



IBM Software Group | Rational software

IBM® Rational® ClearCase MultiSite®

An Introduction to MultiSite

Rational software

@business on demand.

© 2008 IBM Corporation
Updated May 19, 2008

The slide features a blue header with the IBM logo and 'IBM Software Group | Rational software'. The main content area is white with the title 'IBM® Rational® ClearCase MultiSite®' and subtitle 'An Introduction to MultiSite'. Below the title is the 'Rational software' logo. A horizontal bar with various icons (a person, a globe, a refresh symbol, a network diagram, etc.) is positioned below the logo. The footer is blue with '@business on demand.', copyright information, and an update date.

This module will cover an introduction for IBM Rational ClearCase MultiSite versions 2003.06.15 and higher.

Introduction

- This module will cover what MultiSite is used for, VOB data replication, basic concepts and terminology, basic multitool commands and basic administrative tools
- Prerequisites: An understanding of ClearCase



This module will cover MultiSite usage, what VOB data is - and is not - replicated, basic concepts and terminology, basic multitool commands and basic administrative tools. You should already be familiar with ClearCase concepts and terminology before reading this module. For example, you should know what a VOB is and how files are stored in it, how a user would work with files in a VOB, how to run cleartool commands and what basic administrative tools are before continuing.

Module objectives

- The following topics are covered in this module:
 - ▶ What is MultiSite?
 - ▶ VOB data replication
 - ▶ MultiSite concepts and terminology
 - ▶ Basic multitool commands
 - ▶ The MultiSite administration console and shipping.conf file
 - ▶ Synchronization jobs
- Upon completion of this module, you will be able to:
 - ▶ Be more familiar with how MultiSite functions to replicate data
 - ▶ How data is kept in sync with remote sites
 - ▶ Be familiar with the basic administration tools and commands



This course covers the following topics: What is MultiSite, VOB Data Replication, MultiSite Concepts and Terminology, The MultiSite Administration console, shipping.conf files and Synchronization jobs.

Upon completion of this module, you will be more familiar with how MultiSite functions to replicate VOB data across multiple geographical locations and how data is kept in sync with remote locations. You should also gain knowledge of the basic administrative tools and commands.

What is MultiSite?

- A solution to replicate VOB data to multiple geographical locations
- VOBs are replicated by a user and sent to a remote location within a network
- Changes in a VOB are replicated to remote sites
- Can be used as a disaster recovery tool



What is MultiSite? ClearCase MultiSite provides a solution to replicate VOB data to other geographical locations within an organization. This allows users to be able to access a common set of data across multiple physical locations within an organization. Work can be done on a project in various locations throughout the world. ClearCase VOBs are replicated by a user and sent to a remote location within a network. The data is transferred to a remote location via a standard TCP/IP connection to the remote host. A user at the receiving site imports the new replica and makes it available to the local users.

When elements are added, or data is changed in a VOB that has been enabled for replication, these changes are recorded and replicated to remote sites. The synchronizations between the replicas keep all of the various sites up to date with each other. The synchronizations can be automated through the use of scheduled jobs.

MultiSite can also be used as a disaster recovery tool. There can be a backup server in which replicas of all production VOBs exist. The backups are kept up to date via the regular synchronization methods. When a system goes down in a production environment, there can be a replica waiting on the backup system. An administrator can have users switch over to the replicas until the primary system is back up and running

VOB data replication

- Data that is replicated
 - ▶ Elements, branches and versions
 - ▶ Most kinds of type objects
 - ▶ Metadata
 - ▶ UCM objects
 - ▶ Permanent locks
 - ▶ Checkout records of elements and changes in checked-out directories
 - ▶ Event records
 - ▶ Mastership information



There is a lot of information that get replicated within a VOB family. The following is a list of data that will be propagated to other replicas in a VOB family:

Elements, branches, and versions; this also includes derived object versions.

Metadata which includes version labels, attributes and hyperlinks. This also includes merge arrows and hyperlinks to administrative VOBs.

UCM objects which includes activities, baselines, components, folders, projects and streams.

Permanent locks that result from a user running the 'cleartool lock -obsolete' command.

Checkout records of elements and changes in checked-out directories.

Event records.

Mastership information of objects. (Mastership will be explained in greater detail later in this module)

VOB Data Replication

- Data that is not replicated
 - ▶ Derived objects that have not been checked in as versions
 - ▶ Trigger type objects
 - ▶ Individual “attached” triggers
 - ▶ Temporary locks
 - ▶ Contents of checked-out versions
 - ▶ Mastership request settings
 - ▶ Custom type managers
 - ▶ Changes to text mode property



The information that is not replicated between VOBs is:

Derived objects that have not been checked in as versions. DOs are usually large and short-lived, thus transmitting DOs among multiple replicas would not be very efficient. It is more efficient to rebuild them at each replica.

Trigger type objects are not replicated because they are usually used to implement local policies, and trigger type definitions can include local pathnames that do not exist at other sites.

Temporary locks that were created without the **–obsolete** option. For example if a user were to run the ‘cleartool lock vob:\vobtag’ command to run a backup, this setting would not be replicated.

Contents of checked-out versions, mastership request settings and custom type managers are not replicated.

When you create a new replica, it has the same text mode property as its parent replica, but subsequent changes are not propagated

MultiSite concepts and terminology

- **Replica**
 - ▶ A VOB that has been copied using MultiSite commands
 - ▶ Given a unique name to distinguish it from other replicas
- **Replica Family**
 - ▶ A VOB that has been replicated one or more times
 - ▶ There can be many replicas of a VOB in a given family
 - ▶ Each VOB that has been replicated is part of its own family



A replica is the word used to describe a VOB that has been copied using MultiSite commands. A replicated VOB is located at multiple sites. It is given a unique name to distinguish it from other replicas.

