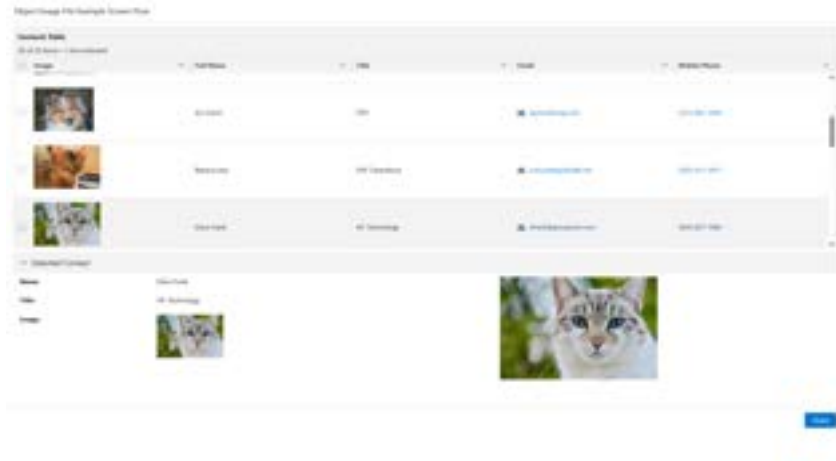


Object Image File View Flow



Edit Image

- **Screen Flows** (as a component within a screen component) - by referencing the Product/Contact image field in a screen component.



Object Image File Example Screen Flow

- **Printable View** - by using the [Printable View] action button on the detail page containing the image.



Product Layout - Image



Contact Layout - Image

- **List Views** - by including the Product/Contact image field.

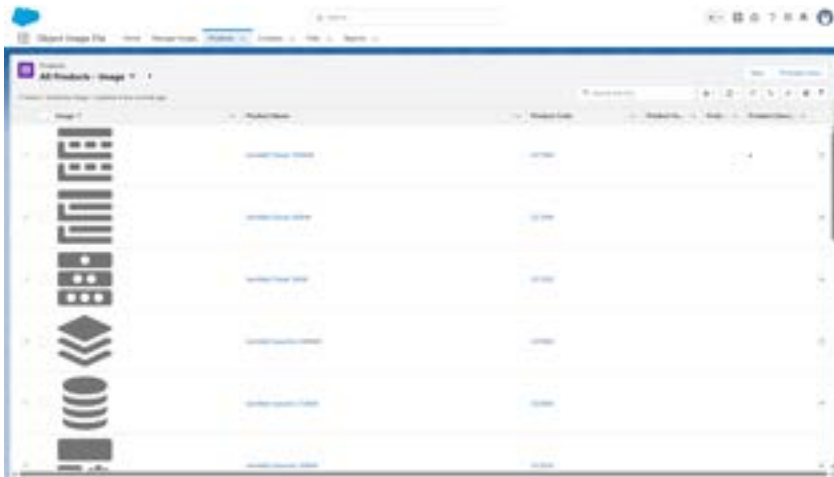


Image	Product Name	Product Code	Product Price	Product Status
	Product Name 1	Product Code 1	Product Price 1	Product Status 1
	Product Name 2	Product Code 2	Product Price 2	Product Status 2
	Product Name 3	Product Code 3	Product Price 3	Product Status 3
	Product Name 4	Product Code 4	Product Price 4	Product Status 4
	Product Name 5	Product Code 5	Product Price 5	Product Status 5

All Products - Image

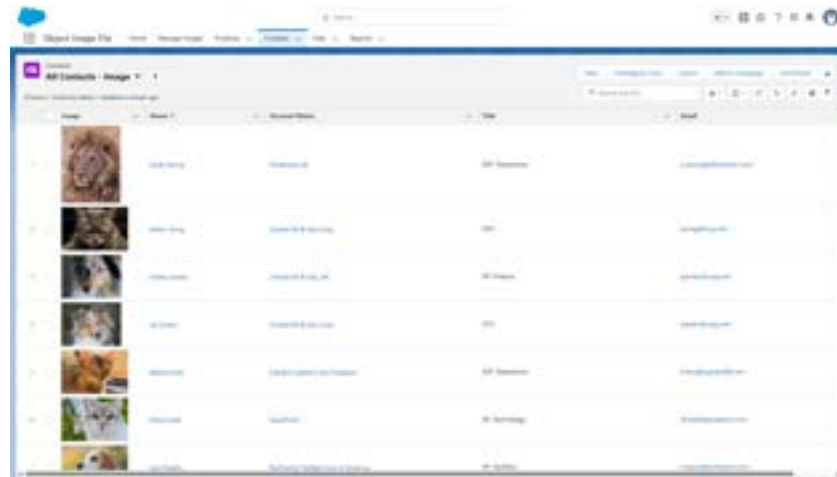


Image	Contact Name	Contact Email	Contact Phone	Contact Address
	Contact Name 1	Contact Email 1	Contact Phone 1	Contact Address 1
	Contact Name 2	Contact Email 2	Contact Phone 2	Contact Address 2
	Contact Name 3	Contact Email 3	Contact Phone 3	Contact Address 3
	Contact Name 4	Contact Email 4	Contact Phone 4	Contact Address 4
	Contact Name 5	Contact Email 5	Contact Phone 5	Contact Address 5

All Contacts - Image

- **Reports** - by including the Product/Contact image field.

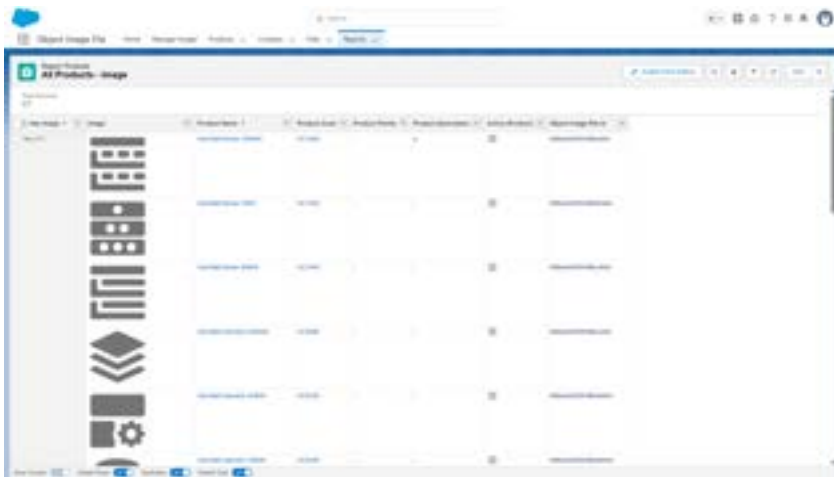


Image	Product Name	Product Code	Product Price	Product Status
	Product Name 1	Product Code 1	Product Price 1	Product Status 1
	Product Name 2	Product Code 2	Product Price 2	Product Status 2
	Product Name 3	Product Code 3	Product Price 3	Product Status 3
	Product Name 4	Product Code 4	Product Price 4	Product Status 4
	Product Name 5	Product Code 5	Product Price 5	Product Status 5

All Products - Image

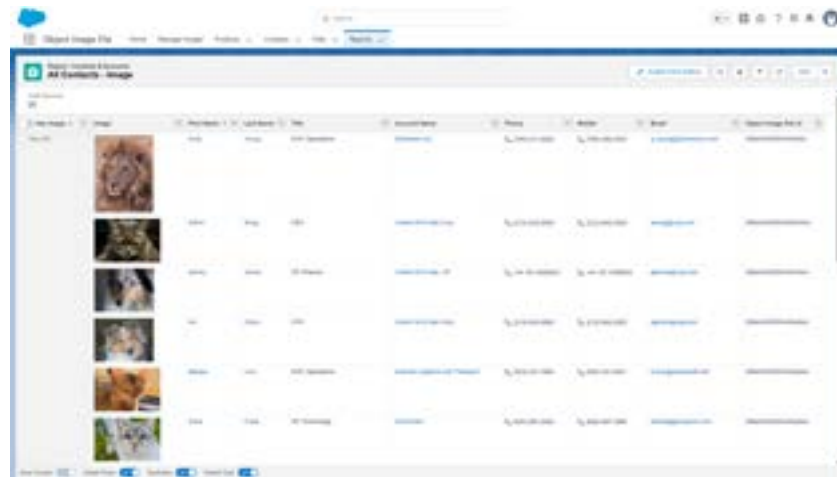


Image	Contact Name	Contact Email	Contact Phone	Contact Address
	Contact Name 1	Contact Email 1	Contact Phone 1	Contact Address 1
	Contact Name 2	Contact Email 2	Contact Phone 2	Contact Address 2
	Contact Name 3	Contact Email 3	Contact Phone 3	Contact Address 3
	Contact Name 4	Contact Email 4	Contact Phone 4	Contact Address 4
	Contact Name 5	Contact Email 5	Contact Phone 5	Contact Address 5

All Contacts - Image

- **Experience Cloud Pages** - by using the image field on Product/Contact Detail/List Object Pages.

Image	Product Name	Price	Status
[Placeholder Icon]	Product Name 1	10.00	Active
[Placeholder Icon]	Product Name 2	15.00	Active
[Placeholder Icon]	Product Name 3	20.00	Active
[Placeholder Icon]	Product Name 4	25.00	Active
[Placeholder Icon]	Product Name 5	30.00	Active
[Placeholder Icon]	Product Name 6	35.00	Active

Product List

Image	Contact Name	Phone Number	Email Address	Status
[Placeholder Icon]	Contact Name 1	123-456-7890	john.doe@example.com	Active
[Placeholder Icon]	Contact Name 2	234-567-8901	jane.smith@example.com	Active
[Placeholder Icon]	Contact Name 3	345-678-9012	john.doe@example.com	Active
[Placeholder Icon]	Contact Name 4	456-789-0123	jane.smith@example.com	Active
[Placeholder Icon]	Contact Name 5	567-890-1234	john.doe@example.com	Active
[Placeholder Icon]	Contact Name 6	678-901-2345	jane.smith@example.com	Active

Contact List

Product Detail Form:

- Product Name:
- Price:
- Status:
- Image:

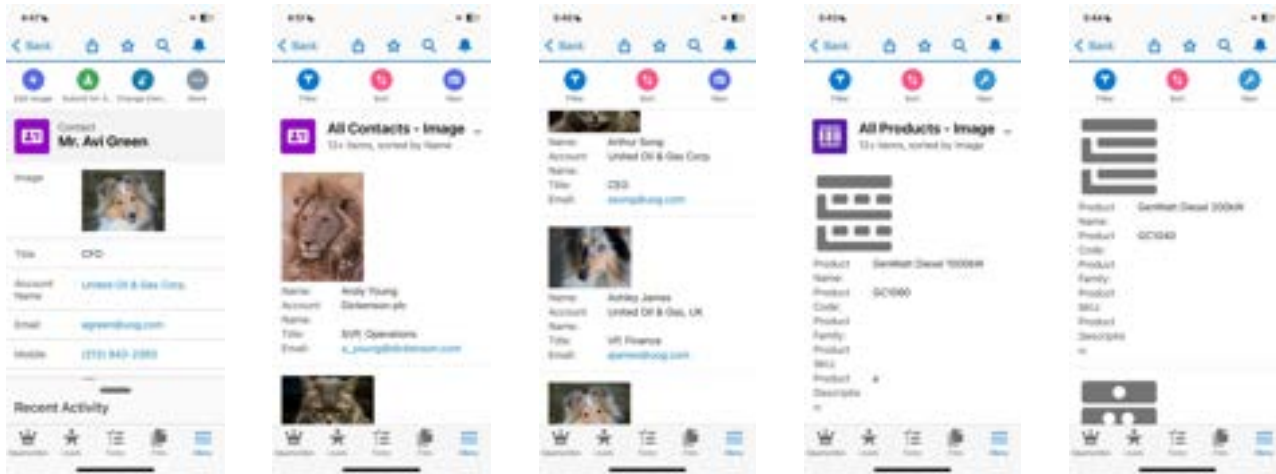
Product Detail

Contact Detail Form:

