



**World Meteorological Organization**

**COMMISSION FOR CLIMATOLOGY**

**Expert Meeting on WMO Climate Data  
Management System Strategy**

Geneva, Switzerland, 12-14.XII.2018

**EM-WCDMSS/Doc  
N(3.2)**

Submitted by:  
Steve Foreman

14.XII.2018

---

## **GOVERNANCE OF DEVELOPMENT AND MAINTENANCE OF THE REFERENCE CDMS TOOL SET**

This document will become Document 6.1(4) of the 2019 workshop.

**REVIEW COMMENTS**

<b>Version / No</b>	<b>Reviewer</b>	<b>Issue</b>	<b>Notes</b>	<b>Response</b>
0.1/1	ET-DRM	Emphasize the project principle of “free open source” development		Amended para 4.1.1.2 of the strategy document. Also updated the purpose statement in this document.
0.1/2	ET-DRM	Version control: users should be encouraged to offer modifications back to the trunk – and provide mechanisms to make it easier to add functionality (including in interfaces) (such as making the interface components part of the database rather than coded in)		Added new paragraph 3.2.3.

## **DISCLAIMER**

### **Regulation 43**

Recommendations of working groups shall have no status within the Organization until they have been approved by the responsible constituent body. In the case of joint working groups the recommendations must be concurred with by the presidents of the constituent bodies concerned before being submitted to the designated constituent body.

### **Regulation 44**

In the case of a recommendation made by a working group between sessions of the responsible constituent body, either in a session of a working group or by correspondence, the president of the body may, as an exceptional measure, approve the recommendation on behalf of the constituent body when the matter is, in his opinion, urgent, and does not appear to imply new obligations for Members. He may then submit this recommendation for adoption by the Executive Council or to the President of the Organization for action in accordance with Regulation 9(5).

© World Meteorological Organization, 2018

The right of publication in print, electronic and any other form and in any language is reserved by WMO. Short extracts from WMO publications may be reproduced without authorization, provided that the complete source is clearly indicated. Editorial correspondence and requests to publish, reproduce or translate this publication in part or in whole should be addressed to:

Chairperson, Publications Board  
World Meteorological Organization (WMO)  
7 bis, avenue de la Paix  
P.O. Box No. 2300  
CH-1211 Geneva 2, Switzerland

Tel.: +41 (0)22 730 84 03  
Fax: +41 (0)22 730 80 40  
E-mail: [Publications@wmo.int](mailto:Publications@wmo.int)

#### **NOTE:**

The designations employed in WMO publications and the presentation of material in this publication do not imply the expression of any opinion whatsoever on the part of WMO concerning the legal status of any country, territory, city or area, or of its authorities, or concerning the delimitation of its frontiers or boundaries.

The mention of specific companies or products does not imply that they are endorsed or recommended by WMO in preference to others of a similar nature which are not mentioned or advertised.

This document (or report) is not an official publication of WMO and has not been subjected to its standard editorial procedures. The views expressed herein do not necessarily reflect those of WMO or its Members.

# GOVERNANCE OF DEVELOPMENT AND MAINTENANCE OF THE REFERENCE CDMS TOOL SET

*Note: for brevity, the tool set is referred to as OpenCDMS in what follows and the project as the OpenCDMS project*

## 1 APPROACH TO DEVELOPMENT AND MAINTENANCE

1.1 Development of open source software depends on volunteers, whether those volunteers are individual developers or organizations providing the time of their staff. Open source projects can also be “forked” - that is, at any time an individual or group of individuals can take a copy of the software and develop it independently of the original project. In consequence, those contributing to the project need to feel they have a say in the direction of the project and feel rewarded by their participation in it.

