



Forsys Catalog

Admin Guide

Date : July 30, 2023

Table of Contents

About This Guide.....	3
Catalog Object Configuration.....	3
Create Quick Action Button for Products Data Refresh Functionality.....	3
Create Quick Action Button for Application Fields Mapping Functionality.....	4
Create Detail Page Button for Display Fields Functionality.....	4
Steps to add the buttons to the Catalog Object Page Layout.....	5
Create the Necessary Fields on Header Object and Line Item Object.....	6
Header Object Fields.....	6
Line Item Object Fields.....	8
Create the Catalog Object Record and Associated Files.....	10
Steps to create the Catalog Record.....	11
Steps to generate the Products Data File.....	12
Steps to generate the Application Fields Mapping File.....	12
Steps to generate the Display Fields File.....	15
Create Detail Page Button on Header Object for Catalog Component.....	18
Catalog Component in Experience Cloud Site.....	19
Steps to add Catalog Component in Experience Cloud Site.....	19
Steps to create Detail Page Button for Catalog Component.....	20
Custom Calculations and Custom Styles.....	23
Steps to add Custom Calculations File.....	23
Steps to add Custom Styles for Buttons.....	23
Steps to update the Custom Metadata for Custom Calculations and Styles.....	24
Override Custom Labels.....	25
Permission Set.....	27

About This Guide

This Guide provides the step-by-step process to set up the Forsys Catalog Solution.

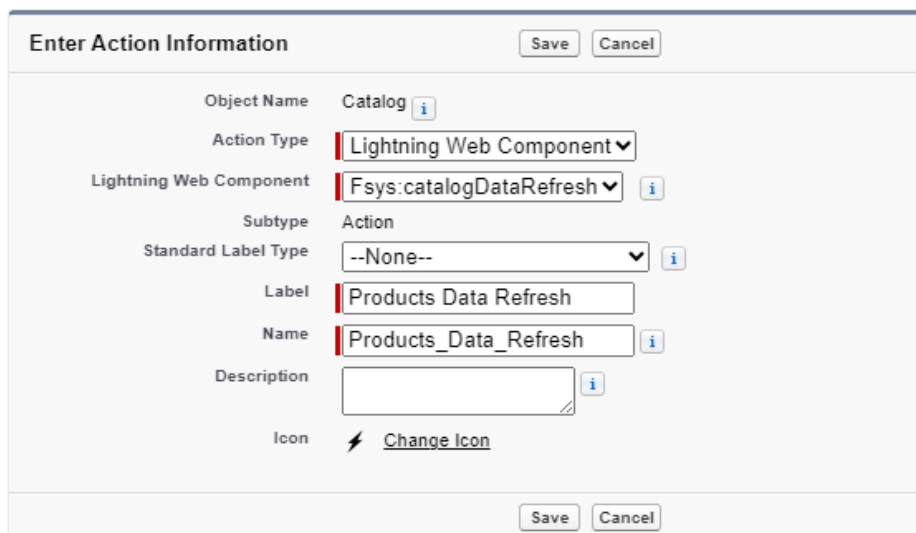
Catalog Object Configuration

Catalog Object needs to have a few Quick Action Buttons and Detail Page Button to generate the necessary files for the Catalog record.

Steps to create Quick Action Button for Products Data Refresh Functionality

- 1) Go to Setup and navigate to Object Manager and select Catalog object.
- 2) Go to Buttons, Links, and Actions on the object.
- 3) Create a Quick Action by clicking on the New Action button.
- 4) Provide the action type as Lightning Web Component and select the Fsys:catalogDataRefresh Component.

Catalog Actions
New Action



The screenshot shows a dialog box titled "Enter Action Information" with "Save" and "Cancel" buttons at the top right. The form contains the following fields:

- Object Name:** Catalog (with an information icon)
- Action Type:** Lightning Web Component (dropdown menu)
- Lightning Web Component:** Fsys:catalogDataRefresh (dropdown menu, with an information icon)
- Subtype:** Action
- Standard Label Type:** --None-- (dropdown menu, with an information icon)
- Label:** Products Data Refresh
- Name:** Products_Data_Refresh (with an information icon)
- Description:** (empty text box, with an information icon)
- Icon:** ⚡ Change Icon

