配置管理规范

|  |  |  |
| --- | --- | --- |
| 编 制 | 审 核 | 批 准 |
| 陆丽娜 |  |  |

深圳市共济科技有限公司

（版权所有）

**说 明**

名 称： 配置管理规范

所 有 者： 深圳市共济科技有限公司开发部 陆丽娜

访问路径：

**版 本 历 史**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 日 期 | 版本 | 人 员 | 说 明 | 审核人 | 审核日期 | 备 注 |
| 2011-01-04 | V1.4.1 | 陆丽娜 | 开发库配置项管理的若干规定--配置规范.doc |  |  |  |
| 2013-3-23 | V1.4.2 | 陆丽娜 | 加入SVN管理相关内容 |  |  |  |
| 2013-4-3 | V1.4.3 | 陆丽娜 | 加入SVN管理基线、分支和标签相关内容 |  |  |  |
| 2013-5-23 | V1.4.4 | 王晶 | 增加配置管理表格模板，修改部分截图及内容 |  |  |  |

目 录

[第1章 简介 7](#_Toc352775957)

[1.1 目的 7](#_Toc352775958)

[1.1.1 编写目的 7](#_Toc352775959)

[1.1.2 适用范围 7](#_Toc352775960)

[1.1.3 预期读者 7](#_Toc352775961)

[1.1.4 术语和缩略语 7](#_Toc352775962)

[1.1.5 配置管理的范围 7](#_Toc352775963)

[1.2 定义 8](#_Toc352775964)

[1.2.1 基本配置项 8](#_Toc352775965)

[1.2.2 产品类主要配置项 8](#_Toc352775966)

[第2章 目录结构 9](#_Toc352775967)

[2.1 SVN目录结构 9](#_Toc352775968)

[2.1.1 驱动库目录结构 9](#_Toc352775969)

[2.1.2 项目库目录结构 9](#_Toc352775970)

[2.2 VSS目录结构 9](#_Toc352775971)

[第3章 入库规范 10](#_Toc352775972)

[3.1 遵守规则 10](#_Toc352775973)

[3.2 SVN驱动库入库规范 12](#_Toc352775974)

[3.2.1 svn/basic/trunk/drivers 12](#_Toc352775975)

[3.2.1.1 Build 12](#_Toc352775976)

[3.2.1.2 Config 12](#_Toc352775977)

[3.2.1.3 Share 2](#_Toc352775978)

[3.2.1.4 Source 2](#_Toc352775979)

[3.2.2 示例1：svn/basic/trunk/drivers/config 2](#_Toc352775980)

[3.2.2.1 目录首字母大写 2](#_Toc352775981)

[3.2.2.2 Database 2](#_Toc352775982)

[3.2.2.3 ReaseNote 2](#_Toc352775983)

[3.2.2.4 功能需求清单 2](#_Toc352775984)

[3.2.2.5 数据库维护更新记录 2](#_Toc352775985)

[3.2.2.6 抓包数据 2](#_Toc352775986)

[3.2.3 示例2：svn/basic/trunk/drivers/source 2](#_Toc352775987)

[3.2.3.1 Code只放代码 2](#_Toc352775988)

[3.2.3.2 安装调试手册 3](#_Toc352775989)

[3.2.3.3 程序维护更新记录 3](#_Toc352775990)

[3.2.3.4 ReaseNote 3](#_Toc352775991)

[3.2.3.5 功能需求清单 4](#_Toc352775992)

[3.2.3.6 安装调试手册 4](#_Toc352775993)

[3.2.3.7 文档命名 4](#_Toc352775994)

[3.2.3.8 程序维护更新记录 4](#_Toc352775995)

[3.2.4 svn/drivers 4](#_Toc352775996)

[3.2.4.1 厂家调试软件目录 4](#_Toc352775997)

[3.2.4.2 资料目录 4](#_Toc352775998)

[3.2.4.3 功能需求清单目录 5](#_Toc352775999)

[3.2.4.4 数据库维护更新记录 5](#_Toc352776000)

[3.2.4.5 协议目录 5](#_Toc352776001)

[3.2.4.6 抓包数据 5](#_Toc352776002)

[3.3 SVN项目库入库规范 5](#_Toc352776003)

[3.3.1 /svn/basic/projects 5](#_Toc352776004)

[3.3.1.1 Build 5](#_Toc352776005)

[3.3.2 项目档案：存放需要跟踪的项目信息 6](#_Toc352776006)

[3.3.2.1 需求分析 6](#_Toc352776007)

[3.3.2.2 项目计划 6](#_Toc352776008)

[3.3.2.3 设计 6](#_Toc352776009)

[3.3.2.4 项目跟踪与维护 6](#_Toc352776010)

[3.3.2.5 测试 6](#_Toc352776011)

[3.3.2.6 评审与总结 6](#_Toc352776012)

[3.3.2.7 成果发布 6](#_Toc352776013)

[3.4 VSS库入库规则 7](#_Toc352776014)

[3.4.1 驱动库的结构 7](#_Toc352776015)

[3.4.2 基本构成 10](#_Toc352776016)

[3.4.3 遵守规则 12](#_Toc352776017)

[3.4.4 文件夹命名和位置的确定 13](#_Toc352776018)

[第4章 SVN版本信息规范 13](#_Toc352776019)

[4.1 目的 13](#_Toc352776020)

[4.2 范围 13](#_Toc352776021)

[4.3 版本号约定 13](#_Toc352776022)

[4.4 版本信息内容要求 14](#_Toc352776023)

[4.4.1 C++ 14](#_Toc352776024)

[4.4.1.1 编辑方法 14](#_Toc352776025)

[4.4.1.2 示例 15](#_Toc352776026)

[4.4.2 C# 17](#_Toc352776027)

[4.4.2.1 编辑方法 17](#_Toc352776028)

[4.4.2.2 示例 19](#_Toc352776029)

[4.4.3 Java 20](#_Toc352776030)

[第5章 软件发布 21](#_Toc352776031)

[5.1.1 产品发布 21](#_Toc352776032)

[5.1.1.1 产品发布流程图 21](#_Toc352776033)

[5.1.1.2 产品发布步骤 21](#_Toc352776034)

[5.1.1.3 建立产品基线(分支)步骤 22](#_Toc352776035)

[5.1.2 任务发布 22](#_Toc352776036)

[5.1.2.1 任务发布流程图 22](#_Toc352776037)

[5.1.2.2 任务发布步骤 22](#_Toc352776038)

[5.1.2.3 建立任务类基线(分支)步骤 23](#_Toc352776039)

[第6章 VSS版本信息规范 23](#_Toc352776040)

[6.1.1 编译目标程序及产品项目命名规则 23](#_Toc352776041)

[6.1.2 程序版本号 24](#_Toc352776042)

[6.1.2.1 格式 24](#_Toc352776043)

[6.1.2.2 标记位置 25](#_Toc352776044)

[6.1.2.3 v5版本的特殊说明 26](#_Toc352776045)

[6.1.2.4 示例 26](#_Toc352776046)

[6.2 工程文件命名 27](#_Toc352776047)

[6.2.1 工程文件(源代码)命名规则 27](#_Toc352776048)

[6.2.2 文档命名及版本号 28](#_Toc352776049)

[6.2.3 文档名称 28](#_Toc352776050)

[6.2.3.1 格式 28](#_Toc352776051)

[6.2.3.2 标记位置 28](#_Toc352776052)

[6.2.3.3 示例 28](#_Toc352776053)

[6.2.4 文档版本号 29](#_Toc352776054)

[6.2.4.1 格式 29](#_Toc352776055)

[6.2.4.2 标记位置 29](#_Toc352776056)

[6.3 设备命名 30](#_Toc352776057)

[6.3.1 设备类型 30](#_Toc352776058)

[6.3.2 设备名称 31](#_Toc352776059)

[6.3.3 配置设备数据模板库的指南 31](#_Toc352776060)

[6.3.4 示例 31](#_Toc352776061)

[6.4 协议 32](#_Toc352776062)

[6.4.1 协议编号格式 32](#_Toc352776063)

[6.4.2 协议编号标识 32](#_Toc352776064)

[6.4.3 示例 32](#_Toc352776065)

[6.5 产品库/项目库介绍 32](#_Toc352776066)

[6.5.1 产品库/项目库结构 32](#_Toc352776067)

[6.5.2 项目文件夹的命名及其位置 33](#_Toc352776068)

[第7章 文档管理 34](#_Toc352776069)

[7.1.1 项目开发资料的类别 34](#_Toc352776070)

[7.1.2 文档的作用和类别 35](#_Toc352776071)

[7.1.3 文档类别及生成阶段 35](#_Toc352776072)

[7.1.4 各阶段产品列表 36](#_Toc352776073)

[7.1.5 文档版本及命名规则 36](#_Toc352776074)

[7.1.5.1 文档版本格式 36](#_Toc352776075)

[7.1.5.2 文档命名规则 36](#_Toc352776076)

[7.2 配置管理表格模板 38](#_Toc352776077)

[7.2.1 配置管理计划 38](#_Toc352776078)

[7.2.2 变更请求单 38](#_Toc352776079)

[7.2.3 变更状态报告 38](#_Toc352776080)

[7.2.4 基线审计报告 38](#_Toc352776081)

[7.2.5 基线发布报告 38](#_Toc352776082)

[7.2.6 分支审计报告 38](#_Toc352776083)

[7.2.7 分支发布报告 38](#_Toc352776084)

[7.2.8 版本发布报告 38](#_Toc352776085)

[7.2.9 配置库管理报告 38](#_Toc352776086)

[7.2.10 配置审计报告 38](#_Toc352776087)

[7.2.11 各类文档模板 38](#_Toc352776088)

[第8章 配置管理的相关概念 39](#_Toc352776089)

[8.1 配置管理基本概念 39](#_Toc352776090)

[8.1.1 配置管理定义 39](#_Toc352776091)

[8.2 版本控制的概念 39](#_Toc352776092)

[8.2.1 版本控制的对象 39](#_Toc352776093)

[8.2.2 版本控制的目的 39](#_Toc352776094)

[8.2.3 版本控制的实际 39](#_Toc352776095)

[8.3 变更控制 39](#_Toc352776096)

[8.4 配置库 40](#_Toc352776097)

[8.4.1 建立配置管理系统（SVN配置库、VSS配置库） 40](#_Toc352776098)

[8.4.2 配置库使用流程图 40](#_Toc352776099)

[8.4.3 配置库使用原则 41](#_Toc352776100)

[8.4.4 配置管理活动及方针 42](#_Toc352776101)

[8.5 基线 42](#_Toc352776102)

[8.5.1 常用的基线 42](#_Toc352776103)

[8.5.1.1 分配基线（Allocated Baseline） 42](#_Toc352776104)

[8.5.1.2 功能基线（Functional Baseline） 43](#_Toc352776105)

[8.5.1.3 产品基线（Product Baseline） 43](#_Toc352776106)

[8.5.2 基线建立指南 43](#_Toc352776107)

[8.5.3 基线命名规定 44](#_Toc352776108)

[8.6 分支 45](#_Toc352776109)

[8.6.1 分支建立指南 45](#_Toc352776110)

[8.6.2 分支命名规定 45](#_Toc352776111)

[8.6.3 分支/标签建立实例 46](#_Toc352776112)

