



IBM Software Group

IBM Rational如何使用ClearCase UCM来开发产品

陈阅

chenyue@cn.ibm.com

Rational. software



ON DEMAND BUSINESS™

议程

- Rational ClearCase的开发环境
 - ▶ Network Appliance as VOB storage
- 源代码管理
 - ▶ 启动UCM项目
 - ▶ 补丁流和并行开发
 - 使用UCM管理补丁流
 - ▶ 构建和发布
 - BuildForge
 - ▶ 全球开发 (MultiSite vs CCRC)
 - ▶ 系统VOBs
 - ▶ 为什么需要管理工具
- 集成 ClearQuest 需求和缺陷管理系统
 - ▶ ClearCase-ClearQuest UCM
- 工程师的开发环境



Rational的产品开发



Rational ClearCase的开发环境

- VOB 信息
 - ▶ 300 公共 UNIX VOBs
 - ▶ 30公共NT VOBs
 - ▶ 8 UNIX VOB servers
 - ▶ 2 NT VOB servers
 - ▶ 100 UCM components
- VOB 存储
 - ▶ Network Appliance
 - ▶ >1TB VOB storage
- MultiSite信息
 - ▶ 75 UNIX replicas
 - ▶ 12 NT replicas
 - ▶ 9 不同different sites
- View 信息
 - ▶ 6000 UNIX views
 - ▶ 8500 NT views
- Windows -> UNIX interop
 - ▶ TAS - samba

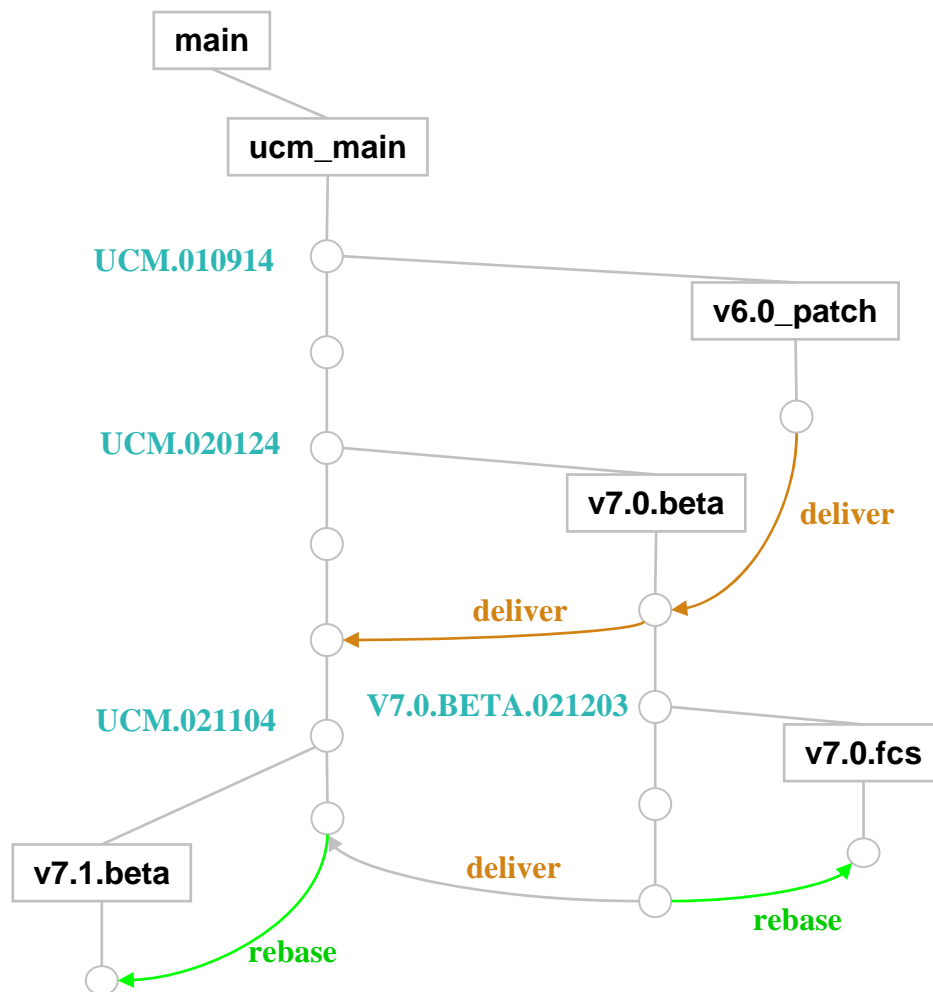


用NetApp作为VOB服务器

- 将所有VOB的存储整合成为一个大的文件系统
 - ▶ 解决了VOB过大难以存储和管理的问题
 - ▶ 通过NetApp降低了在NFS和CIFS的交易压力
 - ▶ 方便了VOB在VOB服务器间移动
 - ▶ 备份的快速和有效
 - 在UCM环境里，同步备份源代码VOB和项目VOB是非常重要的
 - 通过NetApp的快照技术实现在线备份
 - 可以在数据库的快照备份上运行dbcheck命令进行备份检查
 - ▶ 数据都通过NetApp备份，无须在VOB服务器上进行备份



源代码管理



启动一个新的UCM项目

- 创建一个新流之前要回答的问题:
 - ▶ 哪里是起始点?
 - ▶ 如何命名新的版本? (为项目、流和基线定义命名规范)
 - ▶ 流间的关系是什么?
 - ▶ 项目中涉及哪些资源, 哪些可以管理? (硬件和人力资源)
 - ▶ 不同流间的访问权限?
 - ▶ 是否需要使用复合基线?
 - ▶ 是一个新的流还是一个新的项目?



