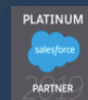




# Smart Files Download User Manual



**CMMISVC/3<sup>SM</sup>**  
CMMISVC 2.0 / Exp. 2023-12-21 / Appraisal #52308





## Application Overview

Currently, Lightning Experience Files support individual file downloads. Every time the user wants to download a file he needs to go to the respective file, click on the action button, and then click on the download.

Consider the scenario where the user wants to download more than one file, then he needs to repeat the above process.

- Features
  - ✓ Supports multiple file downloads.
  - ✓ Supports multiple file deletions.
  - ✓ Download as separate files or as a Zip file in one go.
  - ✓ Supports in the Lightning Experience and Lightning Community record pages.
  - ✓ Drag and drop in the Lightning page or use it as a quick action button.

## Prerequisite

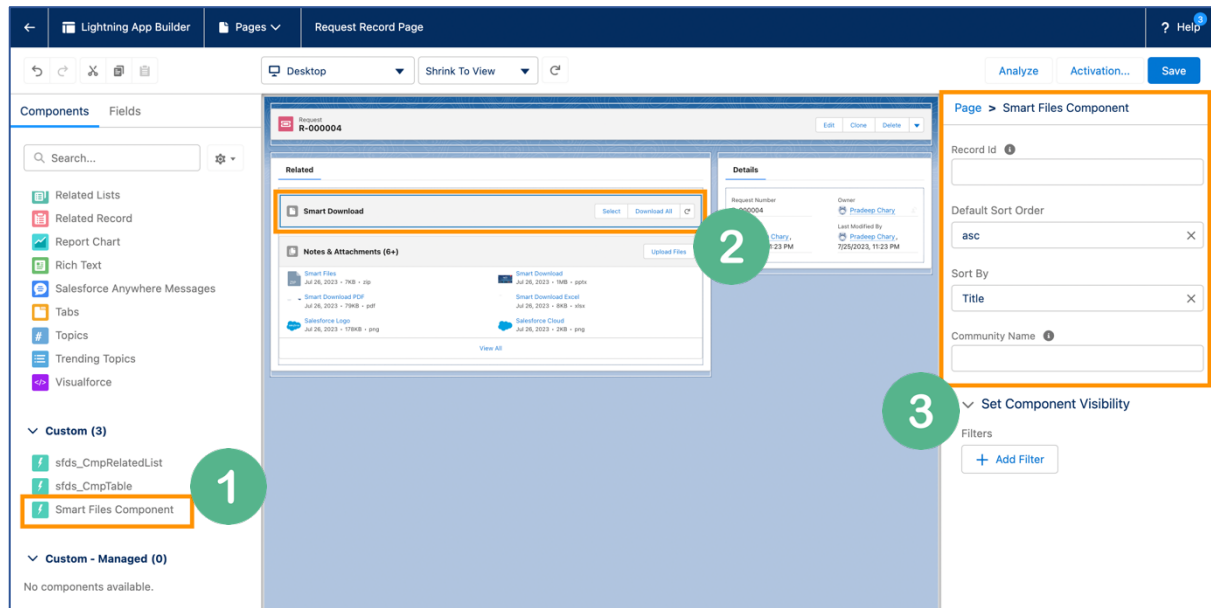
Install the application package in your org and ensure My Domain is enabled in it. If not, please refer to the Salesforce trailhead module 'Customize Your Login Process with My Domain Learning Objectives' [link](#).

- Scenario 1: Lightning Experience Record Page
- Scenario 2: Lightning Quick Action
- Scenario 3: Lightning Community Record Pages

## Scenario 1: Lightning Experience Record Page

Open the Lightning page of the Record page where the component needs to be placed.

- Step 1 - We can do this in two ways
  - Navigate to Setup - Search for Lightning App Builder - Select the required Lightning page.
  - Navigate to Record Page where we place the component - Click on the Setup Icon - Edit page.
- Step 2
  - Drag and drop the component ' Smart Files Component ' on the layout as shown in the screenshot below.



- Step 3 - Check the properties section.

