RepliWeb®
R-1
Version 3.3

Centralized
One-To-Many Distribution System
Over Dynamic IP Networks

User Guide

For Windows and UNIX operating systems

RepliWeb, Inc.

R-1 Version 3.3
User Guide

The information in this manual has been compiled with care, but RepliWeb, Inc. makes no warranties as to its accuracy or completeness. The software described herein may be changed or enhanced from time to time. This information does not constitute a commitment or representation by RepliWeb and is subject to change without notice. The software described in this document is furnished under license and may be used and/or copied only in accordance with the terms of this license and the End User License Agreement.

Please direct correspondence or inquiries to:
RepliWeb, Inc.
351 S Cypress Road, Suite 402
Pompano Beach, Florida 33060
USA

Telephone: 954.946.2274
Fax: 954.958.9787

Sales & General Information: info@repliweb.com
Documentation: docs@repliweb.com
Technical Support: support@repliweb.com
Website: http://www.repliweb.com

Table of Contents

1. Introduction

RepliWeb, Inc. is a full service technology company that develops innovative software for data replication and deployment. Its primary clients are S/M/L Enterprises, Technology Developers, Integrators and Global Service Providers.

1.1. The RepliWeb Family of Products

RepliWeb, Inc. offers a full range of robust, enabling technologies that put an end to the difficulties of maintaining and administering large volume content distribution, data replication and synchronization in both homogeneous and heterogeneous network environments.

A complete list of products and supported platforms can be found at RepliWeb's web site: http://www.repliweb.com.

RepliWeb Deployment Suite™ (RDS)

The **R**epliWeb **D**eployment **S**uite™ (RDS) provides a mission critical cross-platform scheduled content replication and synchronization solution for heterogeneous system environments. With RDS, the replication environment can consist of Windows NT / 2000 / XP, plus many flavors of UNIX. RDS is an extremely robust solution enabling the scheduled replication and synchronization of file systems over all networks, regardless of the content type, volume or production environment.

RepliWeb Deployment Suite™ for Network Attached Storage (RDS/NAS)

The RepliWeb Deployment Suite™ (RDS) for Network Attached Storage (NAS) is a highly secure, software-based, automated, file replication and synchronization technology for distributing business critical data between heterogeneous system environments (Servers, eCDN Appliances) and NAS devices, across any TCP/IP network. RDS/NAS is the version installed on the NAS.

RepliWeb Distribution Manager (R-1)

R-1 is a centralized, high-level, one-to-many distribution system, engineered to perform in complex and dynamic IP networks (LAN, WAN, Internet, eCDN, satellite). R-1 provides unattended, center-to-edge data deployment and synchronization that scales to thousands of target machines.

1.2. About the User Guide

Audience

This guide was written with the following in mind:

- The user is familiar with the various Microsoft Windows operating systems (Windows NT / 2000 / XP / 2003) as well as the UNIX operating system (should the replication job employ it) and their basic administration.
- The user has the necessary privileges for the computer(s) on which the software is to be installed. These will be described more fully throughout the User Guide.
- The user is familiar with Windows-style graphical user interfaces (GUI) and / or command line interfaces (CLI).

User Guide Organization

As mentioned, R-1 has been designed to operate between Windows environments, UNIX environments, and heterogeneous (mixed) environments. With that in mind, all administrative tasks can be performed on both UNIX and Windows machines.

The User Guide reflects this administrative flexibility and is divided into sections pertaining to R-1 administration through a graphical user interface (R-1 Console GUI), administration through the command line interface (R-1 Console CLI) and sections that are independent of the Console interface.

The Console component is explained in detail in the following section of the User Guide. Administration through the Console can include submitting replication jobs, viewing job progress, and aborting jobs.

NOTE: R-1 does NOT create any default user or password. All user names and passwords used in this guide are samples only.

Valid credentials should be used for Center and Edge connections.

2. Overview

This chapter will review RepliWeb's Distribution Manager – R-1, its architecture and features.

2.1. General Concepts

Mission

R-1 is a centralized, high-level, one-to-many distribution system, engineered to perform in complex and dynamic IP networks (LAN, WAN, Internet, eCDN, satellite based, over Unicast and Multicast). R-1 provides unattended, Center-to-Edge data deployment and synchronization that scales in volume of data, number of processes and number of target machines, employing heterogeneous (cross-platform) server environments (Microsoft Windows™, UNIX).

R-1 realizes that enterprise network environment is as diverse as the servers connected to it. R-1 is currently available for - and transparently interoperates between - the following operating systems: Windows NT (SP4 or higher), Windows 2000, Windows XP, Windows 2003, IBM AIX (4.3 or higher), Sun Solaris (2.7 or higher - SPARC), HP-UX (11.0 or higher), Red Hat Linux (kernel 2.2.x & 2.4.x - Intel x86), Open Server (SCO 5.0 or higher), FreeBSD (4.2 or higher), Compaq True64, NAS (Network Attached Storage) (IBM 200/300, DELL 715N).

R-1 distribution is Multicast enabled with a powerful Unicast complement. Integrated Network Awareness™ (iNAT™) technology allows R-1 to optimize distribution policies based on the available network infrastructure (Multicast, Unicast & hybrid networks), for "Application Level Multicast" functionality.

R-1 differs from conventional one-to-many solutions by taking the responsibility for resource intensive tasks related to critical distribution. R-1 Business Intelligence dictates policies based on content-value, transport-cost metrics and network resources, as well as other configurable parameters.

R-1 facilitates a totally unattended, background distribution process. Automation is rooted in a master "fire-and-forget" scheduler that supports priority, time, event and logic as policy definitions.

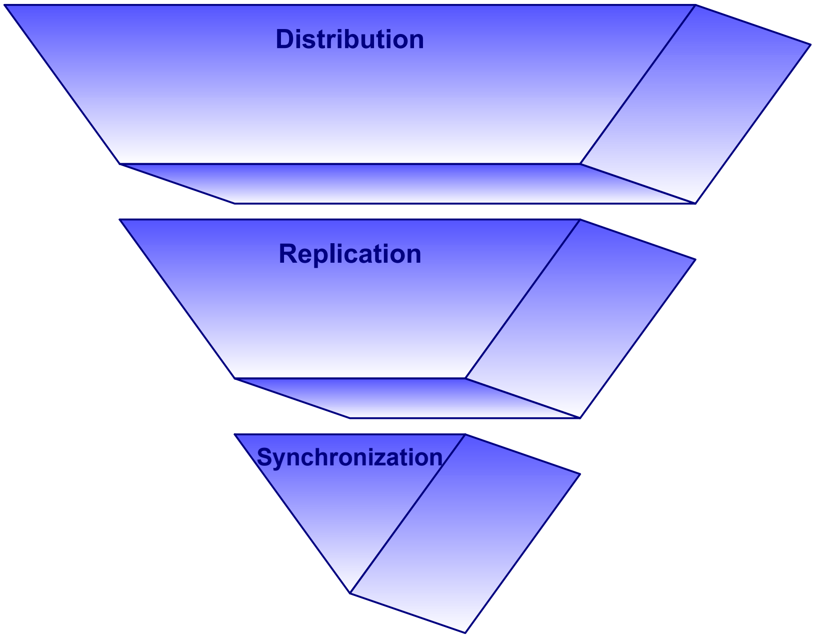
R-1 Features

- High-level, non-intrusive application - installation on production servers does not require reboot or taking applications off-line.
- No dedicated hardware required.

- Advanced, highly configurable scheduled automation.
- Integrated Network Awareness Technology™ (iNAT™)
- Platform Independent – supports most major operating systems
- Remote Management – detached management and control interfaces
- Decoupled reporting engine
- End-to-End authentication and security
- Data modeling determining the differences between Center and Edge data volumes
- Transport optimization computed on-the-fly on a per Edge-node basis
- Differential transfer algorithms distribute deltas between source and target file-systems
- True Edge synchronization - not only loads, but deletes data at target machines
- Integration pathway with third-party Multicast transport engines

2.2. Functional Description

R-1's logical structure may be explained by the following three-tiered pyramid:



1. The basic logical layer is the **Synchronization** layer, in which R-1 determines the minimum amount of data that needs to be transferred. By employing a synchronization-logic powered by RepliWeb's unique **C**omparative **S**napshot **T**echnology™ (CST™) engine, the synchronization time no longer depends on the total volume of data to be replicated, but is solely a function of the volume of changes.
2. The second layer is the **Replication** layer. Here R-1 performs accelerated, one-to-one, data replication by using parallel data streams and maximizing network throughput. It uses a tightly integrated transfer technology layer, providing integrity, recovery, reliability, bandwidth control, and differential transfer.
3. The top layer is the **Distribution** layer. One-to-many data distribution is performed, employing distribution policies based on the available network infrastructure. Here R-1 uses distribution-logic infrastructure powered by RepliWeb's unique **C**omparative **S**napshot **M**atrix™ (CSM™) engine. R-1 minimizes the distribution time by resolving the minimum amount of data that need to be transferred to each of the distribution targets.

   R-1 uses Multicast-enabled distribution with a powerful Unicast complement. Integrated Network Awareness technology™ (iNAT™) allows R-1 to optimize distribution policies based on the available network infrastructure (Multicast, Unicast & hybrid networks), for "Application Level Multicast" functionality.

Distribution is performed using three components:

- **Console -** Used to define and monitor distribution tasks between Centers and Edges.
- **Center** - The initiator of the distribution process.
- **Edge** - Systems that respond to Center distribution and replication requests.

2.3. Architecture

R-1's flexible architecture allows the user to run distribution and replication jobs regardless of network configuration. The target system can also be a Network Attached Storage (NAS) server. Cross-platform synchronization (Windows to UNIX or vice versa) is supported.

- An R-1 job replicates between two computers: a **Cente**r computer containing at least the R-1 **Center** component and an **Edge** computer containing at least the R-1 **Edge** component.
- Numerous systems may be involved in a single distribution job. In such a setup, one **Center** computer distributes data to multiple **Edge** computers.

Each component is independent of the others, meaning that a given system can be a Center, Edge, or Console. The components that should be installed on a given system depend on the role that the system will fill in the replication and distribution topology. Also, any given system can be of a different platform (Windows or UNIX), regardless of the role it plays or the R-1 components that are installed on it.

Terminology

The Center is the initiator of any distribution or replication job. Thus, any reference to the direction of data transfer is, by convention, relative to the Center.

An **Upload job** is a job in which data is being uploaded from the Center to the Edge(s).
A **Download job** is a job in which data is being downloaded from the Edge to the Center.

Distribution is always an **Upload job**. In the distribution process, the data transfer is from the Center to one or many Edges (a one-to-many operation).

The **Replication** process, always takes place between two systems only – a Center and an Edge (one-to-one operation), involving either the uploading or downloading of data.

Center

The Center is the initiator of any R-1 distribution job and must therefore be installed on any system that will initiate a distribution. Once a job is submitted to the Center through the R-1 Console (through the Console GUI or the Console CLI), the background services on the Center take over. Whether the job is set to run immediately or on a predefined schedule, the Center will ensure that the jobs are initiated without any further user interaction.

The Center operates transparently as background processes regardless of the operating system it is installed on. This ensures that a distribution job can be successfully completed whether a user is actively logged on to the system or not.

NOTE: On UNIX systems the Center runs as background processes, while on Windows systems, it runs as services.

Edge

The Edge component must be installed on any system that responds to distribution and replication requests initiated by a Center. It is a passive element that runs as a background process, and is awakened by a distribution request originating from a Center. It then allows R-1 to perform the distribution, again, regardless of whether a user is actively logged on or not.

In a distribution process, more than one Edge may take part as the receiver of the distributed data. In a replication process, only one Edge will be involved, either as the receiver or the sender of the data.

NOTE: On UNIX systems the Edge runs as background processes, while on Windows systems, it runs as services.

Console

The Console is the main means for an end-user to interact with R-1. Whether using the R-1 Console command-line interface (CLI) or R-1 Console graphical user interface (GUI), the Console is a “window” to the Center.

The Console connects the user to any system with the Center component installed and allows the user to perform all forms of administrative tasks with respect to replication jobs defined on the Center.

The Console does not need to be installed on the system with the Center component installed. It can be installed on any system that can form a TCP/IP session with the Center (i.e. via LAN, WAN, VPN or a dial-up connection). This means that there can be multiple Consoles simultaneously administering R-1 jobs on a given Center.

- If the Console is installed on a Windows system, both the R-1 Console CLI and the R-1 Console GUI can be used to administer jobs.
- If the Console is installed on a UNIX system, only the R-1 Console CLI can be used.

Jobs can be administered using a combination of the two operating systems. This is illustrated by the following scenario:

The R-1 Center component is installed on a UNIX system at the office. A Console has been installed both at the office, and on the Network Administrator's home Windows PC. This allows the administrator to monitor jobs from both the home and the office. At the office, the Console is on a UNIX machine (the same machine as the Center), meaning that only the CLI can be used. However, once at home, the administrator can do all R-1 job administrating through either the CLI or the GUI.

A Console always displays information relayed from the Center it is connected to and the Edge involved in the replication process.

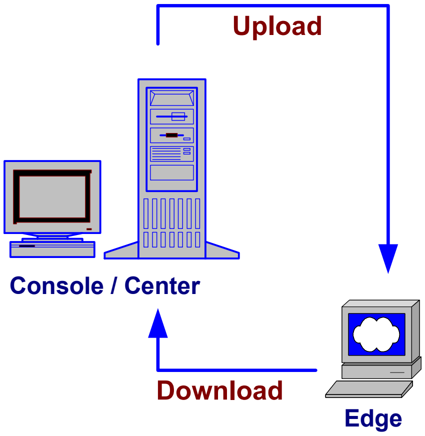
NOTE: R-1 job administration is completely independent of the systems involved in the distribution. For example, although a job may be initiated to mirror a directory between UNIX System (A) and UNIX System (B), all administrative actions pertaining to the job can be fulfilled through either a UNIX system, or a Windows system. The only requirement is that the R-1 Console component be installed on the system in question.

2.4. Distribution & Replication Examples

To demonstrate R-1's flexibility, a few common scenarios are described below. Since R-1 is cross-platform in nature, only the "components" are mentioned. Each system in the topologies mentioned, whether acting as a Console, Center or Edge can be a Windows or UNIX system. This is independent of the other systems involved in the synchronization process.

R-1 operates in complex and dynamic IP networks (LAN, WAN, Internet, eCDN, satellite based, over Unicast and Multicast), and at each point in time determines the best policy for transferring the data. The network connection in the following examples is not specified, as it has no effect on the synchronization process itself.

1. Basic Replication Job

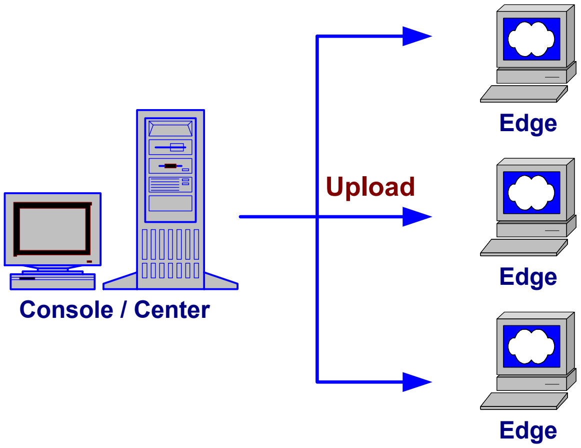


The figure above illustrates a basic R-1 replicating architecture.

The Console and the Center are installed on the same machine. Since the system will initiate the replication, it must have the Center component installed on it. Also, since the jobs will all be submitted and monitored from this system, it must have the Console component installed on it.

The Edge component is installed on a different machine in a different site.

2. Basic Distribution Job

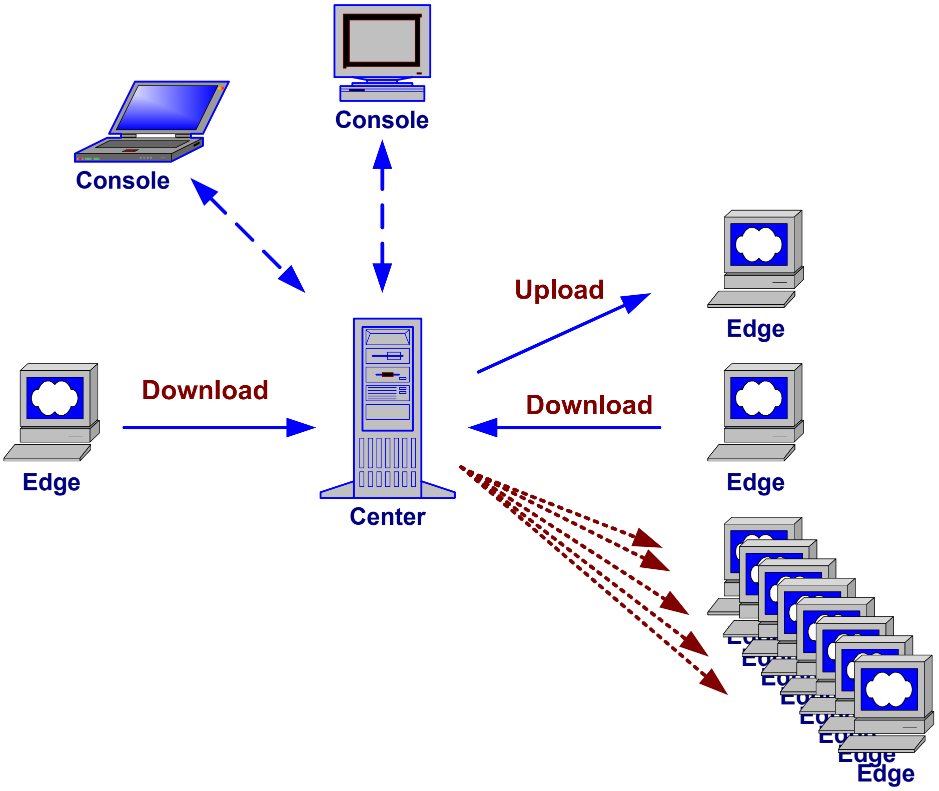


This example demonstrates a distribution topology.

The Console and Center are installed in the company’s central office. Data is being distributed on a daily basis from the company’s site to all its servers around the world.

All distribution jobs in this topology are Uploads jobs. Data is being transferred from the Center to the Edges. No data is being downloaded from any of the Edges to the Center.

3. Replication and Distribution Topology



In this example, hybrid architecture is deployed. A web-site development company updates its web site on a daily basis.