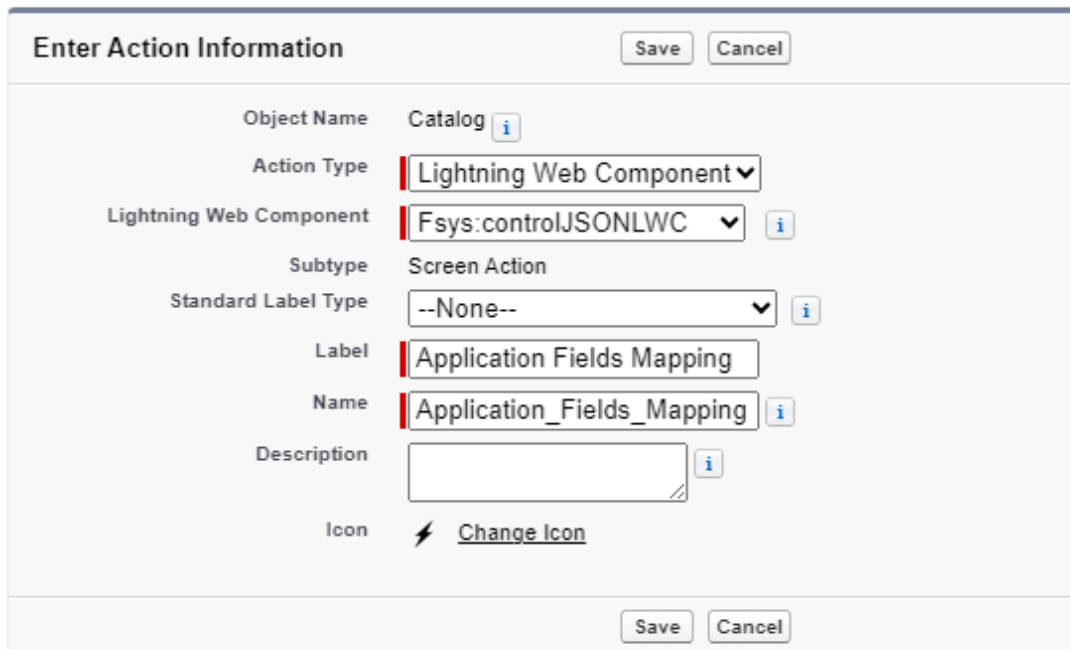
At the bottom right of the dialog box are "Save" and "Cancel" buttons.

Steps to create Quick Action Button for Application Fields Mapping Functionality

- 1) Go to Setup and from Object Manager go to the Catalog object.
- 2) Go to Buttons, Links, and Actions on the object.
- 3) Create a Quick Action by clicking on the New Action button.
- 4) Provide the action type as Lightning Web Component and select the Fsys:controlJSONLWC Component.

Catalog Actions

New Action



The screenshot shows the 'Enter Action Information' form for creating a new action. The form has a title bar with 'Enter Action Information' and 'Save' and 'Cancel' buttons. The form fields are as follows:

- Object Name:** Catalog (with an information icon)
- Action Type:** Lightning Web Component (dropdown menu)
- Lightning Web Component:** Fsys:controlJSONLWC (dropdown menu, with an information icon)
- Subtype:** Screen Action
- Standard Label Type:** --None-- (dropdown menu, with an information icon)
- Label:** Application Fields Mapping
- Name:** Application_Fields_Mapping (with an information icon)
- Description:** (text area, with an information icon)
- Icon:** ⚡ Change Icon

At the bottom of the form, there are 'Save' and 'Cancel' buttons.

Steps to create Detail Page Button for Display Fields Functionality

- 1) Go to Setup and from Object Manager go to the Catalog object.
- 2) Go to Buttons, Links, and Actions on the object.
- 3) Create a Detail Page Button by clicking on the New Button or Link button.
- 4) Set the attributes as mentioned below.

Detail Page Attribute	Value
Display Type	Detail Page Button
Behavior	Display in existing window without sidebar or header
Content Source	URL
URL Value	<code>/lightning/cmp/Fsys__DisplayFieldsGenerator?c__recordId={!CASESAFEID(Fsys__Catalog__c.Id)}</code>

Edit Catalog Custom Button or Link

Display Fields

Custom Button or Link Edit
Save Quick Save Preview Cancel

Label
Name ⓘ
Description
Display Type
☐ Detail Page Link [View example](#)
☒ Detail Page Button [View example](#)
☐ List Button [View example](#)
Behavior [View Behavior Options](#)
Content Source

