IS-ENES3 Deliverable D10.4

CMIP6 documentation

Reporting period: 01/01/2022 - 31/03/2023

Authors: A. Ben Nasser (CNRS-IPSL)

Reviewers: Paola Nassisi (CMCC), Stephan Kindermann (DKRZ)

Release date: 27/01/2023

ABSTRACT

This deliverable reports on the ES-DOC CMIP6 documentation. We describe the operational ES-DOC ecosystem put in place to produce and maintain the CMIP documentation. This document is the final technical review of the ES-DOC software stack and concludes with some recommendations to follow in order to scale the ES-DOC service for CMIP7.

Dissemination Level		
PU	Public	

Revision Table			
Version	**Date**	**Name**	**Comments**
Initial draft	01/12/2022	Ben Nasser Atef	First draft of the document
Release for review	20/01/2023	Paola Nassisi Stephan Kindermann	All sections completed

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 824084

Table of contents

Executive Summary

The IS-ENES3 project is working on developing a climate data infrastructure to support the analysis and processing of large volumes of climate model data from CMIP and CORDEX simulations. The goals of the project include providing a sustainable and scalable data distribution system, supporting interoperability for automated data processing, and maintaining an international documentation infrastructure to support future model intercomparison projects. The project aims to provide the European contribution to the development of the ESGF (Earth System Grid Federation) architecture, which is a distributed data and computing system that provides access to a wide range of earth science data, including climate data.

Through this document, a comprehensive description of the ES-DOC services design provided so far will be presented, as well as some actions taken to ensure the provision of the services on the mid to long term, mainly the progress achieved in cementing the technical documentation, deployment recipes and scripts that would help future-proof the service stack, in case of infrastructure migration for example. This effort also leads to massively reducing the manpower required to maintain the services provided.

1. Introduction

The ES-DOC (Earth System Documentation) project requires a software ecosystem that facilitates both the provision and the consumption of documentation of the CMIP6 workflow and, where possible, automating the various and often complex stages involved.

The ES-DOC eco-system of tools and services are founded upon five pillars:

- a set of controlled vocabularies,
- a common ontology,
- a set of specialisations,
- a public archive,
- an errata registry.

The back-end software stack comprises: i) utility libraries to automate creation and publication of standardised documents from user-supplied content and to store and systemise semantics for applicable controlled vocabularies, models and ontologies; ii) content storage through archive repositories; iii) and a shell script library to facilitate development using these tools on the system on multiple separately version-controlled repositories.

The front-end stack consists of web services to manage documentation and errata databases, for generating and publishing the model documentation, for rewriting varios URLs; and web applications that support the viewing, searching and comparing of the published documentation, as well as serving and displaying other relevant content.

Each pillar is associated with a library and a set of applications and/or web-services. Collectively, they form the foundations of the ES-DOC project.

The ES-DOC suite of software and services is geared towards providing the best possible toolset for the scientific community to create and consume documentation for the CMIP6 simulations and related data as well as the attached metadata.

The ES-DOC community is composed of contributors from NCAR, CNRS-IPSL, CEDA-UKRI and DKRZ. A weekly telco is set up to keep track of work done and reinforce collaboration between the team members. This also leads to face-to-face meetings such as the meeting that took place at IPSL in Paris in November 2022.

This document provides a rundown of the technical infrastructure that made these services possible with the new developments within the IS-ENES3 WP10/JRA3.