项目，流和基线策略

- 一个项目一个集成流
- 不使用缺省的命名而使用下列命名规范
 - ▶ 项目：V7.0.BETA（大写）
 - ▶ 集成流：v7.0.beta（小写）
 - ▶ 基线：V7.0.BETA.YYMMDD.component_name
- 把项目交付到公用项目上，减少流间层叠的情况
- 严格使用一个公用项目做为交付和变基的起点，不要让它和开发过程混杂
 - ▶ ClearCase开发团队使用一个叫ucm_main的流实现这个目的
- 使用复合基线来用一个基线标识整个项目



构建和发布

- 使用构建流和基线、复合基线进行构建
 - ▶ “cpf”复合基线包含ClearCase的所有组件
- 建立一个流程使所有构建过程能够获取需要的基线
 - ▶ 如：复合基线 V7.0.FCS.050818.cpf表示2005年8月18日于V7.0.fcs流上进行的构建
- 使用一个空/干净的视图进行构建，可以保证只发布检入了的文件而不至于意外将私有文件包含到发布包中



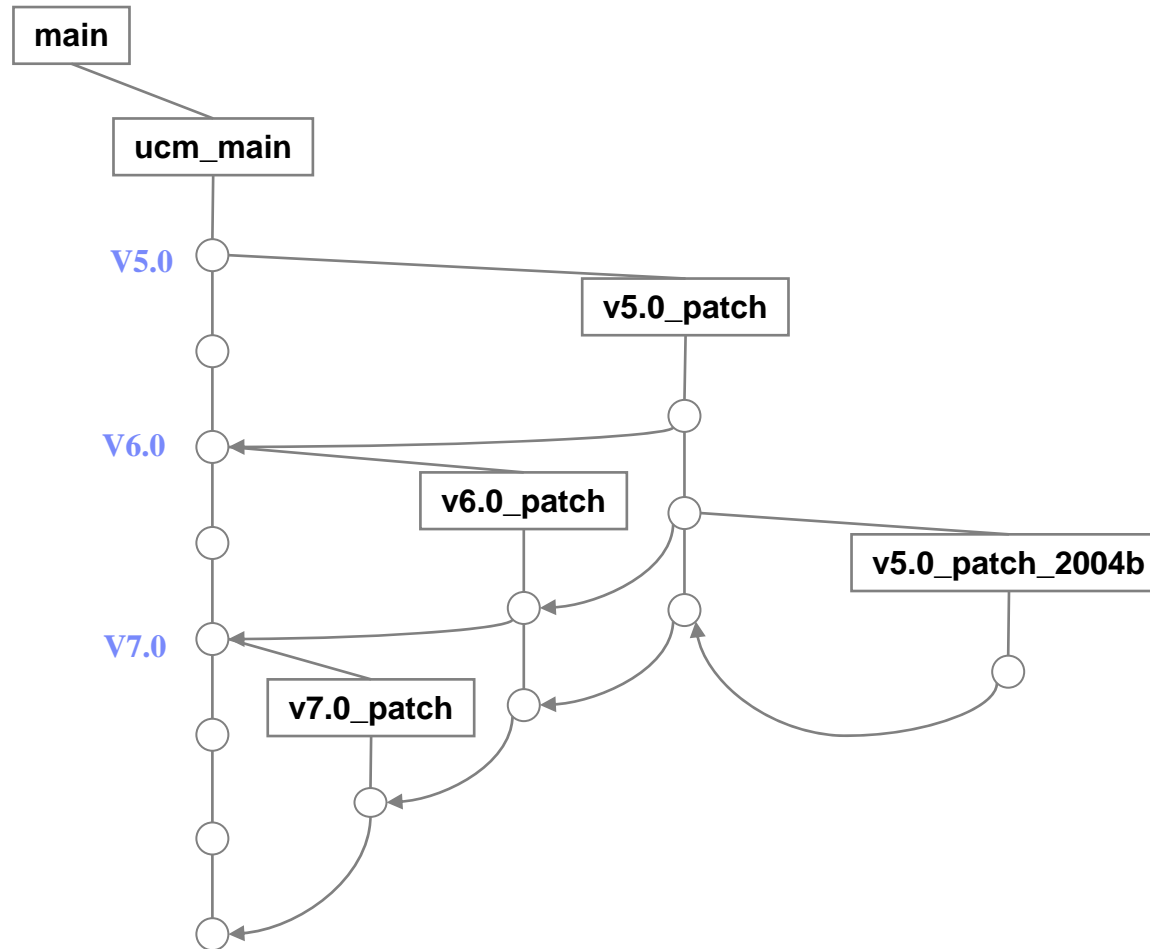
补丁和测试

- 补丁的处理和正常发布流程类似
- 流/分支和基线/标签采用类似的处理原则
- 减少发布周期
- 测试是可选的，**ClearCase**工具可以协助根据开发的情况选择测试的内容
 - ▶ `cleartool diffbl`
 - ▶ `cleartool catcr`
- 并行补丁开发越多，为了保持数据同步，每个人的工作量就越多



