ClearCase 系统管理员手册

Operation Guide

目录

1 引言

在软件开发过程中的变更是不可避免的，而变更更加加剧了项目中软件工程师之间的协调难度和混乱。

配置管理（CM）是运用技术、行政的指令指定和控制以下内容的一种学科：对某个配置项的功能特性和物理特性进行标识并形成文件，对这些特征的更改进行控制，记录并报告更改过程和实施状态，以及验证对规定要求的符合性。[IEEE—STD—610]。软件配置管理(SCM)是贯穿于整个软件过程中的保护性活动。因为变化可能发生在任意时间，SCM 活动应该包括动态的标识变化，控制变化，保证适当地实现这些变化，以及向其它相关的人员报告变化的情况。

软件配置管理涉及在给定时间点，或者在变更发生时，动态的标识并报告软件的配置状态，系统地控制对配置的更改并维护在整个软件生存周期中配置的完整性和可跟踪性。软件配置管理的任务包括识别那些应该纳入到软件配置管理之下的软件配置项，建立起相应的软件基线和基线库，通过软件配置管理的更改控制和配置审核功能，系统地控制基线的更改和那些根据软件基线库构造的软件产品的发放。

1．1 目的

指导 SCM 人员应用 Rational 公司的软件配置管理工具 Clearcase 对项目的 SCM 活动进行策划和实施，从而高效的完成项目的 SCM 活动，同时为公司的软件过程数据库积累原始资料。

1．2 适用范围

本指南文件规定了在软件项目组应用 Clearcase 实施软件配置管理时应该遵循的统一的基本要求，适用于全公司范围内的软件工程项目的 SCM 活动，但是具体项目在实施软件配置管理时，要结合具体项目的特点对本程序文件的条目进行裁剪得到其子集，也可以在此基础之上再添加一些新特性，以适用于具体项目的要求。

1．3 引用标准和参考资料

国际标准

Capability Maturity Model [SM] for Software Version 1.1

国家标准

GB/T 11457 软件工程术语

GB 8566 计算机软件开发规范

GB 8567 计算机软件产品开发文件编制指南

GB/T 12504 计算机软件质量保证计划规范

公司标准

QP309 软件配置管理

QR102-03 评审总结及评审记录

QR309-03 软件变更通知单

QR307-01 软件问题报告

QR401_03 会议记录

QR501-03 文件变更申请

参考资料

Roger S. Pressman McGraw-Hill Software Engineering a Practitioner’s Approach

Fourth Edition

软件工程指南	朱三元等编译	上海翻译出版公司
实用软件工程	郑人杰 殷人昆 陶永雷	第二版 清华大学出版社
软件工程导论	张海藩	第三版 清华大学出版社
软件工程（高级）	郑人杰	第一版 清华大学出版社

1．4 术语及解释

软件生存周期 SLC: Software Life Cycle 同任何事物一样，软件也有一个孕育、诞生、成长、成熟、衰亡的许多阶段，一般称其为计算机软件的生存期。根据这一思想，把上述基本的过程活动进一步展开，可以得到软件生存期的 6 个阶段工作，即制定计划、需求分析、设计、程序编制、测试及运行维护。（《软件工程（高级）》郑人杰 1999.8 第一版）

软件配置项 SCI: Software Configuration Items 是软件配置管理的对象，指的是软件工程过程中产生的所有信息项。包括计算机可执行的源代码、目标码、数据库等以及计算机不可执行的文文件、源程序清单、测试用例等。（《软件工程（高级）》郑人杰 1999.8 第一版）

软件配置对象 SCO: 在每一个软件项目的 SCM 计划中要对本项目中所有可能的软件工程元素从它的重要性以及和其它软件工程元素之间的关系等方面进行综合的考虑，判断是否应该标识为软件配置项 SCI；在确定本项目所有的软件配置项之后，在实现软件配置管理的活动中，用面向对象的方法组织软件配置项 SCI，将 SCI 按逻辑关系组织成软件配置对象 Software Configuration Objects，每个 SCO 都有自己的标识，属性，并通过关系和其它对象联结。软件配置对象有基本的和聚集的两种类型，都将被归类到项目数据库中。（QG309_01V1.2）

软件配置对象标识 SCOI: 为了控制和管理软件配置项 SCI，每个软件配置项都必须被唯一的命名，Software Configuration Objects Identification 的工作就是分析所有这些软件配置项之间的逻辑关系，然后用面向对象的方法将 SCI 组织成软件配置对象 SCO，并为每个 SCO 详细的填写一组唯一地标识它的独特的特征属性：名字、描述、资源表、以及无二义性地标识该对象的一个字符串。（QG309_01V1.2）

软件基线 SB: Software Baseline 是软件生存周期中各开发阶段末尾的特定点，也被称为里程碑（milestone）。他的作用是把各阶段的开发工作划分得更加明确，使得本来连续的工作在这些点上断开，使之便于检验和确认阶段

开发成果。它对变更控制起的作用是，限制不允许跨越里程碑去修改另一阶段的文档。（《软件工程（高级）》郑人杰 1999.8 第一版）

软件基线库 SBL: Software Base Library：用以存放配置项和相应的记录的的仓库的内容。（cmm v1 源文件中文）

软件配置控制委员会 SCCB: Software Configuration Control Board 负责审批配置项更改建议，和确保已批准的更改实施的组。（cmm v1 源文件中文）

版本控制 VC: 软件配置管理应该允许用户选择适当的版本来确定软件系统的配置情况，这可以通过将一些属性结合到各个软件工作产品的某一个特定的版本上，再通过描述所希望的属性集合来确定（或构造）所想要的配置，如为了驱动单色和彩色显示，可能开发了两个不同的显示驱动模块，在构造一个实际系统时就可以根据需要和软件工作产品的版本信息来选用合适的软件工作产品。Version Control 就是对软件开发的各个阶段的软件工作产品的各种版本进行管理和控制，保证软件开发人员所选用的软件工作产品的版本是正确的且是最新的，以避免不必要的错误发生，提高软件生产率。（QG309_01V1.2）

变更控制 CC: Change Control，变更控制就是要把变更严格的控制起来，随时保留变更的有关信息，把精确、清晰的信息传递到开发过程的下一活动或下一任务去，防止出现混乱。（《软件工程（高级）》郑人杰 1999.8 第一版 P343）

软件配置管理 SCM: Software Configuration Management，软件配置管理是标识和确定系统中配置项的过程，在系统整个生存周期内控制这些项的投放和更动，记录并报告配置的状态和更动要求，验证配置项的完整性和正确性。《GB/T 11457（1995）软件工程术语》

版本对象库 VOB: Version Object Base， VOB 是存储文件、目录和原资料等版本对象的永久性资料仓库（加入到 Clearcase 控制下的文件和目录称为元素，每个元素的一个 Check in 版本成为一个“版本”）；

分支 Branches：分支描述了元素的树状版本结构，一个分支指定了一个元素的线性

版本序列；每一个元素都有一个主分支 main，代表着开发的主线，它可以有多个代表独立开发线的子分支 subbranches，并且子分支还可以再有子分支。

标签 Label：标签是用户自定义的，可以附到一个版本上的名字。标签是一种强大的项目管理和系统集成的工具。在软件开发生命周期中的给定的点上，对成组的元素应用标签，可以起到定义和保护文件/目录版本集之间的相互关系。

属性 Attributes：属性是一组由“名称=值”组成的资料对儿，使得可以从另外一种角度获取版本的状态信息。能用来注释一个版本、一个元素，一个 VOB，或一个超链的一个名字或值；Attribute 值有特定的范围和单独的类型。

超链 Hyperlinks：超链允许你定义和保护一个 **VOB** 之内，或者多个 **VOB** 之间的两个元素的关系。这种能力可以实现对程控的需求，例如可以把源文件链接到需求或者设计文档上，并以此追溯文件间的参考一致性。

触发器 Trrigers：触发器允许通过事前或事后执行特定的程序或者可执行脚本，控制 Cleartool 和其它 Clearcase 操作的行为。实际上任何修改元素的操作都可以激发触发器，特殊的环境变量使得相关的信息对这些脚本或者陈程序可用。与其它元数据类型不一样的是，触发器除了存储资料外，还可以定义活动。

锁 Lock：对一个元素加锁，限制除了“Exception”列表外的任何其它用户进行修改。加锁机制对进行临时限制非常有用。例如在集成阶段，可以通过对主分支 main 加锁，拒绝非集成组的成员进行修改。锁的作用范围可大可小，即可以作用到整个 VOB 上，也可以只对某个对象加锁。

1．5 角色与职责：

Clearcase Administrator（Clearcase 系统管理员）：又称配置管理员，其域用户是 ccadmin，制定配置管理策略，实施配置管理活动，同时也是域用户管理员。职责如下：

- 制订基于配置管理工具的开发策略；
- 定制视图模板（View Profile）和配置规则（Config Spec）；

- 定义域用户和组，设置用户权限；
- 执行锁定/开锁（Lock）；
- 给元素或元素组打标签（Label）；
- 定期备份 VOB 库；
- 解决项目组成员日常使用中遇到的工具问题；
- 使用 Clearcase 执行程序文件 QP309 所规定的活动。

Project Manager（项目经理）： 对 Clearcase Administrator 的活动负责，控制进度，版本发布，变更控制等活动。

Leader（业务主管）：负责审查其下属的工作产品。

Member（项目成员）：在预定义的视图中对元素执行检出/入（check out/ in）操作，管理私有分支，完成单元测试。

2 配置管理库结构：

参见 RUP。

3 权限分配策略：

- Clearcase 系统管理员在 PDC 上定义项目的用户和组（pm, reqer,designer,coder, tester），每个组中都包含项目经理（pmm），SQA（sqa），SCM（scm）；如果使用 Windows/Unix 混合平台，则需要在 Unix 服务器上创建同样的用户和组，并把这些用户和组作为相关目录的 owner 和 primary group；
- Clearcase 系统管理员使用 ccadmin 为项目创建 VOB 和 View，以及 VOB 中的目录结构；
- Clearcase 系统管理员使用 protectvob 命令定义 VOB 的权限，设置 owner（-chown）：Clearcase 系统管理员的域账户，设置主组（-chgrp）为 pm 组，把所有属于此项目的组加入到组列表中 pm, reqer, designer, coder, tester，（-add_group）；
- Clearcase 系统管理员使用 protect 命令定义元素（文件元素与目录元素）的权限，设置 owner（-chown）：业务主管，设置主组（-chgrp）为业务组，VOB 根目录的权限模式（-chmod）为 770，之下的目录的权限模式（-chmod）为 775；
- Clearcase 系统管理员对已发布的配置项元素加锁（lock），防止别人修改。
- Clearcase 系统管理员定义 View 的配置规则（Config spec），项目成员在工作视图中执行 Check out 时自动生成私有分支，并在此分支上完成工作任务； 使用基线视

图获取所需的稳定配置项；

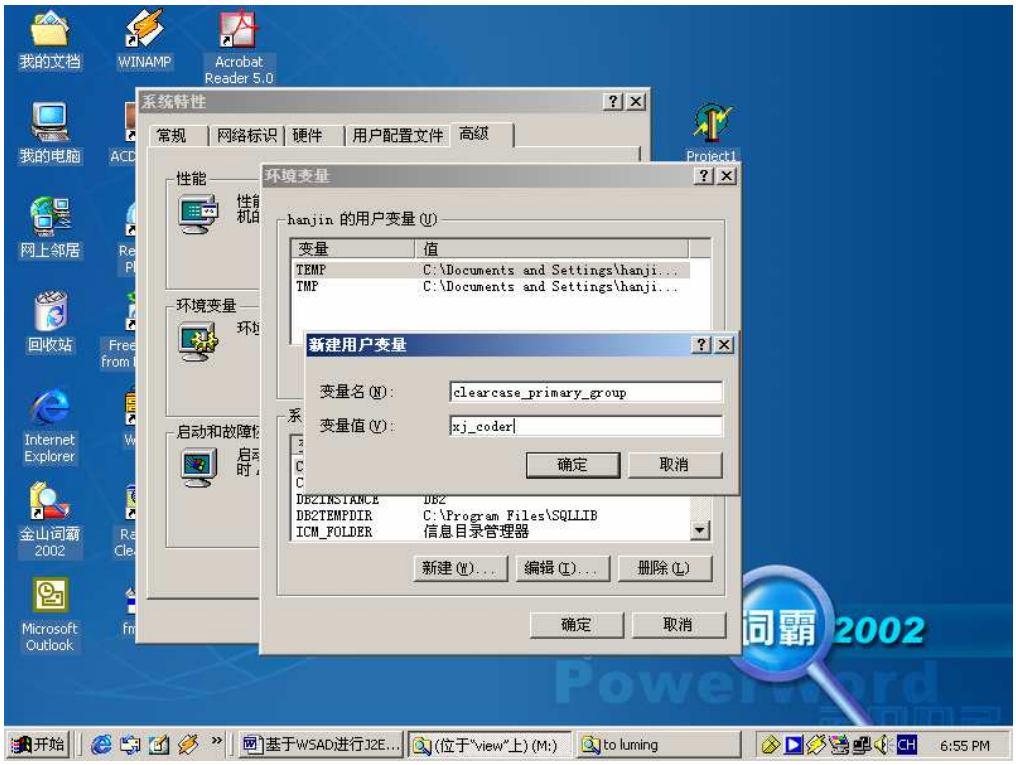
4 准备活动：

4．1　安装和配置服务器：

使用 Clearcase 进行局域网内的配置管理活动的必要前提就是在一台 PDC（主网网域控制器）上完成安装和配置。包括以下关键活动：

- 安装服务器：在服务器（其硬盘经过 License 解密）上安装 Win 2000 Server 或者 Win NT，或者 Unix（Solaris）作为我们的 VOB Server, View Server, Registry Server。
- 配置 PDC：把服务器配置成主域控制器（Primary domain control）。
- 创建账户和组：使用“开始>管理工具>Active Directory 用户和计算机”菜单，创建 ccadmin 账户，添加到 domain admins 组中；创建使用本服务器的项目组用户和组，如 pa 组；使用 Unix 作为 VOB Server 和 View Server 而客户端使用 Windows 时需要在 Unix 上创建同样的用户名和组名并作为 cc-storage 的 owner 和 Primary group，并且客户端要使用 NFS 来进行混合环境下的通信。
- Server 预安装：指定 VOB Server, View S erver, Registry Server；并选择不允许远程管理该计算机（Do not allow any user to administrate this computer remotely）项，以确保服务器的系统安全性；选择所有要在开始菜单中显示的子菜单选项（没有在菜单中显示的功能可以在安装目录下找到可执行文件，如 c:\ program files\ rational\ clearcase\ bin\ …）。请统一设置用户 clearcase_albd 的密码为“albd”以便于进行管理。详情请参考本文附录“4。1 预安装”部分内容。
- Server 安装：选择安装 Server，输入合法的 License，或者在此处跳过，安装完成后可以到控制面板中 ClearCase 服务的 License 页中添加。成功完成安装后，开始菜单中已经包含了 Rational ClearCase Administrator 和 Rational ClearCase 菜单。安装完成后重新启动。详情请参考本文附录“4。2 Server 安装”部分内容。
- Client 安装：项目成员把本机注册到预定义的域中，使用域账户登陆完成 Client 安装；安装完成后重新启动。
- 启动服务：安装完成后确认控制面板中的 ClearCase 的所有服务都已启动。如果 clearcase_albd 没有启动，重新输入 clearcase_albd 用户的密码为安装时定义的密码，然后使用此域的 ccadmin 用户（如：SCM\ccadmin）登陆，重新输入服务 Atria Local Broker 的属性中的登陆账户（域名\clearcase_albd）的密码，启动服务，控制面板中的 clearcase\service startup 中的 ALBD service 就也启动了。如果仍不能启动服务，请查看《ClearCase Technotes》，如果仍不能解决，请详情请参考本文附录“4。3 Client 安装”部分内容。

- 设置 Registry 密码：运行安装目录中 rational\clearcase\bin 目录中的 rgy_passwd.exe 文件，即弹出命令行窗口，带有提示符“password>”，输入您的 Registry 密码即可。
- 设置环境变量：在服务器和客户端均须设置环境变量 clearcase_primary_group；服务器端其值设置为 Domain admins，客户端设置为项目组成员所属的域用户组，如 CHPL_Coder 组。在“我的电脑/属性/高级/环境变量”中新建变量 Clearcase_primry_group，如下图：



- 规划 Storage：包括 VOB storage 和 View storage；使用开始\程序\Rational clearcase administrator\Server storage wizard 菜单指定驱动器，自动创建共享给 everyone（完全控制）的文件夹 clearcase_storage，策划具体项目时使用 Protectvob 命令继续设置其中元素的共享权限；也可以由 Clearcase 系统管理员直接在服务器的资源管理器上创建共享文件夹作为 clearcase_storag。客户端只需要创建 view storage，还可以修改共享的人为自己的域用户。

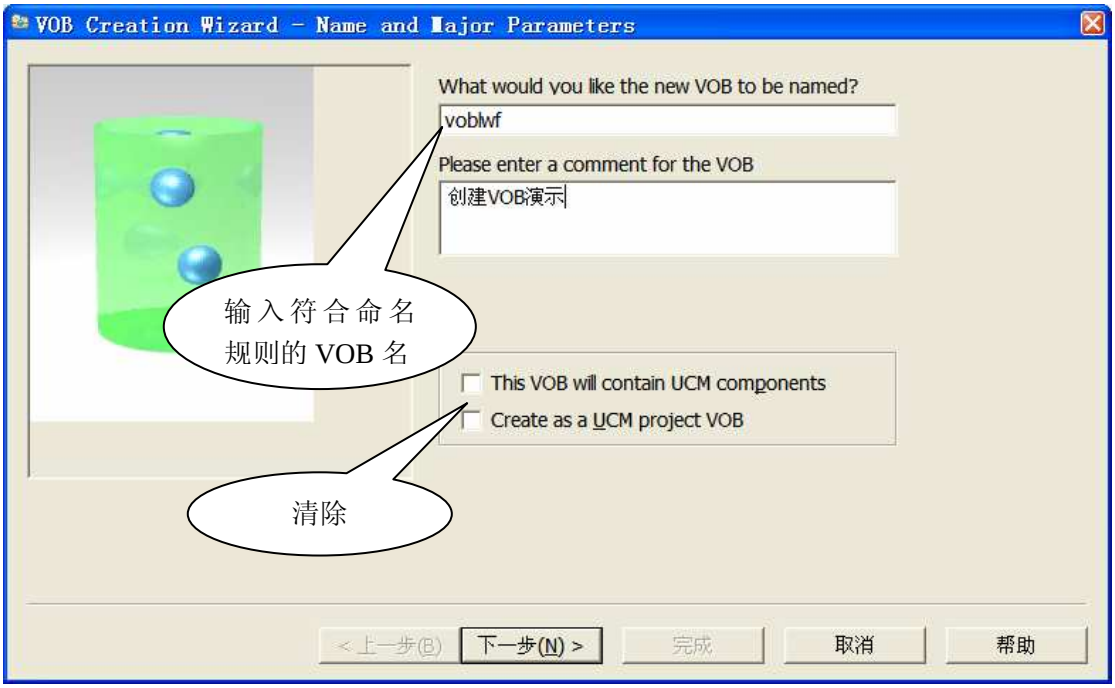
4．2 策划项目策略：

本部分描述使用配置管理工具 Clearcase 进行开发之前需要进行的策划和设置工作。

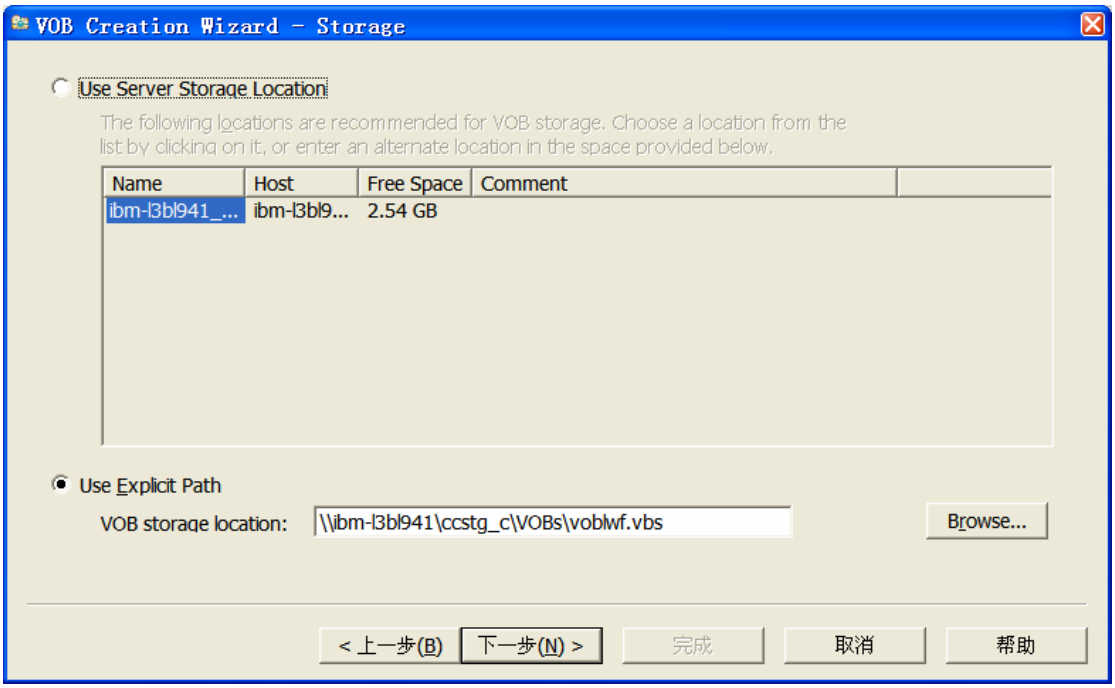
4.2.1 VOB 策略：

- 功能介绍： VOB 是存储版本对象的资料池，同一个 VOB 内的资料可以进行横向比较。

- 应用策略：我们规定为每一个项目创建一个 VOB，不管服务器使用什么操作系统。创建项目 VOB 时，选择使用本文附录中的Administration VOB。
- 命名规则：_项目简称_VOB，在前缀_和后缀_VOB 之间使用项目的缩写。
- 实现步骤：使用开始〉程序〉Rational Clearcase administration > Create VOB 打开向导，输入 VOB 名，清除 “Create as a VOB level component…”前的复选框，点击“下一步”;

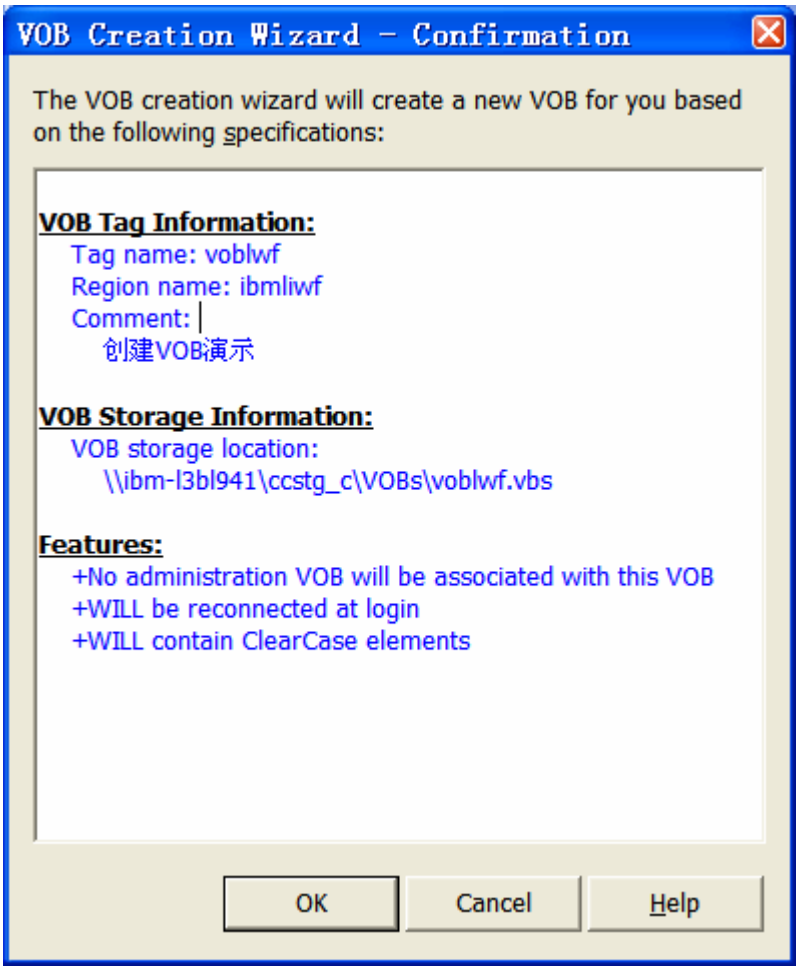


使用右下角的“Browse…”按钮从网络上定位到 VOB Storage，点击“下一步”;