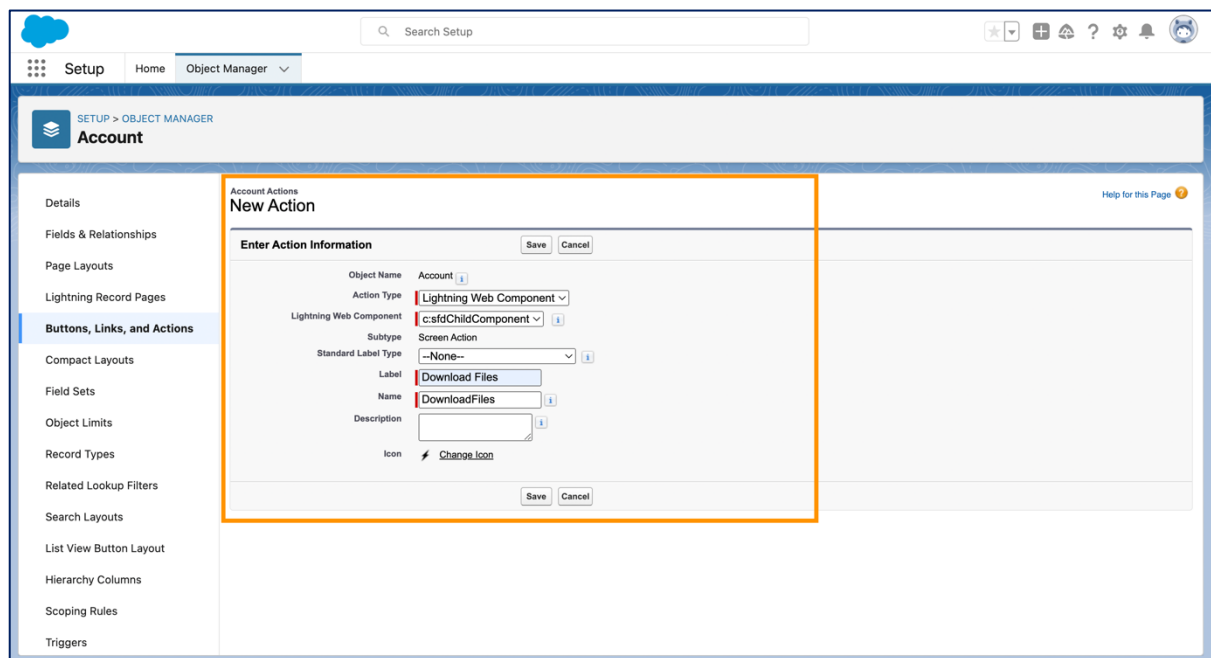
| Property Label     | Description                                                                                                                                                                                                                                                                                      | Default Value |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| Record Id          | Required if the component is placed on a community. The value will be {!recordId}                                                                                                                                                                                                                |               |
| Default Sort Order | The default order of sorting on the load of the table.<br>Available values:<br>- asc (Ascending),<br>- desc (Descending)                                                                                                                                                                         | asc           |
| Sort By            | The default sort by field on the load of the table.<br>- If the table should be on Title, please select Title.<br>- If the table should be on Last Modified Date, please select Last Modified Date.                                                                                              | Title         |
| Community Name     | If the component is placed under the community then the community name should be mentioned here. If it is not mentioned, it results in an error.<br>For Example, if your Community Site name is Account Portal and the URL provided is accountportal, then the value here will be accountportal. |               |

- Step 4 - Save and Activation.

## Scenario 2: Lightning Quick Action

The component can be placed under a lightning-quick action button. To set this up, please follow the below steps.

- Step 1
  - Navigate to Setup - Object Manager - Select the object where you want to place this component - Button, Links, and Actions.
- Step 2
  - Click on New Action, select/enter the details as shown below, and Save.



