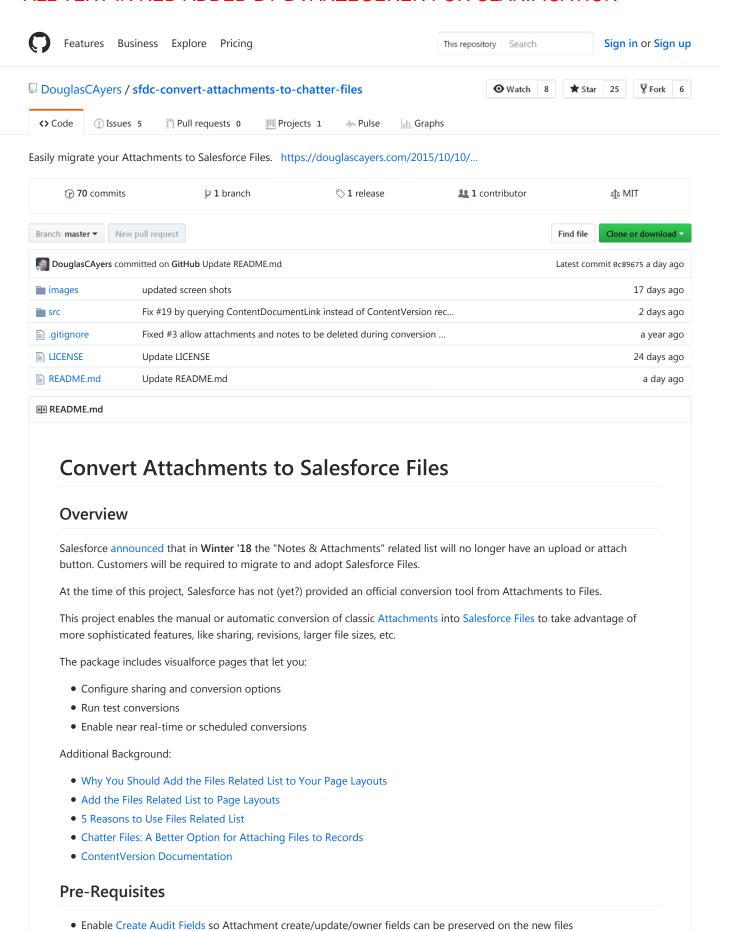
ALL TEXT IN RED ADDED BY DVANZEGEREN FOR CLARIFICATION



Setup>Build>Customize>UserInterface

| Setup | |
|--|--|
| | |
| | |
| | |
| | |
| | |
| Enable "Set Audit Fields upon Record Creation" and "Upda | ate Records with Inactive Owners" User Permissions |
| | |
| | |
| Enable Advanced Setup Search (Beta) | |

• Enable Email-to-Case. You don't have to begin using it, just have it enabled. The code references the EmailMessage object. Setup>Build>Customize>Cases>Email-to-Case



Packaged Release History

Release 1.1 (latest) $_{\mbox{Use}}$ the latest version of app

- Install package
 - Production URL
 - Sandbox URL
- Fixes Issue 19 inconsistency when detecting if a file has already been converted or not due to special access rules on ContentVersion object.
- If error log records are generated then email is sent to all Salesforce Users listed in Setup | Email Administration | Apex Exception Email notifying them to review the error log records. Misc Notes:

Release 1.0

- Install package
 - Production URL
 - Sandbox URL
- Initial managed package offering
- Split out Notes Conversion into its own project

loading/running_issues, turn on Apex
Exception emails:
Setup>Administer>Email
Administration>Apex Exception Email
AND set Email Deliverability to All
Emails: Setup>Administer>Email

Administration>Deliverability

To capture debug logs in case of

Installing the Source Code (Developers)