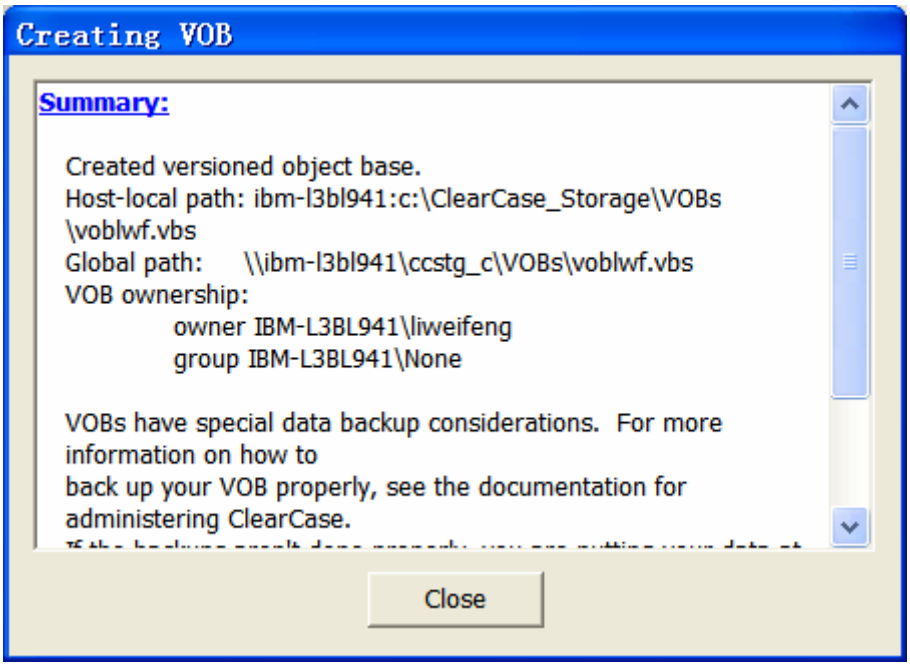


在“What is the administrative VOB that is associated with this VOB”下拉列表中选择“none”，切记不要选择“Make this a public … ”前的复选框，点击“完成”；

将会弹出确认对话框，列出 VOB 的基础信息，如下图，点击“OK”；



确认后，Clearcase 执行创建过程，完成后弹出下框：



由 Clearcase 系统管理员负责创建和维护 VOBs，但是不包含资料导入。

注：其它与 VOB 相关的操作请参考附录 1“管理 VOB”

4.2.2 分支策略：

- 功能介绍：Clearcase 使用分支策略支持并行开发。分支策略受项目的开发对象的影响，同时提供了一种控制代码库变革的机制。有多少组织使用 Clearcase 就有多少种分支策略，但其相似之处就是都反映了“坚持最佳实践”的思想。比较通用的三种分支是任务分支、私有开发分支和集成分支。

- 应用策略： 为了便于操作，初期我们只使用三种分支类型：main， test 和 develop。main 支由元素的系统管理员管理，其上只保存 merge 后的发布和基线版本；test 支由业务主管管理，负责把待评测的元素 merge 到 test 分支上，评测人员通过评测视图从此分支上获取正确的元素版本；开发人员在 develop 分支上进行工作。Clearcase 系统管理员预定义工作视图的配置规则，使得项目成员在工作视图中第一次执行 check out 时自动生成 develop 并在其上完成自己的工作，成熟后由 owner（业务主管）执行归并（merge）操作。

注：如果项目要自定义分支，需要在项目配置管理计划中分析原因及自定义结果。

- 命名规则：分支名字的格式如下：develop。分支的名字说明分支的作用，不能使用分和标签（Label）类型相同的名字。为了便于区别，我们规定分支的名字一律采用小写字母和数字符元，而标签名字全部采用大写字母和数字符元。

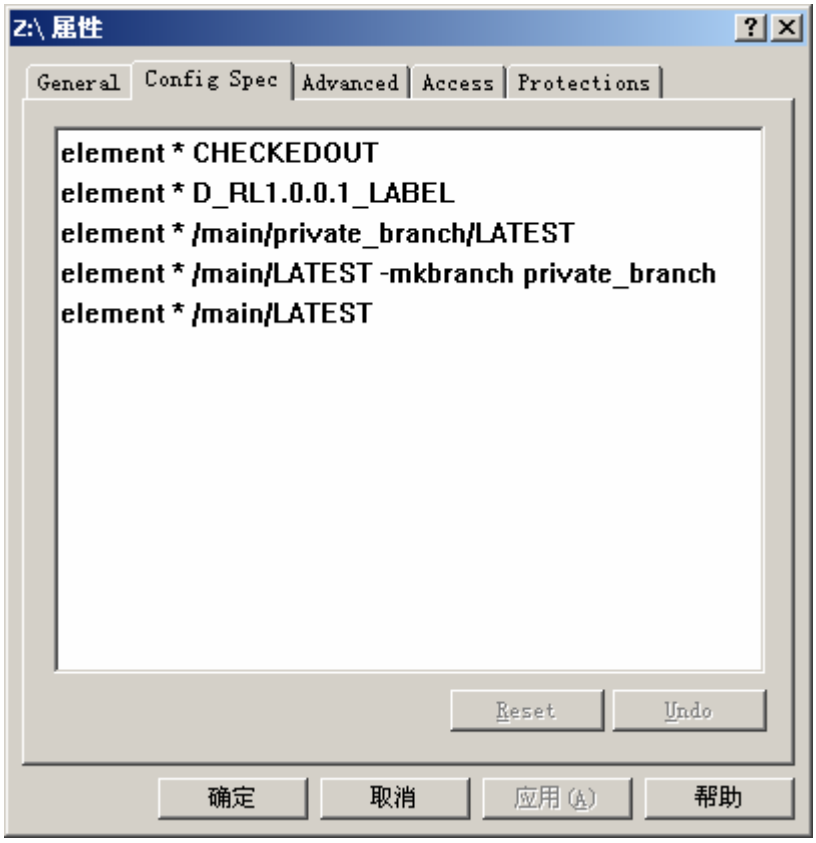
- 实现步骤：

使用 Type Explorer 创建分支类型：

使用-mkbranch 命令创建一个分支实例。有两种方式，可以在视图属性的配置规则（config spec）页面中编辑，如下页图所示。

也可以使用 Cleartool > mkbranch 命令行的方式实现，语法如下：

mkbranch [-c.omment comment | -cfi.le comment-file-pname |-cq.uery | -cqe.ach | -nc.omment] [-nwa.rn] [-nco] [-ver.sion version-selector] branch-type-selector pname ...



4.2.3 视图策略：

- 功能介绍：访问 vob 里的元素必须通过 View 进行。为客户端提供了一个工作的私人空间。每个 View 都有一系列的规则叫做 Configuration Specification（Config Spec），它可以选择所指定任务的每一个文件或目录的适当版本，并呈现它们。一个客户端可以创建多个 view，如用于开发的 view、用于改错的 view 等。

- 应用策略：Clearcase 系统管理员预先定义视图的配置规则（config spec）或者是配置规则的格式（可参考附录 3 配置规则），以及详细的说明文件，使得项目成员可以根据需要创建自己的个人工作视图（快照视图或者动态视图），在其中既能够阅读到其它元素的发布或基线版本，同时也能定位在自己的 develop 上（附录 3 实例的最后一个）； Clearcase 系统管理员在评测时创建公用的评测视图（块昭视图），使得评审/测试人员在最新版本的评测视图中能够读取需要评审/测试的配置项的适当版本；Clearcase 系统管理员在生成基线时创建公用的发布视图（快照视图），使得项目成员在最新版本的发布视图中能够看到最新的可用已发布配置项；Clearcase 系统管理员在生成基线时创建公用的基线视图（快照视图），使得项目成员在最新版本的基线视图中能够看到最新的可用基线。有变更发生时，Clearcase 系统管理员对需要变更的配置项打上变更标签（change label），生成变更分支（change branch），从而生成变更视图（动态视图），使得变更执行人员在变更视图中完成变更的执行。

注：如果项目自定义了分支，则需要在项目配置管理计划中声明该项目使用的视图策略。

- 命名规则：Owner_Type_view，第一部分代表在此视图类型，第二部分说明此视图的版本（记录配置规则的修改历史），加上后缀“_view”说明类型。
- 实现步骤：

工作视图配置规则：选择相应所有元素的最新版本，执行 check out 时自动生成私有分支；其中非此角色工作区的配置项不具备参考价值，请使用基线视图，格式如下：

element * CHECKEDOUT

element * D_RL1.0.0.1_LABEL

element * /main/test/develop/LATEST

element * /main/test/LATEST －mkbranch develop

element * /main/LATEST –mkbranch test

element * /main/LATEST

基线视图配置规则：使用基线标签选择配置项版本，如 ：

element * baseline label;

评测视图配置规则：使用评测标签选择配置项版本，格式同上；

发布视图配置规则：使用发布标签选择配置项版本，格式同上；

变更视图配置规则：使用变更标签选择配置项版本，格式同上；

具体操作请参考附录 2；配置规则语法及已有的、可供参考的配置规则，请参考附录 3。

4．3 其它策略：

为了强制应用开发策略，可以创建一些元数据（Metadata）类型来保护版本的状态信息。为了监控项目过程，可以从元数据类型和事件纪录中捕获的信息生成各种各样的报告。

包括 attributes, hyperlinks, triggers, and locks，统一元数据使用规则如下。

4.3.1 标签（Label）：

- 功能介绍：在软件开发生命周期中的给定的点上，对成组的元素应用标签，可以起到定义和保护文件/目录版本集之间的相互关系。
- 应用策略：统一定义如下标签类型：

标签类型	Type 名称	应用时机	格式	执行人

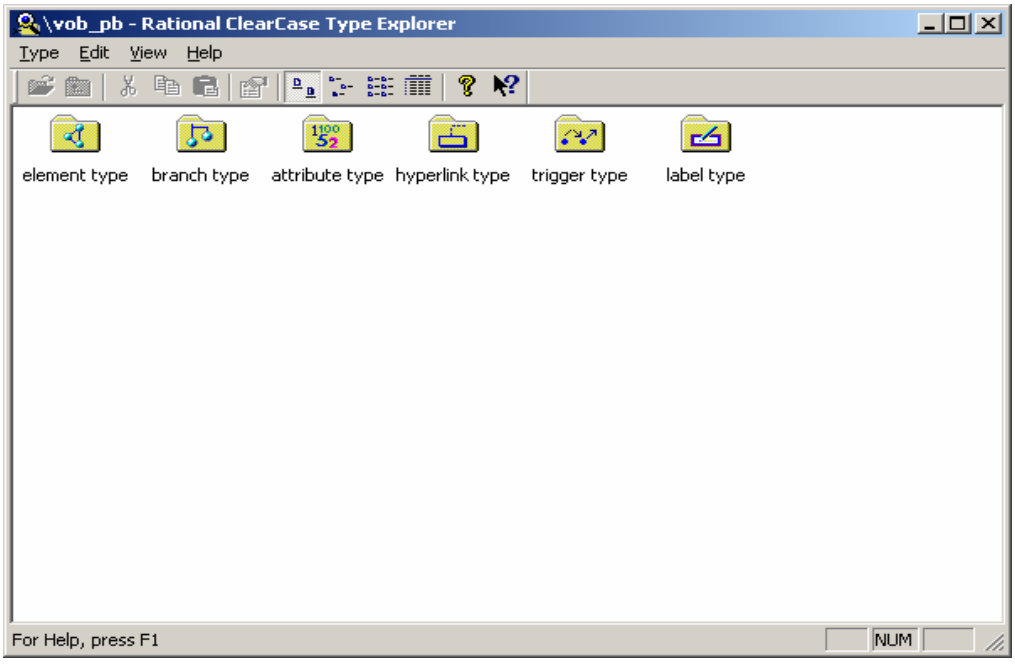
评测标签	Checkpoint	送评/测前	-项目简称-Name_ Type_Ver	配置管理员
发布标签	Release	发布时	-项目简称-Name_ Type_Ver	配置管理员
基线标签	Baseline	生成基线时	-项目简称-Name_ Ver	配置管理员
变更标签	Change	变更提出时	-项目简称-Name_ Type_Ver	配置管理员
增进标签	Enhancement	增进功能提出时	-项目简称-Name_ Type_Ver	配置管理员

- 命名规则：我们规定使用大写字母和数字符元表示标签的名字，格式是：TYPE_Version_ 生成日期，“type”指的是上面描述的 4 种标签，即有 4 个可选项：CP（Checkpoint），RL（Release），CH（Change）,EM（Enhancement），其中 Version 表示版本号（请参见版本控制）；Name 指的是基线：A(ssign), PP(Project Plan), R(equer), D(esign), C(ode), T(est), I(ntegration), P(roduct)。

