

Deploying Highly Available IBM® Rational® ClearCase® On Linux

Ready for

IBM | Rational.

software

Abstract

Linux has become a primary platform for both hardware and software development organizations. Protection of software assets being managed on Linux development platforms is a critical function of corporate IT. IBM Rational ClearCase provides a complete set of tools for managing the complete development lifecycle for software products. Deploying a highly available Rational ClearCase environment provides the benefits of a powerful integrated and distributed development environment with the assurance of protection against loss of data or productivity that can result from planned and unplanned outages.

Introduction

In the past five years, Linux® has gained wide support from major corporations such as IBM, Oracle and Novell for use in business-critical environments and is now a Tier-1 operating system at every major server hardware and software provider (with the notable exception of Microsoft). Linux has also emerged as a platform of choice among IT shops due to its scalability, cost-efficient hardware choices and command set similarity to wellknown UNIX environments. Software development tasks which have been hosted on UNIX platforms are migrating to Linux to leverage these advantages as well as the emerging set of next generation development platforms which are being developed by the Linux community, such as the GNU toolset and a host of JAVA-based development, build and debug tools.

IBM Rational® ClearCase® is a leading solution used by development organizations as they leverage Linux to build, integrate, modernize and deploy software projects. Rational support for Linux is currently focused in two areas: facilitating the development of Linux applications through comprehensive IDE support and reducing the total cost of ownership for software development infrastructure by providing Linux based platform support for managing software assets. With support for all of the roles and activities within the software life cycle, IBM Rational products allow organizations to perform all aspects of product development on Linux with the software asset management capabilities provided by Rational ClearCase supporting teams of any size, from small, co-located teams to globally distributed teams working in parallel. The solutions are available on Linux running on diverse platforms from PCs to mainframe computers.

Rational ClearCase provides complete life cycle management and control for software development projects. Through integrated version control, automated workspace management, parallel development support, baseline management, and build and release management, Rational ClearCase provides the capabilities needed to create, update, build, deliver, reuse and maintain business-critical assets. Rational ClearCase can help increase productivity through parallel development, reduced build/release cycle times, and increased software reuse. Integration with leading IDEs including Rational Application Developer, WebSphere Studio, Microsoft Visual Studio .NET and the open source Eclipse framework further streamlines development.

Rational ClearCase also supports geographically dispersed development teams via its MultiSite feature. With MultiSite, Rational ClearCase can scale from small local development teams to large worldwide enterprise organizations.

Local, remote and web interfaces for Rational ClearCase enable access at anytime and from virtually anywhere. Support for development on Linux, Windows, Unix and mainframe (z/OS) systems provides a broad range of choices for application and build environments and allows organizations moving to Linux to easily migrate from existing platforms and receive all benefits of a Rational ClearCase development environment.

Configuring Rational ClearCase on Linux

The IBM document, Rational ClearCase Install Guide for Linux, provides detailed instructions on installation within a Linux environment. Rational ClearCase supports both 2.4 and 2.6 kernel versions as distributed via RedHat, RHEL 3 and 4, and Novell SUSE, SLES 8, 9, and 10. Due to differences in the kernels on which the Linux distributions are based, the steps required to configure the kernel for Rational ClearCase installation, primarily to re-build the Linux vnode layer, must be followed based on the specific Linux distribution used. Rational ClearCase on Linux supports platforms built on Intel x86, AMD64, Xeon64bit/EM64T, s390 and POWER chipsets. While all IBM BladeCenter, System i, System x, and System p5 platforms which are certified with Linux are supported with Rational ClearCase 2003.06.x, not all systems are supported



on all Linux distributions. Refer to Supported Rational ClearCase Family Releases for specifics on the combinations available. IBM System x Servers such as the x3850, x3755, x3650, and x3550 combined with Linux and Rational ClearCase provide a highly scalable, reliable and affordable solution for application development in organizations ranging from the Fortune 500 to sole proprietorships.

The Importance of Protecting Software Assets

Software assets are development organizations' most precious resources. As such, the source code, development environment, build tools and the resulting binaries which make up these assets must be protected. Protection against data loss and against downtime resulting from both planned and unplanned outages is critical to an organization's continued productivity, success and longevity.

