



OEO CMM 2+ in 2001

Requirements on Odin to Support the CMM Needs of the SCM KPA

Tim Gill & the OCAB

Preface

- Purpose:** *A working document to arrive at a clear statement of requirements on Odin for supporting the needs of the CMM SCM KPA, as expressed in the [15-May-2001 minutes](#) of the OEO SCM KPA team meeting.*
- Audience:** *The OCAB and OEO SCM KPA teams.*
- Related Documents:** *The [template](#) for an HP-UX SCM Plan; the [EIAL SCM KPA Plan](#) guiding some of the OEO SCM KPA work.*
- Terminology:** *TBD... What definitions do we use? There are [3 terms](#) in the [OEO Glossary of Terms](#) with the word "baseline" in them, 2 from an IEEE standard. Are any of these accurate for current HP-UX development practice or are they simply IEEE definitions?*
- Versions** *Not tracked particularly; pay attention to the date in the footer, which is the date the document was last saved. Also look at the history page in Odin for this document.*

Table of Contents

| | |
|--|----------|
| 1. Introduction _____ | 3 |
| 2. Overall Comments on the Initial Requirements _____ | 4 |
| 2.1 Comments: Cash _____ | 4 |
| 2.2 Comments: Roth _____ | 4 |
| 2.3 Comments: Gill _____ | 4 |
| 3. Original Requirements and Comments _____ | 5 |
| 3.1 Some Documents Need to be Included with the Baseline of a Project _____ | 5 |
| 3.2 Need a Unique Identifier _____ | 5 |
| 3.3 Access to Previous Baselined Versions _____ | 6 |
| 3.4 Tracking Versions of Docs, Retention, Pruning _____ | 6 |
| 3.5 Validation _____ | 7 |
| 3.6 Change Control _____ | 7 |
| 3.7 Connect Documents to Relevant Source Code and Vice Versa _____ | 7 |
| 3.8 Retrieval of Older Version for Review and Diffmarking _____ | 8 |
| 3.9 Connect DMS Baselines with SCM Baseline Names _____ | 8 |
| 4. Re-Statement of Clarified Requirements _____ | 9 |
| 4.1 Tim & Paul's Re-Stated Requirements (June 22) _____ | 9 |

1. Introduction

At its May 15 meeting the [OEO SCM KPA Team](#) came up with some requirements that it felt Odin needed to support to allow the SCM KPA Team to implement SCM policies and procedures for the configuration management of documents, and captured them in the [meeting minutes](#). They requested the [OCAB](#) to comment on those requirements. Several e-mail messages among OCAB members provided some initial comments. This document is an attempt to capture those initial requirements and the OCAB comments and allow the OCAB, through a couple of document iterations, to have an agreed upon set of requirements.

The requirements in the minutes were as follows:

Requirements From an SCM Point of View:

1. There are some documents that need to be included with the baseline of the project

How frequent: on the order of each IC

Which documents: defined in the SCM plan

2. Need a unique identifier
A path name such as a URL could be sufficient
3. We must have access to previous baselined versions

What should the "time-to-live" be for a baselined document?

Some baselined versions should be removed over time

On the code side: a baseline is never removed

4. We want to keep track of versions of docs in Odin.
Some versions need to be retained forever (baselined).
Other versions should be pruned on a more regular basis.

Disk space should not be an issue here.

5. Validation.
For all of the documents that are being managed, we need a way of confirming that documents are being managed the way we believe it is.
6. Change control.
eg Requirements doc under CRT control
Need a way to control changes to a document
-> Implies the need of a "gate keeper"
7. Connect documents to relevant source code and vice-versa
8. Retrieval of older versions (for review and diff marking)
9. Connect DMS baselines with SCM baseline names. The DMS and SCM tools need to support a common name space for baselines.

2. Overall Comments on the Initial Requirements

This section contains comments by OCAB members on the overall set of requirements.

2.1 Comments: Cash

- a. One puzzlement I feel about these requirements is precisely what we (or Odin) are supposed to do about them. Is it enough if we assure them that these requirements can, in principle be implemented? Or are we supposed to say how to implement them? Or are we actually supposed to implement them? So, what form should our response take?
- b. One thing--it looks like we are certainly going to be adding some metadata fields. Let's make up a list and do it all at once rather than piecemeal. That way, we get to assess the impact of the changes, eliminate any redundancies, and I only have to rebuild the search indexes once.