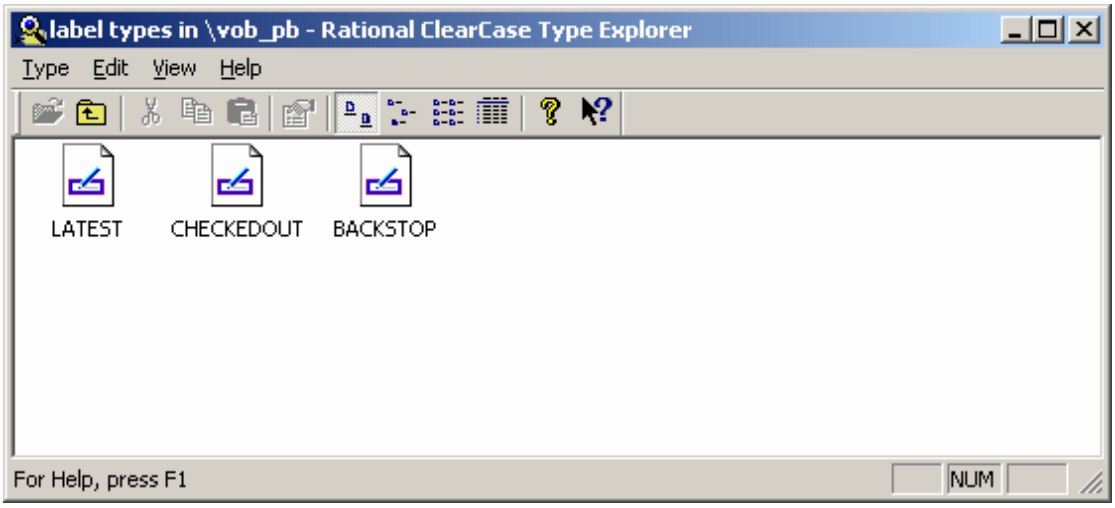
- 实现方法：

a）创建一个标签类型

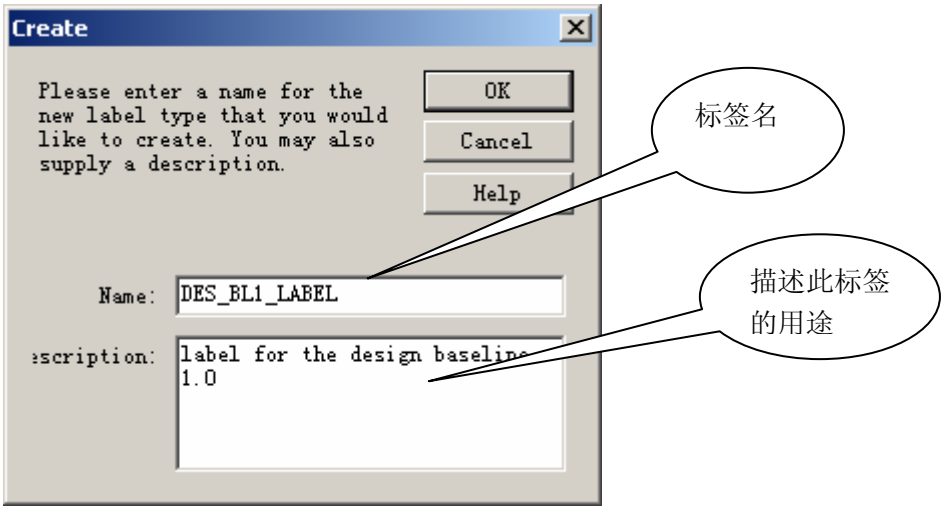
定位到 VOB，快捷菜单打开



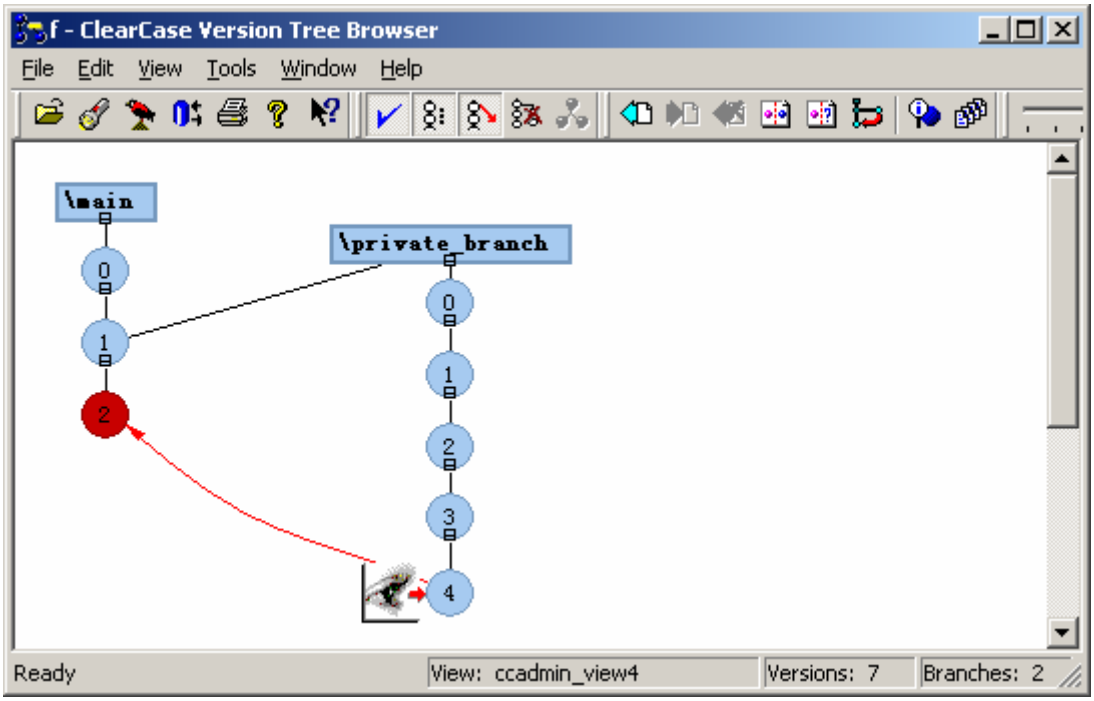
双击 Label type



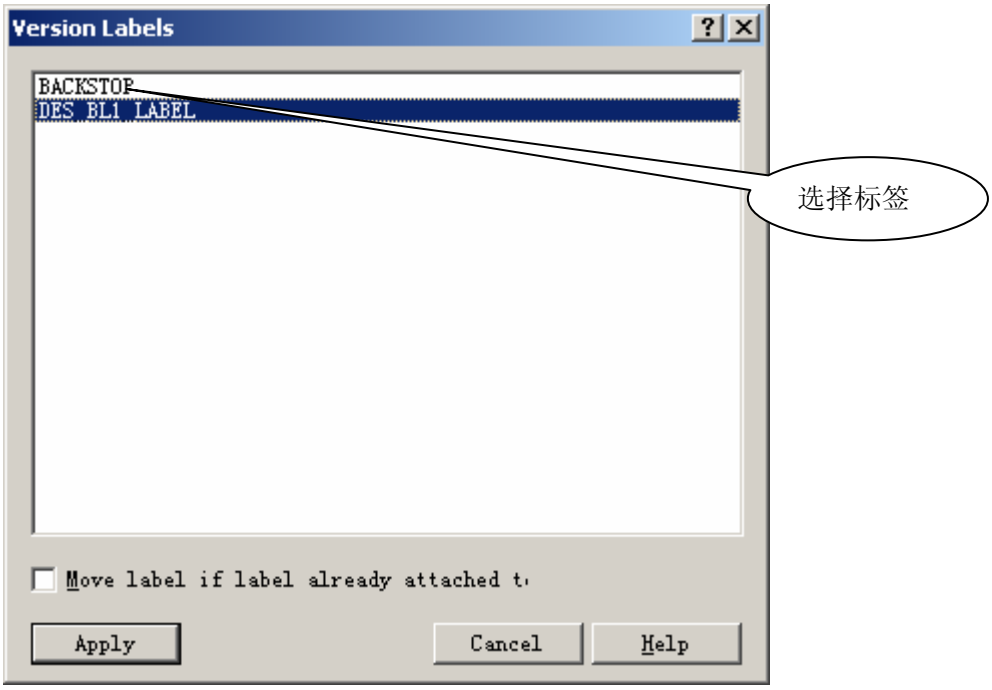
击菜单 Type->Create

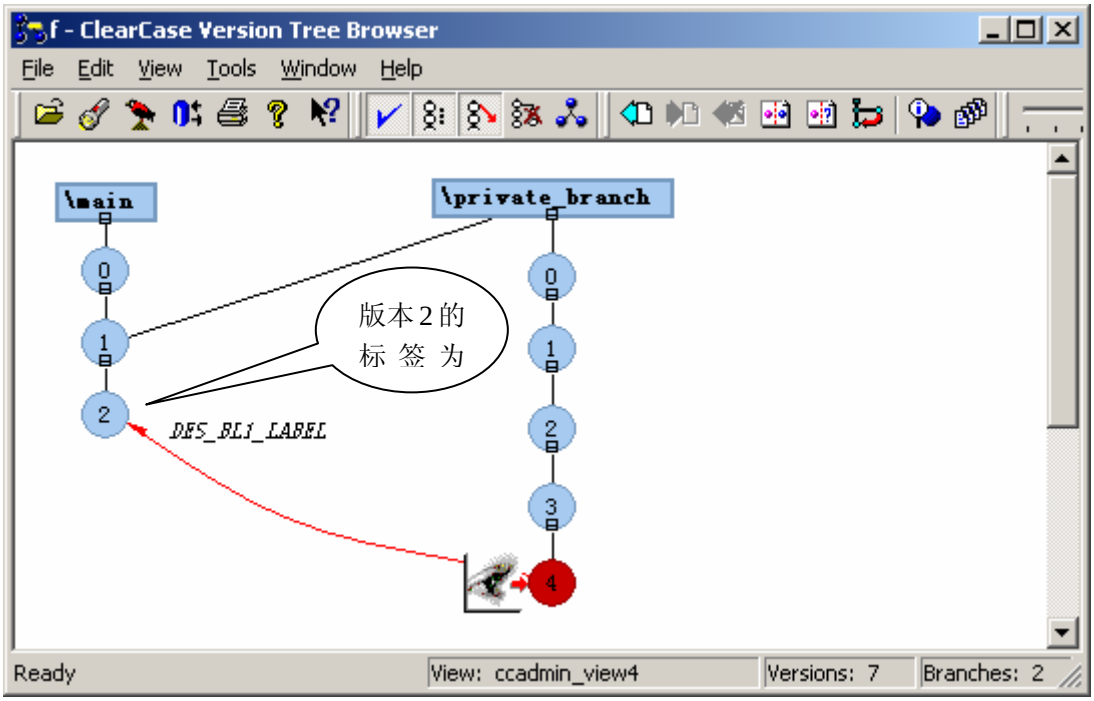


b）为某个元素的一个版本打标签



在选定版本上击右键，选择弹出菜单上的“apply label”命令：





4.3.2 属性（Attribute）：

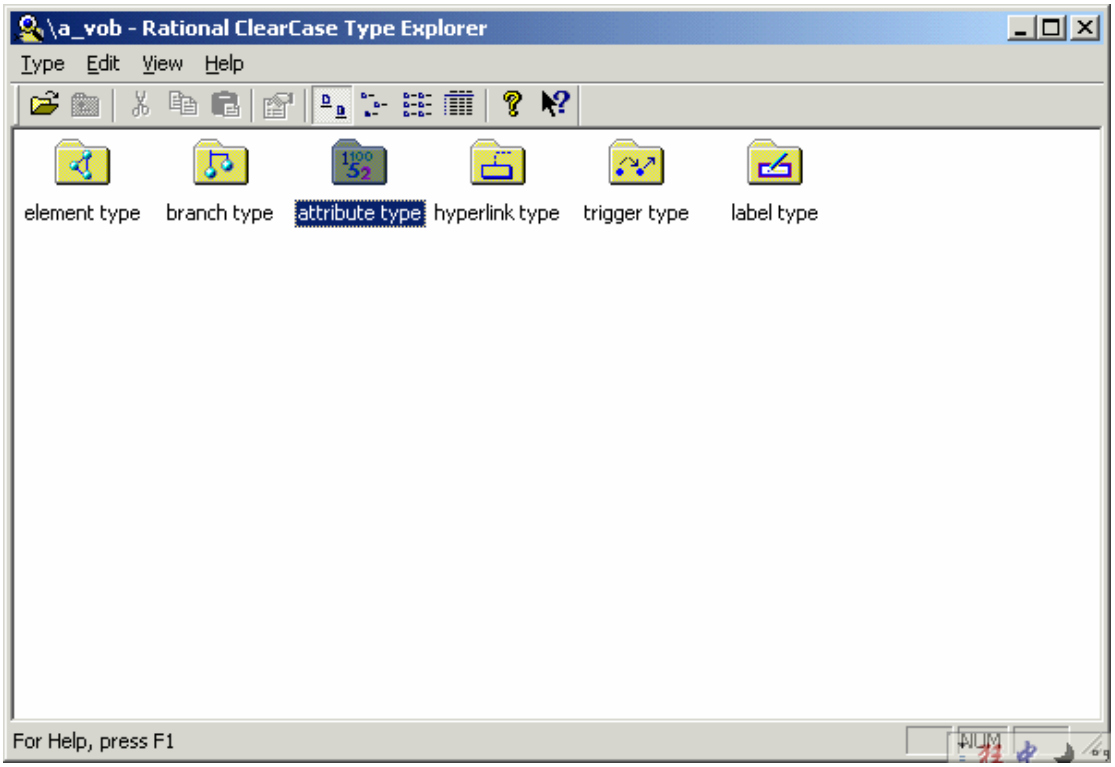
- 功能介绍：属性是一组由“名称=值”组成的资料对儿，使得可以从另外一种角度获取版本的状态信息。Attribute 应用：配置规则：例如 element * \{QAstat＝＝“FALSE”} VOB 搜索：例如 cleartool find . －element attype（BugNum） －print。
- 应用策略：预定义以下属性，要求各业务主管审查其成员的工作产品时公正客观的应用每一个属性。项目组可以根据需要提出自己的 Attribute 变量。CC 系统管理员搜集过程数据库的需求，逐步完善

属性名称	应用时机	用途	取值	
CommentDensity	代码审查	说明代码的注释情况	Unacceptable	不合格
			Low	差
			Medium	中
			High	优
Regularity	规范性	说明文档使用模板的情况	Unacceptable	不合格
			Low	差
			Medium	中
			Standard	标准的
Pages	页数	说明文档的页数多少	5	0-5
			10	6-10
			20	11-20
			40	21-40
			〉40	40 页以上

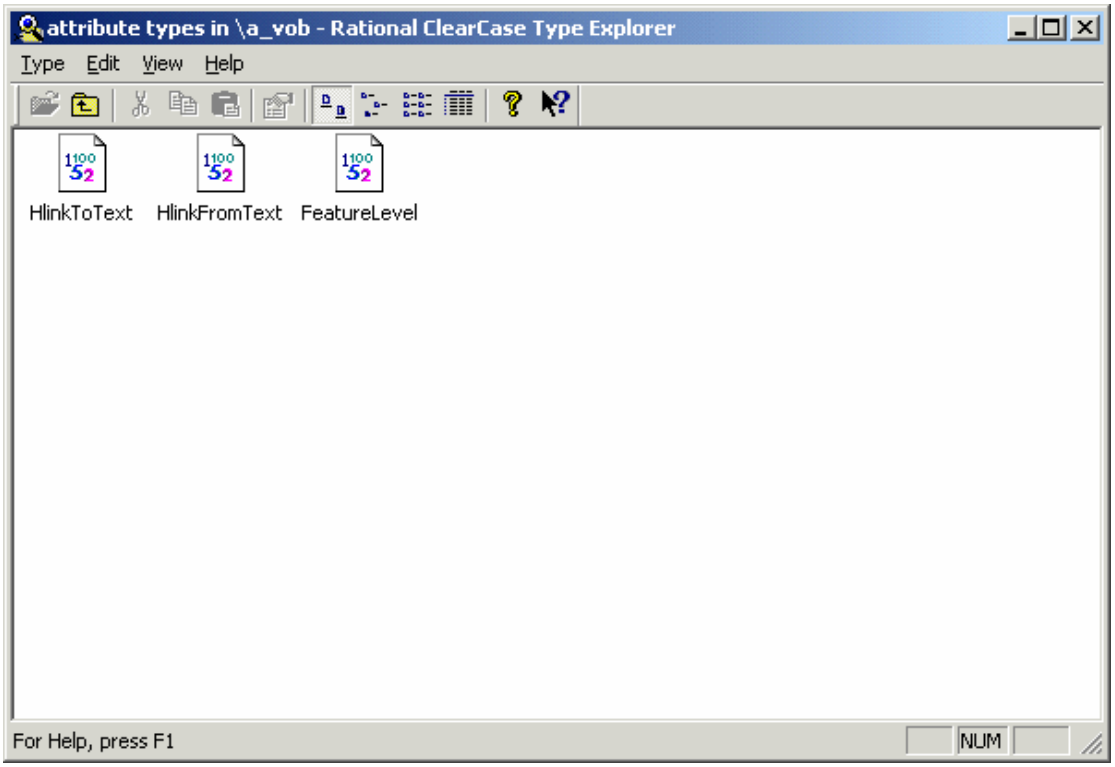
- 命名规则：属性名中每个有具体含义的字段元元的第一个字母采用大写，其它字母使用小写，可取值的第一个字母采用大写。
- 实现方法：

a）建立一个 Attribute　Type

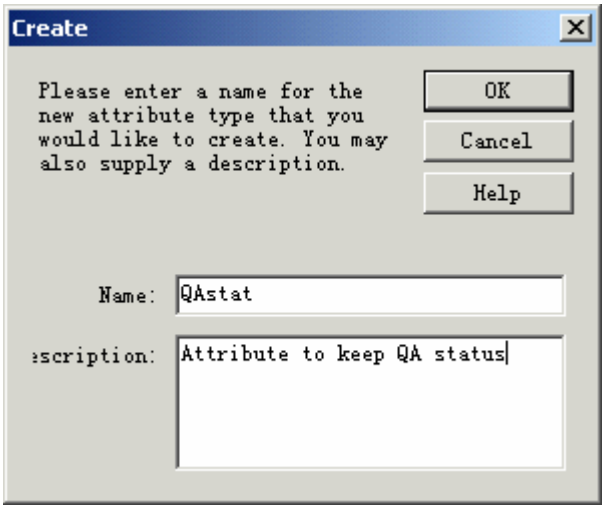
打开 Type Explorer



双击 Attribute Type



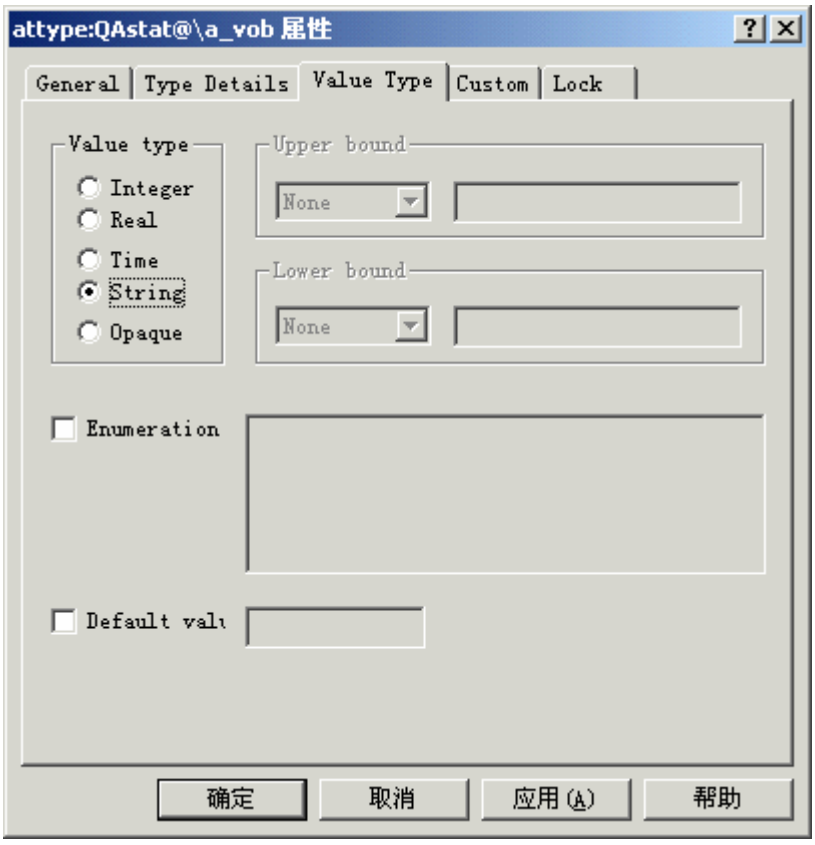
选择菜单 Type>Create



点 OK。

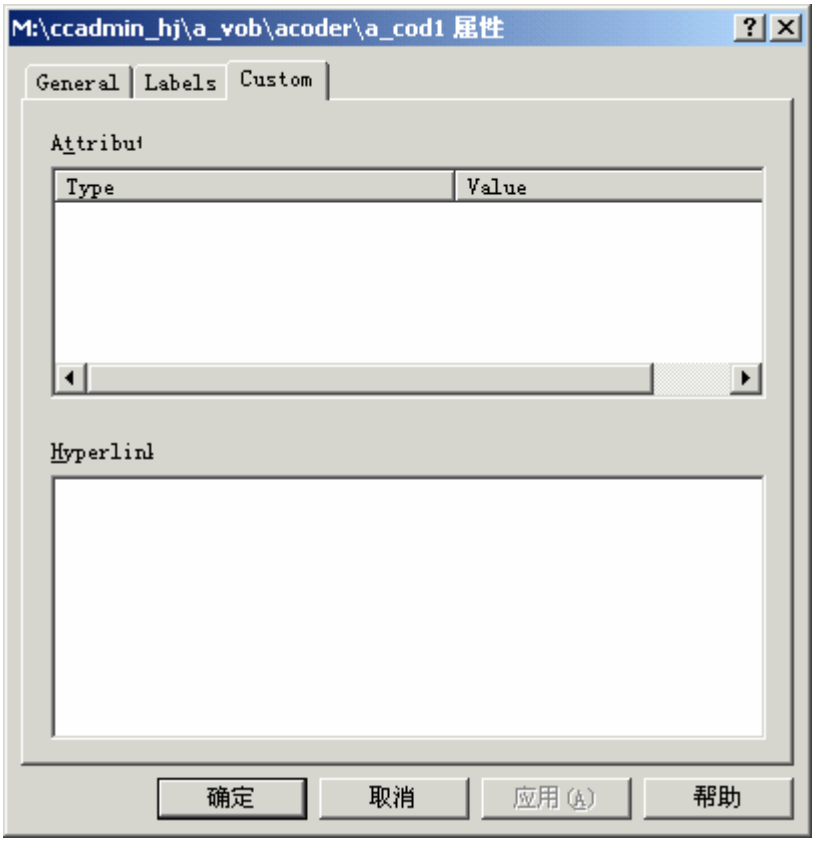
b）制定 attribute 值类型

在属性页中，例如在 Value Type 中选择“String”，点 OK.

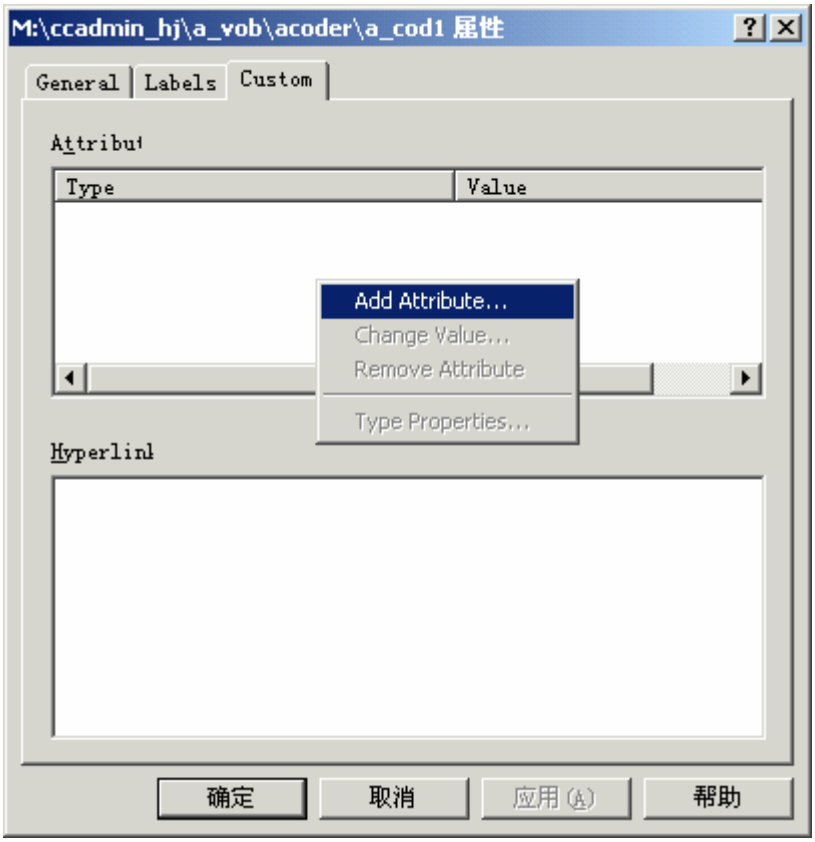


c）Attaching Attributes

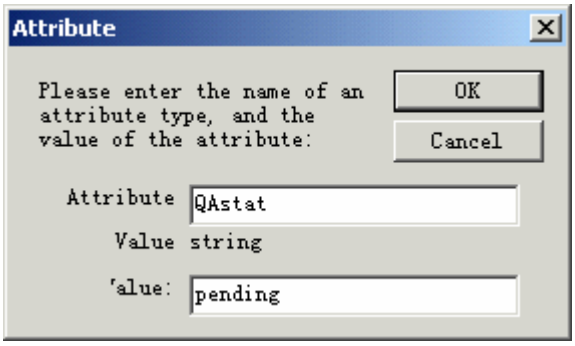
在 ClearCase Explorer 中选择一个活页夹或文件，打开 Properties of Version，选择 Custom 属性页。

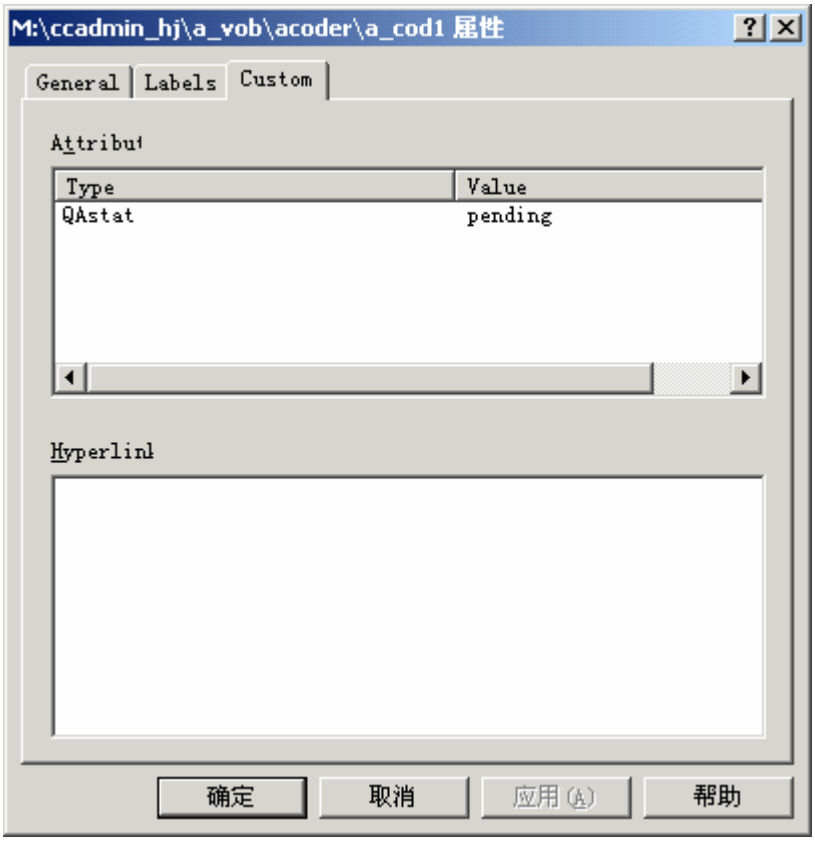


在 Attribute 下的空白区域选择 Add Attribute



在对话框中输入新建的 attribute type， 点 OK





4.3.3 超链（Hyperlinks）

- 功能介绍：创建一个 **VOB** 之内，或者多个 **VOB** 之间的两个元素的关系。
- 应用策略：我们要求每一位项目成员使用超链，把自己的软件工作产品和他参考的上阶段的输出关联起来，要求使用完整的版本扩展名，例如：\\scm-test\M:\doc\详细设计.doc@@main\REL2。每个项目预定义如下超链类型：

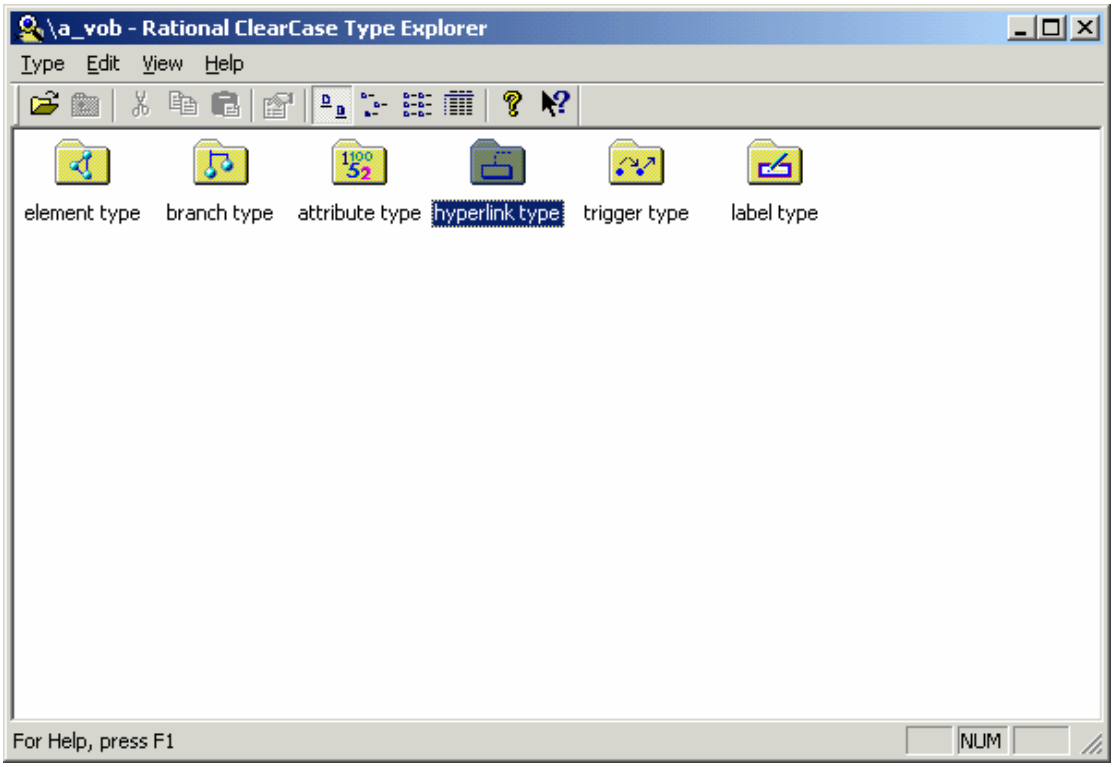
超连类型名	应用时机	功能说明	应用人

DR	设计完成后，提交评审前	设计文档->需求文档之间的参照一致性	设计人员
TR	测试文档完成后	测试需求、测试用例->需求文档之间的参照一致性	测试人员
SD	代码完成后，提交测试前	代码->设计之间的参照一致性	编码人员