[8.7 词汇表 47](#_Toc352776113)

[8.7.1 CM（Configuration management） 47](#_Toc352776114)

[8.7.2 CCB（Change control board） 47](#_Toc352776115)

[8.7.3 CI（Configuration item） 47](#_Toc352776116)

[8.7.4 CR（Change Request） 47](#_Toc352776117)

[8.7.5 PCA（Physical Configuration Audit） 47](#_Toc352776118)

[8.7.6 FCA（Functional Configuration Audit） 47](#_Toc352776119)

[8.7.7 基线（Baseline） 47](#_Toc352776120)

[8.7.8 配置管理库（Configuration Management Library） 47](#_Toc352776121)

[8.7.9 版本（Version） 47](#_Toc352776122)

# 简介

## 目的

### 编写目的

规范VSS和SVN库中配置项的标识和存放。从而提供配置项的存取和检索功能，有利于配置项的更改控制，保证配置项的完整性和可跟踪性。

### 适用范围

适用于所有软件产品和软件项目的配置项管理。配置管理可采用各种工具及手工办法，本文件以VSS和SVN配置管理工具为例，规定公司的配置管理办法，使用其他工具时也可对应本文件的要求参照执行。对于本制度中规定的内容，若与以前发布的制度有冲突的，均以本制度为准。

### 预期读者

开发部全体成员。

### 术语和缩略语

本文件采用NP601100《配置管理》程序使用的术语和缩略语的定义。

### 配置管理的范围

软件配置可包括以下几方面：项目文档，源代码，执行程序，相关设备及资料等。

1）项目文档主要指：立项建议报告、项目启动计划、可行性分析报告、开发计划、需求分析报告、软件功能规格说明书、系统设计报告、数据库表结构、技术报告、总结报告、验收报告以及上述文档的评审记录。

2）相关设备主要指项目开发和运行环境（包括硬件和软件），以及项目开发和测试过程中使用的专用仪器设备，如读卡机、扫描仪等。

3）相关资料主要指客户提供的行业法规，标准及其调研期间提供的业务单据，往来会议记要，传真，电子邮件，重要的电话记录等。

## 定义

### 基本配置项

软件开发过程的输出信息，可分为计算机程序、文档和数据定义及配置文件三个主要类别：其中

1. **计算机程序**指由公司开发部组织自主开发，它包括但不限于下表中所列文件；

|  |  |
| --- | --- |
| **File Groups** | **File Types** |
| VC | \*.bmp;\*.c;\*.cpp;\*.cur;\*.cxx;\*.def;\*.ds?;\*.h;\*.hpj;\*.hpp;\*.hxx;\*.ico;\*.inl;\*.mak;\*.rc;\*.rc2;\*.rgs |
| VB/VBA | \*.asp;\*.bas;\*.cls;\*.ct?;\*.dca;\*.dep;\*.dob;\*.dox;\*.ds?;\*.fr?;\*.log;\*.oca;\*.pag;\*.pgx;\*.res;\*.swt;\*.vb?,\*.xls |
| 目标文件 | \*.dll;\*.exe;\*.ocx；\*.xls |

1. **文档**包括但不限于以下类别的文档，需求说明类；开发设计类；使用帮助类；项目管理类；往来沟通类；第三方提供的协议、软件（含SDK）。
2. **数据**定义及配置文件包括但不限于下表中所列文件；

|  |  |  |
| --- | --- | --- |
| **File Groups** | | **File Types** |
| 数据定义 | Access | \*.mdb |
| SQL | \*.mdf;\*.ldf; |
| XML | \*.xml |
| 配置文件 | | \*.ini |
| 日志文件 | | \*.log |

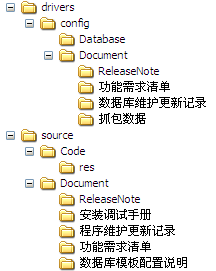
### 产品类主要配置项

|  |  |
| --- | --- |
| 类型 | 主要配置项 |
| 技术合同 | 合同；SOW[[1]](#footnote-2) |
| 计划 | 项目计划；质量保证计划；配置管理计划； |
| 需求 | 需求规格说明书；用户DEMO |
| 设计 | 总体设计说明书；数据库设计；详细设计说明书；设计术语及规范 |
| 编程 | 源程序；编码规则 |
| 测试 | 测试计划；测试用例；测试报告 |
| 提交 | 运行产品；验收报告；用户手册 |

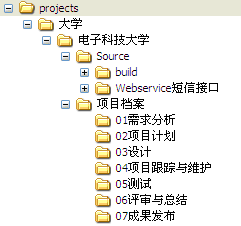
# 目录结构

## SVN目录结构

### 驱动库目录结构

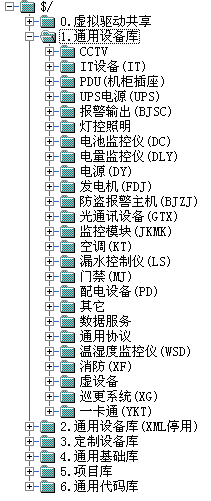


### 项目库目录结构



## VSS目录结构

VSS-Dyanmic2



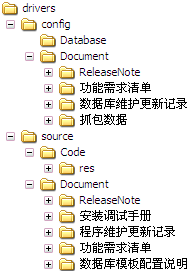
# 入库规范

## 遵守规则

VSS和SVN的目录由配置组建立，根据任务开启相应路径权限

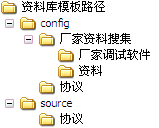
### 驱动目录建立

驱动目录按照库上最新结构，由配置组建立(svn/basic/trunk/drivers/share/documents/SCM/drivers)



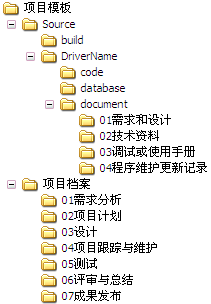
### 驱动资料目录建立

驱动资料目录按照库上最新结构，由配置组建立(svn/drivers/share/资料库模板路径)



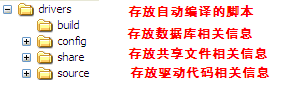
### 项目目录建立

项目目录按照库上最新结构，由配置组建立(svn/basic/projects/项目模板)



## SVN驱动库入库规范

### svn/basic/trunk/drivers

库主要有4个：。

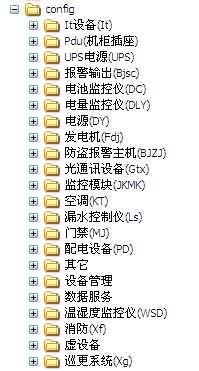
#### Build

存放编译的脚本和ReleaseNote ,具体目录如下



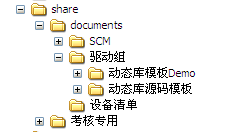
#### Config

专门存放设备数据库配置相关信息，具体目录如下：



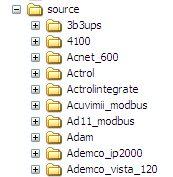
#### Share

专门驱动开发共享的模板、资料、规范等相关信息，具体目录如下：



#### Source

专门驱动代码、维护记录等相关信息，具体目录如下：

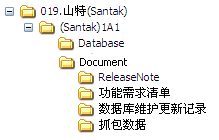


### 示例1：svn/basic/trunk/drivers/config

UPS电源(UPS)/019.山特(Santak)/(Santak)1A1目录如图：

#### 目录首字母大写

数据库、文档的名称使用“类型\_品牌\_型号\_类型”的规范



#### Database

仅放置设备数据库

#### ReaseNote

放置用户阅读的修正要点

#### 功能需求清单

#### 数据库维护更新记录

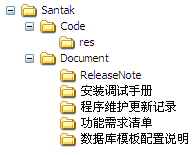
记录必须使用服务器上最新模板，按规范样例里的要求修改后放置。



#### 抓包数据

### 示例2：svn/basic/trunk/drivers/source

/Santak目录如图



首字母大写，其它小写

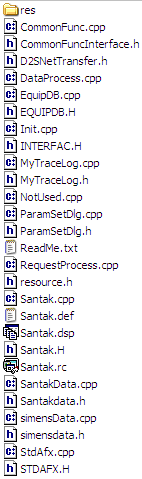
#### Code只放代码

代码是要删除dsw,plg,opt,ncb,aps,clw这些文件后的代码。

Res只放代码资源

若有特殊目录的。Code下必须按磁盘上的目录建立，保证下载后编译直接能过。

Code：



Res:



#### 安装调试手册



#### 程序维护更新记录



#### ReaseNote

放置用户阅读的修正要点

#### 功能需求清单

#### 安装调试手册

放置代码功能使用手册。

现场通过文档可以直接调试成功

#### 文档命名

必须按设备类型\_文档类型.文件类型

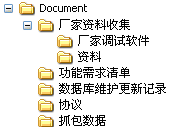


#### 程序维护更新记录

需要从服务器上获取最新的版本文档，按样例里的要求填写默认值，新开发程序不用写程序维护更新记录，但文档必须有。

### svn/drivers

主要存放驱动相关的文档资料



#### 厂家调试软件目录

存放厂家调试软件，下载后双击该exe即可运行调试。

#### 资料目录

共享协议，同时并添加厂家其它资料（与开发该设备数据库无关的资料）

#### 功能需求清单目录

#### 数据库维护更新记录

使用最新模板编写的维护记录

#### 协议目录

存放与开发该设备数据库有关的接口资料（协议如果是pdf的。请压缩一份和该pdf一起入库）

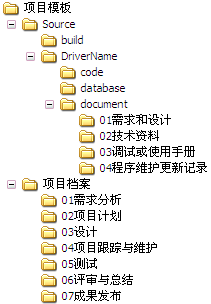
协议资料：放置该代码依赖哪个数据接口开发的。

#### 抓包数据

需要放置调试通过，没有问题的抓包数据，利于模拟测试和调试发包。

## SVN项目库入库规范

### 目录结构



### Source

Source目录包括Build目录和Driver Name 目录，其中每个项目Build目录只有一个，Driver Name目录根据具体程序或模块名进行新建，可有多个。

#### Build

存放编译的脚本和ReleaseNote ,具体目录如下



project.properties：主要由具体开发人员填写，包括基本属性、项目属性、发布属性和构建目标等信息。

ReleaseNote.txt：主要由具体开发人员填写，项目名称、项目描述、程序名称、程序版本、功能及缺陷变更等信息。

#### Driver Name

Driver Name目录根据具体程序或模块名进行新建，例如新开发程序Modbus.dll，则可将目录改名为Modbus

注意：本目录应尽量使用英文名称

Code：用于存放程序源码

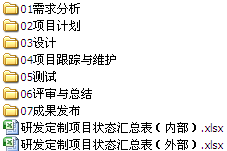
Database：用于存放数据库，若没有可留空

Document：用于存放程序文档，包括01需求和设计、02技术资料、03调试或使用手册和04程序维护更新记录。

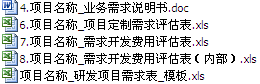
### 项目档案

项目档案：用于存放需要跟踪的项目信息，主要为项目经理使用，具体使用规范请参见：





#### 需求分析



#### 项目计划



#### 设计

#### 项目跟踪与维护



#### 测试

#### 评审与总结



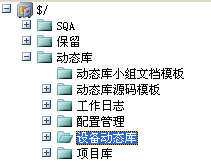
#### 成果发布

## VSS库入库规则

### 驱动库的结构

驱动库vss-dynamic和vss-dynamic1已经于2011-12-01迁移至vss- dynamic2。所有新开发的设备驱动均在vss-dynamic2库上进行。

