

Centralized Documentation Platform

Product Requirement Document



Version 1.0
Espressif Systems
Copyright © 2024

About This Document

The Centralized Documentation Platform(CDP) is an internal and external platform to streamline the company document publishing and management process. It serves to help manage the entire document management workflow and enhance efficiency.

This document outlines the product requirements for the CDP, detailing the necessary functionalities, and specifications to meet the requirements from the document group.

Release Notes

Date	Version	Release Notes
2024.01	V1.0	Initial release.

Technical Support

For any technical issues or suggestions, please kindly send an email to ehr.techsupport@espressif.com. The support team will look into the problem and get back to you soon.

Table of Contents

- 1. System Overview4**
 - 1.1. Key system parameters4
 - 1.2. Roles and permissions4
 - 1.3. Document Publish and Update Flow5
- 2. Sprint Specification.....6**
 - 2.1. Sprint 1 Infrastructure Setup7
 - 2.2. Sprint 2 Auto & Manual Publishing Features8
 - 2.2.1. Entities.....8
 - 2.2.2. Features.....11



1. System Overview

This section provides a high-level view of the proposed CDP, which outlines the main modules of the system, features to be included, and what other interactions needed within/outside the system. It shows the brief scope and roadmap, followed with more detailed descriptions of each component in the subsequent sections.

1.1. Key system parameters

Table 1-1. Key system parameters

Item	Value
System Name	Centralized Documentation Platform (CDP)
User Scope	All Espressif employees with role assigned to the system For available roles, please see table 1-2
Network Access	Internal office network (VPN for remote access)
Key Stakeholders	Documentation group
System Scope	<ul style="list-style-type: none">• Auto Publishing Features• Manual Publishing Features• APIs for Other Internal Systems• Core Website (some parts are not in scope of 1st version)• Subscription Related Features (some parts are not in scope of 1st version)• User Experience Related Features (not in scope of 1st version)

1.2. Roles and permissions

Available roles and relevant permissions are listed below.

Table 1-2. Roles and permissions

Role Name	Role Permissions
Owner (System Admin)	Permissions to all available features and data <ul style="list-style-type: none">• User management: add/delete a new user, change permission
Maintainer	<ul style="list-style-type: none">• Add/Update/Delete document types• Access to different doc groups (sw, hw, certificates, PCNs)
Developer	<ul style="list-style-type: none">• Permissions to view all features and data• Permissions to create/modify/publish/delete documents
Reporter	<ul style="list-style-type: none">• Permissions to view all features and data
Visitors to External Documentation Site	<ul style="list-style-type: none">• Permissions to view the documents and track their subscriptions (Not in current scope)



1.3. Document Publish and Update Flow

The flow chart **Figure 1-1** illustrates the previous flow among the entire document publish and edit process, which is not coherent and user friendly. Users only can create documents manually, and need to delete the document files individually before uploading new attachment to the edit page.

The flow chart **Figure 1-2** illustrates the updated flow among the entire document publish and update process, which include the auto publish the documents from GitLab to official website and manual upload, edit on the CDP page.

