infohold-logo.png

智控-平台-规范

配置管理规范

密 级：□ 普通 □ 秘密 □ 机密 ■ 绝密

文档编号：IH\_CP\_SPC\_001

| 版本号 | 版本说明/变更理由/变更内容 | 作者/日期 | 审批人/日期 | 备注 |
| --- | --- | --- | --- | --- |
| 0.5 | 初始创建/C | 杨明/20110607 |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

变更说明：C：Create，初始创建；A：Add，增加内容；M：Mod，修改；D：Del，删除

目录

[1 概述 4](#_Toc295401824)

[1.1 简介 4](#_Toc295401825)

[1.2 目的 4](#_Toc295401826)

[1.3 适用范围 4](#_Toc295401827)

[1.4 文档读者 4](#_Toc295401828)

[1.5 词汇表 4](#_Toc295401829)

[1.6 参考资料 5](#_Toc295401830)

[2 配置管理规范 5](#_Toc295401831)

[2.1 配置管理流程图 5](#_Toc295401832)

[2.2 角色 5](#_Toc295401833)

[2.3 用户组 5](#_Toc295401834)

[2.4 活动 6](#_Toc295401835)

[2.5 版本号规范 6](#_Toc295401836)

[2.5.1 格式 6](#_Toc295401837)

[2.5.2 策略 7](#_Toc295401838)

[2.6 文档类配置项标识规范 7](#_Toc295401839)

[2.6.1 文档编号规范 7](#_Toc295401840)

[2.6.2 文档标识规范 7](#_Toc295401841)

[2.7 配置库组织结构 9](#_Toc295401842)

[2.7.1 配置库组织结构 9](#_Toc295401843)

[2.7.2 目录使用说明 9](#_Toc295401844)

[3 配置项发布规范 10](#_Toc295401845)

[3.1 文档类配置项发布规范 10](#_Toc295401846)

[3.2 程序类配置项发布规范 10](#_Toc295401847)

[3.2.1 目的 10](#_Toc295401848)

[3.2.2 策略 10](#_Toc295401849)

[4 建议配置安全控制规范 10](#_Toc295401850)

[4.1 用户名配置管理规范 10](#_Toc295401851)

[4.2 用户密码配置管理规范 10](#_Toc295401852)

[4.3 添加用户配置管理规范 11](#_Toc295401853)

[4.4 删除用户配置管理规范 12](#_Toc295401854)

[5 配置环境 13](#_Toc295401855)

[5.1 硬件环境 13](#_Toc295401856)

[5.2 操作系统 13](#_Toc295401857)

[5.3 管理工具 13](#_Toc295401858)

[6 配置库备份规范 13](#_Toc295401859)

[6.1 活动 13](#_Toc295401860)

[6.2 配置库备份文件标识规范 13](#_Toc295401861)

[7 配置管理工具 14](#_Toc295401862)

[7.1 Subversion 14](#_Toc295401863)

[7.2 TortoiseSVN 14](#_Toc295401864)

[7.2.1 安装 15](#_Toc295401865)

[7.3 Subclipse 15](#_Toc295401866)

[7.3.1 安装 15](#_Toc295401867)

# 概述

## 简介

## 目的

本文档指导产品研发交付部开展IH Enterprise Payments Platform V1产品开发的配置管理活动。

## 适用范围

本规范的适用范围包括产品研发交付部的全部工作产品以及相关评审记录。

本规范只在本部门范围内有效，对其他部门没有参考作用。

## 文档读者

本规范的预期读者为智控美信相关领导和参与本部门的所有管理、设计、开发人员，未经智控美信书面允许，请勿扩散。

## 词汇表

iEPP：智控企业结算支付平台（IH Enterprise Payments Platform）

iBPP：业务流程平台（IH Business Process Platform）

CI：配置项（Configuration Item），包含文档、程序

CM：配置管理(Configuration Management)