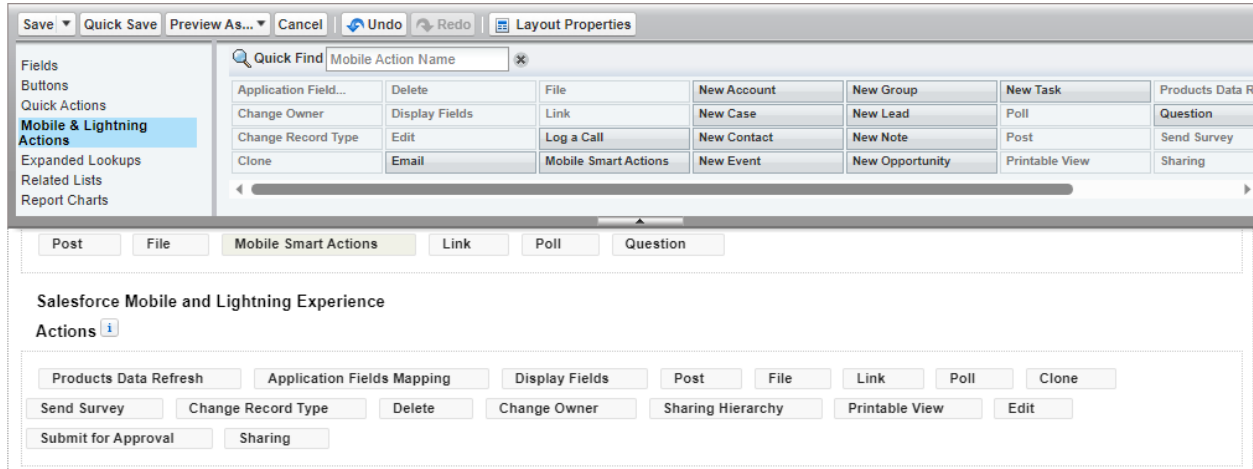
Select Field Type
Insert Field
Insert Operator

```
/lightning/cmp/Fsys__DisplayFieldsGenerator?c__recordId={!CASESAFEID(Fsys__Catalog__c.Id)}
```

Steps to add the buttons to the Catalog Object Page Layout

- 1) Go to Setup and from Object Manager go to the Catalog object.
- 2) Go to Page Layouts and click on Catalog Layout.

- 3) Drag and drop the Quick Actions and Detail Page Button created earlier from the Mobile and Lightning Actions section to the Salesforce Mobile and Lightning Experience Actions.
- 4) Click on Save.



Create the Necessary Fields on Header Object and Line Item Object

Fields which are required on Header Object (Parent Object like Opportunity etc) and Line Item Object (Child Object like Opportunitylineitem etc).

Header Object Fields:

Field Name	Field Description	Field Type
TotalListPrice	Sum of the Prices of all products before applying discounts	Currency
TotalNetPrice	Sum of the Prices of all products after applying discounts	Currency
AverageDiscount	The Average Discount	Percent

Percentage	Percentage applied to the Products	
SummaryOneTime	Sum of the Prices of all One Time Products	Currency
SummaryARR	Sum of the Annual Recurring Revenue of all Recurring Products	Currency
SummaryMRR	Sum of the Monthly Recurring Revenue of all Recurring Products	Currency
CatalogSetup	The Catalog Record associated to the Header Object Record	Lookup to Catalog Object
Term	Default Subscription Term added for the Recurring Products in Catalog Component	Number
Id	Header Object Record Id	Available by Default
Name	Name Field on Header Object	Available by Default
PriceChange	Field which indicates whether the Summary Values Calculation is Calculated or Pending. (Pending indicates an error has occurred while calculating the Summary Values)	Picklist with values (Calculated, Pending)
PricebookId	Pricebook field on the header object	Lookup to Pricebook object (Available by

		Default for Standard Objects)
--	--	-------------------------------

Note: Set the Term and CatalogSetup fields as required on Header Object i.e. the values shouldn't be null on Header Object.

Line Item Object Fields:

Field Name	Field Description	Field Type
MRR	Monthly Recurring Revenue of a Recurring Product.	Currency
ARR	Annual Recurring Revenue of a Recurring Product.	Currency
ExtendedPrice	Total Price of Product before applying discounts.	Currency
TotalPrice	Total Price of Product after applying discounts.	Currency (Available by Default for Standard Objects)
TCR	Total Contract Revenue of a Product i.e Total Price after applying discounts	Currency
UnitNetPrice	Per unit price of Product after applying discounts	Currency
DiscountType	Type of Discount Applied	Picklist with values (Percent, Amount)
DiscountValue	Discount applied to the	Number

	Product	
PricingType	Pricing Type of Product	Picklist with values (One Time, Recurring)
PricingFrequency	Frequency of the Recurring Products	Picklist with values (Monthly, Annually)
ListPrice	List Price of the Product	Currency (Available by Default for Standard Objects)
LineItemTerm	Subscription term of the Product	Number
Quantity	Quantity of the product	Number (Available by Default for Standard Objects)
SalesPrice	Sales Price of the product	Currency (Available by Default for Standard Objects)
HeaderId	Header Object Id	Lookup to Header Object (Available by Default for Standard Objects)
PricebookEntry	Pricebook Entry Id	Lookup to PricebookEntry Object (Available by Default for Standard Objects) and Text Field for Custom Object
ProductId	Product Id	Lookup to Product2 Object (Available by Default for Standard Objects)

PriceChange	Field which indicates whether the Net Price Calculation is Calculated or Pending. (Pending indicates an error has occurred while calculating the Net Price)	Picklist with values (Calculated, Pending)
Id	Line Item Id	Id (Available by Default for Object)
ProductName	Relationship Name or Formula Field which references Name field on Product2 Object through Product2 Lookup Field	Field Creation is not required when Relationship Name (like Product2.Name etc) is used.
AverageDiscountPercent	Discount amount indicated as Percent irrespective of Discount Type applied to Product	Percent
ProductCode	Formula field which references product code field of the Product2 object	Formula (Available by default on Opportunity Object)

Note: The Label of the Fields which are created should be unique and should not have any conflicts with the labels of the Standard Fields present in the Object.

