

# 基于ClearCase UCM的项目开发目录结构

胡协刚

软件架构师 UML/RUP专家

[szjinco@public.szptt.net.cn](mailto:szjinco@public.szptt.net.cn)

中国软件架构师网

[www.soft-arch.net](http://www.soft-arch.net) [www.chinaarchitect.net](http://www.chinaarchitect.net)

# 开发（产品）目录结构

- 开发目录结构是关系到软件项目健康运作的关键因素；
- 开发（产品）目录提供了项目团队进行开发、管理等活动的统一共享场所，它需要满足不同涉众/角色（分析、设计、测试、管理、支持等人员），在不同的阶段（启始、精化、构建、迁移、维护等），对不同类型工件（源码、文档、模型、安装包等）进行访问的多种场景需求；
- 开发目录的结构本身是静态的，但在整个项目生命周期中，其所存放工件内容以至组织结构却在不断地变化，因此它也具有动态的特性。

# 目录结构的组织原则

- 目录必须采用一个明确的工件分类方案，帮助使用者将工件存放到正确的地方，同时方便使用者迅速找到或定位他所需要的工件；
- 目录应当避免出现冗余的工件副本，并且像软件一样具备良好的层次结构，它应当最充分地体现分而治之的组织原则；
- 同一目录内容高内聚、不同目录依赖低耦合，从而帮助使用者将注意力集中于当前的工作上；

# 目录结构的组织原则

---

- 开发（产品）目录应当面向团队协作，既要做到实施员在私有空间开发时相对独立，不受他人工作的影响，又能保证与他人的工作成果同步，并顺利地实现集成。
- 目录与工件都必须受控，这种控制既针对其静态结构，又要涵盖其动态的演变过程，并确保其历史版本记录和基线的备份。

# 源码目录组织原则

- 开发（产品）目录的组织应当尽量在同一级的目录中采用同一种分类标准。
- 目录分类标准有：
  - 按照工件类型  
(src/bin/lib/doc/md1/script/web/xdoc) ,
  - 按照工件功能或用途  
(build/test/interface/example) ,
  - 按照活动组织 (plan/requirement/design) ,
  - 按照归属关系 (第三方库3rd Lib/基础构件Components/子系统Subsystem) ,
  - . . .