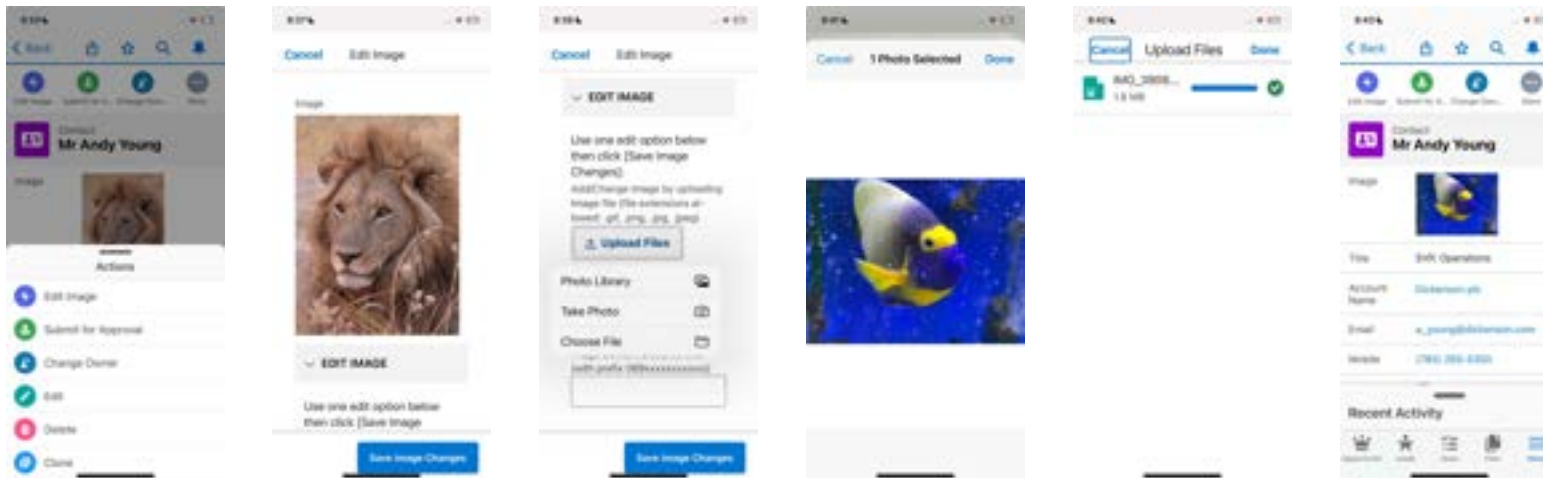
- Contact Name:
- Phone Number:
- Email Address:
- Image:

Contact Detail

- Salesforce Mobile App** - by using the image field on Product/Contact page layouts and list views. The screen flow (Object Image File View Flow) when used as an action on the detail page also provides an add/edit image action button on the mobile app.



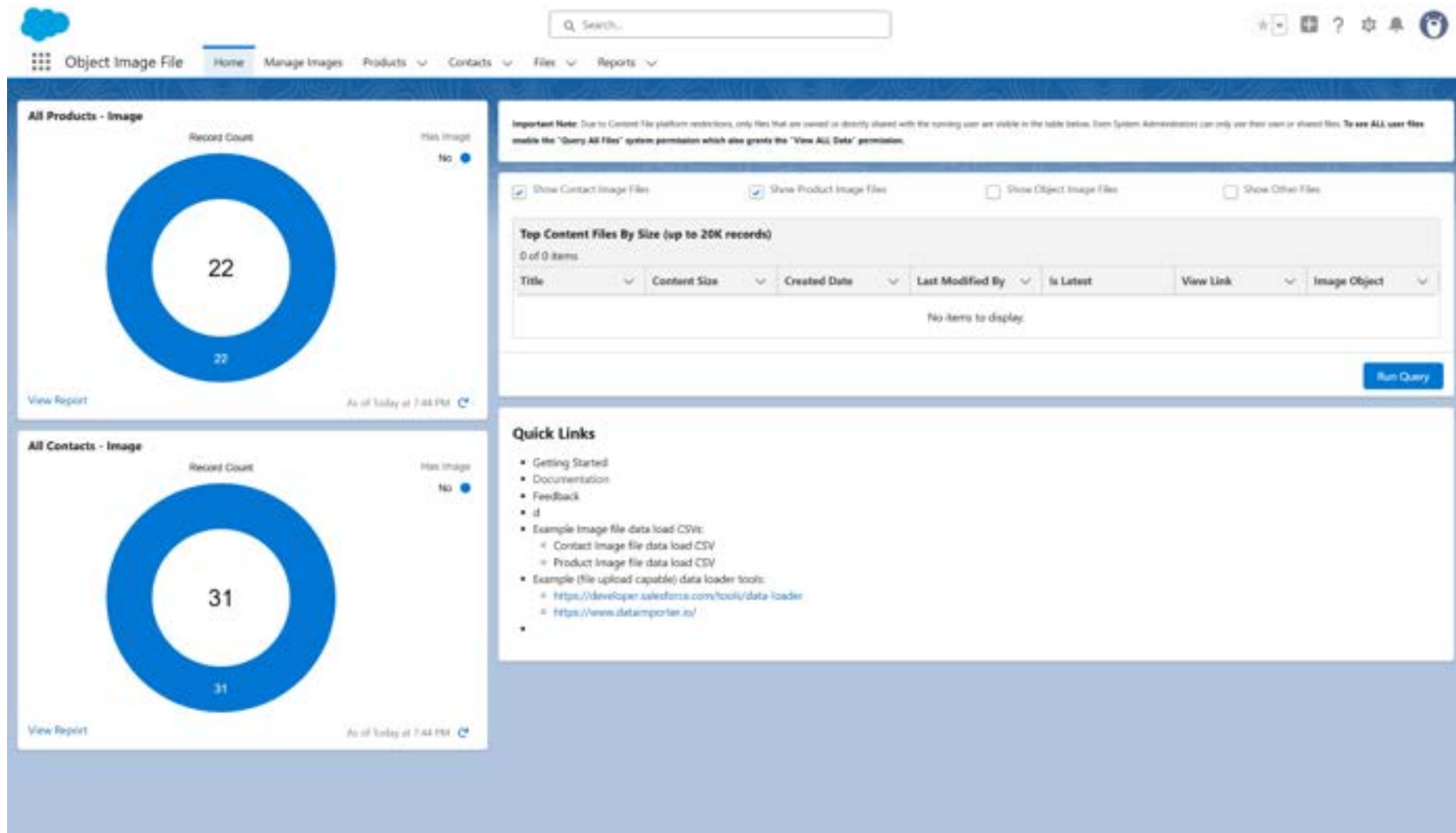
Contact Layout - Image, All Contacts - Image, All Products - Image



Contact Layout - Image, Object Image File View Flow, Edit Image

Other Features

- **Lightning App** - from the launcher is the `Object Image File` app in the Lightning Experience App Manager.
- **Home Page** - contains the following areas:
 - Report charts showing number of images loaded against Contacts/Products.
 - Links to the underlying report shows the records that have no images loaded.
 - The “Top Content Files By Size” table shows image files sorted by largest to smallest size.
 - Filter checkboxes are available to show Product / Contact images files, all image files and/or all other (content) files.
 - Note that only the files that are visible to the running user are shown unless the “Query All Files” permission has been granted.
 - Quick Links provide links to useful related documentation, tools or files.



Object Image File Home

- **Manage Images** - provides the ability to query and manage image (content) files.
 - The default query returns all object image files.
 - Filters are available for Title (contains), Size (less than), Last Modified Date and Image Object type, Image Field Value (starts with).
 - Filter checkboxes are available to show Product / Contact images files, all image files and/or all other (content) files.
 - Query results can be sorted (ascending) by Title, Size or Date.
 - The default number of records returned is 99 and up to 20k records could be returned but this may result in performance issues.
 - Note that only the files that are visible to the running user are shown unless the “Query All Files” permission has been granted.
 - There are three action buttons available to delete files selected in the results table, or delete ALL Product image files or ALL Contact image files.
 - The result table shows relevant object image file details used matching against related Product/Contact records. There are also links to view or download the image file (ContentVersion record). Editing ContentVersion file details can be done by using the View link and then editing from the details page.



Search...



Object Image File

HomeManage ImagesProductsContactsFilesReports

Query Filters

Important Note: Due to Content File platform restrictions, only files that are owned or directly shared with the running user are visible in the query results. Even System Administrators can only see their own or shared files. To see ALL user Content Files enable the "Query All Files" system permission which also grants the "View All Data" permission.

Delete Selected Content Files

Delete ALL Product Image Content Files

Delete ALL Contact Image Content Files

Show Debug

Title contains

Record Limit

* Sort By

☒ Show Object Image Files

☐ Show Other Files

Size greater than or equal to

OR Size greater than or equal to (KB)

Last Modified less than

OR Last Modified after date

Image Object (standard)

Image Object (custom)

☐

Image Field Value starts with

Query Results

0 of 0 items • 0 items selected

Image

Title

Extension

Size

Modified Date

Modified By

View

Download

Image Object

Image Field Value

No items to display.

Run Query

Manage Images

Installation and Setup / Usage

The following steps provide a minimal setup scenario, additional setup or steps may be required for specific org, use case and/or other requirements/processes.

Important Notes:

- When installing the package, Experience Cloud may need to be enabled (but no sites are necessary) in order to allow the installation of templated Contact/Product Experience Cloud object pages that are part of the package.
- When querying or managing (image) files, due to Content File platform restrictions, only files that are owned or directly shared with the running user are visible in list views, reports and tables. Even System Administrators can only see their own or shared files. To see ALL user files enable the "Query All Files" system permission which also grants the "View ALL Data" permission.
- When editing an image on records, the user must have underlying edit permission on the record being edited in order to set the object field `Object Image File Id`.