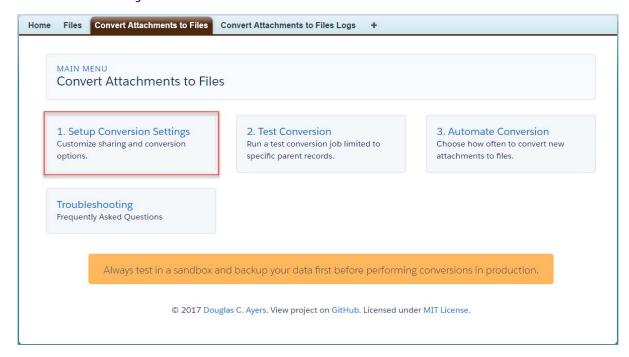
You may install the unmanaged code from GitHub and make any desired adjustments. You are responsible for ensuring unit tests meet your org's validation rules and other requirements.

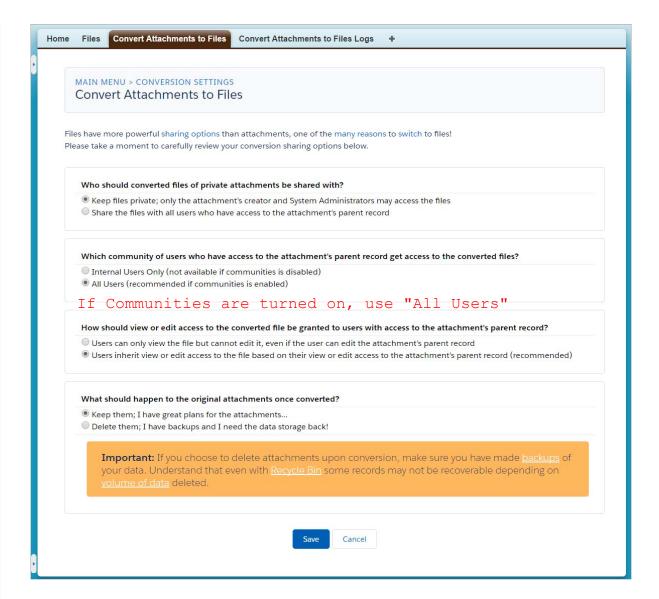
• Deploy from Github

Getting Started

Performed as Pre-Requisite:Setup>Build>Customize>UserInterface

- 1. Enable setting Create Audit Fields so Attachment create/update/owner fields can be preserved on the new files
- 2. Enable setting Files uploaded to the Attachments related list on records are uploaded as Salesforce Files, not as attachments Setup>Build>Customize>Salesforce Files>Settings>General Settings
- 3. Add "Files" related list to your page layouts (e.g. Accounts, Contacts, Tasks, Events, etc.)
- 4. Deploy the package using one of the installation links above
- 5. Assign yourself the permission set Convert Attachments to Files then switch to the app by the same name
- 6. On the **Convert Attachments to Files** tab page, click on **Setup Conversion Settings** to configure sharing and conversion behavior
- 7. Perform a test conversion
- 8. Consider automating conversion





FAQ

Max Documents or Versions Published Governor Limit

When converting classic Notes & Attachments the new data is stored in the Contentversion object. There is a limit to how many of these records can be created in a 24 hour period. If you have a lot of Notes & Attachments to convert plan around this limit and split the work across multiple days.

Field is not writeable: ContentVersion.CreatedById

When you deploy the package you might get error that files are invalid and need recompilation and one of the specific messages might say "Field is not writeable: ContentVersion.CreatedById". The conversion tool tries to copy the attachment's original created and last modified date/user to the converted file. To do so then the "Create Audit Fields" feature must be enabled. Please see this help article for instructions enable this feature.



This app can't be installed.

There are problems that prevent this package from being installed.