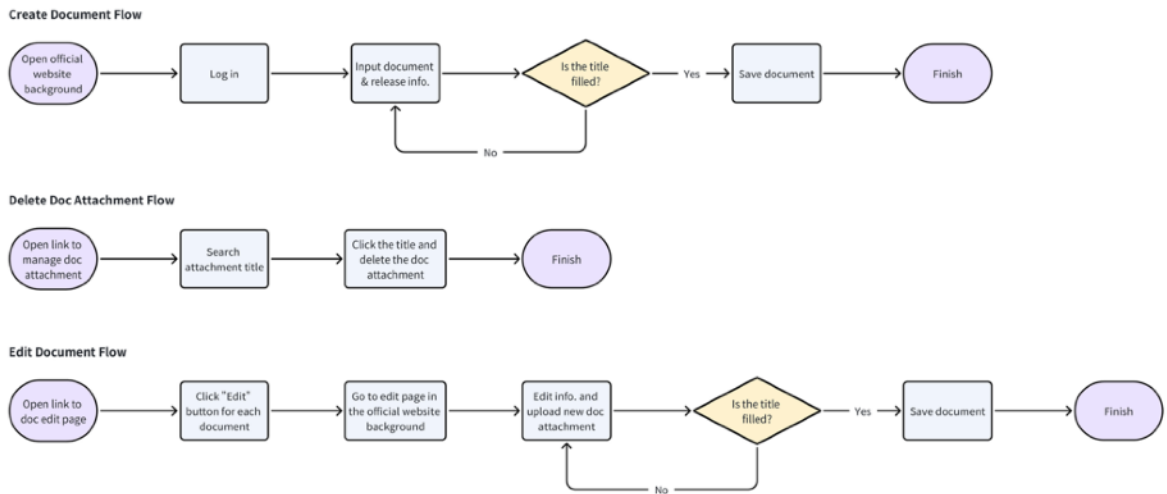


Figure 1-1. Previous flow chart for document publish & edit process

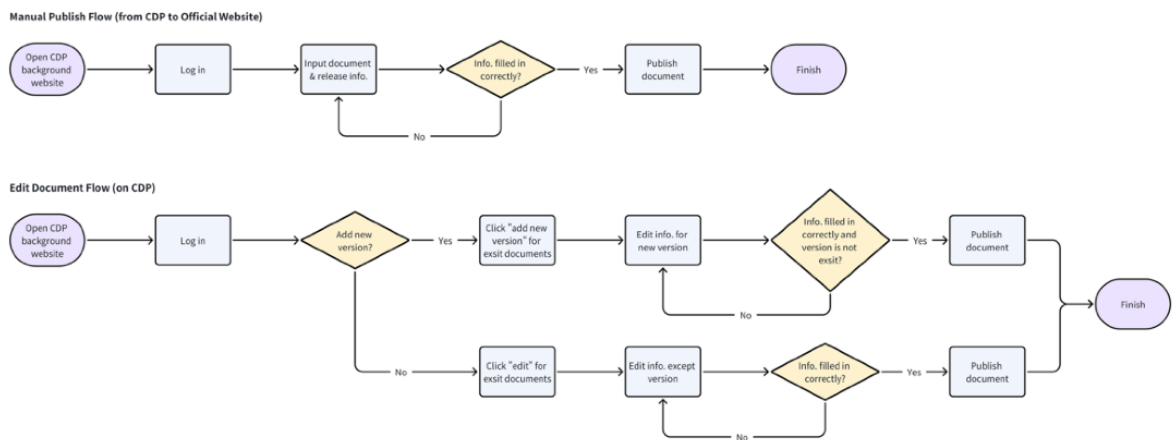


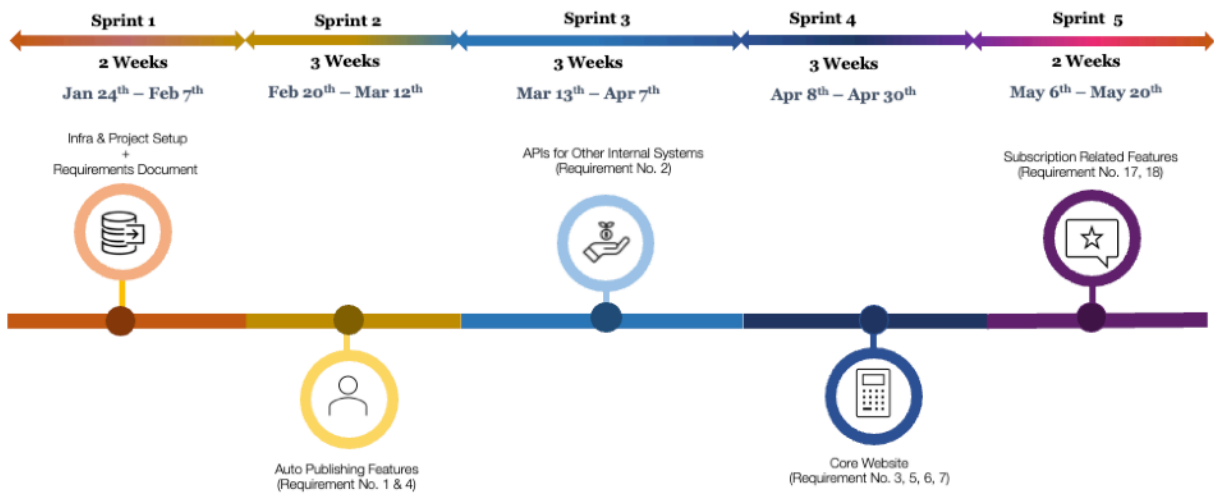
Figure 1-2. Updated flow chart for document publish & edit process



2. Sprint Specification

As the master plan shown below provides the planned timeline for the system P1 features. This section outlines the scope of every sprint, including feature list and estimated development time.