The development team works on a separate site, updating a pre-production machine. The Center initializes a replication job, to mirror this machine to a centralized site. Then it initializes a distribution job, uploading the data to web servers around the world. The network connection to the servers varies between LAN and satellite connections.

In addition, data is being uploaded to and downloaded from other servers of the company, using one-to-one replication processes.

In this scheme, there is more than one Console. The jobs are defined in one place and monitored in a different place, employing the strength and flexibility of the R-1 architecture.

2.5. Synchronization Features

Synchronization Logic

R-1's default synchronization-logic is to mirror a target directory with a source directory. Some situations, however, are better suited for other synchronization-logic. These include **Purge** logic and **Backup** logic. The nuances of each are explained in detail, later in this guide.

Security

R-1 is a completely secured distribution solution, which overcomes the traditional problems associated with Multicast transports. R-1's security mechanisms allow using the Internet and Internet-based VPNs and WANs as an efficient distribution channel, without the concerns of data loss, pilferage or malicious impersonation. R-1 uses SSL end-to-end, making integration with other systems seamless.

R-1 allows authenticated and encrypted data transfer of valuable digital assets between hosts. All access is denied by default, unless specifically permitted and is granular to the file level. Trusted IP addresses, subnets, users and schedules are supported, as well as the total anonymity of user/password/directory information from one host to another, thus allowing for secure transport between untrusted networks. This authentication proxy mechanism adds a layer of autonomy to B2B content distribution, enabling hosts that do not trust each other to synchronize massive content stores without having to divulge anything beyond the machine name or IP address, virtual user and virtual password.

Scalability

R-1 scales to handle infinitely large content repositories without sacrificing performance. R-1 uses RepliWeb's Comparative Snapshot Technology™ (CST™) to shift the burden of synchronization from the network to the servers. CST™ guarantees that only the absolute minimum volume of data necessary to perform the synchronization will be transferred, without burdening the networks with nonessential protocol traffic.

2.6. Replication Features

Recovery

R-1 operates on a “fire and forget” mind set. Once a job has been properly submitted, R-1 will do everything it can to ensure its successful completion. R-1’s recovery procedures are user configurable and will recover from any type of network failure or system crash. Whether it is a network “time-out”, system crash, reboot, or simply loosened network cables, as soon as the systems are back up and running, R-1 will recover and resume all of the interrupted replications.

Every distribution job on a Center runs completely independently of the others, and every stage of a job is independent. If the job failed during a file transfer, the transfer will resume from the exact point of failure within that file, only resending the last block transferred. If, during the down time, the target file was modified at all, R-1 will resume the transfer from the last point of match within the two files. (This is a situation that occurs all too frequently with Operating Systems that employ write-caches).

R-1 makes it possible to control every aspect of the recovery process, ranging from the number of retries to the time between recovery attempts.

Rollback

Rollback functionality provides a means of effectively undoing the effects of multiple replications and distributions on a target system to a user-specified ‘point-in-time’. The benefit of such a system over a simple ‘archiving’ of specific files is its ability to specify a time relative to the targets (e.g. 8 hours ago) and systematically undo the effects of all scheduled or event-driven changes that were replicated. The use of relative time as opposed to absolute time (i.e. DD/MM/YY HH:MM) eliminates the need to account for target systems across different time zones, thereby reducing the complexity of a rollback for the system administrator.

Rollback functionality is employed through the ‘recording’ of all replication related changes on a target system, followed by potentially ‘rolling back’ the content at a later time. The recording phase occurs prior to any propagation of changes to a target system and includes the recording of creation, deletion and modification to files and directories (including ownership and permission changes). Recording can be defined to take place locally on each target system, or on a network-wide file server defined through a UNC share-point (e.g. \\nasstorage\R1_recording) – effectively providing unattended and efficient tape-less backup and restoration. System administrators also custom define the recording history to a time relative to the present, for example, specifying that a distribution should be able to rollback to a state 24 hours in the past. R-1 will then only keep the changes that occurred in the last 24 hours, and automatically purge recorded changes that have expired.

Data Transfer

To achieve the required level of mission-critical reliability and security, R-1 employs a robust transport layer allowing the use of the best transfer method for each network topology and characteristics.

- **LAN Transport** – Employs the most out of a LAN connection, geared for speed and maximal throughput.
- **WAN Transport** – Maximizes the WAN connection, providing:
    - **Recovery** - An interrupted transfer will continue from the exact point of failure within the file, down to the block level.
    - **Integrity** - R-1 can perform an integrity check on each and every file transferred to ensure the accuracy of the transfer.
    - **Compression** – Transparent, asynchronous compression provides excellent performance over wide area networks and eliminates the need for packing and unpacking phases when transferring large volumes.
    - **Bandwidth Control** – Complete control over bandwidth consumption allows execution of the entire operation without having interactive system users experience high network latency, or web site visitors experiencing slow server response times. (See enhanced explanation below).
    - **Differential Transfer (block level)** – The data transfer can be limited to only the differences between the source and target files, reducing both transfer time and bandwidth costs.
- **File Preservation** - NTFS permissions, disk shares, UNIX file / directory security and UNIX UID / GID information can all be preserved during an R-1 job, regardless of the transport layer being used.

Multiple Data Streams

R-1 can perform the data transfer using multiple data streams. In the vast majority of replication environments, the use of concurrent streams will significantly reduce the replication time. This is especially true for replications performed over an Internet-based Virtual Private Network (VPN) where the limiting data transfer rate is much lower than the theoretical maximum.

When replicating over VPN or a high-speed LAN, transfer performance can be boosted when performing simultaneous parallel transfers of partial file lists. In a high-speed LAN environment, using data streams will result in maximizing the network throughput. When used on VPN environments (over the Internet), performance is maximized to nearly the network limit defined by your VPN connection.

Bandwidth Limit

Network activity that includes massive data-transfer operations can result in poor response time for interactive network users and network applications. To avoid this bottleneck in network performance, utilize the Bandwidth Limit feature.

Bandwidth Limit enables specification of the maximum line bandwidth that a particular operation can use. Each job can have its individual bandwidth priority that is set by the network administrator.

Absolute Bandwidth Limit

When using an absolute bandwidth limit, the job transfer rate is adjusted so that it does not exceed the specified bandwidth load.

For example, assume that interactive users are using a 256 Kbps line. While these users are connected, it is necessary to replicate a directory tree with a net difference of 1GB. If this is attempted while the users are still connected, network performance and response time will drop significantly.

By selecting an absolute bandwidth limit, the user can ensure that the R-1 transfer will never exceed its allocated absolute bandwidth.

Example:

> A network has a 256 Kbps line. Since this particular replication is low priority, the user chooses to impose an absolute limit of 56 Kbps. Thus, this job will never exceed a transfer rate of 56 Kbps, leaving 200 Kbps for network users and other applications.

Relative Bandwidth Limit

Relative bandwidth control limits the replication to a given percentage of the available line bandwidth. The total bandwidth used by the job will never exceed the specified percentage of the overall line bandwidth. Increased network traffic, in the form of additional replications or other network activity, will decrease the available bandwidth, and the Bandwidth Limit function will reflect that reduction in the bandwidth used by the job. If network activity subsequently decreases, the job will automatically increase its bandwidth usage until it reaches the set bandwidth percentage.

Bandwidth Limit and Time Frames for a Specific Edge Distribution

As network activity changes over the course of a day, the amount of bandwidth available for a given replication job can change. This Bandwidth Limit option allows the flexibility to specify different bandwidth limits for different time frames on the same replication job. This enables the operation to make use of more bandwidth as network resources become available, or less as the demand for bandwidth from other activities increases.

Example:

A large synchronization job is activated during peak activity hours, but its activity continues through a period of low network activity. This operation can be specified with two bandwidth limits: a lower limit which will be in effect during the time of peak network activity, and a higher limit during off-peak hours. The first bandwidth limit will be in effect in a time frame extending from 8:00 AM (08:00) to 6:59 PM (18:59), while the second bandwidth limit takes effect in a time frame extending from 7:00 PM (19:00) to 7:59 AM (07:59).

Differential Transfer

R-1 allows the transferring of only the differences between two files residing on the source and target. Both files are compared and only those blocks of data that are different are subsequently transferred. Thus, network load, transfer time, and network data exposure are reduced to the minimum required to synchronize the two files.

When this feature is enabled, the user must specify a minimum file size.

There is no advantage at all in using the Differential Transfer option with files smaller than 64K.

Flexibility

R-1 is ideal for heterogeneous network environments running UNIX, Windows or a combination of both. Businesses utilizing distributed development, application, production and web servers gain maximum flexibility with the R-1 modular architecture.

R-1 enables concurrent “push and pull” operations that can be logically grouped to automate business processes. R-1 uses simultaneous data streams to maximize network throughput, overcoming the inherent latencies in TCP networks.

Scheduling

R-1 facilitates a totally unattended, background distribution process. Automation is rooted in a master fire-and-forget scheduler that supports priority, time, event and logic as policy definitions.

Continuous Updates

R-1’s **Continuous Update** event-driven replication, significantly improves performance in projects requiring immediate data consolidation. The RepliWeb Event Driven Agent - REDA is an event driven process - not a device driver - that detects, collects, and filters the events on the source data that are relevant to each of the distribution targets – subsequently directing how R-1 responds to the events.

Through the implementation of the REDA over a scheduled distribution solution, R-1's Continuous Update overcomes the pitfalls of standard real-time replication solutions by regularly performing a full analysis and repair of the target system(s) involved at fixed intervals. The REDA framework allows it to be aware of the number of events generated, ensuring that a flood of events does not overload the system - If such a scenario does present itself, the REDA forces a full synchronization ensuring that a system is always up to date.

For more information on Continuous Update jobs, refer to **Appendix H**.

2.7. Distribution Features

Unique Architecture

R-1's multi-tiered architecture allows complete freedom and customization particularly for complex implementations. An R-1 deployment is comprised of the following components: Edge, Center, Management Console and an optional API Layer.

The distribution process running on the Center may distribute to as many Edges as desired. (The number of Edges is only limited by the R-1 license). The Edge is a passive component that is inactive until called by an authenticated Center. All components are detached from each other and platform agnostic.

A fourth tier may be added for greater flexibility in managing multiple systems simultaneously. The API layer may be used to seamlessly integrate R-1 functionality into external enterprise applications, such as third party web and management systems.

Business Intelligence

R-1 can make scheduling and transport decisions based on external events. A key example is time-shifting Multicast distribution of files over a satellite uplink, to align with the lower rates offered during off-peak hours. This module incorporates costing and content-value decisions while interacting with the other modules to maximize the performance and ROI.

Infrastructure

Integrated Network Awareness Technology™ (iNAT™) provides R-1 with a powerful set of tools to enable it to decide optimal network routing and transports for every distribution. Any hybrid TCP/Multicast network supports R-1 transparently. R-1 maximizes the effectiveness of the network by making the appropriate transport decisions in run-time.

Multicast

R-1 uses a set of properties to enable Multicast data transfer. Multicasting greatly reduces transfer time by improving the distribution process. It is specified for each R-1 Edge whether it is Multicast enabled. R-1 will, by default, send content using Multicast where possible. The distribution method (Multicast / Unicast) is determined dynamically for each file in the Comparative Snapshot Matrix (CSM™) based on predefined job parameters.

> **Minimum Quorum** - A job is ready to be a part of the Multicast transfer once it has completed building its CSM™. The distribution job will wait for a minimum number of jobs to reach that stage before calculating which files should be sent

with Multicast and which with Unicast. There is a time limit for the Quorum to assemble.

The value represents the number of edges that must have completed their own local snapshots and are ready for the next stage of the deployment, either the file transfer itself, or pre-transfer processing. Therefore, if a distribution has a total of a hundred (100) edges, a minimum quorum of eighty (80), specifies that once eighty edges have taken their respective snapshots and have transparently sent them to the Center. The Center can begin to form the CSM for the distribution.

Wait Quorum For – The field specifies the amount of time the distribution should wait to reach the Minimum Quorum before failing the distribution due to an unattained Minimum Quorum. Once the Minimum Quorum has been reached, the **Wait to Maximize** option indicates the length of time (in seconds) the distribution should wait for additional edges to report before the CSM for the distribution is generated.

The Minimum Quorum does not reflect the minimum number of edges capable and ready to receive a multicast data transfer. Any Edge that has not reported by the time the CSM is generated, will not receive files broadcast with Multicast. If the distribution is set to Multicast without Edge or File Level Complement, the Edge will be excluded from the current distribution.

The CSM outlines whether a given file should be delivered over Multicast, and if so, to which Edges. The CSM is then being filtered according to criteria files must satisfy in order to be multicast. The multicast criteria is based on **TTL** (in hops), a **File Quorum** (multicast the file only if it must be sent to *n* Edges) and **Minimum / Maximum File Size** (multicast the file only if its size in KB is in range.) Any files remaining in the matrix after the filtering are then multicast.

TTL - When a packet is incident on a network router, the router must decide whether to relay the packet to the next packet, or discard it. Each time a packet reaches a router, the router subtracts at least one count from the TTL header, when a packet with a TTL value of zero reaches a router, it is discarded.

By Convention:

- 0 is restricted to the same host
- 1 is restricted to the same subnet
- 32 is restricted to the same site [Default]
- 64 is restricted to the same region
- 128 is restricted to the same continent
- 255 is unrestricted

What is done with the files no longer in the matrix is dependant on the distribution type (Multicast / Multicast with Unicast complement.) If in Multicast only mode, only files remaining in the CSM will be transferred in multicast. Thus, files removed from the CSM will be not be transferred.

The Multicast **Edge / File Level Complement** transfer logic adds two layers of Unicast complementing to a Multicast deployment.

Employing **File Level Complement** will instruct R-1 to Unicast any files removed from the CSM due to filtering options of the respective Edges. This complement option only applies to Edges that have reported for the quorum and are included in the Multicast stage. A typical scenario that would benefit from this feature would be when a portion of the data consists of many small files and the multicast network is a satellite link. The latency in the satellite makes it unreasonable to use it to multicast the many small files, so it is agreed that any files under 500 KB will be Unicast. A filter is set up to multicast files above 500 KB, and the File Level Complement feature is enabled. When creating the CSM, files smaller than 500 KB will be removed from the multicast matrix and be sent in Unicast to each Edge requiring the respective file.

The second level of Unicast complement, **Edge Level Complement**, instructs R-1 how to handle an Edge that is excluded from the multicast because it did not report in time for the CSM stage. Before it was mentioned that any edges that have not reported when the Minimum Quorum has been reached would be excluded (aborted) from the current distribution. With Edge Level Complement, every Edge that does not report in time for multicast distribution will have all its data transferred in a parallel Unicast stream.

An Edge that was part of the Multicast transfer but failed to receive content will continue and recover using Unicast.

Files that were marked for Multicast, and their Edge cannot receive Multicast transmission (or did not meet the Multicast Quorum) will be submitted using Unicast, if the File Level Complement option is selected, even if the Edge Level Complement option is not.

File Packing - When the source directory contains many files, it is recommended to pack them into larger packages for better performance, as the overhead of multicasting a file take longer than actually transmitting that file. Multicasting just a few large files it is recommended not to use file packing.

Openness

R-1 is an open system, and can distribute content using external transfer tools, in addition to its internal transfer tools. As well as R-1's Multicast and Unicast transfer engines R-1 is designed for integration with other multicast technologies thus leveraging the benefits of customers' existing infrastructure.

Parallelism

R-1 uses a set of dynamic algorithms that simultaneously distribute data via Multicast (if applicable) and guarantee completion of each transmission with unique differential transfer algorithms that distribute only the deltas between source and target file-systems. These parallel TCP transfers correct errors in the Multicast datagram in run-time, over any number of Edge nodes.

Heuristics

Even the most advanced enterprise solution becomes a management problem if it lacks the power to solve problems independently. R-1 has the ability to identify difficulties and make recovery decisions based on network events, as well as from other external sources.

2.8. Distribution Process

A distribution process consists of the following steps:

Pre-Distribution Processing

This stage performs the pre-distribution processing stages designated by the user on the Center and/or on the Edge(s). R-1 can execute scripts and programs both locally and remotely to give the network administrator the opportunity to prepare the source and target systems prior to the distribution.

The pre-distribution process may also include a download/collection/consolidation operation, providing a unified process of downloading data from one location and distributing it over the network.

Planning

The planning stage is where R-1 differentiates itself from other distribution and replication technologies. Implementing CSM™ and CST™, snapshots of the source and target directories are taken, listing file sizes, modification times and dates, security permissions, and shares (where applicable). The remote snapshot is then transferred to the Center where it is analyzed locally looking for differences between the source directory tree and target directory tree. The analysis results in a Comparative Snapshot Matrix™ that lists the absolute minimum that must be transferred to perform the requested synchronization. The Comparative Snapshot Matrix™ is a matrix of files and Edges, listing exactly which files need to be transferred to which Edges in whole, which files need only security permissions updated, and so forth.

Compare this method to the alternative: Look at the first file in the directory, send a request to all Edges to see if the file exists. If it does, reply to the request asking whether the file should be overwritten, or ignored, if it doesn't, then start to transfer the file. Repeat this for every file in the directory tree for every Edge involved in the distribution process. Not only is this method highly inefficient, it also drastically increases network packet traffic. CSM™ leads to faster synchronization times, lower bandwidth usage fees, and reduced network stress.

Transfer

The transfer stage uses the Comparative Snapshot Matrix™ generated in the planning stage to synchronize the systems. Depending on the content and topology, R-1 can use one of the internal transfer methods to transfer the data, optimized for WAN environments, or for LAN environments. This can lead to greatly reduced transfer times over conventional file transfer protocols.

R-1 encompasses Multicast-enabled distribution with powerful Edge and File Level complement. Integrated Network Awareness Technology™ (iNAT™) allows R-1 to optimize distribution policies based on the available network infrastructure (Multicast, Unicast & hybrid networks), for “Application Level Multicast” functionality.

R-1 will distribute files using Multicast where possible. If Edge content cannot be distributed with Multicast to a certain Edge, the Center will send the content to that Edge using Unicast, while other nodes get their content via Multicast. R-1 will check for valid Multicast transfer at the beginning of each distribution process (depending on the user’s configuration). It may happen that in the next distribution process that Edge will be able to receive Multicast transmission, and content will be sent to it using Multicast, thus improving the distribution process.

Post-Distribution Processing

Like in the pre-distribution process, in this stage R-1 performs operations designated by the user on the Center and/or on the Edge(s), following the distribution process. R-1 can execute scripts and programs both locally and remotely to give the network administrator the opportunity to complete the process by performing various tasks on the systems that took part in the distribution process.