* 1. Vss-dynamic,鸟瞰图如下

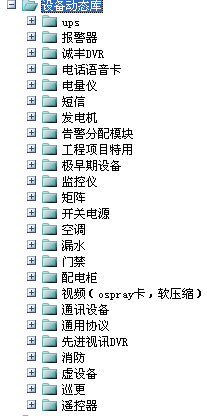


目录如上图，开发人员的主要工作路径是**$\动态库\**，子目录的用途分别是（未提到的目录表示目前已经不使用了）：

**动态库小组文档模板**：存放驱动组常用的文档或模板供新手阅读学习。

**动态库源码模板**：驱动组开发设备驱动的源代码模板。

**设备动态库**：按**类别\厂家\型号**存放已经开发好的设备驱动



**项目库:**按**项目名称\模块**存放为专门的项目定制设备驱动或程序模块。

* 1. VSS-Dynamic1结构图如下：

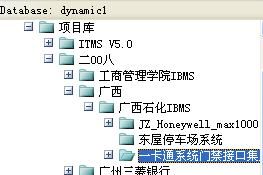


目录说明：

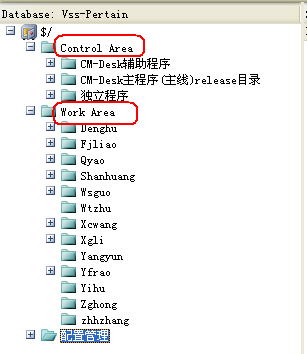
独立程序：存放用于驱动的独立程序，比如设备调试工具等。

通用设备库：按**类别\厂家\型号**存放已经开发好的设备驱动

项目库：按**项目时间\区域\项目名称\模块**存放为专门的项目定制设备驱动或程序模块。如下图：



1. 辅助程序库(Vss-Pertain)的开发结构



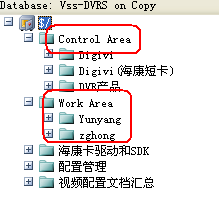
Vss-pertain库存放CM-Desk平台的辅助程序模块及独立于平台运行的小程序或工具

分为控制区(Control Area)和工作区(Work Area)。开发人员的日常开发不写入控制区，有需求要修改代码时开发人员向配置管理提出开权限请求，配置管理员给开发人员开放指定目录读权限。开发人员获取控制区最新的版本然后在工作区的个人工作目录中开展工作，在下个版本发布之前由配置管理统一开放控制区目录写权限给相应开发人员将工作区代码合入到控制区。

控制区(Control Area)：按**程序类型\模块名称\[不同版本]**存放。

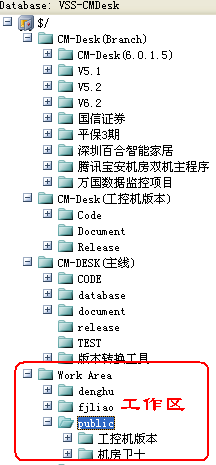
工作区(Work Area)：按**开发人员名字\模块名称\[不同版本]**存放。开发人员对自己的目录默认有读写权限。

1. 视频库(Vss-DVRS)的开发结构



Vss-dvrs库存放CM-Desk平台视频模块的所有程序源代码。其基本结构与vss-pertain原理一致，工作模式也是相同的。

1. 主程序库(Vss-CMDesk)的开发结构



Vss-CMDesk库存放平台主程序的源代码，也分为控制区和工作区，这里工作区之外的同级目录都列为控制区，包括三个主目录：CM-Desk(Branch)、CM-Desk (工控机版本)、CM-Desk(主线)。

**CM-Desk(Branch)**：主程序源代码的分支，专为项目定制。按**主版本号\项目名称**存放

**CM-Desk (工控机版本)**：存放最新发布版本地站\管理站的主程序源代码。属于旧的产品平台，仅用于老项目的维护不再输出给新项目。

**CM-Desk(主线)**：存放最新发布版机房卫士\管家的主程序源代码。目前所用的产品平台的主程序。

工作区跟vss-pertain，vss-dvrs一样按人员名字分。因为主程序通常由多人修改，所以在工作区建了一个目录public供多人共同协作开发。发布的时候指定一人负责编译并更新到控制区。

### 基本构成

一个开发项目，在CQ库中须有一份开发任务单与之对应。

在开发设计时，一个开发项目可由1个或多个模块构成，一个模块由1个或多个子模块构成。

在开发库里，一个开发项目对应一个项目文件夹，与此项目相关的计算机程序、文档和数据定义及配置文件等必须按其类别分类放入项目文件夹；一个开发项目由一个或多个元素构成，原则上一个元素对应一个目标文件及其代码；若此项目由多个模块构成，一个模块对应一个项目子文件夹，一个子模块是一个元素。

以Actrol门禁为例：

|  |  |  |
| --- | --- | --- |
| **开发设计结构示意图** | **开发库实际结构** | **开发库结构模型示意图** |
| 开发项目  模块1  　　子模块  模块2  　　子模块  模块3  　　子模块 |  | 项目文件夹  项目子文件夹1  　　　　元素  项目子文件夹2  　　元素  项目子文件夹3  　　元素 |

**元素的一般结构如下图所示：**



**文件在元素各级目录中存放的规则如下表所述**：

|  |  |  |
| --- | --- | --- |
| **一级目录** | **二级目录示例** | **文件存放规则** |
| CODE |  | 用于存放代码，可根据代码工程的结构建立多级子目录。 |
| DATABASE |  | 用于存放数据定义文件，可根据需要建立多级子目录。  文件存放的一般规则：   1. 属于公司平台型软件产品指定的数据定义文件存放在DATABASE文件夹下； 2. 根据用户需求定义的其它类型数据定义文件，在DATABASE文件夹下建立子文件夹存放； |
| DOCUMENT |  | 用于存放文档，可根据需要建立多级子目录。  文件存放的一般规则：   1. 开发人员在开发过程中编制的文档放在DOCUMENT文件夹下；此类型的文档须符合相应的文档模板规定的格式和内容要求。 2. 往来沟通文档放在DOCUMENT文件夹下；以txt或doc文件格式存放，文件内容需记录交流的主题、参与人、时间、交流方式和详细内容。 3. **第三方软件子文件夹**。此文件夹在必要时建立。对于第三方软件，开发人员必须在项目结项时提供一份文档，文档用于说明第三方软件中的每个文件在开发过程中的作用，文档名称为《第三方软件清单》，文档内容由项目经理或开发组长审核。 4. **协议子文件夹**。对于针对设备的开发项目，必须建立此子文件夹。该文件夹由配置管理员建立并以协议的编号命名，开发人员在将协议的电子档入库时，请在文件名中加入协议编号信息；若此协议没有电子文档，开发人员可在配置管理组借阅此编号的协议。 5. 项目管理类文档放在项目级的Document文件夹中。 |
| RELEASE |  | 用于存放编译后的目标文件及其必须的配置文件，可根据需要建立多级子目录。  文件存放的一般规则：   1. 编译后的目标文件、必须的配置文件直接放在RELEASE文件夹中； 2. 此项目在发布时，需同时发布的程序、数据定义文件和文档等文件，用共享方式放至文件夹； 3. 此项目提交测试时，需提供给测试人员的程序、数据定义文件和文档等文件，用共享方式放至文件夹；   针对设备动态库文件的特别说明：   1. 若新开发的设备使用已有的动态库文件（即，此元素中不包括CODE部分），则此文件夹可以不存放编译后的可执行文件，但必须在RELEASE文件名称上用括号标记出使用的动态库文件名称。 2. 2006年起CM-Desk系统的动态库文件采用新接口方式。若工程中需要对老接口方式的文件进行维护时，它在功能实现上又需要与新接口版本的文件保持一致时，只需保存老接口方式的动态库文件，而不必保留其代码。采用新接口方式的动态库文件入在Release文件夹下，采用老接口方式的动态库文件在Release文件夹下建立名为“老接口方式”的文件夹存放；新、老接口的动态库文件在版本号上有区别，详见3.2.3节。 |
| TEST |  | 临时性文件夹，仅用于在测试过程中存放《测试需求单》。开发人员在提出测试需求时建立，对应的CQ任务单关闭时由配置管理员删除。 |

元素的结构可根据实际情况，适当裁减。

### 遵守规则

为了方便VSS库的管理及操作的方便，请开发人员务必遵守以下规则

1、开发一个新模块，包括驱动（不管有没有动态库）都必须写软件需求说明书（如：XXXX\_RS\_软件需求说明书，XXXX指模块名，下同）和软件使用说明书(如：XXXX\_SUD\_软件使用说明书)。

2、修改任何程序模块或设备驱动必须填写程序维护更新记录(如：XXXX\_PMD\_程序维护更新记录)。

3、文档，代码，数据库按类存放后，将Document中的”XXXX\_SUD\_软件使用说明书.doc”、”XXXX\_PMD\_程序维护更新记录.doc”链接到Release目录中。将Database中的用到的设备模板数据库文件链接到Release目录中。

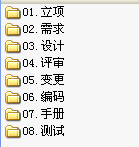
操作方法：可以直接用**拖拽**方式。

**注：上面1、2、3中提到的文档名称及简写请参考本文中文档格式说明。**

1. 各类文档都有模板。参考文档的模板统一放置在：

[\\192.168.1.9\部门管理\2.开发管理\过期模板\2011年软件开发模版](file:///\\192.168.1.9\部门管理\2.开发管理\过期模板\2011年软件开发模版)

需求文档必须有包含发布程序配置信息。



5、各类标准文档及协议的命名必须按照本文所规定。

### 文件夹命名和位置的确定

在开发库中，项目文件夹由配置管理员建立，其路径在对应的CQ库任务单的VSS工程中应有详细说明。项目文件夹名称由配置管理员和开发人员协商确定。

项目子文件夹由开发人员根据开发设计的需要建立，其名称由开发人员确定。

项目文件夹和项目子文件夹名称须用简洁的文字标识出其特性。

# SVN版本信息规范

## 目的

文件版本信息包括公司名称、产品版本、版权声明、描述等信息，为了在版本标识方面更加专业和严谨，需统一产品发布后的文件版本信息风格，特制订本规范。

本文档主要针对公司产品/项目中几种常用的编程语言和开发工具来对版本信息的编辑进行要求。

## 范围

以下人员需知悉本规范：

* 产品经理
* 项目经理
* 软件工程师（C++、C#、Java）
* 配置工程师

## 版本号约定



**V＜X.Y.Z.M＞**：为版本标记，不可省略。X、Y、Z由产品经理控制，M由开发内部控制。

* V为版本号起始标记。
* **X为主版本号（以下简称X）**，从1开始编号，最大值没有限制；当此程序有重大修改时，根据产品经理意见**X**增1；当**X**递增时，**Y**、**Z、M**归0。
* **Y为次主版本号（以下简称Y）**，从0开始编号，最大值没有限制；此程序新增功能时，根据产品经理意见**Y**增1；当**Y**递增时，**Z、M**归0。
* **Z为从版本号（以下简称Z）**，从0开始编号，最大值没有限制；程序功能增加或修改时根据产品经理意见**Z**增1，M归0。
* **M为编译版本号（以下简称M）**，从1开始编号，最大值没有限制。**M**专用于记录开发人员编译次数。开发人员自己做单元测试时M保持不变。