2.2 Comments: Roth

[Replying to Cash]: I think we're expected at least to figure out how to implement these things. I think we're probably expected to own the implementation as well, although we're certainly free to use the SCM team's leverage to get help if needed.

2.3 Comments: Gill

It seems to me that these requirements are on both the capabilities and configuration of Odin as well as on the SCM procedures defined to the SCM KPA (which are the responsibility of the SCM KPA and not of the OCAB); I think the final requirements must reflect that. Some of what the SCM KPA team is trying to accomplish to meet the CMM objectives requires such procedures, and making them part of Odin may not be the right way to do it.

I am also particularly concerned about the notion of having baselined versions of documents done every time there is an IC, as this seems unnecessary and wasteful. What is the real need here? We need to dig deeper.

As I think about the plain use of DocuShare versioning to handle some of these requirements, I am concerned by what appears to be DocuShare's lack of indexing on older versions of documents. For example, if version 11 in Odin of a requirements doc is the approved version of a feature in "release 1.2" of HP-UX and version 23 in Odin is the same for "release 1.3" of HP-UX, searching Odin will not turn up document text or attributes for version 11 that are not in version 23. Yes, this is an implementation issue, but it is important to really get the requirements clear in this regard.

In summary, I think that getting these requirements right, and doable, as quickly as possible requires that the people who sit on both the OEO SCM KPA team and the OCAB need to work this all out.

3. Original Requirements and Comments

This section simply breaks the minutes' version into its parts and includes the e-mail and other comments of OCAB members on those initial requirements.

3.1 Some Documents Need to be Included with the Baseline of a Project

1. There are some documents that need to be included with the baseline of the project

How frequent: on the order of each IC
Which documents: defined in the SCM plan

3.1.1 Comments: Cash

I still don't have a handle on what this actually means. One way to read it is that the documents must be checked into ClearCase. Another reading would be that the baseline must be identifiable in some way--for example, there is a list of links that pertains to each baseline where the links point to the relevant documents in Odin. In either case, the project team would have to take care of this, either by checking the docs into ClearCase, or by drawing up a list of hyperlinks.

3.1.2 Comments: Roth [in reply to Cash]

- a. I think what it means is a little more generic than that. It simply means that at various points in time, probably on the order of monthly, we will be expected to take some form of a snapshot of certain documents in Odin. (What form is up to us.) That snapshot would be given a certain label, not under our control. It must then be possible, forever after, to retrieve the copy of any document that was current at the time of any labeled snapshot (i.e., baseline).
- b. In Odin terms I expect this means that: (a) there is a specific keyword that we will tell people to associate with any document that should be included in baselines (b) we will need a custom script, to be run when a baseline is taken, that will find all documents with this keyword and copy (not link) them to a different part of Odin (c) that different part of Odin will be organized by baseline label (d) those copied documents would be read-only and never purged
- c. Notice that the document owner has no tasks in this procedure, except for applying the correct keyword to the document.

3.1.3 Comments: Gill

- a. The phrase "on the order of each IC" implies being able to retrieve 10-20-30 versions of a document, since there are often that many IC's of the software. Are there every that many usable versions of a document, and do they really want to be able to retrieve that many (interim) versions? I cannot imagine this is useful or really what is wanted. Do they care about versions of documents in between "Approved" states? (Perhaps this whole topic hinges to some degree on whether one views (controlled) documents as documenting code that is written or defining what is to be coded...).
- b. What does "include" mean here? Baselines in ClearCase are typically marked with a ClearCase label, allowing retrieval of a particular version by name. Do they want a way to retrieve a document version by baseline label, or just the ability to associate a particular version of a document with a software baseline?
- c. Does "baseline" mean something different when talking about a document as opposed to source code? Are they using the definition of baseline in the [OEO glossary](#) (and there are 3 terms with "baseline" in them)?

In response to Steve's comment, I think he is suggesting a solution based on a face-value reading of the initial requirements statement. I think the requirement needs to be discussed and clarified before we think of implementation, because what they seem to be asking for seems excessive.