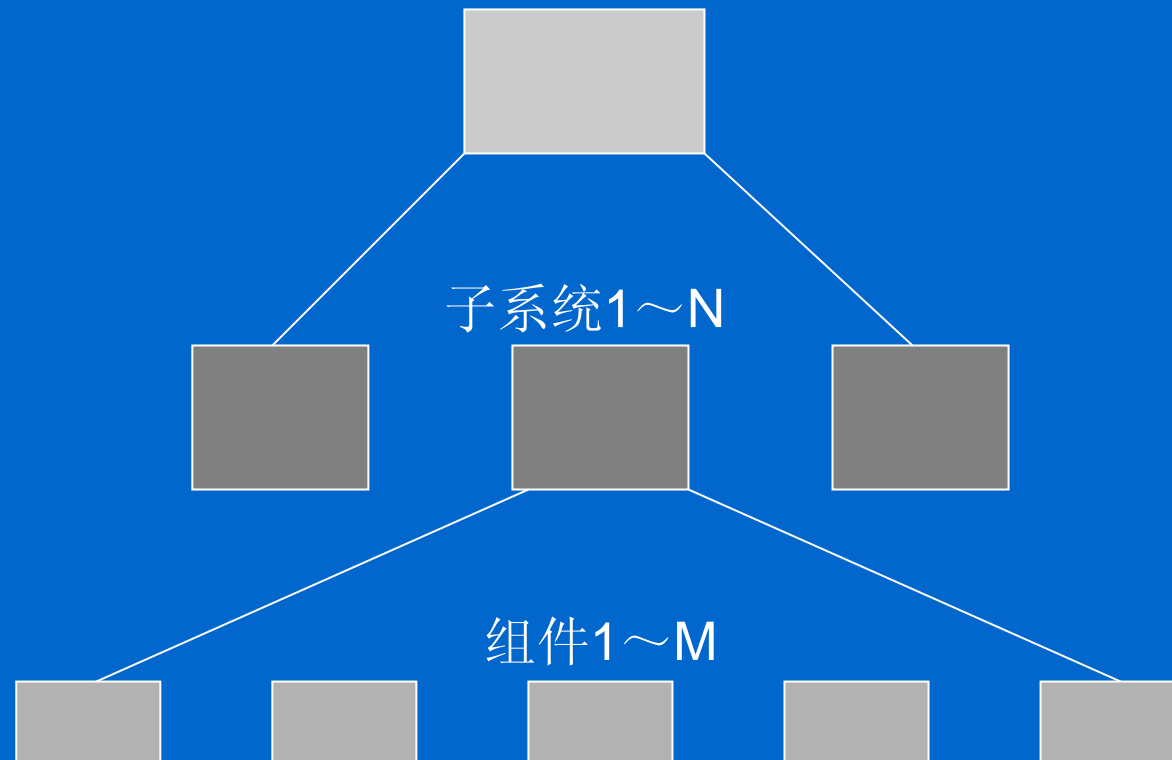
# 示例：产品目录嵌套结构

—产品目录结构为所有具有版本号的产品提供逻辑嵌套的占位符

系统级：

子系统级：

构件级：





# 示例：系统级产品目录结构

系统需求	模型	用例模型	用例包
		用户界面原型	
	文档	软件需求规约	
	报告	用例报告	
系统设计 与实施	模型	分析模型	用例实现
		设计模型	设计子系统
			接口
			测试包
		数据模型	
	文档	软件设计文档	
	子系统—1	子系统目录结构	
	子系统—N	子系统目录结构	

# 示例：系统级产品目录结构

系统集成	计划	系统集成建立计划
系统测试	计划	
	测试用例	
	测试数据	
	测试结果	
系统部署	计划	
	文档	发布说明
	手册	用户手册、培训材料
系统管理	计划	
	评估	
工具		
标准与指南		



# 示例：子系统级产品目录结构

子系统—N 系统需求	模型	用例模型	用例包
		用户界面原型	
	文档	软件需求规约	
	报告	用例报告	
子系统—N 系统设计与实施	模型	分析模型	用例实现
		子系统 设计模型	设计包
			接口包
			测试包
		数据模型	
	文档	软件设计文档	
	构件/模块—1	构件目录结构	
	构件/模块—N	构件目录结构	

# 示例：构件/模块级产品目录结构

构件	源代码
	可执行文件
	接口
	测试代码
	可执行文件测试脚本
	测试数据
	测试结果

# 目录结构的实施依赖于配置管理工具

---

- 做为项目团队进行开发、管理等活动的统一共享场所，开发目录的实施首先必须支持并发操作；网络目录等提供了一定的支持，但实际上只有专门的配置管理工具才能较好地胜任这些工作
- 目录本身处于动态的变化过程中，这意味着需要有工具来支持目录的版本管理；直接支持目录版本控制的配置管理工具主要有IBM Rational的ClearCase

# 概念： 软件配置管理

所谓配置管理是指——在整个软件生命周期过程中对软件的配置及其变更进行管理和控制；（即对开发目录内容的动态变化进行控制）

配置管理的目标有——

- 建立和维护软件产品在整个项目生命周期中的完整性和可跟踪性
  - ✓ 所有配置项（工件）被标识出来
  - ✓ 所有的变更都是受控的
  - ✓ 过程可跟踪
- 支持并行开发（提供团队共享工作环境）
- 提供项目管理的基础
- 最大限度地减少错误，提高生产率