Create the Catalog Object Record and Associated Files

Catalog Object record needs to be created.

Products Data File contains all the Products and their Pricing Information.

Application Fields Mapping File (Control File) contains all the data in the form of key-value pairs associated with the Header and Line Item Object.

Display Fields File contains all the data of the fields which are used to generate the columns of the cart in the Catalog Component. Each field added to the component represents a column in the Cart.

Steps to create the Catalog Record

- 1) Add the Catalog Object Tab in the Navigation bar.
- 2) Go to Catalog Object Tab.
- 3) Create a New Record with the values mentioned below.

Field Name	Field Value
Name	Catalog Record Name
Parent Object	Header Object API Name
Line Item Object	Line ItemObject API Name
Price Book	Pricebook Record

Note: Do not assign any value to the Products File Generation Status Field.

New Catalog

* = Required Information

Information

* Catalog Name ↶

* Parent Object ↶

* Line Item Object ↶

Owner
 Saas Catalog Testing

Price Book ↶

✕

Products File Generation Status

--None--
▼

Cancel

Save & New

Save

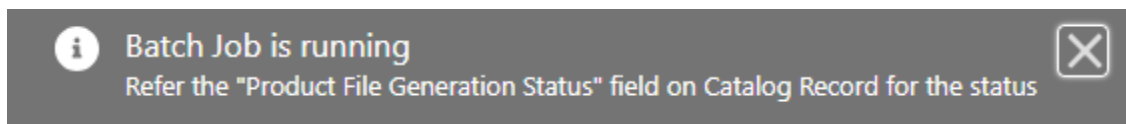
Steps to Generate the Products Data File

- 1) In the Catalog Record, click on the Quick Action Button which is created for Products Data Refresh Functionality earlier, then a Toast message appears.
- 2) Refresh the Page.
- 3) Products File Generation Status Field indicates:

Products File Generation Status	Description
Pending	Products File Generation is in Progress
Completed	Products File Generation is Completed
Failed	Products File Generation is Failed

Note:

1. If the Products File Generation Status is Failed, verify if the Current User is assigned to the “Saas_Catalog” Permission Set and check whether the Associated Pricebook has any products.
2. All the Active Pricebook Entries present in the associated pricebook irrespective of the value in the active field in Product record will be included in the generated file.



Steps to Generate the Application Fields Mapping File

- 1) In the Catalog Record, click on the Quick Action Button which is created for Application Fields Mapping Functionality earlier.
- 2) Select Fields created earlier from a picklist related to each field.
- 3) If you want to add any new field click on add field button.

- 4) Enter the Field label and select the Field value from the picklist.
- 5) If you want to add any relationship field check the Is Relationship Field checkbox and enter the field value as {RelationshipName.FieldAPIName}.

Note:

- 1) Append “__r” to the Relationship Name for the Custom Relationships.
- 2) The fields mentioned in Application Fields Mapping Component will be used in Catalog Page.
- 3) The Current User who needs to view the Catalog Component should have read access on all the fields specified in Application Fields Mapping Component, else an error message will be displayed.
- 4) If mapping is changed in the Application Fields Mapping component, then the Display Fields Component needs to be re-saved.
- 5) The Static Field Names present in the Component are used for the default calculations provided in the package, so you can’t delete them. If you have customized the calculations and some Static Field Names are not necessary for your calculations, then provide any temporary field.
- 6) The Static Field Name of any added field should not contain any spaces.
- 7) The relationship field’s Api Name is case sensitive.

Parent Fields Mapping
[Add Field](#)

	Static Field Names	Is Relationship Field	Parent Fields	
1	TotalListPrice		Total List Price	
2	TotalNetPrice		Total Net Price	
3	AverageDiscountPercentage		Avg Discount Percent	
4	SummaryOneTime		Summary One Time	
5	SummaryARR		Summary ARR	
6	SummaryMRR		Summary MRR	
7	CatalogSetup		Catalog	
8	Term		Default Subscription Term	
9	Id		Opportunity ID	
10	Name		Name	
11	PriceChange		Price Change	
12	Pricebookid		Price Book ID	

Next
Revert Values
Cancel

Application Fields Mapping

Child Fields Mapping

[Add Field](#)