A replica family is a set of replicas of a VOB. There can be many replicas of a VOB in a given family. Each replica includes a full set of data containers and a complete copy of the VOB database. At its site, a replica appears to be a regular VOB; developers can check out, edit, and check in; build software; attach metadata to objects; and so on

Each VOB that has been replicated is part of its own family. For example, VOB_A is replicated to generate VOB_A_Replica. VOB_B is replicated to generate VOB_B_Replica. VOB A and B are 2 separate replica families.

MultiSite concepts and terminology

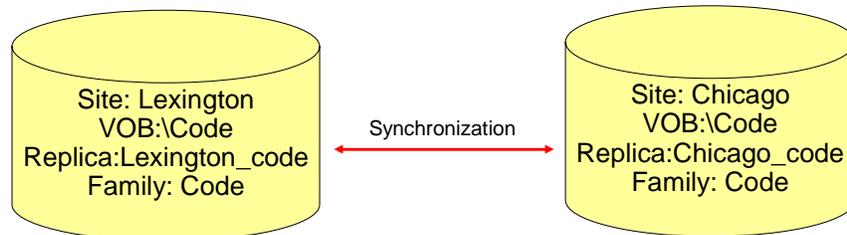
- Site
 - ▶ A collection of clients and servers
 - ▶ Each site can only have one replica of a VOB
- Host
 - ▶ A name or IP address of the network node that contains the database of the VOB replica
 - ▶ Each site can use a host that can run the Microsoft® Windows® or UNIX® operating systems



A site is a collection of clients and servers. A site usually refers to a physical geographical location where a replica of a VOB is located. Each site can have only one replica of a VOB. At each site there is a host which contains the replica of the VOB. The host is the name or IP address of the machine holding the replica. In a replicated environment you do not have to limit the type of operating systems the hosts run. You can run a mixture of UNIX and Windows hosts and replication can still occur between the two sites running different operating systems.

MultiSite concepts and terminology

Replica/family illustration



In this illustration there are 2 different sites, Lexington and Chicago. The Lexington site contains the \Code VOB and the Lexington_code replica of the VOB. The Chicago site also contains the \Code VOB but the Chicago_replica of the VOB. Together these two replicas make up the Code VOB family. The two replicas exchange updates with each other via synchronization so that both sites can have the same set of data.

MultiSite concepts and terminology

- Multitool
 - ▶ The utility that is used to create replicas of VOBs and keep them in sync
 - ▶ It functions in the same manner as *cleartool*
- Oplogs
 - ▶ Recorded in the VOB database
 - ▶ Keeps track of operations that were performed within the replicated VOB
 - ▶ Every operation is recorded with a specific ID and sequence number
 - ▶ Oplogs must be applied to remote replicas in the sequence that they were recorded in



Multitool is the utility that is used to run the set of commands to create replicas of VOBs and keep them in sync with each other. It is also used to run other diagnostic commands for troubleshooting purposes. It functions in the same manner as *cleartool*; multitool must be entered before the desired command can be executed. It's located in the CCHOME\ClearCase\bin directory on Windows and on UNIX.

Operation logs are recorded in the VOB database. They keep track of operations that were performed within the replicated VOB. Every operation is recorded with a specific ID and sequence number in which the operations occurred. Oplogs must be applied to remote replicas in the sequence that they were recorded in.

For example: A user checks an element out in a replicated VOB, generating oplog number 5. The user checks the element back in after the changes are made generating oplog number 6. Oplog number 5 must be applied to a remote replica before oplog number 6.

MultiSite concepts and terminology

- Epoch table
 - ▶ Each replica keeps track of its own oplogs in the epoch table
 - ▶ Each replica estimates how many oplogs other replicas have
- Divergence
 - ▶ When two or more replicas in a family do not have the same operation recorded in the same oplog number
 - ▶ Most common cause of oplog divergence between replicas is an improper restore from backup



Every replica in a family keeps track of its own oplogs in the epoch table. Each replica also estimates how many oplogs the other replicas in the family has.

Divergence is a term used to describe when two or more replicas in a family do not have the same operation recorded in the same oplog number. The most common cause of oplog divergence between replicas is an improper restore from backup.

When you restore a replica from backup, its epoch row is rolled back to the last changes that were made before the backup occurred. If you do not run a command called `restorereplica` on the replica before resuming development in it, divergence will occur.

For example, operations 1-800 are created in a replica and exported to sibling replicas. The replica is then restored from backup and its epoch number becomes 700 (operations 701-800 occurred after the backup copy was created). If `restorereplica` is not run, development resumes and new operations are recorded in the oplog starting with ID 701. These operations have the same ID as the operations that were exported to other replicas before the restoration, but the operations themselves are different. The restored replica has diverged from the other replicas. The only way to recover from a diverged replica is to recreate it from another replica in the family.