1.2 As an open source project, OpenCDMS has to adopt a management style that is suitable for the open source way of working. It has to balance ensuring that the project delivers the key functionality needed to implement the climate data management system specifications while ensuring that contributors feel that their needs and concerns are been met and addressed. Individual developers will need to feel valued and their contributions recognized and, even though the decisions may not always be what a developer would hope, they have to feel as if they have had a fair hearing. If these conditions are not met, then either the WMO community will be disappointed because the software does not deliver what was intended, or the software development may stagnate because there are too few developers available to develop and maintain OpenCDMS.

1.3 The approach taken by OpenCDMS is to have a very clear statement of the purpose of the project against which development proposals can be assessed, together with an open, consensus based decision making process on the content, design and internal standards of the project. WMO technical regulations, standards, specifications and recommended practices (even those in the Climate Data Management System Specifications, WMO-N0. 1131) will not cover all the procedures and algorithms that need to be implemented in OpenCDMS. The versions of the procedures and algorithms implemented in OpenCDMS are likely to become *de facto* standards, but the decision time scales needed by the OpenCDMS project will be much shorter than the WMO decision making processes. OpenCDMS, therefore, has to take care that its internal decision making process for such items produces a decision that is likely to be endorsed by a subsequent WMO process – a challenge that the majority of open source projects do not need to address.

## 2 MANAGEMENT STRUCTURE TO SUPPORT DEVELOPMENT AND MANAGEMENT

### 2.1 INITIAL MANAGEMENT

2.1.1 OpenCDMS cannot be launched as an open source project until its aims have been clarified, an initial overall architecture, design and development standards have been created, and core software infrastructure has been developed (Fogel, 2018, <https://producingoss.com>). This stage of the development will involve fewer developers than later stages, and the management structure can have more direct involvement with WMO structures than when the project is fully open source. Even at this stage, all discussions, development and software should be publicly visible so that potential open source contributors can gain confidence in how the project is being managed as open source even though it has not been launched formally.

2.1.2 Members of WMO and HMEI will be invited to provide developers to support the initial development of OpenCDMS and to offer technical facilities to support the project.

2.1.3 Initial development of OpenCDMS will be managed by a management team supported by technical advisors.

2.1.3.1 **Chair of management team:** A member of the CCI ET-DRM (Expert Team on Data Rescue and Management).

2.1.3.2 **Members of team:** a member from each of:  
CCI TT-CSTD (Task Team on Climate Services Toolkit and Downscaling)  
Vice President of Chy  
JCOMM ET-MC (Expert Team on Marine Climatology)  
CBS TT-IM (Task Team on Information Management)  
CliDE project  
ClimSoft project  
MCH project  
HMEI.

2.1.3.3 **Formal advisors to the team:** a member from each of:  
CCI ET-DDS (Expert Team on Data Development and Stewardship)  
CCI TT-GCP (Task Team on the Guide to Climatological Practices)  
CCI TT-TCI (Task Team on Tailoring Climate Information)  
JCOMM ETDMP (Expert Team on Data Management Practices).

2.1.4 The management team will provide quarterly progress reports to the CCI Management Group.

2.1.5 The prime tasks for the initial management team are:

2.1.5.1 Develop and publish the statement of purpose for OpenCDMS;

2.1.5.2 Develop and publish an initial architecture for OpenCDMS;

2.1.5.3 Develop and publish an initial design for OpenCDMS including the key interfaces needed for interoperability with other systems;

2.1.5.4 Prioritize the modules in the design, and identify those elements that have to be in place (in an initial form) before the project can launch itself as an open source project;

2.1.5.5 Set up the infrastructure to support development of OpenCDMS;

2.1.5.6 Set up the management structure for the open source project;

2.1.5.7 Set up the governance policies for OpenCDMS

2.1.5.7.1 Publish the development guidelines for OpenCDMS (the guidelines used by the GeoServer project are a useful guide to the types of policy that are required: <https://docs.geoserver.org/latest/en/developer/policies/index.html>);

2.1.5.7.2 Publish the Code of Behaviour for OpenCDMS;

2.1.5.7.3 Publish the software licence for OpenCDMS;

2.1.5.7.4 Publish the Contributor Agreement for OpenCDMS;

2.1.5.11 Set up any required legal framework for OpenCDMS;

2.1.5.12 Register any trade marks needed to ensure that the “official” version of OpenCDMS can be identified;

2.1.5.13 Launch the open source project.

## **2.2 STATEMENT OF PURPOSE**

2.2.1 The statement of purpose for OpenCDMS will provide the reference against which decisions on the development and maintenance of OpenCDMS will be judged. It will be used to prioritize development and to set the scope of OpenCDMS.