补丁分支的例子

- Cascading patch merges for 3 releases



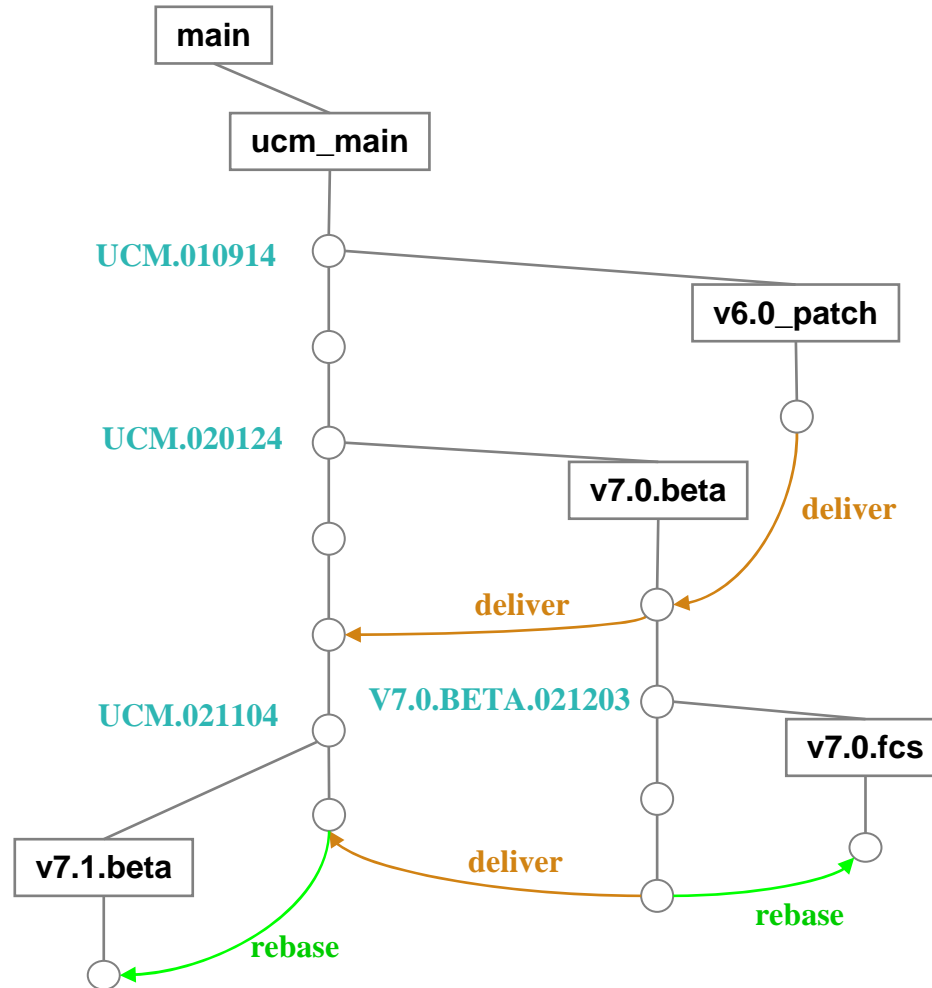
并行开发

- 从整体掌握开发流程
- 保证版本的兼容和一致
- 预先计划流的创建和合并策略
 - ▶ 每次迭代完成后都要进行交付和变基
- 确定最终流在哪里，进行集成
- 一次不要进行多个流的合并
- 目前**Rational ClearCase**要维护3个版本的补丁，同时有两个或以上的流在开发
- 同时维护核心版和本地化版本



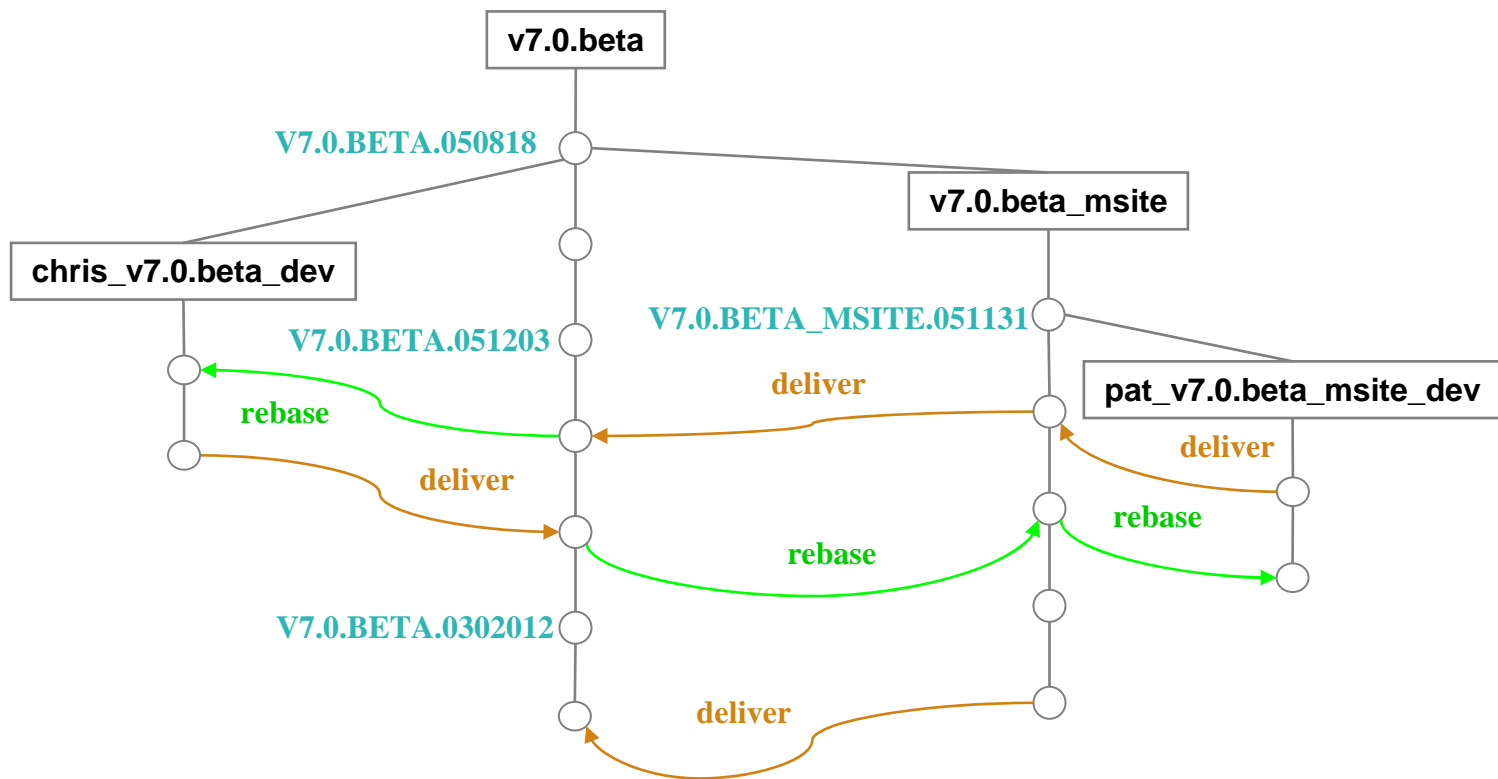
并行开发的例子

■ 3个流的分支结构



嵌套流的例子

- 从集成流衍生出的项目流



稳定流

- 隔离QE和开发
- 修复缺陷时不至于引入不需要的代码
- 在需要而不是开发结束的时候交付新的代码
- 允许多个叠代并行开发
- 对于发布团队会增加额外的构建工作
 - ▶ $N + 1$ 次构建 (N = 架构流的个数)
 - ▶ 需要更多次的交付和变基