Epoch table example

```

M:\testview\Lexington>mt lsepoch
For VOB replica "\Lexington":
Oplog IDs for row "Chicago" (@ Chicago_host):
oid:737e2b9b.f08240c8.968e.5c:67:8d:1b:54:d2=21      (Chicago)
oid:92ccc961.6ffb4b51.93b4.da:94:75:29:8e:ae=0     (Chicago.deleted)
oid:cb53e85b.95244fff.8646.f2:6a:71:44:26:48=118  (Lexington)
Oplog IDs for row "Lexington" (@ Lexington_host):
oid:737e2b9b.f08240c8.968e.5c:67:8d:1b:54:d2=18  (Chicago)
oid:92ccc961.6ffb4b51.93b4.da:94:75:29:8e:ae=0   (Chicago.deleted)
oid:cb53e85b.95244fff.8646.f2:6a:71:44:26:48=125 (Lexington)
  
```

Identify site and host where the replica is located by the (@ host name)

The number of changes that originated in the Lexington replica that Chicago has

Each row (or site) keeps track of its own oplogs

The number of changes that the Lexington replica has from Chicago



In the lsepoch output example, the \Lexington VOB contains two replicas, Lexington and Chicago. Each replica is listed on its own separate row in the table. You can tell which site and host the replica is sitting on the (@ hostname) notation. The replicas also keep track of which replicas were deleted as indicated by the .deleted notation.

Each replica lists the current number of oplogs it has for itself. Each row will also list the number of oplogs that it has from its sibling replica as well what it thinks the remote site has in its epoch table. This number may not always be accurate due to a number of circumstances such as lost packets, site connectivity or sync intervals. You can see that in this example Chicago and Lexington are not in sync with each other as the Lexington replica has 18 out of 21 changes from Chicago. Chicago only has 118 out of the 125 changes that originated in Lexington.

MultiSite concepts and terminology

▪ Sync packets

- ▶ Packets that contain the updated information recorded in the ologs
- ▶ Generated at a sending site by running multitool commands

▪ Shipping bays

- ▶ Physical storage locations on a hard drive to store incoming and outgoing packets
- ▶ Directories on a file system
- ▶ The default shipping bays are C:\Program Files\Rational\ClearCase\var\shipping on Windows and opt/rational/ClearCase/var/shipping on UNIX



Sync packets are packets that contain the information recorded in the ologs to be applied at remote sites. Sync packets are generated at a sending site by running multitool commands. The packets are forwarded to remote sites via a TCP/IP connection so that they may be applied to the sibling replicas to keep them in sync.

Shipping bays are physical storage locations on a hard drive that MultiSite uses to store incoming and outgoing packets in.

Shipping bays are directories on a file system. The default shipping bays are C:\Program Files\Rational\ClearCase\var\shipping on Windows and opt/rational/ClearCase/var/shipping on UNIX.

MultiSite concepts and terminology

- Storage class
 - ▶ A set of incoming and outgoing shipping bays
 - ▶ One or multiple storage classes may exist
- Store-and-forward
 - ▶ A term used to describe the forwarding of sync packets
 - ▶ Refers to the process of storing a packet in a shipping bay and then forwarding it to the next host



A storage class is a set of incoming and outgoing shipping bays. You may have one or multiple storage classes set up in an environment.

For example, you may want different storage classes to hold packets shipping to different VOB families.

The term store-and-forward is used to describe the storing and shipping of sync packets. The store-and-forward facility consists of storage classes, shipping bays and the shipping server process. The storage classes define where the shipping bays are located, the shipping bays store the packets once they are generated and the shipping server process forwards the packets to their next destination.

MultiSite concepts and terminology

- Shipping order
 - ▶ A file that is generated when update packets are created
 - ▶ Contains the following information:
 - arrival destination
 - arrival route
 - creation date of the shipping order
 - expiration date
 - ▶ Shipping order is processed by the shipping server



A shipping order is a file that is generated when update packets are created. The shipping order contains information such as arrival destination, arrival route, creation date of the shipping order and expiration date. The shipping order is processed by the shipping server when it runs.

Shipping order file example

An example of a shipping order file that was generated from running the multitool syncreplica command

```
# ClearCase MultiSite Shipping Order
# Version 1.0
%IDENTIFIER db214213.989e4f64.bb21.3e:ac:97:e1:34:c5
%CREATION-DATE 1206026348 20-Mar-08.11:19:08
%EXPIRATION-DATE 1207235949 03-Apr-08.11:19:09
%ORIGINAL-DATA-PATH "C:\Program Files\Rational\ClearCase\var\shipping\ms_ship\outgoing\sync_Lexington_20-Mar-08.11.19.08_3668"
%LOCAL-DATA-PATH "C:\Program Files\Rational\ClearCase\var\shipping\ms_ship\outgoing\sync_Lexington_20-Mar-08.11.19.08_3668"
%DESTINATIONS
  my_remote_host
%ARRIVAL-ROUTE
%DELIVERIES
%FAILED-ATTEMPTS
%NOTIFICATION-ADDRESSES
%COMMENT
```

Note the creation date and expiration date lines

Destination is where the packet is being routed to

Storage location where the sync packets reside



The shipping order will be created with a name of sh_o_ <originating replica name>_<date and time sync was run>. The shipping order files can be found in the outgoing bay of a host when an update packet is generated. The file will log its creation date and time as well as the packet expiration date and time. The expiration date and time specifies the time at which the store-and-forward facility stops trying to deliver the packet and generates a failure mail message.

If a new shipping order file needs to be generated the mkorder command can be used. Mkorder is not a multitool subcommand as it is its own utility which resides in the CCHOME\ClearCase\bin directory. The mkorder command can be used with a -pexpire switch in which a user can specify a date and time the packets will expire. The expiration date and time indicates when the store-and-forward facility will stop trying to forward packets to a remote host.