	Static Child Field Names	Is Relationship Field	Child Fields	
1	MRR		MRR	
2	ARR		ARR	
3	TCR		TCR	
4	ExtendedPrice		Extended Price	
5	TotalPrice		Total Price	
6	UnitNetPrice		Unit Net Price	
7	DiscountValue		Discount Value	
8	DiscountType		Discount Type	
9	PricingFrequency		Pricing Frequency	
10	PricingType		Pricing Type	
11	ListPrice		List Price	
12	LineItemTerm		Subscription Term	
13	Quantity		Quantity	
14	SalesPrice		Sales Price	
15	HeaderId		Opportunity ID	
16	PricebookEntry		Price Book Entry ID	
17	ProductId		Product ID	
18	PriceChange		Price Change	
19	Id		Line Item ID	
20	ProductName	✓	Product2.Name	
21	AverageDiscountPercent		AverageDiscountPercent	
22	ProductCode		Product Code	

[Previous](#)
[Save](#)
[Revert Values](#)
[Cancel](#)

Steps to Generate the Display Fields File

- 1) In the Catalog Record, click on the Quick Action Button which is created for Display Fields File Functionality earlier.
- 2) Add the Fields by clicking the Add Field button.
- 3) Select the Object i.e Header or Line Item Object.
- 4) Select Field from Picklist (Picklist has all the keys present in Application Fields Mapping Component) depending on the object selected.
- 5) Enter the Field label and select the Field Type for the field (Supported field types are Text, Number, Currency, Lookup, URL, Picklist, Percent, Date, Datetime and Boolean).
- 6) Add the Field Type Attributes for the field if required.

Field	Type Attribute	Description
Currency	MinimumFractionDigits	Indicates the minimum number of Decimal Digits needed to represent a currency field.
	MaximumFractionDigits	Indicates the maximum number of Decimal Digits needed to represent a currency field.
Number	MinimumFractionDigits	Indicates the Minimum Number of Decimal Digits needed to represent a number.
	MaximumFractionDigits	Indicates the Maximum Number of Decimal Digits needed to represent a number.

URL	URL Value	Select the field from the picklist which has the URL Value.
	IsSalesforceId	Checkbox indicates whether the URL value is a Salesforce Id.

7) Select the Checkboxes.

Check Box	Description
Is Editable	Indicates whether the column is Editable or not
Is Required	Indicates whether the value in the column should not be null or blank while editing the column, if the column is editable.
Is Mobile Displayable	Indicates whether the column should be displayed in Mobile View (i.e Mobile Devices).

- 8) Enter the Column Width (in px), if required.
- 9) Select the Alignment of data in column, if required.
- 10) After clicking 'Save', you can rearrange the fields by dragging and dropping one onto another as needed. The columns in the cart will appear in the newly arranged order.

Note:

- 1) The URL Fields, Header Object Fields and Relationship Fields are not editable, so the Is Editable checkbox will be disabled.

- 2) The requiredness of the lookup field will be managed by the field properties present on the object. So the `isRequired` button will be disabled for the lookup fields.
- 3) The `Is Required` Checkbox will be only enabled if the column is editable.
- 4) If you want to remove any key of any object (Header or Line Item) from the Application Field Mapping Component, then first all the references to that field in the Display Fields Component should be replaced or removed.
- 5) If you want to delete any field from the Org on any object (Header or Line Item), then first all the references in Display Fields Component and Application Fields Mapping Component should be replaced or removed in respective order.
- 6) By default, the Subscription Term is editable only for Recurring Products.
- 7) By default, the Quantity cannot be blank or null in Cart
- 8) By default, the Subscription Term For Recurring Products cannot be blank or null in Cart.
- 9) For lookup fields, the field will be visible in cart if the current user has read access to the Lookup object.


Display Fields
Edit Fields
Delete Fields

	Field Object	Field Name	Display Label	Field Type	Type Attrib...	Is Editable	Is Required	Is Mobile Dt...	Width	Alignment	
☐	Line Item	Headend	Opportunity	Lookup		☒	☐	☐	Width	Select Align...	
☐	Line Item	Contact	Contact	Lookup		☒	☐	☒	Width	Select Align...	
☐	Line Item	ProductName	Product Name	URL		☐	☐	☒	Width	Select Align...	
☐	Line Item	ProductCode	Product Code	Text		☐	☐	☒	Width	Select Align...	
☐	Line Item	ListPrice	List Price	Currency		☐	☐	☒	Width	Select Align...	
☐	Line Item	LineItemTerm	Subscription Term	Number		☒	☒	☒	Width	Select Align...	