Clean Up

Any remaining job-specific temporary files generated by R-1 during job execution are removed from the local and remote systems. After Clean Up completes, R-1 executes any On Exit commands requested. These can include notification via e-mail as to the status of the completed job or any other customized procedure on the Center or Edge(s).

NOTE: The Distribution process is somewhat different when running Continuous Updates Jobs. Please refer to **Appendix H** for more information.

3. Creating a Distribution Job: GUI

In order to initiate a job, its parameters must be defined. The parameters control every aspect of the job including the data to be replicated, bandwidth consumption, scheduling, and Center / Edge pre and post-distribution command processing. All job definition and configuration may be performed using R-1's Graphical User Interface (GUI) called the **Console GUI**. The Console GUI will only be available if R-1's Console component was installed on a Windows system. Each parameter is discussed in detail in this chapter.

Some distribution options are platform dependent. Since the R-1 Console GUI can connect to both a UNIX and Windows Center, platform specific properties will be clearly outlined.

NOTE: Users who are not familiar with R-1's features and components are advised to experiment with previously backed up files and directory structures only.

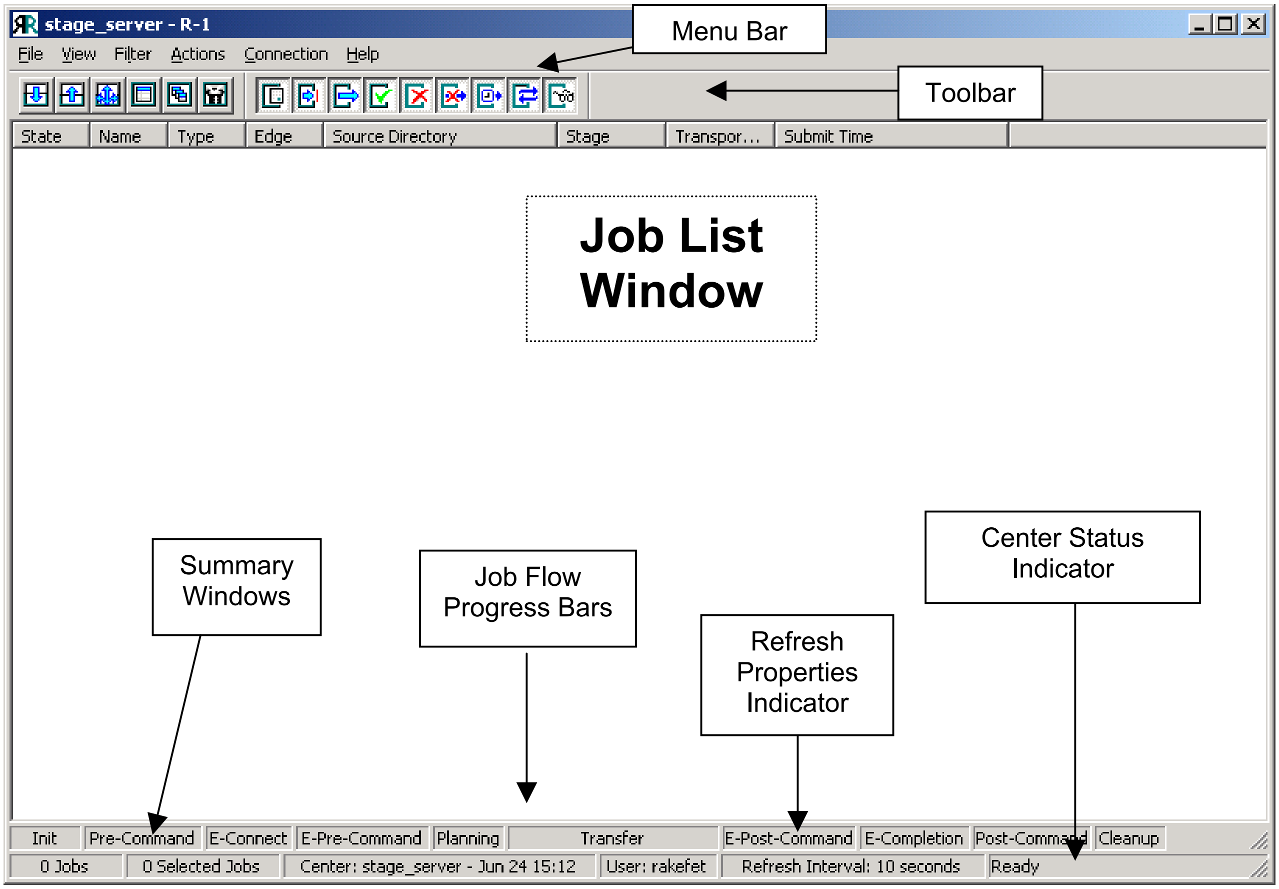
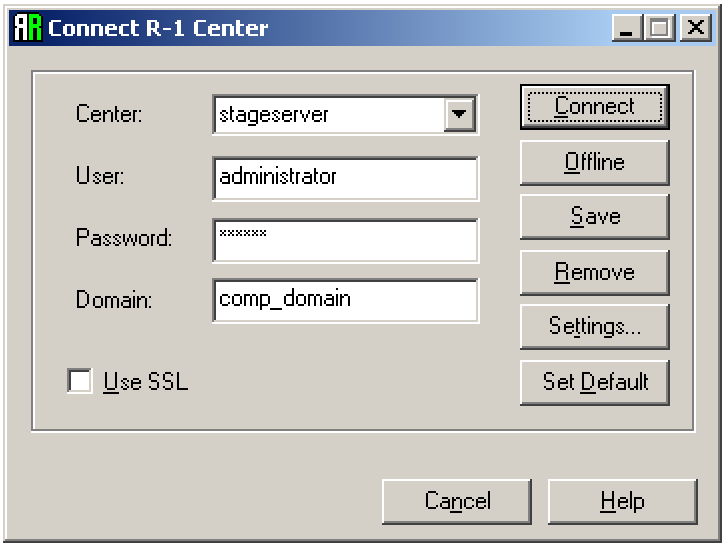


Figure 1 - The R-1 Console

3.1. Connecting Via the R-1 Console GUI



Center: stageserver

User: administrator

Password: ******

Domain: comp_domain

□ Use SSL

Figure 2: Center Log On

The first step in creating a distribution job is opening the R-1 Console and connecting to a machine with the R-1 Center component installed on it. This will allow the user to view, submit, and administer any jobs associated with the Center in question.

A dialog box appears with the following fields:

- **Center**- Enter the host name or IP address of the R-1 Center being accessed by the Console. R-1 saves, according to user settings, the connection credentials of each of the Centers the Console is connecting to, so the Center can be selected from the list if the user already connected to it.
- **User** - Enter the user name that all jobs on the Center will use to perform the synchronization.
- **Password** - Enter the password for the account specified in the **User** field.
- **Domain** - If **user** is part of a domain, enter the domain here, if it is not, leave this field blank.
- **Use SSL** – Specify whether to connect to that Center using SSL security.

  For explanation about R-1 security, see the **Overview Chapter**.

NOTE: The reason one connects to a Center in the first place is to have the ability to submit, modify, or monitor jobs. For security purposes, authentication for the respective Center must take place. This means first supplying either the IP address or DNS name of the Center in question, and a real (domain or local) account and password for that computer.

Any operation that takes place on a system (Windows or UNIX) must have a user context it is operating under. When a user is logged in to the system, the user context is that user. However, when a background process (such as an R-1 job) performs operations, it must take on a user context. The account supplied to the Center will be the account under which R-1 will perform all tasks on the Center machine. If it were to write files to a certain directory on the Center, it would need standard NT security privileges to write to the given directory.

This also means that any pre / post commands on the Center will be run under the above user's security context.

This window allows saving the list of Centers the user connects to from the Console. The list is saved locally and is user specific. Connection credentials can include the password with/without confirmation if the user selects to do that; the password is saved in an encrypted format.

When all the necessary fields are filled in, click **Connect**.

- **Connect** – Connects to the Center specified. Connection credentials are saved according to the settings specified by the user. The user can elect to always add the Center's connection credentials to the list upon successful connection to the Center.

- **Offline** – Disconnect from the current Center (if already connected). Do not connect to any Center. Allows the creation of local job templates.

- **Save** – Save connection credentials according to the settings specified by the user regardless of the validity of the values entered.

- **Remove** – Remove a Center's connection credentials from the list.

- **Settings** - Specify whether the connection credentials will be saved with/without the password, and whether saving confirmation is required. This window also allows specifying that the connection credentials will be automatically saved upon successful connection to the Center.

  NOTE: When saving passwords, the password is always encrypted!

- **Set Default** - Use the current connection credentials as the default for R-1 to use the next time the console is opened.

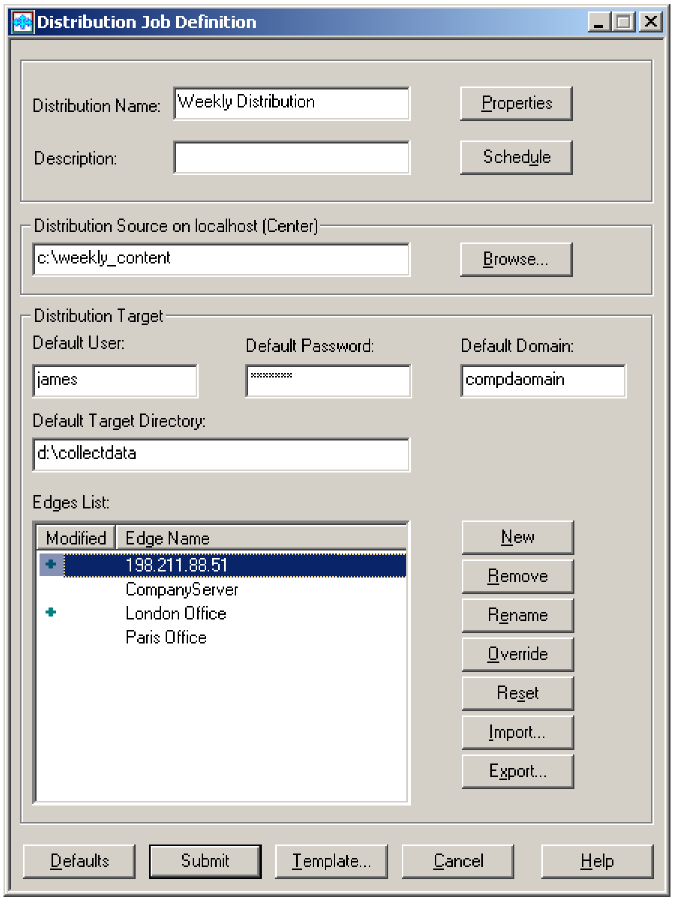
NOTE: When connecting to a Center as a specific user, one will only see jobs that were submitted by that user. **user** bob will only see jobs that were submitted when connected to the given Center as **user** bob.

The user name entered in the **user** field is also case-sensitive, so **user** Bob will not see any jobs submitted by **user** bob and vise versa. This is true for both Windows and UNIX systems.

3.2. Distribution Job Definition and Properties

A distribution job is defined in the **distribution Job Definition** window. This window is reached by selecting the Distribution icon on the button bar, or selecting the option from the **File** menu. Frequently performed jobs may be saved as a template. Templates are described in detail in the following sections. In effect, a template simply saves the job definition, including all options selected and information entered, for later submission or editing.

Connection to the Center automatically identifies one of the machines involved in the distribution (the Center is the source for distribution and upload jobs and, conversely, the target for a download job). The above convention holds regardless of where the Console is.



Distribution Name: Weekly Distribution

Description:

Distribution Source on localhost (Center)

c:\weekly_content

Distribution Target

Default User: james

Default Password: *******

Default Domain: compdaomain

Default Target Directory: d:\collectdata

Edges List:

Modified	Edge Name
+	198.211.88.51
	CompanyServer
+	London Office
	Paris Office

Figure 3: Defining a Distribution Job

The job definition fields are:

- **Distribution Name** – Enter the name of the distribution job.
- **Distribution Description** – Enter the job's description.
- **Source Directory** – The Distribution's Source directory.

- **Default Target Directory** – The Distribution's default target directory. If not specified in each of the Edges (override) the same target directory will be used in all Edges.

  The source and target **Directory** paths must be specified in full. For example, if the following path is entered as the target directory for a distribution job – `d:\collectdata` - all distribution activities will occur in the `collectdata` sub-directory on drive 'd'). If this directory does not exist on the target, R-1 will create it.

NOTE: It is **EXTREMELY** important to remember that all path locations and commands **MUST** be written in a native format, i.e., if the source is a Windows machine, the source path must be in Windows format; if the target is a UNIX machine, the target path must be in UNIX format.

See **Cross Platform Formats and Escaping** description in **Chapter 5** for detailed explanation.

- **Distribution Edges –** Define all properties related to the Edges the job will connect:
  - **User** – Enter the username that all jobs on the Edge will use to perform the synchronization.
  - **Password** – Enter the password for the account specified in the **User** field.
  - **Domain** – If the Target platform is Windows, the target Domain will also need to be specified.
  - **Default Target Directory** – Specify the Target distribution directory on all Edges. This may be overridden for each or all Edges.
  - **Edges List** - The Edge(s) (source or target, depending on the job type) must be specified. The Edge may be identified by name or IP address. This represents a system that has the Edge component installed.
    - **New –** Add a new Edge (target) to the distribution job.
    - **Rename–** Rename an Edge.
    - **Remove–** Remove an Edge (target) from the distribution job.
    - **Override** – Opens the Edge Override dialog box.
    - **Reset** – Reset all Edge Override changes. All Edge properties will be taken from the main job.
    - **Import** – Import a text file listing Edges.
    - **Export** – Export the job's Edge List to a text file.

    Modified Edges will be marked by a + sign, marking that some of their properties are different than the default properties of the job.

It was noted above that the user inputted in the Center settings dialog box gave the user context that all operations on the Center would be carried under. Similarly, the **User** entered in the Edge box defines the user context that all operations on the Edge will be carried under.

Note that the user entered above must have whatever permissions necessary to write / read / modify / delete in the directory specified in the Edge. Any pre / post commands on the Edge will be run under that user's security context.

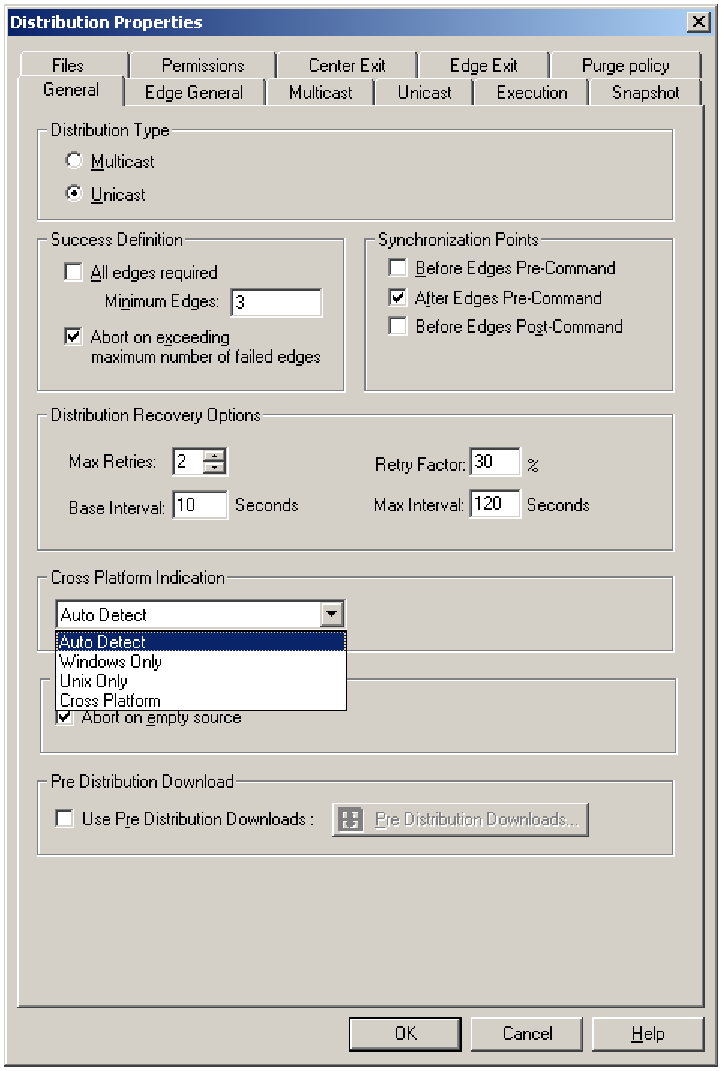
Note also that the Center's user credentials do NOT have to bear any relation to the Edge's; there is no linkage between the user context on the Center and that on the edge.

NOTE: A rule of thumb is that if the user entered in the Center or Edge field can be actively logged-in to the systems with the Center and Edge systems respectively and create and delete files in the respective directories, R-1 can too.

- **Defaults** – Press this button to save all entered properties as defaults for following job submissions.
- **Submit** – Submits the specified distribution job.
- **Template** – Opens a dialog box allowing selecting and saving job definition templates.
- **Properties** – Opens the Distribution Properties dialog box.
- **Schedule** – Opens the Schedule dialog box.

Once all job properties have been defined, proceed to submit the job or, if desired, save the job as a template. Additional parameters can be set through the job **Properties** and **Schedule** screens.

3.2.1. Distribution General Tab



Distribution Properties

Distribution Type

○ Multicast

● Unicast

Success Definition

☐ All edges required

Minimum Edges: 3

☑ Abort on exceeding maximum number of failed edges

Synchronization Points

☐ Before Edges Pre-Command

☑ After Edges Pre-Command

☐ Before Edges Post-Command

Distribution Recovery Options

Max Retries: 2

Retry Factor: 30 %

Base Interval: 10 Seconds

Max Interval: 120 Seconds

Cross Platform Indication

Auto Detect

- Auto Detect
- Windows Only
- Unix Only
- Cross Platform

☑ Abort on empty source

Pre Distribution Download

☐ Use Pre Distribution Downloads :

Figure 4: General Distribution Properties Tab-Defining Job Type and Recovery Options

NOTE: The Console is connected to a Windows Center. Screen will vary slightly if the Center is UNIX.

- **Distribution Type** – Specify the distribution type:
  - **Multicast** – Distribute files using Multicast. Properties need to be set in the **Multicast Tab**.
  - **Unicast** – Distribute all files using Unicast. Properties need to be set using the **Unicast Tab**.