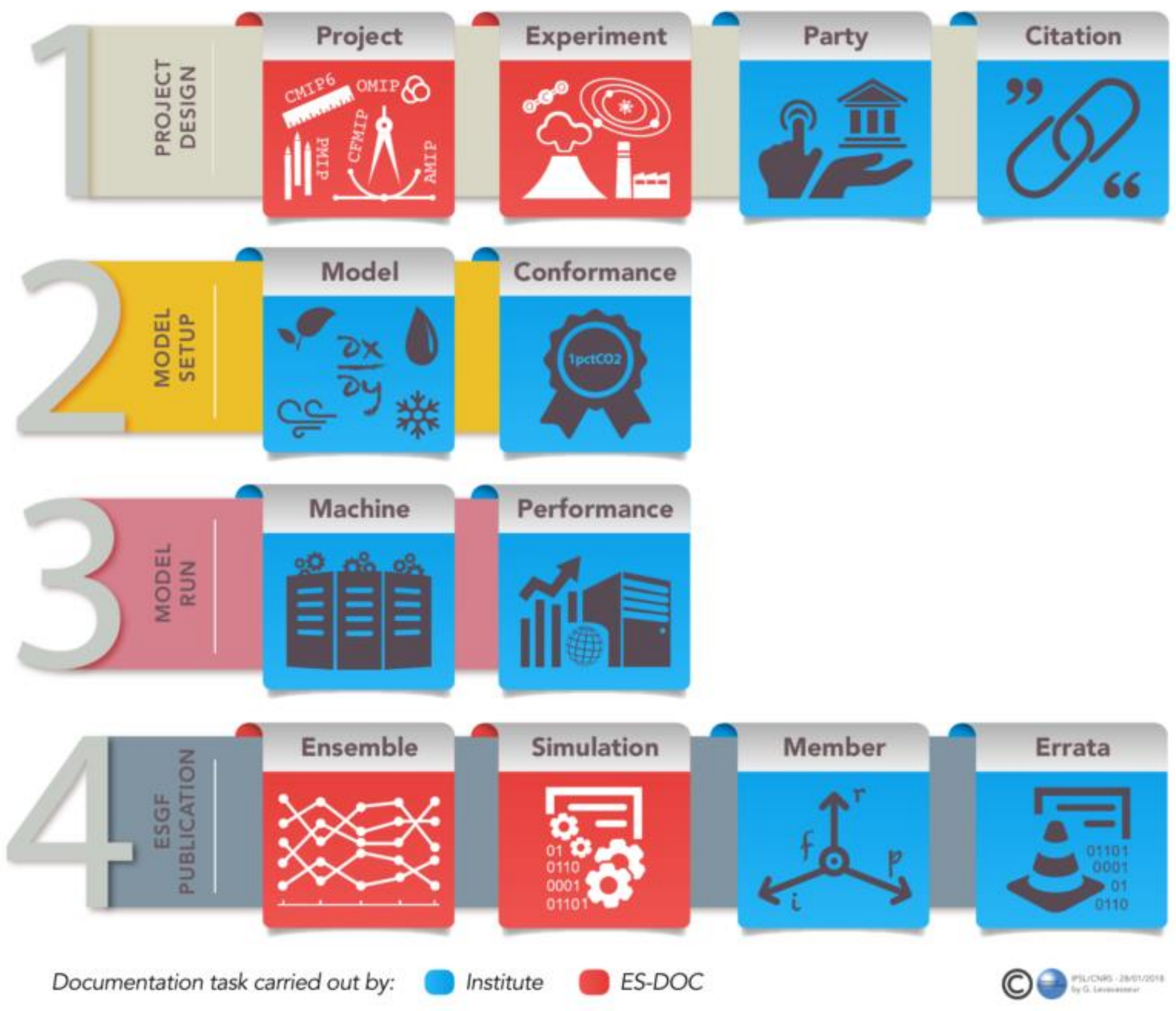
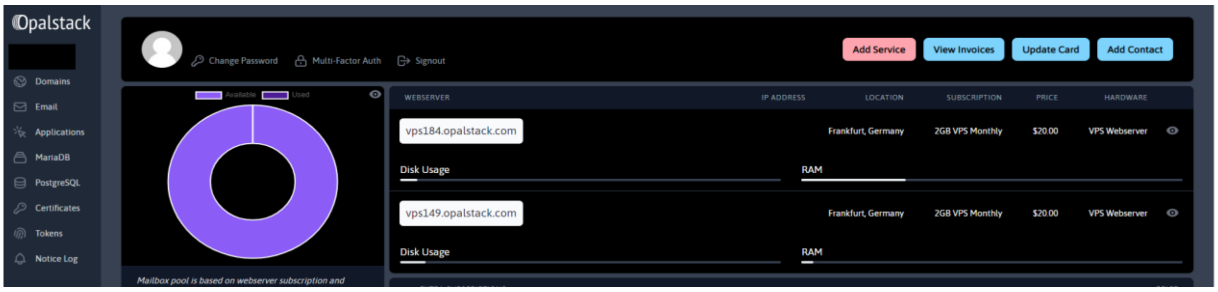


Figure 1: ES-DOC documentation coverage summarised in this document. Building on CIM concepts, the following documents (in bold) are created for the CMIP6 **Project**. A **Simulation** is a run of a configured **Model** which has **Conformance** to the numerical requirements of an **Experiment**, runs on a **Machine** with an HPC **Performance** and produces output Datasets (archived on ESGF). An **Ensemble** is a set of **Simulations** whose Axis **Members** describe how the simulations differ. A **Party** describes a person or organisation involved in the modelling process and a published reference is recorded in a **Citation**. Datasets may be associated with **Errata** if any problems are discovered after publication to ESGF.

2. Hardware infrastructure and dev-ops

The ES-DOC service suite relies on “OpalStack”, a 3rd-party service provider, to secure the necessary hardware and infrastructure for hosting. The service is charged yearly and CNRS-IPSL takes responsibility for due payments. “OpalStack” provides two virtual machines for ES-DOC (named vps184 and vps149), physically hosted in Frankfurt (Germany). These machines are used for production and test deployments for ES-DOC services. The data backup strategy and snapshots are managed by “OpalStack” which alleviate some administration pressure from ES-DOC personnel, leaving only the development, deployment and management of the services to the human resources.