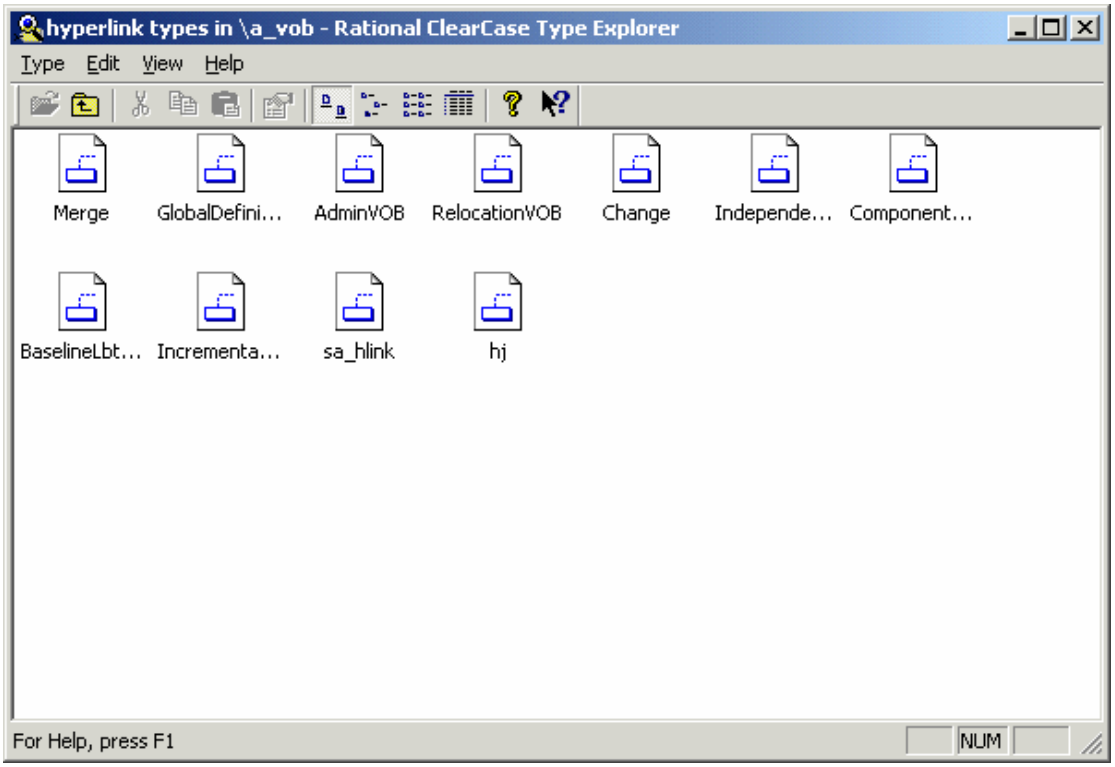
- 命名规则：预定义以上三种超链类型，使用统一的命名方式。其中 D 代表 design（设计），R 代表 requer（需求），S 代表 src（源代码），T 代表 test（测试）。
- 实现方法：

a)建立一个 Hyperlink Type

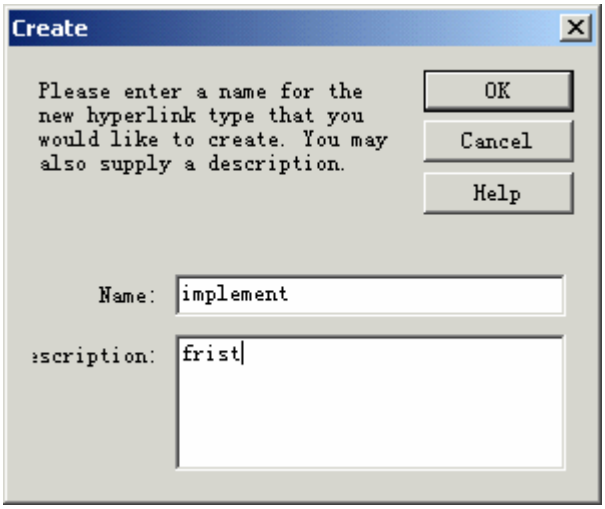
打开 Type Explorer



双击 Hyperlink Type



击 Type>Create



点 OK，完成

b）Attaching a Hyperlink

使用 Cleartool mkhlink 命令

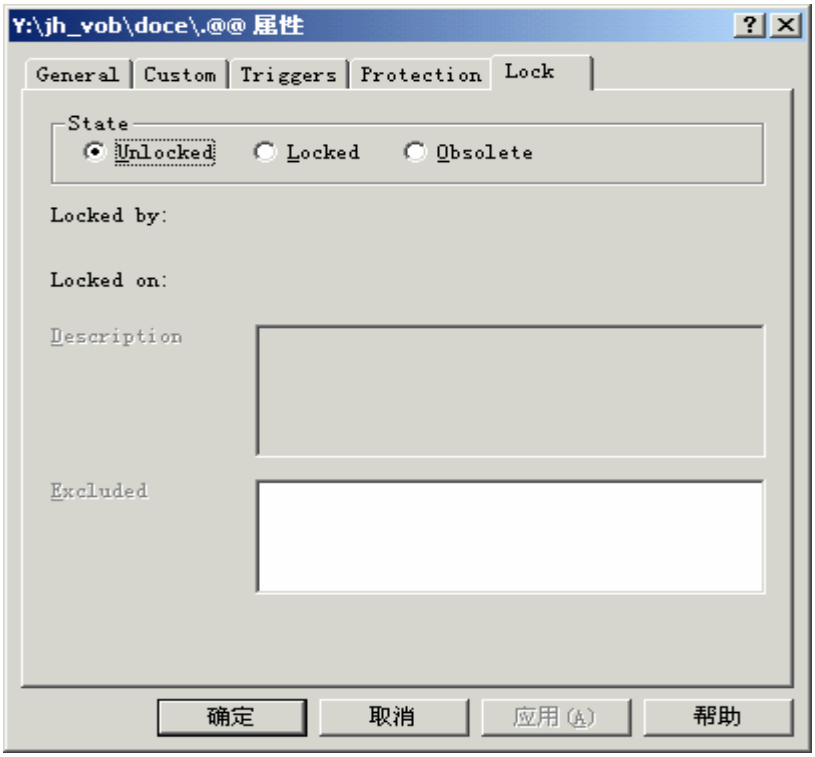
4.3.4 触发器（Triggers）

- 功能介绍：在 GUI 中不能建立 trgger，只有 vob 的 owner 用户和 clearcase 组成员有权建立 trigger，trigger 对 vob 的建立删除不起作用，trigger 在 clearcase 操作前或后被触发执行，trigger 的建立不一定必须两步， 通常定义 trigger 时，首先建立预定义的可执行程序，然后定义 trigger 起作用的条件；但也可以用“ccperl –e die()”代替预定义可执行文件，默认返回 false 值。
- 应用策略：统一由 Clearcase 系统管理员进行，其它人员可以向系统管理员提出使用触发器的需求。CC 管理员在创建 VOB 时必须应用两个触发器，防止随意删除和防止 checkout 时不加注释。
- 命名规则：根据其功能命名，并形成清单，放到 PM\SCM 目录下。
- 实现方法：请参考附录 4：触发器语法和实例。

4.3.5 锁（Locks）

- 功能介绍：限制除了“Exception”列表外的任何其它用户进行修改。
- 应用策略：元素通过评测发布时，要求配置管理员锁定此元素，任何人不得修改；
- 实现方法：

选择一个元素，打开元素的属性窗口，在 Lock 页进行属性设置



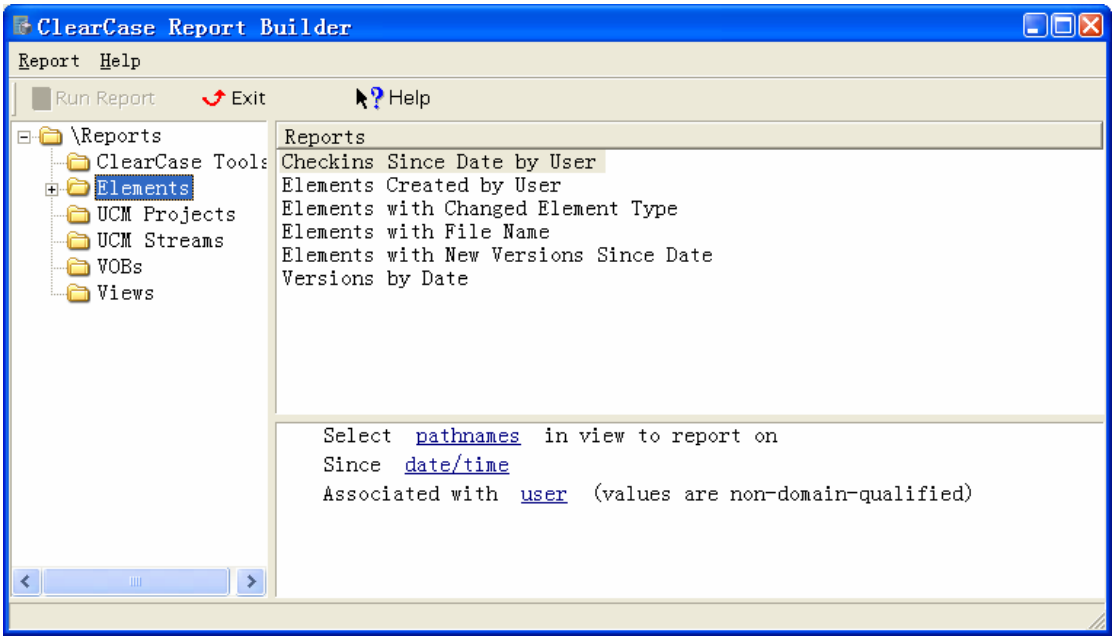
4.3.6 生成报告：

- 功能介绍：当元素被修改或者合并时，Clearcase 创建并保存事件记录，事件记录中包括：用户名、日期、注释、主机名以及修改描述。我们可以对事件记录和原数据类型进行选择和过滤，进而生成我们所需的报告。
- 应用策略：由于系统预定义的格式的限制，我们暂时还需要手工生成《配置状态报告》、《基线审核报告》以及《产品配置清单》，《基线发布通知单》可以由报告生成器的标签过滤器来实现。
- 命名规则：由报告生成器生成的报告根据其功能命名。
- 实现方法：

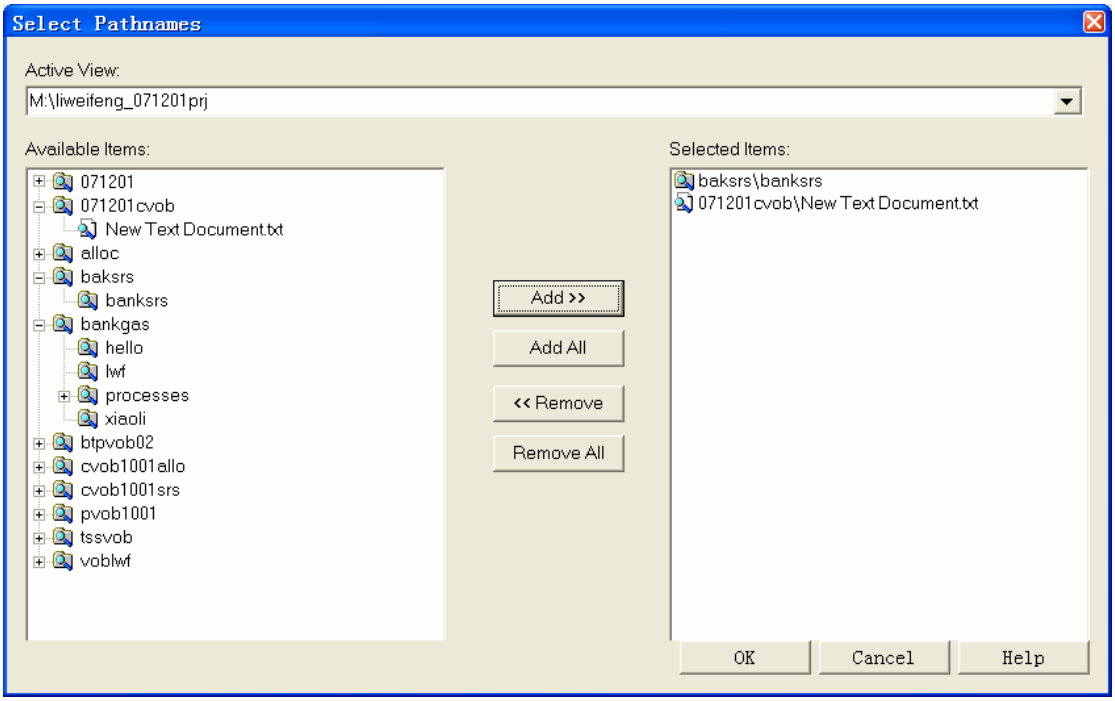
ClearCase 报告生成器预定义了六类共计 27 种报告格式（其中 14 种是针对 UCM 的），每种格式使用 2-3 个字段作为过滤器和显示的字段。生成的报告只能保存为*.HTML,*.XML 以及*.cvs 格式。

可以通过编写 Perl 语言脚本的方式自定义报告格式。

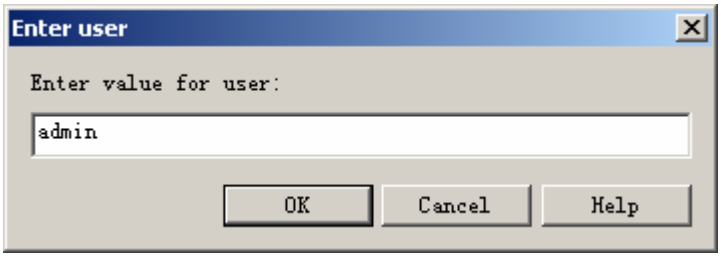
打开开始>程序>Rational ClearCase Administration>ClearCase Report Builder 应用程序，就会打开下面的窗口。



点击右下栏的连接设置过滤条件，设置路径的窗口如下图：

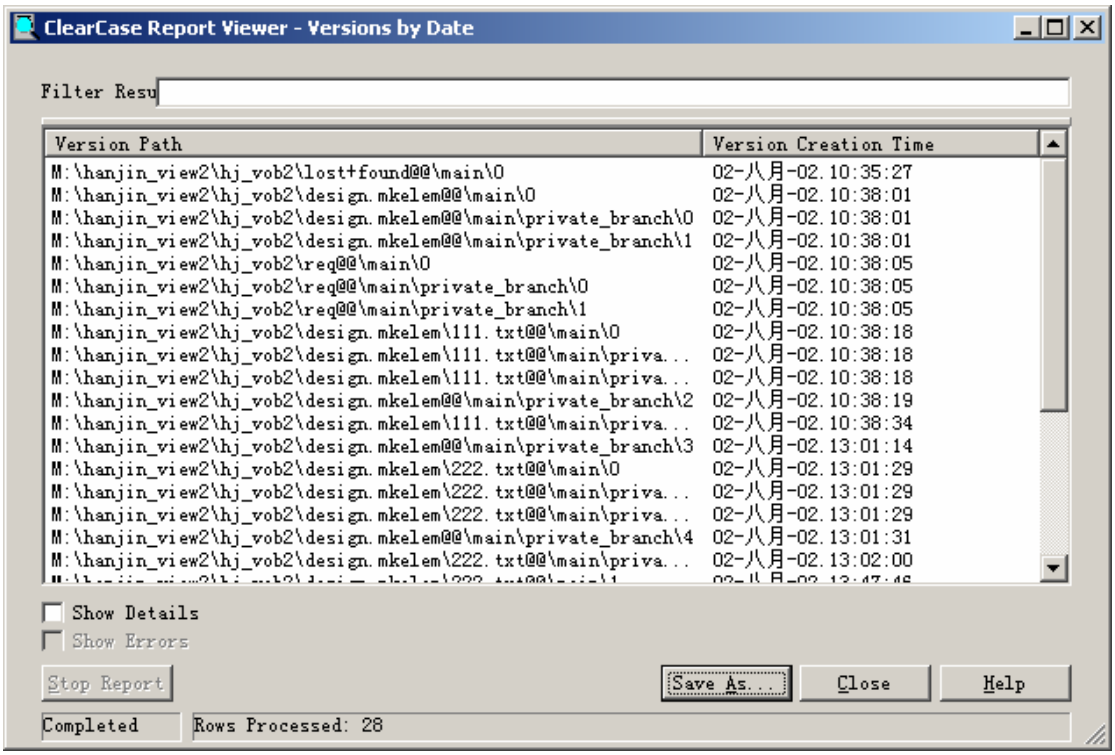


选择用户的窗口是一个要求手工输入的文本框，如下：



还有其他过滤条件的设置，在此就不一一赘述了。

生成的报告的格式图下：



4.3.7 备份与恢复

- 功能介绍： Clearcase 系统管理员定期或者在事件驱动下，使用 CC 自带的工具进行备份和恢复。备份和恢复 clearcase 对象包括 VOBs（份功能可能跳过打开的可写文件；保证所有的 vob 数据都备份你需要：设置你的备份功能去保存打开的可写文件，或锁定这个 vob 并且拷贝这个 vob 或 view 存贮目录到其它位置，然后备份这个拷贝，或在备份之前停止 clearcase；），Views，registries 和 license hosts。
- 应用策略： 根据计划定期执行标准备份。
- 实现方法：

vob 备份选项：标准备份（备份整个 vob；标准备份一般要锁定这个 vob 并且备份整个 vob(vob 数据库，和 vob 存储池和在 vob 存储目录的各种文件)使用标准 PC 备份功能。在备份期间这个 vob 必须锁定。）和半备份（备份 vob 的子目录（数据库和资源池）；半备份使用 clearcase 功能，备份 vob 的两个主要的部分。这个方法缩短 vob 锁定的时间，而且需要较多的磁盘空间。恢复过程比标准备份复杂）。

标准备份

- 备份下面 vob 几个部分：
  - 顶层目录
  - vob 库（db 子目录）

- 资源池（s 子目录）

◆ 派生的对象池（d 子目录）应该备份

- 如果没有，你可能需要重新同步这个库并且使用 cleartool rmdo 命令删除“missing” DOs 来派生对象池

◆ cleartext 池（c 子目录）不需要备份

例：

1 锁定 vob

c:\>cleartool lock vob:\sabela

locked versioned object base “\\sabela\vobs\vob1_vbs”.

2 停止 clearcase,不能 copy vob 库中的 db

3 执行备份

4 重新开始 clearcase

5 解锁 vob

c:\>cleartool unlock vob:\sabela

Unlocked versioned object base

“\\sabela\vobs\vob1_vbs

为了防止丢失“open for write”文件,这个例子使用 vob 拷贝来达到备份的目的

步骤 3

确认备份包括

顶层存储目录和他的文件

数据库目录和文件的拷贝

资源池

派生对象和 cleartext 池，如果规定

备份：使用数据的一个拷贝

1 锁定这个 vob

2 拷贝 vob 库到另外的位置

3 解锁这个 vob

4 备份这个锁定库的拷贝

5 备份后删除这个库的拷贝

拷贝这个 vob 库（db 子目录）到磁盘其它位置。

备份如下：

- 顶层存储目录和他的文件
- vob 库和他的文件的拷贝
- 资源池（s 子目录）
- 派生对象池(d 子目录)

半备份：

包括备份两个主要的 vob 片断：

vob database-设置 vob 使其有 vob_snapshot 功能来周期性的锁定 vob 和 copy 这个 vob database 到其它位置。这个 vob 的拷贝是作为一般备份过程的部分代表的备份

vob storage pools-日常的备份 vob storage directory,没有锁定 vob

半备份的主要好处是减少 vob 加锁的时间，因为 vob storage directory 可以备份已经解锁的 vob

缺点是：

因为必须将将两个 vob 的库放在磁盘上所以需要较多的空间

vob 恢复过程较复杂

在恢复池和恢复库有差异时恢复 vob 可能会丢失数据

准备：

- 使用 vob_snapshot_setup modparam t 加一个 vob 到备份 vob 的列表中
  - 你必须是 vob 的 owner 权限或 clearcase 组成员登录来执行这个命令
  - 这个 vob 必须是在 clearcase 3.x 以上版本的格式
- 参数
  - -snap_to ：指定一个目标目录
  - -db_check：运行库的连接和完整 check

<ccase-home-dir>\etc>vob_snapshot_setup modparam –dbcheck yes –snap_to \\sabela\vobs\snaps\project_a \project_a

语法

<ccase-home-dir>\etc\vob_snapshot_setup modparam –snap_to snap-dir-pname{ -dbcheck yes| -dbcheck no} –notify login-name[,…]vob-tag

NOTE:在<ccase-home-dir>\etc 目录，使用全部参数去运行 vob_snapshot

备份运行

- vob_snapshot 命令是*<ccase-home-dir>\config\cron\ccase_day.bat* 脚本的一部分
- 在每天晚上这个不冲突的时间进行
- 做以下脚本：
  - 检查 snapshot target 目录存在且可写
  - 加锁这个 vob，如果这个 vob 不能锁定可以作为可疑在这个 snapshot 标记
  - check 到一个目录
  - 建立一个目标目录，使用 uuid 作为他的名字
  - 拷贝这个 vob 目录树到新的目录
  - 解锁这个 vob
  - 对列表的下一个 vob 重复上述步骤
  - 在所有 vob 上运行 db_check 如此设置-如果可疑的 snapshots 通过，这个可疑的指定被删除

what to do

1 登录你做备份的主机

2 准备这个 vob 每天 database snapshots

<ccas-home-dir>\etc\vob_snapshot_setup

3 在本地机的 snapshot 列表确定这个 vob

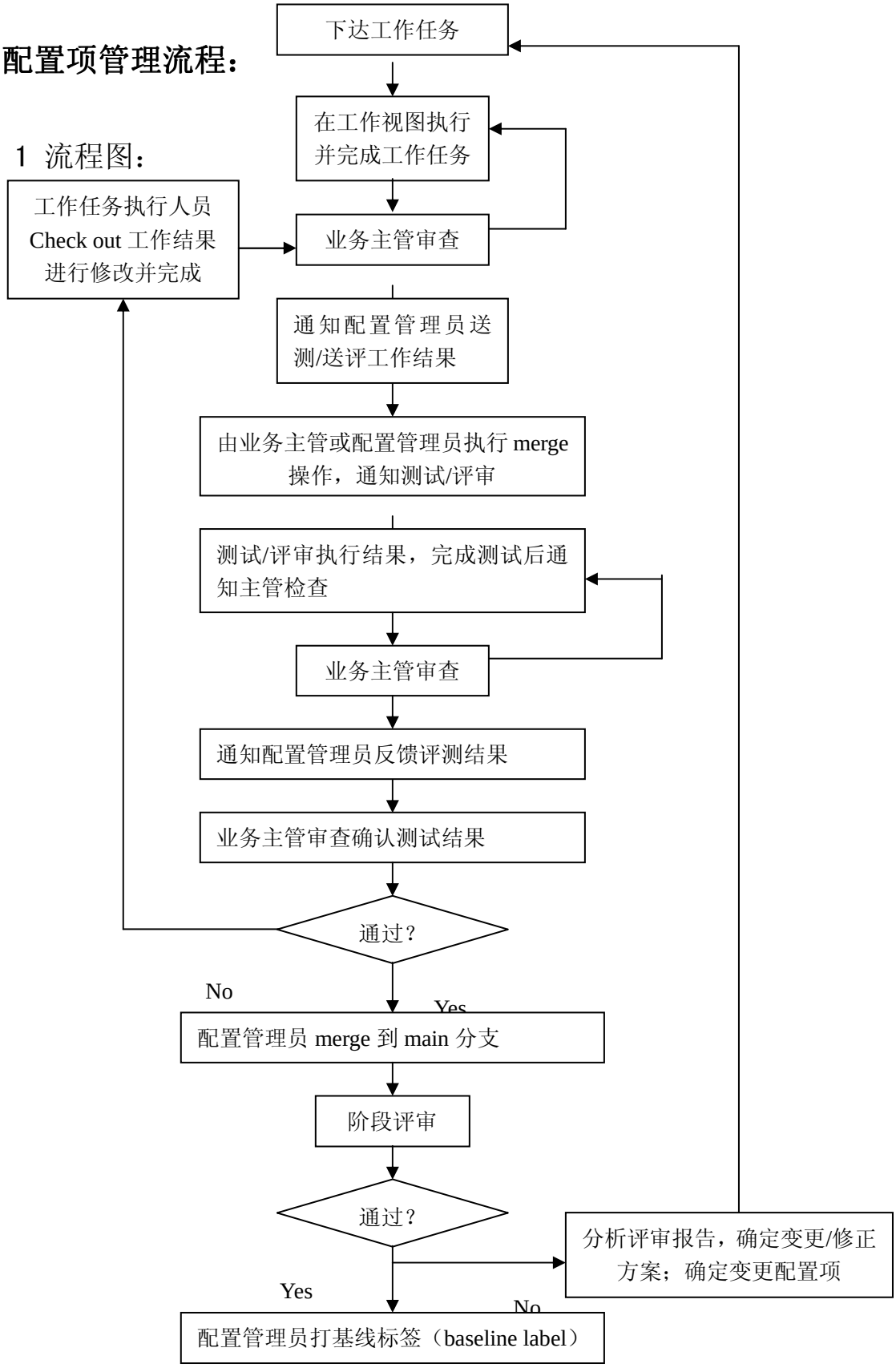
<ccas-home-dir>\etc\vob_snapshot_setup lsvob