在多种平台上构建

- Rational ClearCase 可以在10个平台上运行
- 不同的平台采用不同的view. e.g. v6.0.fcs.sun5, v7.0.beta.hp11_pa, v7.0.fcs.nt_i386
- 如果需要使用同一个视图在多个平台上构建, 则创建视图的私有目录结构
 - ▶ 例如对于lib/widget目录内的源代码, 创建lib/widget/sun5目录并且在该目录下进行构建
 - ▶ 不进行跨平台的winkin, 防止混乱
- 对于不同的操作系统, 分阶段的构建, 避免对于同一文件的同时构建



远程开发

- **ClearCase**开发团队有许多开发人员在采用**CCRC**进行远程开发
 - ▶ 从长期远程开发（**California**和**Peru**）
 - ▶ 到每周都有的在家工作的情况
- **CCRC** 相当稳定和快速
- 大部分开发所需要使用的功能都可以支持
- 大的远程站点仍然采用 **MultiSite**
 - ▶ **Bangalore**
 - ▶ **Raleigh**
 - ▶ **Cupertino**



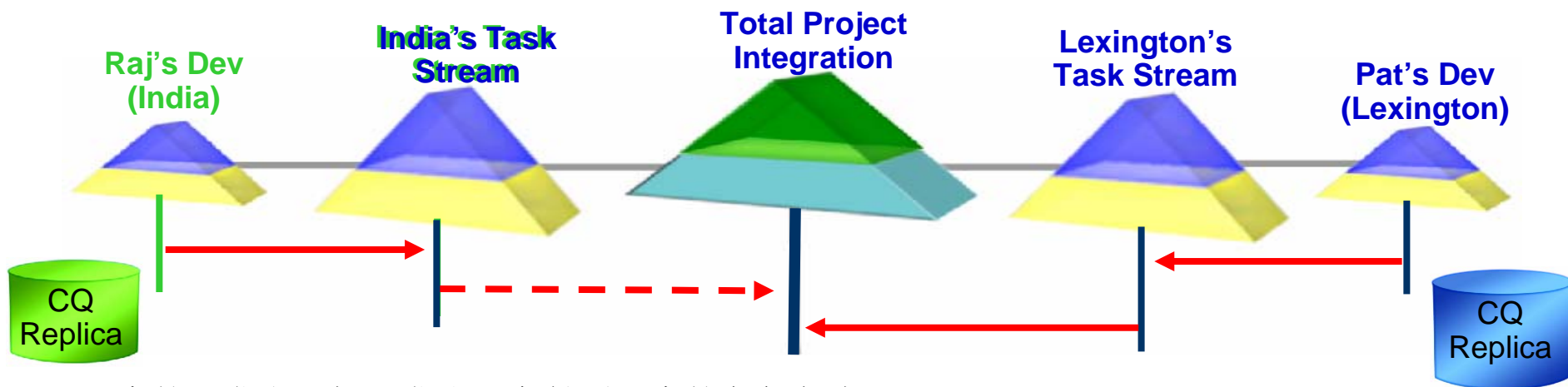
远程开发和全球并行开发

- **Rational Suite** 产品线来自超过10个产品开发组/地区
- 站点间采用 **MultiSite**同步
- 注意问题
 - ▶ 采用通用的分支/标签命名规范
 - ▶ 分割**VOB**使每个开发站点有自己的**VOB**，从而可以控制**VOB**的分支和标签类型
 - ▶ 每天运行一次‘**chepoch -actual**’，保证每个站点都有所有的包



远程开发和全球并行开发

- 每个replica有一个本地控制的子集成流
 - ▶ 使用UCM的流层次结构（推荐）
 - 在远程站点执行本地交付、构建和测试
 - 在远程站点任务流执行到集成流的远程交付



- 印度的开发人员把开发成果交付到印度的主任务流
- Lexington的开发人员把开发成果交付到Lexington的主任务流
- 印度的项目经理执行从印度主任务流到项目主集成流的交付
- Lexington同样可以执行从他们的主任务流到项目主集成流的交付

阶段版本控制库

- 阶段版本控制库(Staging VOBs) – 是什么？如何使用？
 - ▶ 阶段版本控制库是用来存储**derived objects**，构建库和可执行文件的
 - ▶ 在视图被删除的情况下仍然可以访问所有构建产品某个版本所需要的对象
 - ▶ 可以使用 '**cleartool catcr**'来查看任何已构建对象的完整清单
 - ▶ 这些VOB增长得非常快，必须经常清理
 - ClearCase开发团队使用一个脚本来删除所有未打标签的版本
 - 为了使上述脚本更有效，需要周期性的删除不重要的标签
 - 为了运行'**rmver**'命令，上述VOB都是**base CC VOBs**，因为UCM不允许用户删除已基线化的版本



阶段版本控制库

- 为什么阶段版本控制库不使用UCM?
 - ▶ 允许用户执行构建而无须为检入阶段元素而创建活动
 - ▶ 可以用与构建所使用的UCM基线匹配的标签来标识阶段元素
 - ▶ 可以很容易的在config spec内构造分段规则



构建环境

- ~builder – 它为我们做什么？
 - ▶ 所有构建都使用该用户名进行
 - ▶ 所有构建都通过builder的crontab启动，从而可以简化构建计划
 - ▶ 构建使用的视图通过定义用户组的权限来防止开发人员意外破坏生产视图，但允许发布团队的成员很容易的使用这些视图
 - ▶ 是所有发布脚本和资源文件的缺省目录



使用BuildForge统筹管理构建

- 通过web控制台自动化、一致的管理构建过程
 - ▶ 将所有环节连接起来
 - 构建基线
 - 在相关平台上开始构建
 - 成功后进行镜像
 - 实现对应构建的测试
 - ▶ 对于失败的组件进行构建
 - ▶ 通过代理在多平台上实现构建
 - ▶ 可以集成包括checkvob、dbcheck在内的CC管理任务