Centralized Documentation Platform Master Plan – Phase 1 (all P1 features)



2	API	Information on the Documentation page can be easily reused by other Espressif platforms, such as individual product pages and product selector.	P1
3	Webpage Layout	Documentation page should have a layout that mimics the new product page, which separates hardware and software documents	P1
4	Auto Publish	Publish documents maintained on GitLab or GitHub with metadata to www.espressif.com automatically	P1
5	Webpage Layout	Create a fitler for archived/NRND products	P1
6	Webpage Layout	Allows customers to sort documents from the most recently updated (This is the logic we have now).	P1
7	Webpage Layout	Allows customers to see the documents with the largest number of clicks. (Implementing this feature in the backend only is ok).	P1
17	Subscription	Add an option to allow subscribers to subscribe to our "new documents" that do not exist by the time they make subscription.	P1
18	Subscription	Update the logic when subscribers subscribe to "All" documents, then these "All" documents should also include future documents by default.	P1
9	User Experience	Provide personalized experience (example: Nordic)	P2
16	Subscription	Add automated testing if the subscription system actually works (for example, send a test notification to a certain email before sending emails to all subscribers) to avoid issues described in DOC-3074	P2
8	User Experience	Support enhanced search function	P3
10	User Experience	Integrate Chat with Espressif Bot, ref. EAI-1	P3
11	Subscription	Provide notification on update of an individual document (we support notification for group of documents and products, not individual documents).	P3
12	Subscription	Give option to receive notifications instantly (once per day), once per week (this is what we do now), and once per month.	P3
13	User Experience	On the documentation page, show all the documents user has subscribed to.	P3
14	Subscription	Add a short change description when sending notifications to customers.	P3
15	User Experience	List all updates with option to select specific period.	P3



2.1. Sprint 1 Infrastructure Setup

Sprint 1	Infrastructure Setup
Time	2024.01.24 - 2024.02.07 (2 weeks)
Targets	<ul style="list-style-type: none">• Project codebase and foundation setup• Requirement collection



2.2. Sprint 2 Auto & Manual Publishing Features

Sprint 2	Auto & Manual Publishing Features
Time	2024.02.20 - 2024.03.12 (3 weeks)
Targets	<ul style="list-style-type: none">Documents automatically publish by documentation team members from GitLab to official websiteDocuments manually publish by documentation team membersView auto published document records

2.2.1. Entities

• User

Field	Length	Data Type	Optionality
User Name 用户名称	150	String	Mandatory
Email 邮箱地址	250	String	Mandatory
Date Joined 创建时间	/	DateTime	Auto
Is Active 状态	/	Boolean	Mandatory

• Permission

Field	Length	Data Type	Optionality
ID	11	Integer	Auto
Name 权限名称	255	String	Mandatory
Description	255	String	Mandatory

• Role

Field	Length	Data Type	Optionality
ID	11	Integer	Auto
Name 权限组名称	150	String	Mandatory

• Role Permission

Field	Length	Data Type	Optionality
ID	11	Integer	Auto
Role 权限组	/	Foreign Key - Role	Mandatory
Permission 权限	/	Foreign Key - Permission	Mandatory



- **User Role**

Field	Length	Data Type	Optionality
ID	11	Integer	Auto
User 用户	/	Foreign Key - User	Mandatory
Role 权限组	/	Foreign Key - Role	Mandatory

- **Document Management**

- Field names in **purple** are columns default shown in the list search page

Metadata Mapping Between DMF and the existing Website: <https://espressifsystems.sharepoint.com/:x/r/sites/Documentation/Workspace/DOC-5005/mapping%20table.xlsx?d=w75c64435783249cbbd44966b382fde02&csf=1&web=1&e=9CL98F>

