

ClearCase四大功能详述（版本控制）PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/141/2021_2022_ClearCase_E5_c29_141562.htm

ClearCase的核心功能是版本控制，它是对在软件开发进程中一个文件或一个目录发展过程进行追踪的手段。ClearCase对所有文件系统对象（包括文件、目录和链接）增强了版本控制系统功能。可定版本的文件包括源代码、可执行文件、位图文件、需求文档、设计说明、测试计划、和一些ASCII和非ASCII文件。目录的版本记录了整个组织基础资源的发展状况，包括源文件的建立、重新命名、重新构造和删除操作等。这种版本控制系统提供了先进的版本分支和归并功能用于支持并行开发。控制任何文件的版本

ClearCase可以对每一个软件组件或元件的版本进行维护和控制。ClearCase也可以维护一个非文本文件、目录和工具的版本。正如：它可以管理库文件、编译器、需求文档、测试包和数据库而不仅仅是源代码。ClearCase的元件类型可以管理版本内容。用户可以定义自己的元件类型，也可以使用ClearCase中的预定义类型：文本文件、压缩文本文件、文件、压缩文件和二进制增量文件。ClearCase可以利用增量算法将文本文件存储在一个特殊结构的文件容器中。ClearCase采用标准的压缩技术和增量算法存储一个压缩文本文件。（这比以往的存储形式节省了50%—70%的存储空间。）这种元件类型文件和压缩文件可以被用于控制任何操作系统文件

比如，可执行程序、程序资源库、结构数据库和结构文档文件。二进制增量文件类型可以随时被用于二进制文件格式。在版本树中组织元件发展的过程在ClearCase中，元件版

本的组织体现在版本树结构中。一个版本的结构可以按目录结构定制，还可以包含多层分支和子分支。在一个典型的开发环境中，很多元件的版本树结构最初仅包含一个分支，即，元件的版本排列在同一条线型队列中。随着时间的发展，当用户做一些错误修复、代码的组织、一些实验性修改或指定平台的开发时，它们可以给一些相关元件定义子分支，从而脱离主干进行开发。ClearCase可以支持多级的分支操作，还可以给版本或分支命名。对目录和子目录进行版本控制

ClearCase可以对目录和子目录进行版本控制，允许开发者对他们数据的组织发展过程进行追踪。目录版本对一些改变进行控制，如：建立一个新文件、修改文件名、建立新的子目录或在目录间移动文件等。ClearCase也支持对目录自动进行比较和归并的操作。存储数据在一个可访问的版本对象类中（VOBS）ClearCase把所有版本控制的数据存放在一个永久、安全的存储区中，这个存储区被称为版本对象类（Version Object Bases），项目团队（或管理者）可以决定它们所需要的VOBs的数量，可以决定什么样的目录或文件需要被维护。VOBs不仅是一个可连接的文件系统而且也是网上的资源

主机可以连接任何数量的VOBs。ClearCase VOBs的组成模式跟UNIX、Windows NT的文件系统和分布式的数据库系统非常类似。ClearCase采用Raima数据管理机制维护VOB数据库。当在ClearCase中连接和访问时，VOB象一个标准的软件作为目录树的形式出现在客户面前，包含标准的文件对象：目录、文件、符号链接和硬链接。但事实上，文件系统已经有广泛的版本控制组件：它包含目录元素、目录元素版本、文件元素、文件元素版本、VOB动态链接和VOB硬链接。开发

者也可以查看和这些文件系统对象相关的数据。这些数据包括事件记录，建立审核以及用户定义的项如：版本标签和属性。使用常见的检出/编辑/检入范例 ClearCase的命令可以控制元素的变化，确保存储区有序的繁衍并使数据损坏的程度达到最小。ClearCase采用一种检出/编辑后检入的范例，类似于传统的版本控制工具如：RCS和SCCS。ClearCase除了可以进行检出、检入以及非检出操作外，它还可以通过命令设置另外的操作，如：删除版本、建立/删除分枝、可按时间顺序排列或结构排列顺序列出版本历史、比较版本间的差异，并且可以归并并行开发的版本。当开始对于一个指定的文件进行工作时，该文件具有只读属性 这意味着它不能被编辑或删除。而检出操作可以对该文件的最近版本形成一个可编辑的拷贝。它无须将文件拷贝到另一区域工作。检出的注释可以被提供。当编辑完成后，该文件被检入，于是在版本树中形成一个新的版本并且将可编辑的拷贝删除。为了检验文件的变化，在检入过程中可以填入注释信息。文件一旦被检入，即刻回复到只读状态成为共享数据，可被所有成员使用。ClearCase支持两种检出，保留以及非保留。保留检出可以保证版本历史形成的正确范围，并且同时只允许一个人做保留检出的操作。非保留检出无须保证建立一个成功的版本，如果多个用户同时对同一元素执行非保留检出，也企图进行检入操作，那么第一个检入操作被允许，而其他用户必须通过归并操作合并它们的结果。丰富的注释信息和版本数据的报表 ClearCase存储了和文件系统对象相关又截然不同的信息类。这些信息实际上并不包含在对象中，它是一些额外数据。这些数据可以由ClearCase产生，也可以由用户自己定义。