# 概念：统一变更管理 (UCM)

---

- 统一变更管理 (UCM) 是 Rational 用来管理从需求到发布的软件系统开发过程中的变更的途径。

UCM 贯穿整个开发生命周期，确定如何管理对需求、设计模型、文档、构件、测试用例和源代码的变更。

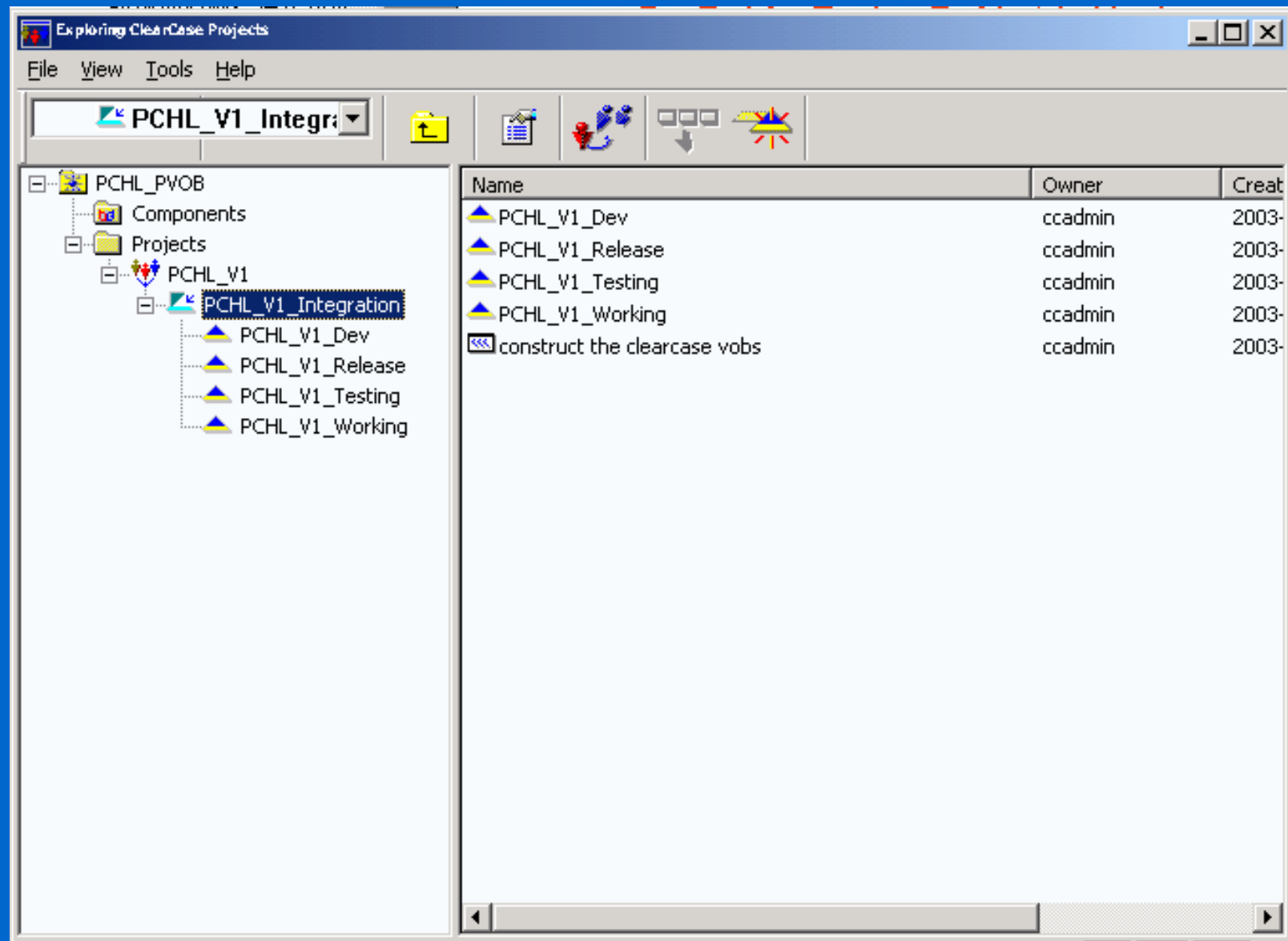
- UCM 模型的一个重要方面在于，它使得用于计划与追踪项目进展的活动与正在实施变更的工件统一起来。