Add field
Save
Discard changes
Back

Configure Field Type Attributes

minimumFractionDigits
2

maximumFractionDigits
2

Close

Configure Field Type Attributes

URL Value
ProductId

☒ Is Salesforce Id

Close

Create Detail Page Button on Header Object for Catalog Component

- 1) Go to the Setup, and navigate to the Object Manager tab.
- 2) Select the Header Object (ex. Opportunity, Order, Quote).
- 3) Click on Buttons, Links, and Actions tab.
- 4) Click on New Button or Link.
- 5) Fill the data as below.

Field	Value
Label	Label of the Button
Name	API name of the button
Display Type	Detail Page Button
Behavior	Display in existing window without sidebar or header
Content Source	URL
URL value for Single Currency Org	/lightning/cmp/FSYS__SaasCatalogAura?c__recordId={!CASESAFEID(HeaderObjectIdField)}&c__catalogId={!CASESAFEID(HeaderObjectCatalogLookupId)}
URL value for Multi Currency Enabled Org	/lightning/cmp/FSYS__SaasCatalogAura?c__recordId={!CASESAFEID(HeaderObjectIdField)}&c__catalogId={!CASESAFEID(HeaderObjectCatalogLookupId)}&c__currencyIsoCode={HeaderO

bjectCurrencyIsoCodeField}

- 6) Click on Save.
- 7) Click on Page Layouts and select the page layout.
- 8) Drag and drop the Detail Page Button created earlier from the Mobile and Lightning Actions section to the Salesforce Mobile and Lightning Experience Actions.
- 9) Click on Save.

Edit Opportunity Custom Button or Link
Catalog Page

Custom Button or Link Edit
Save Quick Save Preview Cancel

Label
Catalog Page

Name
Catalog_Page

Description

Display Type
☐ Detail Page Link [View example](#)
☒ Detail Page Button [View example](#)
☐ List Button [View example](#)

Behavior
Display in existing window without sidebar or header [View Behavior Options](#)

Content Source
URL

Select Field Type
Opportunity

Insert Field
-- Insert Merge Field --

Insert Operator

/lightning/cmp/Fsys__SaasCatalogAura?c__recordId={!CASESAFEID(Opportunity.Id)}&c__catalogId={!CASESAFEID(Opportunity.CatalogId__c)}

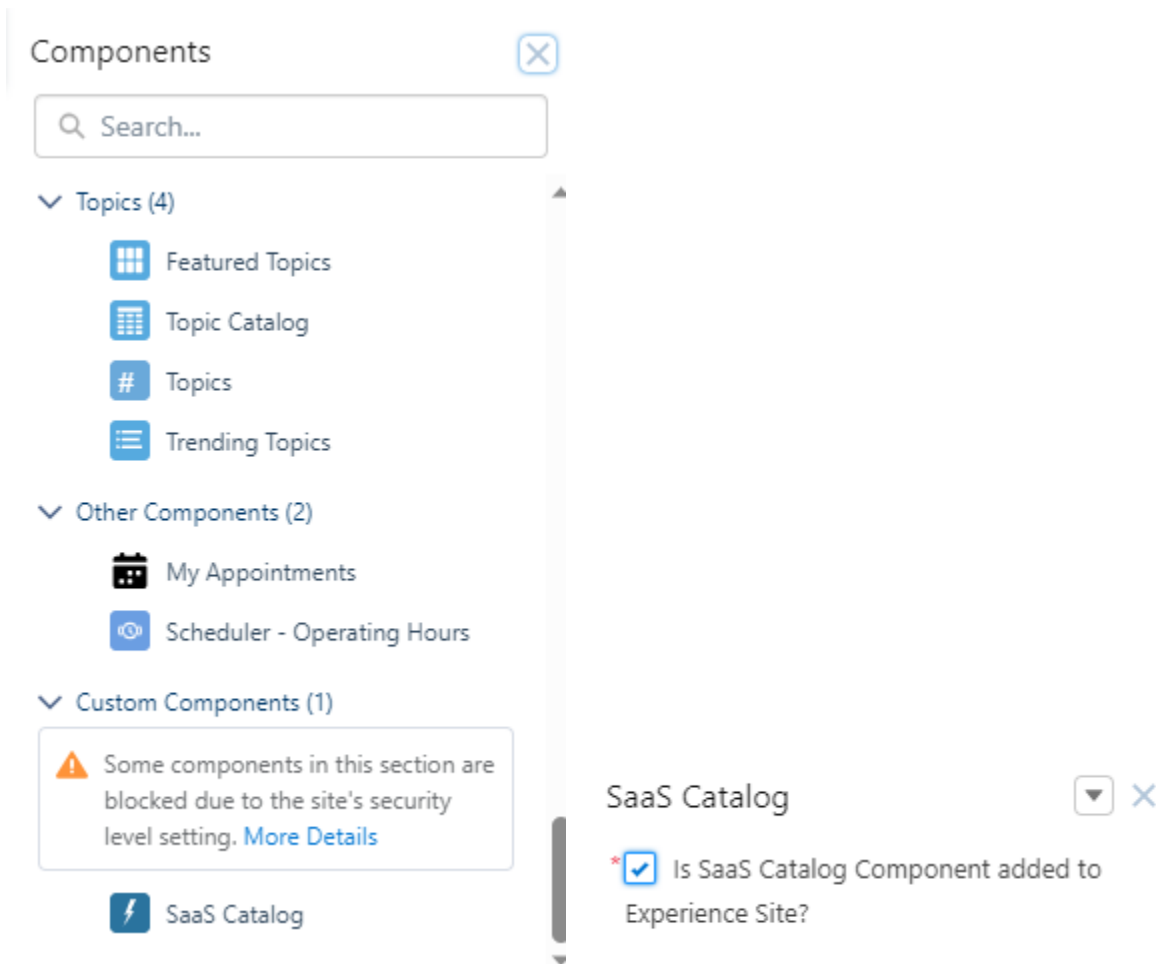
Catalog Component in Experience Cloud Site