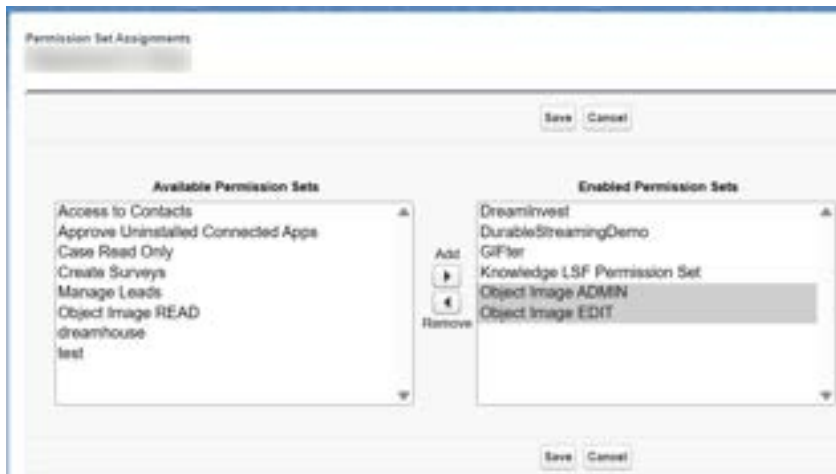
1. Install package

- Use the "Install for Admins only" optionAssign system administrator/image librarian "Object Image ADMIN" and "Object Image EDIT" permissions.



Assign user/profile permission sets

- Assign system administrator/image librarian “Object Image ADMIN” and “Object Image EDIT” permissions.
- Assign users who can edit images the “Object Image EDIT” permission. Underlying edit permission on the object whose image is being edited must be provided separately, otherwise the edit will fail with a flow error.
- Assign users who can see images the “Object Image READ” permission.

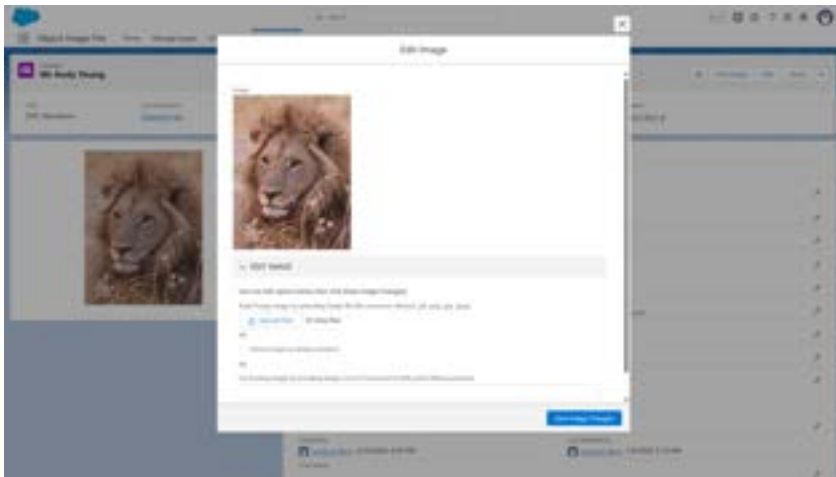


2. Start the “Object Image File” app

- As an “Object Image ADMIN” user, use the launcher to navigate to the “Object Image File” app. Under the “Home” or “Manage Images” tabs there should be no images loaded against any Product/Contact records until the bulk image load or manual image edit flow are used.

3. Load images

- Content image files can be individually loaded under the Product/Contact record or bulk loaded/imported using a data loader tool that supports file loading.
- An initial image can be loaded for a Product and/or Contact record under the “Products” or “Contacts” tab by selecting a Product/Contact record and using the details page [Edit Image] button or “Image” tab.



4. Assign page layouts/lightning pages

- Assign the appropriate Product/Contact page layouts, lightning record/object pages, list views, and/or reports to any users who will view/edit images.
- The layouts, record/object pages, list views, reports, screen flows, Experience Builder pages can be created, cloned and modified as required using the image field, image flow component or image Lightning Web Component (with the record ID as the input parameter).
- Example layouts and pages are provided for reference/cloning in the screenshots above.

Configuration and Metadata

The following metadata and configuration is used:

- The image is displayed using a custom (formula) field named “Image” with API name `kltv_dev3__Object_Image_File_View__c` (using the IMAGE function) on the Product/Contact object. The field label cannot be directly changed for a managed package field however the Translation Workbench > Override option could be used to change the label by an administrator.
- A second custom image formula field `kltv_dev3__Object_Image_File_View_Small__c` is also available for cases where a thumbnail size image is more appropriate such as list, report, table and highlight views. Both the large and small image sizes can be displayed depending on the display mode/format/type being used.
- The custom image formula field is also available as a Lightning Web Component (LWC) so that it can be used almost anywhere where an LWC is placed (and a record ID can be provided).
- Image display size can be changed by configuring a specially formatted Custom Metadata Type record `Object Image File Setting > Image Height Width`. The value is in fixed character format with each line representing the Object Id (3-letter) Prefix, a dash/minus character, then the (zero padded) 3 digit pixel height and width of the image separated by a comma. For example, `001-300,200` - specifies the Account object image (prefix 001) will have a height of 300 pixels and a width of 200 pixels.
- There are three additional special height/width values that can be used: `LGEThmb`, `MEDThmb`, `SMLThmb`, a height width value of `000,000` and a height custom value with a width value of `-01`.
 - The Large, Medium, Small Thumbnail values generate dynamically resized (and compressed) images which are optimised for faster loading. These images are preferred for most display purposes (especially for mobile device display). For example, `003-SMLThmb` generates an optimised smaller thumbnail size (120 x 90) Contact photo ideal for list, report, table and highlight views. `MEDThmb` generates 240 x 180 size images and `LGEThmb` generates 720 x 480 size images. These sizes are preset Salesforce image sizes and cannot be changed.
 - The height width value of `300,-01` generates an image with a height of 300 pixels and retains the original image proportions without stretching or shrinking of the original image to fit within a predefined size. This is ideal for when image sizes are not known in advance and/or cannot be formatted/resized before loading.
 - The height width value of `000,000` displays the original image size which can cause display and performance problems if the original loaded image files have not been resized appropriately or have file sizes that are too large. This is not an ideal setting unless the original loaded image files have been checked and optimized for display and loading.

- An additional secondary smaller/thumbnail size image field setting `Object Image File Setting > Image Height Width_Small`
- Configuration settings are found in the Custom Metadata Type `Object Image File Setting`. In general, the settings do not need to change except if a custom image height width is required or to deactivate the ContentVersion trigger that matches and links the image file to the Product/Contact object.
 - **Field Name For Image File Id** - the field name containing the ContentVersion image ID.
 - **Image Default URL** - the image url to use if there is no image loaded against an object. By default an X image is used.
 - **Image File Visibility** - the default visibility of the image when linking image to object.
 - **Image Height Width** - controls the image size for different object types based on the object ID 3-character prefix.
 - **Image Height Width Small** - controls the smaller/thumbnail image size.
 - **Trigger Delete Existing Match** - enables automatic deletion of existing matching image files when loading a new image.
 - **Trigger Disable** - bypasses the logic in the ContentVersion trigger that matches and links the image file to the Product/Contact object. The trigger still fires but does nothing instead.
- Images are stored as ContentVersion records/files against the related Product/Content record.
- The Lightning Web Component (`objectImageDisplay`) has settings to override default values. The “label” setting changes how the field label is displayed, possible values include:
 - **standard** - label is positioned based on the screen component form’s density
 - **label-hidden** - hides the label
 - **Enter Your Own Text Value** - provides the ability to override the label just by specifying the custom label text desired

The other settings are reserved for future use.
- The screen flow (`Object Image File View Flow`) has configurable parameters to control how to whether to display the edit options:
 - **varHideDeleteOption** - remove delete option from flow screen
 - **varHideEditOptions** - remove all edit options from flow screen
 - **varHideUseExistingOption** - remove use existing option flow screen

The other parameters are reserved for future use.

Possible/Potential Issues or Error Scenarios

The list below contains a (non-exhaustive) list of possible/potential issues/errors that might be encountered:

- An X image is displayed if the `Object Image File Id` field is empty. A valid/existing `ContentVersion` record ID (prefix 068) should be loaded in this field.
- After loading a new image and using the (recommended) thumbnail rendering options (`LGEThmb`, `MEDThmb`, `SMLThmb`) there can be a delay for the rendering/caching to occur. Refresh the page after a delay to give the image time to render and cache. The reverse delay also applies when clearing/deleting the image, it can take some time for the cached image to be removed from the browser.
- A broken image icon may indicate an `Object Image File Id` field that contains an invalid value, e.g a `ContentVersion` record Id for a file that has been deleted in some way. This can be fixed by clearing the field manually, or use the `Manage Images` tab to clear all fields and files, or adding a new or restoring the original image file.
- It is possible to have “orphaned” image files if the `Object Image File Id` field is cleared or updated with some other value. All image file (`ContentVersion`) records should have a valid `n object linked entity name` and `object linked field (matching)` value. These images can be restored by using the existing file’s `ContentDocument ID` (prefix 069) in the `Edit Image` action.
- When bulk data loading image files it is possible there may be multiple/duplicate records or no matching records when doing linked entity lookup, and only the first one will be matched or no image matched against the `Product/Contact` record. Ensure there is a single unique matching value (e.g. similar to an external ID) for each image file loaded against each linked entity `Product/Contact` record.
- If no image fields can be seen in the layout/page at all then the user may not have the correct security/permissions for the image field. Check that an `Object Image File` permission set has been assigned to the user and/or other security on the field or related metadata.
- Sometimes there can be errors while data loading image files, due to invalid linked entity field names or values. Ensure that each record has a valid entity field name and value similar to the example CSV files. A partially successful load can cause images not to load, it may be better to use smaller batch sizes with validated field names/values that ensures no errors occur and help isolate problematic records.
- If a flow error happens when saving while editing/adding image flow, it may be due to insufficient edit permission on the object’s image file id field, e.g. the `Product` object is not usually in most profiles, so edit permission on the object whose image is being edited must be provided separately.

Considerations and Limitations

- Due to Content File platform restrictions, only files that are owned or directly shared with the running user are visible in the table below. Even System Administrators can only see their own or shared files. To see ALL user files enable the "Query All Files" system permission which also grants the "View ALL Data" permission.
- It is expected that the System Administrator(s) will perform all image data loads and manage image files, however it may be possible that a single (non-admin) user without the "View all files" (and the "View All Data" permission) could be assigned the central (and only) record owner "Image Librarian" role which would ensure that all Product/Contact image files are loaded, validated and maintained correctly. Perhaps all image file record ownership could be automatically changed to the librarian user for approval/maintenance purposes.
- Large volumes (thousands) of images may require more testing. When data loading, it may be better to use smaller batches. When managing/querying image files it may be better to use more selective filters to return smaller volumes.
- Large volumes and/or sizes of image files use more file storage capacity, check file storage usage remains within org limits.
- Due to the design principles (below) there are some constraints with certain features, functionality and usability, such as the lack of pagination, limited query filter options, layout/component placement/structure/flexibility, as well as style/look-and-feel.

Helpful Tips

- Use the small thumbnail render image size option (`SMLThmb`) for resize/compression especially on list views/reports/mobile app for faster and efficient image loading performance.
- Example layouts, pages and other metadata are provided as reference and can be copied/cloned for learning and a faster setup.
- Example CSV files (with sample image files) are provided as a template for a quicker start to data loading/import.