WEBSERVER	IP ADDRESS	LOCATION	SUBSCRIPTION	PRICE	HARDWARE
vps184.opalstack.com		Frankfurt, Germany	2GB VPS Monthly	$20.00	VPS Webserver
vps149.opalstack.com		Frankfurt, Germany	2GB VPS Monthly	$20.00	VPS Webserver

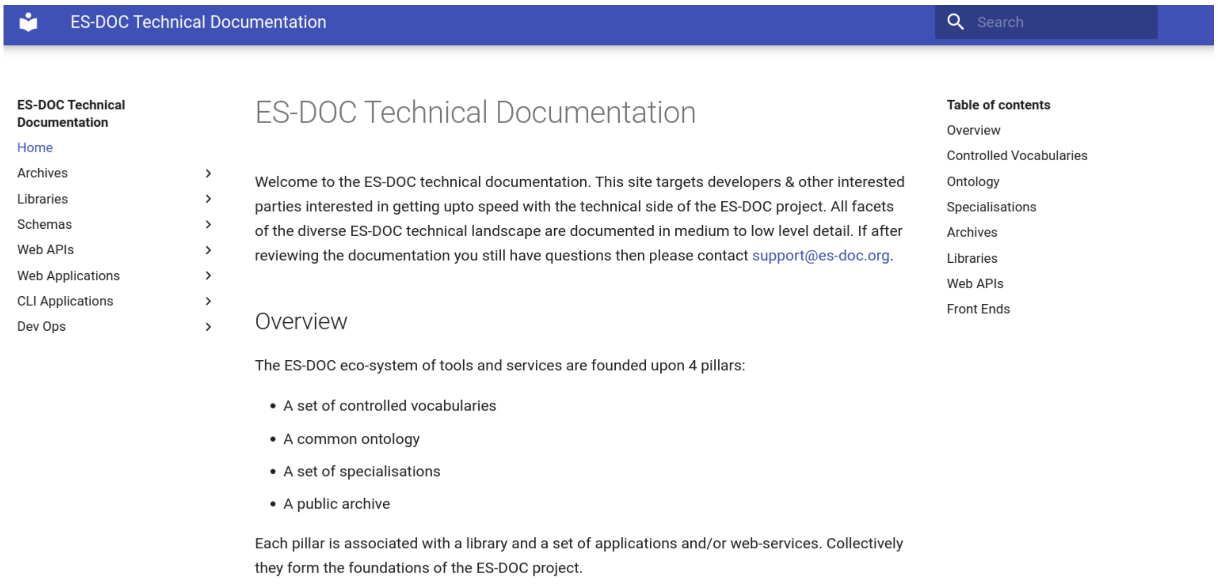
Mailbox pool is based on webserver subscription and

Figure 2: OpalStack system administration dashboard

2.1. Technical documentation and deployment recipes

One of the highest priority goals for ES-DOC in WP10 was to establish a complete technical documentation that would ensure the maintenance of the service and its resilience to key personnel unavailability. This resulted in the establishing of deployment recipes independent from hardware provision, as well as a full description of the stack. This documentation is currently available to consult at https://technical.es-doc.org/, the key and sensitive information is however only distributed internally via a specific dev-ops github repository at https://github.com/ES-DOC/devops

The ES-DOC face-to-face meeting that took place in November 2022 in Paris saw the completion of these deployment recipes and technical documentation, and they were put to the test on fresh virtual machines provided by IPSL system administrators.



ES-DOC Technical Documentation

Welcome to the ES-DOC technical documentation. This site targets developers & other interested parties interested in getting upto speed with the technical side of the ES-DOC project. All facets of the diverse ES-DOC technical landscape are documented in medium to low level detail. If after reviewing the documentation you still have questions then please contact support@es-doc.org.

Overview

The ES-DOC eco-system of tools and services are founded upon 4 pillars:

- A set of controlled vocabularies
- A common ontology
- A set of specialisations
- A public archive

Each pillar is associated with a library and a set of applications and/or web-services. Collectively they form the foundations of the ES-DOC project.

Figure 3: ES-DOC technical documentation

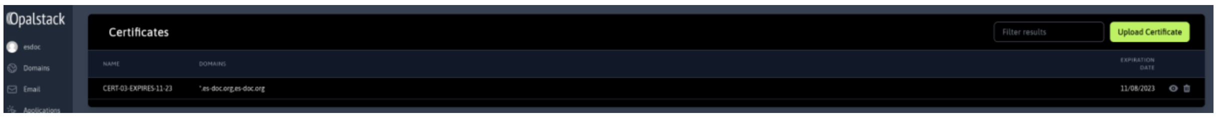
2.2. OpalStack migration

One of the reasons that motivated the ES-DOC team to undertake the documentation tasks described above is the unforeseen change in infrastructure provision that occurred at the end of 2020 and forced some unexpected tasks on the personnel.

The "WebFaction" cloud hosting service that had been used for some years withdrew some of the features that ES-DOC service relies on, prompting the need to find a new hosting service. Transferring to the new "OpalStack" service highlighted areas where more resilience was needed and also provided a documentation and training opportunity so that other members of the team (or anyone else) could learn how to deploy the ES-DOC services.