Field	DMF Name	Data Type	Optionality	Example or Note
	Document Information			
ID 文档编码	doc_id	String	Optional 1.Existing documents: 4-digit 2.New documents: Blank.	E.g.: 6659 * If the doc_id field empty in the DMF, then CDP will get a doc ID from espressif.com and show in the doc management page. * If have the doc_id field in the DMF, CDP will find this doc and update the information.
Title 文档名称	document_name	String length: 50	Mandatory	E.g.: ESP32-S3-PICO-1 Datasheet
Language 语言	language	Enum 1.English 2.Chinese 3.Russian	Mandatory	should be managable so that languages can be added or deleted (not in current sprint)
Document Storage 存储平台	document_storage	Enum 1.CDP 2.External	Mandatory	* PDF, .zip, and documents in other formats that need to be uploaded: CDP , which means the documents are stored directly on CDP. * HTML and documents stored in other places that do not need to be uploaded: External .
Document 文件	document_link	Upload	Optional	* If the platform is cdp, attachment is required. * If the platform is external, attachment is blank.
Link 链接	document_link	String	Optional	* If the platform is external, link is required. * If the platform is cdp, link is blank.



Field	DMF Name	Data Type	Optionality	Example or Note
Summary 文档描述	document_summary	String length: 500	Mandatory	E.g.: This document introduces the specifications of ESP-WROOM-02 hardware, including overview, pin definitions, functional description, peripherals, and electrical characteristics, etc.
Product Type 产品类别	product_type	Enum 1.Software 2.SoCs 3.Hardware/ Devkits 4.Hardware/ Modules 5.Equipment	Mandatory	* For chip datasheet, use: SoCs * For module datasheet, use: Hardware/Modules
Product Name 产品名称	product_name	Enum Any product names maintained in the product structure	Mandatory	E.g.: ESP32-MINI-1/1U If a product type is selected, new products under this type need to be added by default
Document Type 文档类别	document_type	Enum 1.Datasheet (Datasheet & Errata on the website) 2.Errata (Datasheet & Errata on the website) 3.Reference 4.Book 5.Guide 6.Application Note	Mandatory	should be manageable so that types can be added or deleted (not in current sprint)
Version 版本	version	String length: 50	Mandatory	* CDP documents: show the v by default, only need to provide number 0.5, 1.0, or any other version numbers. * External documents: latest, or any other release version, such as release-5.0.
Marketing Status 市场营销状况	marketing_status	Enum, auto filled 1.Mass Production 2.NRND 3.Obsolete	Optional	the status should be maintained in the product structure
Document Owner 文档负责人	document_owner	String	Mandatory	E.g.: wangfang@espressif.com Search in the user list, editable
Technical Owner 技术负责人	technical_owner	String	Optional	E.g.: wangfang@espressif.com Input box, editable
	Release Information			



Field	DMF Name	Data Type	Optionality	Example or Note
Release Date 发布日期	schedule_release_date	DateTime	Optional	E.g.: 2024.01.25 18:00:00 * Real-time: CDP provides the time when the document is actually published. * Schedule time: Any reasonable date in the future, accurate down to the hours within a month.
Send Add/Update Notification 是否发送新增/更新消息	notify_subscribers	Enum 1.Yes 2.No	Mandatory	
Visibility Status 显示状态	visibility_on_website	Enum 1.Hide 2.Show	Mandatory	
Authored by 编著人	authored_by	String	Mandatory	E.g.: wangfang@espressif.com Auto filled with the email address of the current user. Search in the user list, editable
Authored on 编著日期	authored_on	DateTime	Mandatory	Real Time, not editable E.g.: 2024.01.25 18:00:00 Auto filled with the real time, accurate down to the seconds .
Tracking Reference 跟踪参考	tracking_reference	String	Optional	E.g.: https://gitlab.espressif.cn/6688/espressif/esp-hardware-design-guidelines/-/merge_requests/32 This item is not included in LBCF, but should be generated automatically by a script according to the time when the MR is merged. Only the release method is "Manual" can be input and edit. Only show this part in the list page: esp-hardware-design-guidelines/-/merge_requests/32
Release Method 发布形式	/	Enum 1.Auto 2.Manual	Mandatory	Auto filled, not editable

2.2.2. Features

Document Management

- **Auto Publish Documents from GitLab to Official Website**

The flow chart **Figure 2-1** illustrates the auto publish flow from GitLab to official website. After document developers developing the documents on the GitLab, it will



initiate a CI/CD job for CDP system to synchronize the document. This process happens in real time. If this step is failed, CDP will send an error message to GitLab. If it is succeed, CDP will continue to auto-publishing the documents to the official website. If this step is failed, CDP will send an email and notification to the user. If it is succeed, the official website will show the published documents.

