



Document Identification



IMproved QUality SOlutions

MANAGEMENT SUMMARY

White paper that explains some aspects of document identification structures



TABLE OF CONTENTS

1	PURPOSE.....	3
2	DOCUMENT IDENTIFICATION INTRODUCTION	3
3	GENERAL STATEMENTS	4
4	USAGE OF MULTIPLE CODES IN A DOCUMENT-ID	5
5	ABOUT THE SEQUENCE NUMBER.....	5
6	WHAT ARE THE BEST PRACTICES?.....	6
7	SO WHAT DOCUMENT ID STRUCTURES I USE?	7
8	HOW TO HANDLE OBSOLETE DOC-IDS.....	7
9	CONFIGURATION MANAGEMENT SYSTEM CAVEATS	7
10	SAMPLE CODE STRUCTURES	8
10.1	DOCUMENT CODES FOR A QUALITY MANAGEMENT SYSTEM	8
10.2	DOCUMENT CODES FOR (DEVELOPMENT) PROJECT DOCUMENTS.....	9
10.3	DISCIPLINE CODES.....	9
10.4	SOFTWARE ENGINEERING CLUSTERS	10
10.5	INSTANCE CODES	10
10.6	GENERAL DOCUMENT TYPES	10

1 Purpose

This document provides some do's and don'ts on how to create a document identification structure.

2 Document Identification Introduction

There are a lot of opinions and points of view on document identification. Some have recognized the need and defined common accepted standards for document identification generally for a specific purpose. Examples are: ISBN, Library of Congress Cataloging-in-Publication Data, Number on your passport, driver license number, etc.

Most of them are not a solution for identifying a document in an (development) organisation. And as we all say that every organisation is different, so is the document identification they use.



3 General Statements

A document-id can consist of a combination of codes with operators.

Generally this is:

- A fixed prefix,
 - one or more selectable code values
 - separated by one or more different types of separators and
 - a sequence number
- that makes the entire doc-id unique.

Examples are: ImQuSo-WP-2008-001, IQS-CustomerX-002, IOS-Proc-ReV-01

Here are some general remarks based on an above like document identification

- Make your doc-id as short as possible.
Long document-ids take up a lot of place in lists, header or footer and are hard to remember.
- I would use the doc-id as the filename of the document in the storage archive.
The doc-id can handle strange characters in but some Operating Systems it can't handle strange characters (if doc-id is the filename)
- Once a doc-id is issued it should never change.
If you don't use it or it is a bad id, make it obsolete.
For note on obsolete doc-id see: § 8 How to handle obsolete Doc-ids.
- Do not put the document status, document version number or document version date in the doc-id.
If you store these documents in a Configuration Management System this can lead to non optimal use. See § 9 Configuration Management System Caveats.
This makes only sense for private storage during a long edit period.
- Do not allow a 'free format' postfix for the document id to have a more descriptive doc-id.
This clutters the identification and causes possible CM storage problems.
- Remarks on the use of codes in the doc-id like e.g. Department-code, Doc-type-code, etc.
 - Do not link codes to a physical storage path in the archive
 - You can't place the document in another location
 - If you change the doc-id of the file you have to move the file also to a new location.
 - Make sure that a code also covers all your future needs
If you run out of meaningful codes and you have to fall back to 'strange abbreviation' the better part of the code will be meaningless for (novice) users.



IMproved QUality SOlutions

- Organisational (code) names can/will change often
Use department numbers instead of department code names.
- Make sure that if your organisation structure/hierarchy changes your doc id structure stays the same.
- A year field in the doc id structure (just before the sequence number).
 - It gives an indication how obsolete the document information is.
 - It makes archiving (in year folders) easier (for a secretary)
- Make sure your sequence number has enough digits to cover your future needs!
- How are you planning to issue doc-ids?
 - Via manual action from a central issue point?
 - Via a tool that generates doc-ids?

This choice has a large impact on the option you can implement!

4 Usage of multiple codes in a document-id

Defining multiple codes for easy document recognition and easy document search/browse facility is not a good idea.