4 保证这个-snap_to 目录和 vob 存储目录以日常一般备份的方式被存储

5 使用 ccas_day.bat 自动锁定这个 vob 而且用这个 vob_snapshot script 拷贝他的库到磁盘上。

5 配置项管理流程：

5．1 流程图：



5．2 流程说明：

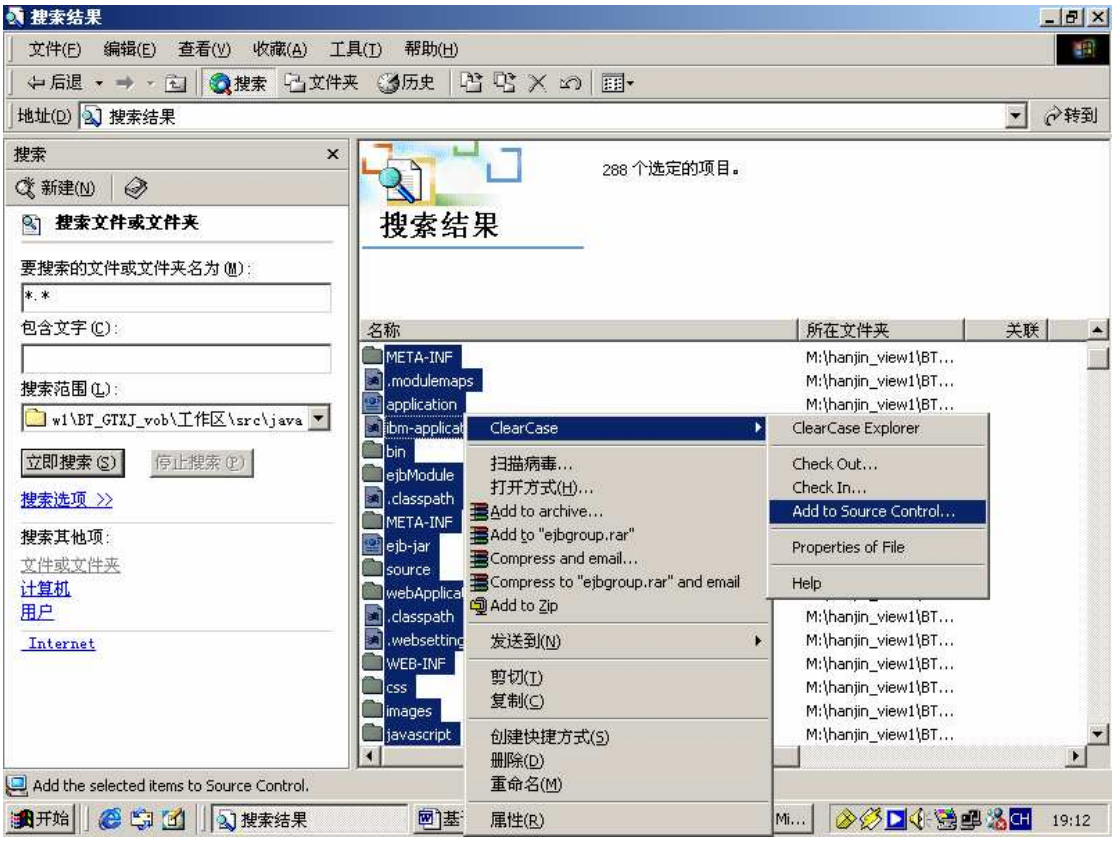
配置项纳入配置管理的流程如下：

1） 业务主管（项目经理、需求分析主管、设计主管、编码主管、测试主管）根据项目开发计划制订具体开发任务给项目成员，并在项目组内达成一致；

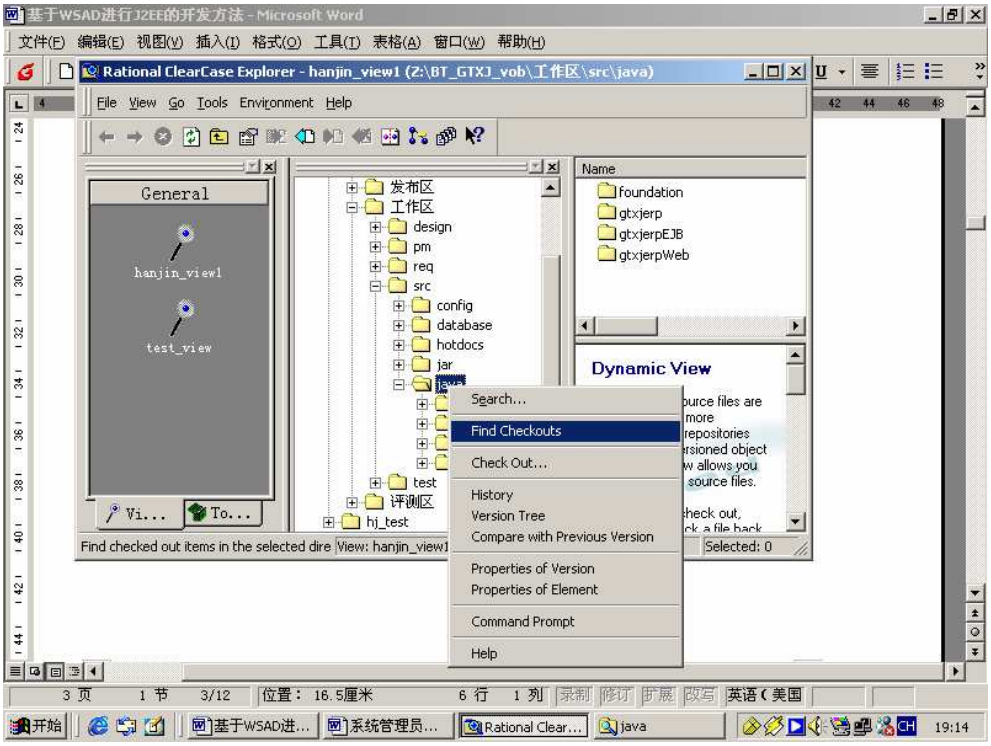
2） 项目成员得到工作任务指派后，在本地完成开发任务，保存（或复制）到相应的工作视图中，并执行 Add to Source Control 命令加入配置控制；视图将根据配置规则自动创建工作分支 Develop，项目成员在工作分支上完成自己的工作任务；

小技巧：

新建的文件要全部“加入到源代码控制之下”，才能进行版本控制并共享给他人；把新建的文件放到视图或者资源管理器映射位置的对应目录下，使用搜索功能选择出此目录下的所有元素（使用选择条件*.*即可），全部加入到源代码控制之下。如下图：



编辑文件时须先检出，完成后要确保所有被检出的元素都被检入，别人才能看到你所做的修改。可以使用“Find checkout”命令找到所有被检出的元素。以上属于使用 ClearCase 进行团队开发的内容，如有问题请与配置管理员联系。



3） 完成工作后，主动通知主管审核，业务主管应用“属性变量”=值的方式标识审核结果；如果没有通过审核，则把审核意见反馈给编码人员进行修改，直至审核通过为止。

4） 审核通过，业务主管将提交的工作产品归并到 test 分支上，并通知配置管理员进行评审或测试，由配置管理员/业务主管打上评测标签（checkpoint label），同时生成评测视图（snapshort view），通知相关人员使用此评测视图进行评审/测试；

注：如果项目组采用基于 WEB 的开发工具，测试人员可以直接在应用服务器上进行测试。

5） 评审/测试人员得到评测通知后，从评测视图中读取相应的文件进行评审/测试；评审完成后，使用评审记录（《同行评审记录》或者《评审记录》）意见反馈给评审材料负责人，由其负责整理汇总（形成《同行评审总结报告》或者《评审总结报告》），由配置管理员录入到缺陷跟踪工具 ClearQuest 中（并在 ClearQuest 中跟踪此缺陷直至 Closed），经大家确认后执行修改，并由第二人验证即为通过，执行第 6 步；测试完成后，通知测试主管检查测试结果，检查通过，则测试主管把测试结果 merge to test 支，通知配置管理员反馈测试结果，否则，要求测试人员重新组织测试，直至测试通过（如果使用 ClearQuest 记录和跟踪 bug，则不需要在 ClearCase 中操作）；

6） 配置管理员/业务主管给相应版本打上评测标签（checkpoint label），使得测试结果能够在评测视图中可见，通知编码主管接收测试结果；

7） 编码主管确认测试结果，若接受则分配编码人员进行修改，否则通知配置管理员退回；编码人员根据测试结果修改完后继续送测，直至测试通过；

8） 配置管理员把通过评测的版本 merge to main 分支上，打发布标签（release label），并应用配置规则生成发布视图；

9） 当某一阶段的配置项全部通过评测后，项目经理组织相关人员进行阶段评审，如果评审通过，则相应的阶段基线确定，由配置管理员给组成基线的所有配置项打上基线标签（baseline label），并应用配置规则生成基线视图；

10） 系统测试通过，由配置管理员根据与用户的约定（合同或者项目备忘），对组成产品的配置项打上产品标签（production label），并应用配置规则生成产品基线视图。

11） 配置管理员按照要求把缺陷录入 ClearQuest 中，并跟踪直至解决；

12） 开发完成后锁定私有分支并使它作废，以免别人在这个分支上继续开发。

5．3 变更流程：

对于变更的控制流程请参考公司程序文件 QP309《软件配置管理》的相关部分。

需要说明的是对需要变更的配置项打上变更标签（change label）

暂不考虑与 ClearQuest 的集成。

6 版本控制：

参见 RUP

7 配置项命名：

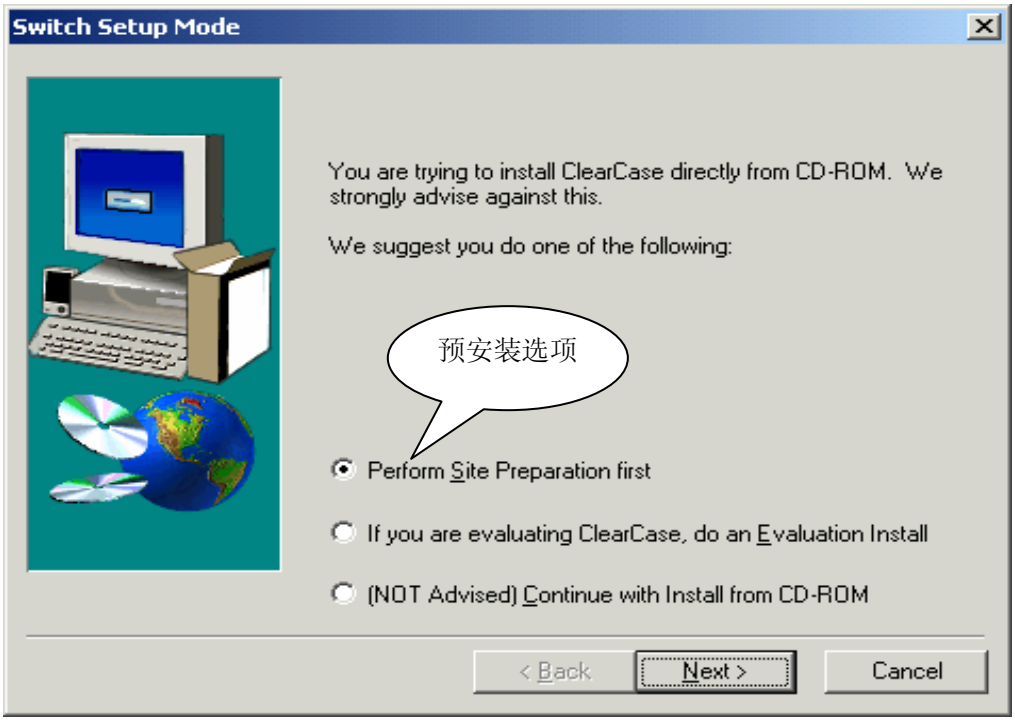
参见 RUP

8 附录 1：安装 Clearcase

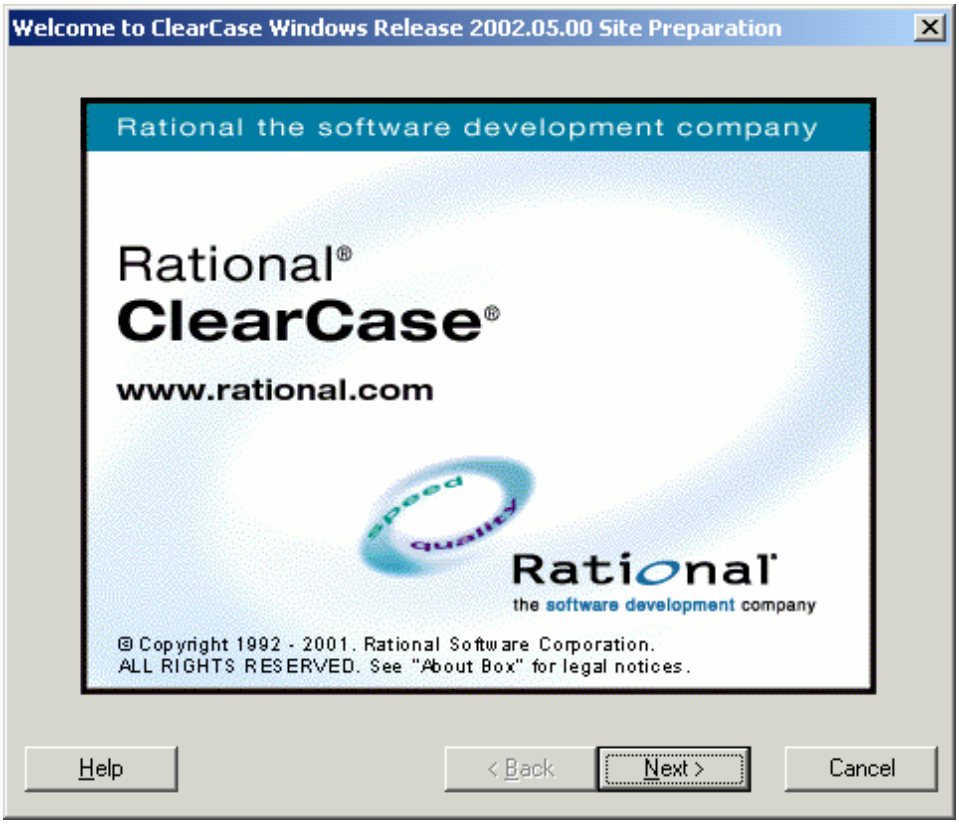
此部分使用图形接口详细描述了 Clearcase 的安装过程。

8．1 预安装：

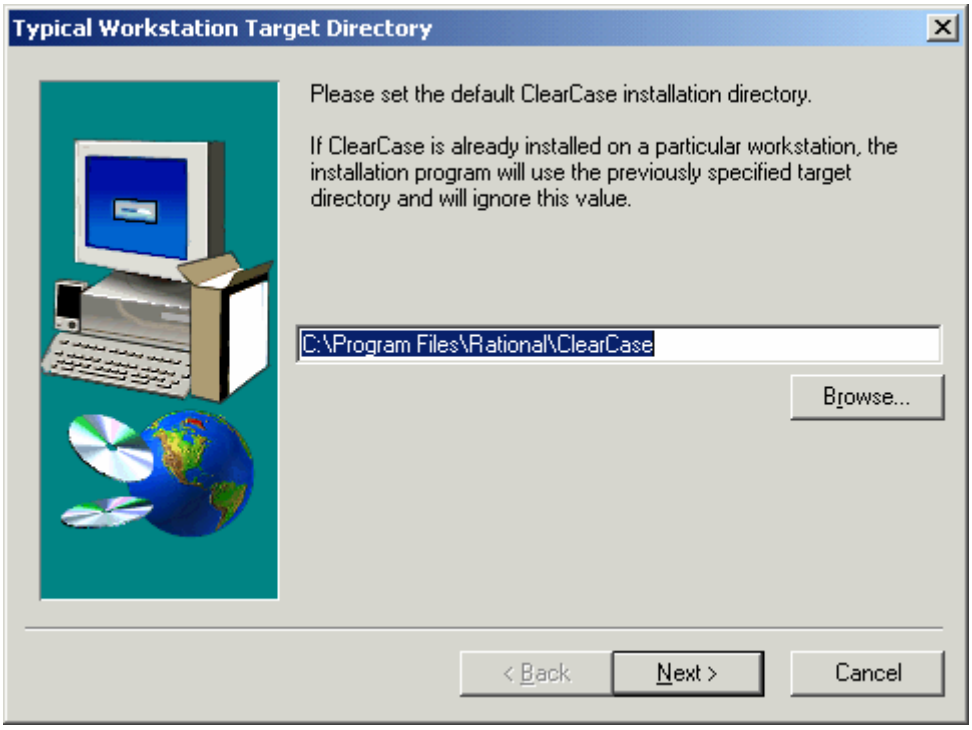
a)运行安装光盘 cpf\nt_i386\setup.exe，如果第一次安装就会显示如下图所示的安装接口，选择预安装选项，点 Next 按钮执行下一步。



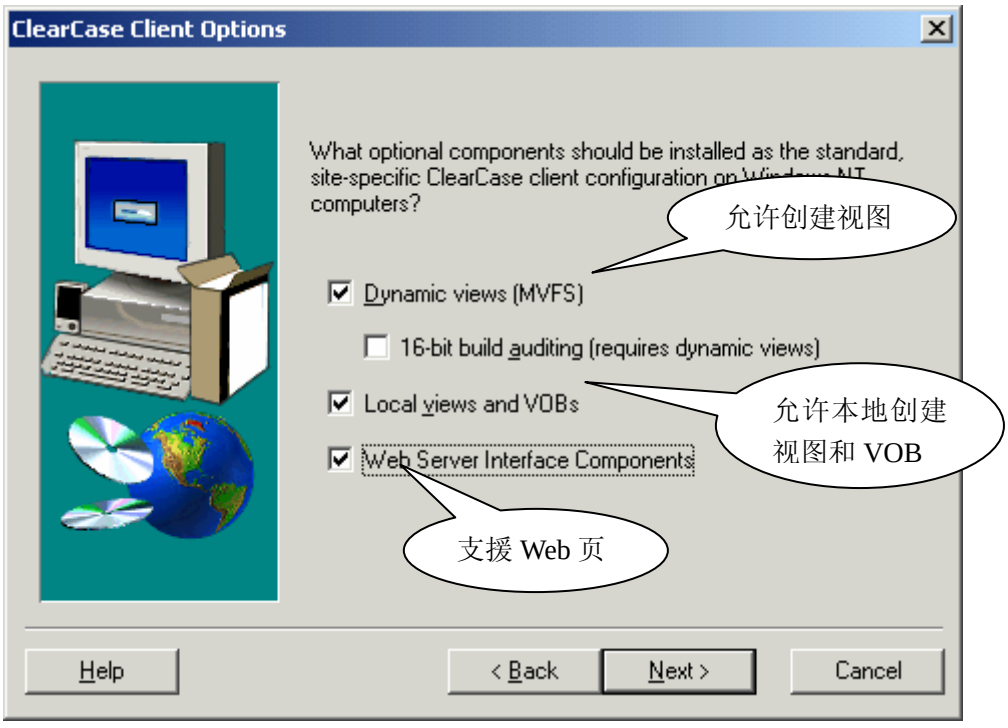
b)执行下一步



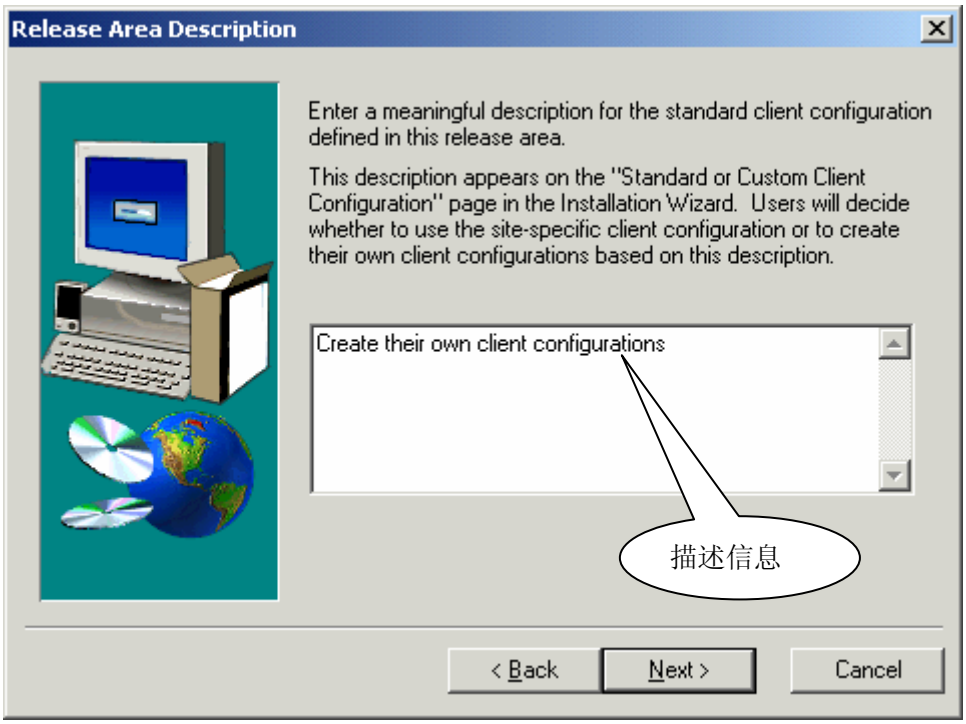
c)指定安装路径，执行下一步



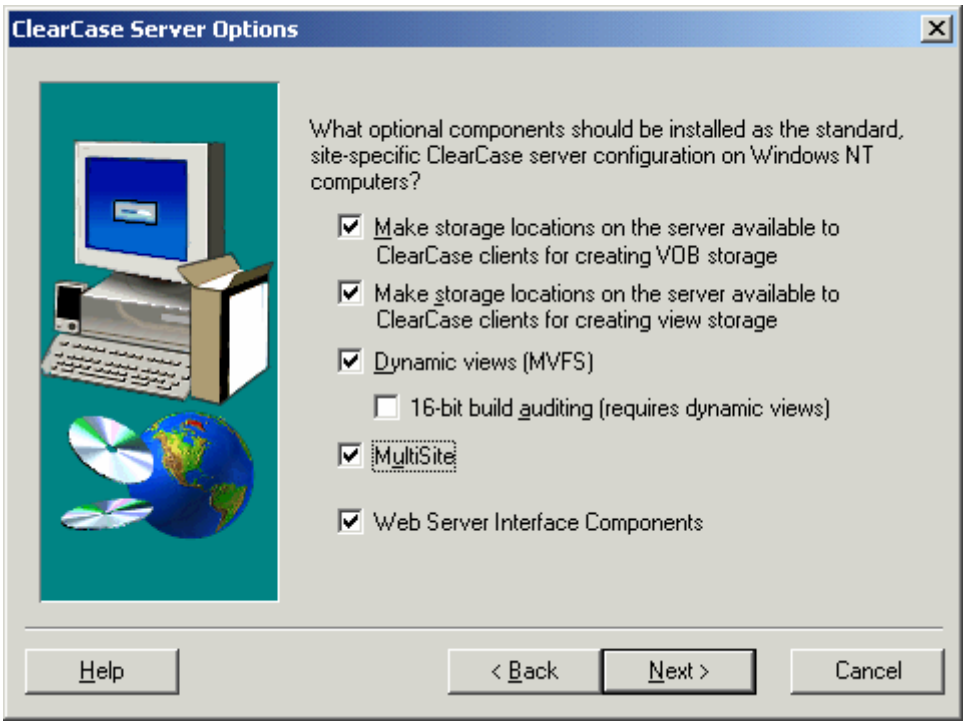
d)按照下图所示设置选项，执行下一步



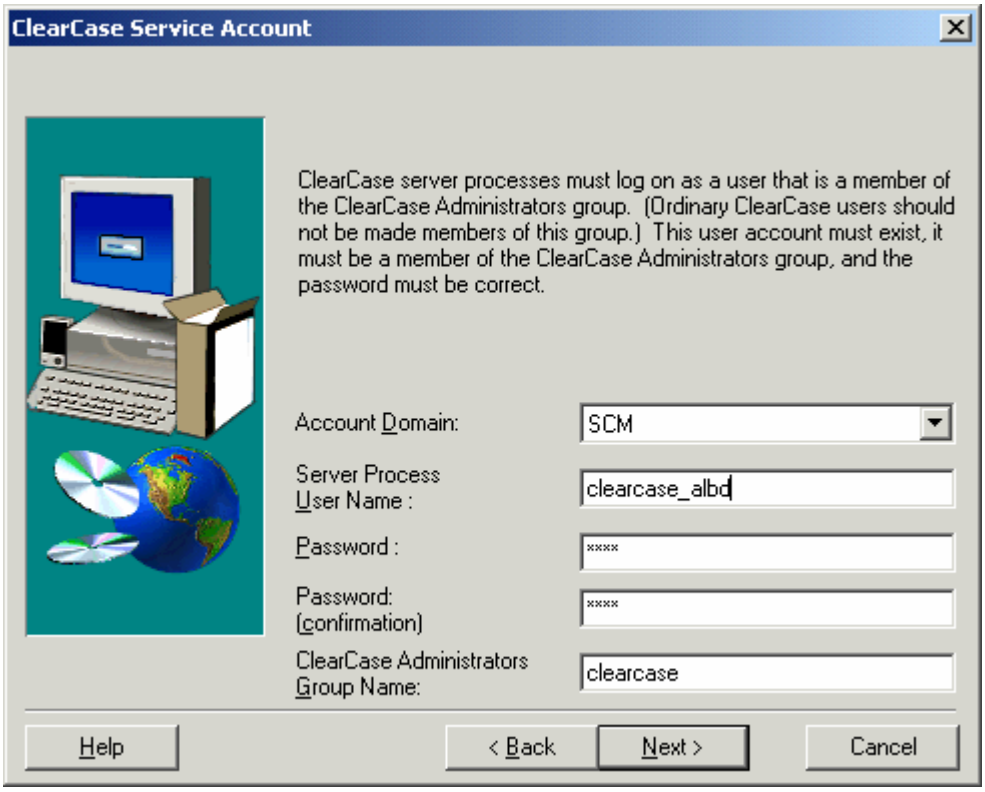
e)输入描述信息，用来提示客户端用户应设置哪种安装选项，它将在客户端安装的“Standard or Custom Client Configuration”接口显示.。执行下一步。