2.3. SSL certificates management

Currently the ES-DOC stack relies on an SSL certificate provided by GoGetSSL. The certificate has to be renewed bi-annually, albeit a confirmation process is necessary every year in order to maintain an operational certification process. This is one of the tasks undertaken by the ES-DOC dev-ops personnel currently on "OpalStack" infrastructure.



NAME	DOMAINS	EXPIRATION DATE
CERT-03-EXPIRES-11-23	*.es-doc.org,es-doc.org	11/08/2023

Figure 4: Opalstack certificate management dashboard

3. Controlled vocabulary

The ES-DOC eco-system revolves around **controlled vocabularies**, i.e. scoped collections of terms mandated by vocabulary authorities. Such authorities include ES-DOC, WCRP, ECMWF and Corpernicus.

Processing vocabularies from a diverse set of sources is problematic. Serving them in a standardised fashion to a plethora of tools in need of accessing this key data is complicated. To address this issue the ES-DOC team created a special purpose utility library called **Pyessv** (**P**ython **E**arth **S**cience **S**tandard **V**ocabularies). The primary task of Pyessv is to **normalise** vocabularies and expose them to upstream tools & applications. To do this, Pyessv relies on an archive, a python library and a web-service, which while helps with versioning, hinders the use and exploitation of the information by 3rd-party applications.

The archive's directory layout is shaped around the Pyessv data model. This model consists of a set of vocabulary **terms**, organised under a scoped **collection**, having been mandated by an **authority**.

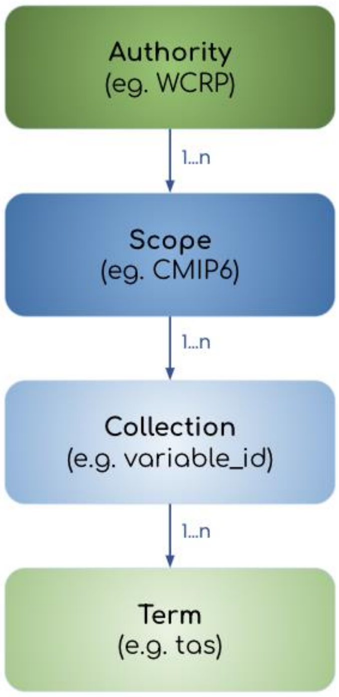


Figure 5: Pyessv domain model

3.1. Pyessv archive

The Pyessv archive hosts a collection of standardised vocabularies written in a specific normative form. This archive is populated from official vocabulary sources provided by official sources such as WCRP and ECMWF, mostly in the form of JSON files stored in Github repositories. Under each and every scope level, a manifest file details the set of collections and terms held underneath. Every term is mapped to a single JSON file as follows:

```
{authority}/{scope}/{collection}/{term}.json
```

For example:

```
wcrp/cmip6/experiment/piControl.json
```

3.2. Pyessv library

The primary task of Pyessv is to **normalise** supported vocabularies. Once normalised, the vocabularies are published to an archive and made accessible via a simple web-service API and a dedicated python library, mainly revolving around setting the archive directory, which is necessary for the functioning of the package.

The Pyessv Python package is distributed on Pypi and is supposed to be a lightweight, pragmatic and simple to use manager of controlled vocabulary destined to 3rd-party application users.

Technically speaking, the package uses the same data model explained above in the archive paragraph. The Pyessv library not only normalises the data model associated with a controlled vocabulary entity, but also normalises the **canonical name** associated with any vocabulary entity.

The original names processed by the writer are still retained by Pyessv (as the raw_name field). Furthermore names that may appear in a user interface are stored as a label field. Thus whilst the canonical name is normalised, both the raw name and label associated with a vocabulary entity is retained at the **discretion** of the vocabulary authority.

As stated above, ES-DOC vocabularies are derived from a diverse array of sources. For each source, e.g. ECMWF, a dedicated Pyessv **writer** is allocated the task of mapping the source vocabularies to the Pyessv data model, i.e. the task of normalising.

The Pyessv writers are simple python scripts that rely on existing vocabulary sources to convert them to Pyessv archive.

The Pyessv library is also available in javascript form in order to satisfy the front-end application needs and requirements such as the errata service or the ES-DOC explorer.

3.3. Pyessv web service

The Pyessv web API exposes a set of endpoints for exposing controlled vocabularies to upstream tools & applications. It runs over the top of an in-memory cached instance of the Pyessv-archive. It serves vocabularies in JSON format on demand.

The purpose of the web API form of the Pyessv toolkit is to provide 3rd-party applications with Pyessv functionalities without having to install the archive and the Python library in their environment. The idea is to simply query the proper endpoints when needed. The web service endpoints are as follows:

- **https://pyessv.es-doc.org:**
  - **HTTP GET**
  - Returns web service operational status
- **https://pyessv.es-doc.org/1/retrieve/{authority}/{scope}/{collection}/{term}**
  - **HTTP GET**
  - Returns a vocabulary term or set of terms
  - Valid authorities : copernicus, ecmwf, esdoc, wcrp
- **https://pyessv.es-doc.org/1/validate-identifier**
  - **HTTP GET**
  - Validates an identifier against archived vocabularies
  - Param: identifier
  - Param: identifier type

    - Param: project
- **https://pyessv.es-doc.org/1/validate-identifier-set**
    - **HTTP POST**
    - Validates a set of identifiers against an archive
    - Param: identifier
    - Param: identifierType
    - Param: project

The web service is currently deployed on an "OpalStack" infrastructure, the deployment process as well as the management of day-to-day tasks of maintenance and system administration are described in the ES-DOC technical documentation.