You should use a doc-id for its elementary purpose that is unique identification and not misuse it for other reasons.

You can better invest in a tool for document-id issuing that permits you to define additional information tags to a document-id without adding them to the doc-id itself.

If the tool is equipped with a good search engine that can search on all of these kind of 'meta tags' attached to a document id you are better off.

5 About the sequence number

A sequence number is generally used to make the doc-id unique in case you have multiple documents with the same doc-id code part.

The unique part of the doc-id can be interpreted differently.

Consider the following options:

- A. The sequence number makes the entire doc-id unique
{Only if the part before the sequence number is the same the sequence number is incremented.}
e.g. CompA-Dep1-Memo-001, CompA-Dep1-Memo-002,
CompA-Dep1-Letter-001, CompA-Dep2-Memo-001
CompB-Dep1-Memo-001, CompA-Dep1-Memo-003
- B. The sequence number makes the doc-id unique by making a small part of the doc-id unique.



IMproved QUality SOlutions

- Solution 1: {Independent of the Comp value}
e.g. CompA-memo-Dep1-2008-001, CompA-memo-Dep1-2008-002,
CompA-memo-Dep2-2008-001, CompA-memo-Dep3-2008-001,
CompA-letter-Dep1-2008-001, CompB-memo-Dep3-2008-002
- Solution 2 {only dependant of the year value}
e.g. CompA-memo-Dep1-2008-001, CompA-memo-Dep1-2008-002,
CompA-memo-Dep2-2008-003, CompA-memo-Dep3-2008-004,
CompA-letter-Dep1-2008-005
- C. The sequence number it self is unique.
e.g. CompA-Dep1-00001, CompA-Dep2-00002, CompA-Dep1-00003

Solution C can lead to very large sequence numbers and does not have my preference.

Solution A & B1 will cause problems with manual document-id issuing.

For solution B2 & C any code used (except year in B2) will have no identification purpose. Consider omitting this code. See: § 4 Usage of multiple codes in a document-id

6 What are the best practices?

My best practice has evolved over the years, but know I think that there is no single doc-id structure that fits all purposes, even within the same department/organisation.

Because if you want to make the doc-id structure as short and efficient as possible you probably will have one or more doc id structures:

- Official company documents available in a company wide archive
- Document of a organisational quality system
- Documents for use inside the organisation
- Document dedicated to (development) projects

Combining some/all of these into a single document-id structure is possible.

But you need to consider:

- Who issues document ids
- Where and in what archive should it be stored
- Who is responsible for the completeness of the archive
- Who has to store documents in the archive
- Is there an electronic and a paper archive?
(are responsibilities different for both?)

If you have identified multiple roles and responsibilities here then consider having multiple doc id structures.



7 So what document ID structures I use?

For general purpose documents:

Start with a fixed prefix for company

Secondly you can add a global document code

Put a year field in the code.

Define a sequence number with enough digits.

(This is reset to 1 at the start of a new year.)

This is how the doc-id for this whitepaper is defined:

ImOuSo-WP-2008-001

- ImQuSo Company abbreviation
- WP Category identification: White Paper
- YYYY year code of document-id (not necessary publish date.)
- 001 Three digit sequence number

For customers I use:

- ImQuSo Company abbreviation
- PLE Three letter customer abbreviation
- BLL Three letter document purpose abbreviation (Bill)
- YYYY year code of document ID
- Three digit sequence number

8 How to handle obsolete Doc-ids

Do not simply remove obsolete doc-ids from the registration system of archive.

The best way to proceed is:

Put a descriptive text in the document like:

This doc-id declared obsolete by Mr X at <date>.

Please look for the original content in doc-id ????

Put the document with the dummy content in the archive.

Create a new (correct) doc-id if needed.

9 Configuration Management System Caveats

Filename issues:

In most CM systems a file is stored under its Operating System name. If a new version of the same file(name) (same file path!) is entered in the CM system a new instance (private system copy) is created for that file(name). This way the CM system constructs a history of changed files or file changes for that filename. (Internally it keeps track of the different instances and the optionally changes between two instances.)