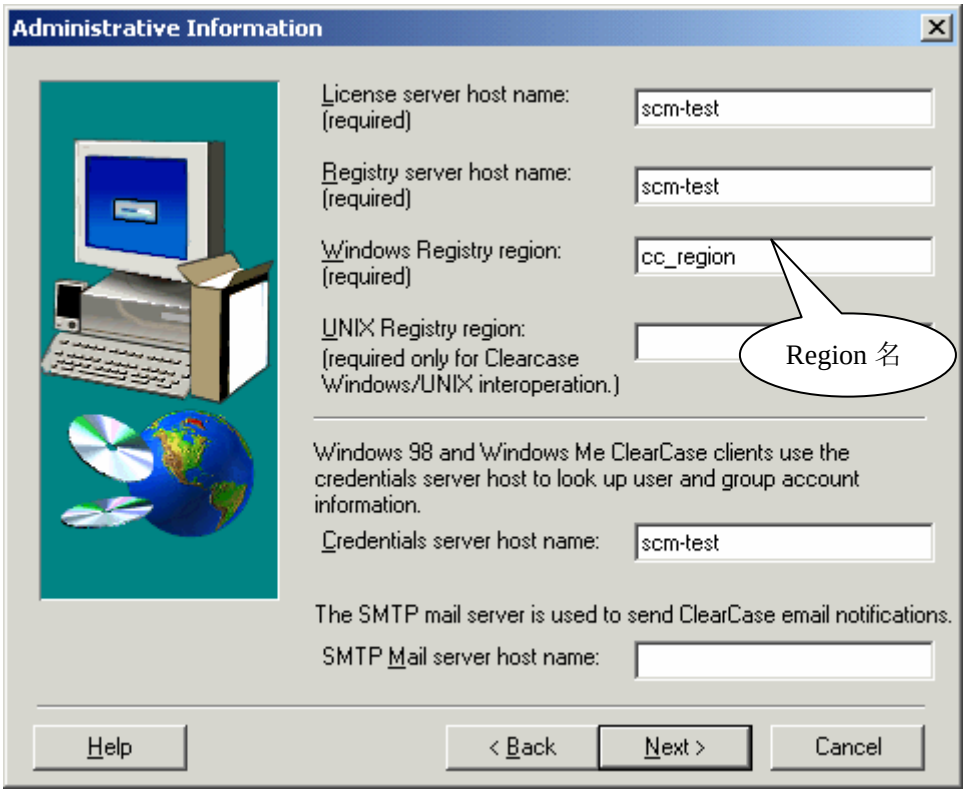
f)按照下图所示设置安装选项，执行下一步。



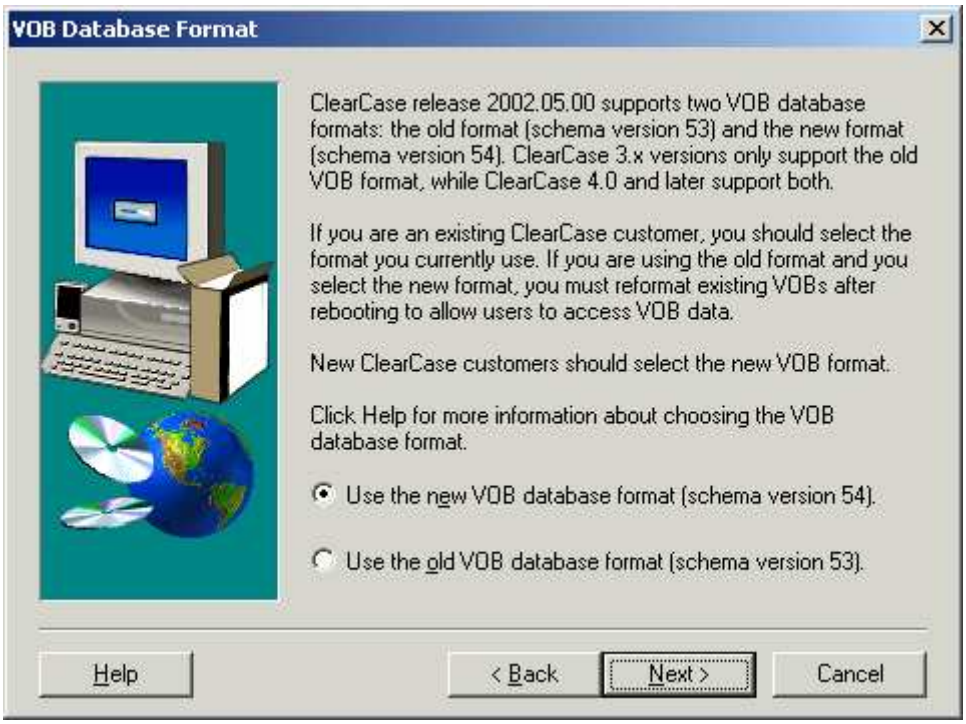
g)输入密码，制定域名，执行下一步



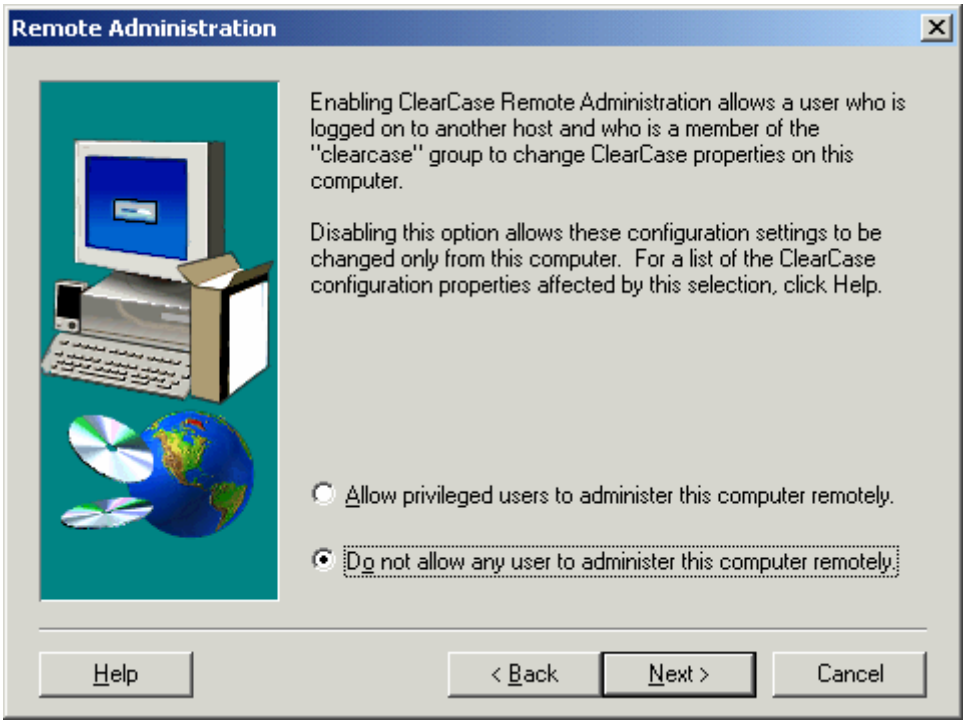
h)在下图所示接口输入主机名和 Region 名，执行下一步。



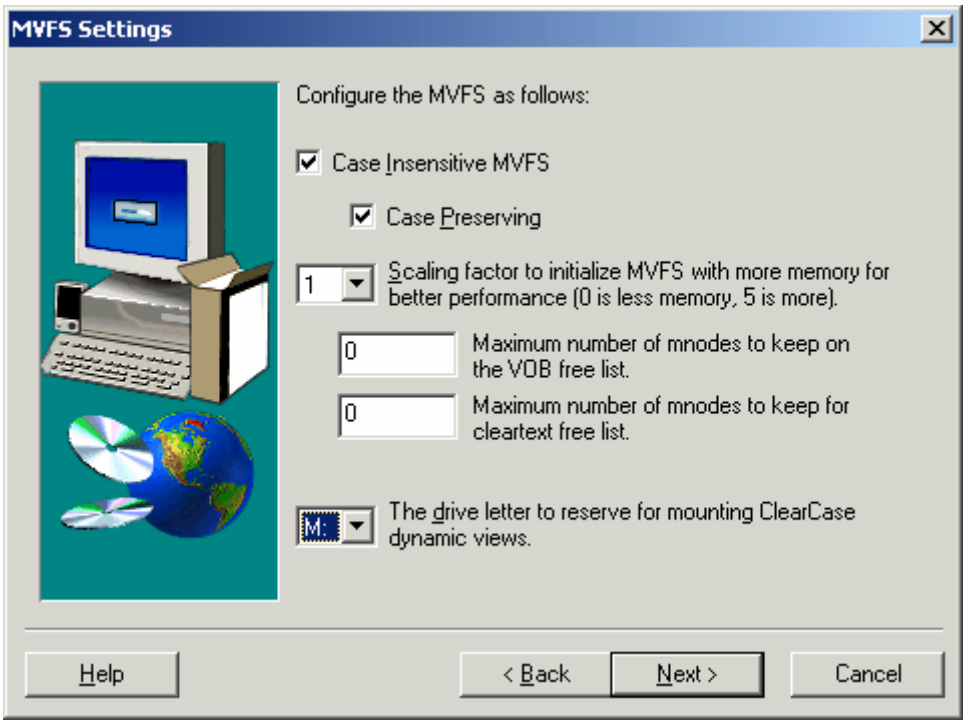
i)按下图设置，执行下一步。



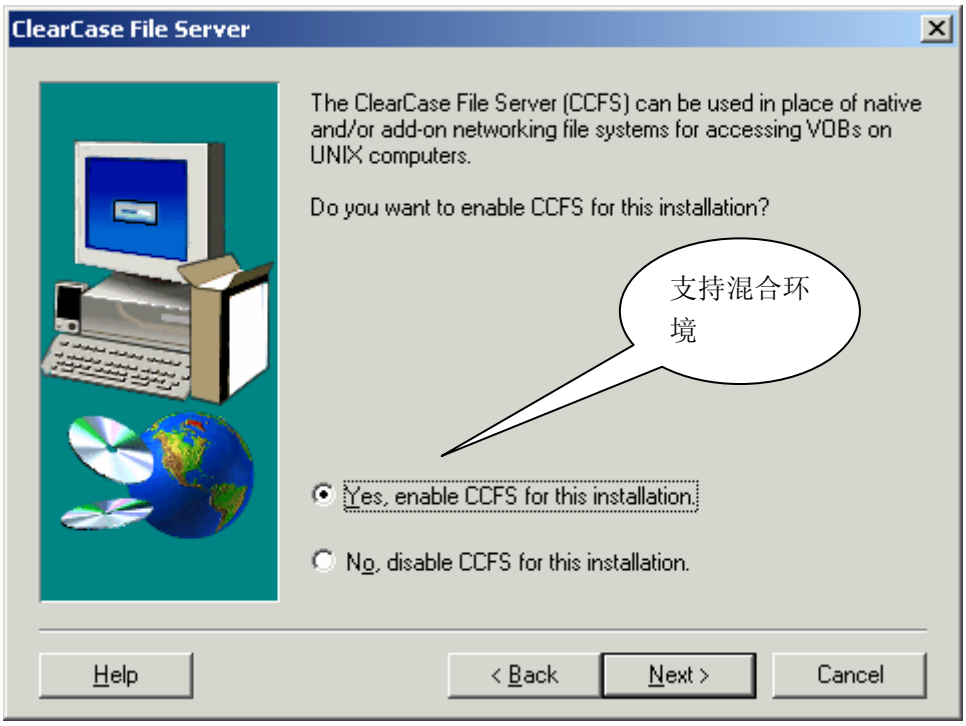
j) 按下图设置，执行下一步。



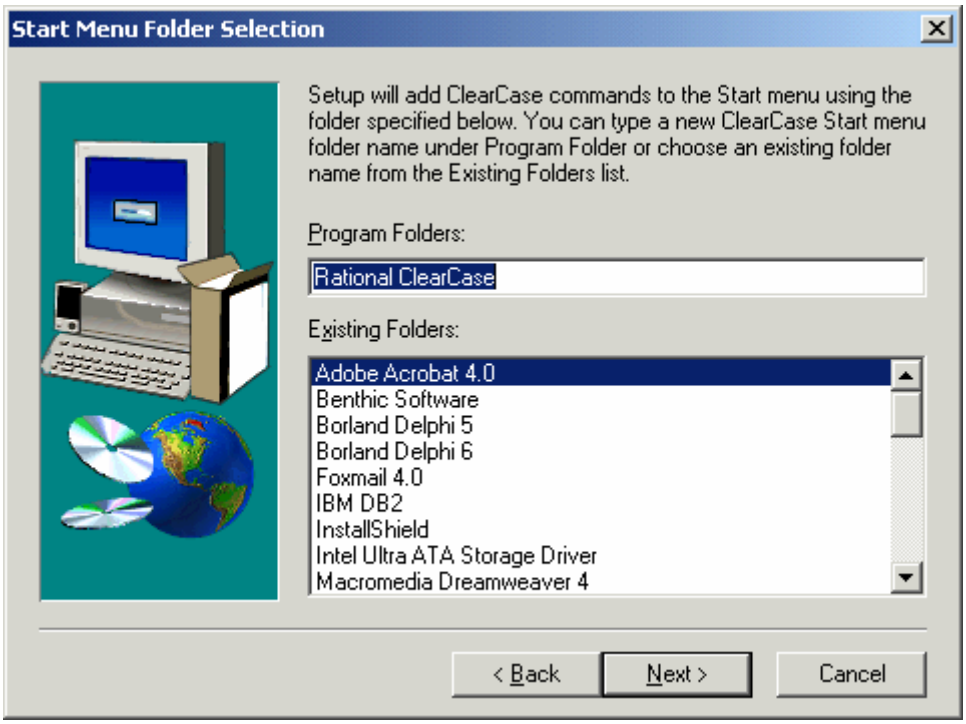
k) 按下图设置，执行下一步。



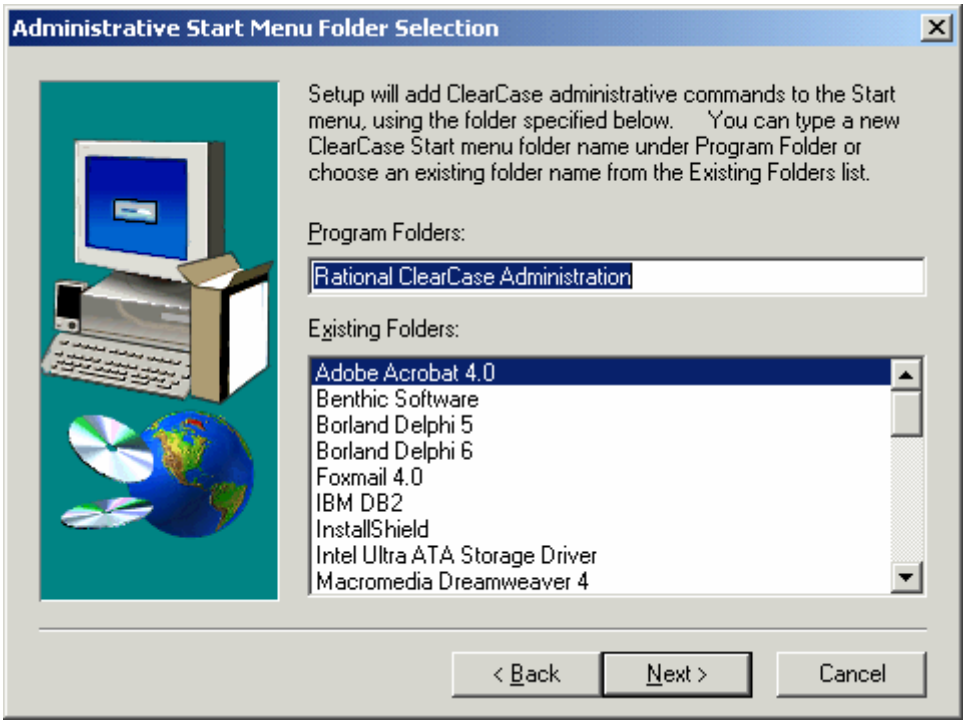
m) 按下图设置，执行下一步。



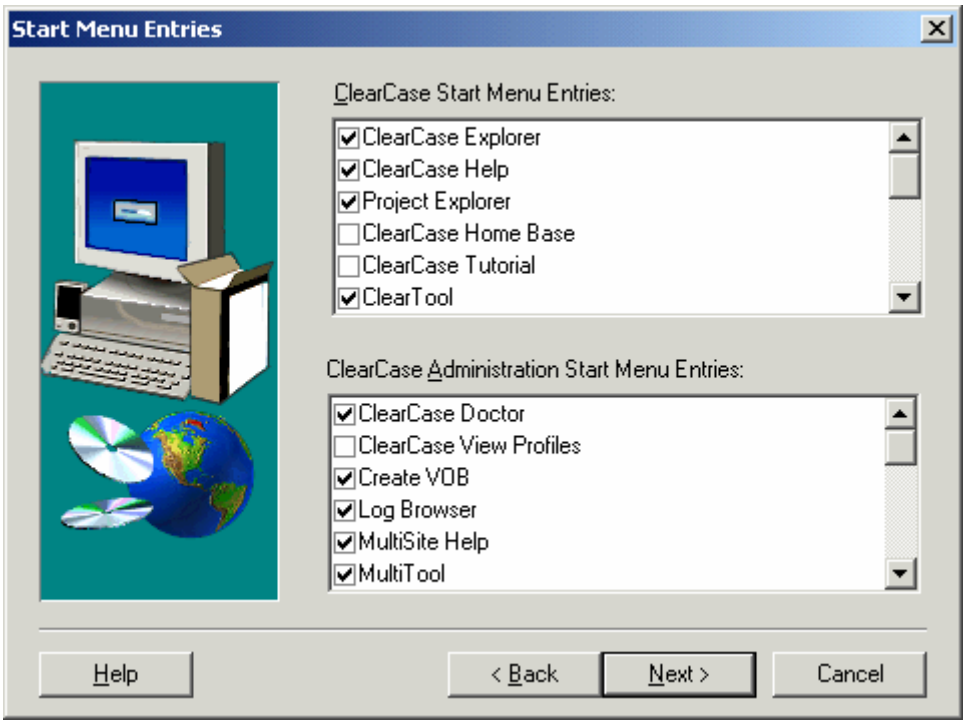
n)执行下一步



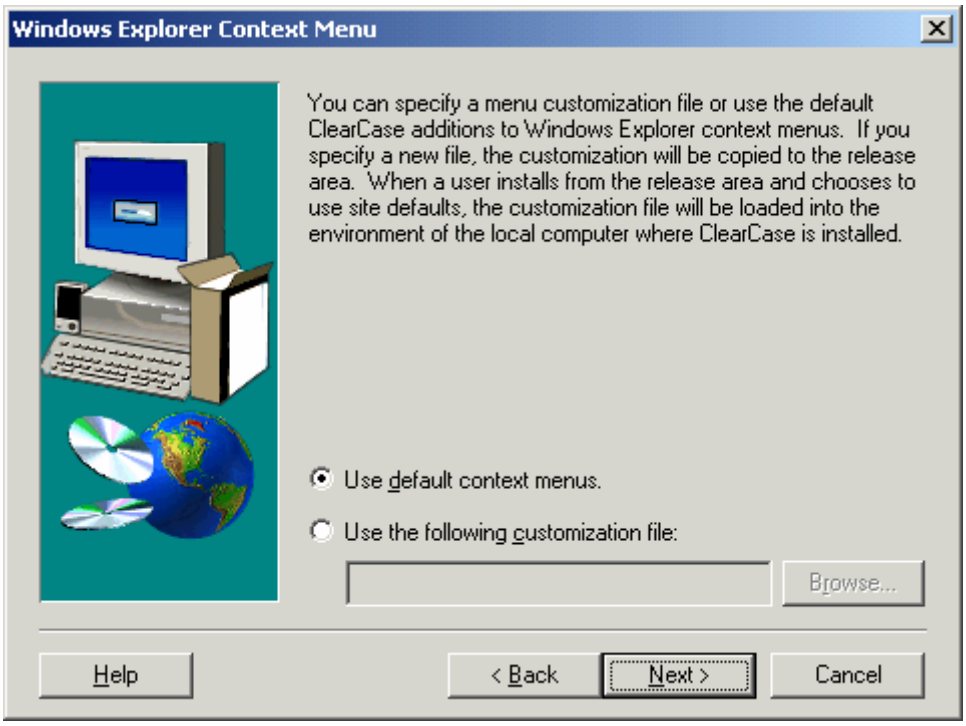
o)执行下一步



p)进行开始菜单设置，执行下一步

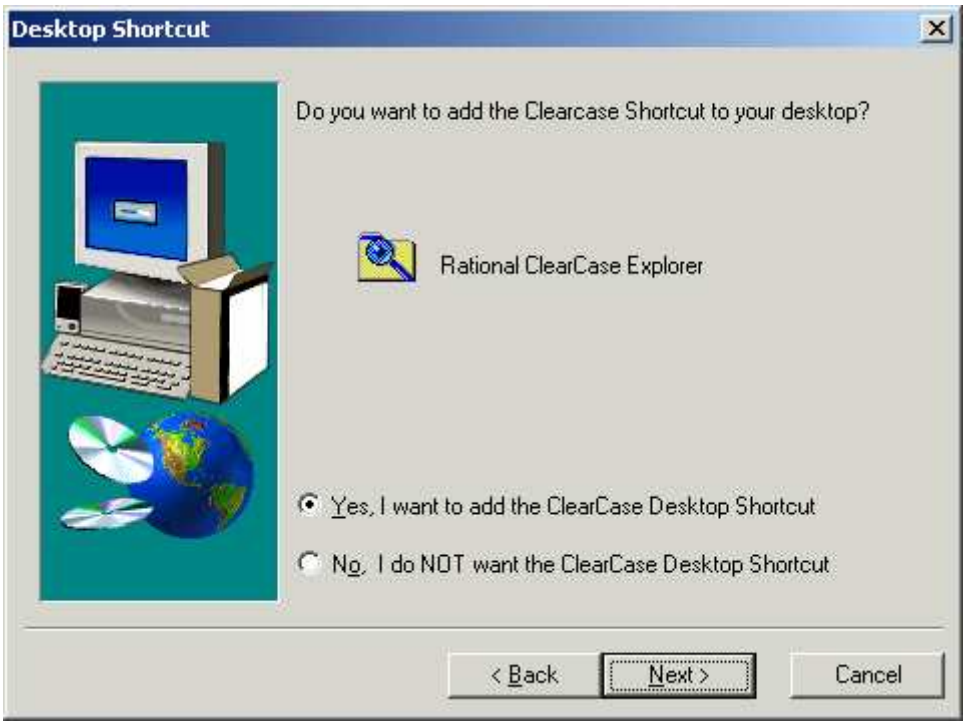


q) 按下图设置，执行下一步。

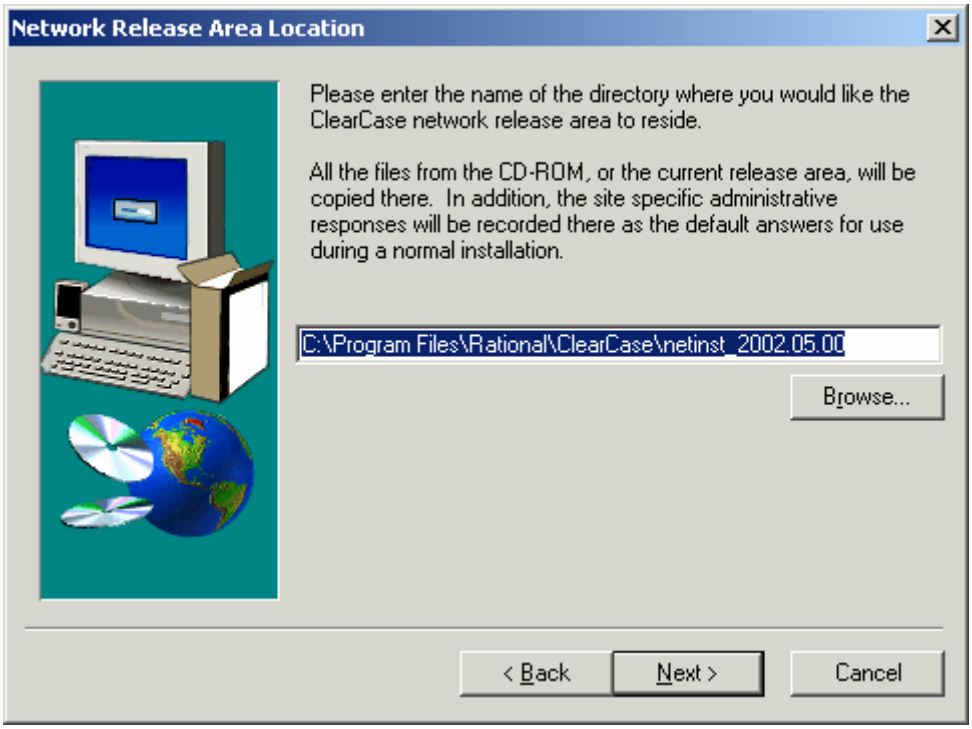


r) 按下图设置，执行下一步。

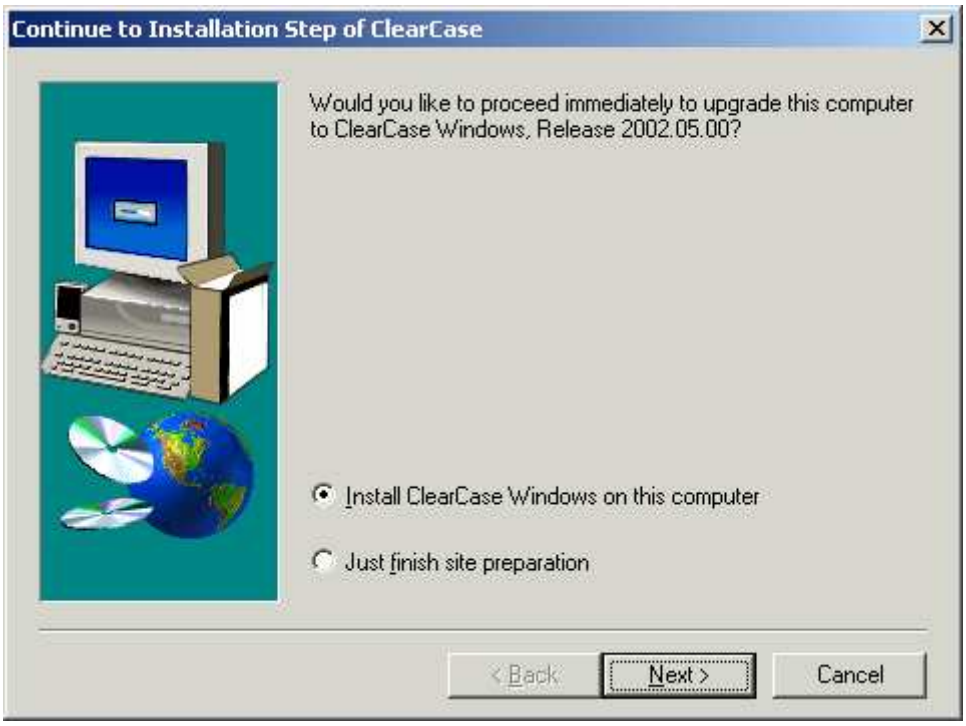
s)执行下一步。



t)指定路径执行下一步



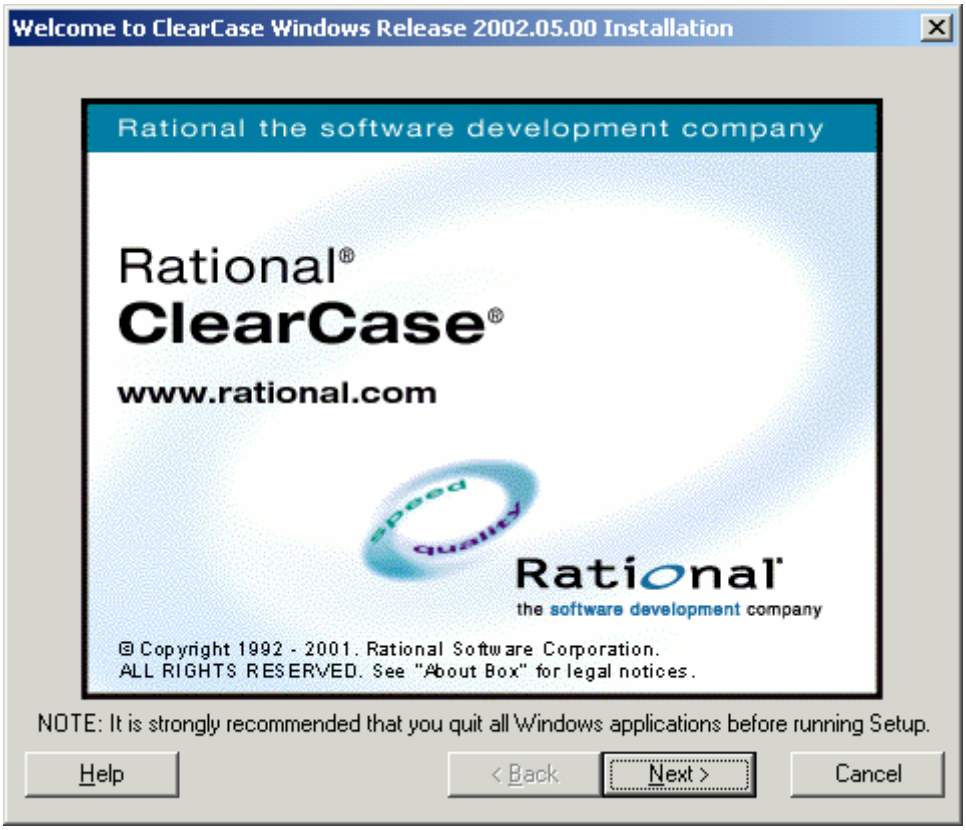
u)若想进行其它安装选择第一个选项，这里选择第二个选项完成预安装



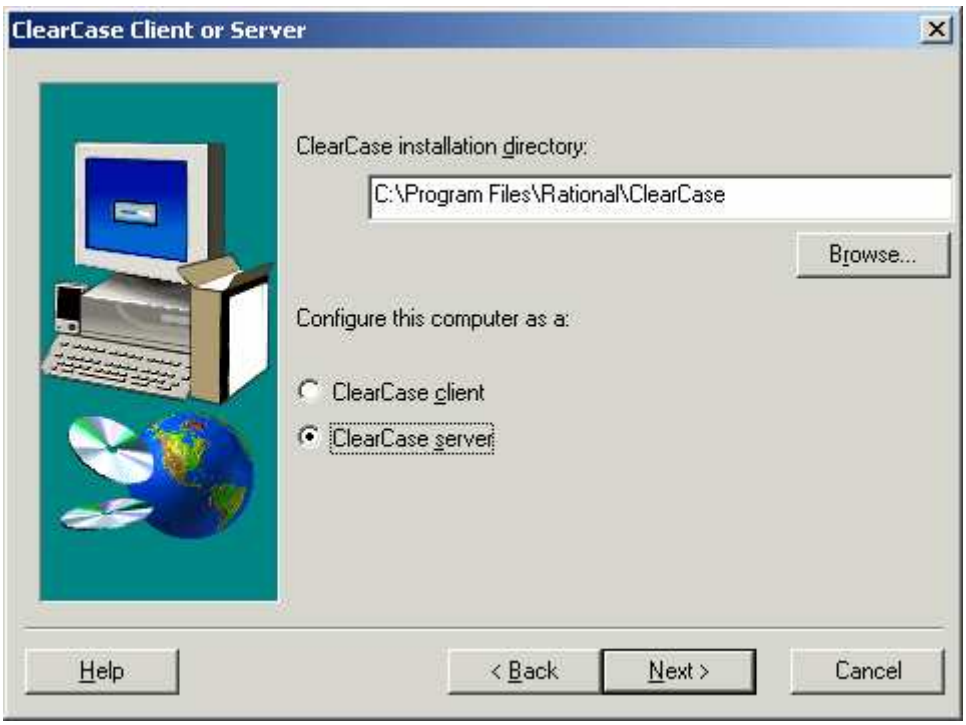
8. 2 Server 安装：

a)运行安装目录 Rational\ClearCase\netinst_2002.05.00\setup.exe,打开安装接口，执行下一

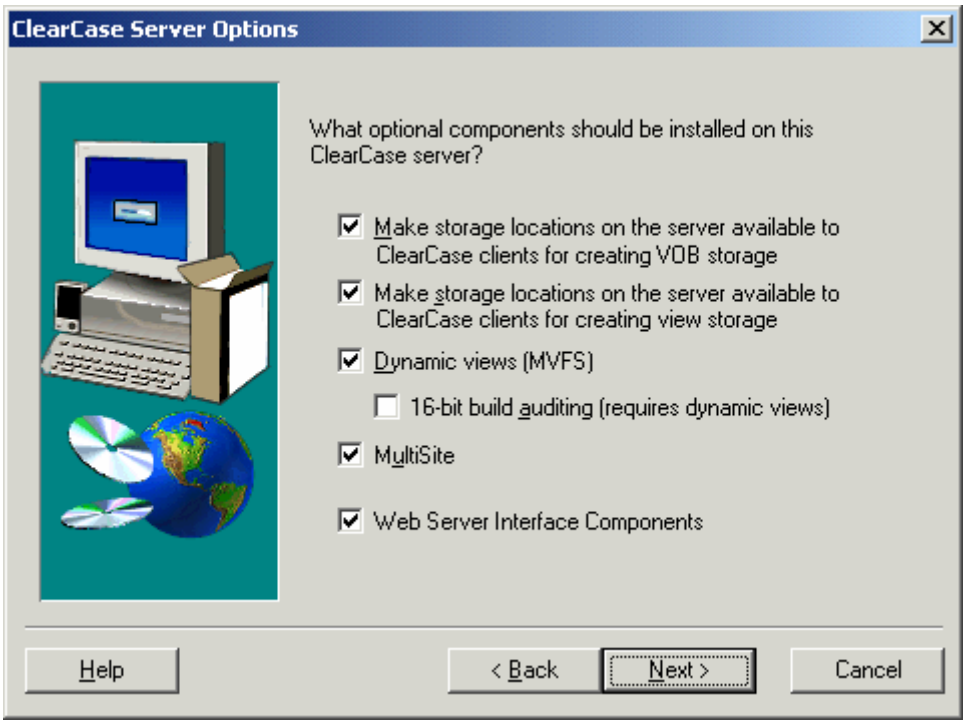
步



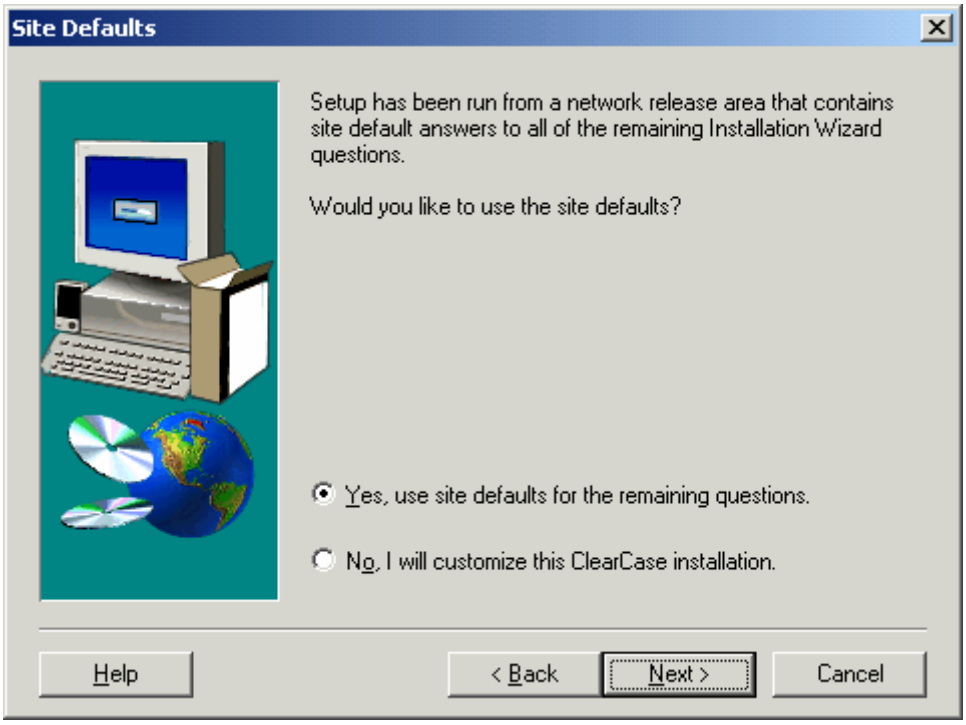
b)指定安装路径，选择服务器安装选项，执行下一步。



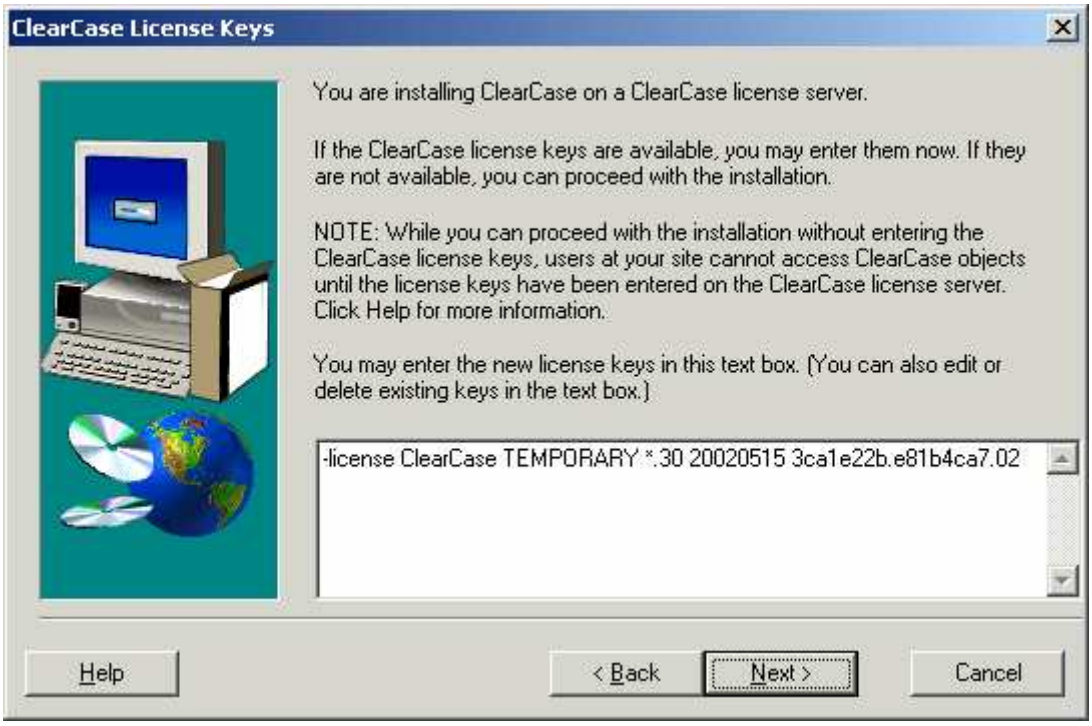
c)按照下图所示进行设置，执行下一步



d)按照下图所示进行设置，执行下一步



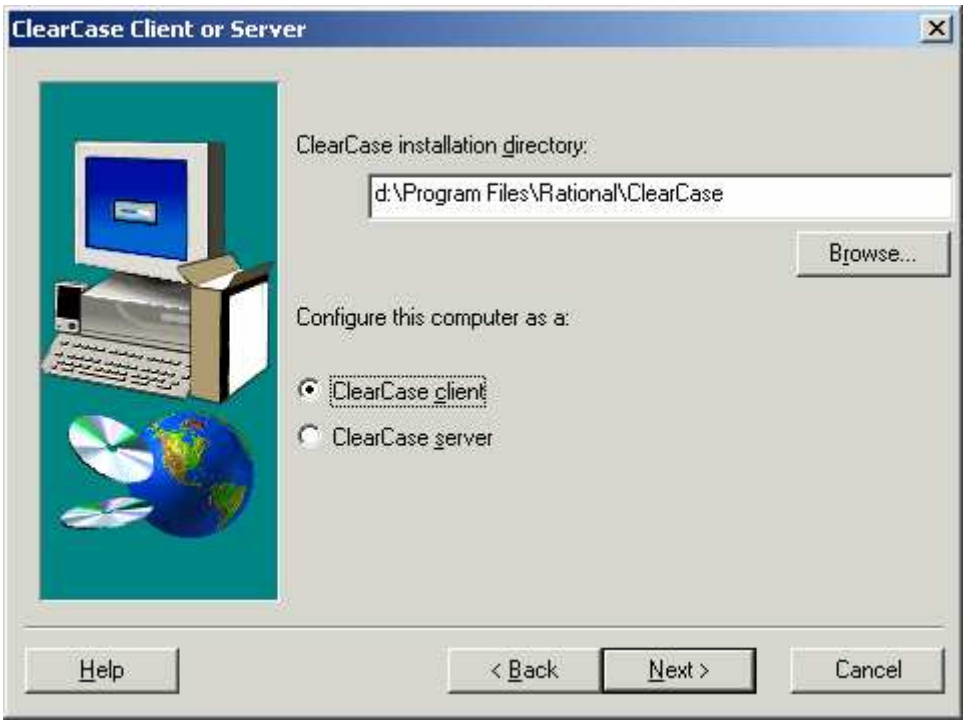
e)执行下一步



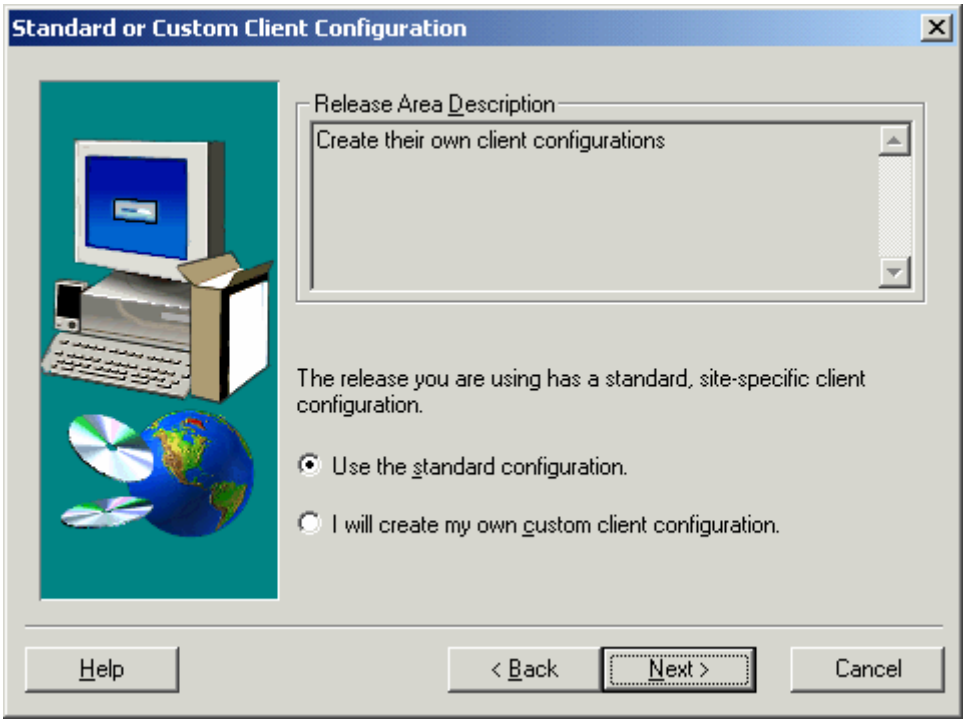
f)完成安装。重启动前需执行解密。

8. 3 Client 安装：

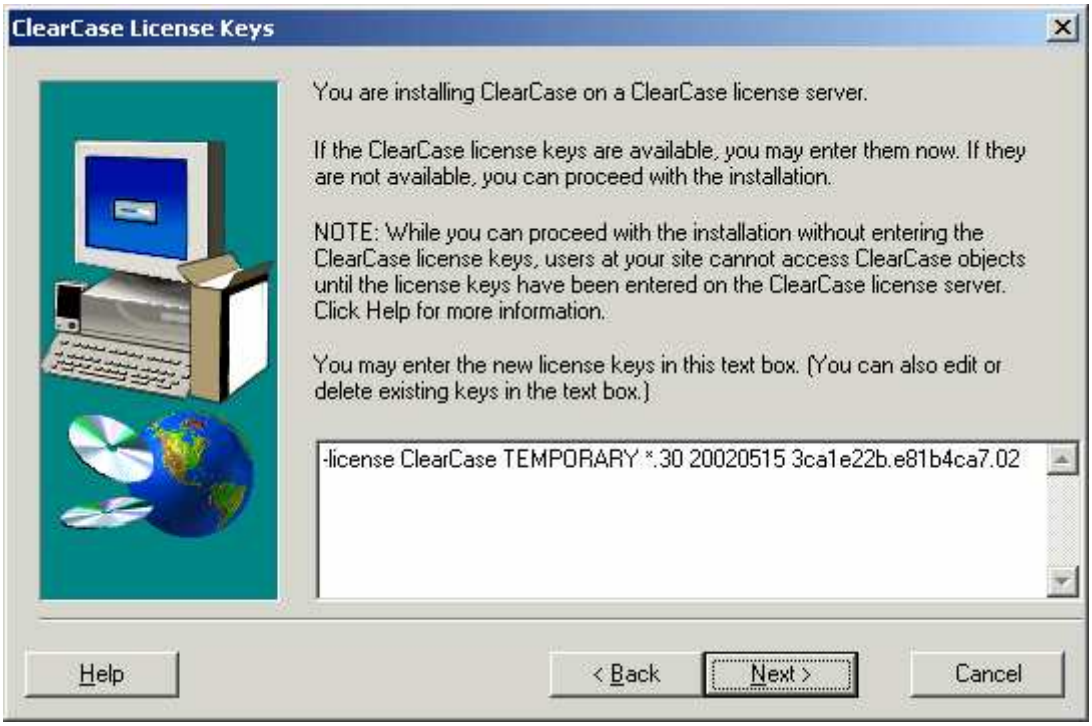
a)运行安装 setup.exe,显示如下图的安装接口选择客户端安装



b)按下图所示设置，执行下一步



c)执行下一步



d)执行下一步完成安装，在重启动前需执行解密。

9 附录 2：管理 VOB

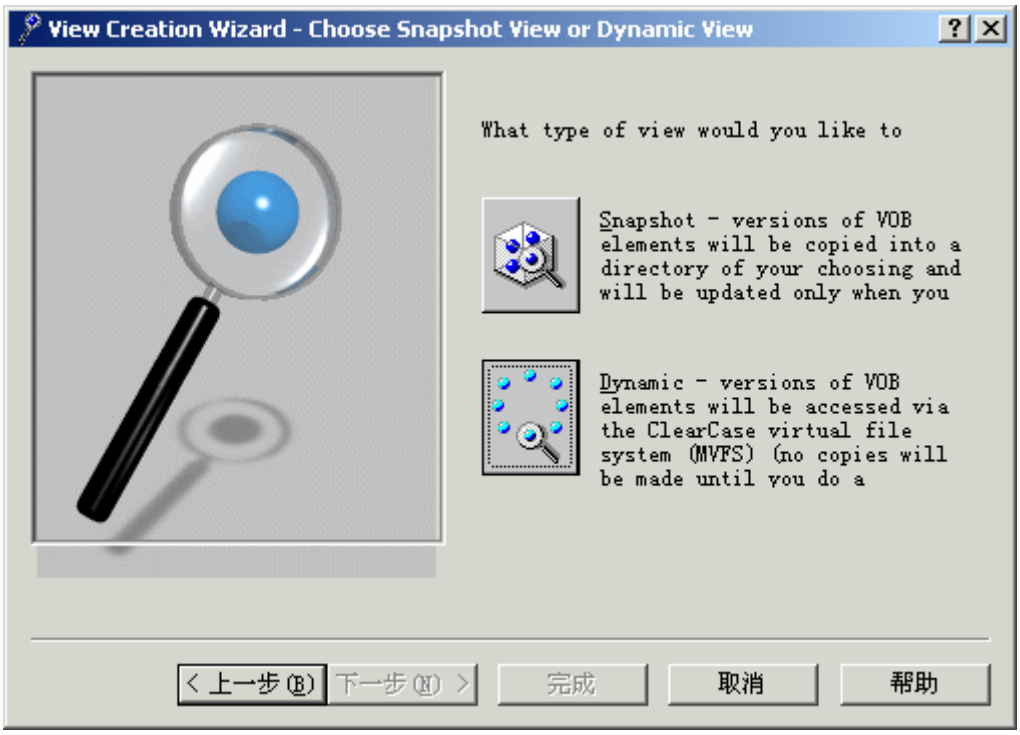
此部分内容较多，需要的话可以在软件中心支持组查阅 Rational 手册原文或译文。

10 附录 3：管理 View

1） 创建 view

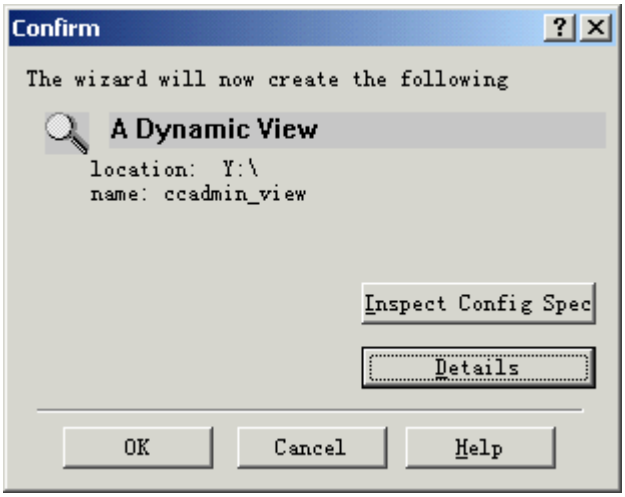
打开试图建立窗口：打开 ClearCase Explorer，点击 Toolbox，点击 Base ClearCase，点击 Creat View 打开窗口如图

选择 NO 点下一步，选择创建的视图的类型这里选择 Dynamic



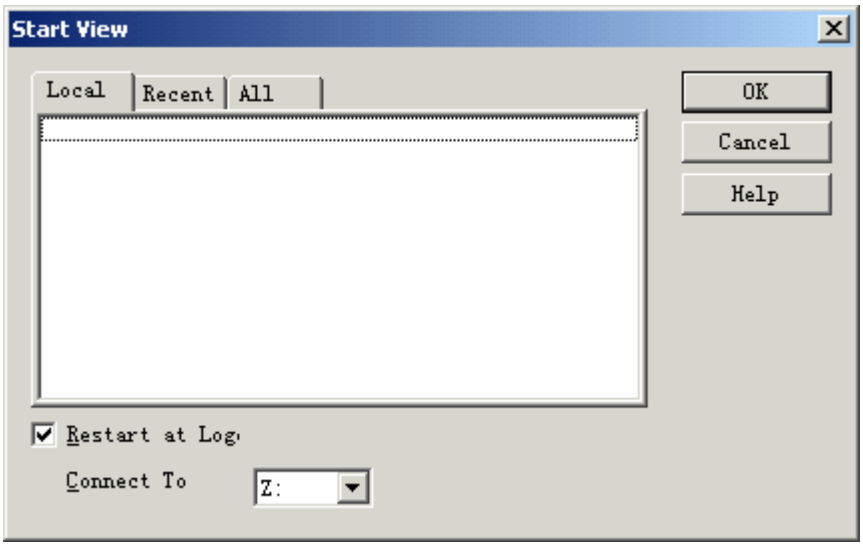
输入视图名，例如视图命名为 ccadmin_view，选择连接此视图的驱动器符如 Y:，在最后的编辑框输入必要的注释信息，在这里输入“Creat view”，注释可以为空，点完成

点 OK 完成创建。可以点‘Inspect Config Spec’和‘Details’查看相关信息



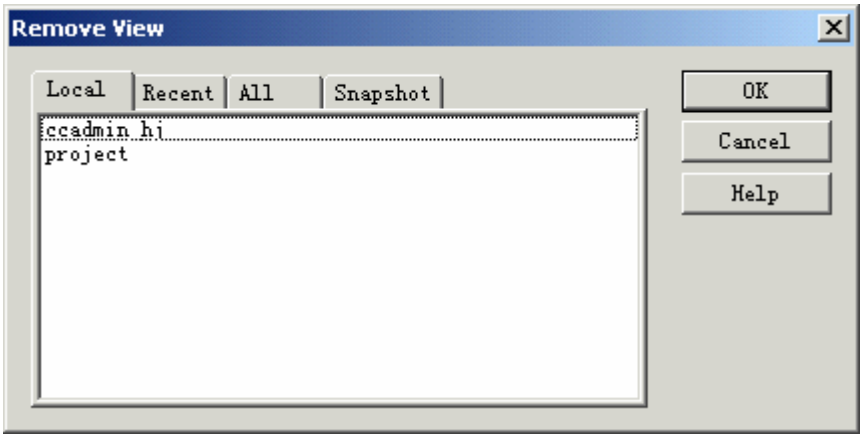
2） 启动 view

打开 ClearCase Explorer 点击 Toolbox 然后点击 Base ClearCase>Start View 打开对话框选择一个视图和驱动盘符点 OK 完成



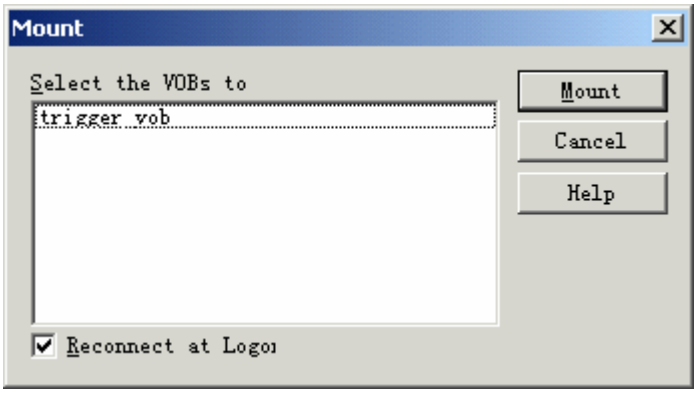
3） 删除 view

删除之前首先确保所有重要的私有文件都备份到其它位置或已经 add to source control，使用删除 view 工具，在 ClearCase Explorer 中点 Remove View 选择要删除的 view 点 ok.。



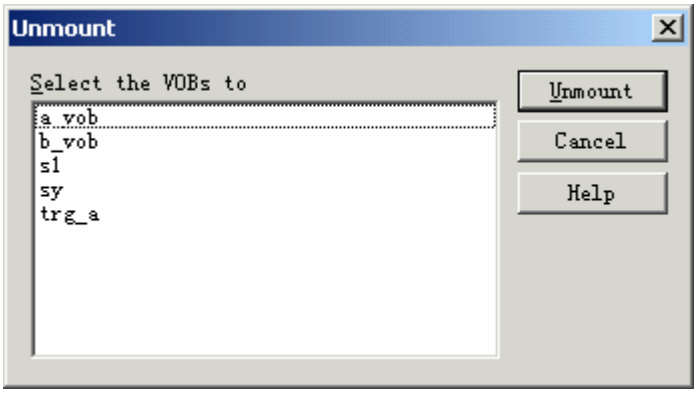
4） Mounting / Unmounting a VOB

Mounting a VOB 指连接要访问的 VOB,方法是在浏览器中选择动态视图映像的驱动器，右键快捷菜单选择 Mount VOB，选择要连接的 VOB，点 Mount 按钮完成。



当 Mount vob 后 vob 就以活页夹的形式显示在映像驱动下或 view 下。

UNmount a VOB 指断开与某个 VOB 连接方法同上在右键快捷菜单选择 Unmount VOB



11 附录 4：配置规则：

1. 配置规则语法：

0 * CHECKEDOUT

element * \main\LATEST

- **<scope>**
  指元素的类型，例如：

  element, element –file, element –directory, element –eltype text_file

- **<pattern>**
  指元素名，可以使用通配符。例如：

  ***.c， *.c， \project_a\...*.c**

- **<version-selector>**
  版本选择器

2. 相关配置规则：

缺省 config spec

element * CHECKEDOUT

element * \main\LATEST

要呈现特殊的版本的 config spec

element * REL2

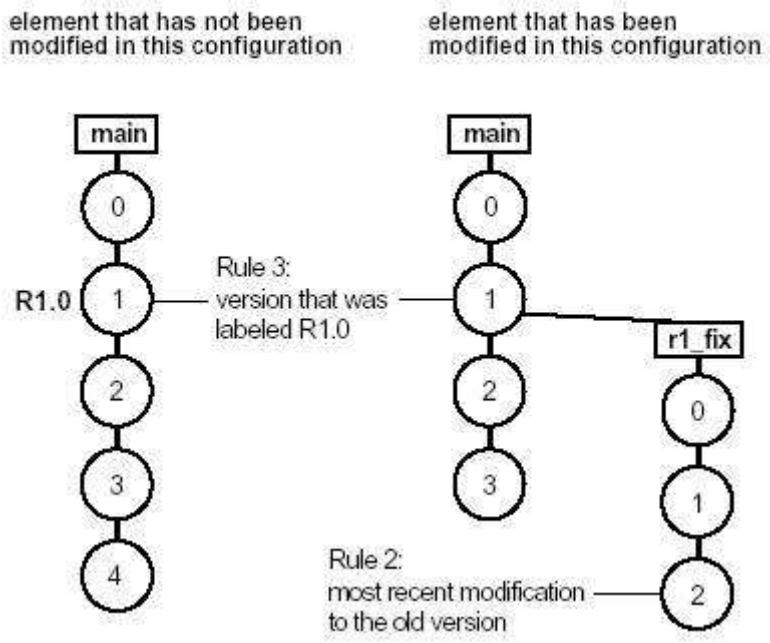