Useful links to everything Pyessv are listed below:

- https://github.com/ES-DOC/Pyessv
- https://github.com/ES-DOC/Pyessv-archive
- https://github.com/ES-DOC/Pyessv-js
- https://github.com/ES-DOC/Pyessv-writers
- https://github.com/ES-DOC/Pyessv-ws

4. Common ontology and specialisations

Climate scientists wishing to document their work in a highly-structured manner can leverage a special-purpose ontology, i.e. a formalised data model, called the CIM (Common Information Model). Currently at version 2.2, the CIM has evolved over time based upon community feedback for the express purpose of documenting climatology experiments, simulation & models. The underlying schema is governed via a light-weight process that permits the ontology to evolve whilst still ensuring that associated tools & services are kept in sync.

The CIM schema is defined declaratively via a pythonic domain specific language. From the schema a wide range of assets are forward engineered such as mind-maps, code, documentation, etc. Such assets are used by upstream tools & applications, e.g. online documentation search engines.

The ES-DOC eco-system is built upon the CIM, a nascent standard capturing a wide range of information relating to the earth system modelling process. The CIM is now being leveraged by a variety of international model intercomparison projects.

The canonical CIM schema consists of a set of package scoped classes, in which each class encapsulates a set of properties. Classes may be organised hierarchically via a standard inheritance mechanism. A subset of the CIM classes are marked as documents thereby becoming eligible for inclusion within the ES-DOC Archive. In practical terms the CIM schema is defined declaratively as a set of python modules. Within each module are declared a set of class factory functions, each function returns a dictionary containing the metadata representing the class definition.

For scientists wishing to document their work in a loosely structured fashion, the ES-DOC project provides support for so-called specialisations. On a project by project basis, groups of scientists can come together to declare a hierarchy of topics whereby each topic is associated with a fine-grained set of questions.

As with the CIM, specialisations are declared by the community using a pythonic domain specific language. From the definitions are forward engineered assets that are in turn leveraged by upstream tools & applications, e.g. the specialisations web viewer. Each specialisation set consists of a so-called top-level topic plus a set of high-level topics. The topics are declared within python modules as structured dictionaries.

4.1. Metadata collection

We can illustrate the ES-DOC documentation workflow with the experiment documentation.

The experiments are documented using verbose **spreadsheets**. For each experiment some standard information is defined, then the relationships to other experiments, then the experiment's requirements.

The requirements are of different types: temporal constraint (simulation length describes the simulated period), ensemble, multi-ensemble, configuration and forcings. There are separate spreadsheet tabs for each type of requirement. The spreadsheet is parsed and transformed by the following set of scripts into CIM document, the scripts can be found under the Github repository here:

https://github.com/ES-DOC/esdoc-shell/tree/master/bash/cmip6/experiments/write_cim_documents

Those CIM documents are the documentation, and the ES-DOC tooling can then be used to visualise them in a number of ways. The model documentation follows the same workflow.