Design Principles and Constraints

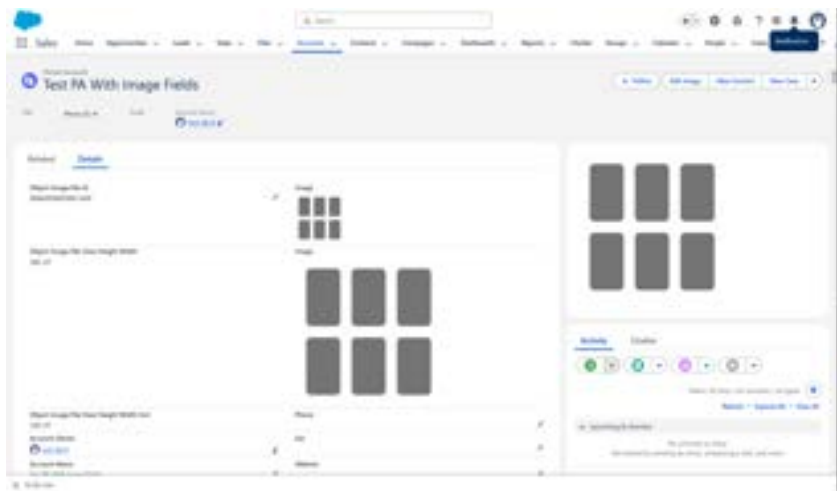
The following design principles and constraints were used to build this app:

- Configuration over code, minimize code and use only where necessary. A low code, no code approach has been taken, and is preferred over better custom-built options (as seen in the user interface compromises).
- Pre-built/standard/flow-based user interface/components over custom-built/coded components.
- Minimize resources/time/effort.

Potential Features and Ideas

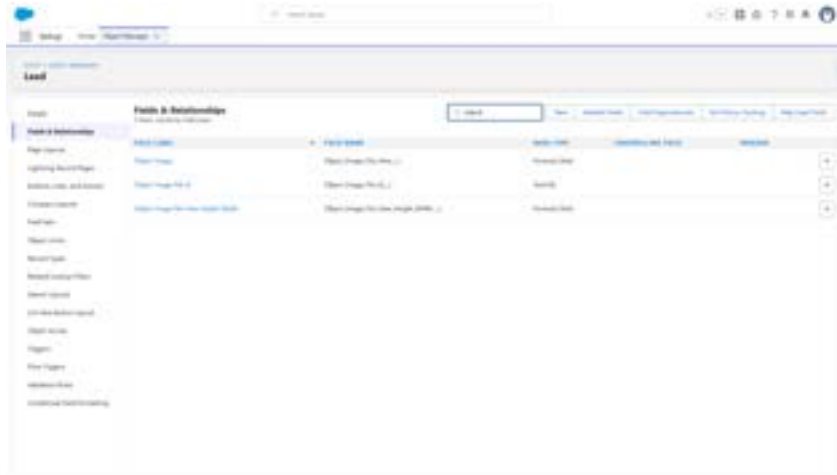
The following ideas and features may be considered in future based on further feedback and level of interest/demand:

- **Image field for other standard and custom objects** - other standard and custom Object support, e.g. Accounts and Leads? (Although Account image fields are provided, an unmanaged package may be provided in future)
 - **Account/PersonAccount image field** - available in package, manual set-up to add to pages, layouts, list views, etc, same as Contact/Product which is already provided which can be “copied”. This should also work with the PersonAccount object because it shares the same Account image field but has a separate layout.



- **Lead and other custom objects** - not available in package but could be manually configured with current package by:

1. Copying the existing image fields and recreating them in the desired object with the same name (e.g. Object_Image_File_Id__c, Object_Image_File_View_Height_Width__c, Object_Image_File_View__c)



2. Then modifying any formulas to remove the namespace prefix to match the manually created image fields.

Object_Image_File_View__c

```
IF(ISBLANK(Object_Image_File_Id__c), IMAGE($Label.kltv_dev3__Object_Image_Default_URL, ''),

IF(LEFT(Object_Image_File_View_Height_Width__c,3)='SML', IMAGE(SUBSTITUTE(LEFT($Api.Partner_Server_URL_630, FIND( '/services',
$Api.Partner_Server_URL_630)), 'my.sales', 'lightning.') +
'sfc/servlet.shepherd/version/renditionDownload?rendition=THUMB120BY90&operationContext=CHATTER&versionId='+Object_Image_File_Id__c, ''),
IF(LEFT(Object_Image_File_View_Height_Width__c,3)='MED', IMAGE(SUBSTITUTE(LEFT($Api.Partner_Server_URL_630, FIND( '/services',
$Api.Partner_Server_URL_630)), 'my.sales', 'lightning.') +
'sfc/servlet.shepherd/version/renditionDownload?rendition=THUMB240BY180&operationContext=CHATTER&versionId='+Object_Image_File_Id__c, ''),
IF(LEFT(Object_Image_File_View_Height_Width__c,3)='LGE', IMAGE(SUBSTITUTE(LEFT($Api.Partner_Server_URL_630, FIND( '/services',
$Api.Partner_Server_URL_630)), 'my.sales', 'lightning.') +
'sfc/servlet.shepherd/version/renditionDownload?rendition=THUMB720BY480&operationContext=CHATTER&versionId='+Object_Image_File_Id__c, ''),

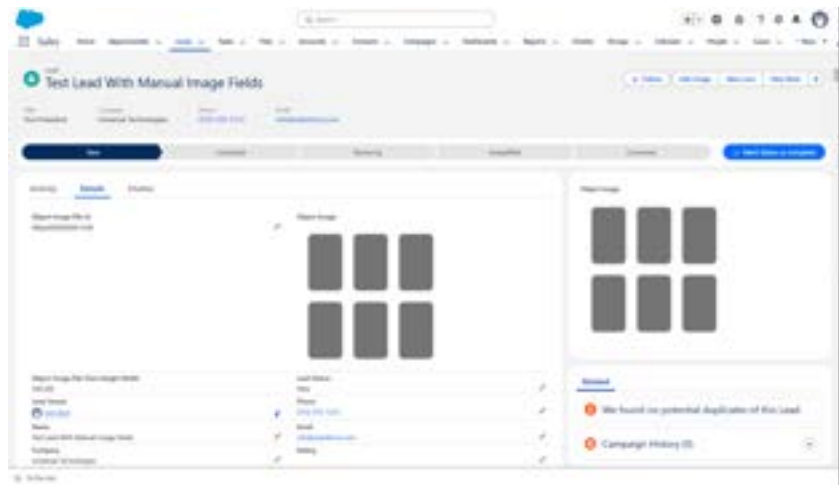
IF(OR(VALUE(LEFT(Object_Image_File_View_Height_Width__c,3))=0, VALUE(RIGHT(Object_Image_File_View_Height_Width__c,3))=0),
IMAGE(SUBSTITUTE(LEFT($Api.Partner_Server_URL_630, FIND( '/services', $Api.Partner_Server_URL_630)), 'my.sales', 'lightning.') +
'sfc/servlet.shepherd/version/renditionDownload?rendition=ORIGINAL_Png&operationContext=CHATTER&versionId='+Object_Image_File_Id__c, ''),
IMAGE(SUBSTITUTE(LEFT($Api.Partner_Server_URL_630, FIND( '/services', $Api.Partner_Server_URL_630)), 'my.sales', 'lightning.') +
'sfc/servlet.shepherd/version/renditionDownload?rendition=ORIGINAL_Png&operationContext=CHATTER&versionId='+Object_Image_File_Id__c, '',
VALUE(LEFT(Object_Image_File_View_Height_Width__c,3)), VALUE(RIGHT(Object_Image_File_View_Height_Width__c,3)))
```

```
)
)
)
)
)
```

Object_Image_File_View_Height_Width__c

```
IF(FIND(LEFT(Id, 3), $CustomMetadata.kltv_dev3__Object_Image_File_Setting__mdt.kltv_dev3__Image_Height_Width.kltv_dev3__ValueTextArea__c) = 0,
'000,000',
MID($CustomMetadata.kltv_dev3__Object_Image_File_Setting__mdt.kltv_dev3__Image_Height_Width.kltv_dev3__ValueTextArea__c, FIND(LEFT(Id, 3),
$CustomMetadata.kltv_dev3__Object_Image_File_Setting__mdt.kltv_dev3__Image_Height_Width.kltv_dev3__ValueTextArea__c) + 4, 7)
)
```

3. Then adding these new fields to layouts, etc.



Another slight modification when using the `objectImageDisplay` LWC is to change the `imageFieldName` parameter to "Object_Image_File_View__c" (remove "kltv_dev3__" namespace)

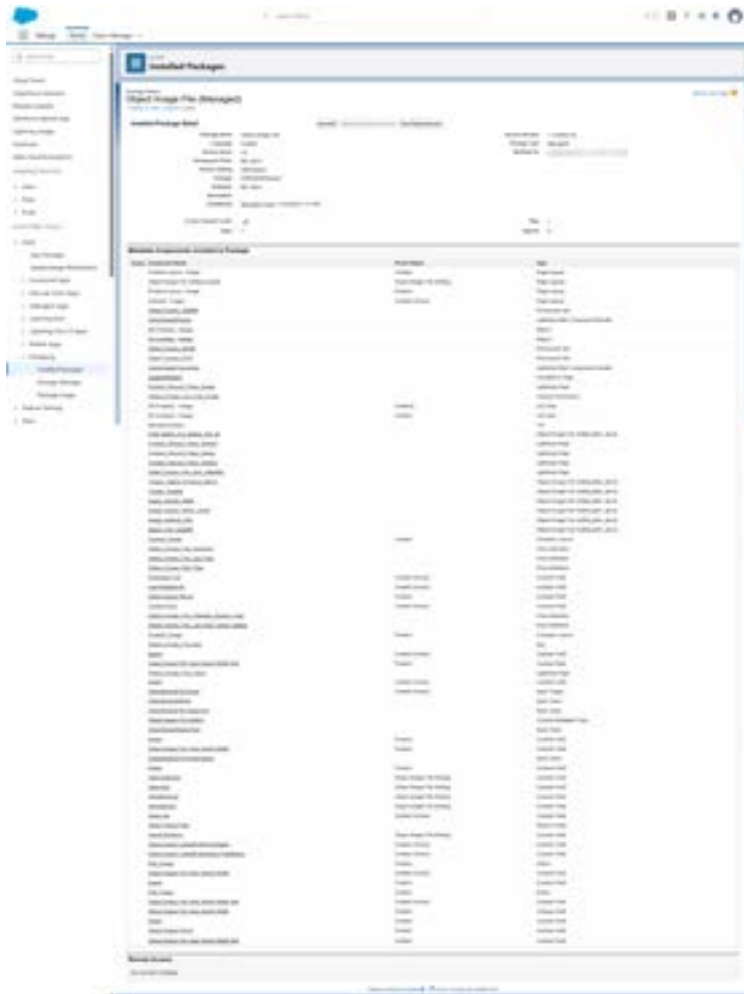
- **Multiple images per object** - load and display with a tile or carousel display component (LWC) to show multiple files at once.
- **Better SLDS2 styling** - change some styling to better match SLDS2, e.g. rounded/circle images for contact photos?

- **Easier bulk image export** - Easier options/steps to to bulk export images (ContentVersion) records with image files.
- **URL images** - Provide an option to use URLs in addition to files for images that are externally hosted/managed, e.g. Product images.
- Additional unmanaged package for potential future features: Account logos, custom object image, multiple images per object.