IMproved QUality SOlutions

If you put the document version and/or document status in the filename the OS filename of the new file is different from the previous one.

For the simple CM system the same rules apply and thus this new file is handled as a new file (even if the file path is the same) and not as a new instance of an existing file. That is what you really want!

Note: Some CM systems have symbolic filenames and can handle different OS filenames that are linked to that symbolic CM filename.

Keyword replacement:

Most CM systems support keyword replacement for e.g. source code files.

In that case the version number, last editor, last edit date, etc. is replaced in the source file by the CM system with the correct/actual values.

You don't want the CM system do keyword replacement on binary file like e.g. MS word documents. In that case that the CM system discovers a keyword pattern somewhere in a binary file and it replaces it with the 'intended' text most likely the word file will be corrupt after such 'accidental' keyword replacement.

Most CM systems that do keyword replacement have an option to disable this for binary files types like e.g. .zip, .exe, .com, .doc, .ppt, .xls, etc.

10 Sample Code Structures

Below you will find a couple of (incomplete) lists of different views on documents that can be used as part of a document Identification structure

10.1 Document codes for a Quality Management System

Code	Quality System Document type
QP	Procedure (Quality Policy)
QC	Process description (Approach)
QI	Instruction (Mandatory Way of Working "How")
QG	Guideline (Advised Way of Working "How")
QT	Forms, Templates and Tools
QR	Reports
QM	Manual (Describes how to use a tool, etc.) ("Why", "How")
QD	Other documents
QJ	QJ Will not be used to avoid confusion with QI



10.2 Document codes for (Development) Project Documents

Code	Development document type
CMP	Configuration Management Plan
CRS	Customer Requirements Specification
DD	Detailed Design
Eval	Evaluation
FeaS	Feasibility Study
FRS	Function Requirement Specification
GTP	Global Test Plan
IRS	Industrial Requirements Specification
PP	Project Plan
QAP	Quality Assurance Plan
TLD	Top Level Design
TLD-CA	Top Level Design: Conceptual Architecture
TLD-MIA	Top Level Design: Module Interconnection Architecture
TLD-HSP	Top Level Design: HW SW Partitioning
TLD-HSI	Top Level Design: HW SW interface
TLD-SSI	Top Level Design: SW SW interface
TLD-EA	Top Level Design: Execution Architecture
TLD-CodA	Top Level Design: Code Architecture
TP	Test Plan
TR	Test Report
TRN	Training
TS	Test Specification

10.3 Discipline codes

Code	Discipline
ACO	Acoustical
ELE	Electrical
HYD	Hydraulical
MEC	Mechanical
ORG	Organisation (everything outside the project)
PJM	Project Management
PJO	Project Office
QD	Quality Department
SW	Software
HW	Hardware (All not software)
SYS	System (Software & Hardware in complete product)

Note: not all of the disciplines or projects need to be present in your organisation.



10.4 Software Engineering clusters

Code	Engineering cluster
Arc	Architecture
Dev	Development
CM	Configuration Management
Man	Manuals etc.
PM	Project Management
QA	Quality Assurance
Tst	Testing

10.5 Instance Codes

Code	Instance description
BP	Best/good Practice
Exa	Example
Tpl	(empty) Template
<PP>	Project Code
<OO>	Organisation/department code

10.6 General document types

Code	General Document type
Ann	Announcement
Ass	Assignment
Bin	Binary files
Ckl	Checklist
Dsg	Design
Des	Description
Frm	Form
Inv	Invitation
Mai	Mail
Man	Manual
Mem	Memo
Pln	Plan
Pst	Poster
Rep	Report
Spc	Specification
Src	Source files
Stu	Study



IMproved QUality SOlutions

*

ImQuSo Improved Quality Solutions.

Po Box 169

5540 AD REUSEL, The Netherlands

All mentioned names are used for identification purposes only and are trademarks or registered trademarks of their respective companies.

© Copyright 2008-2009 ImQuSo.

ALL RIGHTS RESERVED.

Doc. ImQuSo-WP-2008-013 Rev. 1.1 2009-04-15