BL：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

CML：配置管理库（Configuration Management Library），存储产品工件的所有版本，即存储产品的定义的配置项

CCB：变更控制委员会（Change control board）

## 参考资料

# 配置管理规范

## 配置管理流程图



## 角色

|  |  |
| --- | --- |
| 角色名称 | 职责 |
| 配置管理经理 | 配置管理经理负责为产品开发团队提供全面的配置管理基础设施和环境，负责制定配置管理计划，负责对文档类的配置项建立基线，负责基线审计，报告配置状态 |
| CCB | 评审变更 |
| 任意角色 | 提交配置项 |

## 用户组

|  |  |
| --- | --- |
| 组名称 | 描述 |
| Manager | 管理组，具有管理权限 |
| Developers | 开发者组，主要针对源代码库进行操作 |
| Implementers | 实施组，主要针对产品实施库进行操作 |
| Others | 其他，现阶段未进行分类的只具有浏览权限的人员组 |

## 活动

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 活动名称 | 角色 | 活动描述 |
| 1 | 计划配置管理 | 配置管理经理  CCB | 制定配置管理策略  制定变更控制策略  编写配置管理计划  评审配置管理计划 |
| 2 | 创建配置管理环境 | 配置管理经理 | 设置硬件环境  设置网络环境  设置软件环境  安装配置管理工具  建立一个配置管理库，储存项目中定义的配置项  提供配置管理培训 |
| 3 | 配置项标识 | 配置管理经理 | 对文档类的配置项进行的标识  对程序的配置项进行标识 |
| 5 | 报告配置状态 | 配置管理经理 | 根据配置管理计划，收集配置活动数据， 编写配置状态报告。 |
| 7 | 变更控制管理 | CCB | 文档类：任意角色提交变更请求，CCB进行评审  程序类：任意角色提交变更请求，CCB进行评审 |

## 版本号规范

版本号是版本的标识号，它能使用户了解所使用的操作系统或软件是否为最新的版本以及它所提供的功能与设施。每一个版本号可以分为主版本号与次版本号两部分。

### 格式

本规范中版本号采用GNU风格：

**主版本号.子版本号[.修正版本号] [后缀]**

说明：

1. 主版本号：格式为VX，其中“V”为常量，X代表自然数
2. 子版本号：取自然数
3. 修正版本号：格式为YYYYMMDD[HHMMSS],语义为发布或者构建时间
4. 后缀：修饰性、说明性的词语，如：alphal（α）、beta（β）等，详见附录《版本号修饰符说明》

### 策略

1. 初版本时，版本号可定为0.1或0.1.0，也可定为1.0或1.0.0；
2. 进行局部修改或bug修正时，主版本号和子版本号都不变，修正版本号加1；
3. 原有基础上增加部分功能后，主版本号不变，子版本号加1，修正版本号复位为0，可以忽略；
4. 进行重大修改或局部修正累积较多，导致产品整体发生全局变化时，主版本号加1;

## 文档类配置项标识规范

### 文档编号规范

详见附录三《文档编号说明》

### 文档标识规范

#### 软件过程文档标识规范

格式：IH Enterprise Payments Platform V1文档主题[\_功能名称]\_版本号

说明：

1. 文档主题，如需求分析报告、概要设计说明等
2. 功能名称：如业务流程平台、资产管理等

例：

1. 《IH Enterprise Payments Platform V1需求分析报告\_V0.6.0》
2. 《IH Enterprise Payments Platform V1概要设计说明\_业务流程平台\_V1.0.20100324》

#### 管理过程文档标识规范

1. 工作计划

格式：IH Enterprise Payments Platform V1\_XX工作计划[\_YYYYMMDD]

说明：

1. XX工作计划，某工作的工作计划
2. YYYYMMDD为计划开始的日期，其中：YYYY为年，如2010；MM为月，如08（不足两位的前面补零）；DD为日（不足两位的前面补零）

例：《IH Enterprise Payments Platform V1\_业务流程平台升级\_20071201》

1. 与会通知

格式：与会通知[\_XX组]\_主题[\_YYYYMMDD]