2.2.2 The initial statement of purpose is in the implementation plan for OpenCDMS:

*“The overall aim of the strategy for the reference CDMS tool set is to provide all WMO Members with a software solution based on an electronic database to assist with management of climate-relevant information that is provided under a perpetual and irrevocable licence that does not require payment for using the software and that permits modifications to the software. In its initial stages of development, the reference CDMS tool set in itself will not provide sufficient functionality to allow Members to implement it as their core CDMS tool set. This plan therefore seeks to allow Members to inter-operate their existing systems with their current software, and to allow Members without current access to an electronic CDMS tool set to implement one of the existing open source solutions (that do not have a sustainable model for maintenance and support) with the confidence that they can evolve to using the reference CDMS tool set.”* [Note - when developing the documents, the version of the purpose statement here has to be kept the same as in the project plan.]

## **2.3 TRANSFER TO OPEN SOURCE PROJECT**

2.3.1 The management structure of the project has to change when it is launched as an open source project. Although the main method of decision making will be by consensus, there will be times when action will be needed to trigger decisions or activities. This means that a light-touch management team will still be needed. Although the initial membership of that team will be decided by the management team for the first stage of the project, as soon as there are enough voting contributors for a vote to be meaningful, the management team should be chosen by vote. The management team itself should decide on its chair.

2.3.2 Voting contributors will be selected by vote among those already eligible to vote. The primary criterion for proposing someone as a voting contributor will be the value of their contributions to the project. Those contributing large amounts of code to the project are likely to be nominated, but in addition those who contribute to other aspects (such as documentation) should also be considered. Some contributors may not contribute anything tangible to the project, but frequently provide insight that supports or changes the direction of the project – such contributors might also be proposed as voting contributors. Any contributor may nominate anyone to be a voting contributor – but the decision is made by in secret by the voting contributors.

2.3.3 Major tasks in the developments may need a coordinator or leader. If such a leader does not arise naturally, then the management team might wish to suggest someone (after first checking with the individual). The response to the nomination might be consensus, or another proposal on which consensus can be reached.

2.3.4 A key role of the management team will be to make sure that the code of conduct is respected. Contributors are volunteers, and they and their contributions have to be valued even if they are not adopted by the project.

2.3.5 OpenCDMS will be developed as a set of releases that each builds on the functionality of the previous release. The management team will be responsible for proposing the critical content of each release (the content that must be present before the release is issued) and for leading the discussions towards agreement on the content.

## **2.4 COORDINATION WITH WMO**

2.4.1 Although WMO will provide the initial management team, in the longer term WMO as an organization will not have any direct role in the management of OpenCDMS. It has to be involved indirectly because of the interaction between OpenCDMS and WMO's regulatory regime.

## **3 TOOLS TO SUPPORT DEVELOPMENT AND MANAGEMENT**

3.0.1 OpenCDMS is expected to receive contributions from developers around the world. Early experience during the development of WIS demonstrated that care needs to be taken when choosing the tools to support a project – tools that are complex to use, or that do not prompt the user when an action is needed, are unlikely to be successful unless they are part of the day-to-day working environment of the developer. The tools used by OpenCDMS should be widely used and prompt users by email or social media when an action is needed. OpenCDMS hopes to create a large community of developers, reviewers and users, and the licensing for any tools supporting the development of OpenCDMS has to take this large number of potential users into account.