- **Success Definition** – It may happen that not all Edges are available at the execution time. Define here the number of Edges required for the successful completion of a Job and when the distribution job determined its completion status.

    - **All Edges Required** – Successful distribution to all Edges is required for a successful completion of the Distribution Job.

    - **Minimum Edges** – If distribution to the minimum number of Edges succeeded, the Job succeeds.

    - **Abort on exceeding maximum number of failed edges** – This optimization option specifies **when** the distribution process determines its completion status.

      By default, once an Edge job fails, the distribution process evaluates its completion status based on the **Minimum Edges** or **All Edges Required** options, as specified by the user. If the conditions are not met, the job will fail.

      When this option is used, if one or more of the Edge jobs fail, and the job will not meet the minimum of number of edges criteria, the distribution job will abort immediately all its running Edge jobs.

      For example, assume you defined a distribution job to 10 Edges, and specified the **Success Definition** to minimum of 6 Edge jobs. If 5 Edge jobs fail, it is obvious that the distribution job will fail, as it will not meet its success definition. Instead of waiting for the other 5 Edge jobs to complete (successfully or not), the distribution job can abort immediately these 5 Edge jobs and determine its completion status- as failed.

      If the success definition requires that all Edge jobs complete successfully, one failed job is sufficient to determine the completion status of the job and the job will abort immediately.

- **Synchronization Points** – The Center and Edge(s) components may synchronize at certain points of the job execution. All processes will hold, until all Edges reach the specified breakpoint before continuing. More than one breakpoint may be selected:

    - **Before Edges Pre-Commands**
    - **After Edges Pre-Commands**
    - **Before Edges Post-Commands**

- **Distribution Recovery Options** – Distribution operations can recover after unexpected errors such as communication line or network faults, user interruption, or even a system crash. In anticipation of these errors, the Center maintains an ongoing record of its progress through each distribution job.

  In the event of failure, R-1 obtains all necessary information regarding the failed operation from the record generated, including the original property configurations and the exact point of failure.

  The **Recovery** parameters allow complete configuration of R-1's recovery mechanism. Parameters that may be manipulated include:

  - **Max Retries** - The maximum number of recovery attempts.
  - **Base Interval** - The time interval before the next attempt is made.
  - **Retry Factor** - An incremental increasing factor used to geometrically increase the interval length between attempts.
  - **Max Interval** - The maximum period of time to which the interval between recovery attempts may accumulate based on the **Base Interval** and the **Retry Factor**.

Example:

A job has a Max Retries of 6, a Base Interval of 10 minutes (600 seconds), a Retry Factor of 50%, and a Max Interval of 60 minutes (3600 seconds).

The job was submitted at 5:05 AM:
Recovery attempt 1 of 6 will occur at 5:15 (10 minutes after the failure)
Recovery attempt 2 of 6 will occur at 5:30 (15 minutes after the last failure)
Recovery attempt 3 of 6 will occur at 5:53 (23 minutes after the last failure)
Recovery attempt 4 of 6 will occur at 6:27 (34 minutes after the last failure)
Recovery attempt 5 of 6 will occur at 7:18 (51 minutes after the last failure)
~~Recovery attempt 6 of 6 will occur at 8:34 (76 minutes after the last failure)~~
Recovery attempt 6 of 6 will not occur since the interval has increased to 76 minutes surpassing the Max Interval of 60 minutes.

- **Cross Platform Indication -** By default R-1 queries all Edges for their Operating System to decide what kind of Snapshot it should create. This may cause the job to hold, as the Snapshot phase will not be completed without information from all Edges. Using this qualifier, the user can specify what kind of snapshot to create, so the snapshot phase will proceed immediately.

  - **Windows only** or **UNIX only**: Use when all Edges are of the same Operating System as the Center. This will result in an enhanced snapshot using all file system qualifiers (including permissions).

  - **Cross Platform**: R-1 will create a snapshot containing only information that is common to both Operating Systems. Permissions will not be transferred.
  - **Auto Detect**: Default. R-1 will query all Edges for their Operating System and will build the Snapshot accordingly.

- **Abort on empty source**– If the source directory of the distribution is empty, **ALL** files on the target run the risk of being deleted. If this option is selected the job will abort if the source list is empty.

- **Use Pre-Distribution Download** – Define Download jobs to be submitted before the distribution process begins. Only after successful completion of the Download processes, will content be distributed to the Edge(s).

3.2.2. Pre-Distribution Download Window

The **Pre-Distribution Download** Window allows adding, removing, renaming and updating the Download job definitions.

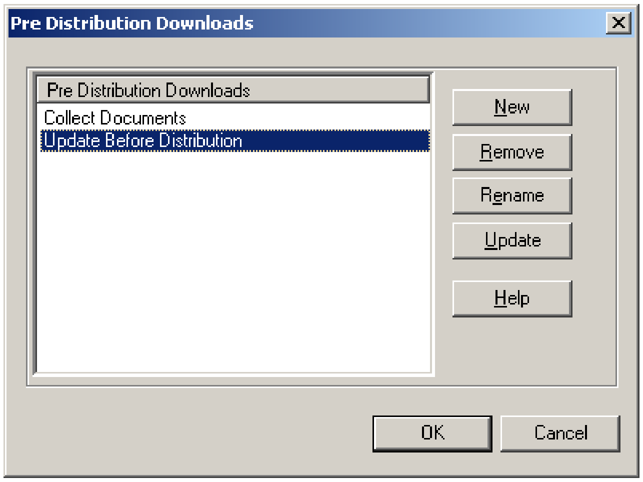

Pre Distribution Downloads
Pre Distribution Downloads
Collect Documents
Update Before Distribution
New
Remove
Rename
Update
Help
OK
Cancel

Figure 5: Defining Pre-Distribution Download Jobs

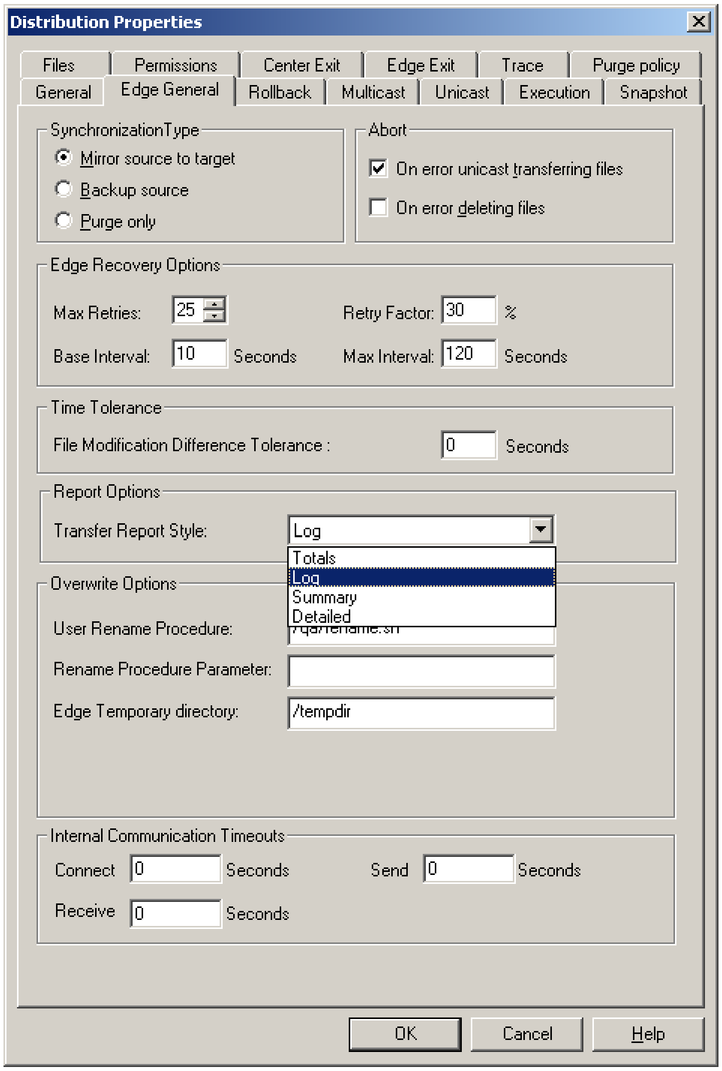
The Pre-Distribution Download jobs will be submitted before the distribution process begins. Only after successful completion of the Download processes, will content be distributed to the Edge(s).

Pressing the **New** button will open a Download Job Definition window.

NOTE: The submission order of the Pre-Distribution download jobs is not affected by the order of the jobs in the list.

3.2.3. Edge General Tab

The Edge properties set in this tab, are the default Edge settings. To override these settings for a specific Edge, open the **Edge Override** dialog.



Distribution Properties

SynchronizationType
- ● Mirror source to target
- ○ Backup source
- ○ Purge only

Abort
- ☑ On error unicast transferring files
- □ On error deleting files

Edge Recovery Options

Max Retries: 25

Retry Factor: 30 %

Base Interval: 10 Seconds

Max Interval: 120 Seconds

Time Tolerance

File Modification Difference Tolerance : 0 Seconds

Report Options

Transfer Report Style: Log

Overwrite Options

User Rename Procedure: /qa/rename.sh

Rename Procedure Parameter:

Edge Temporary directory: /tempdir

Internal Communication Timeouts

Connect 0 Seconds

Send 0 Seconds

Receive 0 Seconds

Figure 6: Edge General Tab - Defining Common Edge Properties

- **Synchronization Type** – Specify Edge synchronization logic:
  - **Mirror Source To Target –** At the end of the synchronization process, the Source and Target directories are exactly the same.
    - Files that exist on source but do not exist on target are copied to target.
    - Files that exist on both source and target – with any differing properties (size / modification date) - are copied from the source to the target.
    - Files that do not exist on source but exist on target are deleted from target.

If the **Purge Files Older Than** is checked, only files older than specified are deleted.

- Files existing on both source and target having differing NTFS permissions only are not transferred, however the NTFS permissions may be replicated by putting a check mark in the preserve/detect NTFS checkbox.
- Files that exist on both source and target that are identical will **NOT** be copied.

NOTE: Using **Mirror** logic, an additional option can be enabled to only delete files on the target that are older than a preset parameter. To activate this option, place a check mark in the **File Tab/Purge Files Older Than** checkbox and complete the required fields. This means “Do not purge files that were created / modified within the past **Days / Hours / Minutes**.”

- **Backup Source** - Files from source are copied to target but no deletions are made on target machine.
  - Files that exist on source but do not exist on target are copied to target.
  - Files that exist on both source and target are copied from the source to the target only if they have a more recent creation / modification date.
  - Files that do not exist on source but exist on target are left unchanged.
  - **Important for Windows Target:** Mismatch - a file (directory) on the source has the same name and path as a directory (file) on the target - results in deletion of the file (directory) on the target and the creation of the directory (file) from the source to the target.

Example:

During a previous backup operation, file “Document” existed on the source but not the target, and was subsequently copied to the target. In the interim, the file “Document” was removed from the source. Since the job is being run under **Backup Source** logic, the file “Document” is not removed from the target. Now, however, a directory has been created on the source called “Document”. Since Windows machines do not allow duplicate file / directory names, the directory “Document” cannot be copied to the target since the file “Document” still exists there.

R-1 will delete the conflicting file / directory on the target, copying the newer file / directory from the source.

  - **Purge Only** - Files and directories that exist on target but do not exist on source are deleted from the target. This logic involves no transfer of files from the source to the target.
    - Using Purge logic, an additional option can be enabled to only delete files on the target that are older than a preset parameter. To activate this option, place a check mark in the **Purge Files Older Than** checkbox and complete the required fields. This means "Do not purge files that were created / modified within the past **Days / Hours / Minutes**."
    - Using a **Continuous Updates** job, if a directory is renamed, it is purged (deleted) from the target directory.

- **Abort** - Specify the conditions in which a job will abort. More than one option may be specified.
  - **On Error Unicast Transferring Files** – Abort the job if failure occurred while transferring or deleting any of the files to/from the Edge(s) using unicast.
  - **On Error Deleting Files** – Abort the job if a failure occurred during the deletion of target files during the synchronization process.

- **Edge Recovery Options** –Edge operations can also be configured individually to recover after unexpected errors, so that each edge can have its own independent recovery policy. In the event of failure, R-1 obtains all necessary information regarding the failed operation from records generated, including the exact point of failure. The values set for the Edge recovery may differ from those set for the complete job.

  Together, the **Recovery** parameters allow complete configuration of R-1's recovery mechanism. Parameters that may be manipulated include:
  - **Max Retries** - The maximum number of recovery attempts.
  - **Base Interval** - The time interval before the next attempt is made.
  - **Retry Factor** - An incremental increasing factor used to geometrically increase the interval length between attempts.
  - **Max Interval** - The maximum period of time to which the interval between recovery attempts may accumulate based on the **Base Interval** and the **Retry Factor.**

- **File Modification Difference Tolerance -** The replication process may be fine tuned by adjusting the age threshold used by R-1 to decide that a target file is too old thus replacing it with a current version. This parameter sets a threshold used by R-1 to compare the modification date/time of similar files on the source and target.

  One application for the file time tolerance parameter would be to allow for the difference in time stamping when replicating between FAT and NTFS drives. In this case, a value of two (2) seconds would avoid the possibility of unnecessary file transfers.

A Word About Time Stamps on FAT and NTFS Drives

The following is an excerpt from Microsoft's web based documentation on the subject:

File time stamps on FAT drives are rounded to the nearest two seconds (even number) when the file is written to the drive. The file time stamps on NTFS drives are rounded to the nearest 100 nanoseconds when the file is written to the drive.
When files are copied from NTFS drives to FAT drives, some file time stamp rounding has to occur; the file time stamp is rounded up to the next even second.
To preserve exact NTFS file times, always copy files from NTFS drives to other NTFS drives. If writing a program to compare file times between NTFS and FAT drives, accommodate for the expected rounding.
For example, when comparing time stamps between an original file in NTFS and a file in FAT, which is copied from NTFS, the following appears:
NTFS time stamp: 7 hours 31 min 0 sec 000.
FAT time stamp becomes 7 hours 31 min 0 sec 000.
NTFS time stamp: 7 hours 31 min 0 sec 001.
FAT time stamp becomes 7 hours 31 min 2 sec 000.
NTFS time stamp: 7 hours 31 min 1 sec 000.
FAT time stamp becomes 7 hours 31 min 2 sec 000.
NTFS time stamp: 7 hours 31 min 1 sec 999.
FAT time stamp becomes 7 hours 31 min 2 sec 000.

- **Report Options - Transfer Report Style –** Available report types are:
  - **Totals** – Displays the least amount of job detail. Information includes job start/end time and number of files transferred.
  - **Log** - This style includes all **Totals** information, as well as source/target (including path) filenames and the file sizes.
  - **Summary** – All information from **Log** and **Total** styles is included, as well as the transfer, start and end times of any file transferred.
  - **Detailed** - All of the above information as well as block level transfer details per file transferred.

    This report file may be very large. If the job includes the transfer of many files, or the Center has a limited amount of free space available, it is recommended that **Log** or **Total** styles be used.

    NOTE: Using the **File List** option in the **On Exit Tab**, specify here either **Summary** or **Detailed**. Any other report style will result in an **empty** file list being created.

  For more information about the available reports refer to **Appendix M – Reports**.

- **Overwrite Options**

NOTE: **Overwrite After Completion of Transfer** feature is activated automatically by R-1 using Multicast or the LAN transfer engine. Using WAN engine, this option needs to be explicitly checked in the **Unicast Tab** under the WAN transfer engine properties.

- **User Rename Procedure – Rename Hook -** Enter the name of the rename hook script. The purpose of the Rename Hook is to replace R-1's regular rename process, that happens after a file is being copied to the temporary directory in the Edge, and moves the file from this temporary directory to the target directory.

  Instead of using the default permissions preserving, users can now add the "inherit target permissions" and develop a rename process that moves the file and sets its permissions according to the real target parent directory or any other mechanism.

  The procedure has to reside on the target machine.

  It accepts the following environment variables -

  ```
  FASTCOPY_TEMP_FILE_NAME
  FASTCOPY_TARGET_FILE_NAME
  FASTCOPY_RENAME_HOOK_PARAM
  ```

  The script may look like that:

  ```
  copy "%FASTCOPY_TEMP_FILE_NAME%" "%FASTCOPY_TARGET_FILE_NAME%"
  REM Use the following line for WAN transfer only
  DEL "%FASTCOPY_TEMP_FILE_NAME%"
  exit 0
  ```

  In case of success the rename procedure should return 0.

  NOTE: Temporary files are always deleted when using the LAN transfer. When using WAN transport engine, add the DEL "%FASTCOPY_TEMP_FILE_NAME%" command to your script.

  NOTE: Wrap the environment variables in "" for files with spaces in their name.

- **Rename Procedure Parameters-** Enter the parameters to be passed to the rename hook script.

- **Edge Temporary Directory – Overwrite After Completion–** Specify the temporary directory on the Edge. If this is not set, R-1 will use its installation directory.

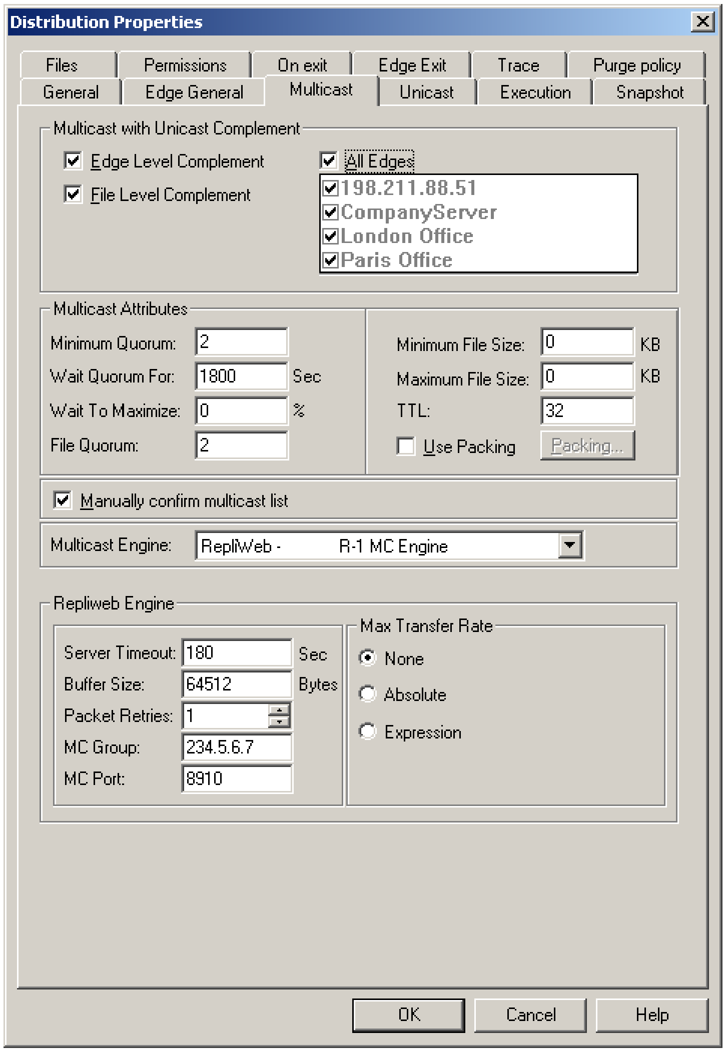
- **Preserve UNIX Files Permissions to Temporary Files** – Preserve UNIX permissions on the temporary files.
- **Preserve UNIX Files Ownership to Temporary Files** – Preserve UNIX ownership on the temporary files.