Catalog Component can be added in Aura Framework Experience Cloud Sites and invoked from Header Object Record only.

Steps to add Catalog Component in Experience Cloud Site

- 1) Go to the Setup, and enter Digital Experiences in Quick Find.
- 2) Click on All Sites.

- 3) Click on Builder of any Experience Cloud Site(Community) in which the Catalog Component is needed.
- 4) Create a new Standard Page with Flexible Layout or 1 Full-Width Column Layout.
- 5) Drag and Drop the “SaaS Catalog” Component from the Components section on any component holder and check the “Is SaaS Catalog Component added to Experience Site?” checkbox.



Steps to create Detail Page Button for Catalog Component Page on Header Object

- 1) Go to the Setup, and navigate to the Object Manager tab.
- 2) Select the Header Object (ex. Opportunity, Order, Quote).

- 3) Click on Buttons, Links, and Actions tab.
- 4) Click on New Button or Link.
- 5) Fill the data as below.

Field	Value
Label	Label of the Button
Name	API name of the button
Display Type	Detail Page Button
Behavior	Display in existing window without sidebar or header
Content Source	URL
URL value for Single Currency Org	<code>/ {CommunityURL} /s/ {PageURL} ?c__recordId= {!CASESAFEID(HeaderObjectIdField)} &c__catalogId= {!CASESAFEID(HeaderObjectCatalogLookupId)}</code>
URL value for Multi Currency Enabled Org	<code>/ {CommunityURL} /s/ {PageURL} ?c__recordId= {!CASESAFEID(HeaderObjectIdField)} &c__catalogId= {!CASESAFEID(HeaderObjectCatalogLookupId)} &c__currencyIsoCode= {HeaderObjectCurrencyIsoCodeField}</code>

Note:

- 1) CommunityURL refers to the URL of Experience Cloud Site(Community) and PageURL refers to the URL of the page on which the Catalog Component has been added.
- 2) CommunityURL can be found beside the Experience Cloud (Community) in AllSites section of Digital Experiences. It will be in form of <https://{ORGURL}/{CommunityURL}>

- 3) If no CommunityURL is set up for the site, then remove the `/{{CommunityURL}}` in the URL Value of the Detail Page Button.
- 4) PageURL can be found in page properties of a page in the Builder Section of Experience Cloud Site(Community). It will be in form of `/{{PageURL}}`
- 6) Click on Save.
- 7) Click on Page Layouts and select the page layout which has been assigned to Community or Experience Cloud User.
- 8) Drag and drop the Detail Page Button created earlier from the Mobile and Lightning Actions section to the Salesforce Mobile and Lightning Experience Actions.
- 9) Click on Save.

Note: The Experience Cloud Site User needs to have access to all the fields referred in the Application Fields Mapping Component.

Catalog Page For Community

Custom Button or Link Edit
Save Quick Save Preview Cancel

Label
Catalog Page Community

Name
Catalog_Page_Community

Description

Display Type
☐ Detail Page Link [View example](#)
☒ Detail Page Button [View example](#)
☐ List Button [View example](#)

Behavior
Display in new window
View Behavior Options

Content Source
URL

Select Field Type
Order

Insert Field
-- Insert Merge Field --

Insert Operator

/s/catalog?c__recordId={!CASESAFEID(Order.Id)}&c__catalogId={!CASESAFEID(Order.CatalogId__c)}

Custom Calculations and Custom Styles

The user can customize the calculations and CSS in the Catalog.

Steps to add Custom Calculations File

- 1) Go to Setup, search for Static Resources in Quick Find.
- 2) Click on the NetPriceCalculation file.
- 3) Click on the view file.
- 4) Download the Javascript code that is opened in a new window.
- 5) Modify the Javascript code as per your requirement and upload it to the Static Resources.