### **3.1 PROJECT COMMUNICATION TOOLS**

3.1.1 OpenCDMS needs a website to provide a central point of reference that allows potential users and contributors to find out more about the project. Although it may not itself provide downloads of the OpenCDMS software, it should link to where this can be done. Similarly, it should link to where the majority of conversations about OpenCDMS take place.

3.1.2 When they are small, many projects rely on ending emails among the participants as a way of communication. This has two major drawbacks for an open source project. The first is that email communications are private to those involved in the email exchange, whereas the communications in an open source project should (with very few exceptions) be open for anyone to see. The second is that even with the best will in the world, people will be left off email conversations. Open source projects typically use list servers and/or online forums as their main means of communication. A prime requirement of both of these is that the system should provide a record of the conversations with tools to allow topics to be linked and sorted, making it easier for users to find and follow discussions. OpenCDMS should seek to limit itself to a single communications tool, otherwise conversations will become disjoint and difficult to find as they move between tools. This does not prevent a solution that uses web forms, social media and email as ways of sending and receiving messages, as long as all the conversations are archived and visible from a single location.

3.1.3 Spanning the communications system and the version management system is a bug tracking system. More than just a means of recording discussions, the bug tracking system has to link between the bug report, discussions on how it may be resolved, and the modification to the software that corrects the bug. Many version management systems (see below) include a simple bug tracker. It may be sufficient to use the tools provided in the version management system as a bug tracker for the first few releases.

## 3.2 VERSION MANAGEMENT

3.2.1 With many developers working in parallel on openCDMS, it will be important that each can see what others are doing, and that the developments can be tested alongside each other. This is achieved using a version management system. There are strong advantages to using a version management system that works closely with the communication tool so that version changes can be linked to the discussions leading up to them, and issues with a software version can be linked back to it.

3.2.2 Choice of the version management tool is a key task for the initial management team – migrating between different version management systems can be difficult and lose contextual information, and even moving between installations of the same system on different hosts can lose information on the links between the contents of the version management system and the communications system.

3.2.3 Users will be allowed to modify their own copies of openCDMS. Experience with the franchise CDMS has demonstrated that this can cause problems if those users subsequently install an updated version of the software because changes to the official version (“trunk”) may overwrite or be in conflict with changes made by the users. OpenCDMS should therefore encourage users who modify the software for their own purposes to offer those changes back to the development project so that they may be included in future releases of the trunk or, if that is not possible, compatibility of the changes with updates to the trunk can be tested before the final release.

## 4 PROJECT MANAGEMENT

4.1 To thrive, OpenCDMS must be seen to issue new releases to the promised time scale. Although run as an open source project with volunteer effort, project management will be needed to achieve this.

4.2 Like traditional project managers, those developing openCDMS will need to know what has to be delivered, how much effort and time it will need to deliver each of the items that are planned for a release, who is working on creating them, whether some items rely on others that are also under development being ready before they can be tested, and whether items are going to be delivered on time. Whereas in a traditional project, this information would allow the project manager to re-direct resources, negotiate time scales and deliverables or to increase (or decrease) the resources available, an open source project does not have such a direct control mechanism.

4.3 Within an open source project, the developers themselves decide where they are going to make their contribution and when they are able to devote time to the project. They will make those decisions based on their own interests and availability, but also on the needs of the project. So, all traditional project information is needed, but instead of being driven from the top, the decisions are mostly made by the developers themselves. Inevitably, there will be parts of the release that are not receiving the attention they need. In this case the management team will need to draw attention to these and encourage participation from contributors to them.

4.4 The most important project information for a stage in the development of OpenCDMS is the specifications of the required content for that stage, with indications of the effort that is expected to be needed for each item together with the skills required to deliver them, and the dependencies between the items (so that people know the order in which things have to be done). Once the development of the stage starts, communication between the developers on who is working on each item and whether there are issues becomes a critical resource. Some projects structure this communication through kanban boards (<https://en.wikipedia.org/wiki/Kanban>). This may prove adequate for OpenCDMS, at least for the initial stages of development.