MultiSite concepts and terminology

- Shipping server
 - ▶ Processes sync packets
 - ▶ Sends the packets to the destinations specified in the shipping order file
 - ▶ A stand-alone shipping server can be installed on a Linux\UNIX host
- Receipt handler
 - ▶ Processes packets that that have arrived at a destination
 - ▶ Will import the new update packets into a replica on the local host



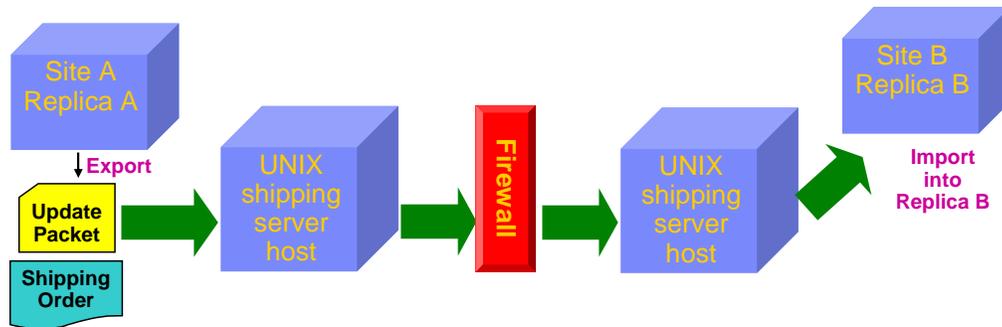
The MultiSite shipping server processes synchronization packets. It sends the packets to the destinations specified in the shipping order file.

A stand-alone shipping server can be installed on a Linux\UNIX host. When you install ClearCase MultiSite onto a UNIX or Linux host, there will be an option for a shipping server only install. No other ClearCase services or processes will be installed when this option is selected. This is also referred to as a store-and-forward host. A store-and-forward host just holds packets and routes them to their next destination.

A receipt handler processes packets that that have arrived at a destination. It will import the new update packets into the replica that is specified in the sync packet on the local host.

MultiSite concepts and terminology

Example of a multiple shipping server environment



An update packet is generated for a replica at site A by running the multitool `syncreplica - export` command. It gets the routing path as specified in the `shipping.conf` file or Multisite Control Panel. It is then forwarded to the shipping server host and arrives at the destination VOB server where it is imported into the target replica. In order for a packet to pass through a firewall special options need to be configured to allow packets to pass on specified ports.

MultiSite concepts and terminology

- **Mastership**
 - ▶ Most objects in a replica have mastership
 - ▶ A replica must have mastership of an object in order for a user to modify it
 - ▶ It enables individual development
 - ▶ Can be requested by a user from a remote site
 - ▶ Transferred in an update packet



Most objects in a replica have mastership. Some objects are assigned a master replica when they are created. After this point, mastership can then be transferred.

A replica must have mastership of an object in order for a user to modify it. Mastership allows a user at one site to work on a file and transfer it to another site when another user needs to modify it. Without mastership in a replicated environment, there would be complete chaos. For example, if version 5 were created off the main branch in 4 different replicas at the exact same time there would be no way to tell which one was the real version 5.

Mastership can be requested by a user from a remote site via command line or GUI methods and is then transferred in an update packet.

MultiSite concepts and terminology

Mastership example from ClearCase Explorer

When replication is enabled in a VOB there is a Mastership tab

The current master of the element is the Lexington replica

The properties window of the java_code.java element

Special privileges are required to change mastership

M: \testview\Lexington\Some Changes\java_code.java@ ...

General | Mastership | Custom | Triggers | Protection | Lock

Current replica: Lexington

Master replica: Lexington

Change...

OK Cancel Apply Help

20

An Introduction to MultiSite

© 2008 IBM Corporation

In the example, the properties page of an element is displayed from ClearCase explorer. A user can tell that a VOB has been replicated by the mastership tab. The mastership tab does not appear in non-replicated VOBs.

This tab will display the current mastering replica of the element being displayed. To change mastership of non-ucm objects you must be the object creator (except for replicas), object owner (except for replicas), VOB owner, **root** (UNIX systems), or a member of the ClearCase administrators group (Windows). There is no special identity required for UCM objects, except baselines.

Basic Multitool commands

- mkreplica
 - ▶ Used to create new replicas of existing VOBs
 - ▶ Generates replica creation packets to be sent to a remote site
- rmreplica
 - ▶ Used to remove an existing replica
 - ▶ Once a replica is removed a synchronization packet must be sent to all replicas in the family informing them of the change



Mkreplica is used to create new replicas of existing VOBs. It will generate replica creation packets to be sent to a remote site. Mkreplica can be run with a number of different switches. For example, there are options to send the packets immediately or specify the maximum size a create packet should be.

Rmreplica is used to remove an existing replica. Rmreplica does not remove the physical VOB storage at the local site. It is highly recommended to run this command before the rmvob command. Once a replica is removed, a synchronization packet must be sent to all replicas in the family informing them of the change. To remove a replica object, the current replica where the command is run from must have mastership of the object. After the replica has been removed, the rmvob command may run at the local site to remove the physical VOB storage.

Basic Multitool commands

- **restorereplica**
 - ▶ Replaces missing operations in a replica that has been restored from backup
 - ▶ Must be run immediately after a restore from backup is performed or divergence will occur
- **lsreplica**
 - ▶ Will list all of the replicas of a specified VOB
 - ▶ Can be used to determine how many replicas are in a family



Restorereplica replaces missing operations in a replica that has been restored from backup. The command must be run immediately after a restore from backup is performed or divergence will occur. Restorereplica replaces missing changes in a VOB replica that has been restored from backup, as follows:

It causes the current replica to create special update packets that contain update requests to other replicas.