组织维护工作的 config spec

element * CHECKEDOUT

element * \main\r1_fix\LATEST

element * R1.0 –mkbranch r1_fix

Figure 34 Making a Change to an Old Version



呈现一定时间后的版本的 config spec

element *.h \main\LATEST –time now

element * \main\LATEST –time 22-Feb-96.9:00

不允许 check out 的 config spec

element * REL1 –nocheckout

使用 attribute 来选择版本的 config spec

element * CHECKEDOUT

element * \main\major\temp\LATEST

element –file src* \main\major\{QAOK= =”Yes”} –mkbranch temp

element * \main\LATEST

呈现一个用户修改的版本的 config spec

element * ‘\main\{created_by(jackson) && created_since(25-Apr)}’

element * \main\LATEST –time 25-Apr

用户既可以看到参考的文档的 main 支上的稳定版本，又能看到自己工作产品的 develop 上的对应版本，例如：编码人员不但可看自己私有分支的文件，而且也可以看发布的设计或需求文档。将发布的设计文档打上标签（如 D_RL1.0.0.1_LABEL）编码的视图规则设置如下：

element * CHECKEDOUT

element * D_RL1.0.0.1_LABEL

element * /main/develop/LATEST

element * /main/LATEST –mkbranch develop

element * /main/LATEST

12 附录 5：触发器语法和实例

1. trigger 语法:

cleartool mktrtype -element –all -preop rmelem -eltype text_file
1 2 3 4 5
-exec “ccperl –e die()” NO_RMELEM
6 7

1.进入到 cleartool 命令提示符下
2.trigger 类型 –element,-element –all, -type
3.参数 preop 表示前触发，后触发参数为 –postop
4.clearcase 操作，例如 checkin、checkout 等
5.clearcase 操作的对象类型，不包括 triger 类型
6.trigger 触发后执行的动作
7.trigger 类型名

2. 实例：

1 防止用户对所有 element 进行 checkin：

command：cleartool mktrtype –element –all –preop checkin –exec “exit 1” no_ci

结果：任何用户进行 checkin 操作时出现报错，提示 checkin error；当上述 trigger 建立后在 type plorer 中显示建立的 trigger

2 修改已建立的 trigger：

command：cleartool mktrtype －element –all –replace –preop checkin –nuser dcrowe –exec”exit 1” no ci

结果：使用－replace 参数可以修改已定义的 trigger

3 使用－type 参数可以在用户建立类型时触发 trigger

command：cleartool mktrtype －type －preop mktype －brtype －all －exec 。。。

结果：可以使用 trigger 来约束用户建立类型

4 防止用户 check out 一个文件版本时没有添加注释（用 modify_data，执行 pel 脚本程序来判断注释是否添加）脚本参见练习册 p1－12

command：cleartool mktrtype －element －preop **modfy_data** －exec。。。。

结果：可以按脚本的设计弹出提示信息

5 防止用户随意删除 element

command：cleartool mktrtype －element －all －preop remelem －exec。。。

结果：删除 element 时触发报错，如果没有定义 －nuser 任何用户均不能删除

6 trigger 作用范围

进入 vob 的不同文件目录下建立 trigger

结果：trigger 影响一个 vob 下的所有 element

7 用.bat 文件定义提示信息

结果：使用 clearpromp proceed －prompt “提示信息” －mask proceed 来实现参见教程 7－31

8 可以 lock 、raname、remove a trigger type

结果：参见教程 7－36

13 附录 7：用户权限分配和管理

CC 的权限控制是基于 windows 的用户和组，按组来设置权限，每个用户都可以属于几个组，但只能属于一个主组。

对于一个 CC 对象，其 owner 和 CC 管理员有最高权限。**注意：CC 对象的 owner 缺省情况下是该对象的创建者，但也可以被改变，如一个 VOB 的 owner 可以改变一个 element 的 owner。**

CC 管理员在 CC 中有最高权限，他能做任何操作，但为避免操作上的失误，建议 CC 管理员在做开发时用一个一般的用户帐号，只有在做管理操作时才使用该帐号。

在 CC 中，权限级别又高到低分别是：CC 管理员－>VOB owner－>element owner－>metadata type owner－>version owner－>同组的人－>其它的人。其中 element owner 和 metadata type owner 为同一级别，对一个 CC 对象的操作只有其创建者和比他权限高的人才能做。如只有 element owner 和比他权限高的人才能删除一个 element。

除 CC 自身所设置的权限外，我们还可以使用 trigger 来对 VOB 的对象操作进行控制，如只允许 VOB owner 可以删除元素，则创建一个 trigger 如下：