※※ 强烈建议：开发人员要养成良好习惯，每编译一次M加1。在“特殊内部版本说明”中注明：编译人、编译时间。

**说明：**

**如在项目中有支线版本，则主线版本保持不变，可增加支线版本（项目中版本号，共两位，可参照程序版本号中的“Z.M”）信息，在“特殊版本标识符”进行描述，下文将有具体示例。**

## 版本信息内容要求

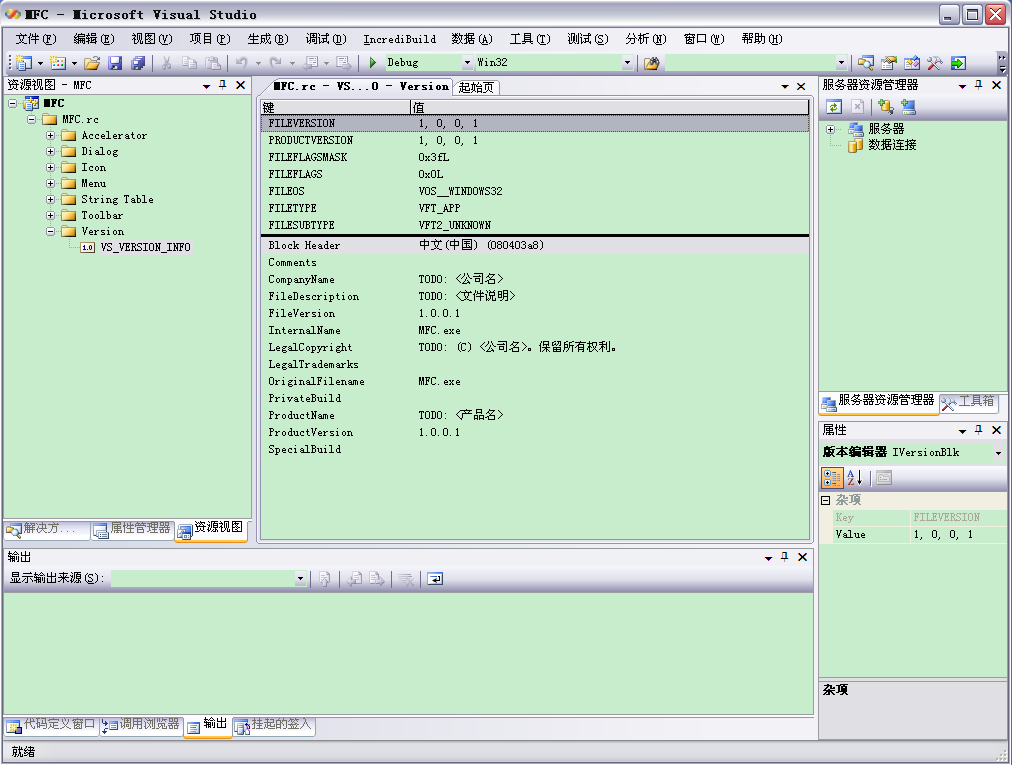
### C++

#### 编辑方法

目前，不同的产品/项目在使用C++作为编程语言时，选取的开发工具有：VC6.0、VS2008、VS2010，这三种开发环境下，都可使用Microsoft Visual C++的MFC应用程序资源编辑器来对版本信息进行编辑，其中包含了版本信息编辑器（Version Information Editor）。

**方法**：

在资源视图（ResourceView）中双击版本信息资源（VS\_VERSION\_INFO），如下图：

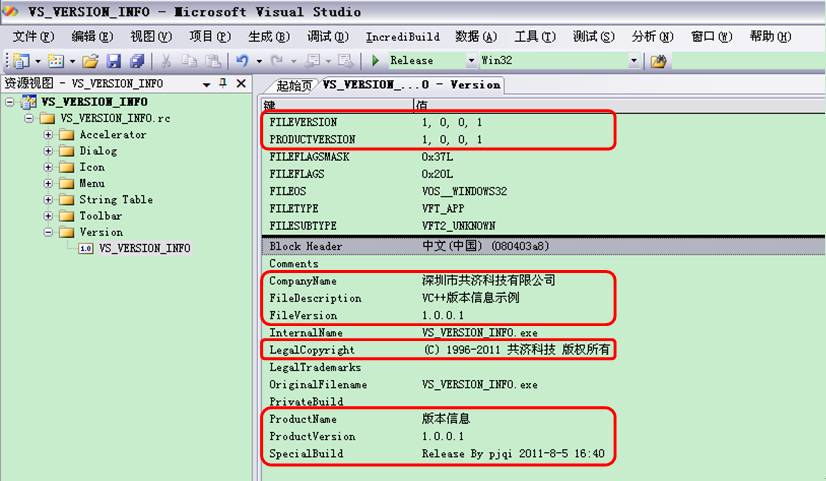


在对象列表中可看到很多项，填写要求如下表所示（必填项用蓝色强调）：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本信息编辑项** | **对应编译后的**  **文件版本信息** | **填写说明** | **是否**  **必填** | **示例** |
| FILEVERSION |  | X.Y.Z.M | 是 | 6.2.0.9 |
| PRODUCTVERSION |  | X.Y.Z.M | 是 | 6.2.0.9 |
| Comments | 备注 |  | 否 |  |
| CompanyName | 公司 | 填写公司中文全称  （若是英文版，则需使用英文全称） | 是 | 深圳市共济科技有限公司  Xbrother Technology Co.,LTD |
| FileDescription | 描述 | 说明文件的用途 | 是 | 虚拟设备驱动 |
| FileVersion | 文件版本 | X.Y.Z.M\_特殊版本标识符[[2]](#footnote-3) | 是 | 6.2.0.9\_北京工商局1.4 |
| InternalName | 内部名称 |  | 否 |  |
| LegalCopyright | 版权 | 1996-2012 共济科技 版权所有  Copyright 1996-2012 Xbrother Technology. All Rights Reserved. | 是 | 1996-2012 共济科技 版权所有  Copyright 1996-2012 Xbrother Technology. All Rights Reserved. |
| LegalTrademarks | 合法商标 |  | 否 |  |
| OriginalFilename | 源文件名 |  | 否 | VirtualDevice.DLL |
| PrivateBuild | 个人用内部版本说明 |  | 否 |  |
| ProductName | 产品名称 | 程序名称 | 是 | 虚拟设备动态库 |
| ProductVersion | 产品版本 | X.Y.Z.M\_特殊版本标识符（与FileVersion一致） | 是 | 6.2.0.9\_北京工商局1.4 |
| SpecialBuild | 特殊内部版本说明 | Release By XXX YYYY-MM-DD HH:MM | 是 | Release By xgli 2011-6-5 15:20 |

#### 示例

在对版本信息进行编辑时，下图中用红色框线标记的为必填项。



编译之后生成的文件，当点击右键“属性”，在其版本标签页可以查阅，如下图所示：

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

### C#

#### 编辑方法

目前，公司的产品/项目在使用C#作为编程语言时，选取的开发工具主要是VS2010，可使用“程序集信息”来对版本信息进行编辑。

“程序集信息”对话框用于指定 .NET Framework 全局程序集特性的值，这些特性存储在项目自动创建的AssemblyInfo文件中。若要访问此对话框，在“解决方案资源管理器”中选择项目节点，然后在右键菜单上单击“属性”。 当“项目设计器”出现时，单击“应用程序”选项卡。 在“应用程序”页上，单击“程序集信息”按钮。如下图所示：



填写要求如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **程序集信息编辑项** | **对应编译后的**  **文件版本信息** | **填写说明** | **是否必填** | **示例** |
| 标题 | 描述 | 说明文件的用途 | 是 | CM\_Desk报表系统 |
| 说明 | 备注 | 特殊版本说明[[3]](#footnote-4) | 否 | 北京工商局1.4 2011-XX-XX |
| 公司 | 公司 | 填写公司中文全称  （若是英文版，则需使用英文全称） | 是 | 深圳市共济科技有限公司  （Xbrother Technology Co.,LTD） |
| 产品 | 产品名称 | 程序名称 | 是 | 虚拟设备动态库 |
| 版权 | 版权 | © 1996-YYYY 共济科技 版权所有  （若是英文版，则需使用英文方式） | 是 | © 1996-2011 共济科技 版权所有  （© 1996-2011 Xbrother Technology. All rights reserved.） |
| 商标 | 合法商标 |  | 否 |  |
| 程序集版本 | 产品版本 | X.Y.Z.M | 是 | 5.1.0.9 |
| 文件版本 | 文件版本 | X.Y.Z.M | 是 | 5.1.0.9 |
| GUID |  |  | 否 |  |
| 非特定语言 |  |  | 否 |  |

#### 示例

编辑“程序集信息”时，下图中，红色标记部分为必填项。



|  |  |
| --- | --- |
|  |  |
|  |  |

### Java

目前，公司的产品/项目在使用Java作为编程语言时，使用的开发工具主要是Eclipse或MyEclipse。

在工程生成Jar/War文件包时，会自动生成一个META-INF目录，其中包含MANIFEST.MF文件，在此文件中对版本信息进行编辑说明。

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **填写内容** | **是否必填** | **示例** |
| Name | 产品/程序名称 | 是 | 监控管理平台 |
| Version | 产品/程序版本 | 是 | 2.1.0.6 |
| Description | 描述本产品/程序用途 | 是 | 提供综合监控管理服务 |
| CompanyName | 填写公司中文全称  （若是英文版，则需使用英文全称） | 是 | 深圳市共济科技有限公司  （Xbrother Technology Co.,LTD） |
| Copyright | © 1996-YYYY 共济科技 版权所有  （若是英文版，则需使用英文方式） | 是 | © 1996-2011 共济科技 版权所有  （© 1996-2011 Xbrother Technology. All rights reserved.） |
| Comments | 其他需要说明的事项 | 否 |  |

格式如下：

Name：监控管理平台

Version：2.1.0.6

Description：提供综合监控管理服务

CompanyName：深圳市共济科技有限公司

Copyright：© 1996-2011 共济科技 版权所有

Comments：

# 软件发布

### 产品发布

1. 测试版本发布：不符合正式发布条件，需要继续发布直至形成正式版本
2. 正式版本发布：符合正式发布条件，产品经理负全责，通知配置正式发布到全公司
3. 其他基线(分支)发布：需要项目经理，开发经理，组长确认并通知配置

建立相应的计划基线、需求基线、设计基线、编码基线、测试基线等

#### 产品发布流程图



#### 产品发布步骤

1. 发布前准备：项目经理负责将产品《发布申请表》和《ReleaseNote》内容填写完整，并检查代码是否测试通过
2. 产品发布申请：《发布申请表》和《ReleaseNote》向配置组申请发布
3. 配置发布：根据《发布申请表》进行产品发布，输出物：安装包
4. 测试验证：根据配置发布的成果和《测试需求单》，输出物：测试报告
5. 开发经理：负全责，确定是否向外发布
6. 产品发布：
7. 测试版本发布：
8. 正式版本发布：