It locks the current replica's VOB object and marks the replica as being in the process of restoration.

It increments the recovery incarnation for the replica.

It causes lsreplica -long to indicate which replicas must send restoration updates to the current replica.

Lsreplica will list all of the replicas of a specified VOB. It can be used to determine how many replicas are in a family and when they were created. The -long switch will also list more detailed information such as the VOB owner and group, feature level, identities and permissions status, and whether or no the replica has connectivity to other sites.

Basic Multitool commands

- syncreplica
 - ▶ Used to generate an update packet to be sent to a remote replica
 - ▶ Specific switches can be specified to ship packets immediately or place them in the shipping bay for future processing
- lsepoch
 - ▶ Provides a print out of the current replicas epoch table
 - ▶ It lists how many oplogs it has for the replica the command is run against, and estimates the number of oplogs for the other replicas in the family



Syncreplica is used to generate an update packet to be sent to a remote replica. Specific switches can be specified to ship packets immediately or place them in the shipping bay for future processing. Many other switches also exist such as specifying the maximum packet size, work directories, packet expiration time and which view to use.

Lsepoch provides a print out of the current replicas epoch table. It lists how many oplogs it has for the replica the command is run against as well as estimates how many oplogs the other replicas in the family have. Lsepoch is a very useful troubleshooting command that can be used for many different situations.

Basic Multitool commands

- **lspacket**
 - ▶ Lists all packets that are currently in the shipping bays
 - ▶ Can be used to help determine if a packet has been shipped or has been lost
- **chmaster**
 - ▶ Changes mastership of a specific object to a remote site
 - ▶ An update packet must be generated after the command is run and mastership will be transferred.



Lspacket will list all packets that are currently in the shipping bays on a host. It can be used to help determine if a packet has been shipped or has been lost.

Chmaster changes mastership of a specific object to a remote site. An update packet must be generated after the command is run so mastership can be transferred. There are restrictions that apply when running the chmaster command as mentioned in the mastership GUI screen capture example slide.

Basic Multitool commands

Ispacket example output

C:\>multitool lspacket

Packet is: C:\Program
Files\Rational\ClearCase\var\shipping\ms_ship\outgoing\sync_Lexington_20-Mar-0

8.11.19.08_3668

Packet type: Update

Packet fragment: 1 of 1

VOB family identifier is: afdda1ab.fd0842d0.857b.dd:d5:cf:3f:44:ba

Comment supplied at packet creation is:

Packet intended for the following targets:

Chicago

The command output will show the exact path of the packet location

The VOB family oid

The output will also indicate what replica it is being sent to



The lspacket command will show the user the exact path of the packet location. It will also list what the VOB family object identifier (oid) is. Finally the output will list any comments that were made when it was created and which replica the packet is intended for. There is also a -long switch that will provide additional diagnostic data such as which replica the packet originated from and the first oplog ID contained in the packet.

Basic Multitool commands

- reqmaster
 - ▶ Issues a request for mastership of a given object to a specified remote site
 - ▶ reqmaster contacts a remote system directly
 - ▶ Mastership of the object is then transferred back in an update packet
 - ▶ The user who issues the reqmaster command must have the appropriate permissions
 - ▶ Requests for mastership are not enabled by default
 - ▶ reqmaster will not function through a firewall



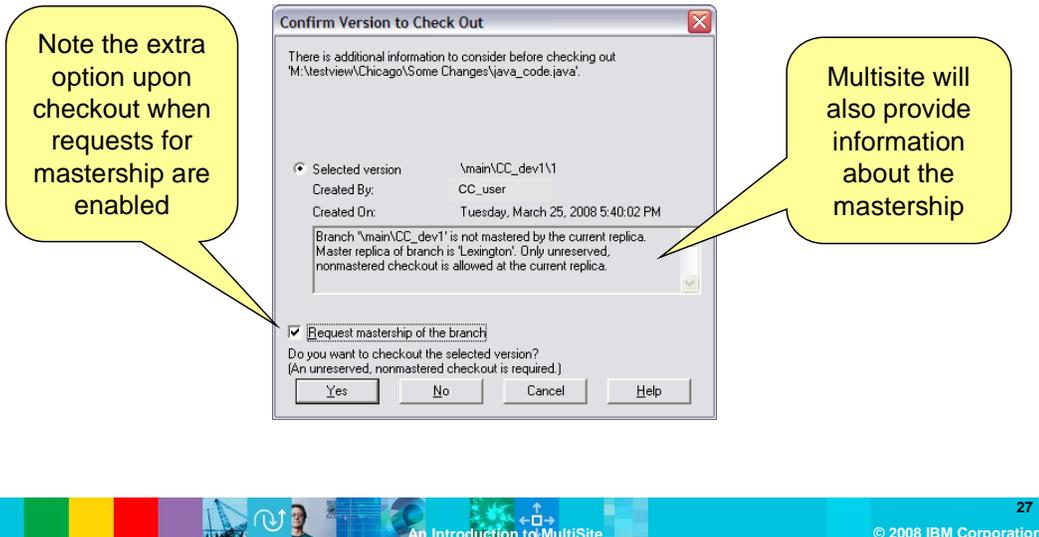
Reqmaster issues a request for mastership of a given object to a specified remote site. The reqmaster command then contacts the remote host where the VOB replica resides directly. If the user is allowed to request mastership at the local site, mastership of the object is then transferred back in an update packet.