# ClearCase 与 ClearQuest

---

- ClearCase 管理软件工程生成的所有工件；而 ClearQuest 则管理项目的任务、缺陷和增强请求等活动，并提供图表工具来追踪项目的进展情况。
- 在实施 UCM 时，ClearCase 的 UCM Project 与 ClearQuest 的 User Database 必须关联一体，并将 ClearCase 中的 Activity 与 ClearQuest 的变更请求记录 CR 一一映射。

# 实例：ClearCase UCM工程





# 实例：产品目录结构(流划分)

- 开发（产品）目录结构的规划是软件配置管理的主要内容，由过程工程师、构架师配合配置经理来共同制定，并在配置工具的支持下予以实施；
- ClearCase Stream（流）的划分——

- PCHL_V1_Integration	整个项目所有工件的最终整合场所，用于项目级基线构建、进行集成和确认集成测试
--- PCHL_V1_Working	需求、设计、管理、过程等工作场所
--- PCHL_V1_Dev	项目所有构件的开发场所
--- PCHL_V1_Testing	测试组的测试场所
--- PCHL_V1_Release	发布演示场所

# 实例：产品目录结构(VOB划分)

- 项目目录参照RUP的工件集来组织
- Clearcase VOB的划分——

<b>Infrastructures</b>	内部开发的相对独立、复用度强的基础应用包，针对第三方产品进行封装的调用接口等
<b>Libraries</b>	从组织外部获取的源码库、开发组件等，主要来自于开放源码；
<b>PCHL_Components</b>	项目组针对目标系统开发的构建于底层开发包、基础设施之上的所有构件
<b>PCHL_SubSystems</b>	构建于底层开发包、基础设施与所有构件之上的可执行子系统（或系统），通常是目标系统的最终交付实体
<b>PCHL_System</b>	项目目标系统主体工件目录，包含需求、设计、集成、测试等
<b>PCHL_Management</b>	项目的管理工件目录
<b>PCHL_Supports</b>	项目的过程、环境、标准等支持工件目录

# 演示：开发目录结构(内容)

Rational ClearCase Explorer - ccadmin\_PCHL\_V1\_int (Z:\PCHL\_System\Design\Documents)

File View Go Tools Environment Help

General  
PCHL\_V1

ccadmin\_PCHL\_V1\_int

scts\_v1  
SCTS\_Web

ccadmin\_PCHL\_V1\_int

- My Activities
- Infrastructures
- Libraries
  - ace
  - boost
  - boost-build
  - cppunit
  - Library\_Repository
  - log4cplus
  - lost+found
  - mockpp
  - xercesc
- PCHL\_Components
- PCHL\_Management
- PCHL\_SubSystems
- PCHL\_Supports
- PCHL\_System
  - Context\_Analysis
  - Deployment
  - Design
    - Documents
    - Models
  - Integration
  - lost+found
  - Requirement
  - Runtime\_Image
  - Test

Name	Size	Kind	Modified
Design_Model_Surveys	89	View-p...	2003-12-16
Detail_Designs	91	Direct...	2003-12-16
Interface_Designs	317	Direct...	2003-12-16
UI_Designs	89	View-p...	2003-12-16
Software_Architecture_Document.txt	0	File El...	2003-12-16
Workload_Analysis_Document.txt	0	File El...	2003-12-16
通道技术方案对比分析.doc	117248	File El...	2003-12-16
证券统一通道平台结构图.vsd	373248	File El...	2003-12-16
证券统一通道平台软件构架文档.doc	1236992	File El...	2003-12-16

### UCM Dynamic Integration View

This is an integration view for the PCHL\_V1 project.