视图管理

- 在构建机器上创建相应的视图
 - ▶ 可以将视图存储在NetApp上
- 建立通用区存储构建规范
- 使用的视图越多,构建越慢
 - ▶ 除去没用的view
 - ▶ **Built objects**仍然存在于阶段版本控制库中, 随时可以访问
 - ▶ 通过控制台可以到查看有**DO**的视图并决定是否要删除这些视图



系统VOBs

■ 系统 VOBs

- ▶ 包括操作系统headers、库、编译器及构建所需的其它工具
- ▶ 除了VOB之外，构建还需要基于这些对象进行，因此ClearCase开发团队利用此库版本化构建所需的操作系统和编译器
- ▶ 可以实现跨机器的构建的移动
- ▶ 更加方便测试以及回退
- ▶ 这些对象在 UCM项目中是只读的



补丁版本的生命周期

- 补丁版本其实是一个压缩了的版本周期
 - ▶ 接受交付
 - 补丁的内容由变更管理委员会(CCB)管理
 - 补丁的内容可通过CQ中的补丁请求监控
 - ▶ 内容冻结
 - 创建稳定流
 - 下一补丁(n+1)接受交付
 - ▶ 稳定流
 - 稳定流完全由CCB控制
 - 前面的内容继承到后续的补丁
 - 前面的内容继承到后续的版本
 - 开始下一补丁的开发

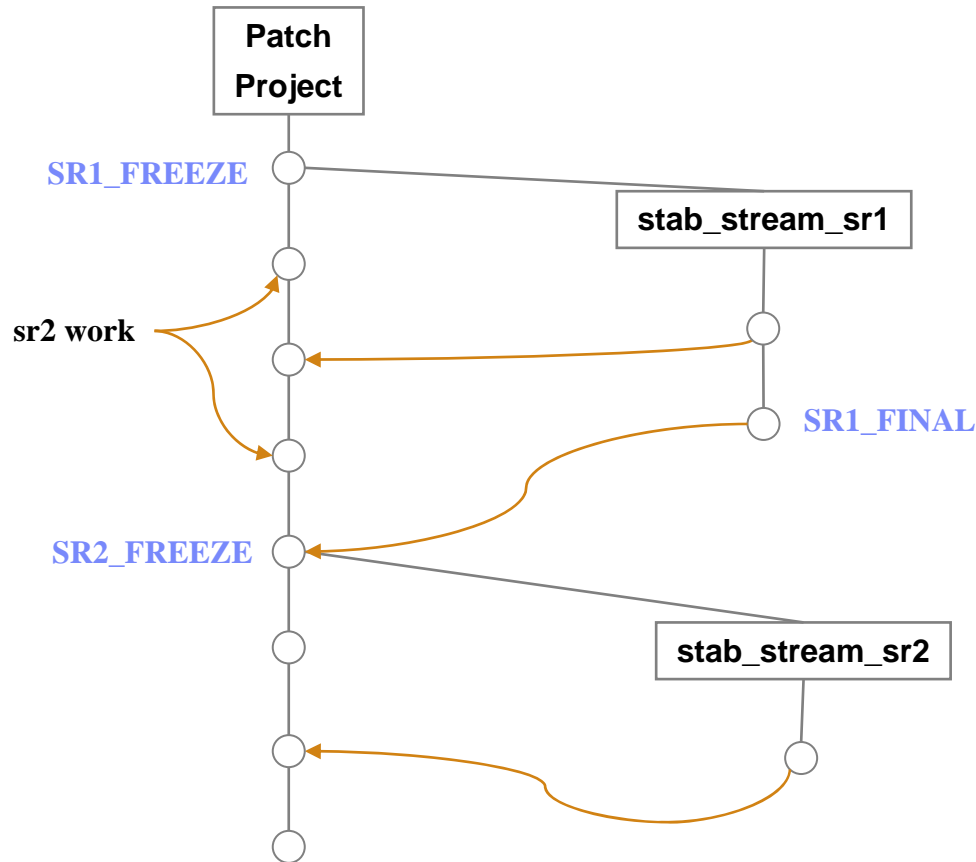


补丁版本的生命周期

- ▶ 为内部部署做准备
 - 允许开发人员在正常测试进行的过程中开始对下一补丁版本进行测试
 - 帮助开发人员在将补丁发布给客户之前发现产品的问题
- ▶ 校验构建对象
 - 在已发布的构建对象上运行**catcr**，以保证他们已包含开发交付的所有变更集
 - 检查交付到补丁流上的变更集，确认他们包含了适当的控制标签以保证构建的可重复性
 - 使用**diffbl**使**QE**了解每个补丁构建之间的变更
- ▶ 确认变更已合并到所有流
 - 使用**cleartool deliver -preview**保证所有最近的开发活动都已经合并到主补丁分支和下一个主要的发布版本
- ▶ 把内部部署机器升级到补丁的生产版本
- ▶ 创建最终**UCM**基线
- ▶ 把补丁发布给生产



Patch Project and Stabilization



使用UCM进行补丁管理

- 使用UCM管理补丁流
 - ▶ 每个Release一个UCM项目
 - 每个补丁请求一个UCM活动
 - ▶ 每个补丁版本一个稳定流（由CCB控制、锁定）
 - ▶ 每个补丁版本一个主要UCM基线
 - 标记最终发布的内容
 - ▶ 补丁发布是有依赖性的
 - 所有前面的内容继承到后续的补丁版本



使用UCM进行补丁管理

- 开发人员在多个**UCM**项目中工作
- 许多补丁同时包含两个以上补丁版本的内容
 - ▶ 先在一个流上进行开发
 - ▶ 将变更（通常是向前地）合并到其他分支上
 - ▶ 将最终的补丁流向前合并到下一个主要版本
- 先在开发流上工作，再交付到集成流
 - ▶ 交付必须在**CCB**批准以后进行
- 先在子开发流上工作
 - ▶ 开发人员从稳定流创建自己的私有流
 - ▶ 私有流使开发人员可以在提交**CCB**之前检出和测试他们的变更，因为此时稳定流是锁定的