#### 建立产品基线(分支)步骤

1. 先由项目经理提出建立产品基线(分支)申请，由总工程师/开发经理确认受影响的相关人员（如：项目组成员、测试人员、配置管理员、质量保证工程师、相关业务部门）；
2. 举行建立产品(分支)基线评审，由总工程师/开发经理主持
3. 评审通过之后，配置进行基线(分支)的建立。

### 任务发布

1. 程序发布：只需要单个程序的发布
2. 补丁包发布： 整合程序生成补丁包发布
3. 其他基线(分支)发布：需要项目经理，开发经理，组长确认并通知配置

建立相应的计划基线、需求基线、设计基线、编码基线、测试基线等

#### 任务发布流程图



#### 任务发布步骤

1. 任务单发布申请：开发人员《发布申请表》和《ReleaseNote》向配置组申请发布
2. 配置发布：根据《发布申请表》进行发布，输出物：程序、补丁包或安装包
3. 测试验证：根据配置发布的成果和《测试需求单》，输出物：测试报告
4. 项目经理：确保配置发布，确保测试通过，确定是否向外发布

必须确保为配置发布的输出物

任务单的输出物不完全都通过测试组结论报告，需要项目经理自行判断是否发布，

#### 建立任务类基线(分支)步骤

1. 先由项目经理提出建立任务类基线(分支)申请，由总工程师/开发经理确认受影响的相关人员（如：项目组成员、测试人员、配置管理员、质量保证工程师、相关业务部门）；
2. 举行建立任务类基线(分支)评审，由总工程师/开发经理主持
3. 评审通过之后，配置进行基线(分支)的建立。

# VSS版本信息规范

### 编译目标程序及产品项目命名规则

* 程序名称中的有效字符仅限于字母、数字和下划线。
* 程序名中用到的文字必须拼写正确，并且符合语法和缩略写习惯。
* 一般情况下，程序的命名遵守望名知意的原则，即程序名能反应出程序实现的功能或特性。
* 设备动态库文件的命名还应遵守以下几项特殊规则：
  1. 程序名称中需包含厂家/品牌、型号等信息，如AcTrol门禁的设备动态库名称为：Actrol.dll；
  2. 设备的功能模块名称格式如“＜品牌名称＞＿＜模块功能标识符＞”所示，模块功能标识符遵守望名知意的原则，如AcTrol门禁历史查询功能模块的程序名称为：ActrolHistoryQuery.exe；
  3. 对于通用协议和使用设备管理方式的功能模块其命名遵守望名知意的原则，如Modbus协议的动态库名称为：Modbus.dll，南京海关短信服务的动态库名称为：SmsNJ.dll；
  4. 对于应用以上规则命名后，若仍会与已有程序的文件产生重名的情况，则在文件名称中附加区别标识符，以示区别。区别标识符须能反应两者间的区别。
  5. 对于VSS库中已经存在的动态库,但在个别项目中需要重写的时候,需要将该动态库重新命名,命名的原则是加上项目名称的简写。
  6. 对于VSS库中已经存在的动态库，但在个别子系统（未与CM-Desk集成）有相同功能模块，命名的原则是加上子系统的简称。
  7. 配置文件（扩展名为ini）名称与对应的程序名称一致。
  8. 日志文件（扩展名为log）名称与对应的程序名称一致。
* 产品型项目命名规则：

1. 发布后的交付物名称是：产品名称\_v版本号(发布日期)，产品名称有中文名用中文名，没有中文名用英文名。例如:机房总管客户端\_v7.0.2.4(090824)
2. 产品配置库的基线的命名规则是：Release\_产品名称\_v版本号(发布日期)

产品名称用英文简称，例如：Release\_SMP7\_v7.0.2.4(090824)

### 程序版本号

#### 格式

＜程序名称＞＿V＜X.Y.Z.M＞＿［特殊版本标识符］

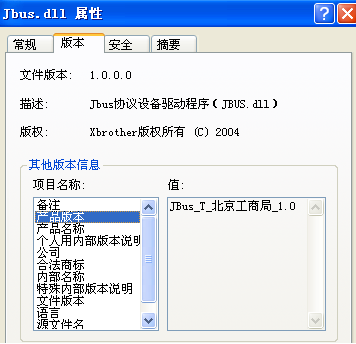
**程序名称**：规则如3.1节所述，不可省略。

**V＜X.Y.Z.M＞**：为版本标记，不可省略。

* V为版本号起始标记。
* **X为主版本号（以下简称X）**，从1开始编号，最大值没有限制；当此程序有重大修改时，根据开发意见**X**增1；当**X**递增时，**Y**、**Z**归0。
  + 驱动类程序：旧平台的以5开始编号，新平台的以6开始编号，旧平台中已经存在的（以5开始的）也用于新平台，不用版本更新。
  + 视频模块：以7开始编号
  + 门禁模块：以6开始编号
  + 告警分配模块（暂未成独立模块）
* **Y为次主版本号（以下简称Y）**，从0开始编号，最大值没有限制；此程序新增功能时，根据开发意见**Y**增1；当**Y**递增时，**Z**归0。
* **Z为从版本号（以下简称Z）**，从0开始编号，最大值没有限制；程序功能增加或修改时根据开发意见**Z**增1，M归0。
* **X，Y，Z和版本号升级由开发决定。**
* **M为调试版本号（以下简称M）**，从1开始编号，最大值没有限制。**M**专用于记录平台组开发人员（平台软件）提交给测试组测试（如第一次提交V6.2.0.1，下次提交V6.2.0.2）,和驱动组开发人员（设备驱动）发给工程方面调试的版本（如第一次发送V6.0.1.1，下次发送V6.0.1.2）。开发人员自己做单元测试时M保持不变。**※※ 强烈建议：开发人员要养成良好习惯，每提交/发送一个新的版本M加1。**
* 在调试版本M要保持不变的情况下，比如多人编译同一版本，需要提供内部编译的版本号，即：X.Y.Z.M build nnnn。用于区别同一版本的不同编译次数。

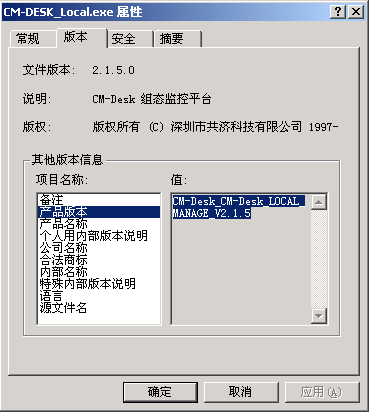
**特殊版本标识符**：必要时使用，可省略。

* 当需对某一程序做特殊功能开发，其在CM-Desk系统中的程序文件名必须与普通版本的一样时，版本号上增加特殊版本标识符，特殊版本标识符须能反应两者间的区别。如下图用于北京工商局的特殊版本。



#### 标记位置

在Microsoft Visual C++中程序携带ProductVersion和FileVersiong两个版本信息。当点击该程序的右键属性时，在其版本标签页可以查阅，如下图所示。



* 产品版本（ProductVersion），规则如本文所述。
* 文件版本（FileVersion），与产品版本的数值部分一致。

#### v5版本的特殊说明

CM-Desk v5监控平台中动态库文件版本的特殊说明

1、CM-Desk本地站/管理站监控平台于2006起改用接口方式，所以在该平台下使用的动态库文件，其版本号的主版本号为5，次主版本号和从版本号的演进规则遵守3.2.1节所述规则。

2、CM-Desk机房卫士/机房管家监控平台于2008起在原本地站的基础上升级，所以在该平台下新开发的的动态库文件，其版本号的主版本号为6，次主版本号和从版本号的演进规则遵守3.2.1节所述规则。在该平台下维护老的动态库，不须硬性升级到6,在原版本号上继续维护。

3、若工程中需要对老接口方式的文件进行维护时，它在功能实现上又需要与新接口版本的文件保持一致时，新、老接口方式的动态库文件的文件名相同，但版本号不同，新接口的主版本号用‘5’，老接口用‘3’；次主版本号和从版本号两者一致。

4、为使CM-Desk监控平台中的程序同时能应用于特殊版本输出，开发人员在编译程序时，对于程序的Version，仅需填文件版本、产品版本和特殊内部版本说明。“特殊内部版本说明”只需注明程序编译的时间和编译者名，示例如下图。



#### 示例

以下将分三种情况描述程序版本演进的过程。产品版本和文件版本必须相同。下面以产品版本做说明。

* **情况一：新开发驱动程序，只需经工程验证**

首次编译，产品版本：×××\_V6.0.0.1

开发人员做单元测试，M不用递增，版本演进状态为，

　　　　　产品版本：×××\_V6.0.0.1

单元测试通过，将V6.0.0.1版本发给工程方面试用，出现BUG，对其进行修改，版本演进为，

　　　　　产品版本：×××\_V6.0.0.2

将V6.0.0.2版本发给工程方面试用，出现新BUG，对其进行修改，版本演进为，

　　　　　产品版本：×××\_V6.0.0.3

继续修改，直到V6.0.0.5版本，工程方面验证通过并通知驱动组项目经理（黎冲），开发人员确定此版本可做发布版本使用。

　　　　　产品版本：×××\_V6.0.0.5

对CQ里的任务单，进行Resolve操作；同时《需求说明书》里程序的发布版本号定为V6.0.0.5。

以上流程对于修改BUG同样适用。

* **情况二：软件平台产品提交测试**

首次编译，产品版本：×××\_V6.2.0.1

开发人员做单元测试，M不须递增，版本状态为，

　　　　　产品版本：×××\_ V6.2.0.1

单元测试通过，提交配置组编译发布V6.2.0.1。

开发人员验证V6.2.0.1。验证过程发现问题修改版本号保持不变。验证通过后配置提交给测试组测试。

产品版本：×××\_V6.2.0.1

测试组测试，测试出来的问题往CQ Bug库提交Bugs。开发人员修改BUG重新编译M加1。

　　　　　产品版本：×××\_V6.2.0.2

V6.2.0.2再提交测试，还有BUG需要修改。

　　　　　产品版本：×××\_ V6.2.0.3

继续测试和修改，直到V6.2.0.5版本，测试验收通过，测试人员确定此版本可做发布版本使用，则发布版本为，

产品版本：×××\_ V6.2.0.5

## 工程文件命名

### 工程文件(源代码)命名规则

在实际的开发工作中，新开发的模块大部分是从已经存在的模块拷贝出来再修改的，这样做的好处是方便，节省时间。但是开发人员在拷贝的时候造成新工程名与已存在的那个工程名相同，每个模块应该有自己的工程名，所以规定开发人员在开发新模块时一定要将工程改名，新名称遵循以下规则。

* 原则上与程序名称相同。如adam.dll的工程名应为adam。
* 如果有输出程序名称相同，但应用范围不同的，如中文版与英文版，普通版与串口版等类似情况，应该在原有工程名后面加下划线”\_”和标志符号。
* 其他的规范同程序的命名规范。