An integration stream identifies the shared versions to be used to build and test either the entire project, or a major feature of the project. The shared versions are delivered from child integration or development streams below it in the project hierarchy.

If you are a developer, use this view to see the work other team members have delivered, and to test your own work during [delivery](#).

If you are a project manager, use the integration view to:

- [Create new baselines](#) as development proceeds. Developers can then rebase to the new baselines to stay current with project changes.
- Promote or demote baselines. The promotion level indicates the quality and stability of the baseline to all developers.

Ready View: ccadmin\_PCHL\_V1\_int Activity: construct the clearcase vobs Items: 9 Selected: 0

# 实例：项目的主要文档

---

参照RUP核心工件，根据项目需要确定在项目  
中开发和使用如下文档——

- 《前景文档》
- 《补充规约》
- 《软件构架文档》
- 《XXXX详细设计文档》
- 《软件开发计划》
- 《测试计划》
- 《系统性能测试报告》
- 《系统安装与配置手册》

# 演示：源码目录结构

**Rational ClearCase Explorer - huxg\_PCHL\_V1\_Dev (D:\Shared\_Views\PCHL\_V1\_Dev\PCHL\_Components\invocation\src)**

File View Go Tools Environment Help

**General**  
PCHL\_V1

huxgShared\_PCHL\_V1\_Dev  
huxg\_PCHL\_V1\_Dev  
huxg\_PCHL\_V1\_Working

**SCTS\_V1**  
Vi... To...

**huxg\_PCHL\_V1\_Dev**

- My Activities
- Infrastructures
- Libraries
- PCHL\_Components
  - common\_facilities
  - conversation\_intf
  - invocation
    - bin
    - build
    - include
    - lib
    - src
  - lost+found
  - marshal\_supporting
  - protocols\_api
  - ucs\_impl
- PCHL\_SubSystems
- PCHL\_System
  - Deployment
  - Integration
    - bin
    - build
    - doc
    - example
    - include
    - lib
    - Plans
  - Test

Name	Size	Kind	Modif
callee_handler_base.cpp	183	File Element Version	2004
exception.cpp	541	File Element Version	2004
exception_msg_imp.cpp	1410	File Element Version	2004
execute_framework.cpp	2700	File Element Version	2004
<b>general_msg_base.cpp</b>	1204	File Element Version	2004
handler_factory.cpp	1236	File Element Version	2004
request_msg_base.cpp	907	File Element Version	2004
response_msg_base.cpp	1037	File Element Version	2004
server_factory_base.cpp	619	File Element Version	2004
server_obj_base.cpp	3688	File Element Version	2004