There are numerous ways to back-up and protect the information you have stored in a Rational ClearCase repository on Linux. Using standard back-up to tape archive procedures, you can store away information securely so that in the event of a serious failure, you can use these archives to restore your repositories. However tape-based restoration can take significant time and will result in loss of all data since the last back-up was taken.

If Rational ClearCase availability is critical to your organization, using an approach which includes real-time continuous data protection and/or high availability clustering to maintain a redundant replica enables you to restore normal operations faster and, if using data replication for continuous data protection, a catastrophic outage will cause lost data only since the last replication (typically only minutes) instead of the last back-up (typically the previous day or before). A high availability solution can result in dramatic decreases in lost Rational ClearCase data and dramatic increases in developer productivity.

This white paper explains how you can build a highly available Rational ClearCase environment to enable quick switching to a back-up server to prevent end user downtime in the event of either planned or unplanned outages of components in the Rational ClearCase architecture. SteelEye Technology's LifeKeeper Protection Suite for Rational ClearCase is shown as one example of a high-available clustering solution that can be used to build a redundant environment. Other solutions for providing Rational ClearCase high availability on Linux exist, but none have specific monitoring and recovery modules built for Rational ClearCase as exist in LifeKeeper. The LifeKeeper Protection Suite for Rational ClearCase is the first high availability clustering product to receive "Ready for Rational" certification.

Rational ClearCase High Availability

Rational ClearCase is made up of several independent yet interacting components as represented in Figure 1 below. While these are shown as separate physical servers, the services are often consolidated in a Rational ClearCase configuration. So, a single server can host one or more of the functions depicted.

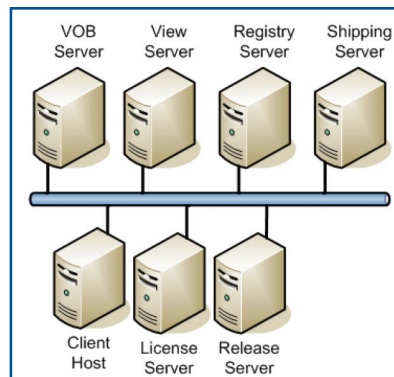


Figure 1: Rational ClearCase Components

Rational ClearCase requires several network wide resources: a license server host (one or more), a registry server host (1 plus an optional back-up) and a release host. If Rational ClearCase is used in a geographically dispersed environment via the use of MultiSite, then a shipping server is also required. Additionally, the configuration can contain a number of client hosts and server hosts. The function of each of these critical system components and the requirements for high availability protection is described below:

License server host – Most functions in Rational ClearCase must have a valid license to perform the requested task. The Rational ClearCase license server performs the role of allocating a license when requested by a Rational ClearCase command. It also ensures that the maximum number of licenses allowed as defined in the license.db file is not exceeded. This server can be made highly available, or as an alternative, Rational ClearCase does allow multiple license servers to prevent license server access from being a single point of failure.

Registry server host – The Rational ClearCase registry server contains databases used to determine the locations of Rational ClearCase data structures. This information is stored within the server host in files in the directory /var/adm/rational/clearcase/rgy. This server must also be highly available and can be protected by LifeKeeper to ensure its availability.

Release host – The Release host serves as the network-wide housing area that provides storage for the entire Rational ClearCase product distribution including executable files, configuration files, and online documentation. This host is essentially a file server. Rational ClearCase provides a link install type, which creates symbolic links to the Release Area via NFS mount points. Thus, it must be highly available if an install with links has been selected.

Shipping server host – The Shipping server host allows automatic replication and synchronization of ClearCase repositories. This server supports the MultiSite features of Rational ClearCase.

Client host – A Client host is a system that runs the Rational ClearCase client programs which constitute the user-level interface to Rational ClearCase. This interface includes cleartool, checkout, etc. A Client Host by itself does not require high availability protection.

Server host – A Server host stores Rational ClearCase data such as VOB and View storage directories. Rational ClearCase server processes execute on these hosts, as needed, to communicate with client programs through remote procedure calls. Because of the storage directories and server processes, high availability protection is required on these hosts to manage NFS resource for local storage access and IP resources for storage on NAS devices.