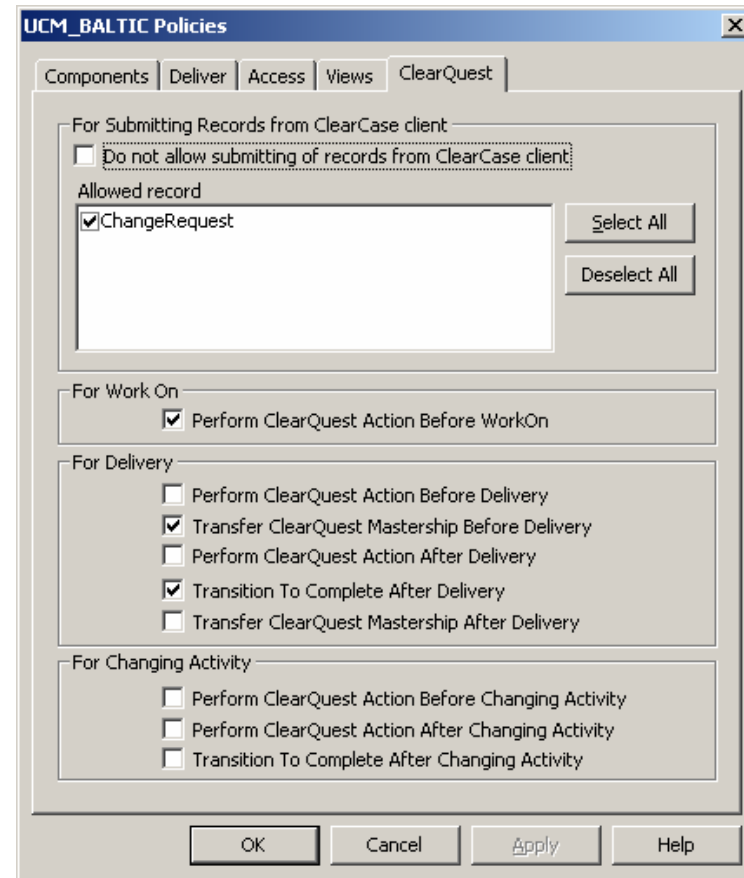
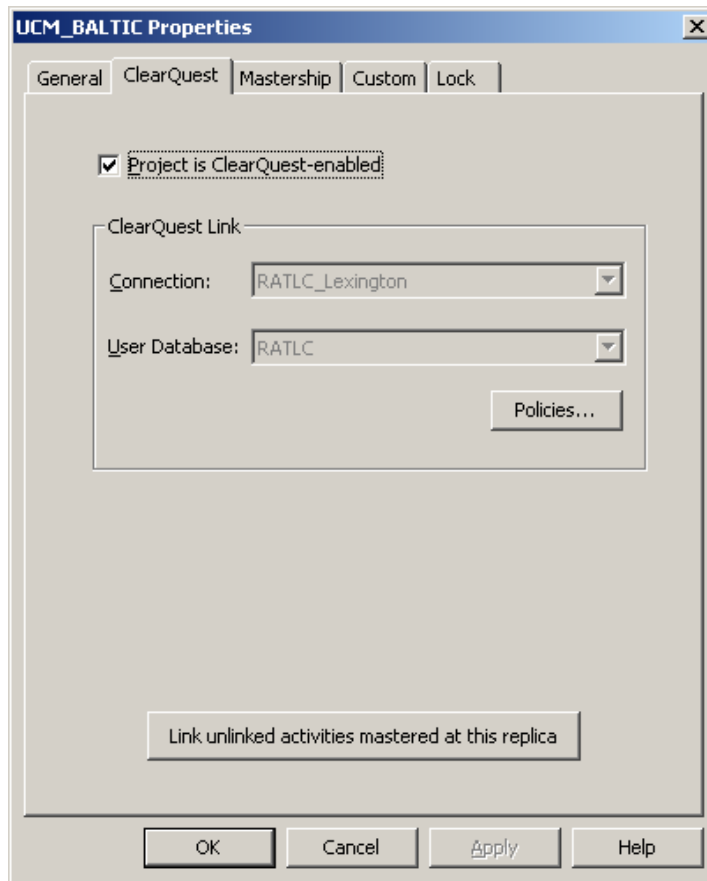
使用UCM进行补丁管理

- 一个UCM活动对应一个版本上特定问题的补丁活动
 - ▶ UCM活动的变更集提供
 - 补丁管理
 - 代码走查
 - 质量保证和测试
 - 基于活动区分基线的内容



集成 ClearQuest 需求和缺陷管理系统

- 通过UCM集成ClearCase和ClearQuest RATLC



集成 ClearQuest 需求和缺陷管理系统

- ClearCase中可以直接访问ClearQuest的内容

Exploring ClearCase Projects

File View Tools Help

yyin_UCM_BALTIC

ccprojects

- ClearCase
 - BALTIC
 - ematcheyTest
 - Experiments

Name	Owner	Created On	ID	Activity State
C3: SQUID operation causes segmentation fault	yyin	10/12/2005 02:01:53 PM	RATLC00756171	Closed
C4: Can't add double quotes to headline assoc. w/ a UCM a...	yyin	11/28/2005 10:16:01 AM	RATLC00763617	Closed
C4: Finish Activity in CQ-enabled context should clear activi...	yyin	10/12/2005 01:51:17 PM	RATLC00759445	Closed
C4: Unable to create new UCM activity from xclearcase	yyin	11/07/2005 11:54:33 AM	RATLC01102948	Closed
			RATLC00763075	Closed
			RATLC00763040	Closed
			RATLC00760919	Closed
			RATLC00761699	Closed
			RATLC00761641	Closed
			RATLC00760978	Complete
			RATLC00763824	Complete
			RATLC00691980	Closed

View ChangeRequest RATLC00756171

History Configuration Unified Change Management Patch Other Systems

Main Customer Doc and Triage Resolution APAR Notes

Headline: C3: SQUID operation causes segmentation fault

IBM Confidential ID: RATLC00756171 State: Closed Status: Fixed indirectly

Change Request Type: Defect Defect Severity: DS1 Business Priority: BP1 How Was Problem Found: Functional test