### 文档命名及版本号

此章节仅适用于开发人员在开发过程中编制的文档.。

### 文档名称

#### 格式

＜项目名称＞＿＜文档类型简称＞＿＜文档名称＞

**项目名称**：以该文档拟在VSS库中存放处的项目文件夹和子文件夹名称名称命名。

**文档名称**：开发过程中编制的文档如下表所示。

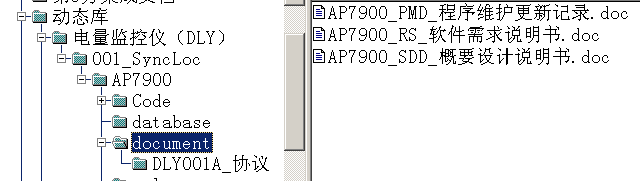
|  |  |
| --- | --- |
| **文档类型简称** | **文档名称** |
| PMD | 程序维护更新记录 |
| DMD | 数据库模板维护更新记录 |
| RS | 软件需求说明书 |
| SDD | 概要设计说明书 |
| SUD | 软件使用说明书 |
| DDD | 详细设计说明书 |
| ESD | 设备使用说明 |
| PD | 项目开发计划 |
| TCD | 动态库模板配置说明 |
| CD | 交流纪要 |

#### 标记位置

该文档电子版的文件名称

#### 示例

以SyncLos厂家的型号为AP7900的信号采集器为例，其文档软件需求说明书的名称为：AP7900\_RS\_软件需求说明书，其中，项目名称为AP7900，也即该文档存放处的项目文件夹的名称；文档类型简称为RS；文档名称为软件需求说明书。其在VSS库中的位置如下图所示：



### 文档版本号

#### 格式

**VX.Y.Z**

* + - V为版本号起始标记。
    - X为主版本编号，从1开始编号；当此文档有重大修改时，根据评审组意见主版本号增1；当主版本号递增时，次版本号版归0。
    - Y为次版本号，从0开始编号，此文档每修改一次，次版本号增1。
    - Z由文档作者自行定义。
    - 当文档处于草稿状态时，主版本号为0，次版本号由作者自行确定，以便于在交流时能区分不同的版本；首次正式发布的版本号为1.0.0，此后的版本演进遵循如上所述规则。

#### 标记位置

在文档的版本历史处标记。

## 设备命名

此章节仅适用于CM-Desk组态监控平台的CM-Desk数据库。

### 设备类型

CM-Desk组态监控平台上监控的设备是由CM-Desk数据库集中管理.。CM-Desk.mdb文件的Equip\_OPT表记载了所有的设备，其中“设备类型”字段，每一个字段值用于唯一标识一个设备，此字段值的命名规则如下：

格式：＜设备分类＞＿＜厂家名/品牌名＞＿［型号/系列］＿［区别信息］

**设备分类**

　　CM－Desk里的设备分类如下表所示：

|  |  |  |
| --- | --- | --- |
| **设备分类** | **中文释义** | **备注** |
| JZ | 矩阵 |  |
| IT | IT设备 |  |
| UPS | UPS电源 |  |
| BJSC | 报警输出 |  |
| DC | 电池监控仪 |  |
| DLY | 电量监控仪 |  |
| DY | 电源 |  |
| FDJ | 发电机 |  |
| BJZJ | 防盗报警主机 |  |
| GTX | 光通讯设备 |  |
| JKMK | 监控模块 |  |
| KT | 空调 |  |
| LS | 漏水控制器 |  |
| MJ | 门禁 |  |
| PD | 配电设备 |  |
| WSD | 温湿度监控仪 |  |
| XF | 消防 |  |
| LSJ | 冷水机 |  |
| ―― | DVR | 一般情况下，此类型的设备在确定设备类型字段值时，设备分类信息可以省略。 |
| ―― | 其它 |
| ―― | 数据服务 | 一般情况下，此类型的设备在确定设备类型字段值时，可不遵循上述格式，而以特殊方式命名。 |
| ―― | 项目专用 |

**厂家名/品牌名**

　　　　厂家名/品牌名可使用中文和英文，中/英的拼写要求准备无误，同时尽可能使用英文。此部分不可省略。

**型号/系列**

　　　　此部分可省略。若工程方面能提供确切的型号或产品系列信息，则填写。

**区别信息**

按上述命名规则命名后，若仍会与已有设备的设备类型字段值产生重名的情况，则附加区别信息，以示区别。区别标识符须能反应两者间的区别。

### 设备名称

在工程中，设备名称用作设备在设备树上的标识，此字段值的命名规则如下：

＜设备名称＞

**设备名称**

　　　　设备的中文名称，此部分不可省略。

### 配置设备数据模板库的指南



### 示例

以SyncLos厂家的型号为AP7900的信号采集器为例，

设备类型字段值：[DLY\_SyncLoc\_AP7900]；

设备分类

型号

厂家名

设备名称字段值：信号采集器。

## 协议

协议有电子文档和纸质文档两种状态，需开发设备的协议至少有电子文档或纸质文档中的任一种，电子文档由开发人员放入对应开发项目的项目文件夹，纸质文档由配置管理组集中保管。协议编号是识别协议唯一标记。

### 协议编号格式

＜设备分类＞＿＜厂家/品牌编号＞＿＜序列号＞

**设备分类**

　　设备分类，见5.1节的设备分类表。

**厂家名/品牌编号**

　　用三位数字表示，从001开发编号，按1增长。一个编号代表一个厂家。

**序列号**

　　用大写英文字母表示，从A开始，按字母顺序增长。一个设备一个序列号。

　　注：此规定实行后新开发设备的协议按此规则编号，目前已有的协议仍按原编号。

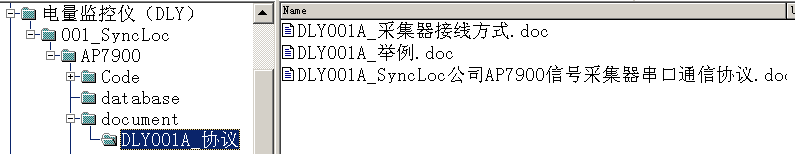
### 协议编号标识

电子文档协议，在入开发库时，在文件名前加注协议编号。

纸质文档协议，在入协议库时，在档案盒上标记。

### 示例

以SyncLos厂家的型号为AP7900的信号采集器为例，协议编号：DLY001A，其中DLY是设备分类；001厂家编号；A是序列号，由厂家提供的协议在入库时也在文件名上加注了协议编号，其在VSS库中的存放状态如下图所示：

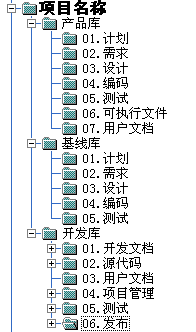


## 产品库/项目库介绍

### 产品库/项目库结构

产品库/项目库目前使用Vss-Product和Vss-Project (产品库/项目库)和Vss-TS(文档库)。所有新开发的产品均在Vss-Product和Vss-Project库上进行。

Vss-Product和Vss-Project,鸟瞰图如下



目录如上图，开发人员的主要工作路径是**$/项目名称/开发库/02.源代码/XXX模块**，子目录的用途分别是：

在项目开发过程中，配置库可分开发库、基线库和测试区三个区域，其各自存放的内容及存取的规定为：

（1）开发库：——配置管理每日构建

存放开发过程中需要保留的各种信息，供项目组成员使用，项目组每日入库。

开发人员的日常开发写入开发库，有需求要修改代码时开发人员向配置管理提出开权限请求，经项目经理同意后，配置管理员给开发人员开放指定目录读写权限。

开发库存放项目组所遵循的过程标准、参考资料、所有未经批准的配置项、已经批准但未纳入基线的配置项，此区域中的配置项由项目经理负责和控制，项目总结结束后删除。

（2）基线库：——存储成果级别的代码和评审完成的文档。并作为发布的目录，进行软件版本发布编译的路径。

在软件开发的某个阶段工作结束时，将工作产品存入或将有关的信息存入。

开发库中达到成果级别就入基线库，但必须通知配置管理，由配置管理统一代码检验后才可入基线库。由配置管理员联合编译，发布版本。

基线库存放基线。此区域的配置项由项目经理或CCB评审批准后，由配置管理员从开发库更新而来，此区属配置管理员所有。

（3）产品库：——基线库中形成了一个对外发布的版本

在开发的软件产品完成系统测试之后，作为最终产品存入库内，等待交付用户或现场安装。

### 项目文件夹的命名及其位置

在产品库/项目库中，项目文件夹由配置管理员建立。项目文件夹名称由配置管理员和开发人员协商确定。

项目源代码子文件夹由开发人员根据开发设计的需要建立，其名称由开发人员确定。

项目文件夹和项目代码子文件夹名称须用简洁的文字标识出其特性。

目录结构如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **一级目录** | **二级目录** | **三级目录** | **四级目录** | **五级目录** | **软件资料成品** |
| **项目名称** | 基线库 | 计划 |  |  | 可行性研究报告，项目开发计划 |
|  |  | 需求 |  |  | 项目开发计划，软件需求说明，数据要求说明 |
|  |  | 设计 |  |  | 概要设计说明，详细设计说明 |
|  |  | 编码 |  |  |  |
|  |  | 测试 |  |  | 测试计划，测试分析报告 |
|  | 产品库/项目库 | 计划 |  |  | 可行性研究报告，项目开发计划 |
|  |  | 需求 |  |  | 项目开发计划，软件需求说明，数据要求说明 |
|  |  | 设计 |  |  | 概要设计说明，详细设计说明 |
|  |  | 编码 |  |  |  |
|  |  | 测试 |  |  | 测试计划，测试分析报告 |
|  |  | 可执行文件 |  |  |  |
|  |  | 用户文档 |  |  | 用户手册，操作手册，维护修改建议 |
|  | 开发库 | 开发文档 | 计划 |  | 可行性研究报告，项目开发计划 |
|  |  |  | 需求分析 |  | 项目开发计划，软件需求说明，数据要求说明 |
|  |  |  | 概要设计 |  | 概要设计说明 |
|  |  |  | 数据库设计 |  |  |
|  |  |  | 详细设计 |  | 详细设计说明 |
|  |  |  | 参考资料 |  |  |
|  |  | 源代码 |  |  |  |
|  |  | 用户文档 |  |  | 系统的安装维护使用手册等 |
|  |  | 项目管理 | 配置管理 |  | 基线发布报告，基线审计报告，配置变更状态报告，配置权限分配记录 |
|  |  |  | 质量保证 |  |  |
|  |  |  | 跟踪和监控 |  |  |
|  |  |  |  | 项目例会 |  |
|  |  |  |  | 项目周报 | 开发进度月报 |
|  |  |  | 评审和报告 |  | 项目开发总结 |
|  |  | 测试 | 系统测试 | 方案 | 测试计划 |
|  |  |  |  | 用例 |  |
|  |  |  |  | 报告 | 测试分析报告 |
|  |  | 发布 | 版本号 |  | 该目录下可以存放发布记录 |

# 文档管理

### 项目开发资料的类别

以下表格中的资料为开发资产，相应的有效资产需及时入库存储。

|  |  |  |
| --- | --- | --- |
| 分类 | 阶段 | 举例 |
| 环境类 | 软件开发环境或软件维护环境 | 编译器、操作系统、编辑器、数据库管理系统、开发工具、项目管理工具、文档编制工具 |
| 定义类 | 需求分析与定义阶段结束后得到的工作产品 | 需求规格说明书、项目开发计划、设计标准或设计准则、验收测试计划 |
| 设计类 | 设计阶段结束后得到的工作产品 | 系统设计规格说明、程序规格说明、数据库设计、编码标准、用户界面标准、测试标准、系统测试计划、用户手册 |
| 编码类 | 编码及单元测试结束后得到的工作产品 | 源代码、目标码、单元测试数据及单元测试结果 |
| 测试类 | 系统测试完成后的工作产品 | 系统测试数据、系统测试结果、操作手册、安装手册 |
| 维护类 | 进入维护阶段以后生成的工作产品 | 以上任何需要变更的软件配置项 |