说明：

1. XX组，如资产管理组
2. 会议主题
3. YYYYMMDD为计划开始的日期，其中：YYYY为年，如2010；MM为月，如08（不足两位的前面补零）；DD为日（不足两位的前面补零）

例：《与会通知\_业务流程平台升级方案讨论会\_20071201》

1. 会议记录

格式：产品研发交付部\_会议类型[\_XX模块组] [\_会议主题]\_YYYYMMDD

说明：

1. 会议类型：如例会、管理协调会议、技术讨论会、需求变更评审会等
2. XX模块组，如资产管理组
3. YYYYMMDD为会议开始的日期，其中：YYYY为年，如2010；MM为月，如08（不足两位的前面补零）；DD为日（不足两位的前面补零）

例：

《产品研发交付部\_例会\_20090921》

《产品研发交付部\_技术讨论会\_首页性能改进方案\_20091025》

1. 个人文档（建议）

格式：文档类型-NN-YYYYMMDD-文档主题

说明：

1. 文档类型：如工作总结报告等
2. NN：撰写人姓名
3. YYYYMMDD为会议开始的日期，其中：YYYY为年，如2010；MM为月，如08（不足两位的前面补零）；DD为日（不足两位的前面补零）
4. 文档主题：文档的主题。

例：《工作总结报告-张三-20070901-8月份工作报告》

## 配置库组织结构

### 配置库组织结构

详见《附录二 配置库组织结构》

### 目录使用说明

* trunk：主版本，存储最新版开发内容
* branches：子版本存储，基于主线生成，用于发布
* tag：发布版本，branch中测试完成版本进入tag

选择性merge

选择性merge

branch 3.2

branch 3.3

tag3.3.2

tag3.3.1

tag3.2.2

tag3.2.1

trunk 3….

如图所述：

所有开发计划中规划内容在trunk中进行开发，当开发完成后经过确认批准，冻结trunk代码；从trunk中标记出branch，解冻trunk代码，并在进行集成、集中测试修改后，经过确认批准将代码标记为tag，此tag中代码作为版本发布基线，tag中代码为冻结状态，不允许修改；在完成tag后，再发现bug，在相应版本的支线上进行修改，累积到一定程度（或时间）后标记处升级版本的tag，并检出相应补丁；修改完成的bug选择性merge回trunk中。

# 配置项发布规范

## 文档类配置项发布规范

## 程序类配置项发布规范

### 目的

构建可跟踪的产品发布过程。

### 策略

#### 内部测试发布策略

步骤一：第一次发布采用完全检出的方式，记录版本，执行步骤二；

步骤二：每日工作结束后，在从版本库中进行增量提取，并提交说明，执行步骤三；

步骤三：测试组次日进行测试，提交步骤二发布版本测试报告，后执行步骤四；

步骤四：若发生产品版本发生重大变更，执行步骤一；否则执行步骤二；以上两种情况都不符合，执行步骤五；

步骤五：产品研发结束，结束流程；其他情况具体问题具体分析；

# 建议配置安全控制规范

## 用户名配置管理规范

（1）使用用户姓名全拼，且均为小写字母

（2）若出现同名用户则在后添加用用户名后增加区别标识，如用户加入日期

## 用户密码配置管理规范

（1）使用用户姓名全拼，且均为小写字母

（2）使用MD5进行加密

## 添加用户配置管理规范



添加用户命令： htpasswd -m /etc/httpd/conf.d/svn\_user\_list xx

其中：

1、使用root用户登录，使用获取root执行权限；

2、/etc/httpd/conf.d/svn\_user\_list密码保存文件

3、xx代表用户名，一般使用全拼

4、密码要求格式如：ym's\_svn^20110406，“'s\_svn^”为固定字符串，“'”前为用户名简拼， “^”后为添加用户日期

5、添加完用户后需要对其权限进行指定，即修改/etc/httpd/conf.d/svn\_projects.access，进行权限配置，将所添加用户指定至适合组，目前有3个组：manager（管理人员）、developer（开发者）、qa（质量管理），其中r代表只读，w代表可写，rw代表可读可写。