Metadata References

References to configuration metadata are written in `Courier New` font and can be found (and in some cases directly linked to) under the list of managed package components in:

Installed Package > Package Details - Object Image File (Managed) > Metadata Components Included in Package



Bulk Image Import (using data loading tools)

Image files can be bulk imported using data loading tools that support ContentVersion file uploads. Image files are matched against their related Product/Contact records using a (unique) reference field such as a Product Code or Contact Email field. This is similar to (but not the same as) importing related records with an External ID during an upsert.

Many data loading tools do not support file uploads, however the following tools have been found to support file uploads:

- Salesforce Data Loader client (<https://developer.salesforce.com/tools/data-loader>).
- Dataimporter (<https://www.dataimporter.io/>) including the free-tier plan.

CSV Files and Image Files for Import

Each CSV file contains a list of ContentVersion records with each record referring to its image file.

Example CSV files contain a list of ContentVersion records for a set of sample image files that are matched against sample Product/Contact records that are found in the default sample data created in new Developer orgs. These files are provided as a template for testing purposes, for demonstrating usage, and to provide a quick starting point.

Each line in the CSV contains the following key header fields:

- **Object_Image_LinkedEntityName_FieldName__c** - the related object and field name to lookup, e.g. "Contact.Name" will match the image file to the Contact object Name field.
- **Object_Image_LinkedEntityFieldValue__c** - the value of the field to match against the related object record, e.g. "Andy Young" will look for the ("first") Contact record with this Name and relate the image file to this record.
- **PathOnClient** - specifies the path and/or filename of the image file to load depending on the data loading tool being used, e.g. "lion-8225628_1280-240x320.jpg" for Dataimporter or "C:\Users\oscar\Downloads\Object_Image_Import\Object_Image_Import\malinois-3386471_1280.jpg" for Salesforce Data Loader.
- **VersionData** - the data for the image file, can be optional depending on the data loading tool being used. It is a mandatory field for the Salesforce Data Loader tool which requires the value to be set to the same value as PathOnClient.

- **Title** - the title of the file, can be optional depending on the data loading tool being used, it will usually default to the file name.

Example Bulk Import Steps for Salesforce Data Loader

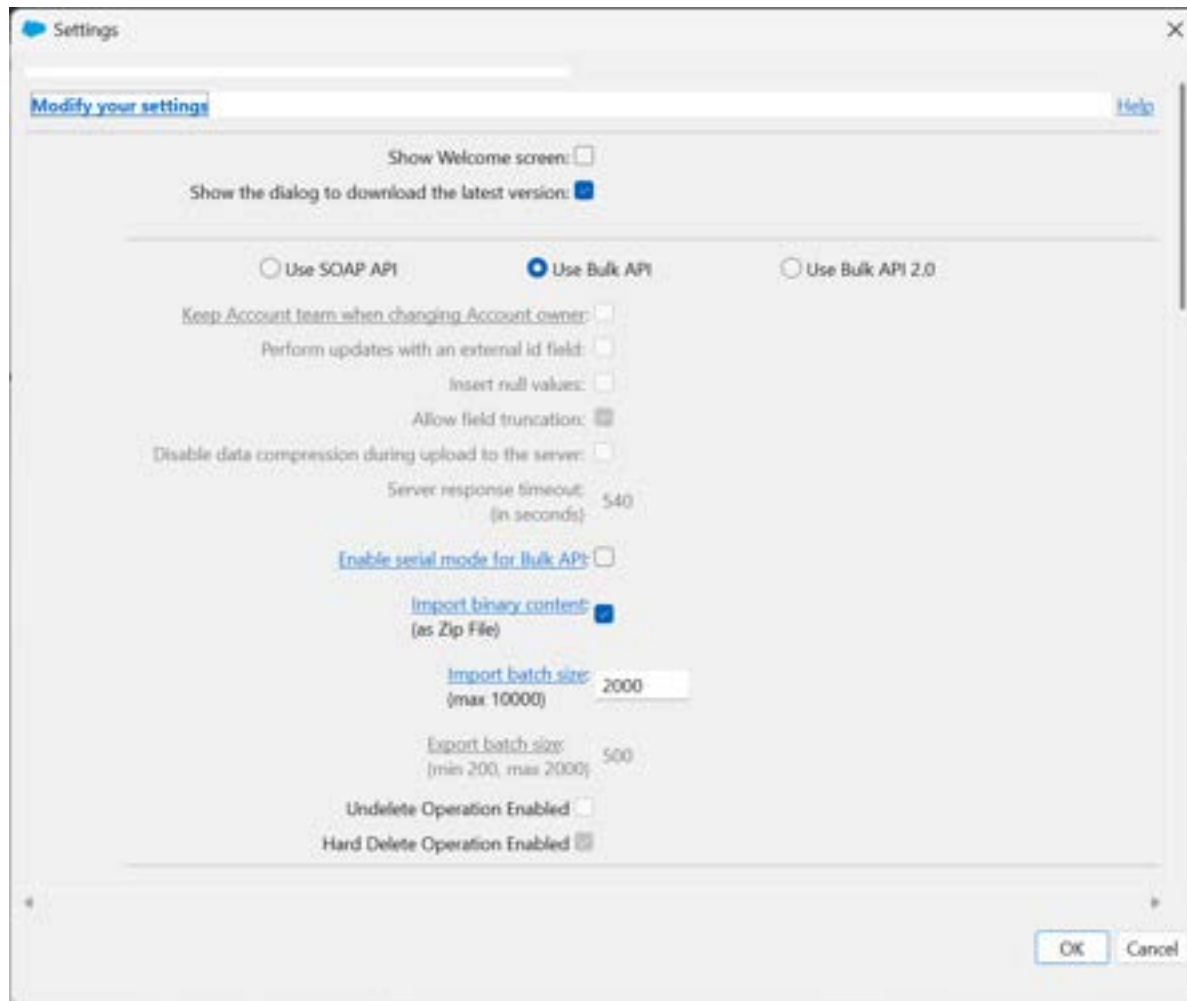
The following steps/screenshots provide example steps for loading image files against sample Product/Contact records found in new Developer orgs.

1. Prepare CSV Import File and Image Files

- In this example, use the “**Contact_Image_Import-dataloaderLocal.csv**” file to load sample images (ContentVersion records) against the (related) Contact object. Each image file is matched to the Contact’s Full Name.
- Extract the zip file to a folder on the local machine that can be accessed by the Salesforce Data Loader application. Note the path to the folder where the images are extracted and edit the CSV file records to use this path, otherwise the image files cannot be found and imported.
- For other orgs, edit the CSV file as needed, provide image files with valid image file name and path, check each ContentVersion record/image file has a matching related object record field value for the target import org.
- For sample Product images use the “**Product_Image_Import-dataloaderLocal.csv**” file.

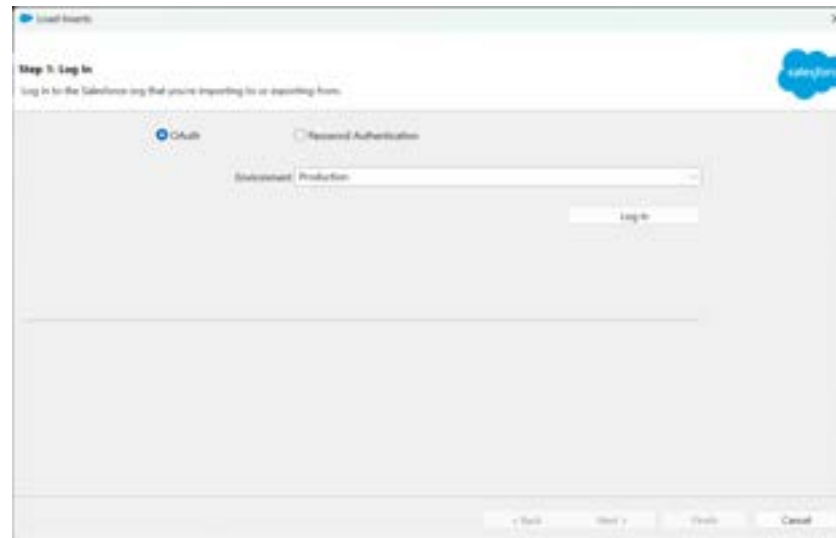
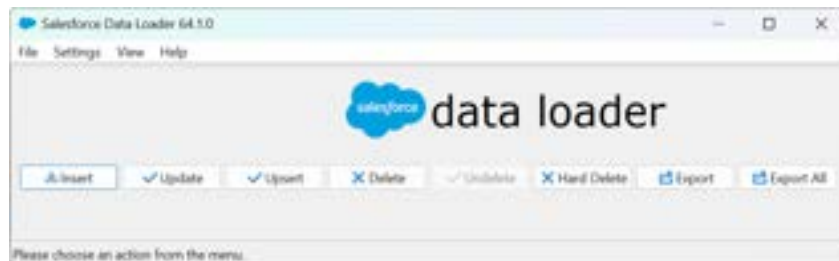
2. Change Data Loader Settings

- After installing/running Data Loader, use the Settings menu to change the tool to support Bulk API loading and image file binary content import as a Zip File.



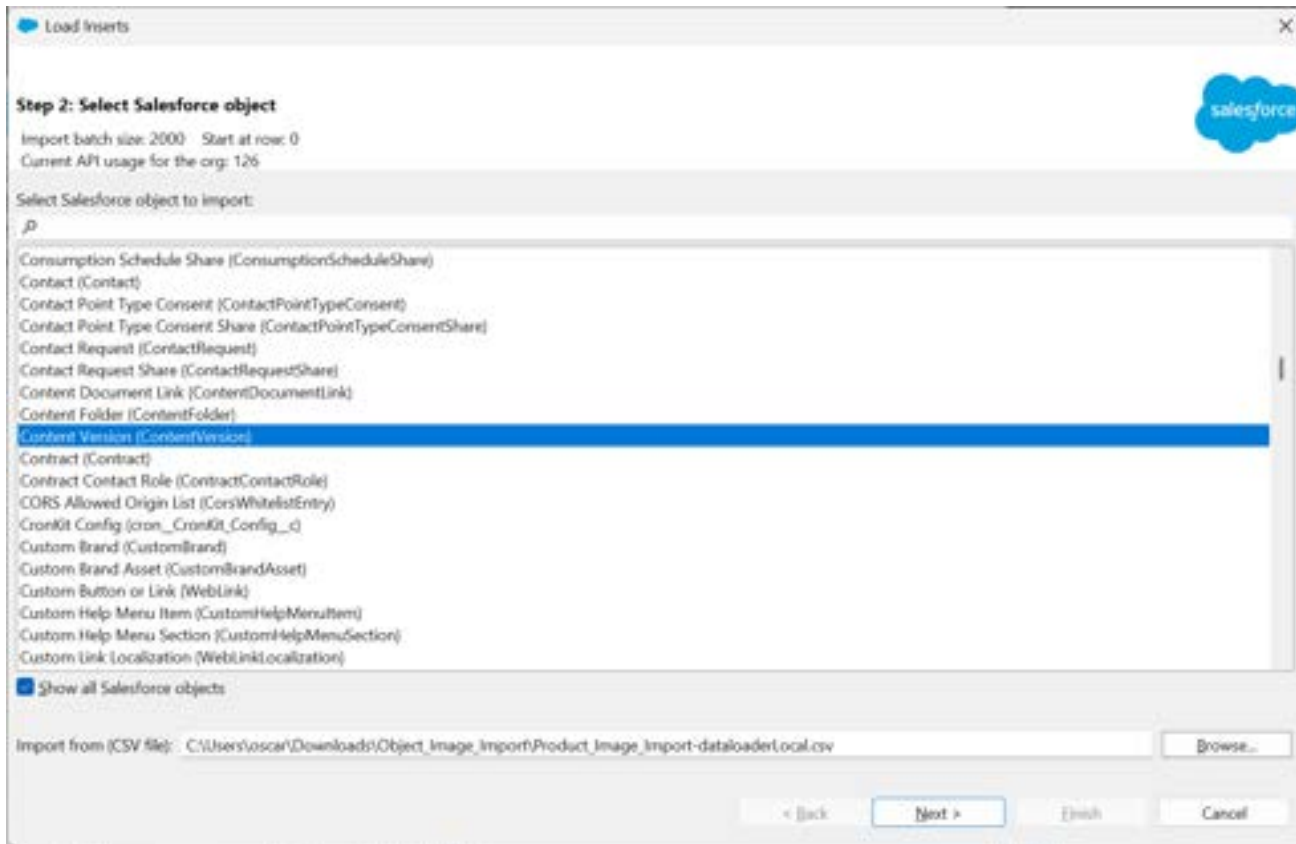
3. Select Insert operation and Login

- Use the Insert operation and login to the target org.