Field is not writeable: ContentVersion.CreatedById ConvertAttachmentsToFilesService: Field is not writeable: ContentVersion.CreatedById

Dependent class is invalid and needs recompilation: ConvertAttachmentsToFilesTrigger: Dependent class is **dca_cnvrt_file.convertattachmentstofilesqueueable:** invalid and needs recompilation:

line 12, column 18: Invalid type: dca_cnvrt_file.convertattachmentstofilesqueueable: line

ConvertAttachmentsToFilesService.ConversionResult12, column 18: Invalid type:

ConvertAttachmentsToFilesService.ConversionResult

| Setup |
|--|
| |
| |
| |
| |
| |
| ☑ Enable "Set Audit Fields upon Record Creation" and "Update Records with Inactive Owners" User Permission |
| ☑ Enable Custom Object Truncate |
| ☑ Enable Improved Setup User Interface i |
| ☑ Enable Advanced Setup Search (Beta) i |
| |

Visibility InternalUsers is not permitted for this linked record.

When the conversion tool shares the file to the attachment's owner and parent record the **ContentDocumentLink.Visibility** field controls which community of users, internal or external, may gain access to the file if they have access to the related record.

When communities are **enabled** then both picklist values Allusers and Internalusers are acceptable. When communities are **disabled** then only the picklist value Allusers is acceptable.

This error usually means communities are **disabled** in your org and you're trying to set the visibility of the converted files to Internallisers

To fix then either (a) enable communities or (b) change the visibility option to Allusers.

FIELD_INTEGRITY_EXCEPTION, Owner ID: id value of incorrect type: 035xxxxxxxxxxxxxxxx: [OwnerId]

Prior to Spring '12, Salesfore customers could have Self-Service Portals, which pre-date the modern Communities we have today. This error means the Attachment is owned by a Self-Service User and ContentVersions cannot be owned by them. You may want to consider changing ownership of those attachments to actual user records whose IDs start with **005** prefix.

Are there any objects that don't support attachment conversion?

Yes, the EmailMessage object. Although technically you can convert their attachments to files, you cannot **share** the files to the email message records. You will receive error INSUFFICIENT_ACCESS_OR_READONLY, You can't create a link for Email Message when it's not in draft state.: [LinkedEntityId].

How are private attachments converted?

Classic Notes & Attachments have an IsPrivate checkbox field that when selected makes the record only visible to the owner and administrators, even through the Note or Attachment is related to the parent entity (e.g. Account or Contact). However, ContentVersion object follows a different approach. Rather than an explicit 'IsPrivate' checkbox it uses a robust sharing model, one of the reasons to convert to the new Files to begin with! In this sharing model, to make a record private then it simpy isn't shared with any other users or records. The caveat then is that these unshared (private) Files do not show up contextually on any Salesforce record. By sharing the new File with the original parent record then any user who has visibility to that parent

record now has access to this previously private attachment. Therefore, when converting you have the option to specify whether the private attachments should or should not be shared with the parent entity once converted into new File.

If I run the conversion multiple times, do duplicate files get created for the same attachments?

No, no duplicate files should be created once an attachment has been converted once. When attachments are converted into files we store the Attachment.ID in the ContentVersion.Original_Record_ID_c field for tracking purposes. The conversion logic first checks if there exist any files that have been stamped with the attachment id, if yes then we skip converting that attachment again.

Of course, if you choose the conversion option to delete the attachments upon conversion then no such attachment would exist the second time around. But if you choose to keep the attachments post conversion they will not be converted again if you run conversion process multiple times.

Disclaimer

This is not an official conversion tool by salesforce.com to migrate Attachments to Salesforce Files. This is a personal project by Doug Ayers to assist customers in migrating to and adopting Salesforce Files. Although this tool has been successfully tested with several customers since 2015 that have between dozens to tens of thousands of attachments, please do your own due diligence and testing in a sandbox before ever attempting this in production.

Always make a backup of your data before attempting any data conversion operations.

You may read the project license here.

Special Thanks

- Arnab Bose (@ArBose), Salesforce Product Manager
- Haris Ikram (@HarisIkramH), Salesforce Product Manager
- David Mendelson, Salesforce Product Manager
- Neil Hayek, Salesforce Chatter Expert
- Arthur Louie, Salesforce Chatter Expert
- Rick MacGuigan, a very helpful early adopter and tester!
- And to everyone who has provided feedback on this project to make it what it is today, thank you!