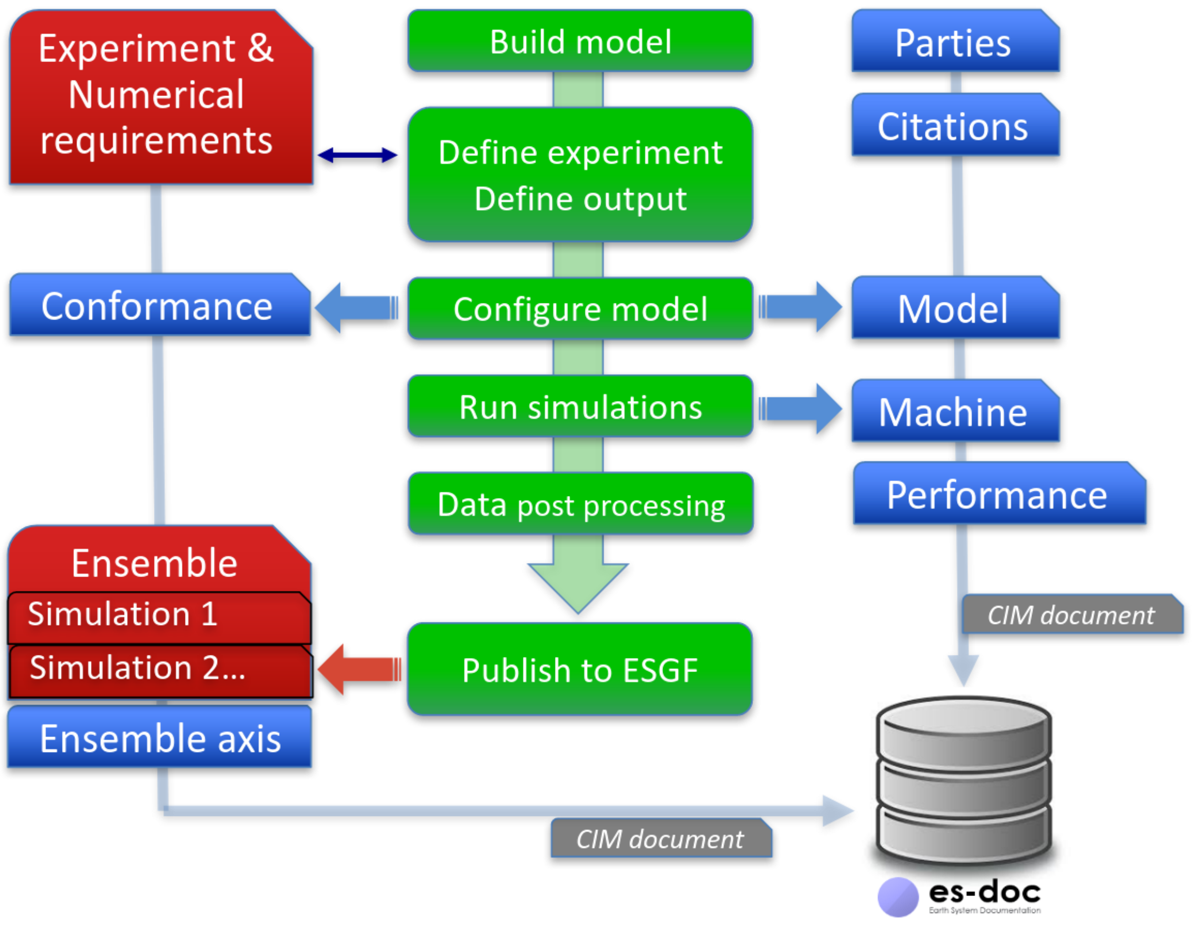


Figure 6: ES-DOC metadata collection workflow

4.1.1. Spreadsheets

The documentation of the CMIP6 project is captured using **spreadsheets** that will be auto generated based on some specialisation files. Those spreadsheets are then transformed into CIM documents that are accessible by the ES-DOC tooling.

The process of creating the model documentation begins with specifying so-called specialisations. Specialisation is simply a long word for a "topic map", i.e. a mind-map that structures and links the properties (e.g., dynamics, radiation scheme, etc.) of a climate system component (e.g. the atmosphere). A specialisation defines the scientific questions/sections that would be required to document the model. It can be seen as a list of questions to format into a form to be filled for any survey. For each "question" the specialisation details the title, if it is multiple choice question or free text, the allowed values, etc.

The specialisations revolve around a batch of so-called top-level specialisations, and then the realm specific specialisations. Each specialisation has its own Github repository that contains a set of python files in which the specialisations are defined - the definitions are actually quite similar to JSON files.