### 文档的作用和类别

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 阶段文档 | 什么 | 何处 | 何时 | 谁 | 如何 | 为何 | 文档类别 |
| 可行性研究报告 | **﹡** |  |  |  |  | **﹡** | 开发文档 |
| 项目开发计划 | **﹡** |  |  |  |  |  | 开发文档/管理文档 |
| 软件需求说明 | **﹡** | **﹡** | **﹡** | **﹡** |  |  | 开发文档/用户文档 |
| 数据要求说明 | **﹡** | **﹡** |  |  |  |  | 开发文档 |
| 概要设计说明 |  |  |  |  | **﹡** |  | 开发文档 |
| 详细设计说明 |  |  |  |  | **﹡** |  | 开发文档 |
| 测试计划 |  |  | **﹡** | **﹡** | **﹡** |  | 管理文档 |
| 用户手册 |  |  |  |  | **﹡** |  | 用户文档 |
| 操作手册 |  |  |  |  | **﹡** |  | 用户文档 |
| 测试分析报告 | **﹡** |  |  |  |  |  | 管理文档 |
| 开发进度月报 | **﹡** |  | **﹡** |  |  |  | 管理文档 |
| 项目开发总结 | **﹡** |  |  |  |  |  | 管理文档 |
| 维护修改建议 | **﹡** |  |  | **﹡** |  | **﹡** | 用户文档 |

### 文档类别及生成阶段

各阶段文档生成情况如下，在项目开发完成时，应保证相应文档已入库存储。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 阶段文档 | 可行性研究与计划 | 需求分析 | 设计 | 代码编码 | 测试 | 运行与维护 |
| 可行性研究报告 |  |  |  |  |  |  |
| 项目开发计划 |  |  |  |  |  |  |
| 软件需求说明 |  |  |  |  |  |  |
| 数据要求说明 |  |  |  |  |  |  |
| 概要设计说明 |  |  |  |  |  |  |
| 详细设计说明 |  |  |  |  |  |  |
| 测试计划 |  |  |  |  |  |  |
| 用户手册 |  |  |  |  |  |  |
| 操作手册 |  |  |  |  |  |  |
| 测试分析报告 |  |  |  |  |  |  |
| 开发进度月报 |  |  |  |  |  |  |
| 项目开发总结 |  |  |  |  |  |  |
| 维护修改建议 |  |  |  |  |  |  |

### 各阶段产品列表

|  |  |  |
| --- | --- | --- |
| 序号 | 阶段 | 工作产品 |
| 1 | 需求阶段 | 软件开发计划 |
| 2 |  | 要件定义书 |
| 3 |  | 需求跟踪列表 |
| 4 | 概要设计阶段 | 系统设计书 |
| 5 |  | 基本设计书 |
| 6 | 详细设计 编码阶段 | 详细设计书样书 |
| 7 |  | 单元集成系统测试式样书 |
| 8 |  | 编码规范 |
| 9 |  | 程序代码 |
| 10 |  | 单元测试报告 |
| 11 |  | 系统α版本 |
| 12 | 测试阶段 | 集成测试报告 |
| 13 |  | 系统β版本 |
| 14 |  | 系统测试报告 |
| 15 |  | 系统release版本 |
| 16 | 验收阶段 | 用户使用手册 |
| 17 |  | 系统安装说明书 |

### 文档版本及命名规则

#### 文档版本格式

**VX.Y.Z**

* + - V为版本号起始标记。
    - X为主版本编号，从1开始编号；当此文档有重大修改时，根据评审组意见主版本号增1；当主版本号递增时，次版本号版归0。
    - Y为次版本号，从0开始编号，此文档每修改一次，次版本号增1。
    - Z由文档作者自行定义。
    - 当文档处于草稿状态时，主版本号为0，次版本号由作者自行确定，以便于在交流时能区分不同的版本；首次正式发布的版本号为1.0.0，此后的版本演进遵循如上所述规则。

#### 文档命名规则

这里只统一文档的命名，源代码部分遵循公司的“编码规范”，以下是项目文档命名规范：

**<项目名>V<发布版本号><文件种类>\_<子系统名称>[<模块名称>]**

*示例：*

|  |  |
| --- | --- |
| **文件种类** | **文档名称** |
| 需求分析说明书 | PECM V1.0.0软件需求分析说明书 |
| 软件项目计划 | PECM V1.0.0软件项目计划 |
| 总体方案 | PECM V1.0.0总体方案 |
| 软件模块详细说明书 | PECM V1.0.0软件模块详细说明书 |
| 数据库设计说明书 | PECM V1.0.0数据库设计说明书 |
| 程序配置清单 | PECM V1.0.0程序配置清单 |
| 测试计划 | PECM V1.0.0测试计划 |
| 测试报告 | PECM V1.0.0测试报告 |
| 用户手册 | PECM V1.0.0用户使用手册 |
| 操作维护手册 | PECM V1.0.0操作维护手册 |
| 项目总结 | PECM V1.0.0项目总结 |

## 配置管理表格模板

### 配置管理计划



### 变更请求单



### 变更状态报告

### 基线审计报告



### 基线发布报告



### 分支审计报告

### 分支发布报告

### 版本发布报告



### 配置库管理报告



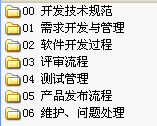
### 配置审计报告



### 各类文档模板

参考文档的模板统一放置在：\\192.168.1.9\流程和规范\工程

产品类/项目类开发基本包含以下文档：



# 配置管理的相关概念

## 配置管理基本概念

### 配置管理定义

1. 配置管理：包含版本控制、工作空间管理、并行开发控制、过程管理、权限管理、变更管理等内容。
2. 软件配置管理：是在贯穿整个软件生命周期中建立和维护项目产品的完整性
3. 目标：
4. 软件配置管理的各项工作是有计划进行的；
5. 被选择的项目产品得到识别，控制并且可以被相关人员获取；
6. 已识别出的项目产品的更改得到控制；
7. 使相关小组和个人及时了解软件基准的状态和内容。

## 版本控制的概念

### 版本控制的对象

软件开发过程中涉及的所有文件系统对象，包括文件、目录和链接。文件包括源代码、可执行文件、位图文件、需求文档、涉及说明和测试计划等。

目录的版本记录了目录的变化历史，包括新文件的建立、新的子目录的创建、已有文件或子目录的重新命名及删除等。

### 版本控制的目的

在于对软件开发过程中文件或目录的发展过程提供有效的追踪手段，保证在需要时找到旧的版本，避免文件的丢失、修改的丢失和相互覆盖，通过对版本库的访问控制避免未经授权的访问和修改。另外软件控制是实现团队并行开发、提高开发效率的基础。

### 版本控制的实际

实际上，对版本的控制就是对版本的各种操作的控制，包括控制检入检出、版本的分支和合并、版本的历史记录。

## 变更控制

1. 对于大型的软件开发项目，无控制的变更将迅速导致混乱，使整个项目无法顺利进行下去而失败。
2. 变更控制就是通过结合人为的规程和自动化工具，以提供一个变化控制的机制。
3. 变更控制的对象主要指配置库中的各基线配置项
4. 变更管理的一般流程是：
5. 由开发人员或系统分析人员提出变更需求；
6. 由CCB（变更控制委员会）或项目经理审核并决定是否批准；
7. 配置管理员根据CCB或项目经理 的决定开放相应的权限，并形成记录备案；
8. 变更申请人员执行相应的变更。

## 配置库

1. 在项目开发过程中，配置库可分开发库、基线库和产品库三个区域，其各自存放的内容及存取的规定为：
2. 开发库：开发库存放项目组所遵循的过程标准、参考资料、所有未经批准的配置项、已经批准但未纳入基线的配置项，此区域中的配置项由项目经理负责和控制，项目总结结束后删除。
3. 基线库：基线库存放基线。此区域的配置项由项目经理或CCB评审批准后，由配置管理员从开发库更新而来，此区属配置管理员所有。
4. 产品库：产品库存放基线库中形成了一个对外发布的版本。此区域的配置项由项目经理或CCB评审批准后，由配置管理员从基线库更新而来，此区属配置管理员所有。

### 建立配置管理系统（SVN配置库、VSS配置库）

不论采取何种存储方式，只要存放的存储库，均叫做配置库：记录配置项有关的所有信息，存放受控的配置项

分支：开发申请分支建立审批通过后，才能标记为开发库，即为Branch；开发人员在分支库上进行开发工作

主干：测试通过的正式发布的代码，才能标记为受控库，即为Trunk。分支测试稳定后，组长定期合入Truck库

标签：申请基线建立审批通过后，才能标记为产品库，即为Tag

trunk通常是放主干程序的，最近最新的可编译执行的代码。

Branches 中存放该项目的一些分支，通常是一些新功能的添加或者bug修复打出的分支。最终分支的程序会merge到trunk中。

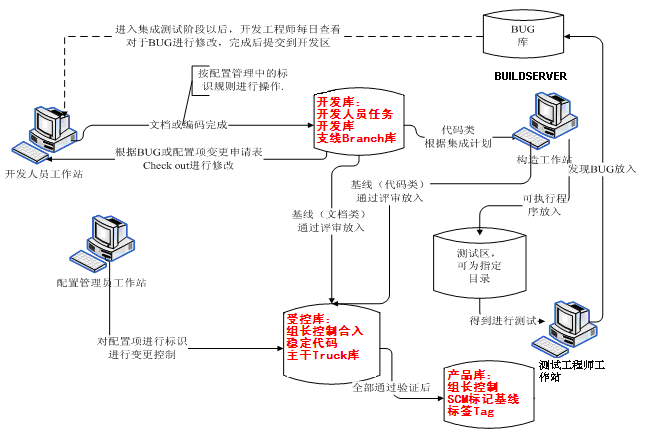
Tags一般只有增加权限，没有删除修改权限，tags中存放一些比较稳定的已经或者曾经上线的版本。

对应关系：

|  |  |  |  |
| --- | --- | --- | --- |
| SVN权限 | SVN | VSS | VSS权限 |
| R（Read）  R/W（Read/Write）  No Access | 分支（branch） | 开发库 | R（Read）  C（Check Out/CheckIn）  A（Add/Rename/Delete）  D（Destory） |
| 主干（trunk） | 受控库 |
| 标签(tag ) | 产品库 |