NOTE: These two UNIX options deal with the temporary file only. To preserve permissions of the target files, use the options in the **Permissions Tab**.

- **Internal Communication Timeouts -** R-1 enables controlling TCP/IP communication timeouts, by using Application Timeouts. Setting these fields will override the operating system TCP timeouts. Value 0 means using the operating system timeouts. Specify here overall R-1 communication timeouts. Timeouts can also be set for each of the transport engine separately for the transfer stage.
    - **Connect -** Specify, in seconds, the TCP connect session timeout for general communication within R-1 components. Default is 60 seconds.
    - **Send -** Specify, in seconds, the TCP send operation timeout, for general communication within R-1 components. Default is 60 seconds.
    - **Receive -** Specify, in seconds, the TCP receive operation timeout, for general communication within R-1 components. Default is 60 seconds.

3.2.4. Multicast Tab

This tab enables setting of Multicast properties. For a complete explanation regarding Multicast Transfer see **Distribution Features – Multicast** in the **Overview** chapter.



Multicast with Unicast Complement

☑ Edge Level Complement

☑ File Level Complement

☑ All Edges

☑ 198.211.88.51
☑ CompanyServer
☑ London Office
☑ Paris Office

Multicast Attributes

Minimum Quorum: 2
Wait Quorum For: 1800 Sec
Wait To Maximize: 0 %
File Quorum: 2

Minimum File Size: 0 KB
Maximum File Size: 0 KB
TTL: 32
☐ Use Packing Packing...

☑ Manually confirm multicast list

Multicast Engine: RepliWeb - R-1 MC Engine

Repliweb Engine

Server Timeout: 180 Sec
Buffer Size: 64512 Bytes
Packet Retries: 1
MC Group: 234.5.6.7
MC Port: 8910

Max Transfer Rate
● None
○ Absolute
○ Expression

Figure 7: Defining Multicast Properties

- **Edge Level Complement** – Multicast to the Edges where possible. If Multicast is not enabled by the Edge or the Edge did not meet the Multicast criteria, use Unicast.

- **File Level Complement** – Multicast files to the Edges where possible. If the file did not meet the File Quorum, its Edge cannot use Multicast or its transfer has failed, use Unicast.

- **Edges List** – Select which of the defined Edges is a Multicast enabled. Check **All Edges** if content should be sent using Multicast to all Edges.

- **Manually confirm multicast list** – After calculating the list to be sent in Multicast transmission, the job will hold for manual preview by the user. Selecting this option will pause the distribution job after the planning stage.

  Select the job (identified by a unique preview icon in the **State** column) by clicking the right-mouse button, and choose the **Preview** option from the menu. A list of all of the transfers and deletions required will be displayed. Any of the changes indicated may be cancelled. To cancel a change, right-mouse-click on the file and select **Ignore**.
  - To confirm a preview, click **Confirm**.
  - To reject a preview, click **Cancel**, and proceed to abort the job.

  NOTE: When **Manually confirm multicast list** is selected, the job will not proceed past the preview stage without user input. This should only be used in running tests and not in a scheduled production scenario. If this option is selected, the job will not run until the snapshot is manually confirmed.

  This option is NOT available for Continuous Updates jobs.

- **Multicast Attributes** – Define setting for the Multicast transfer:
  - **Minimum Quorum** – Set the minimum number of Edges required to start a Multicast distribution. If the number of Multicast enabled Edges is lower than the set number, and Edge Level Complement is selected, Unicast will be used.
  - **Wait Quorum For** – Set the number of seconds to wait for the Multicast Quorum to be reached.
  - **Wait to Maximize** – Once Multicast Quorum is reached, defines the time that the distribution waits for additional Edges to join the Multicast group. This is in percent of the amount of time it took the quorum to assemble.
  - **File Quorum** – The minimum number of Edges a given file has to be sent to qualify for Multicast transmission. Files that should be sent to fewer Edges than specified are not sent with Multicast but with Unicast, if the File Level Complement option is selected.
  - **Minimum File Size** – The minimum size of a file (in KB) for Multicast transmission. Smaller files are not sent with Multicast but with Unicast
  - **Maximum File Size** – The maximum size of a file (in KB) for Multicast transmission. Larger files are not sent with Multicast but with Unicast.
  - **TTL** – Specifies the number of hops a buffer can do before Multicast transmission fails. Default is 32.

- **Multicast Engine** – Select from the list the Multicast engine to be used. This will enable setting specific engine properties.
  - **RepliWeb - R-1 MC Engine**
    - **Server Timeout** – The maximum number of seconds the Edge will wait to receive data from the Center in Multicast transmission
    - **Buffer Size** – Specifies the buffer size in Bytes for Multicast transmission.
    - **Packet Retries** – Specifies the number of retries for each packet.
    - **Max Transfer Rate -** Determine the maximum Multicast transfer rate.
      Values can be **None**, **Absolute** (MB|Mb|KB|Kb / second) or **Expression**.

      See **Appendix B** for setting Time Frames expressions.

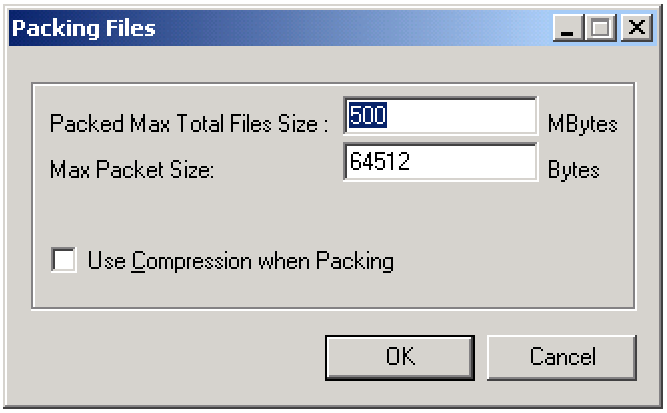
NOTE: Overwrite After Completion of Transfer feature is activated automatically by R-1 using Multicast or the LAN transfer engine. To override the Rename procedure or the default temporary directory, fill in the **Overwrite Options** in the **Edge General** Tab.

- **Use Packing** – Select this option to pack transferred files into packages before transmission. Pressing the Packing button will open the **Packing Files Window**.

3.2.5. Multicast – File Packing Window

This window enables setting the properties of the File Packing option.

When the source directory contains many files, it is recommended to pack them into larger packages for better performance, as the overhead of multicasting a file take longer than actually transmitting that file. Multicasting just a few large files it is recommended not to use file packing.



Packing Files

Packed Max Total Files Size : 500 MBytes

Max Packet Size: 64512 Bytes

☐ Use Compression when Packing

Figure 8: Set Packing Properties

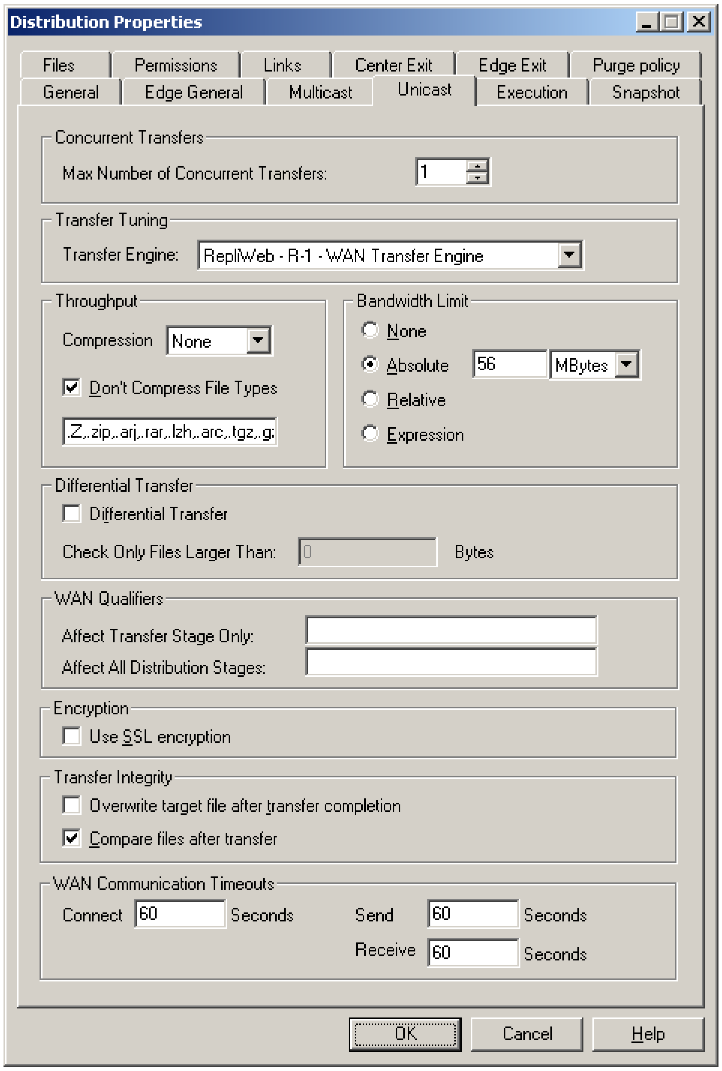
- **Packed Max Total Files Size** – The maximum total size of files to be packed in one package, in MB.

- **Max Packet Size**– Each package is splitted into buffers. Specify here the maximum size of each buffer, in Bytes. The minimum value is 64512 bytes.

- **Use Compression -** Select this option to compress the files while packing. This will allow more files to be sent in one package.

3.2.6. Unicast Tab

Sets Unicast properties. For a complete explanation regarding Unicast Transfer see the **Overview** chapter.



Concurrent Transfers

Max Number of Concurrent Transfers: 1

Transfer Tuning

Transfer Engine: RepliWeb - R-1 - WAN Transfer Engine

Throughput

Compression None

☑ Don't Compress File Types

.Z,.zip,.arj,.rar,.lzh,.arc,.tgz,.g:

Bandwidth Limit

○ None

● Absolute 56 MBytes

○ Relative

○ Expression

Differential Transfer

☐ Differential Transfer

Check Only Files Larger Than: 0 Bytes

WAN Qualifiers

Affect Transfer Stage Only:

Affect All Distribution Stages:

Encryption

☐ Use SSL encryption

Transfer Integrity

☐ Overwrite target file after transfer completion

☑ Compare files after transfer

WAN Communication Timeouts

Connect 60 Seconds Send 60 Seconds

Receive 60 Seconds

Figure 9: Setting Unicast Distribution Properties

- **Concurrent Transfers** – Set the maximum number of concurrent transfers.

  Transfer performance can be boosted when performing simultaneous parallel transfers of partial file lists.

- **Transfer Tuning –** Select the Unicast (upload/download) transport engine.

  - **R-1 LAN Transfer Engine -** Bandwidth limit selection, compression, and the differential transfer options are all voided. If these options are not needed, use this engine to significantly reduce transfer times.

Relative performance increases will be especially large if the content consists of a large number of small files.

- **R-1 WAN Transfer Engine -** The Wide Area Network transport options can significantly reduce transfer time in WAN environments by employing advanced features such as asynchronous compression.

WAN Transfer Engine Properties

- **Throughput –** Select the compression type to be used and specify file types that should not be compressed.

  The volume of data being transferred and the network throughput rate are both significant factors in determining the overall time required to transfer data. The relative change in transfer time if compression is employed will depend on many factors, including the CPU speed and network throughput rate.

  Compression is performed transparently, without performing any intermediate spooling of compressed data.

  Two factors determine the total file transfer time:
  - **Block Compression Time** - This is influenced by the processing resources of the machine and the nature of the data being compressed.
  - **Compressed Block Transmission Time** - This is influenced by the network throughput rate.

  If compression of the subsequent block to be transmitted is completed before completing the transmission of the current block, transmission of the next block will not be delayed. Alternatively, if the block compression takes more time than block transmission, the transfer will be held up waiting for the compressed block to be ready for transport.

  Thus, if the transfer is being performed over a high-speed network and the server is heavily stressed, implementing compression can actually increase the total transfer time.

  Several compression options are available via the pull-down box:

  - **None** - No compression is used on the data to be transmitted.
  - **Normal** - A limited compression algorithm is utilized to compress the data.
  - **AP** - A moderately efficient compression algorithm, providing a higher degree of compression than the **Normal** algorithm, is utilized to compress the data.

  - **Power** - The strongest compression algorithm utilized, produces better compression results on executable files than the **Normal** and **AP** algorithms, but consumes considerable resource usage.

  Each compression algorithm employs different amounts of CPU resources. For most needs, AP is the optimal compression algorithm.

  - **Don't Compress File Types** – Files of the specified types will not be compressed.

    Files that have already been compressed will yield no net compression after undergoing R-1 compression. Trying to compress them will result in undue CPU time.

  See **Appendix A** for a complete explanation on selecting a compression algorithm and file types.

- **Bandwidth Limit** – Set the desired bandwidth option:
  - **None** – No bandwidth limit is set.
  - **Absolute** – Set absolute bandwidth limit in **MB|Mb|KB|Kb.**
  - **Relative** – Set relative bandwidth limit in **%**
  - **Expression** – Set bandwidth using an expression..

  See **Appendix B** for a complete explanation on setting Bandwidth limits.

- **Differential Transfer** – R-1 will only transfer the changes in a given file during transfer, down to the block level.

  - **Check only files larger than** - Applying differential transfer to small files can actually increase the total replication time. File size is in Bytes.

    There is no advantage at all in using the Differential Transfer option with files smaller than 64K.

- **Additional WAN Qualifiers** – Add here any qualifiers to affect the complete Distribution or the Transfer stage only, in a Wide Area Network. Normally only used under supervision of RepliWeb Professional Services Engineers.

- **Use SSL Encryption** – Check this option if connection to the Edge should be encrypted using **SSL.**

  For explanation about R-1 security, see the **Overview Chapter**.

- **Transfer Integrity**
    - **Overwrite Target File After Completion Of Transfer -** If a user attempts to access a file that is in the process of being transferred, or if the distribution fails in the middle of distributing a file, only part of the file will exist on the target therefore users may access corrupted data.

      Using this option, the file will be transferred to a temporary location on the target, leaving the original file on the target intact. Once the entire file has been transferred to the target, the original file is overwritten, and the temporary file removed. This will ensure that users will only access a complete file.

      Using this option, fill in the **Overwrite Options** in the **Edge General Tab**.

      This option is automatically available when using The LAN Transfer or the Multicast engines.

    - **Compare Files After Transfer** - R-1 will analyze each distributed target file to ensure that it matches the source. This prevents problems in data integrity caused by external processes, such as another user or application altering either source or target data during distribution. When comparing files on the Center and Edge nodes, a CRC (Cyclic Redundancy Check) calculation on each file is performed and the results are compared. Only the results of the calculation are transmitted for purposes of comparison, so the comparison itself is immediate. Enabling this option can significantly decrease performance.

      NOTE: On networks with fast lines and few or no communication problems, such as LANs, the use of these verification measures may slow down transfer operations. In this case, it might be best to dispense with some or all of these steps. On the other hand, they are essential for foolproof distribution on networks with frequent communication difficulties

- **WAN Communication Timeouts -** R-1 enables controlling TCP/IP communication timeouts, by using Application Timeouts. Setting these fields will override the operating system TCP timeouts. Value 0 means using the operating system timeouts. Specify here timeouts to be used by the WAN transport engine at the transfer stage.
    - **Connect -** Specify, in seconds, the TCP connect session timeout to be used during transfer using the WAN engine. Default is 60 seconds.

  - **Send -** Specify, in seconds, the TCP send operation timeout to be used during transfer using the WAN engine. Default is 60 seconds.

  - **Receive -** Specify, in seconds, the TCP receive operation timeout to be used during transfer using the WAN engine. Default is 60 seconds.

LAN Transfer Engine Properties

- **LAN Communication Timeouts -** R-1 enables controlling TCP/IP communication timeouts, by using Application Timeouts. Setting these fields will override the operating system TCP timeouts. Value 0 means using the operating system timeouts. Specify here timeouts to be used by the LAN transport engine at the transfer stage.

  - **Connect -** Specify, in seconds, the TCP connect session timeout to be used during transfer using the WAN engine. Default is 0, using Operating system defaults.

  - **Send -** Specify, in seconds, the TCP send operation timeout to be used during transfer using the WAN engine. Default is 0, using Operating system defaults.
  - **Receive -** Specify, in seconds, the TCP receive operation timeout to be used during transfer using the WAN engine. Default is 0, using Operating system defaults.

NOTE: Override After Completion of Transfer feature is activated automatically by R-1 using Multicast or the LAN transfer engine. To override the Rename procedure or the default temporary directory, fill in the **Overwrite Options** in the **Edge General** Tab.

3.2.7. Distribution Execution Commands Tab

Data distribution is often part of a wider operation. In that event, pre or post processing are needed and become an integral part of the distribution operation. The built in recovery mechanism considers an operation incomplete if the post-distribution processing was not completed successfully, even if the distribution itself was. These commands can be run on either the Edge or Center systems, or both.

Commands can be:

- **Center and Edge Commands**
  - **Edge Pre-Command** – The command will be executed on the Edge prior to the submission of the Job.
  - **Edge Post-Command** – The command will be executed on the Edge after a successful execution of the Job.
  - **Center Pre-Command** – The command will be executed on the Center prior to the submission of the Job.
  - **Center Post-Command** – The command will be executed on the Center after content distribution is successfully completed for a specific Edge.