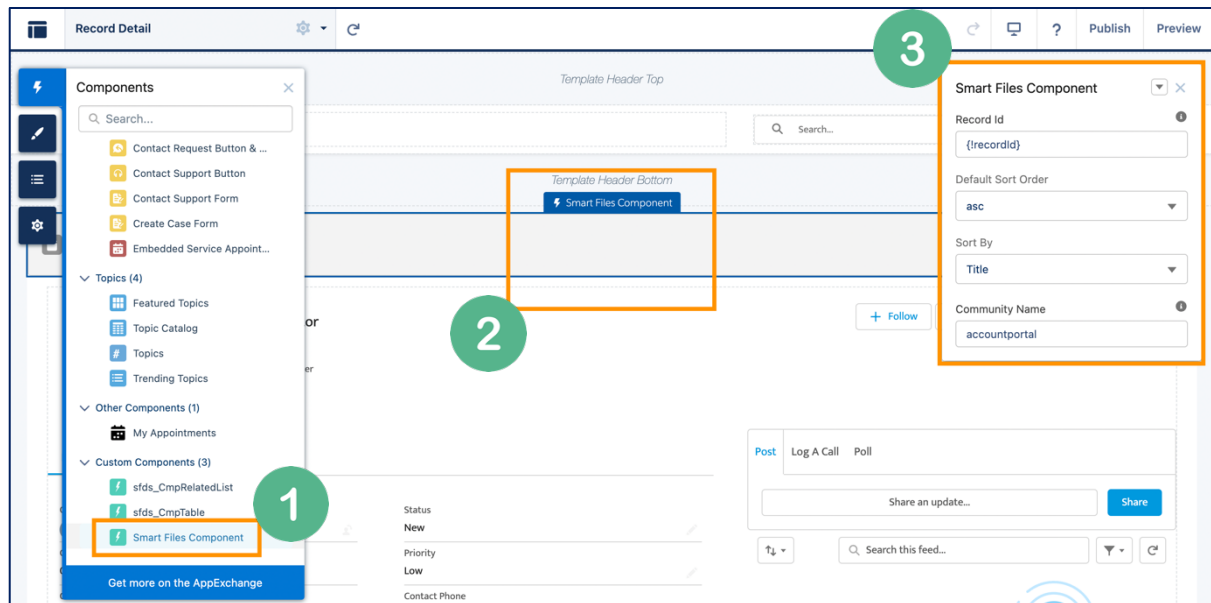
| Enter Action Information | Value                                     |
|--------------------------|-------------------------------------------|
| Action Type              | Lightning Web Component                   |
| Lightning Component      | sfdChildComponent                         |
| Standard Label Type      | None                                      |
| Label                    | Button name to be displayed on the layout |
| Name                     | API Name                                  |

- Step 3 - Add the button to the layout
  - Go to Page Layout, select the page layout where the button needs to be added.
  - Under Mobile & Lightning actions, select the button just created and drag-drop on the layout.
- Step 4 - Save and view under the layout quick actions.

## Scenario 3: Lightning Community Record Page

The component can be placed under a community record page. To set this up, please follow the below steps. Note that communities need to be enabled for this.

- Step 1 - Navigate to Setup - Search for All Communities - Under the selected Community click Builder.
- Step 2 - Go to the Record Detail page, drag and drop the component 'Smart Files Component' on the layout as shown in the screenshot below.

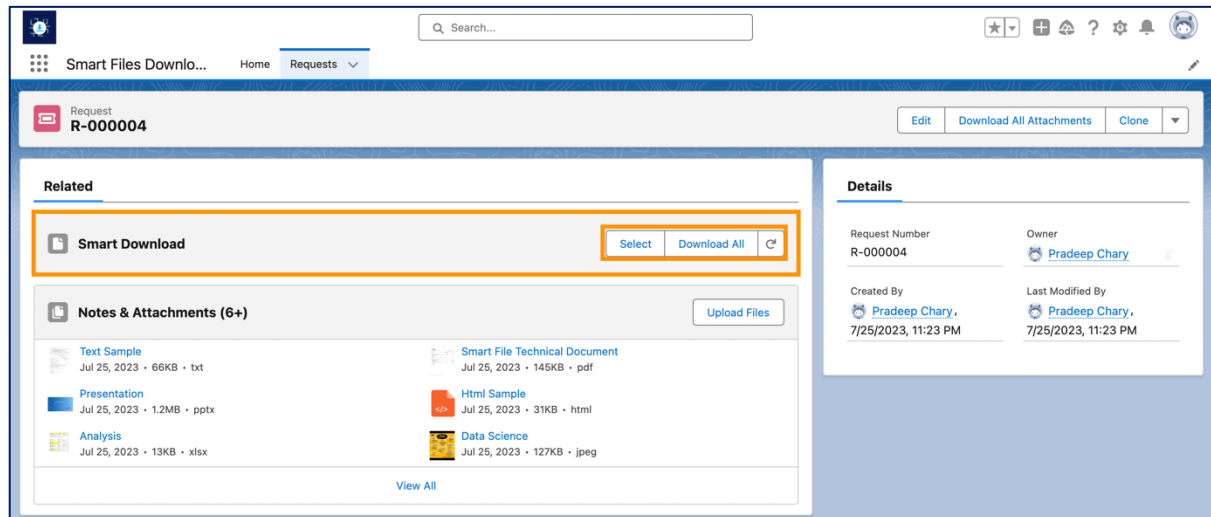


- Step 3 - Update the properties as shown under Scenario 1, step 3.
  - Important property is Community Name, and it is mandatory to update it.
- Step 4 - Publish the Community.
  - Under the Community profile, provide access to the Apex class 'slds\_ClassDownloadAll'.