I:\project_hw>cleartool mktrtype –element –all –preop rmelem –nuser vobadm –exec \\VOBSERVER\triggers\no_relem.bat NO_RELEM

1 对 windows 目录的共享权限

由于 CC 需要在 windows 上建立共享目录，所以又涉及到对共享目录的权限设置，在这里，建议对于 ClearCase_Storage 目录只有 CC 用户有完全控制权限；对于存储目录下的 VOB 目录 CC 用户只有读的权限；View 目录 CC 用户有完全控制的权限。且分配权限的时候最好以组为单位，而非单个的用户。这样设置的好处是只有 CC 管理员能够创建 VOB。

2 修改 VOB 的权限

只有 VOB owner 和 CC 管理员才能修改 VOB 的权限。

cleartool protectvob 可以改变 VOB 的权限。

- **要改变 VOB 的 owner 和 group：**

C:\>cleartool protectvob –chown newvobadm \\VOBSERVER\cc_store\project_hw.vbs

C:\>cleartool protectvob –chgrp ccadmin \\VOBSERVER\cc_store\project_hw.vbs

- **要添加或删除附属组：**

C:\>cleartool protectvob –add_group doc \\VOBSERVER\cc_store\project_hw.vbs

C:\>cleartool protectvob –delete_group tempgrp \\VOBSERVER\cc_store\project_hw.vbs

我们还可以使用锁来更进一步地控制访问权限。可以锁住一个 element，也可以锁住一个 branch，甚至可以锁住整个 VOB。一个对象被锁住后，只有 CC 管理员、VOB owner 或者创建这个锁的人可以修改（他们也可以将锁打开）。

3 修改 View 的权限

改变 view 的权限只能通过命令行 chview –readwrite/-readonly 来实现，可以用 lsview –properties –full 显示 view 的信息。

4 目录和文件的权限控制

每个 VOB 里的目录和文件的权限控制都基于 owner、group 和 other，访问模式分为 r（读）、w（写）、x（执行）。**注意：对于一个 file，写权限被屏蔽掉，有没有写的权限是根据能不能 check out 来决定的。**

在这里，我还要介绍一个概念：Hijacked files。Hijacked files 指未经 check out 而被修改的文件。当一个用户未 check out 就修改一个文件后想要生成新版本，CC 会提示你这是一个 Hijacked file，此时你只有两个选择，一是将该文件 check out，然后再 check in；一是放弃所做的修改。

4.1 修改目录的权限

protect –recurse 可以改变一个目录和目录下所有的子目录和元素的权限改变，例如：

将 src 目录下的所有元素的 group-ID 改为 user：

protect –recurse –chgrp user src

4.2 修改元素的权限

protect –chmod 可以改变一个元素的权限，在这里，使用 u（user）、g（group）、o（other）、a（all）、r（read）、w（write）、x（excute）来具体指定权限。例如：

cleartool protect –chmod o+r doc.txt　即为 other 赋予读的权限

cleartool protect –chmod g-w doc.txt　即 group 的写权限取消

cleartool protect –chmod 775 doc.txt　即为 owner、group 和 other 分别赋予相应的权限。

5 查看和修改访问控制设置

访问控制工具包括：

- cleartool describe
- cleartool protectvob
- creds command：查看当前用户的帐号信息，在<ccase_home_dir>\etc\utils\creds

lsacl command：显示系统级别权限控制信息和 windows 的访问控制信息，在<ccase_home_dir>\etc\utils\lsacl 。如：<ccase_home_dir>\etc\utils\lsacl \\VOBSERVER\cc_store\GUI.vbs

15 附录 8：Administrtation VOB

ClearCase 支持使用单一管理 VOB 来管理公有数据的功能。ClearCase 在线帮助定义管理 VOB 是“一个 VOB，它含有能够复制到委托 VOB 的、必需的全局类型对象，当用户希望在委托 VOB 中创建这个类型对象的实例时”。简而言之，管理 VOB 就是其中存储了一簇 VOB 的公有数据，并且 ClearCase 将会管理这些 VOB 中的每一个可获得的数据类型的公有的（或全局的）定义。

如果你在此之前已经是 ClearCase 用户了，并且已经为你的站点建立了管理 VOB，那么 PVOB 和包含组件的 VOB 就可以使用这个已存在的管理 VOB。

> ▲ 提示：
> 管理 VOB 与委托 VOB 之间的关系是通过创建一个 VOB 至今的超链建立的。当在 Windows 上使用 VOB 创建向导时，这项工作将自动完成。使用命令行方式，这项工作就必须独立于 VOB 创建操作单独完成。

以上内容引自于《Software Configration Management Straegies And Rational ClearCase》。

任意 VOB 都能被用作管理 VOB，并且能够把其中自定义的元数据类型传递给引用它的那个 VOB。我们的管理 VOB 中包含以下自定义元数据：

16 附录 9：View Profile

使用 View Profile 能够预定义视图，包括视图规则、选择的目录（快照视图），其定义

会保存在一个目录下，共享给全组人使用。其他人要想使用，首先要在本机的控制面板上 ClearCase 服务中的 option 页面中关联到此目录上。

17 附录 10：Merge manager

Merge Manager 是一种简单易用的归并工具，以下是一些注意事项：

1、 只能应用于文件，不能应用于目录；

2、 要想使用它归并一个文件，首先必须先对其父目录进行手动归并；

3、 归并管理器的第一步需要您选择一个视图，这个视图定位在您归并的目的分支上；

4、 选择了要归并的目录后，将要让您选择归并的源分支。