  - **Pre Commands Timeout** – Check here to specify the time to wait (in seconds) until the pre command on the Center and/or the Edge will execute. If the timeout have passed, the execution of the pre-command fails.
  - **Post Commands Timeout** – Check here to specify the time to wait (in seconds) until the post command on the Center and/or the Edge will execute. If the timeout have passed, the execution of the pre-command fails.

- **Distribution Commands**
  - **Distribution Pre-Command** – The command will be executed on the Center prior to the content distribution.
  - **Distribution Post-Command** – The command will be executed on the Center after content distribution is successfully completed to all edges involved in this Job.

Valid **Command** entries are the full path to any single batch file or script that can be run from a command-line. Valid **Parameters** entries are any switches, qualifiers, etc., that the script or batch file accepts. Multiple parameters must be separated by a single space.

```
<command> <variable1> <variable2>
```

Where `<variable1>` and `<variable2>` from the batch file refer to user-entered values in the **Parameters** field(s).

Example:

If the parameters to be passed to the batch file are the path c:\dev\www and the value 350, the **Parameters** field should appear as follows:

```
c:\dev\www 350
```

Multiple parameters are separated by a space.

Note: The full paths to the commands are always required, as the batch files are run without any environment context, such as system search path, etc. Paths to batch files should always be entered in the native format of the system the command is being run on.

Example:

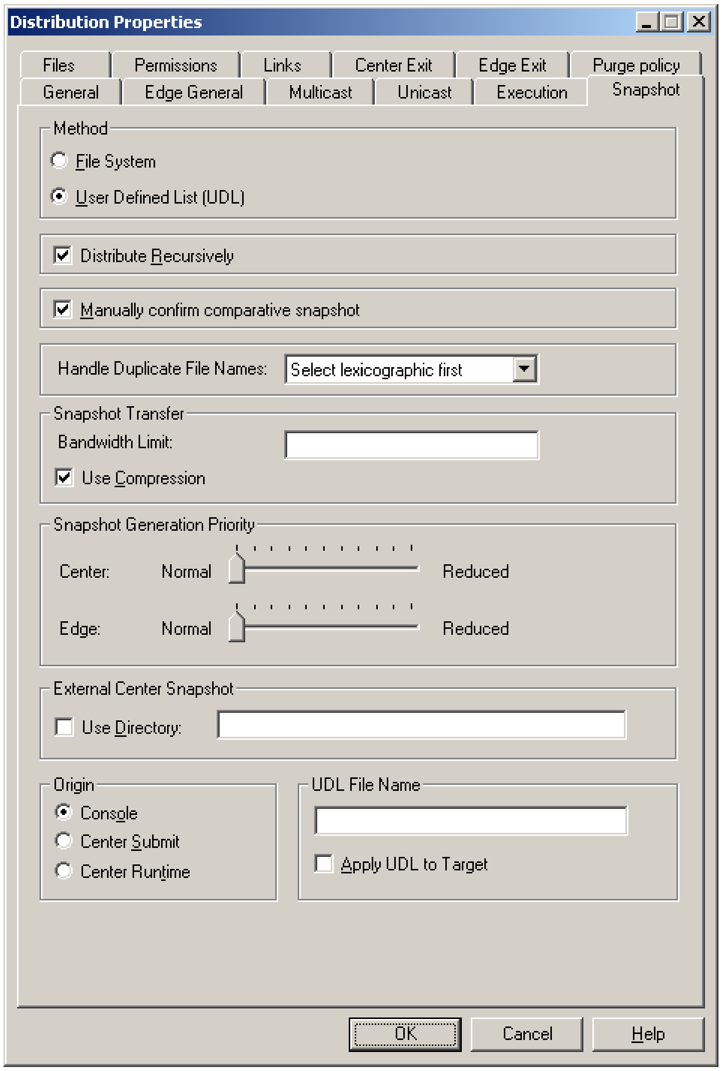
If an Edge post command is being run, and the Edge is a UNIX machine, it should be written as:

```
/scripts/restart.sh
```

Standard output from any command run via **Execution** is saved in R-1 reports. The report name corresponds to the command type. For example, standard output from an **Edge Post Command** would appear in a report titled “Post-Commands: Edge.”

3.2.8. Snapshot Properties Tab

This tab allows defining the fine points of how the Snapshot is created. In the planning phase, Snapshots of the source and target directories are taken, listing file sizes, modification times and dates, security permissions, and shares (where applicable). The remote snapshot is then transferred to the Center where it is analyzed locally looking for differences between the source directory tree and target directory tree. The analysis results in a Comparative Snapshot Matrix™ (CSM™) that lists the absolute minimum that must be transferred to perform the requested synchronization.



Method
○ File System
● User Defined List (UDL)

☑ Distribute Recursively

☑ Manually confirm comparative snapshot

Handle Duplicate File Names: Select lexicographic first

Snapshot Transfer
Bandwidth Limit:
☑ Use Compression

Snapshot Generation Priority
Center: Normal Reduced
Edge: Normal Reduced

External Center Snapshot
□ Use Directory:

Origin
● Console
○ Center Submit
○ Center Runtime

UDL File Name
□ Apply UDL to Target

Figure 10: Defining Snapshot Properties

- **Method** – Specify how the snapshot will be created:
    - **File System -** The files to be synchronized will be obtained directly from the directory specified in the **Source Directory** in the Job Definition window, being filtered by the use of file and exclude specifications as applicable.

  - **User defined list (UDL) -** The file list that will be synchronized will be taken from a text file supplied by the user. The file can either be created directly by the user or generated by an outside process. For a full explanation regarding the format and behavior of the User Defined List option, see **Appendix E**.

- **Distribute Recursively** – Specifies whether to distribute the current directory only or all directories beneath it.

- **Manually Confirm Comparative Snapshot** – This function offers an additional method to assist in creating a perfectly customized distribution job. Selecting this option will pause the distribution job after the planning stage.

  Select the job (identified by a unique preview icon in the **State** column) by clicking the right-mouse button, and choose the **Preview** option from the menu. A list of all of the transfers and deletions required will be displayed. Any of the changes indicated may be cancelled. To cancel a change, right-mouse-click on the file and select **Ignore**.

  - To confirm a preview, click **Confirm**.
  - To reject a preview, click **Cancel**, and proceed to abort the job.

NOTE: When **Manually confirm Comparative Snapshot** is selected, the job will not proceed past the preview stage without user input. This should only be used in running tests and not in a scheduled production scenario. If this option is selected, the job will not run until the snapshot is manually confirmed.

This option is NOT available for Continuous Updates jobs.

- **Handling Duplicate File Names** –When R-1 encounters a situation where only one of either the source or target systems is case sensitive (i.e. UNIX), the user must decide how the discrepancies will be handled.

Example:

If the source contains files: File1, file1, and FILE1 and the target are a Windows machine, the target can only have a single “file1”, regardless of case.

Select the handling method from the list:

  - **Error Exit -** Exit the job with an error.
  - **Select Newest** - Transfer the file with the newest modification date.
  - **Select Oldest** - Transfer the file with the oldest modification date.

- **Select Lexicographic first** - Transfer the file that is first alphanumerically.
- **Select Lexicographic last** - Transfer the file that is last alphanumerically.

NOTE: Lexicographic order interprets capital letters before lower-case letters.

- **Snapshot Transfer** – The snapshot is being transferred from the Edges to the Center. In case of very large snapshot files (large distribution trees), the snapshot transfer can be optimized.

  - **Bandwidth Limit** - Set bandwidth using an expression. See **Appendix B** for a complete explanation on setting Bandwidth limits.

  - **Use Compression** – Compress the snapshot file prior to its transfer to the Center.

- **Snapshot Generation Priority** – The snapshot generation stage is a CPU intensive process. On especially large directory trees with a high number of files (hundreds of thousands → millions) this process can potentially leave few CPU resources for other active processes.

  Use the slider bar to reduce the CPU utilization of a given snapshot generation stage. These can be set individually for both Center and Edge. This will lengthen the time required for the snapshot generation, but leave system resources available for other processes.

- **External Center Snapshot** – Enter the **Directory** in which the external snapshot files reside.

  In cases where the CST™ and CSM™ processes may take a long time, the user can create the local (Center) snapshot lists. Use this option to specify to R-1 not to perform the Center's snapshot but rather use a previously built one. All files that must reside in the given directory.

  See **Appendix G** for a complete explanation on this feature.

The following fields are enabled only if the Snapshot creation method is UDL:

NOTE: The methods in which R-1 deals with any errors in the User Defined List text file can be found in **Appendix F**. Before using this option, please consult the appendix carefully. ***Failure to do so may result in incorrect functionality***.

- **UDL Origin** – If the file list is taken from a UDL, there are four choices for the origin of the User list. These values state where and when the list will be taken from. Although the absolute path of the list will be from the field **UDL File Name**, the system (Console / Center / Edge) must be selected.
  - **Console -** The list will be copied from the system the Console is being run from at the time the job is submitted to the Center. If the job is scheduled, the same list will be used for all subsequent jobs.
  - **Center Submit -** The list will be copied from the Center at the time the job is submitted. This copy will be used as the list for the job. If the job is scheduled, the same list will be used for all subsequent jobs.
  - **Center Runtime -** The list will be taken from the Center at the time the job is actually run. Since a scheduled job spawns an Edge Job to run at each scheduled time, each scheduled job will use the list as is when it is spawned.

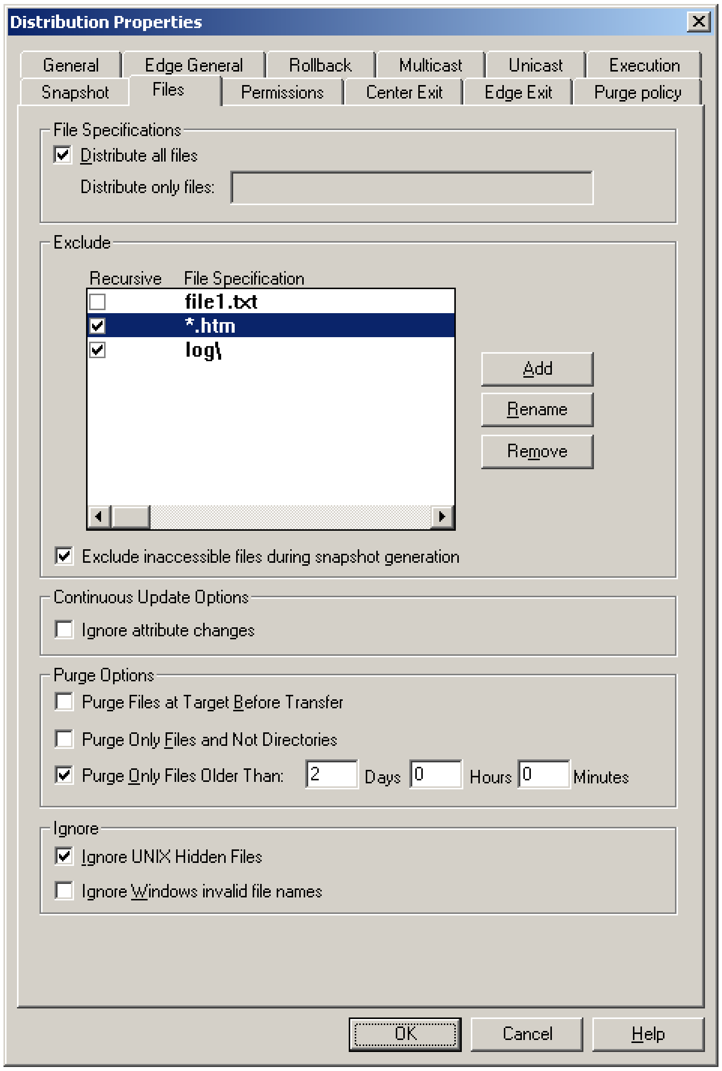
- **UDL File Name** – Enter the name of the UDL text file. A full path is required.

- **Apply UDL to target -** The use of a UDL files on the source machine results in greatly reduced system resources during the snapshot generation stage of the replication. This is because the snapshot is assumed to be the UDL itself. However, on the target machine, a standard snapshot is still issued. When this option is enabled, the UDL is transferred to the target and only the files present in this list are examined for existence and modification. This greatly reduces the associated costs of the target snapshot for large content repositories.

NOTE: For a full description of the UDL feature, please refer to **Appendix F**.

3.2.9. File Specifications Tab

This tab allows setting handling files properties. This includes selecting the files that will be distributed or excluded from distribution jobs, and also define the Purging policy.



File Specifications

☑ Distribute all files

Distribute only files:

Exclude

Recursive	File Specification
☐	file1.txt
☑	*.htm
☑	log\

☑ Exclude inaccessible files during snapshot generation

Continuous Update Options

☐ Ignore attribute changes

Purge Options

☐ Purge Files at Target Before Transfer

☐ Purge Only Files and Not Directories

☑ Purge Only Files Older Than: 2 Days 0 Hours 0 Minutes

Ignore

☑ Ignore UNIX Hidden Files

☐ Ignore Windows invalid file names

Figure 11: Distribution File Specifications

NOTE: Use the **Manually Confirm Comparative Snapshot** feature available from the **Snapshot Tab** to view and verify that the appropriate files will be excluded from the replication, and that the appropriate files will be transferred.

- **Distribute All Files -** By default, R-1 will synchronize the entire directory structure (**Recursively** from the source **Path**) and all of its contents (**All Files**).
  - **Distribute Only Files** - File specification may be used to limit the distribution to a specific file, or files of a specific type. File specification

can include wildcards.

Examples: `file1.txt,*.htm,file*.*`

Entire subdirectories may also be synchronized in the same way. Simply enter the folder name followed by a slash and asterisk.

Example: `dirname\*`
folder1\subfolder1\subfolder2\*,folder5

By default, file specification is checked recursively. Therefore, specifying an explicit file name in the list will cause the whole directory tree to be built at the target machine. Only the files meeting the file specification are transferred but the complete tree is created.

NOTE: Files that do not meet the file specifications will not be transferred, created, or deleted on the source and target systems.

- **Exclude Files** –
    - Specify the files to be excluded from the distribution. This can include a specific file name, or files of a specific type (e.g. `index.html or *.exe`). Here too, directory names may be entered to exclude a whole directory from being distributed.

      Examples: `file1.txt, .htm, file*.*`
      `dir1\, dir1\subdir1\`

    - **Recursive** – Check this option to exclude the file spec from anywhere in the source replication tree.

      In the example shown on the **Figure** above, directories named `logs` and all the files within them, and all `*.htm` files, will be excluded from the replication, regardless of their location on the source tree. The file `file1.txt` will only be excluded from the source directory level.

      For more information and samples on excluding files and directories, please refer to **Appendix K**.

NOTE: Files that meet the Exclude file specifications will not be transferred, created, or deleted on the source and target systems.

- **Exclude inaccessible files during snapshot generation** – During snapshot generation some of the files may not be accessed (mainly due to permission issues). This will cause the job to fail. Check this option to exclude these files from the replication process and transfer the rest of the replication tree.

  NOTE: During the Snapshot Generation phase, files are accessed for retrieving details only. If the user has access to retrieve details of these files, it will pass this phase successfully. However, if that user does not have access to read or delete these files, the job will fail at the Transfer stage.

  To overcome access problems at the Transfer stage, uncheck the **Abort on error transferring files** at the **Edge General Tab**.

  NOTE: Files that could not be accessed during the snapshot generation phase, will be listed in the **Exclude** report.

- **Continuous Update Options – Ignore Attribute Changes –** In a Continuous Update job, do not replicate attribute changes if that is the only change that occurred.

- **Purge Options –** Specify when and which files to purge.

  - **Purge Files at Target Before Transfer** - Check this option when transferring large files and there are also many files to delete from the target directory. This will prevent the replication from failing due to lack of disk space on the target machine.

    The default is to delete files from the target directory AFTER transferring changed files.

  - **Purge Only Files and Not Directories** – If this option is selected, the target directory tree remains intact even if files are purged.

  - **Purge Files Older Than** – R-1 will not purge files that were created or modified within the specified time. If this option is un-checked, all files that should be deleted will be purged (see Replication Logic).

- **Ignore UNIX Hidden Files** - R-1 will not transfer UNIX hidden files (i.e. files beginning with a period).

  - **Ignore Windows invalid file names** – If the source is UNIX and the target is Windows, UNIX files whose names include invalid characters in

Windows will not be transferred. If this option is not selected, and files of that kind exist, the job will fail.

3.2.10. Override Edge Properties Tabs

To specify Edge specific properties use the **Edge Override** dialog. This dialog consists of tabs similar to those in the Job Properties dialog. There, properties were defined for the whole job, including all Edges involved in the distribution process.

Use this option to define Edge specific properties in a convenient way, so only the unique Edge properties need be specified, where the default values are taken from the Job properties for all Edges.

Edges that have been modified, will be marked by a + sign in the Job Definition window.

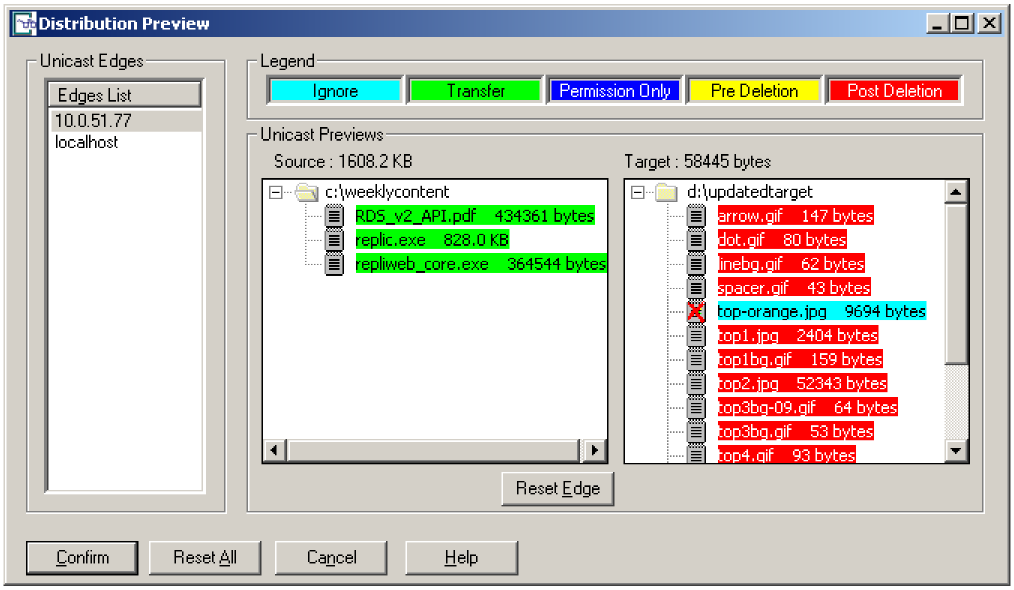
Once unique parameters were set for an Edge, changes performed for the whole job do not automatically propagate to that Edge. After making changes in the Job Properties tabs, the user will be prompted to propagate those changes to those Edges that already have override information.