* If the doc ID is blank, it means it's a new document. CDP needs to get a doc ID from Espressif.com and show it in the doc management page. If GitLab provides a 4-digit ID, it means it's a new version for existing document. CDP needs to search through the exist documents and update the latest version. (If can not find document based on the provided ID, CDP will send an error message: The document ID does not exist.)

Auto Publish Flow (from GitLab to Official Website)

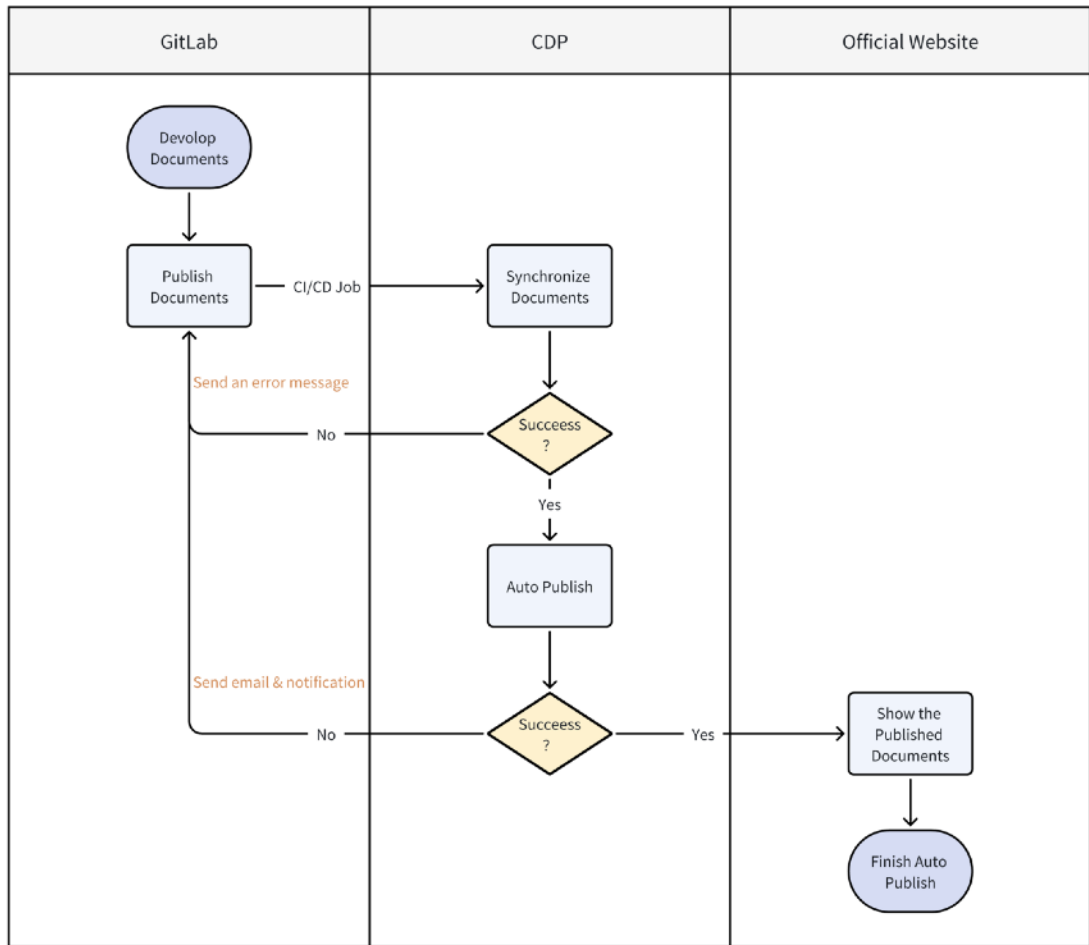


Figure 2-1. Auto publish flow from GitLab to official website

Email template:

Subject: Publishing failed for {document_name} because (of) {reason}

Hello {user},

Publishing of {document_name} failed because (of) {reason}. For details, please check the Synchronization page (ideally a link to that page).