3.2 Need a Unique Identifier

2. Need a unique identifier
A path name such as a URL could be sufficient

3.2.1 Comments: Cash

Odin can certainly provide that.

3.2.2 *Comments: Gill*

- a. DocuShare supports a “permanent” URL to the latest version of a document, or to a folder. How this capability is used will depend on final requirements, but DocuShare seems to provide the needed mechanism.
- b. If we only use the versioning provided by DocuShare, then we may have problems in providing the capabilities desired. DocuShare appears not to provide full-text search of *previous versions*, so if one wants to search for the “IC9” version of a document (based on a document attribute or contents of the document) when the current version is IC20, we may not be able to do that if just using the DocuShare notion of versions.
- c. The URL that DocuShare provides is to a particular format of the document (e.g., source file in Word or HTML version). How this is used needs to be worked out.

3.3 Access to Previous Baselined Versions

3. We must have access to previous baselined versions

What should the "time-to-live" be for a baselined document?
 Some baselined versions should be removed over time
 On the code side: a baseline is never removed

3.3.1 *Comments: Cash*

- a. Sure, as long as the docs remain in Odin, they’re accessible. But you have to know how to find them (e.g., have an URL to a specific historical version of a doc).
- b. What does "time to live" mean? Are they saying they actually want us to remove documents from Odin? Why? If they don’t want to reference certain docs as "baseline" docs, there’s nothing to force them to do so.

3.3.2 *Comments: Roth [in reply to Cash]*

- a. I don’t think using Odin’s history mechanism will satisfy the SCM folks. Even omitting concerns about automatic purging of old versions, there’s no convenient way to associate a particular baseline label with a particular old version.
- b. They’re saying that some docs may be baselined every month during a two-year release cycle, but that after the release is over, the old baselined copies aren’t needed any more. Other docs may be needed forever.

3.3.3 *Comments: Gill*

- a. What is the “time to live” of a source code baseline?
- b. Are there particular kinds of baselines for which the earlier versions of documents are desired? In my experience, one wants access to the “approved” versions that are used as the basis for the corresponding development activity (requirements, design, testing, etc), and there may be a few approved versions. But do they need access to the various draft versions?
- c. Do they care what format of a document the access is provided for (Word, Frame, PDF, HTML, etc?) This affects the URL used to reference it.

3.4 Tracking Versions of Docs, Retention, Pruning

4. We want to keep track of versions of docs in Odin.
 Some versions need to be retained forever (baselined).
 Other versions should be pruned on a more regular basis.

3.4.1 *Comments: Cash*

- a. Do they want some kind of support for doing this, such as a new metadata field for "version"? Or will it be enough just to put what version the document is in the comments?
- b. Again, I’m not sure why SCM should require "pruning" of any sort. In general, aging documents and getting rid of old ones is a good idea, and it is certainly important to identify docs you want to archive. But I’m not sure what requirements SCM wants to impose with regard to what gets pruned.

3.4.2 *Comments: Roth [in reply to Cash]*

- a. [null]
- b. I don't think they are either. This is not a well-formed requirement as yet.

3.4.3 *Comments: Gill*

- a. Odin supports simple version control: copies of n versions. The default is 4 but this can be set on a per document basis (and perhaps at a higher level, such as a folder). How these versions map to the versions that get retained forever and the ones that do not must be defined.

3.5 Validation

- 5. Validation.
For all of the documents that are being managed, we need a way of confirming that documents are being managed the way we believe it is.

3.5.1 *Comments: Cash*

Sounds like they want a script that reads some kind of list of documents, then verifies whether they are present in Odin.

3.5.2 *Comments: Roth [in reply to Cash]*

That's the easy part. They also want a way to make sure that each baselined version really does contain the document that was current at the time of the baseline, i.e., that Odin's history mechanism is working. I think we can finesse this one.

3.5.3 *Comments: Gill*

- a. Assuming the documents being managed are defined in an *SCM Plan for Documents* for each project, this sounds like just being able to find the current and any baselined versions of the documents defined to be under control. This can be done simply using the web UI of Odin. Do they want a programmatic ability to do this, such as a script to be run on the Odin server to verify the state of the defined documents?
- b. Does "managed" mean *version control* or *change control*?