Edges that do not have override information will inherit those changes automatically.

3.2.11. Distribution Preview Window

The distribution preview window enables review of the files that are about to be transferred or deleted during the distribution process, for each of the Edges in the distribution process.

This window is available whenever the user selected to **Manually Confirm Comparative Snapshot**.



Source : 1608.2 KB

```
c:\weeklycontent
    RDS_v2_API.pdf   434361 bytes
    replic.exe   828.0 KB
    repliweb_core.exe   364544 bytes
```

Target : 58445 bytes

```
d:\updatedtarget
    arrow.gif   147 bytes
    dot.gif   80 bytes
    inebg.gif   62 bytes
    spacer.gif   43 bytes
    top-orange.jpg   9694 bytes
    top1.jpg   2404 bytes
    top1bg.gif   159 bytes
    top2.jpg   52343 bytes
    top3bg-09.gif   64 bytes
    top3bg.gif   53 bytes
    top4.gif   93 bytes
```

Figure 12: Distribution Snapshot Confirmation

The Preview pane shows the source and target directory, and the files located in each of them, for each of the Edges involved. The handling of the files during distribution depends on the Synchronization Type (Mirror, Backup, Purge).

The legend shows how the file will be handled during the distribution process:

- **Ignore** – Right click on a file and press ignore to ignore the file during distribution.
- **Transfer**– The file will be transferred to the target directory.
- **Permissions** – Only the file's permission will be transferred.
- **Pre Deletion** – In some cases, files on the target need to be deleted before the transfer takes place. For example, if the source has a file named `content` and the target has a directory named `content`, the target directory needs to be deleted prior the transfer for the transfer to take place.
- **Post Deletion** – The file will be deleted from the target directory after the transfer is complete.

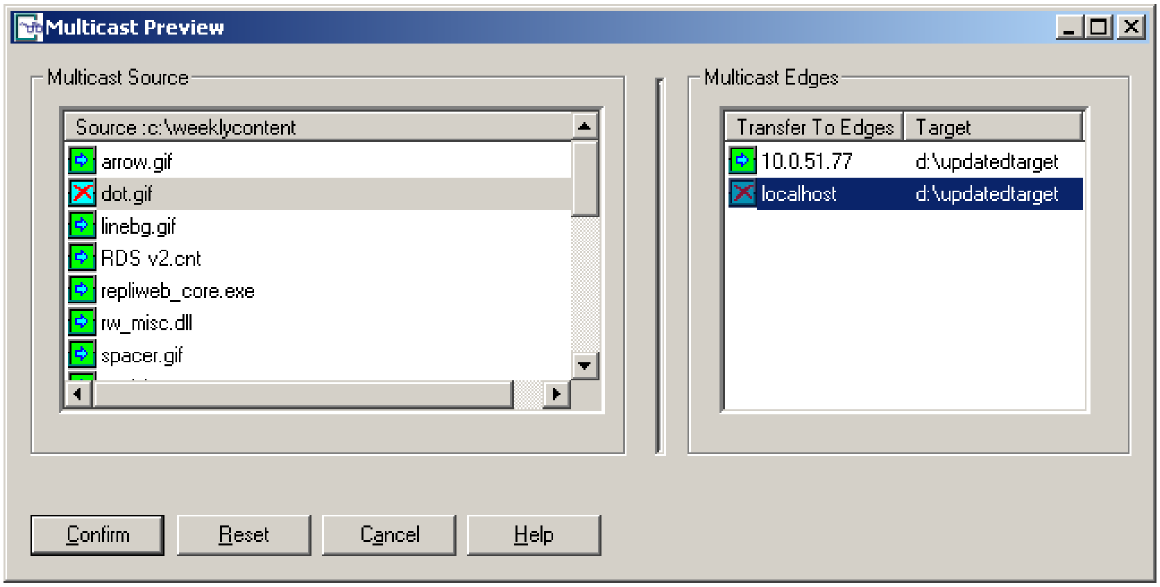
- **Reset All** – Cancel all changes made in the preview list.

- **Cancel** - Cancel all changes made in the preview list and close the Preview window. The job will remain in Preview state until Confirm is selected.
- **Confirm** – Apply changes in the list (if any) and continue the job.

3.2.12. Multicast List Preview Window

The Multicast List preview window enables review of the files that are about to be transferred using Multicast, for each of the Edges in the distribution process.

When the user select the **Manually confirm multicast list**, the job will also enter the Preview state. The following screen will be displayed, showing the files that are about to be sent using Multicast transmission.



Source :c:\weeklycontent

- arrow.gif
- dot.gif
- linebg.gif
- RDS v2.cnt
- repliweb_core.exe
- rw_misc.dll
- spacer.gif

Transfer To Edges	Target
10.0.51.77	d:\updatedtarget
localhost	d:\updatedtarget

Figure 13: Distribution Multicast List

The Preview pane shows the Multicast source directory and target edges.

Using the right mouse user can

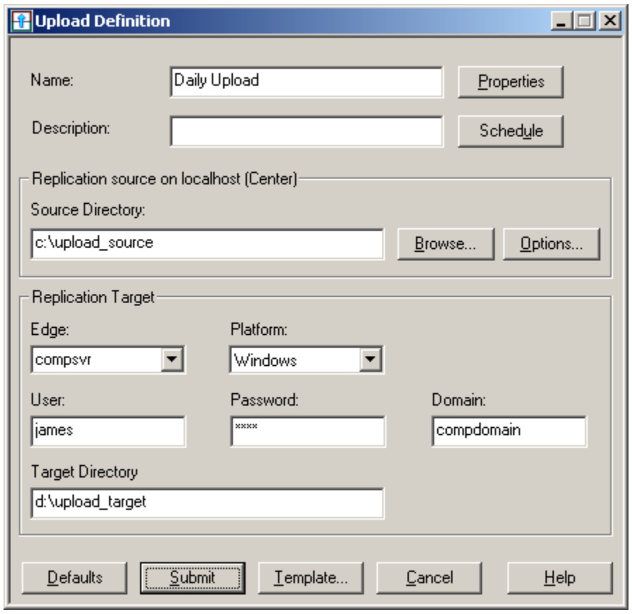
- **Ignore files** – ignored files will not be transferred at all.
- **Ignore edges** - no files will be transmitted to the ignored Edge.

- **Reset All** – Cancel all changes made in the preview list.
- **Cancel** - Cancel all changes made in the preview list and close the Preview window. The job will remain in Preview state until Confirm is selected.
- **Confirm** – Apply changes in the list (if any) and continue the job.

3.3. Replication Job Definition and Properties

A replication job is defined in the **Replication Job Definition** window. This window is reached by selecting the Upload or Download icons on the tool bar, or selecting the appropriate option from the **File** menu. Frequently performed jobs may be saved as a template. Templates are described in detail in the following sections. In effect, a template simply saves the job definition, including all options selected and information entered, for later submission or editing.

Connection to the Center automatically identifies one of the machines involved in the replication (the Center is the source for upload jobs and, conversely, the target for a download job). The above convention holds regardless of where the Console is.



Upload Definition

Name: Daily Upload

Description:

Replication source on localhost (Center)

Source Directory: c:\upload_source

Replication Target

Edge: compsvr

Platform: Windows

User: james

Password: ****

Domain: compdomain

Target Directory: d:\upload_target

Figure 14: Defining an Upload Job

NOTE: The **Download** job screen hold the same fields, the Source box holds the connection information to the Source Edge and the Target is located on the Center.

The upload job definition fields are:

- **Name** – Enter the name of the job.
- **Description** – Enter the job's description.
- **Source Directory** – The Replication's Source directory.

  The source and target **Directory** paths must be specified in full. For example, if the following path is entered as the target directory for a download job - `c:\development\www` - all download replication activity will occur in the `www` sub-directory of `development` on drive '`c`'). If this directory does not exist on the target, R-1 will create it.

NOTE: It is **EXTREMELY** important to remember that all path locations and commands **MUST** be written in a native format, i.e., if the source is a Windows machine, the source path must be in Windows format; if the target is a UNIX machine, the target path must be in UNIX format.

See **Cross Platform Formats and Escaping** description in **Chapter 5** for detailed explanation.

- **Replication Target –** Define all properties related to the Edge the job will connect:
  - **Edge**– Enter the name or address of the target Edge.
  - **Platform**– Specify the Edge's platform – Windows or UNIX.
  - **User** – Enter the username that all jobs on the Edge will use to perform the synchronization.
  - **Password** – Enter the password for the account specified in the **User** field.
  - **Domain** – If the Target platform is Windows, the target Domain will also need to be specified.
  - **Target Directory** – Specify the Target replication directory on the target Edge.

It was noted above that the user inputted in the Center settings dialog box gave the user context that all operations on the Center would be carried under. Similarly, the **User** entered in the Target box defines the user context that all operations on the Edge will be carried under.

Note that the user entered above must have whatever permissions necessary to write / read / modify / delete in the directory specified in the Edge. Any pre / post commands on the Edge will be run under that user's security context.

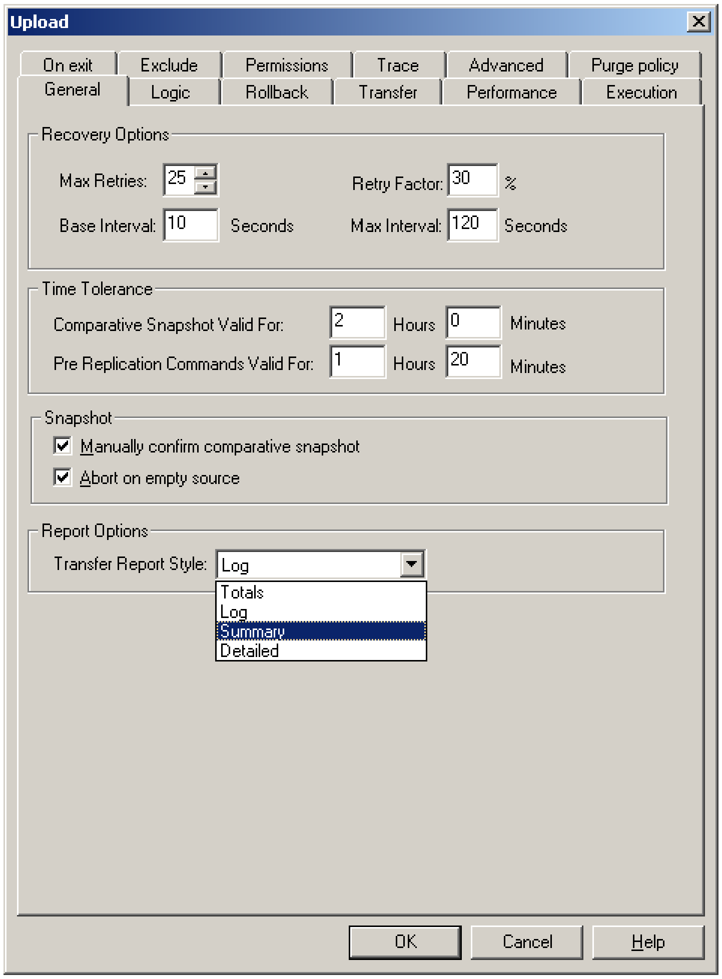
Note also that the Center's user credentials do NOT have to bear any relation to the Edge's; there is no linkage between the user context on the Center and that on the Edge.

NOTE: A rule of thumb is that if the user entered in the Center or Edge field can be actively logged-in to the systems with the Center and Edge systems respectively and create and delete files in the respective directories, R-1 can too.

- **Defaults** – Press this button to save all entered properties as defaults for following job submissions.
- **Submit** – Submits the specified distribution job.
- **Template** – Opens a dialog box allowing selecting and saving job definition templates.
- **Properties** – Opens the Properties dialog box.
- **Schedule** – Opens the Schedule dialog box.
- **Options**– Opens the Options dialog box.

Once all job properties have been defined, proceed to submit the job or, if desired, save the job as a template. Additional parameters can be set through the job **Properties** and **Schedule** screens.

3.3.1. Replication General Tab



Recovery Options

Max Retries: 25 Retry Factor: 30 %

Base Interval: 10 Seconds Max Interval: 120 Seconds

Time Tolerance

Comparative Snapshot Valid For: 2 Hours 0 Minutes

Pre Replication Commands Valid For: 1 Hours 20 Minutes

Snapshot

☑ Manually confirm comparative snapshot

☑ Abort on empty source

Report Options

Transfer Report Style: Log

Totals
Log
Summary
Detailed

Figure 15: General Replication Properties Tab

- **Recovery Options** – Replication operations can recover after unexpected errors such as communication line or network faults, user interruption, or even a system crash. In anticipation of these errors, the Center maintains an ongoing record of its progress through each distribution job.

  In the event of failure, R-1 obtains all necessary information regarding the failed operation from the record generated, including the original property configurations and the exact point of failure.

  The **Recovery** parameters allow complete configuration of R-1's recovery mechanism. Parameters that may be manipulated include:

  - **Max Retries** - The maximum number of recovery attempts.
  - **Base Interval** - The time interval before the next attempt is made.

- o **Retry Factor** - An incremental increasing factor used to geometrically increase the interval length between attempts.
- o **Max Interval** - The maximum period of time to which the interval between recovery attempts may accumulate based on the **Base Interval** and the **Retry Factor**.

Example:

A job has a Max Retries of 6, a Base Interval of 10 minutes (600 seconds), a Retry Factor of 50%, and a Max Interval of 60 minutes (3600 seconds).

The job was submitted at 5:05 AM:

Recovery attempt 1 of 6 will occur at 5:15 (10 minutes after the failure)
Recovery attempt 2 of 6 will occur at 5:30 (15 minutes after the last failure)
Recovery attempt 3 of 6 will occur at 5:53 (23 minutes after the last failure)
Recovery attempt 4 of 6 will occur at 6:27 (34 minutes after the last failure)
Recovery attempt 5 of 6 will occur at 7:18 (51 minutes after the last failure)
~~Recovery attempt 6 of 6 will occur at 8:34 (76 minutes after the last failure)~~

Recovery attempt 6 of 6 will not occur since the interval has increased to 76 minutes surpassing the Max Interval of 60 minutes.

- **Time Tolerance -** R-1 begins the distribution process by creating a directory tree listing of both machines involved in the replication (Center and Edge). These snapshots are analyzed for differences and a Comparative Snapshot Matrix™ (CSM™) - a listing of the differences between the two directory tree listings - is created (see description of the **Planning** phase).

  If the replication process is interrupted, a preset time threshold will be utilized to determine if the target tree used in creating the Comparative Snapshot Matrix™ is still valid. If the threshold has been exceeded, the Job rolls back and goes into Recovery, repeating the snapshot if permitted to do so.

  A zero value in the **Comparative Snapshot Valid For** field indicates that it is valid for the life of the job.

  A similar concept is implemented in controlling the execution of **Pre-Distribution Commands**. In order to prevent the commands from being executed with each new attempt to run the job, a time limitation may be set. Only when a job is attempted at a point in time beyond this threshold will the pre-execution commands be repeated.

- **Manually Confirm Comparative Snapshot** – This function offers an additional method to assist in creating a perfectly customized distribution job. Selecting this option will pause the distribution job after the planning stage.

Select the job (identified by a unique preview icon in the **State** column) by clicking the right-mouse button, and choose the **Preview** option from the menu. A list of all of the transfers and deletions required will be displayed. Any of the changes indicated may be cancelled. To cancel a change, right-mouse-click on the file and select **Ignore**.

- To confirm a preview, click **Confirm**.
- To reject a preview, click **Cancel**, and proceed to abort the job.

NOTE: When **Manually confirm Comparative Snapshot** is selected, the job will not proceed past the preview stage without user input. This should only be used in running tests and not in a scheduled production scenario. If this option is selected, the job will not run until the snapshot is manually confirmed.

This option is NOT available for Continuous Updates jobs.

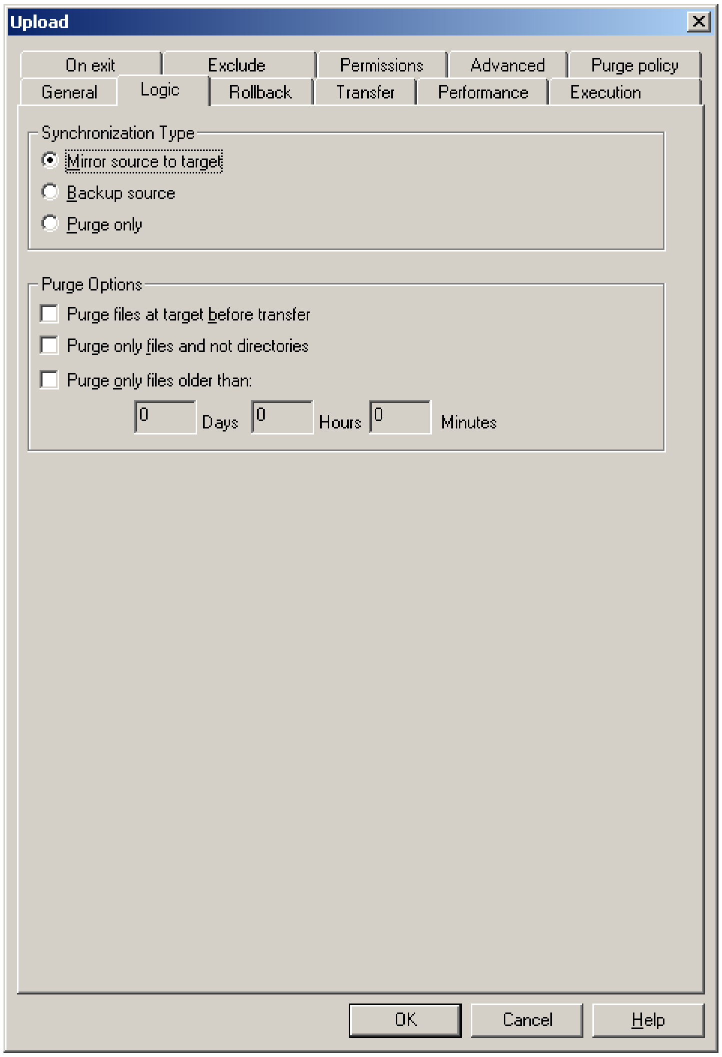
- **Abort on empty source**– If the source directory of the distribution is empty, **ALL** files on the target run the risk of being deleted. If this option is selected the job will abort if the source list is empty.