To allow requests for mastership, a MultiSite administrator has to set access controls at each replica. Mastership settings are not replicated.

Requests for mastership are not enabled by default. This setting needs to be enabled on a per replica basis. The reqmaster command also will not function through a firewall device. To workaround this limitation the chmaster command must be used instead.

Basic Multitool commands

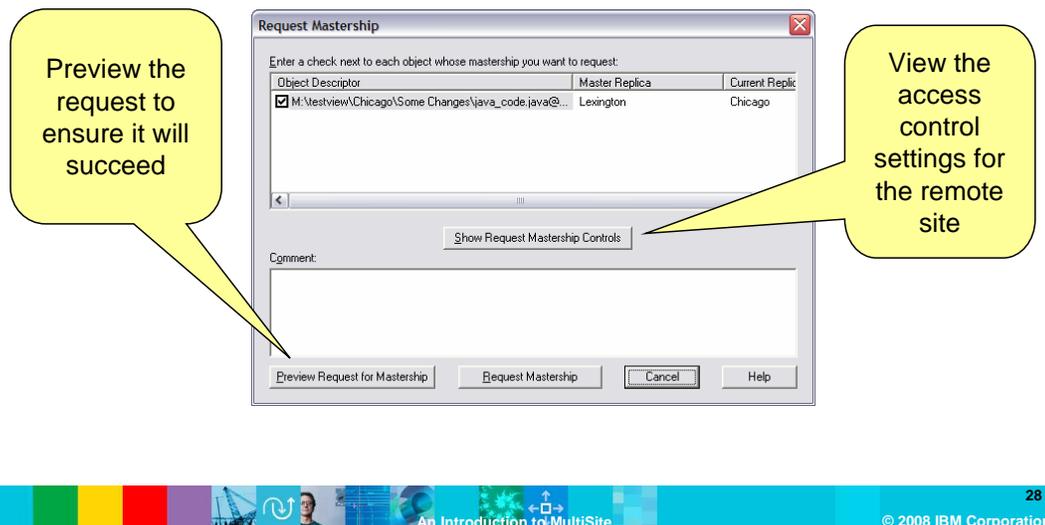
Example of reqmaster from ClearCase Explorer



In this example the request for mastership was initiated on a Windows system upon a checkout of an element version. When requests for mastership are enabled in a replica there will be an extra option listed to send the request to the remote site. There is also an extra description box that will display details about the mastership. It will usually say that the current branch or element is not mastered by the current replica. If a user does not request mastership in this screen, only a unreserved, nonmastered checkout can be performed.

Basic Multitool commands

Example of reqmaster from ClearCase Explorer



Once the Yes button is clicked the Request Mastership window is shown. In this window a user can click the “Preview Request for Mastership” button to see if the request will succeed or not. A user can also view the access control settings for the remote site. This is useful to see if the user does or does not have permissions to request mastership at the remote site. This button will display the ACL for the remote site. Once the user is ready, he or she can click the “Request Mastership” button which will complete the request.

The Multisite control panel and shipping.conf file

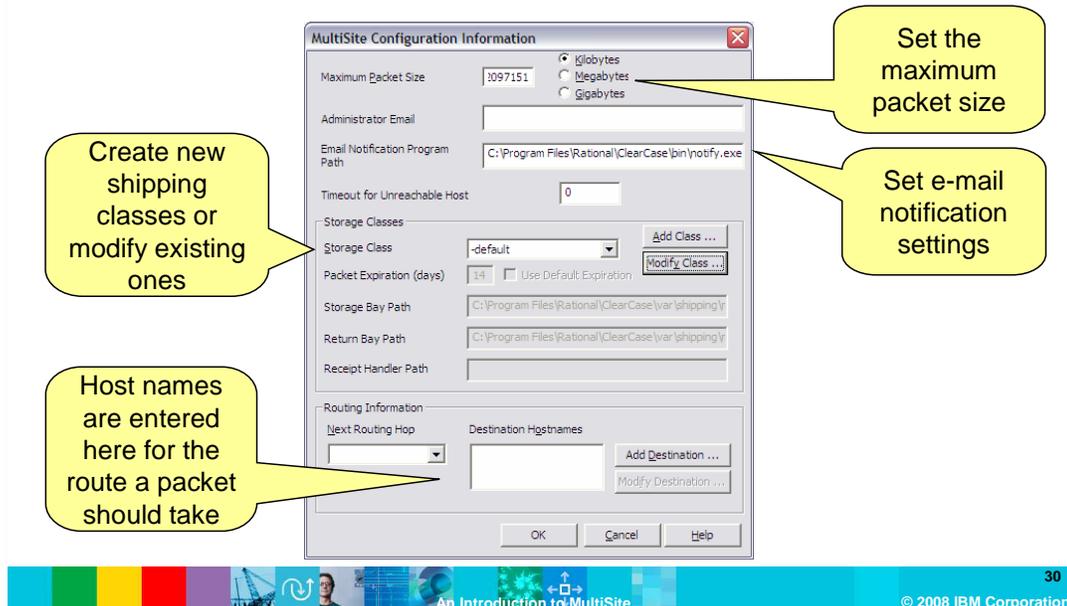
- Multisite control panel (Windows)
 - ▶ The control panel is located in the Windows Control Panel
 - ▶ This is the location where you specify customized storage classes, setup e-mail notifications, set the maximum size for sync packets and the routing path for packets



The Multisite administrative console is located in the Windows Control Panel. This is the location where you specify customized storage classes, setup e-mail notifications, set the maximum size for sync packets and the routing path for packets. You can also set the time in which the store-and-forward facility will stop trying to forward a packet.