Whether running on individual dedicated servers or grouped into a fewer number of machines, the Rational ClearCase registry server, release (area) server, license server, shipping server, and VOB and View storage servers represent potential single points of failure and need to be protected in a high availability cluster configuration.

Providing a mechanism to recover protected Rational ClearCase services from a failed primary server onto a back-up server and to detect failures at both the server level via a heartbeat mechanism and at the individual Rational ClearCase resource level by monitoring the Rational ClearCase daemons, is a primary function of high availability clustering software. The remainder of this document presents several highly available configurations that can be deployed for Rational ClearCase on Linux.

High Availability Configurations

This section contains definitions and examples of several typical highly available Rational ClearCase configurations which can be built using SteelEye LifeKeeper for Linux.

Active/Standby Configuration with Local Shared storage

In this Active/Standby configuration, NodeA is the primary Rational ClearCase server providing Rational ClearCase Registry, Release Area, MultiSite Shipping/Return Bay, and storage locations for Views (the system is a View server). All storage resides on a shared array between the cluster servers. The LifeKeeper Protection Suite for Rational ClearCase provides monitoring of the Rational ClearCase functions and recovery as needed. NodeB acts as a hot-standby for Rational ClearCase while also running other applications/services.

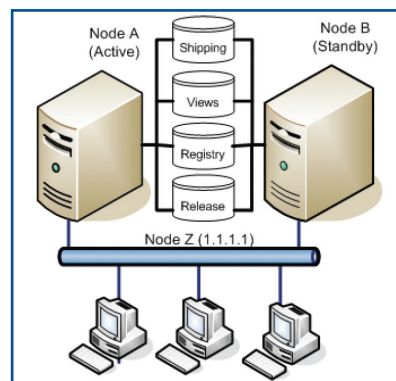


Figure 2: Active/Standby Local Shared Storage Configuration

Configuration Notes

- The clients connect to the Rational ClearCase servers using the DNS entry nodeZ designated to float between the servers in the cluster. The name uses the protected IP address (1.1.1.1).
- The rgy_hosts.conf file on all Rational ClearCase systems is set to nodeZ.
- The alternate_hostnames file on nodeA (the primary server) contains the following entries: nodeA and nodeZ (one name per line in the file).
- The alternate_hostnames file on nodeB (the back-up server) contains the following entries: nodeB and nodeZ.

Active/Standby Configuration with Network Attached Storage

In the Active/Standby configuration, nodeA is the primary LifeKeeper Server. It protects storage locations for VOBs. All storage is located on a NAS device. While nodeB may be handling other applications/services, it acts only as a back-up for the Rational ClearCase resources in LifeKeeper's context.

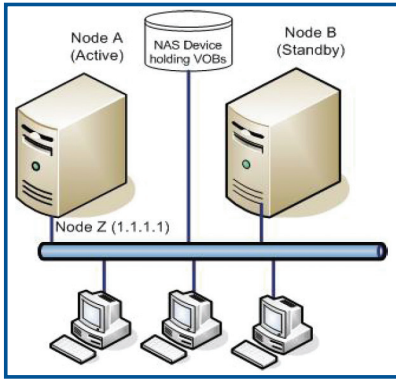


Figure 3: Active/Standby with Network Attached Storage Configuration

Configuration Notes

- The clients connect to the Rational ClearCase servers using the DNS entry nodeZ designated to float between the servers in the cluster. The name uses the protected IP address 1.1.1.1.
- The rgy_hosts.conf file on all Rational ClearCase systems is set to nodeZ.
- The alternate_hostnames file on nodeA contains the following entries: nodeA and nodeZ.
- The alternate_hostnames file on nodeB contains the following entries: nodeB and nodeZ.

Active/Active Configuration with Local Shared Storage

In the Active/Active configuration below, both nodeA and nodeB are primary LifeKeeper servers for Rational ClearCase resources. Each server is also the back-up server for the other. In this example nodeA protects the shared storage array for Views, the Registry, and MultiSite Shipping/Return Bays as the primary server. nodeB protects the shared storage array for VOBs as the primary server. Additionally, each server acts the back-up for the other, which in this example means that nodeB is the back-up for the protected View storage and Registry on nodeA, and nodeA is the back-up for the protected VOB storage on nodeB.