**Your selection**

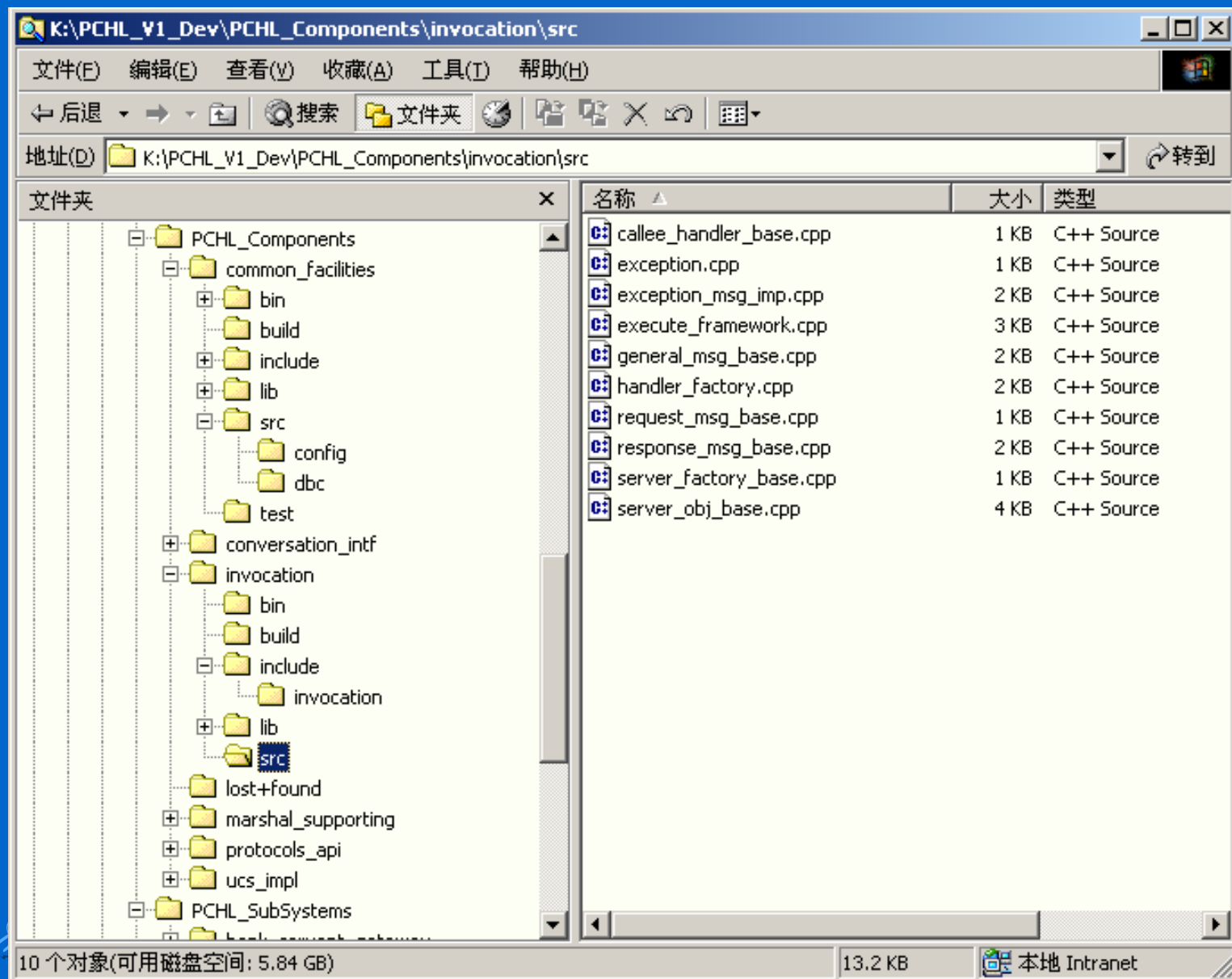
**About** **Uses**

**Name:** general\_msg\_base.cpp  
**View Tag:** huxg\_PCHL\_V1\_Dev  
**View Path:** D:\Shared\_Views\PCHL\_V1\_Dev\PCHL\_Components\invocation\src  
**UCM Project:** PCHL\_V1  
**Kind:** File Element Version  
**Modified:** 2004-01-27 11:13:42  
**Version:** \main\PCHL\_V1\_Dev\1  
**Rule:** ...\PCHL\_V1\_Dev\LATEST

**general\_msg\_base.cpp** is a versioned file under ClearCase source control.

Ready View: huxg\_PCHL\_V1\_Dev Activity: refactor the System directory structure Items: 10

# 示例：开发构件构建方案目录





# 示例： 目录说明

目录	说明	备注
build	构建脚本以及IDE项目文件	纳入配置管理
include	构件的公开头文件（即Interface）目录，依赖它的构件将通过设置环境变量来增添一条指向它的头文件包含查找路径	纳入配置管理
src	构件的源代码，包括私有的头文件	纳入配置管理
doc	用户参考文档	纳入配置管理
test	验证构件是否成功编译、安装的测试代码	纳入配置管理
lib	构件最终提供给项目其它构件引用的静态库和动态库	通常由构建脚本在编译步骤生成
bin	可执行文件	通常由构建脚本在编译步骤生成



---

# Q&A

# 谢谢！