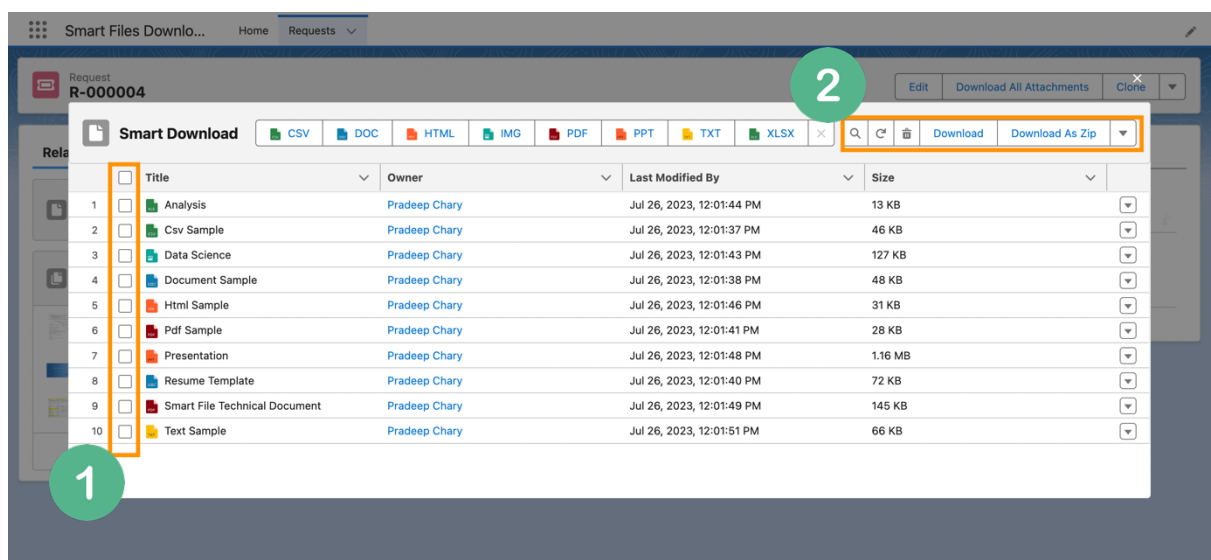
The setup is complete. Now, how this works?

Let's open the component as a related list under the Lightning Record page like below. As we see, the component is a related list and it is named Smart Download. We have three buttons under the related list.

- Select - To open a modal that displays all the Files available under the record.
- Download All - On click, all the files available under the record will be downloaded in the zip format.
- Refresh Icon - If some new files are added then the refresh button is available to get the new list.



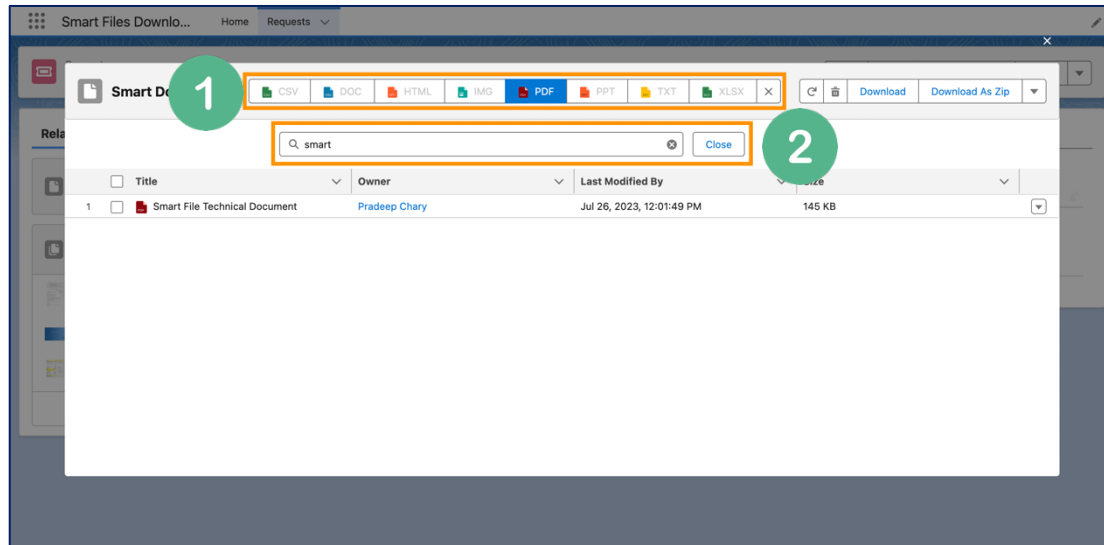
Let's click on the button Select. A modal with the list of the files available under the particular record will be displayed below.