建库模式：按配置项类型分类建库、按任务建库

### 配置库使用流程图



### 配置库使用原则

1. 软件工程师按如下原则使用配置库：
2. 只能访问开发库。
3. 在添加配置项后，按公司版本的约定打标识，给定一个初始版本；
4. 签入/签出不需要更新标识；
5. 当工作产品完成之后，签入后，按公司版本约定打标识；
6. 如果需要再修改，则签出；
7. 修改完成后签入，三级或四级版本号加一，按打上面版本约定打标识
8. 依次类推，直到该配置项完全定稿。
9. 配置管理员按如下原则使用基线配置库：
10. 拥有基线库的全部权限，建立基线库并分配操作权限；
11. 把评审通过的配置项根据评审后确定的版本，打上发布版本标识；
12. 根据审计过的版本控制表生成发布基线，从开发库把配置项移到基线库；之后，使用基线库发布该版本的工作产品；
13. 负责基线配置库的日常维护及备份；
14. 发布时定期或事件驱动从基线配置库生成配置状态报告。
15. 配置管理员按如下原则使用产品库：
16. 拥有产品库的全部权限，建立产品库并分配操作权限；
17. 把评审通过的配置项根据评审后确定的版本，打上产品版本标识；
18. 根据审计过的版本控制表生成产品基线，从基线库把配置项移到产品库；之后，锁定该版本的工作产品；
19. 负责产品库的日常维护及备份；
20. 发布时定期或事件驱动从配置库生成配置状态报告。

### 配置管理活动及方针

1. 项目组软件配置管理应有专人负责（称配置管理员）；中小型项目，由项目经理或指定专人担任配置管理员，负责项目配置管理。大规模项目，应建立配置管理小组（CM组），在项目经理领导或授权下负责项目配置管理。
2. 配置管理贯穿软件生命周期全过程，但分两个阶段：从需求到产品发布的开发阶段，配置管理由项目经理或指定专人负责；发布后进入产品维护阶段，由负责该产品技术支持部门指定的配置管理员负责。
3. 在整个开发阶段，各类工作产品（配置项）及其变更是项目配置管理的重点；而开发环境、测试环境和运行环境的描述文档则只作为配置项纳入配置管理，受到控制。
4. 在产品维护阶段，配置管理的重点则包括变更控制、版本控制和基线管理。
5. 项目启动后就应开始配置管理活动，包括：定义、标识配置项，定义基线，建立配置库和基线库，确定访问权限，控制配置库/基线库的签出（Check out）和签入（Check in）。
6. 在项目计划阶段，应编写配置管理计划（CM计划），与项目开发计划一起提交评审；在产品发布后进入维护阶段，也应编写CM计划。
7. 按评审确认的CM计划建立基线、审计配置库和基线，及时报告配置状态。
8. 每一个产品的所有配置项的变更均应得到管理和控制。
9. （每一个项目组的）软件产品最终集成（产品发布基线，或产品发布后的产品维护阶段定期生成的基线），由项目配置管理员负责实施，技术支持的配置管理员负责监督。

## 基线

1. 基线，由一个或若干个通过（正式）评审并得到确认的配置项组成，是项目进入下一个生命周期阶段的出发点（或基准）。
2. 基线是软件文档或源码(或其它产出物)的一个稳定版本，它是进一步开发的基础，只有经过授权后才能变更。建立一个初始基线后，以后每次对其进行的变更都将记录为一个差值，直到建成下一个基线。
3. 建立基线的原因：
4. 重现性：及时重新生成软件系统给定发布版本的能力，重新生成开发环境。
5. 可追踪性：建立项目工作产品之间的前后继承关系，确保设计满足要求、代码满足设计及用正确的代码编译系统。
6. 报告：来源于基线之间内容的比较，有助于调试并生成发布说明。
7. 建立基线的优点：
8. 为开发工作提供了一个定点和快照。
9. 新项目可以从基线提供的定点建立，作为一个单独分支，新项目将与随后对原始项目所进行的变更进行隔离。
10. 各开发人员可以将建有基线的工作产品作为他在隔离的私有工作区中进行更新的基础。
11. 当认为更新不稳定或不可信时，基线为团队提供一种取消变更的方法。

### 常用的基线

对于配置管理，有以下三种基线：分配基线（需求）、功能基线（设计）和产品基线（测试）。

#### 分配基线（Allocated Baseline）

分配基线指在软件需求分析阶段结束时，经过正式评审和批准的软件需求规格说明。分配基线是最初批准的分配配置标识。

#### 功能基线（Functional Baseline）

功能基线指在系统分析与软件定义阶段结束时，在经过正式评审和批准的系统设计规格说明书中对开发系统的规格说明；或是指在经过项目委托单位和项目承办单位双方签字同意的协议书或合同中，所规定的对开发软件系统的规格说明；或是由下级申请并经上级同意或直接由上级下达的项目任务书中所规定的对开发软件系统的规格说明。功能基线是最初批准的功能配置标识。

#### 产品基线（Product Baseline）

产品基线指在软件组装与系统测试阶段结束时，经过正式评审和批准的有关软件产品的全部配置项的规格说明。产品基线是最初批准的产品配置标识。

### 基线建立指南

基线，由一个或若干个通过（正式）评审并得到确认的配置项组成，是项目进入下一个生命周期阶段的出发点（或基准）。 有定义可知，只有经过**正式评审**的基线申请，才能形成基线



所有基线建立，需提出人填写申请单，组织评审，通过后方可建立

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 一类基线 | 二类基线 | 基线  名称 | 建立时机 | 基线说明 | 是否可裁剪 |
| 项目计划  基线 |  | Plan | 项目启动和项目计划阶段完成之后 | 可以包括立项申请，各种计划，以及技术合同等资料 | 可裁剪 |
| 需求分析  基线 |  | Srs | 需求分析阶段完成之后 | SRS，评审记录，测试文档等需求阶段相关文档 | 不可裁剪 |
| 概要设计  基线 |  | Bld | 概要设计阶段完成之后 | 概要设计相关文档以及需求等上游相关文档 | 不可裁剪 |
|  | 详细设计  基线 | Lld | 每一个迭代的详细设计阶段完成之后 | 详细设计相关文档以及需求等上游相关文档 | 可裁剪 |
|  | 代码基线 | Code | 在迭代中，某些集成测试完成之后，并且代码的质量达到开发要求，相关代码已经稳定之后 | 可以有多条这样的基线，再配置管理计划中进行计划 | 可裁剪 |
|  | 系统测试  基线 | alpha | 在每一个迭代阶段，当测试计划，测试用例等通过评审之后；正式实施系统测试之前 | 系统测试相关文档，代码，用户手册等以及上游相关文档。  本基线可以有多条，因为系统测试可能会beta开展多次，以达到交gold付要求。 | 可裁剪 |
| 交付基线 |  |  | 系统交付申请通过评审之后，正式交付用户使用之前 | 交付相关文档，用户手册，代码，验收测试相关文档，所有系统相关文档。交付可能有多个版本，如alpha ,beta等，都要作出基线 | 不可裁剪 |
| 产品基线 |  |  | 项目结束之后 | 所有相关文档 | 不可裁剪 |

### 基线命名规定

基线(BaseLine)由一组配置项组成，这些配置项构成了一个相对稳定的逻辑实体。基线中的配置项呗“冻结”了，不能再被任何人随意修改。基线通常对应开发过程中的里程碑(Milestone)，一个产品可以有多个基线，也可以只有一个基线。

基线的命名规则如下ProduceName+Vx.x.x.x-[ BaseLineName-N]

1. Produce：表示产品名称
2. Vx.x.x.x：表示产品的版本号，X可以是一位数字，也可以是多位数
3. BaseLineName：标识基线名称，常见的名称有：
4. Plan：表示计划基线
5. Srs：表示软件需求规格基线
6. Hld：表示软件概要设计基线
7. Lld：表示软件详细设计基线
8. Code：表示软件代码基线
9. Alpha：表示内部系统测试版本基线
10. Beta：表示外部测试版本基线，如提交给客户验收测试，客户试用的
11. Gold：表示正式发行的版本，是一个比较稳定的版本
12. N：表示基线序号，从1开始编号，对于正式发行的版本，无此编号

例如：

机房系列V6.6.3.5-alpha-2，表示机房系列V6.6.3.5内部alpha测试第二版

机房系列V6.6.3.5-gold，表示机房系列V6.6.3.5正式版本

## 分支

### 分支建立指南



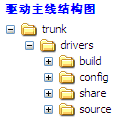
所有分支建立，需提出人填写申请单，组织评审，通过后方可建立

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 分支类型 | 分支名称 | 建立时机 | 分支说明 | 分支发起人 | 分支执行人 |
| 驱动整体发布分支 | Drivers\_NUM\_功能说明 | 需要整体发布的分支申请 | 以发布整体程序的分支申请建立 | 组长/产品经理  项目经理 | 配置组 |
| 驱动单独程序分支 | 程序名\_版本号\_功能说明 | 需要单独发布的分支申请 | 以发布单独程序的分支申请建立 | 组长/产品经理  项目经理 | 配置组 |
| 产品整体分支 | 版本号\_NUM\_功能说明 | 需要整体发布的分支申请 | 以发布整体程序的分支申请建立 | 组长/产品经理  项目经理 | 配置组 |
| 产品单独程序分支 | 程序名\_版本号\_功能说明 | 需要单独发布的分支申请 | 以发布单独程序的分支申请建立 | 组长/产品经理  项目经理 | 配置组 |

### 分支命名规定

|  |  |
| --- | --- |
| 分支类型 | 分支名称 |
| 驱动整体发布分支 | Drivers\_NUM\_功能说明 |
| 驱动单独程序分支 | 程序名\_版本号\_功能说明 |
| 产品整体分支 | 版本号\_NUM\_功能说明 |
| 产品单独程序分支 | 程序名\_版本号\_功能说明 |

### 分支/标签建立实例



## 词汇表

### CM（Configuration management）

配置管理。

### CCB（Change control board）

变更控制委员会。

### CI（Configuration item）

配置项，包含文档、程序。

### CR（Change Request）

变更请求，对提出的要变更工件或流程的任何请求的统称。在变更请求中记录的信息是有关当前问题、提议解决方案及其成本的起源和影响的信息。

### PCA（Physical Configuration Audit）

物理审计，在配置管理系统中建立基线的工件是否为“正确”版本。

### FCA（Functional Configuration Audit）

功能审计，核实软件配置项的实际性能是否符合它的需求。

### 基线（Baseline）

已通过复审和批准的工件发布版，由此构成进一步演进或开发的公认基础，并且只能通过正式程序，例如变更管理和配置控制才能进行更改。

### 配置管理库（Configuration Management Library）

存储项目工件的所有版本，即存储项目的定义的配置项。

### 版本（Version）

某个工件的变体，工件的后期版本一般是在初期版本的基础上进行的扩展。

1. [工作说明书](http://baike.baidu.com/view/478773.htm)(Statement Of Work,缩写为SOW)是合同的附件之一，具有与合同正文同等的法律效力。工作说明书详细规定了合同双方在合同期内应完成的工作，如方案论证、设计、分析、试验、质量控制，可靠性、维修性、保障性、标准化、计量保证等；应向对方提供的项目，如接口控制文件、硬件、计算机软件、技术报告、图纸、资料，以及何时进行何种评审等，因此，工作说明书以契约性文件的形式进一步明确了顾客的要求和承制方为实现顾客要求必须开展的工作，它使产品的管理和质量保证建立在法律依据之上，成为合同甲方（顾客）对乙方(承制单位)进行质量监控的有力工具。工作说明书的详细要求可查阅GJB 2742－96。工作说明书的内容是质量功能展开的重要输入。 [↑](#footnote-ref-2)
2. “特殊版本标识符”必要时才使用，存在项目支线版本时，可加入项目名称、项目中的版本号等信息。 [↑](#footnote-ref-3)
3. 存在项目支线版本时，可加入项目名称、项目中的版本号、编译日期等信息。 [↑](#footnote-ref-4)