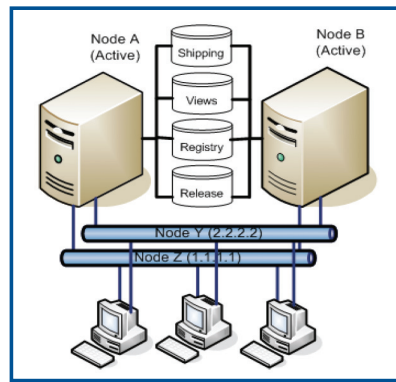


Figure 4: Active/Active Local Shared Storage Configuration

Configuration Notes

- The clients connect to the Rational ClearCase Servers using the DNS entries nodeZ and nodeY designated to float between the servers in the cluster. The names use protected IP addresses 1.1.1.1 and 2.2.2.2, respectively.
- The rgy_hosts.conf file on all ClearCase systems is set to nodeZ.
- The alternate_hostnames file on nodeA contains the following entries: nodeA, nodeY, and nodeZ.
- The alternate_hostnames file on nodeB contains the following entries: nodeB, nodeY and nodeZ.
- When the Rational ClearCase resources for nodeA are brought in-service on nodeB, Rational ClearCase must be stopped and restarted for the albd_server process to understand it is now providing access to the Registry.

Active/Standby Configuration with Replicated Storage

In this Active/Standby configuration, NodeA is the primary Rational ClearCase server providing Rational ClearCase Registry, Release Area, MultiSite Shipping/Return Bay, and storage locations for Views (the system is a View server). All storage is replicated between cluster servers, across a network connection using SteelEye Data Replication. LifeKeeper provides monitoring of the Rational ClearCase functions and recovery as needed. NodeB acts as a hot-standby for Rational ClearCase while also running other applications/services.

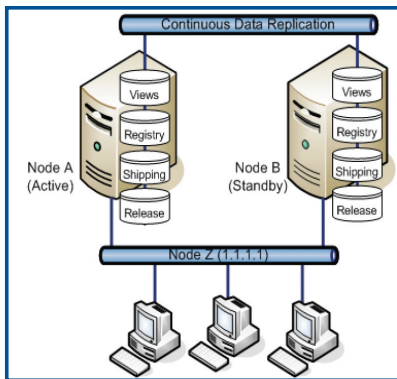


Figure 5: Active/Standby Replicated Storage Configuration

Configuration Notes

- The clients connect to the Rational ClearCase servers using the DNS entry nodeZ designated to float between the servers in the cluster. The name uses the protected IP address 1.1.1.1.
- The rgy_hosts.conf file on all Rational ClearCase systems is set to nodeZ.
- The alternate_hostnames file on nodeA contains the following entries: nodeA and nodeZ.
- The alternate_hostnames file on nodeB contains the following entries: nodeB and nodeZ.

Active/Active Configuration with Local Shared Storage

There are a number of considerations when building a Rational ClearCase high availability configuration. These are required to ensure that the path to critical Rational ClearCase resources (VOB storage, Registry, Release Area) are highly available via the use of virtual IP addresses and an associated system name in DNS. Rational ClearCase services are provided by a number of daemon processes. The main daemon processes are albd_server and lockmgr (this process no longer exists in ClearCase version 7 running on Linux), which must be monitored and protected in a high availability configuration. To provide a highly available Rational ClearCase Application, a number of configuration changes are required.

The need for these changes can be illustrated via the following example. A standard Rational ClearCase installation requires a host to be designated as the Registry server, nodeR. When Rational ClearCase is installed on nodeR, a number of files are created in the directory /var/adm/rational/clearcase/rgy that act as the repository of information for the operation of Rational ClearCase. When other systems are installed, they are set-up to contact nodeR via the setting in /var/adm/rational/clearcase/rgy/rgy_hosts.conf. If nodeR fails, then the Rational ClearCase Registry is no longer accessible and the client cannot perform any Rational ClearCase related commands, such as checking out a file for edit. If a back-up Registry was defined then someone with root access must switch the client machine to point to the back-up. If nodeR is clustered with another server, nodeB, then access to the Registry from a client can be made highly available via the following configuration changes.