Notification message template on CDP:



Publishing of {document_name} failed because (of) {reason}(ideally apply a hyperlink to the entire sentence and the link should link to the synchronization page, similar to the notification message on eHR).

Note: If {reason} is too long then provide short reason in the e-mail subject and long reason in the e-mail body.

通知里用 because 还是 because of 取决于你们提供的原因是句子还是名词短语

{user} 可以用 @espressif.com 前面的姓名拼音, 也可以用邮箱地址

Official Website Database Schema: https://www.icloud.com.cn/pages/0d4MGnKVlboxd_0X5JKRZQtRbA#Official_Website_Documentation_Database_Schema

• **Manual Create New Documents/Add New Version and Publish on Official Website**

The flow chart **Figure 2-2** illustrates the manual publish & edit flow from CDP to official website. Developers log in the CDP first, and click “Create New Document” to input the document and release information. System need to check if the document title, language and version is exist in the current document pool. If yes, system needs to send an error message: *The Document “{title}” exists*. If no, system will publish the new documents to the official website.

If the developers want to add the new version of the existing document, they need to click “Add New Version” icon in the list or detailed page. The system will automatically brings up the latest version of the document. Developers can update the content and publish it again. System needs to check if the version exist for this doc ID. If yes, system will send an error message: *The {version} of Document {ID} exists*.

If the developers want to edit the exist document, they can click “Edit” icon in the list or detailed page. The system will automatically brings up the latest version of the document. Developers can update the content and publish it again. *The version cannot be edited by users*.

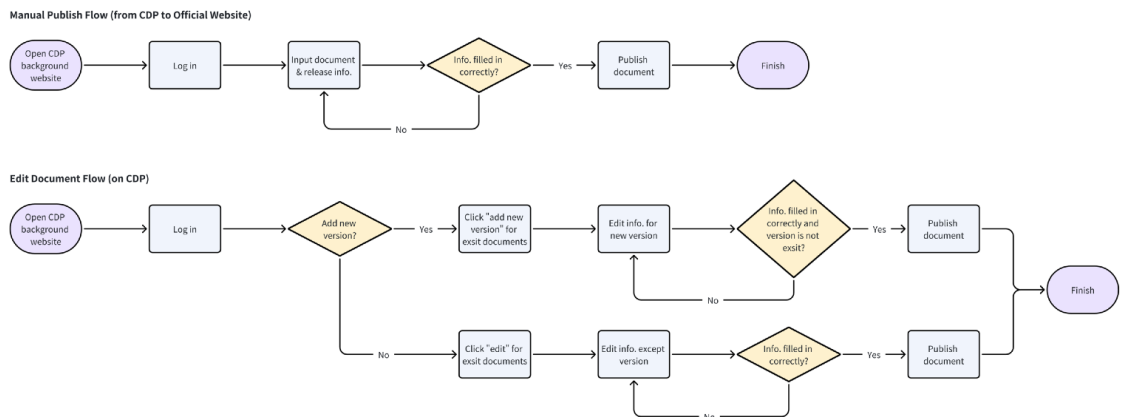


Figure 2-2. Updated flow chart for document publish & edit process



Action - Submit to create a new document	Implementation
Permission Check (Frontend & Backend)	Owner (system admin), Developer
Data Check	<ol style="list-style-type: none"> All mandatory fields are filled and in correct formats (Frontend & Backend) "Document Title" must be unique (Backend)
Data Auto Completion (Frontend & Backend)	<ol style="list-style-type: none"> ID Release Date (if choose "Real Time") Authored on (if choose "Real Time") Tracking Reference Release Form
Fields	Document Information <ol style="list-style-type: none"> Title Language Platform Attachment Link Summary Product Type Product Name Document Type Version Document Owner Technical Owner Release Information <ol style="list-style-type: none"> Release Date Send Add/Update Notification Visibility Status Authored by Authored on
Operate	<ol style="list-style-type: none"> Cancel Publish