The MultiSite control panel and shipping.conf file

Control panel screen capture



An administrator can set the maximum packet size by entering a number and specifying whether that number will be kilobytes, megabytes or gigabytes. This value is used by syncreplica and mkreplica commands unless the `-maxsize` option is specified. The administrator e-mail can have one or multiple e-mail addresses entered in the box for which notifications will go to.

The timeout for an unreachable host value specifies the number of minutes for the shipping server to wait before trying to contact a target host that was previously identified as unreachable.

The storage class section is where an administrator can set which storage class is to be used on the system. If the user does not want to use the default he or she can click the "Add Class..." button which will create a new class with user specified parameters. Once multiple classes exist, a user can specify the `-sclass` option in syncreplica and mkreplica commands.

When the mkorder or syncreplica command is executed with the `-pexpire` option to generate a new shipping order file, it overrides the expiration period specified in the Packet Expiration (days) option for the storage class in the shipping.conf file or MultiSite Control Panel.

An administrator can also add host names of shipping servers that are in the path that a packet will need to travel along to get to its final destination. A host name can be specified with the name of the host itself or the IP address. The Next Routing Hop specifies the next host the packet will be forwarded to.

The Multisite control panel and shipping.conf file

- shipping.conf (UNIX)
 - ▶ The shipping.conf file contains the same settings as the Windows Control Panel.
 - ▶ It also contains settings for the CLEARCASE_MIN_PORT and CLEARCASE_MAX_PORT variables which are used in an environment where a firewall is present
 - ▶ It is located in /var/adm/rational/clearcase/config



The shipping.conf file contains the same settings as the Windows Control Panel. It also contains settings for the CLEARCASE_MIN_PORT and CLEARCASE_MAX_PORT variables which are used in an environment where store-and-forward servers and a firewall are present. On Windows these variables are set as system environment variables. It is important to note that if these variables are to be set on a Windows system, no other ClearCase processes can be running on the system. If other processes do run it will cause unknown behavior due to the way the albd_server dynamically assigns ports to it's child processes.

The shipping.conf file It is located in /var/adm/rational/clearcase/config

Synchronization jobs

- Jobs that run in the ClearCase scheduler to keep all replicas on a system in sync
- There are four default MultiSite jobs
 1. Daily Multisite Export
 - Can be set up to automatically generate synchronization packets to ship to remote sites
 - Can be set to generate packets for all replicas on a system or just a set of them with a customized sync export script
 2. Daily Multisite Shipping Poll
 - Polls the shipping bays for packets that may have been left behind by the sync export job
 - It can be set to poll user defined shipping bays
 - Also used in conjunction with the sync export job to ship packets after they are generated



Synchronization jobs are the jobs that run in the ClearCase scheduler to keep all replicas on a system in sync

There are four default MultiSite jobs

The Daily Multisite Export which can be set up to automatically generate synchronization packets to ship to remote sites.

The export job can be set to generate packets for all replicas on a system or just a set of them with a customized sync export script created by a user. By default the job will run the `sync_export_list.bat` and will export sync packets from all replicated VOBs found on the local system in which it runs. The `sync_export_list` also has a number of options that can be specified. For example it can work with a custom user defined script in which specific replicas are listed and shipping classes can be specified.

The Daily Multisite Shipping Poll polls the shipping bays for packets that may have been left behind by the sync export job

It can be set to poll user defined shipping bays. It is also used in conjunction with the sync export job to ship packets after they are generated. For example a sync export job is set to run every 10 minutes. A shipping poll job runs every 20 minutes to forward the packets that were generated from the sync export command. Staggering the jobs will help to conserve system resources and reduce network traffic.

Synchronization jobs (continued)

3. Daily Multisite Receive
 - Processes any packets that have arrived in the incoming shipping bays
 - It can also be set to processes packets in custom user defined shipping bays
 4. Daily CC MultiSite Family Health Generation
 - Runs the familysyncstatus command to display a summary of synchronization health status reports
 - Does not run by default as the arguments field is blank
 - It can be set to run against VOBs specified in the arguments field
- Custom sync jobs may also be created to ship to different groups of replicas at specified time intervals



The Daily Multisite Receive job processes any packets that have arrived in the incoming shipping bays and applies them to the corresponding replica. It can also be set to processes packets in custom user defined shipping bays. This job will run the sync_receive.bat batch file in the background which actually imports the packets into the VOBs specified in the packets.

The Daily CC MultiSite Family Health Generation job runs the familysyncstatus command to display a summary of synchronization health status reports. The job does not run by default as the arguments field is blank. The job can be set to run against VOBs specified in the arguments field.

Custom sync jobs may also be created to ship to different groups of replicas at specified time intervals; the default jobs do not have to be used.

Review

- **What is MultiSite?**
 - ▶ What functionality does MultiSite provide to ClearCase?
 - ▶ What other strategies can MultiSite be used for?
- **VOB Data Replication**
 - ▶ What data in a VOB is replicated?
 - ▶ What data in a VOB is not replicated?
- **MultiSite Concepts and Terminology**
 - ▶ What is a replica?
 - ▶ What is a replica family?
 - ▶ What is the store-and-forward facility?
 - ▶ How is work performed in replicated VOBs?



In review this module discussed what MultiSite is and also what it can be used for. What Functionality does MultiSite provide to ClearCase users? What other strategies can be implemented with the use of MultiSite?

Not all VOB data is replicated in a replicated environment. List the data that is and is not replicated. Why is certain data not replicated?