1. Place the directory /var/adm/rational/clearcase/rgy on a shared disk (via the use of shared storage, NAS devices or via data replication) so that either system in the cluster can have access to the data.
2. Designate a name (nodeZ) and IP address (1.1.1.1) that can float between nodeR and nodeB (the name and address must be in DNS).
3. Define an IP resource to represent this new floating (or virtual) IP address and create a file system resource that represents the Registry directory. Both of these new resources will be monitored and recovered by the high availability software subsystem.
4. Edit the rgy_hosts.conf file on the client and alter the name from nodeR to nodeZ. Perform the same edit on the Registry server.

Now if nodeR fails, the HA software will detect the failure and migrate both the IP and file system resource to nodeB so that any client accessing the Registry on nodeZ via IP address 1.1.1.1 will now find it transparently on nodeB without any intervention on the part of an administrator and without any disruption in access to the Rational ClearCase services.

A similar configuration change will be required for VOB and View storage locations. When creating a VOB or View with storage on a local system you have the option of specifying the host, global path to the storage area and the local path to the storage area or letting Rational ClearCase determine it. If you let Rational ClearCase determine that information, it creates paths using the system name (from `uname -n` output). For a cluster with names of nodeA and nodeB creating storage on nodeA and letting Rational ClearCase determine values will lead to a host name of nodeA and paths starting as `/net/nodeA`.

As was done in the Registry example above, we need to use a system name and an IP address in DNS for the host name and path entries during creation of the storage area that can float between the servers in the cluster. So if nodeZ with IP address 2.2.2.2 is designated to float between servers in the cluster the VOB creation would use **-host nodeZ, -gpath /net/nodeZ...** and **-hpath /net/nodeZ...** to allow the storage to float between the servers in the cluster. Note that the storage area must reside on storage that each server in the cluster can access either through the use of shared storage, via a NAS device or through data replication. To limit the configuration changes to the Rational ClearCase Registry for an existing Rational ClearCase installation that has correctly defined paths, the current server name could be used for the virtual name which would then require a change to the actual system server name. For example, if `uname -n` currently returns "nodeA", make "nodeA" the virtual name and simply change the real name to return "nodeZ" on a `uname -n` call.

Using floating server names requires the use of the Rational ClearCase `alternate_hostnames` file located in `/var/adm/rational/clearcase/config` directory. In this file are listed all of the names by which the system is known. Using the Registry protection example from above, the primary server would have "nodeR" and "nodeZ", and the back-up server would have "nodeR" and "nodeB". The file `/var/adm/rational/clearcase/rgy/rgy_hosts.conf` also requires changes related to the floating server names. This file points to the Registry host server and will need to be modified on all systems running Rational ClearCase in order to switch the name from the true host name to the floating host name ("nodeR" to "nodeZ" in the above example).

If snapshot views are used, then the storage and database directories must be co-located on the same file system. Rational

ClearCase only stores the location of the database files in the Registry and not the View Storage. Co-location solves this problem and ensures that the high availability solution is protecting both directories.

If the MultiSite Shipping sever function is to be protected by LifeKeeper, then the directories specified in `/var/adm/rational/clearcase/config/shipping.conf` for STORAGE-BAY and RETURN-BAY must be on shared storage, network attached storage, or on a replicated file system.

Configuration Steps for LifeKeeper High Availability

This section provides steps which should be followed to configure Rational ClearCase resources within a LifeKeeper cluster in order to ensure the highest levels of availability protection.

Step One: Plan your Rational ClearCase configuration.

This includes the following:

- Determine what Rational ClearCase services to protect, e.g., all services on the server including VOB and/or View storage, Registry, Release Area, and MultiSite Storage and Return Bays or protect just VOB and/or View storage.
- Determine the permanent and floating server name(s) to be used in the cluster and ensure DNS has been updated to contain all the names and IP addresses.
- If the Rational ClearCase Registry is to be protected, ensure it is located on a LifeKeeper shared resource (shared storage, replicated storage or NAS device).
- Determine the changes required to the Release Area `site.dat` file as well as the changes required to the Registry if adding this kit to an existing Rational ClearCase installation.