Product: ClearCase Found In Version: Baltic Component: UCM CQ

SubComponent: Found In Platform: Linux RedHat Owner: pchyu

Found In Packaged Product:

Found In Iteration:

Impacted Products:

Found in Build(s): baltic.D050829

Found in Build(s) UCM: Add

Description:

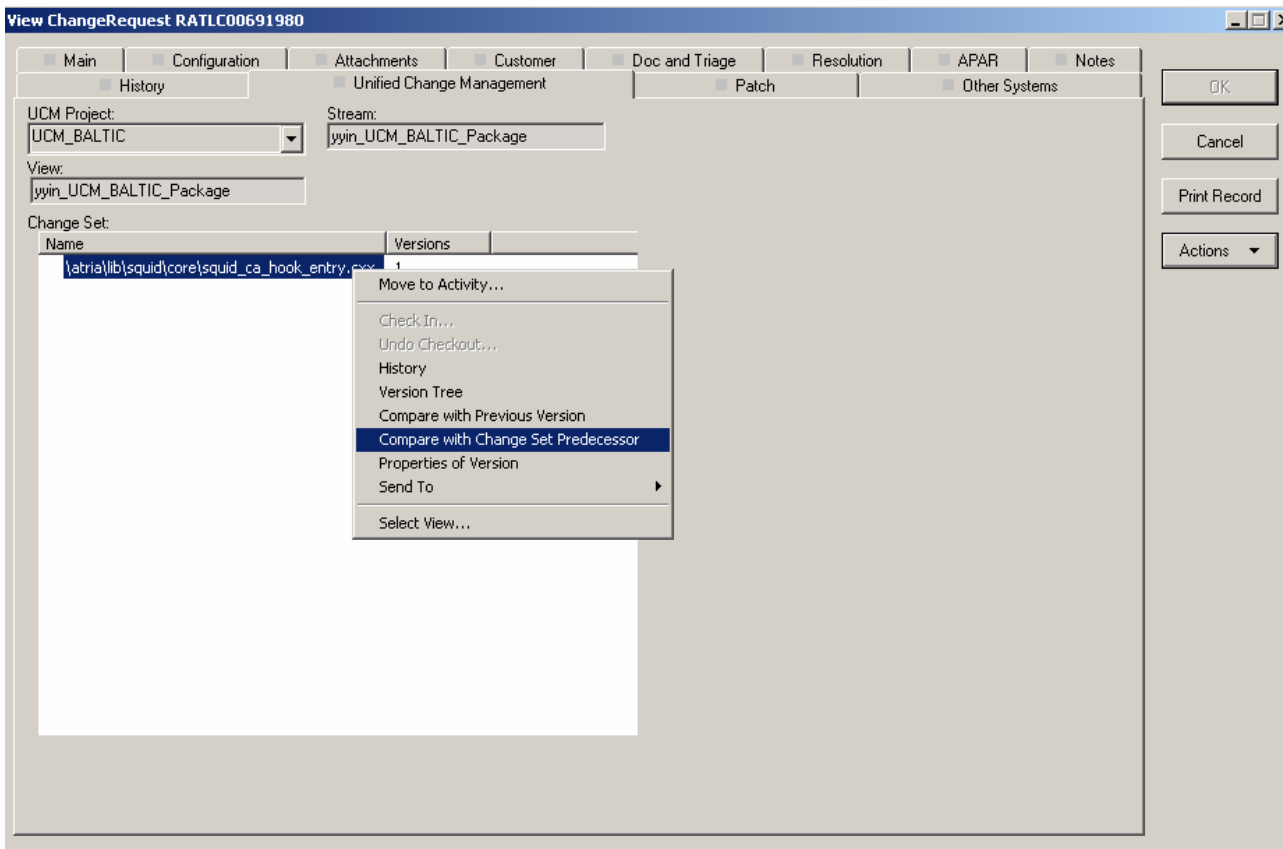
Squid operation "chactivity" is causing Segmentation fault but core file could not be located or it does not exists.
The same ATE test cases ran fine on SUN and NT but it seems to happenning consistently on Linux.

Sorry about the readability but the pasted lines shows the flow and segmentation fault.

```
=====
qvm1038 /var/tmp/vob2_basic_ucm_squid_accept_MEJ91 > ct setactivity
SAMPL00000049
Set activity "SAMPL00000049" in view "view6_SingleInt_ucm_squid_accept_MEJ91".
qvm1038 /var/tmp/vob2_basic_ucm_squid_accept_MEJ91 > ls
lost-found SAMPL00000048_file1.txt SAMPL00000048_file2.txt
qvm1038 /var/tmp/vob2_basic_ucm_squid_accept_MEJ91 > ct co -nc .
Created SAMPL00000048 from SAMPL00000049
```