4. Select ContentVersion object for insert

- Check the “Show all Salesforce objects” checkbox and select the ContentVersion object for insert.
- Specify the full path to the prepared CSV import file above.



Load Inserts

Step 2: Select Salesforce object

Import batch size: 2000 Start at row: 0
Current API usage for the org: 126

Select Salesforce object to import:

Consumption Schedule Share (ConsumptionScheduleShare)
Contact (Contact)
Contact Point Type Consent (ContactPointTypeConsent)
Contact Point Type Consent Share (ContactPointTypeConsentShare)
Contact Request (ContactRequest)
Contact Request Share (ContactRequestShare)
Content Document Link (ContentDocumentLink)
Content Folder (ContentFolder)
ContentVersion (ContentVersion)
Contract (Contract)
Contract Contact Role (ContractContactRole)
CORS Allowed Origin List (CorsWhitelistEntry)
CronKit Config (cron__CronKit_Config__c)
Custom Brand (CustomBrand)
Custom Brand Asset (CustomBrandAsset)
Custom Button or Link (WebLink)
Custom Help Menu Item (CustomHelpMenuItem)
Custom Help Menu Section (CustomHelpMenuSection)
Custom Link Localization (WebLinkLocalization)

☒ Show all Salesforce objects

Import from (CSV file): C:\Users\oscar\Downloads\Object_Image_Import\Product_Image_Import-data-loader\local.csv Browse...

< Back Next > Finish Cancel

5. Skip Optional Relationship Step 2b

- No lookup relationship fields are needed.

Load Inserts

Step 2b: (Optional) relate using lookup field

Relationships of ContentVersion are listed below.
Select a related object and its lookup field if the CSV refers to the related object using the selected lookup field.

Relationship	Related Object	Lookup Field of Related Object
ContentBody	ContentBody	<Not selected>
ContentDocument	ContentDocument	<Not selected>
ExternalDataSource	ExternalDataSource	<Not selected>
FirstPublishLocation	Account	<Not selected>
Owner	User	<Not selected>

< Back **Next >** Finish Cancel

6. Specify CSV Field Mappings

- Select the “Choose an Existing Map” and specify the path to the example “**Object_Image_Import-dataloaderLocal.sdl**” file (which should be in the same location as the CSV file).

Load Inserts

Step 3: Mapping

Import batch size: 2000 Start at row: 0
Current API usage for the org: 126
API Limit for the org: 15,000
Bulk API version: 64.0

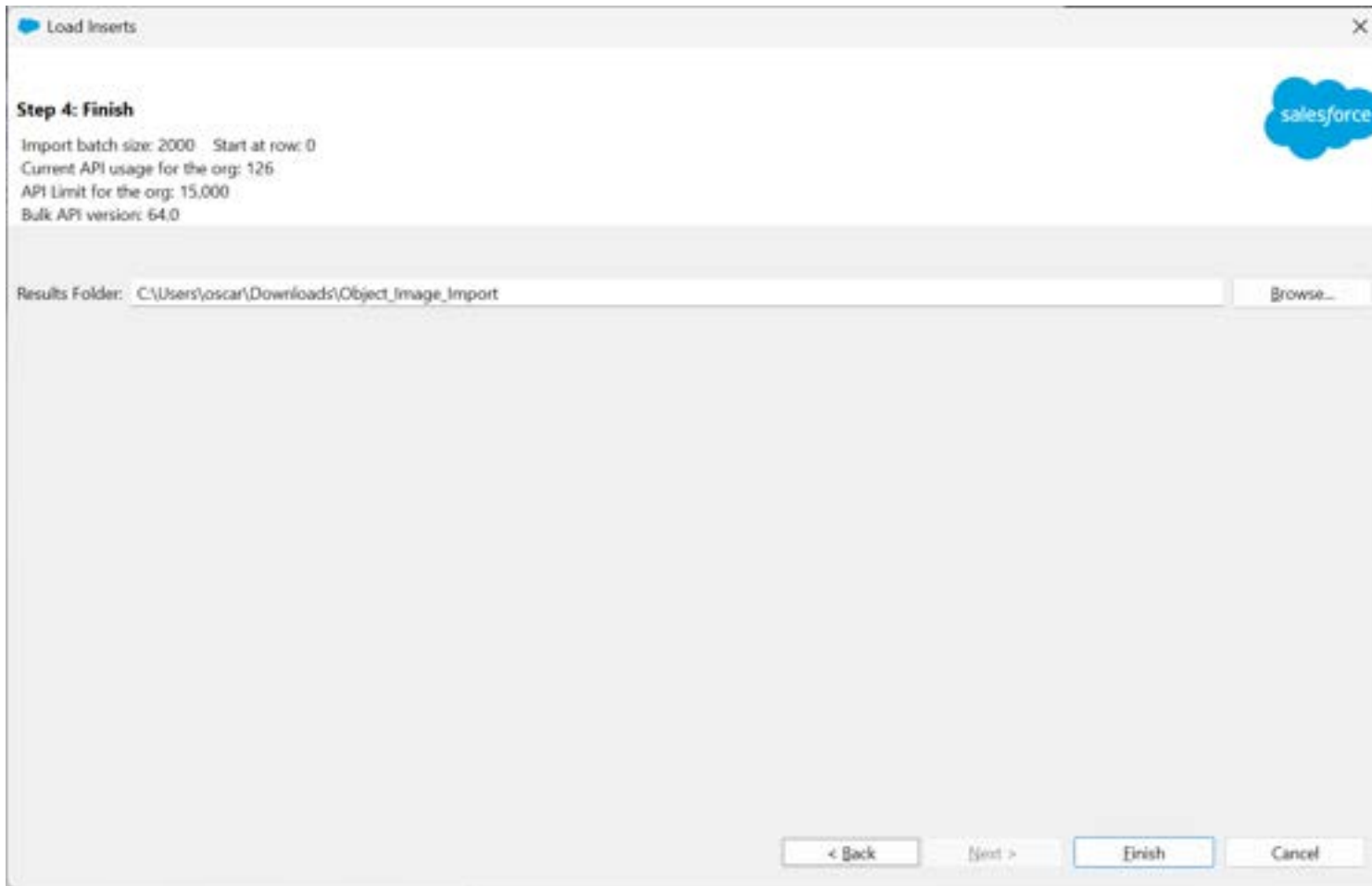
Choose an Existing Map Create or Edit a Map

CSV Column Header	Salesforce Object Field Name
Object_Image_LinkedEntityName_FieldName_c	ldtv_dev3__Object_Image_LinkedEntityName_FieldName_c
Object_Image_LinkedEntityFieldValue_c	ldtv_dev3__Object_Image_LinkedEntityFieldValue_c
PathOnClient	PathOnClient
VersionData	VersionData
Title	Title

< Back Next > Finish Cancel

7. Specify Path to Results Folder

- Enter the location path to a folder where the results, success and error logs will be placed.



Load Inserts

Step 4: Finish

Import batch size: 2000 Start at row: 0
Current API usage for the org: 126
API Limit for the org: 15,000
Bulk API version: 64.0

Results Folder:

8. Check Import Results

- The success CSV file will contain the loaded image records and the error CSV file will contain the records which failed to import and the error encountered.

The screenshot displays the Salesforce Import Wizard interface. A 'Step 4' dialog box indicates that the operation has fully completed, with 10 successful inserts and 0 errors. Below this, a 'Progress Information' dialog box shows 'Loading Using Bulk API insert' and 'Retrieving Bulk Job Results'. The main window shows the 'Results Folder' as 'C:\Users\user\Downloads'. At the bottom, a 'CSV Viewer' window displays a table of import results.

Row Number	ID	Object	Image_LinkedEntityName_FieldName_1	Object_Image_LinkedEntityName_Value_1	RefOnClient	VersionData	File	Status
1	068nu00000poCafAU	Contact	Name	Andy Young	C:\Users\...	C:\Users\...	Item 8227628_1280_148x12	Item Created
2	068nu00000poCafAU	Contact	Name	Arthur Song	C:\Users\...	C:\Users\...	animal-5146751_1280.jpg	Item Created
3	068nu00000poCafAU	Contact	Name	Ashley James	C:\Users\...	C:\Users\...	dog-4819324_1280.jpg	Item Created
4	068nu00000poCafAU	Contact	Name	Art Green	C:\Users\...	C:\Users\...	dog-1807549_1280.jpg	Item Created
5	068nu00000poCafAU	Contact	Name	Babara Levy	C:\Users\...	C:\Users\...	female-1246732_1280.jpg	Item Created
6	068nu00000poCafAU	Contact	Name	Julia Frank	C:\Users\...	C:\Users\...	cat-4152965_1280.jpg	Item Created
7	068nu00000poCafAU	Contact	Name	Jack Rogers	C:\Users\...	C:\Users\...	teagles-4166818_1280.jpg	Item Created
8	068nu00000poCafAU	Contact	Name	Jane Gray	C:\Users\...	C:\Users\...	cat-662689_1280.jpg	Item Created
9	068nu00000poCafAU	Contact	Name	John Bond	C:\Users\...	C:\Users\...	maltese-1386471_1280.jpg	Item Created
10	068nu00000poCafAU	Contact	Name	Josh Davis	C:\Users\...	C:\Users\...	animal-3471311_1280.jpg	Item Created

To open the CSV in the associated external program such as Microsoft Excel, click the button below.

Open in external program Close

9. Check Imported Images

- Use the “All Contacts - Image” list view to see the images loaded against the related Contact records.