Note:

- 1) Whenever Products are edited in Cart, all Products or Line Item records are sent to the Static Resource Calculation Function.
- 2) Make sure to convert the necessary data to numeric datatype in Custom Calculation Logic.
- 3) If any Product is added to the Cart, then all the Products present in the Cart are sent to the Static Resource Calculation Function.
- 4) If any error occurs while calculating net price of any product, then the whole transaction will be reverted i.e the changes which triggered the net price calculation will be reverted.
- 5) If there is no Discount Type on the Product or Line Item record, then Percent would be considered as Discount Type while calculating Net Price.
- 6) If there is no Pricing Type on the Product or Line Item record, then the Product would be considered as One Time Product.

Steps to add Custom Styles for Buttons

- 1) Go to Setup, search for Static Resources and select Static Resources.
- 2) Click on the CustomStyles files.
- 3) Click on the view file.
- 4) Download the css file that has opened in a new window.

- 5) Modify the css file as per your custom requirements and upload it to Static Resources.

Component	Associated CSS Class Name
Add Button	.add-btn
Current page in Pagination Component	.selectedpagecss
Remaining pages in Pagination Component	.atag
Pagination Navigation buttons	.navigationbutton

Note: Add !important to your Custom CSS properties, only then the Custom CSS will be applied.

Steps to update the Custom Metadata Types for Custom Calculations and Styles

- 1) Go to Setup, enter Custom Metadata Types in Search and select it.
- 2) Click on Manage Records for SaasCatalogResourceMapping.
- 3) Click on Edit for saasCatalogResources record.
- 4) Enter the custom Javascript Static Resource name in the jsResourceName field.
- 5) Enter the custom CSS Javascript name in the cssResourceName field.
- 6) Click on Save.

Note: If you want to use the default Calculations and default CSS provided in the package, then restore the jsResourceName and cssResourceName field values.

Field	Value
jsResourceName	Fsys__NetPriceCalculation

cssResourceName	Fsys__CustomStyles
-----------------	--------------------

Note:

1. The method names, parameters and output format in the NetPriceCalculations Javascript file should not be changed. You can write your custom logic inside the methods that are declared in the NetPriceCalculations file.
2. The Current User needs to have access to this Custom Metadata Type to load the static resources when they invoke the Catalog Component.

Override Custom Labels

Forsys Catalog solution's messages can be overridden by editing these custom labels using Translator Workbench.

Custom Label	Usage
CartChangesWarningLabel	The value in this label is displayed when there are changes made to the cart and the user clicks on the back button.
DiscountValueErrorLabel	The value in this label is displayed when the discount value is greater than 100.
EmptyCartErrorLabel	The value in this label is displayed when there are no Products present in the cart.
ErrorLabel	The value in this label is displayed when any error occurs while loading Catalog Component.
ExperienceSiteUserErrorLabel	The value in this label is displayed when any error occurs while Saving Products in Cart or User Session of Experience Site has expired.

InvalidCartDataLabel	The value in this label is displayed when any required field is edited to have blank or empty value.
MassActionsErrorLabel	The value in this label is displayed when there are no Products or Line item records selected to apply mass actions.
NetPriceErrorLabel	The value in this label is displayed when any error occurs while calculating the Net Price of Products.
PricebookErrorLabel	The value in this label is displayed when the Pricebook Id on Header Object record differs from the Pricebook Id on Catalog Record.
ProductAddedLabel	The value in this label is displayed whenever a Product is added to the Cart.
QuoteTotalsErrorLabel	The value in this label is displayed when any error occurs while calculating the Summary Values(Quote Totals).
SaveRecordsErrorLabel	The value in this label is displayed when any error occurs while saving records on the Line Item Object.
SaveRecordsSuccessLabel	The value in this label is displayed when Products or Line Item records successfully save on the Line Item Object.
StaticResourceErrorLabel	The value in this label is displayed when any error has occurred while loading the static resource.
RestrictCheckoutErrorLabel	The value in this label is displayed when any error has occurred and the user clicks on save button.

DisplayFieldsSaveErrorLabel	The value in this label is displayed when any error occurs while saving the fields data to the backend in Display Fields Component.
DisplayFieldsNoFieldsLabel	The value in this label is displayed when users click on save, if there are no fields present in Display Fields Component.
DisplayFieldsNavigateBackLabel	The value in this label is displayed when there are changes made to the Display Fields Component and the user clicks on the back button.
DisplayFieldsSaveSuccessLabel	The value in this label is displayed when fields data is successfully saved to the backend in Display Fields Component.
DisplayFieldsMissingFieldDataLabel	The value in this label is displayed when any required information for the field is missing.

Permission Set

The Permission Set “Saas_Catalog” has all the required permissions to use the Forsys Catalog Solution.

Permission set needs to be assigned to the admin user to access the Forsys Catalog Solution.