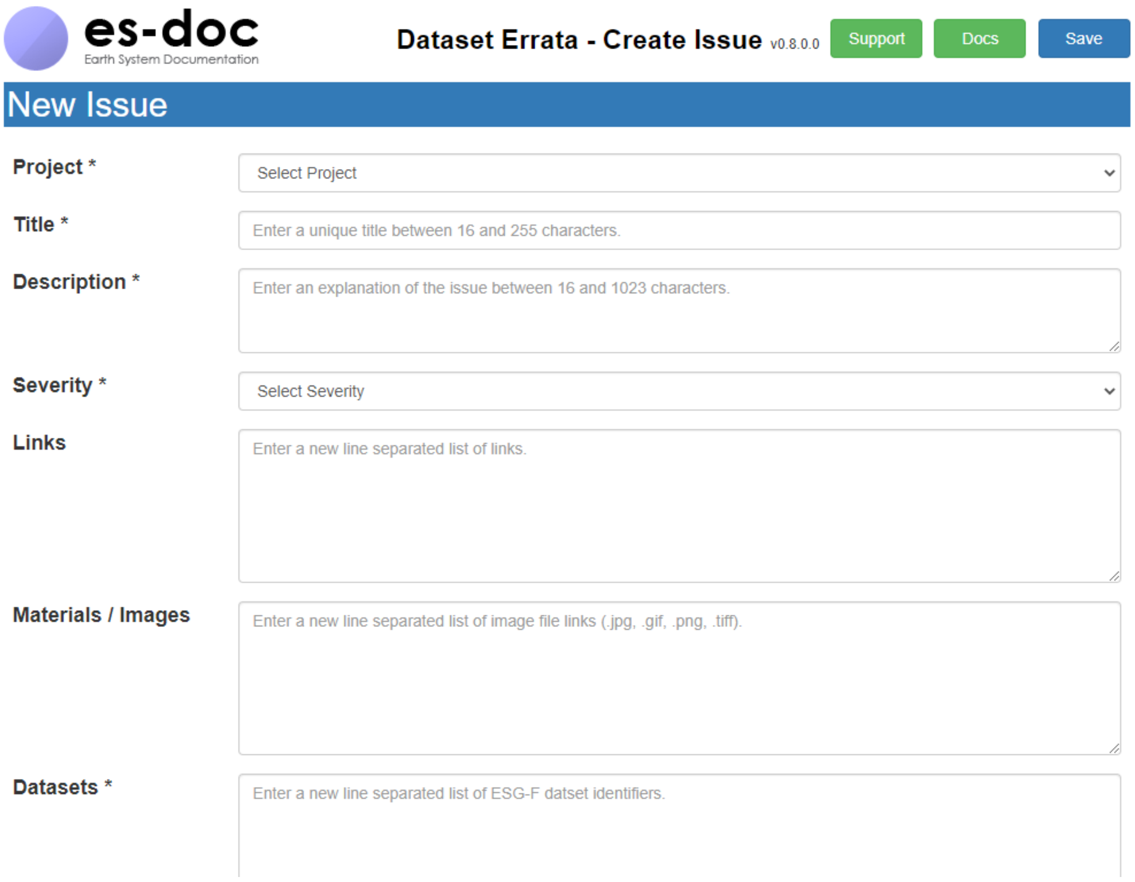
4.1.2. cdf2cim

Ultimately the ES-DOC eco-system serves to facilitate the creation of CIM compliant documentation. Once created they may be archived in the ES-DOC archive Github repository (https://github.com/ES-DOC/esdoc-archive) thereby being eligible for processing by upstream applications such as search engines & documentation viewers. Other than the Pyessv archive already discussed in 3.1, ES-DOC relies on two other forms of archive, CDF2CIM and Documentation archives. The CDF2CIM archive is a collection of JSON files extracted automatically via the ESGF publisher from netCDF files. This will be further explained in an archives dedicated section later on.

4.1.3. Web forms

Another way of harvesting information from different actors within the ES-DOC community and the climate simulation community in general is through webforms. This is a more user friendly way of indicating a specific list of information, however it requires a higher technical investment to put in place and user support to help users when in need.

The ES-DOC errata service relies on web-forms to collect errata details, the fields are either mandatory or optional and are clearly described through box texts. It relies on backbone JS framework. When a field is wrongly filled, a descriptive error message is displayed in a prompt to the user to help them enter the correct value.


Dataset Errata - Create Issue v0.8.0.0

New Issue

Project *	Select Project
Title *	Enter a unique title between 16 and 255 characters.
Description *	Enter an explanation of the issue between 16 and 1023 characters.
Severity *	Select Severity
Links	Enter a new line separated list of links.
Materials / Images	Enter a new line separated list of image file links (.jpg, .gif, .png, .tiff).
Datasets *	Enter a new line separated list of ESG-F datset identifiers.

Figure 7: Errata creation webform

4.2. Pyesdoc: document manager

The CIM ontology combined with project specific specialisations form the backbone of the ES-DOC project. A pythonic declarative mechanism, built upon a domain specific language, allows the scientific community to take the **lead role** in defining the documentation requirements.

Taking as input the vocabulary, ontology & specialisation definitions, a code generator forward-engineers code that is copied into the **pyesdoc** library. The pyesdoc python library underpins the entire ES-DOC tooling chain. It streamlines vocabulary, ontology & specialisation usage. It thus allows upstream assets, tools & applications to be generated and/or developed.

The Pyesdoc package can be used to manage the lifecycle of the information, allowing the developer to programmatically create, validate, read, write and delete CIM documents. It also allows programmatic search queries against the ES-DOC api. The library also allows viewing of archived ES-DOC documents that are stored in JSON format, in either HTML or PDF as required. The HTML formatted documents are subsequently rendered via the ES-DOC web front-ends. In a similar fashion, Pyesdoc also allows users to interact programmatically with specialisation definitions.

4.3. Archives

Ultimately the ES-DOC eco-system serves to facilitate the creation of CIM compliant documentation. Once created they may be archived in the ES-DOC archive Github repository (https://github.com/ES-DOC/esdoc-archive) thereby being eligible for processing by upstream applications such as search engines & documentation viewers. Other than the Pyessv archive already discussed in 3.1, ES-DOC relies on two other forms of archive, CDF2CIM and Documentation archives.

The CDF2CIM archive hosts a collection of JSON files extracted from NetCDF files. The file extraction process is an extension of the ESGF publisher. By using the appropriate flag when running the publisher, a directory scan is performed and for each NetCDF file a cdf2cim JSON file is created, this file is given a unique identifier based on the hash of its contents.

The ES-DOC archive on the other hand hosts a collection of CIM document files, published within the context of various projects. Most documents start life as spreadsheets which are then mapped to CIM documents via dedicated scripts. Other documents have been directly published as CIM documents using the Pysedoc library.