Action - Submit to add new version for the exist document	Implementation
Permission Check (Frontend & Backend)	Owner (system admin), Developer
Data Check	<ol style="list-style-type: none"> All mandatory fields are filled and in correct formats (Frontend & Backend) Version for the doc ID must be unique (Backend)
Data Auto Completion (Frontend & Backend)	<ol style="list-style-type: none"> ID Release Date (if choose "Real Time") Authored on (if choose "Real Time") Tracking Reference Release Form



Action - Submit to add new version for the exist document	Implementation
Fields	Document Information 1. Title 2. Language 3. Platform 4. Attachment 5. Link 6. Summary 7. Product Type 8. Product Name 9. Document Type 10. Version 11. Document Owner 12. Technical Owner Release Information 13. Release Date 14. Send Add/Update Notification 15. Visibility Status 16. Authored by 17. Authored on
Operate	1. Cancel 2. Publish

Action - Submit to edit the exist project	Implementation
Permission Check (Frontend & Backend)	Owner (system admin), Developer
Data Check	1. All mandatory fields are filled and in correct formats (Frontend & Backend) 2. "Document Title" must be unique (Backend)
Data Auto Completion (Frontend & Backend)	1. ID 2. Release Date (if choose "Real Time") 3. Authored on (if choose "Real Time") 4. Tracking Reference 5. Release Form
Fields	Document Information 1. Title 2. Language 3. Platform 4. Attachment 5. Link 6. Summary 7. Product Type 8. Product Name 9. Document Type 10. Version (cannot be edited) 11. Document Owner 12. Technical Owner Release Information 13. Release Date 14. Send Add/Update Notification 15. Visibility Status 16. Authored by 17. Authored on
Operate	1. Cancel 2. Publish



• **Document Record List**

- Field names in **purple** are columns default shown in the list search page

The users can switch “product type” tab at the top of the table the to view documents under them. By default, data is presented in reverse order by publication time and in pagination.

Action - Submit to create a new project	Implementation
Permission Check (Frontend & Backend)	Owner (system admin), Developer, Reporter
Fields	<ol style="list-style-type: none"> 1. ID (clickable, to the document detail page) 2. Title 3. Language 4. Platform 5. Attachment 6. Link 7. Summary 8. Product Type 9. Product Name 10. Document Type 11. Version 12. Document Owner 13. Technical Owner 14. Release Date 15. Send Add/Update Notification 16. Visibility Status 17. Authored by 18. Authored on 19. Tracking Reference 20. Release Form
Operate	Owner (system admin), Developer: Header <ol style="list-style-type: none"> 1. Create New Document 2. Fields Setting Each Record <ol style="list-style-type: none"> 3. Edit 4. Add New Version 5. Delete

• **View Document Information**

Users can click the ID in the list and go to the detailed page for this document. Document information is displayed by version. By default, information about the latest version is displayed. Information about other versions is hidden.

Action - View the exist project	Implementation
Permission Check (Frontend & Backend)	Owner (system admin), Developer, Reporter



Action - View the exist project	Implementation
Fields	1. Doc ID Display the corresponding information by version: 2. Title 3. Version 4. Release Date 5. Language 6. Platform 7. Attachment/Link 8. Product Type 9. Product Name 10. Document Type 11. Document Owner 12. Technical Owner 13. Summary 14. Send Add/Update Notification 15. Visibility Status 16. Authored by 17. Authored on 18. Tracking Reference 19. Release Form
Operate	Owner (system admin), Developer: 1. Edit 2. Add New Version 3. Delete

Synchronization Record

The synchronization record page will show the record for auto publishing documents. By default, data is presented in reverse order by Sync Time (from GitLab to CDP) and in pagination. If the sync status is succeeded, page will also show the sync time. If the sync status is failed, page will show the failed reason when the mouse is hovering on the icon beside the “Failed” status.

Action - Submit to create a new project	Implementation
Permission Check (Frontend & Backend)	Owner (system admin), Developer, Reporter
Fields	1. ID 2. Title 3. Tracking Reference: show 30 characters, next... 4. Sync Status (from GitLab to CDP) 5. Sync Time (from GitLab to CDP) 6. Failed Reason (from GitLab to CDP): if the status is failed 7. Sync Status (from CDP to Website) 8. Sync Time (from CDP to Website) 9. Failed Reason (from GitLab to CDP): if the status is failed