Consideration should be given to the type of configuration (Active/Standby vs. Active/Active). It is recommended that if you are protecting the Registry server in an Active/Active configuration that the non-Registry server protects only VOB storage. The reason is that the clean-up which occurs when taking a Rational ClearCase resource out of service will terminate active views on the server, which will occur when the resource protecting the Registry server performs a failover.

Step Two: Set up your Rational ClearCase configuration files.

This includes the `rgy_hosts.conf` file on already installed systems and the `site.dat` file on the Release Host used in installing future systems.

Step Three: Create protected IP addresses under LifeKeeper for the floating server names noted above.

The floating server names must match the host names used in the Registry and when updating the `rgy_hosts.conf` file. Refer to the LifeKeeper for Linux IP Recovery Kit Administration Guide for details on setting up IP resources. Once created, test the protected IP addresses by pinging them from a number of clients and servers.

Step Four: Create LifeKeeper NFS resources for each local (non-NAS) storage area used for VOBs and Views as well as the Release Area. Ensure that the correct IP resource is used with the NFS hierarchy (if the path is `/net/nodeZ/export` make sure that IP resource used resolves to `nodeZ`).

Step Five: For existing Rational ClearCase installations, update the Registry with the new host names, global and local paths.

Step Six: Create the Rational ClearCase resource hierarchies in LifeKeeper to protect the Registry, Release Area, MultiSite Shipping, and Storage Areas as needed.

Step Seven: Finally, test access to the VOB data while the Rational ClearCase hierarchies are in-service and protected on the primary server.

Summary

Rational ClearCase provides a complete and powerful platform for software development projects on Linux. With full support for 2.4 and 2.6 kernel-based distributions from Red Hat and Novell and available for the complete line of IBM System x Servers, users can build a development environment which

is tailored to their requirements today with the assurance easy scalability as business needs change in the future. The availability of development resources is critical to productivity and asset protection; The LifeKeeper Protection Suite for Rational ClearCase provides automated monitoring and recovery of Rational ClearCase and allows the building of multiple configurations that can be used to ensure always-available development resources.

References

IBM Support Policy for Rational ClearCase on Linux
<http://www-1.ibm.com/support/docview.wss?rs=0&uid=swg21159882>

Linux Application Development Environment
<http://www-306.ibm.com/software/rational/linux/>

Supported Rational ClearCase Family Releases (versions 2003.06 and 7)
<http://www-1.ibm.com/support/docview.wss?rs=727&uid=swg21136950>

<http://www-1.ibm.com/support/docview.wss?rs=984&uid=swg21239315>

Rational ClearCase Install Guide for Linux
<http://www-1.ibm.com/support/docview.wss?rs=0&uid=swg27005767>

IBM Ready for Rational
<http://www-304.ibm.com/jct09002c/gsdod/weblisfilter.do?prog=RFIRS&tab=0>

SteelEye LifeKeeper Protection Suite for Rational ClearCase
<http://www.steeleye.com/solutions/>

About SteelEye Technology

SteelEye is the leading provider of data and application availability management solutions for business continuity and disaster recovery for Linux and Windows and virtual environments.

The SteelEye family of data replication, high availability clustering and disaster recovery solutions are priced and architected to enable enterprises of all sizes to ensure continuous availability of business-critical applications, servers and data.

To complement its software solutions, SteelEye also provides a full range of high availability consulting and professional services to assist organizations with the assessment, design and implementation of solutions for ensuring High Availability within their environments.

SteelEye is a subsidiary of SIOS Technology, Inc. To contact SteelEye, visit www.steeleye.com or call:

US/Canada 866.318.0108 Europe + 44 (0)1223 208701 Int'l + 1.650.843.0655

SteelEye Technology, Inc.
4400 Bohannon Drive, Suite 110
Menlo Park, CA 94025

Replicate Any Data. Protect Any Application



SteelEye
TECHNOLOGY INC

www.steeleye.com