## 删除用户配置管理规范



删除用户命令： htpasswd -m /etc/httpd/conf.d/svn\_user\_list xx

其中：

1、使用itsm用户登录，使用获取root执行权限，需要输入itsm密码；

2、/etc/httpd/conf.d/svn\_user\_list密码保存文件

3、xx代表用户名

4、修改/etc/httpd/conf.d/svn\_projects.access，删除用户。

# 配置环境

## 硬件环境

|  |  |
| --- | --- |
| CPU |  |
| 内存 | 8G |
| 硬盘 | 150G |
| 网卡 |  |

## 操作系统

|  |  |
| --- | --- |
| 名称 | Red Hat Enterp Linux |
| 版本 | 5 |

## 管理工具

|  |  |
| --- | --- |
| 版本控制 | Subversion |
| Http服务器 | Apache 2.2 |

# 配置库备份规范

## 活动

|  |  |  |
| --- | --- | --- |
| 活动名称 | 角色 | 活动描述 |
| 定期备份 | 配置管理员 | 1. 每两个自然月在月底进行一次完全备份 2. 每各自然月内每两周进行一次增量备份 |

## 配置库备份文件标识规范

规则：产品研发交付部-备份-配置库标识备份（完整|增量）-YYMMDD

说明：

1. 配置库标识：配置库的唯一标识
2. （完整|增量）：至少为一种备份方式，如
3. YYYYMMDD为会议开始的日期，其中：YYYY为年，如2010；MM为月，如08（不足两位的前面补零）；DD为日（不足两位的前面补零）

# 配置管理工具

## Subversion

Subversion(简称[SVN](http://baike.baidu.com/view/183128.htm" \t "_blank))是近年来崛起的版本管理软件，是[CVS](http://baike.baidu.com/view/20982.htm" \t "_blank)的接班人。目前，绝大多数开源软件都使用[SVN](http://baike.baidu.com/view/183128.htm" \t "_blank)作为代码版本管理软件。

SVN与CVS对比的优点如下：

* + 1. 统一的版本号。CVS是对每个文件顺序编排版本号，在某一时间各文件的版本号各不相同。而Subversion下，任何一次提交都会对所有文件增加到同一个新版本号，即使是提交并不涉及的文件。所以，各文件在某任意时间的版本号是相同的。版本号相同的文件构成软件的一个版本。
    2. 原子提交。一次提交不管是单个还是多个文件，都是作为一个整体提交的。在这当中发生的意外例如传输中断，不会引起数据库的不完整和数据损坏。
    3. 重命名、复制、删除文件等动作都保存在版本历史记录当中。
    4. 对于二进制文件，使用了节省空间的保存方法。（简单的理解，就是只保存和上一版本不同之处）
    5. 目录也有版本历史。整个目录树可以被移动或者复制，操作很简单，而且能够保留全部版本记录。
    6. 分支的开销非常小。
    7. 优化过的数据库访问，使得一些操作不必访问数据库就可以做到。这样减少了很多不必要的和数据库主机之间的网络流量。

## TortoiseSVN

TortoiseSVN是Subversion版本控制系统的一个免费开源客户端，可以超越时间的管理文件和目录。文件保存在中央版本库，除了能记住文件和目录的每次修改以外，版本库非常像普通的文件服务器。你可以将文件恢复到过去的版本，并且可以通过检查历史知道数据做了哪些修改，谁做的修改。

### 安装

1. 下载地址：

\\192.168.5.13\public\Software\Developer\SCM\Version\subversion\TortoiseSVN-1.6.16.21511-win32-svn-1.6.17.msi