Object Image File Home Manage Images Products Contacts Files Reports

Contacts

All Contacts - Image

32 items • Sorted by Name • Updated a few seconds ago

New Import Add to Campaign Send Email Printable View

Search this list...

	Image	Name	Account Name	Title	Email
1		Andy Young	Dickenson plc	SVP, Operations	a_young@dickenson.com
2		Arthur Song	United Oil & Gas Corp.	CEO	asong@uog.com
3		Ashley James	United Oil & Gas, UK	VP, Finance	ajames@uog.com
4		Aul Green	United Oil & Gas Corp.	CEO	agreen@uog.com
5		Babara Levy	Express Logistics and Transport	SVP, Operations	blevy@expresslit.net
6		Edna Frank	GenePoint	VP, Technology	efrank@genepoint.com
7		Jack Rogers	Burlington Textiles Corp of America	VP, Facilities	jrogers@burlington.com
8		Jane Gray	University of Arizona	Dean of Administration	jane_gray@uoa.edu
9		John Bond	Grand Hotels & Resorts Ltd	VP, Facilities	bond_john@grandhotels.com
10		Josh Davis	Express Logistics and Transport	Director, Warehouse Mgmt	jdavis@expresslit.net
11		Lauren Boyle	United Oil & Gas Corp.	SVP, Technology	lboyle@uog.com
12		Liz O'Crut	United Oil & Gas, Singapore	VP, Production	locrut@uog.com
13		Rosea Gonzalez	Edge Communications	SVP, Procurement	rosea@edge.com

Example Bulk Import Steps for Dataimporter

The following steps/screenshots provide example steps for loading image files against sample Product/Contact records found in new Developer orgs.

1. Prepare CSV Import File and Image Files

- In this example, use the “**Contact_Image_Import.csv**” file to load sample images (ContentVersion records) against the (related) Contact object. Each image file is matched to the Contact’s Full Name.
- All images should be in a zip file for upload, use the original “**Object_Image_Import.zip**” already provided.
- For sample Product images use the “**Product_Image_Import.csv**” file.

2. Connect Instance (Org)

- Use the Connections > Instances menu item to connect the target org with the [Connect Org] button.

3. Create New Insert Job

- Use the Process > Jobs menu item to create a new insert job with the [New Job] button. Specify the target Instance (org) and CSV as the source.

dataimporter

Dashboard

Process

Jobs

Migration Templates

Sandbox Seeding

Formulas

History

Connections

Integrations

Instances

Account

Settings

Billing

Support

Help Center

Email Us

Logout

New Job

Name

Contact Image Import

Instance

dev3@kollektivinfo (Dev3 - production)

Type

Insert

Upsert

Update

Delete

Export

Insert will always create new records.

Source

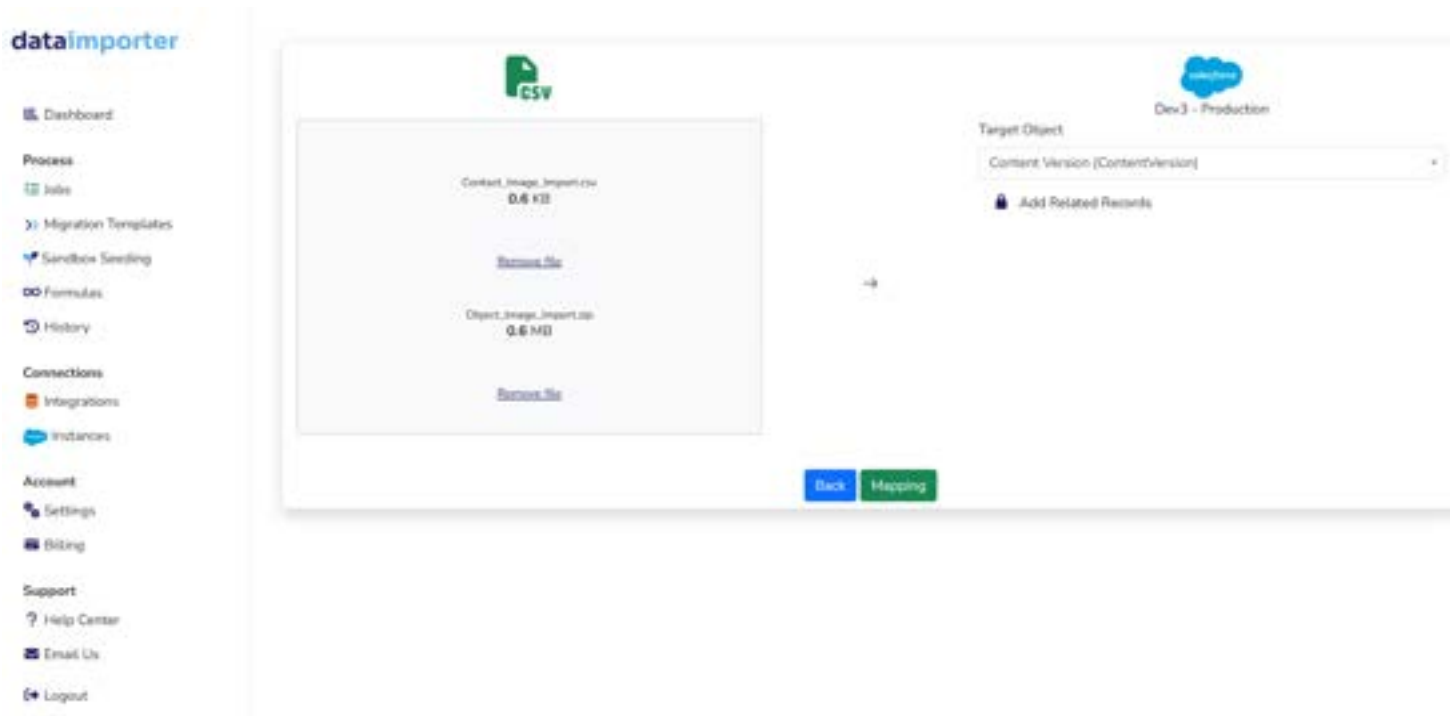
CSV

> Advanced

Save

4. Specify CSV and Zip Files for Import

- Select both the CSV file and Zip file for import.
- Select “Content Version” as the Target Object.



5. Accept Mapping

- Accept the default field mappings by clicking the [Next] button.

dataimporter

Dashboard

Process

Jobs

>> Migration Templates

Sandbox Seeding

Formulas

History

Connections

Integrations

Instances

Account

Settings

Billing

Support

? Help Center

Email Us

Logout


Object_Image_Import.zip

Example

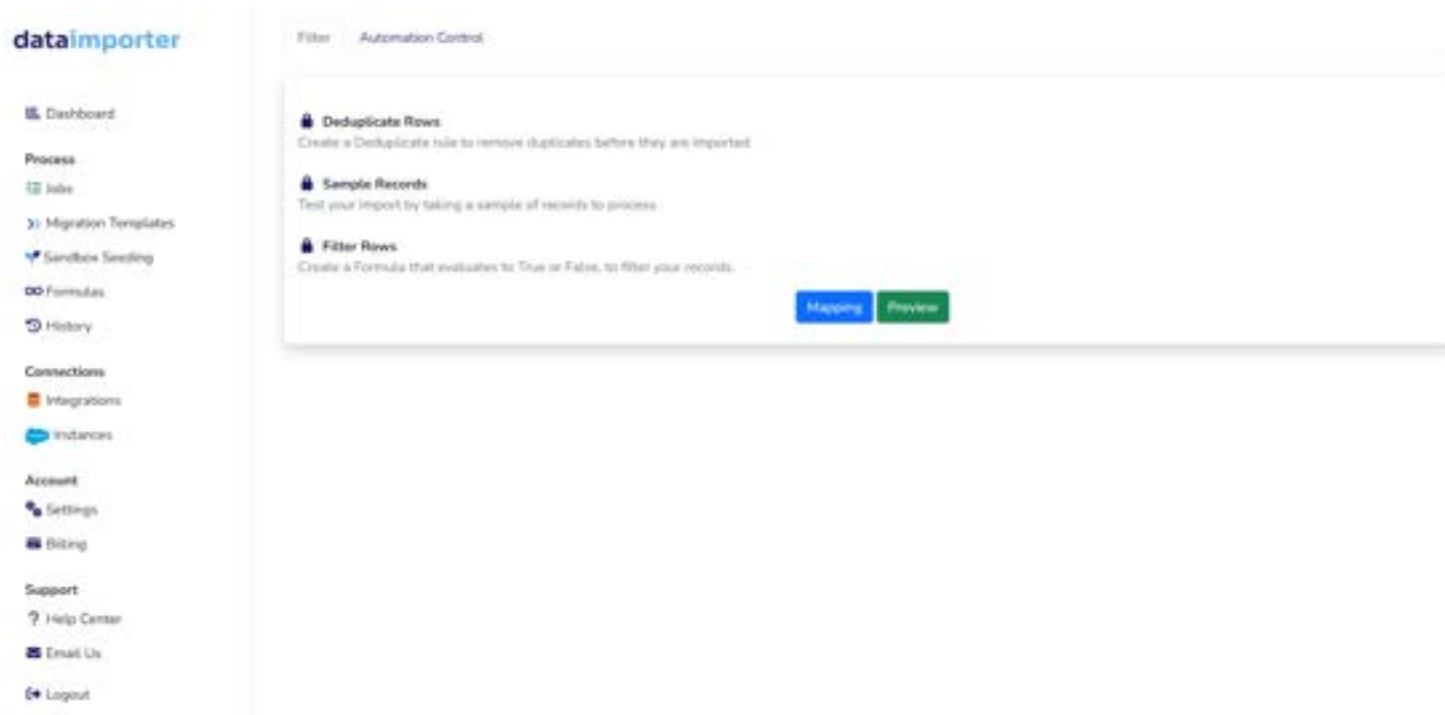

Dev3 - Production: ContentVersion

Object_Image_LinkedEntityName_FieldName__c	Contact.Name	Object Image LinkedEntityName.FieldName (kltv_dev3__O...	✖
Object_Image_LinkedEntityFieldValue__c	Andy Young	Object Image LinkedEntityFieldValue (kltv_dev3__Object_I...	✖
PathOnClient	lion-8225628_1280-24...	Path On Client (PathOnClient)	✖
Title	nan	Title (Title)	✖

[Back](#) [Next](#)

6. Go to Preview

- Go straight to the preview by clicking the [Preview] button.



7. Submit the Job

- Submit the insert job by clicking the [Submit] button and then the [Run Now] button.

dataimporter

Dashboard

Process

Jobs

Migration Templates

Sandbox Seeding

Formulas

History

Connections

Integrations

Instances

Account

Settings

Billing

Support

Help Center

Email Us

Logout

10 records set for insert.