1. The table list has checkboxes, select all the files, or the desired files by checking the respective checkbox.
2. Buttons available
  - a. Search Icon - Search the table with Title and Owner Name.
  - b. Refresh Icon - to refresh the list of the files.
  - c. Delete Icon - Select multiple files and click here to delete the files.
  - d. Download - With the click of this button, it downloads the selected files individually or separately.
  - e. Download As Zip - On click of this button, the selected files are downloaded in Zip format.



## Filters Available



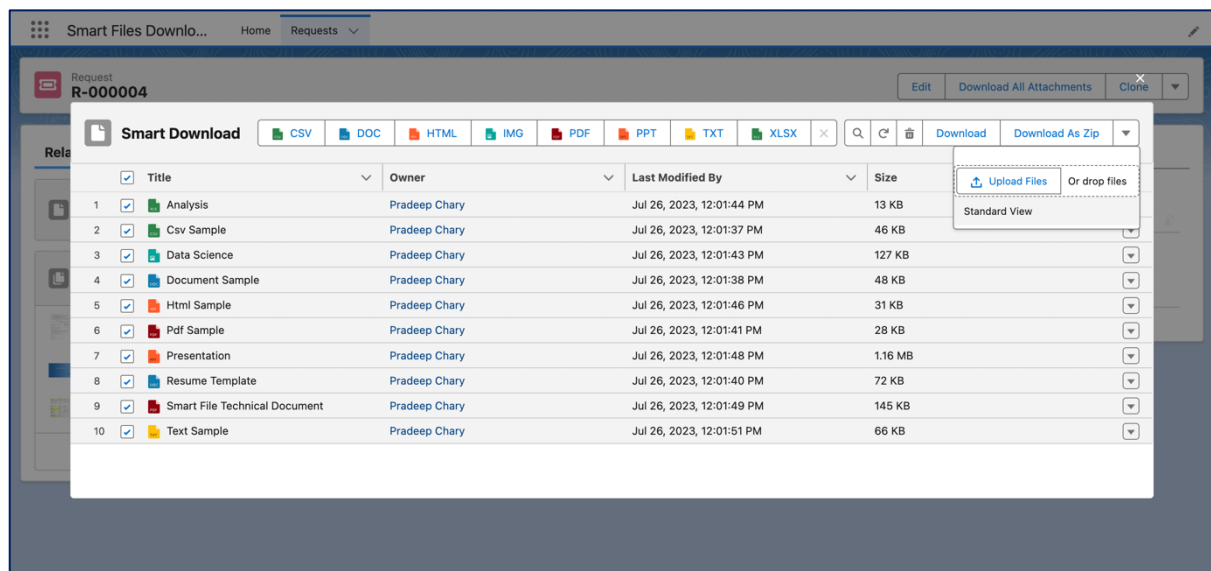
### 1. Content Type Filter: The 'PDF' filter applied

- Filters are available based on the file type. Click on one of the filters, let's say 'PDF', then the files of type image are displayed. Click on cancel (X) to remove the filters applied.

### 2. Search Bar:

- Filters the files by searching with Title or Owner Name.

To Upload more files from the modal only and to navigate to the standard view.