3.6 Change Control

- 6. Change control.
eg Requirements doc under CRT control
Need a way to control changes to a document
-> Implies the need of a "gate keeper"

3.6.1 *Comments: Cash*

This is the responsibility of the owner of the document, isn't it?

3.6.2 *Comments: Roth [in reply to Cash]*

Yes. If they decide to centralize control over key documents, then the central authority becomes the owner from Odin's perspective. Not our problem.

3.6.3 *Comments: Gill*

- a. Is the control at the level of the author or at some higher level, like a manager of change control board?
- b. They are distinguishing *change control* here from *version control*, right?

3.7 Connect Documents to Relevant Source Code and Vice Versa

- 7. Connect documents to relevant source code and vice-versa

3.7.1 *Comments: Cash*

Another meta-data field?

3.7.2 *Comments: Roth [in reply to Cash]*

Perhaps. We discussed this in our meeting, as well.

3.7.3 *Comments: Gill*

- a. For connecting documents to source code (the first sense), do they care if the connection is maintained within the document itself vs. within the Odin metadata?
- b. For connecting source code to documents (the second sense), how ClearCase handles this is not up to the OCAB. Odin can provide a URL that ClearCase can store if desired, once the mechanism/procedure for how baselined document versions are referenced is defined.

3.8 Retrieval of Older Version for Review and Diffmarking

8. Retrieval of older versions (for review and diff marking)

3.8.1 *Comments: Cash*

Retrieval is no problem. It was pointed out during the meeting that "diff marking" must be done by the tool in which the document is written. Odin does not have this functionality.

3.8.2 *Comments: Gill*

- a. Previous requirements (3, 4) provide access to older versions as needed. Since DocuShare does not provide any diff-marking capabilities, this requirement must be met by a procedure defined for doing SCM of documents.

3.9 Connect DMS Baselines with SCM Baseline Names

9. Connect DMS baselines with SCM baseline names. The DMS and SCM tools need to support a common name space for baselines.

3.9.1 *Comments: Cash*

Yet another metadata field? Or can the version field be used for this purpose?

3.9.2 *Comments: Roth [in reply to Cash]*

Agreed. It will also allow us to redesign the properties form for better usability.

3.9.3 *Comments: Gill*

- a. How is this different from requirement 2 on "unique identifier"?
- b. And again, do they care if the connection is maintained within the document itself vs. within the Odin metadata?

4. Re-Statement of Clarified Requirements

Paul Greathouse and Tim Gill met for an hour on 21 June and then again on June 22 to attempt to hash out what the requirements should really be, given our role in both the OCAB and the SCM KPA Team and our interest in the SCM of documents. The 2 sub-sections below contain our attempts from those meetings.

4.1 Tim & Paul's Re-Stated Requirements (June 22)

Some documents in the HP-UX development process must undergo a *baselining* process. This process will associate certain versions of the documents with a particular software baseline such as an IC or an LR of a particular HP-UX release and make those versions of the documents retrievable later. The documents that must be baselined will be defined in the *SCM Plan for Documents* for the particular project.

For this, the combination of Odin and some SCM procedures for documents must, together, support the following requirements for baselined documents:

1. Odin must allow, but not require, each baselined document to have particular versions be put in a baselined state within Odin on the same frequency as the corresponding software is baselined. A particular baseline of a baselined document is retrievable as long as the software baseline exists. No purging of baselines of a baselined document may be caused automatically by Odin administrative functions.
2. The baseline identifiers used in Odin for a document must come from a common namespace for the baselines of the corresponding software. E.g., a document associated with **jazz**-managed source code having the label CUPI80_11.20_LR for the shipped version of 11i on the IA platform could be used to mark the documents baselined with that LR software (baselines from a non-**jazz** system might have a different baseline namespace).
3. Each baseline of a document, as identified by the baseline identifier, must be retrievable using an Odin URL. Some of these URL's may be the "version extended" URL that Odin uses for older versions.
4. A change control process must be followed for documents that have been baselined. This process must be able to prevent changes to particular baselines of a document, even by the author of that document, without appropriate permissions or approvals (how these permissions are granted is currently unclear, however).
5. Just as software baselines are removed over time, the document baselines must be able to be removed over time.
6. A mechanism must exist for validating that documents defined in the *SCM Plan for Documents* for a project (and therein identified as requiring baselining) are in fact baselined.
7. Odin must provide a means for traceability from the document to the relevant source code. For example, a SourceDirectory attribute for a document could contain the name of the top-most directory in the HP-UX source tree of the code affected by the document (i.e., /ux/core/kern/conf).
8. The overall Odin changes and SCM process done to implement these requirements must not prevent the anyone from being able to diffmark 2 versions of a document (where "version" here means any version that gets distributed to reviewers, users, etc), if the authoring tool (e.g., Word or FrameMaker) supports it. And if the implementation makes such diffmarking much easier, this would be helpful.