The errata system is another form of ES-DOC archive, which was a response to a known issue within climate modelisation communities. Throughout the CMIP5 exercise, few local projects spawned trying to archive and index known and reported issues discovered in climate simulation data. Prior to CMIP6 launch, IPSL started the initiative to create a federated system that enables users to create, search and consult errata information. The errata system saw the light in June of 2018 as an integral part of the ES-DOC software stack.

During the WP10/JR3, the errata development was focused around answering growing requirements from the users, also formalised by the ESGF WIP. The main request revolved around relaxing the authentication and authorization requirements put in place to protect the write privileges in the errata system. Instead, a suggestion system is put in place with new moderation features that would allow data providers to either accept or reject suggested errata. Any suggestions that go unanswered for a period of time will be published automatically, albeit a flag will indicate the issue in question wasn't validated by the data provider.

This new mechanism aims at bypassing the bottleneck created by the unavailability of data providers to manage their own errata. Which led to a redesign of the front-end and back-end of the service to support these new features. This evolution has been in the works for months and has been successfully deployed to a test environment by the end of December, with plans to hold a beta testing phase with select test candidates which will hopefully be concluded by a production deployment before June 2023.

4.4. Front ends

The first and most visible inlet for documentation consumption is ES-DOC’s browser-based user interface or ‘Web UI’, which has a landing page domain of es-doc.org. Under this root domain, there are various web services that provide applications which are mapped to specific domains.

The core ES-DOC web assets accessible from there are applications that enable end-users to view and compare documentation across all models. The latter application, the comparator, only provides comparisons for selected aspects of CMIP models (only CMIP5 models at the time of writing - extensions to provide functionality for CMIP6 models are still under development). The view application allows users to search, select and view any available documentation.

There are two further major web resources which ES-DOC interfaces to, namely services for management and dissemination of errata and of data citation.

The (CMIP6 Data) Citation Service is managed by DKRZ (hence also recorded elsewhere for this deliverable) and hosted outside of the ES-DOC web UI. However, it is intrinsically connected to the ES-DOC front-end ecosystem via the “further info URL” landing pages (see next section).

The main access point to the errata system has always been the index page, this page has been a focal point of the evolution towards a more open errata system. The challenge is to make suggested errata visible without compromising the integrity and quality of information.

5. Satellite project support

More document types have been made available yet for creation (by the CMIP6 modelling groups) and consumption (by users of the CMIP6 outputs), namely Machine and Performance documentation. These documents describe the machines on which CMIP6 simulations were run, and the performance characteristics of those simulations (such as the number of model days that were simulated per real day). The experiment documents have been enhanced by the addition of a document versioning framework, and the addition of a new extended description field that allows the documents to be more interoperable with the Copernicus Climate Change Service (C3S) project, and portals with a non-expert user base. The ES-DOC comparator for comparing model descriptions has been updated to work for the CMIP6 models, as well as the existing CMIP5 model functionality.

Documentation support for the CORDEX project has been implemented. This currently comprises the ability to document and publish regional climate model descriptions, which is the primary use case. Most CORDEX experiments are identical to CMIP6 experiments, which are already documented; an extension to document other CORDEX experiments may be required in the future.

A small amount of progress has been made on extending ES-DOC to the obs4mips project, providing observational datasets for model intercomparison projects. This has not resulted in a functional service yet, but the needs of the obs4mips community have been discussed with a view to making a subset of the ES-DOC functionality available during 2022.

ES-DOC experiment documentation was written for the Covid-MIP addition to CMIP6. Covid-MIP experiments investigate the climate implications of different economic recovery scenarios from the COVID-19 pandemic. The ES-DOC information was in place ahead of the deadline for inclusion in the IPCC 6th Assessment Report.

6. Conclusion

In this report we proposed a run-down of the ES-DOC software and services stack, that was made possible through IS-ENES work packages. While the main target set for ES-DOC is to document and provide a set of services and infrastructures that help guide users, inevitable questions arise about the maintenance and the further development of the stack in the future. With IS-ENES 3 project coming to an end, new funding venues are needed to ensure manpower needed for this. This includes but not limited to:

- the efforts of maintaining the hosting infrastructure (currently within a 3rd-party provider, but it still needs attention to resolve certificate issues, manual service restarts, billing details etc.), minor development maintenance effort to mitigate security risks and move away from deprecated technologies,
- providing user support which is a big part in services such as the errata by moderating content and users,
- generating the necessary spreadsheets and coordinating with model providers to feed into the documentation hub, this is necessary for model and experiment documentation but also machine and performance.

However, it was an important goal and task to be achieved, during WP10, to invest in generating the necessary documentation, resources, and skill transfer to minimise the required overhead and resources to maintain the service as a bare minimum target. The work towards this target was crowned during the ES-DOC face to face meeting that occurred in Paris, november 2022, during which dev-ops procedures were documented and tested on independent virtual machines to test their resilience, and important discussions about the different responsibilities of the ES-DOC community members were further cemented.

On a more optimistic note, potential projects that would include semantic web standards implemented in the handling of controlled vocabulary remains a mid to long-term objective for ES-DOC, depending on funding and the community's interest/need for the project. We believe this effort would massively improve the way we catalogue, explore and present data.