MultiSite concepts and terminology were introduced. What is a replica and how are they related to a replica family? What is store-and-forward facility and how does it function? How does MultiSite prevent complete chaos in a replicated environment while users are working and modifying data?

Review

- **Basic multitool Commands**

- ▶ What are the commands used to create a replica and keep it in sync with its siblings?
- ▶ Why would you have to run `restorereplica` and what would it cause if you did not?

- **The MultiSite Administration Console and `shipping.conf` file**

- ▶ Where can these tools be found?
- ▶ What kind of options can be set in each of these and how are they different?

- **Synchronization Jobs**

- ▶ What do the sync export and import jobs actually run?
- ▶ Can you specify any custom arguments in the jobs?



Some of the basic multitool command line commands were introduced. What commands would you use to create a replica and keep it in sync with the other siblings in the family? It is also very important to understand the correct procedures when restoring a VOB from backup. What command needs to be run and why do you need to run it?

The MultiSite administration console and `shipping.conf` file provide administrative functionality. What is the difference between the two tools? What options are set in the administration console and `shipping.conf` file?

Synchronization provide the functionality to keep all replicas in a family up to date and in sync with each other. What are the jobs actually running in the background? What custom arguments can be passed to the jobs when they run?

Summary

- **What is MultiSite?**
 - ▶ It provides a mechanism to replicate VOB data
 - ▶ It can be used to replicate data to multiple geographical locations
 - ▶ Can also be used as a disaster recovery technique
- **VOB data replication**
 - ▶ Not all data in a VOB is replicated
 - ▶ Certain data cannot be replicated due to local site specific values
- **Terminology and concepts**
 - ▶ The basic concepts of replicas and replication were introduced
 - ▶ A replica keeps track of operations with oplogs and epoch tables
 - ▶ Synchronization occurs with update packets that are shipped through shipping bays and shipping servers
 - ▶ Every object in a replica has mastership so that only one user can work on or with an object at a time
- **Basic commands**
 - ▶ Some of the basic commands were listed to administrate and get started with the product
 - ▶ Other troubleshooting commands and methods exist
- **Administrative tools**
 - ▶ The MultiSite control panel and shipping.conf file is where many parameters are set
 - ▶ The scheduled jobs provide automation of replica synchronization



MultiSite provides a mechanism to replicate VOB data across geographically separated sites within an organization. With MultiSite a disaster recovery scenario can be implemented by keeping copies of VOBs up to date for users in the event a system would go down.

It is important to keep in mind that not all data in a VOB is replicated. The reason why certain data cannot be replicated is because it contains values that are specific to a local site such as local paths on server systems.

The basic concepts of replicas and replication were introduced. A replica keeps track of operations with oplogs and epoch tables within the VOB database.

Synchronization occurs with update packets that are shipped through shipping bays and shipping servers to keep all replicas in a family up to date with each other. Every object in a replica has mastership so that only one user can work on or with an object at a time. A user must possess mastership of the object before it can be modified. This provides a mechanism to control modification of data to prevent complete chaos within a replicated environment.

Some of the basic commands were listed to administrate and get started with the product. Note that many other troubleshooting commands and methods exist.

The MultiSite control panel and shipping.conf file is where an administrator can define options such as custom shipping classes, e-mail notification lists, and routing hosts for update packets.

The MultiSite scheduled jobs provide automation of replica synchronization. Multiple jobs can exist to administrate replication to various sites. Custom sync scripts and parameters can be run with the batch files that the jobs run.

Additional resources

▪ Next steps

▶ Review additional documentation

- ClearCase information center
 - <http://publib.boulder.ibm.com/infocenter/cchelp/v7r0m1/index.jsp>
- Technotes
 - **About the ms_rtn directory**
 - <http://www-1.ibm.com/support/docview.wss?uid=swg21253621>
 - **Identifying ClearCase MultiSite packet fragments in version 7.0**
 - <http://www-1.ibm.com/support/docview.wss?uid=swg21258763>
 - **About the restore replica process**
 - <http://www-1.ibm.com/support/docview.wss?uid=swg21131381>
 - **About Feature Levels and ClearCase**
 - <http://www-1.ibm.com/support/docview.wss?uid=swg21119269>
 - **Set the ClearCase MultiSite scheduled job to send packets to a group of replicas**
 - <http://www-1.ibm.com/support/docview.wss?uid=swg21119733>



Now that you have completed the introduction to ClearCase MultiSite module, the next steps would be to review additional documentation. The ClearCase information center contains the ClearCase MultiSite Administration guide which will go into more details on the topics discussed in this course.

There are also technotes listed to help understand additional details on MultiSite concepts and functions. The ms_rtn directory is a directory in which packets are returned to in the event they cannot be delivered. Starting in ClearCase version 7 a new packet fragmentation feature was implemented. There are additional details about the restore replica process in the technote listed on this slide. Finally the technote about setting up a scheduled job to send packets to a group of replicas will go into greater details as to how to create a custom script to accomplish this.

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_RCCv7_IntroductionToMultiSite.ppt

This module is also available in PDF format at: [../RCCv7_IntroductionToMultiSite.pdf](http://RCCv7_IntroductionToMultiSite.pdf)



You can help improve the quality of IBM Education Assistant content by providing feedback.

Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

ClearCase ClearCase MultiSite IBM Rational

A current list of other IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

Rational is a trademark of International Business Machines Corporation and Rational Software Corporation in the United States, Other Countries, or both.

Microsoft, Windows, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2008. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.