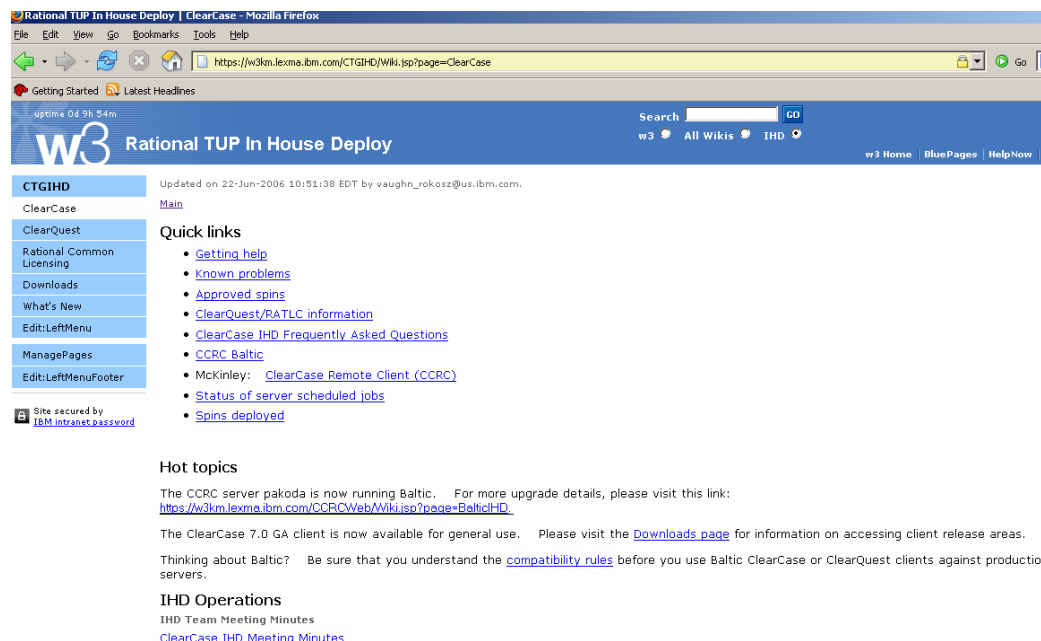
集成 ClearQuest 需求和缺陷管理系统

- 在ClearQuest RATLC中查看UCM变更集和执行ClearCase操作



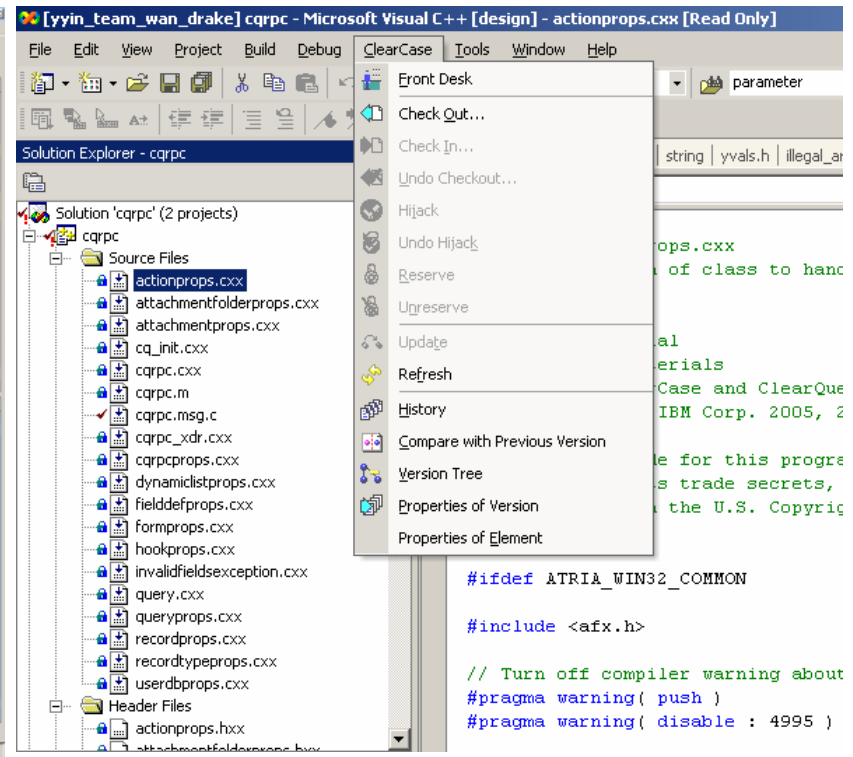
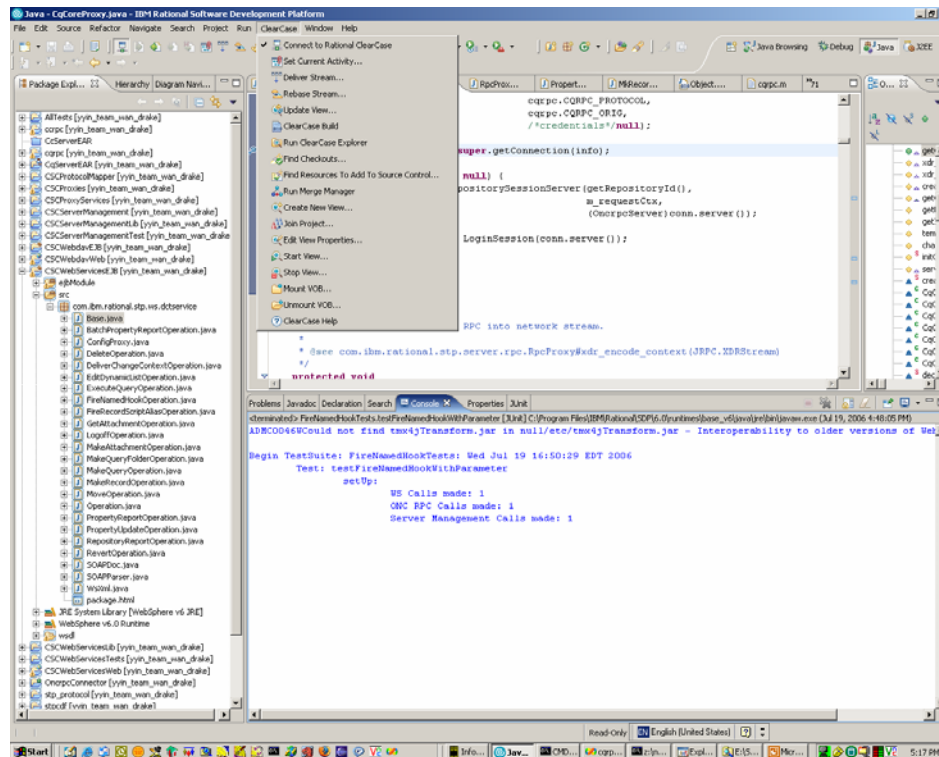
内部使用和部署

- Clearcase开发团队有使用Clearcase开发Clearcase的传统
- IHD (In House Deploy)使我们自己成为自己的用户
- IHD是帮助我们更容易发现问题、保证产品质量的一个重要过程



工程师的开发环境 – ClearCase和IDE集成

■ RAD, VS.Net和ClearCase集成





Questions