4.1.1 Notes on Possible Implementation (June 22)

Paul and I spoke a little about implementation in arriving at the re-phrased requirements above; this helped us conceptualize what we think is needed for CMM (from the SCM KPA point of view) and what was

doable in DocuShare (from the OCAB point of view). Some of the basic ideas of that possible implementation are noted here:

- a. The simple-minded “*n copies of the file*” versioning provided in DocuShare cannot alone handle the needs described in the requirements. It must be compensated for by an OCAB/Odin policy and some additional Odin-specific user interface mechanisms to support the requirements. Without this, we do not think we can make document baselining work easily or cleanly and users would hate it. DocuShare is not ClearCase.
- b. Baselined documents may end up being stored in a folder of their own, in which all baselines of that document are available: ones for the current releases IC’s as well as previous releases. The folder might be named the same as the title of the document, and the contents of the folder might have names that reflect the baseline labels. Such a *baselined document folder* should also have its own Odin-specific icon so that the concept of a baselined document is visually reinforced in Odin.
- c. Such a folder might have a “current document” that is the Odin document for the current release, with a relatively high number of Odin versions allowed (and this might be the version that by default most URL’s to that “document”—an *element* in ClearCase terms—point to). The IC-type baselines for that document might be DocuShare URL objects pointing to the older DocuShare versions of the document, whereas the LR-type baselines, for previous releases, might be separate Odin document objects, created from particular versions of the main document, and having only 1 Odin version.
- d. The baselining policy and mechanism would probably have to be implemented with user interface changes to make it consistent and easy. Aspects of this might include:
 - A new baselined document could be created using a new Odin-specific menu item similar to Add File, perhaps called Add New Baselined Document. This operation would create the sub-folder for the document and setup the right names, permissions, etc. A similar operation could be added for changing an existing document to be a baselined document.
 - An operation to actually baseline a document, and apply a label or such, might need to be restricted to authorized users, in a similar fashion to how **jazz** works with integration.
 - A version marked with a baseline label is no longer owned by the author, to prevent disallowed changes. It could be owned by some administrator-type person (the *document manager* for the release, for example).
 - The Title and Description and other relevant attributes and keywords for baselined versions would conform to a naming and format standard to make all the baselined document folders and their contained documents and baselined versions consistent to users and easy to browse, and for searches to be done and results easily browsed.
- e. We will need to carefully write definitions of a couple terms for this effort. Two that have come to mind already are for the notions of:
 - the *spawning* of a baselined version of a document from the “current” version (which would be done with a UI button or menu choice)
 - the *publishing* of a baselined document

4.1.2 Issues on Possible Implementation (June 22)

Paul and I spoke of a couple of issues we should remember at implementation time:

- The need to understand the various URL formats that DocuShare uses to access various parts of a document (older versions, various representations, attribute information, etc). One issue that came up during our work on this was that changing the filename of the file used to save and then update a doc in Odin changed the URL used for various parts of Odin, and this may affect aspects of any implementation.

- If we were to use URL objects to label baselines by pointing them to a version-extended URL for a previous version of a document, how is that URL and its target handled by the searching mechanism?
- The control mechanisms, process, and people may have parallels to the current UXSCM/**jazz** system but should be much lighter weight.
- Paul thinks that sub-typing of ODIN document types, with attendant variations in available attributes, could possibly help us in this domain.