Object_Image_Import.xls	Example	Example	Example	ContentVersion
Object_Image_LinkedEntryName...	ContactName	ContactName	ContactName	kitx_dev3...Object_Image_Link...
Object_Image_LinkedEntryFiel...	Avi Green	Jane Grey	Ashley James	kitx_dev3...Object_Image_Link...
PathOnClient	dog-3801569_1290.jpg	cat-680689_1290.jpg	dog-4820028_1290.jpg	PathOnClient
Title				Title

Clear

Submit

Submit Job

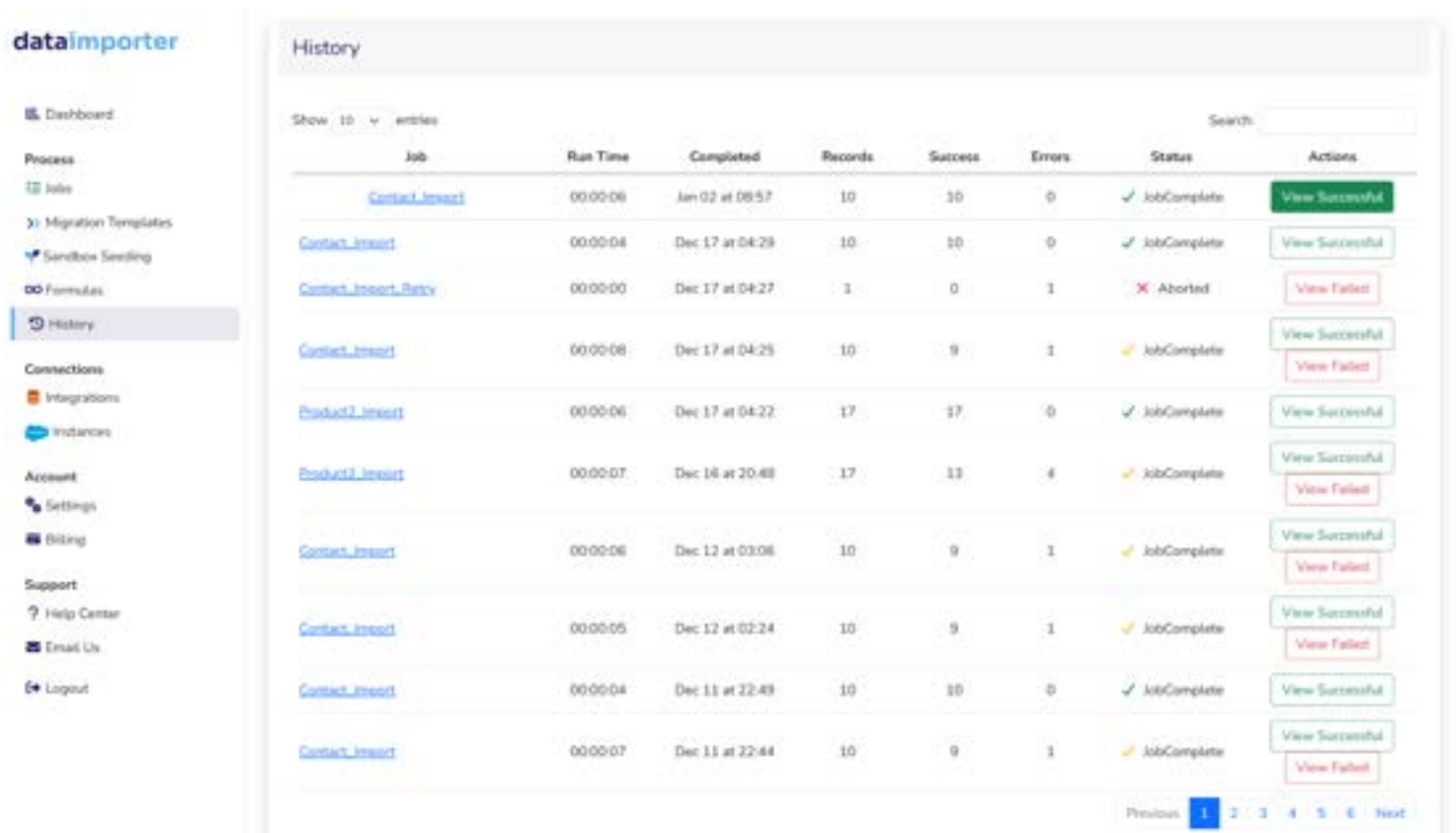
Do you want to run the job **Contact_Import** now?

This will insert **Contact Version** records in the org **dev3 - Production**.

[Preview](#)[Run Now](#)

8. Check Import Results

- Check the Job status is complete and use the [View Successful] or [View Failed] buttons to review the results.



The screenshot displays the 'History' page of the dataimporter application. The left sidebar contains navigation links for Dashboard, Process, Jobs, Migration Templates, Sandbox Seeding, Formulas, History (selected), Connections, Integrations, Instances, Account, Settings, Billing, Support, Help Center, Email Us, and Logout. The main content area shows a table of import jobs. The table has columns for Job, Run Time, Completed, Records, Success, Errors, Status, and Actions. The 'Status' column shows 'JobComplete' with a green checkmark icon. The 'Actions' column contains buttons for 'View Successful' (green) and 'View Failed' (red). The table is paginated, showing 10 entries per page, with a search bar and a 'Previous' button.

Job	Run Time	Completed	Records	Success	Errors	Status	Actions
Contact_Import	00:00:06	Jan 02 at 08:57	10	10	0	✓ JobComplete	View Successful
Contact_Import	00:00:04	Dec 17 at 04:29	10	10	0	✓ JobComplete	View Successful
Contact_Import_Retry	00:00:00	Dec 17 at 04:27	1	0	1	✗ Aborted	View Failed
Contact_Import	00:00:08	Dec 17 at 04:25	10	9	1	✓ JobComplete	View Successful View Failed
Product1_Import	00:00:06	Dec 17 at 04:22	17	17	0	✓ JobComplete	View Successful
Product1_Import	00:00:07	Dec 16 at 20:48	17	13	4	✓ JobComplete	View Successful View Failed
Contact_Import	00:00:06	Dec 12 at 03:08	10	9	1	✓ JobComplete	View Successful View Failed
Contact_Import	00:00:05	Dec 12 at 02:24	10	9	1	✓ JobComplete	View Successful View Failed
Contact_Import	00:00:04	Dec 11 at 22:49	10	10	0	✓ JobComplete	View Successful
Contact_Import	00:00:07	Dec 11 at 22:44	10	9	1	✓ JobComplete	View Successful View Failed

Not happy with the insert? Click on Rollback below to delete the records automatically.



Successful records

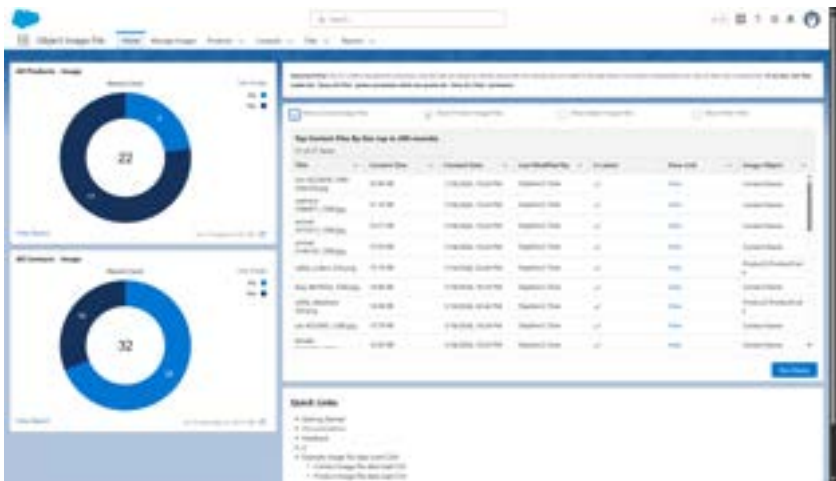
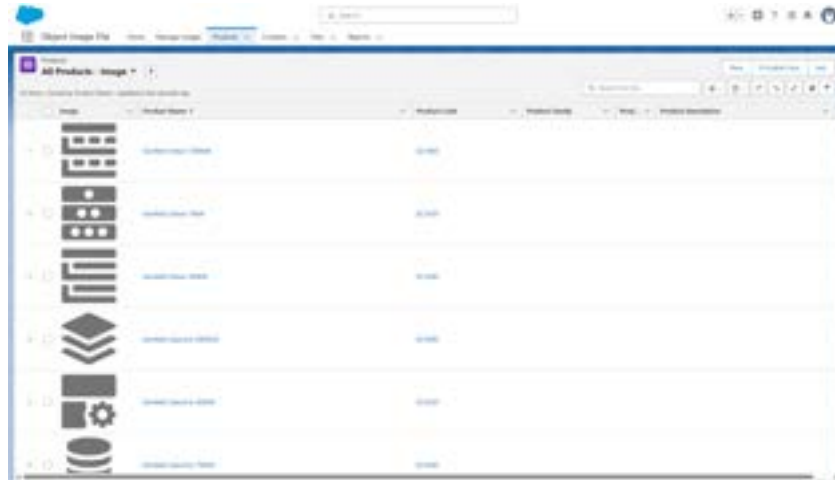
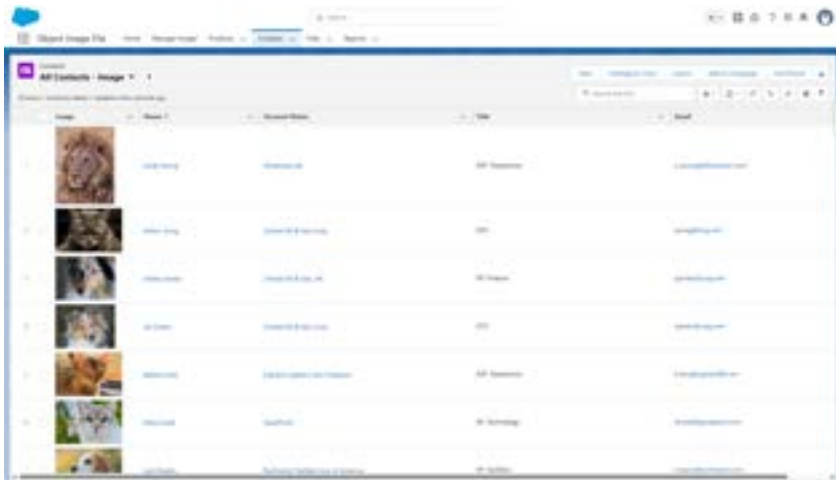
Download

Rollback

Object_Image_LinkedEntityName_FieldName__c	Object_Image_LinkedEntityFieldValues__c	PathOnClient	Title	sf_id	Actions
Contact.Name	Andy Young	lion-8225628_1280-240x320.jpg		068am00000Hv0XAAJ	View in Salesforce
Contact.Name	Arthur Song	animal-5146153_1280.jpg		068am00000Hv0XnAAJ	View in Salesforce
Contact.Name	Ashley James	dog-4810024_1280.jpg		068am00000Hv0XnAAJ	View in Salesforce
Contact.Name	Avi Green	dog-3801569_1280.jpg		068am00000Hv0XnAAJ	View in Salesforce
Contact.Name	Babara Levy	female-1246732_1280.jpg		068am00000Hv0XpAAJ	View in Salesforce
Contact.Name	Edna Frank	cat-4552965_1280.jpg		068am00000Hv0XpAAJ	View in Salesforce
Contact.Name	Jack Rogers	beagle-4168818_1280.jpg		068am00000Hv0XnAAJ	View in Salesforce

9. Check Imported Images

- Use the “All Contacts - Image” list view to see the images loaded against the related Contact records.



END