在VOB数据库中存储了所有的数据。 ClearCase产生的这种数据信息提供了可靠的、面向文件系统的版本注释信息。比如：这些数据可以验证在某一时刻，元素A建立了一个新的版本。用户定义的数据可以用来表达额外的功能 比如：该文件的版本曾被用于构造应用系统的4.31版。 ClearCase的操作（如：检出、检入、和版本归并）可以建立时间记录，记录数据包含这些操作信息。这些记录被存储在VOB数据库中，主要描述了该操作的属性"谁做的、做什么、什么时候、在哪个地方及为什么"，比如：敲入命令的人员的ID号，操作的种类，操作的时间，主机名称及用户填入的描述。可以通过"lshistory"的命令显示存储在VOB中的事件记录，并且可以通过历史信息浏览器提供的图形接口观察VOB中的事件记录。用户可以针对多种目的定义数据，包含分支的名称、版本标签、元素任一版本的注释信息。 ClearCase数据的另一种应用是形成注释的文本文件。注释命令可以通过行显示的形式列出任何一个版本文本文件的内容，这使得我们可以更容易的看到什么时候在不同的地方做了添加或删除的操作。 ClearCase也可以针对文件系统对象建立客户报表。而报表的种类可以由用户自己定制输出格式。通过分支功能支持并行开发 ClearCase支持并行（同时）开发，每一个元素都可以沿着不同的分枝同时发展，即新的版本加到独立的分支上。 ClearCase可以很容易的产生分支，也可以很容易的将不同分支进行合并。这样一来，即便某一部分的工作被冻结或加锁，开发者仍然可以继续自己的工作（如：在软件集成期）。在这种情况下，开发者可以在分支上工作，我们知道， ClearCase的自动化操作和图形归并工具可以让我们很容易的

重新集成新的工作。并行开发是非常重要的，因为：（1）它允许不同的项目在同一时间使用同一资源树。（2）它将目前不可和其他人员共享的修改成果进行隔离。（3）它将绝对不可和其他人员共享的修改成果进行隔离（如：已发布版本中的错误修复）。（4）它使得在软件集成期间开发工作无需停止，程序员可以先在分枝上开发，以后再集成。为了支持并行开发，ClearCase允许进行分支建立，追踪分支的使用，文件比较，自动归并功能。自动的比较和版本间的归并并行开发的特点是对同一元素的不同版本进行定期比较，也需要对版本间内容进行归并。在ClearCase中，对于元素或文本文件进行比较和归并的操作有两种：基于字符型和图形界面型。其中，diff命令执行多文件比较，不执行归并。而归并命令可以处理32个“成员”，并把它们生成一个独立的文件。ClearCase可以自动辨认归并选项并实现归并。ClearCase也可以对需要归并的项目元素进行定位。如果所有的“成员”（归并元素）是同一元素的版本，系统会自动确定基础“成员”，通常是最低版本。此外，ClearCase会记录基础版本和某一归并元素版本间的差异。如果，所有的“成员”间差异互不相同，ClearCase会自动建立归并版本。如果两个或多个归并“成员”文件内容部分不同，归并功能会提示开发者选择归并内容。ClearCase也可以实现反向归并——从主分支向子分支归并。ClearCase的加归并功能可以在归并其它分支时选择指定的版本（那些在分支上自始至终进行变化的版本）。负归并操作可以删除部分版本差异，从而形成一个新的版本，该版本除了那些被删除的变更外包含所有的改变。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com