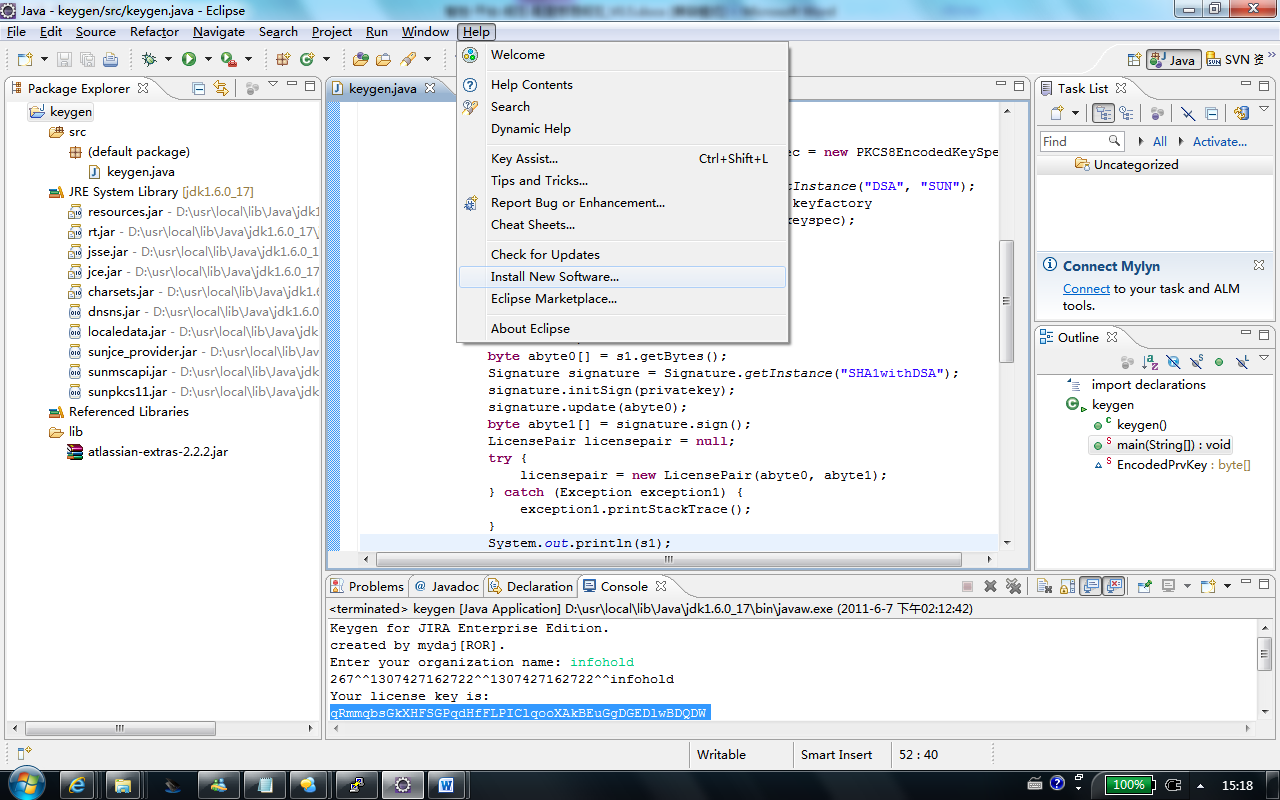
1. 双击进行安装

## Subclipse

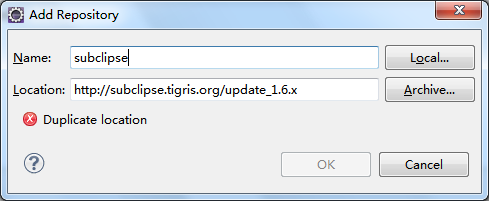
Subclipse是Eclipse开发团队提供的一个Subversion免费开源客户端，以插件的形式运行存在于Eclipse之中。

### 安装

1. 启动Eclipse，依次点击“Help”，“Install New Software …”



1. 点击“add”，添加如下地址：http://subclipse.tigris.org/update\_1.6.x



1. 添加完成在线更新安装subclipse

附录一：版本号修饰符说明

|  |  |
| --- | --- |
| 修饰符 | 说明 |
| alphal(α) | 此版本表示该软件仅仅是一个初步完成品，通常只在软件开发者内部交流，也有很少一部分发布给专业测试人员 |
| beta（β） | 该版本相对于α版已有了很大的改进，消除了严重的错误，但还是存在着一些缺陷，需要经过大规模的发布测试来进一步消除 |
| gamma（γ） | 该版本已经相当成熟，与即将发行的正式版相差无几 |
| trial | 该版本软件进行时间限制，或在功能上做一定的限制 |
| demo | 演示版，该本版集成了正式版中的几个功能，不同的是，demo版一般不能通过升级或注册的方法变为正式版 |
| release | 最终释放版，一般用r代替 |
| standard | 标准版中包含了该软件的基本组件及一些常用功能，满足一般用户的需求 |
| deluxe | 豪华版具有标准版的所有功能外，还包含其他附加功能 |
| professional | 专业版是针对某些特定的用户而言的，其中许多功能是标准版中所没有的，这些内容对于一个特定用户是极为重要的 |
| enterprise |  |
| update |  |