## Additional Feature (Optional)

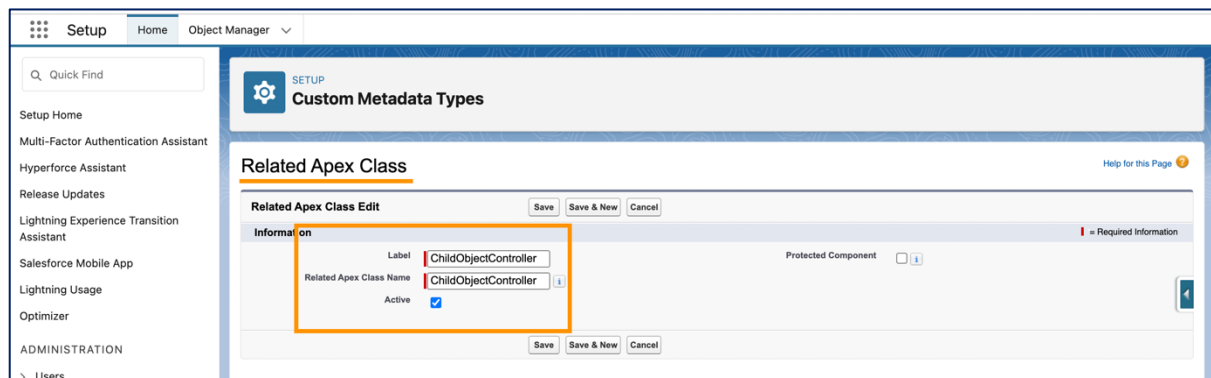
Consider a scenario where you need to show files under the child records on a parent record. Say for example we have a child object called Ticket\_\_c under a parent object Case. This is achievable by doing the below steps.

```

1  global class ChildObjectController implements slds.slds_ChildObject {
2
3      public static Set<Id> getChildIds(String stRecordId) {
4          Set<Id> set_Ids = new Set<Id>();
5          List<Ticket__c> list_Tickets = [SELECT Id FROM Ticket__c WHERE Case__c = :stRecordId];
6          set_Ids = new Set<Id>(new Map<Id, Ticket__c>(list_Tickets).keySet());
7
8          return set_Ids;
9      }
10
11 }

```

1. Create a new Apex Class, call it ChildObjectController (or any). Make sure the class is **global**.
2. Extend it with interface slds.slds\_ChildObject.
3. Create the method getChildIds and pass the parameter parentId to it. Add your logic in the method to return the set of the child record IDs.



4. Add the class to custom metadata type
  - a. Go to Setup -> Custom Metadata type -> Related Apex Class -> Manage
  - b. Click on the New button
  - c. Enter the class created name in the Label, Name, and mark it Active as per the screenshot. Click Save.
5. Now the Ticket record files will be displayed on the parent Case record.
6. Similarly you can do for as other parent-child objects.
7. By default this functionality is available for child objects Email Message and Messaging Session.



## Custom Labels

You can change and add translations to the User Interface text from the Custom labels.  
Below are the custom labels used in this application.

| Custom label | Value                                                 | Short Description          |
|--------------|-------------------------------------------------------|----------------------------|
| CLSFD0001    | No Files                                              | Error Message              |
| CLSFD0002    | Record does not exist                                 | Error Message              |
| CLSFD0003    | Please check record Id and object                     | Error Message              |
| CLSFD0004    | Please check the custom metadata<br>sdfs__Setting     | Error Message              |
| CLSFD0005    | Please select a record                                | Error Message              |
| CLSFD0006    | Select                                                | Component Button Label     |
| CLSFD0007    | Download                                              | Component Button Label     |
| CLSFD0008    | Download As Zip                                       | Component Button Label     |
| CLSFD0009    | Download All                                          | Component Button Label     |
| CLSFD0010    | Smart Files Download                                  | Component Header Title     |
| CLSFD0011    | Preview                                               | Datatable Row action label |
| CLSFD0012    | Title                                                 | Datatable Column Title 1   |
| CLSFD0013    | Owner                                                 | Datatable Column Title 2   |
| CLSFD0014    | Last Modified By                                      | Datatable Column Title 3   |
| CLSFD0015    | Size                                                  | Datatable Column Title 4   |
| CLSFD0016    | Delete                                                | Component Button Label     |
| CLSFD0017    | Delete the Selected Files?                            | Confirmation Message       |
| CLSFD0018    | Standard View                                         | Datatable Row action label |
| CLSFD0019    | Refresh                                               | Component Button Label     |
| CLSFD0020    | Search                                                | Search Button Label        |
| CLSFD0021    | Close                                                 | Close Button Label         |
| CLSFD0022    | Cancel                                                | Cancel Button Label        |
| CLSFD0023    | Unknown error, please contact System<br>Administrator | Error Message              |
| CLSFD0024    | File(s) deleted                                       | Files deleted Message      |
| CLSFD0025    | Download Successful                                   | Successful Message         |

END

This is all about the Smart Files Download application. Let us know your feedback in the comments and please rate the application.



Thank You