附录二：配置库组织结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 一级 | 二级 | 三级 | 四级 | 五级 | 说明 |
| Products | [Platform] | doc | trunk | 01.demand | 需求规格说明书 |
| 02.design | 设计说明书 |
| 03.coding | 编码规范等 |
| 04.testing | 测试相关文档 |
| 05.handbook | 操作手册 |
| 06.change | 变更评审记录 |
| 07.meeting | 会议通知、会议记录 |
| 08.reference | 相关资料、模板 |
| branches |  |  |
|  |  |
| tags |  | 文档基线 |
| code | trunk |  | 源程序主版本存储目录 |
| branches |  | 版本分支源程序存储目录 |
| [分支程序模块] | 子版本源程序存储目录 |
| tags |  | 产品发布目录 |
| Projects | [Project] | code | branches |  |  |
| tags |  | 项目源程序里程碑 |
| trunk |  | 项目源程序主线 |
| doc |  |  | 根据项目特点自行规划 |
| Peronal |  |  |  |  | 个人文档目录，不做基线 |
| Resource |  |  |  |  | 公共资源文档 |

附录三：文档编号说明

附录四：配置项更新操作流程规范

附录五：开发环境支撑软件说明

IDE软件支撑：

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | eclipse-jee-helios-SR2-win32.zip | | |
| 下载地址 |  | | |
| 辅助插件 | 名称 | 下载地址 | 说明 |
| subclipse-1.6.5 |  | Eclipse中SVN客户端插件 |
| Propedit-5.3.3 |  | 国际化资源文件编辑器 |
|  |  |  |

本地数据库环境支撑：

|  |  |
| --- | --- |
| 数据库 | Mysql |
| 版本 | 5.02+ |
| 下载地址 |  |

应用服务器

|  |  |
| --- | --- |
| 名称 | Apache Tomcat |
| 版本 | 6.0+ |
| 下载地址 |  |

版本控制客户端

|  |  |
| --- | --- |
| 名称 | TortoiseSVN |
| 版本 | 1.6.6 |
| 下载地址 |  |

附录六：数据库备份实现

附录七：SVN备份

svn备份一般采用三种方式：1）svnadmin dump 2)svnadmin hotcopy 3)svnsync.

注意，svn备份不宜采用普通的文件拷贝方式（除非你备份的时候将库暂停），如copy命令、rsync命令。

优缺点分析：

第一种svnadmin dump是官方推荐的备份方式：

优点：比较灵活，可以全量备份也可以增量备份，并提供了版本恢复机制；

缺点：如果版本比较大，如版本数增长到数万、数十万，那么dump的过程将非常慢；备份耗时，恢复更耗时；不利于快速进行灾难恢复。

第二种svnadmin hotcopy只能进行全量拷贝，不能进行增量备份：

优点：备份过程较快，灾难恢复也很快；如果备份机上已经搭建了svn服务，甚至不需要恢复，只需要进行简单配置即可切换到备份库上工作。

缺点：比较耗费硬盘，需要有较大的硬盘支持。

第三种svnsync实际上是制作2个镜像库，当一个坏了的时候，可以迅速切换到另一个。

优点：当制作成2个镜像库的时候起到双机实时备份的作用；

缺点：当作为2个镜像库使用时，没办法做到“想完全抛弃今天的修改恢复到昨晚的样子”；而当作为普通备份机制每日备份时，操作又较前2种方法麻烦。

我们使用第一种方式进行备份：

1）定义备份策略：

备份频度：每两个月进行一次完整备份，每个月内每两周进行一次增量备份

备份地点：备份存储路径到/usr/local/backup/svn

备份命名：产品研发交付部-备份-（完整|增量）-YYMMDD

备份时间：23:00

备份检查：

2）建立全量备份脚本：

3）建立增量备份脚本：

4）配置/etc/crontab定期执行