- **Report Options - Transfer Report Style –** Available report types are:
  - **Total** – Displays the least amount of job detail. Information includes job start/end time and number of files transferred.
  - **Log** - This style includes all **Total** information, as well as source/target (including path) filenames and the file sizes.
  - **Summary** – All information from **Log** and **Total** styles is included, as well as the transfer, start and end times of any file transferred.
  - **Detailed** - All of the above information as well as block level transfer details per file transferred.

    This report file may be very large. If the job includes the transfer of many files, or the Center has a limited amount of free space available, it is recommended that **Log** or **Total** styles be used.

    NOTE: Using the **File List** option in the **On Exit Tab**, specify here either **Summary** or **Detailed**. Any other report style will result in an **empty** file list being created.

3.3.2. Logic Tab



Synchronization Type

● Mirror source to target

○ Backup source

○ Purge only

Purge Options

□ Purge files at target before transfer

□ Purge only files and not directories

□ Purge only files older than:

0 Days 0 Hours 0 Minutes

Figure 16: Replication Logic Tab

- **Options** – Specify Edge synchronization logic:
  - **Mirror Source To Target –** At the end of the synchronization process, the Source and Target directories are exactly the same.
    - Files that exist on source but do not exist on target are copied to target.
    - Files that exist on both source and target – with any differing properties (size / modification date) - are copied from the source to the target.
    - Files that do not exist on source but exist on target are deleted from target.

If the **Purge Files Older Than** is checked, only files older than specified are deleted.

- Files existing on both source and target having differing NTFS permissions only are not transferred, however the NTFS permissions may be replicated by putting a check mark in the preserve/detect NTFS checkbox.
- Files that exist on both source and target that are identical will **NOT** be copied.

NOTE: Using **Mirror** logic, an additional option can be enabled to only delete files on the target that are older than a preset parameter. To activate this option, place a check mark in the **File Tab/Purge Files Older Than** checkbox and complete the required fields. This means "Do not purge files that were created / modified within the past **Days / Hours / Minutes**."

- **Backup Source** - Files from source are copied to target but no deletions are made on target machine.
  - Files that exist on source but do not exist on target are copied to target.
  - Files that exist on both source and target are copied from the source to the target only if they have a more recent creation / modification date.
  - Files that do not exist on source but exist on target are left unchanged.
  - **Important for Windows Target:** Mismatch - a file (directory) on the source has the same name and path as a directory (file) on the target - results in deletion of the file (directory) on the target and the creation of the directory (file) from the source to the target.

Example:

During a previous backup operation, file "Document" existed on the source but not the target, and was subsequently copied to the target. In the interim, the file "Document" was removed from the source. Since the job is being run under **Backup Source** logic, the file "Document" is not removed from the target. Now, however, a directory has been created on the source called "Document". Since Windows machines do not allow duplicate file / directory names, the directory "Document" cannot be copied to the target since the file "Document" still exists there.

R-1 will delete the conflicting file / directory on the target, copying the newer file / directory from the source.

- **Purge Only** - Files and directories that exist on target but do not exist on source are deleted from the target. This logic involves no transfer of files from the source to the target.

    - Using Purge logic, an additional option can be enabled to only delete files on the target that are older than a preset parameter. To activate this option, place a check mark in the **Purge Files Older Than** checkbox and complete the required fields. This means "Do not purge files that were created / modified within the past **Days / Hours / Minutes**."
    - Using a **Continuous Updates** job, if a directory is renamed, it is purged (deleted) from the target directory.

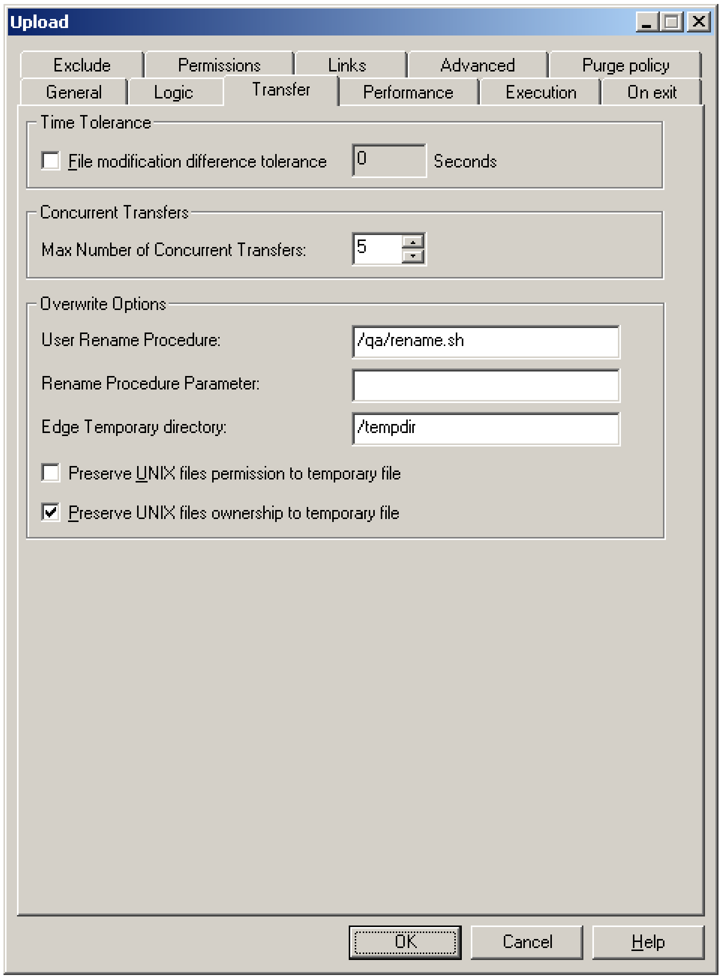
- **Purge files older than** – R-1 will not purge files that were created or modified within the specified time.

- **Purge Only Files and Not Directories** – If this option is selected, the target directory tree remains intact even if files are purged.

- **Purge files on target before transfer** - Check this option when transferring large files and there are also many files to delete from the target directory. This will prevent the replication from failing due to lack of disk space on the target machine.

  The default is to delete files from the target directory AFTER transferring changed files.

3.3.3. Transfer Tab



Time Tolerance

☐ File modification difference tolerance 0 Seconds

Concurrent Transfers

Max Number of Concurrent Transfers: 5

Overwrite Options

User Rename Procedure: /qa/rename.sh

Rename Procedure Parameter:

Edge Temporary directory: /tempdir

☐ Preserve UNIX files permission to temporary file

☑ Preserve UNIX files ownership to temporary file

Figure 17: Transfer Tab

- **File Modification Difference Tolerance -** The replication process may be fine tuned by adjusting the age threshold used by R-1 to decide whether an existing file should be overwritten by a newer version. This parameter sets a threshold used by R-1 to compare the modification date/time of similar files on the source and target.

  One application for the file time tolerance parameter would be to allow for the difference in time stamping when replicating between FAT and NTFS drives. In this case, a value of two (2) seconds would avoid the possibility of unnecessary file transfers.

  See **Time Stamps on FAT and NTFS Drives** for more information.

- **Concurrent Transfers** – Set the maximum number of concurrent transfers.

  Transfer performance can be boosted when performing simultaneous parallel transfers of partial file lists.

- **Overwrite Options**

NOTE: **Override After Completion of Transfer** feature is activated automatically by R-1 using Multicast or the LAN transfer engine. Using WAN engine, this option needs to be explicitly checked in the **Performance Tab** under the WAN transfer engine properties.

  - **User Rename Procedure – Rename Hook -** Enter the name of the rename hook script. The purpose of the Rename Hook is to replace R-1's regular rename process, that happens after a file is being copied to the temporary directory in the Edge, and moves the file from this temporary directory to the target directory.

    Instead of using the default permissions preserving, users can now add the "inherit target permissions" and develop a rename process that moves the file and sets its permissions according to the real target parent directory or any other mechanism.

    The procedure has to reside on the target machine.

    It accepts the following environment variables -

```
FASTCOPY_TEMP_FILE_NAME
FASTCOPY_TARGET_FILE_NAME
FASTCOPY_RENAME_HOOK_PARAM
```

    The script may look like that:

```
copy "%FASTCOPY_TEMP_FILE_NAME%" "%FASTCOPY_TARGET_FILE_NAME%"
REM Use the following line for WAN transfer only
DEL "%FASTCOPY_TEMP_FILE_NAME%"
exit 0
```

    In case of success the rename procedure should return 0.

NOTE: Temporary files are always deleted when using the LAN transfer. When using WAN transport engine, add the DEL "%FASTCOPY_TEMP_FILE_NAME%" command to your script.

NOTE: Wrap the environment variables in "" for files with spaces in their name.

- **Rename Procedure Parameters-** Enter the parameters to be passed to the rename hook script.

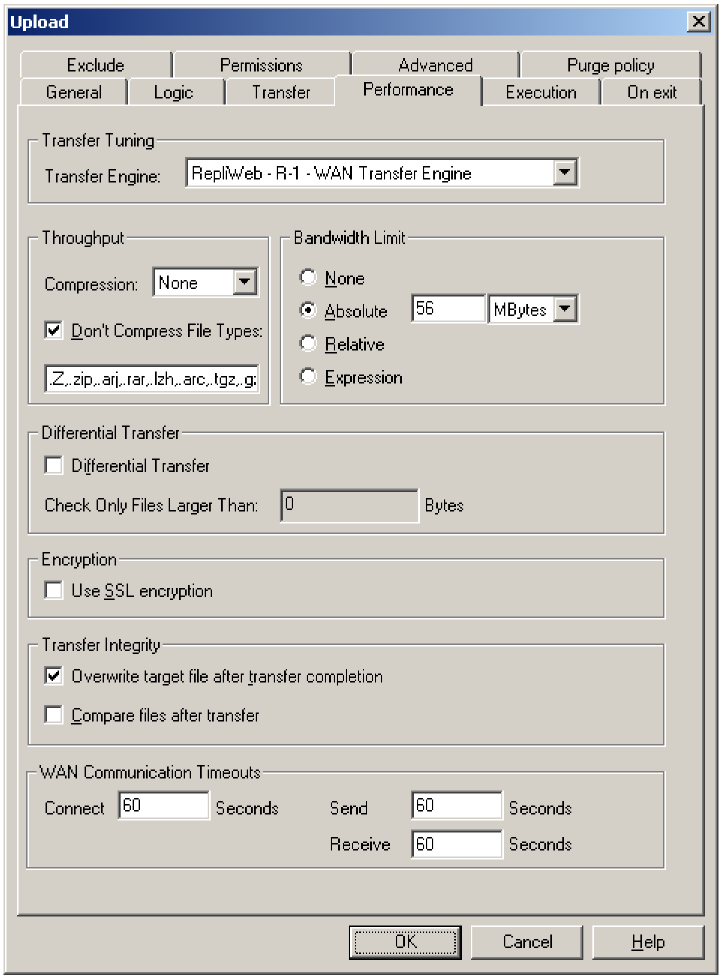
- **Edge Temporary Directory – Override After Completion** Specify the temporary directory on the Edge. If this is not set, R-1 will use its installation directory.

- **Preserve UNIX Files Permissions to Temporary Files** – Preserve UNIX permissions on the temporary files.

- **Preserve UNIX Files Ownership to Temporary Files** – Preserve UNIX ownership on the temporary files.

NOTE: These two UNIX options deal with the temporary file only. To preserve permissions of the target files, use the options in the **Permissions Tab**.

3.3.4. Performance Tab



Transfer Tuning

Transfer Engine: RepliWeb - R-1 - WAN Transfer Engine

Throughput

Compression: None

☑ Don't Compress File Types:

.Z,.zip,.arj,.rar,.lzh,.arc,.tgz,.g:

Bandwidth Limit

○ None

● Absolute 56 MBytes

○ Relative

○ Expression

Differential Transfer

□ Differential Transfer

Check Only Files Larger Than: 0 Bytes

Encryption

□ Use SSL encryption

Transfer Integrity

☑ Overwrite target file after transfer completion

□ Compare files after transfer

WAN Communication Timeouts

Connect 60 Seconds Send 60 Seconds

Receive 60 Seconds

Figure 18: Replication Performance Tab

- **Transfer Tuning –** Select the Unicast (upload/download) transport engine.
    - **R-1 LAN Transfer Engine -** Bandwidth limit selection, compression, and the differential transfer options are all voided. If these options are not needed, use this engine to significantly reduce transfer times.

      Relative performance increases will be especially large if the content consists of a large number of small files.

    - **R-1 WAN Transfer Engine -** The Wide Area Network transport options can significantly reduce transfer time in WAN environments by employing advanced features such as asynchronous compression. For more information see **WAN Transfer Engine Properties**

WAN Transfer Engine Properties

- **Throughput –** Select the compression type to be used and specify file types that should not be compressed.

  The volume of data being transferred and the network throughput rate are both significant factors in determining the overall time required to transfer data. The relative change in transfer time if compression is employed will depend on many factors, including the CPU speed and network throughput rate.

  Compression is performed transparently, without performing any intermediate spooling of compressed data.

  Two factors determine the total file transfer time:
  - **Block Compression Time** - This is influenced by the processing resources of the machine and the nature of the data being compressed.
  - **Compressed Block Transmission Time** - This is influenced by the network throughput rate.

  If compression of the subsequent block to be transmitted is completed before completing the transmission of the current block, transmission of the next block will not be delayed. Alternatively, if the block compression takes more time than block transmission, the transfer will be held up waiting for the compressed block to be ready for transport.

  Thus, if the transfer is being performed over a high-speed network and the server is heavily stressed, implementing compression can actually increase the total transfer time.

  Several compression options are available via the pull-down box:

  - **None** - No compression is used on the data to be transmitted.
  - **Normal** - A limited compression algorithm is utilized to compress the data.
  - **AP** - A moderately efficient compression algorithm, providing a higher degree of compression than the **Normal** algorithm, is utilized to compress the data.
  - **Power** - The strongest compression algorithm utilized, produces better compression results on executable files than the **Normal** and **AP** algorithms, but consumes considerable resource usage.

  Each compression algorithm employs different amounts of CPU resources. For most needs, AP is the optimal compression algorithm.

  - **Don't Compress File Types** – Files of the specified types will not be compressed.

Files that have already been compressed will yield no net compression after undergoing R-1 compression. Trying to compress them will result in undue CPU time.

See **Appendix A** for a complete explanation on selecting a compression algorithm and file types.

- **Bandwidth Limit** – Set the desired bandwidth option:
  - **None** – No bandwidth limit is set.
  - **Absolute** – Set absolute bandwidth limit in **MB|Mb|KB|Kb.**
  - **Relative** – Set relative bandwidth limit in **%**
  - **Expression** – Set bandwidth using an expression.

  See **Appendix B** for a complete explanation on setting Bandwidth limits.

- **Differential Transfer** – R-1 will only transfer the changes in a given file during transfer, down to the block level.

  - **Check only files larger than** - Applying differential transfer to small files can actually increase the total replication time. File size is in Bytes.

    There is no advantage at all in using the Differential Transfer option with files smaller than 64K.

- **Transfer Integrity**

  - **Target File After Completion Of Transfer -** If a user attempts to access a file that is in the process of being transferred, or if the distribution fails in the middle of distributing a file, only part of the file will exist on the target therefore users may access corrupted data.

    Using this option, the file will be transferred to a temporary location on the target, leaving the original file on the target intact. Once the entire file has been transferred to the target, the original file is overwritten, and the temporary file removed. This will ensure that users will only access a complete file.

    Using this option, fill in the **Overwrite Options** in the **Transfer Tab**.

    This option is automatically available when using The LAN Transfer or the Multicast engines.

  - **Compare Files After Transfer** - R-1 will analyze each distributed target file to ensure that it matches the source. This prevents problems in data integrity caused by external processes, such as another user or application altering either source or target data during distribution. When comparing files on the Center and Edge nodes, a CRC (Cyclic Redundancy Check) calculation on each file is performed and the results are compared. Only the results of the calculation are transmitted for purposes of comparison, so the comparison itself is immediate. Enabling this option can significantly decrease performance.

NOTE: On networks with fast lines and few or no communication problems, such as LANs, the use of these verification measures may slow down transfer operations. In this case, it might be best to dispense with some or all of these steps. On the other hand, they are essential for foolproof distribution on networks with frequent communication difficulties

- **WAN Communication Timeouts -** R-1 enables controlling TCP/IP communication timeouts, by using Application Timeouts. Setting these fields will override the operating system TCP timeouts. Value 0 means using the operating system timeouts. Specify here timeouts to be used by the WAN transport engine at the transfer stage.
    - **Connect -** Specify, in seconds, the TCP connect session timeout to be used during transfer using the WAN engine. Default is 60 seconds.
    - **Send -** Specify, in seconds, the TCP send operation timeout to be used during transfer using the WAN engine. Default is 60 seconds.
    - **Receive -** Specify, in seconds, the TCP receive operation timeout to be used during transfer using the WAN engine. Default is 60 seconds.
- **Use SSL Encryption** – Check this option if data transfer to/from the Edge should be performed using SSL encryption.

  For explanation about R-1 security, see the **Overview Chapter**.

LAN Transfer Engine Properties

- **LAN Communication Timeouts -** R-1 enables controlling TCP/IP communication timeouts, by using Application Timeouts. Setting these fields will override the operating system TCP timeouts. Value 0 means using the operating system timeouts. Specify here timeouts to be used by the LAN transport engine at the transfer stage.
    - **Connect -** Specify, in seconds, the TCP connect session timeout to be used during transfer using the WAN engine. Default is 0, using Operating

system defaults.

- **Send -** Specify, in seconds, the TCP send operation timeout to be used during transfer using the WAN engine. Default is 0, using Operating system defaults.
- **Receive -** Specify, in seconds, the TCP receive operation timeout to be used during transfer using the WAN engine. Default is 0, using Operating system defaults.

NOTE: **Overwrite After Completion of Transfer** feature is activated automatically by R-1 using Multicast or the LAN transfer engine. To override the Rename procedure or the default temporary directory, fill in the **Overwrite Options** in the **Transfer** Tab.

3.3.5. Replication Execution Tab

Data replication is often part of a wider operation. In that event, pre or post processing are needed and become an integral part of the replication operation. The built in recovery mechanism considers an operation incomplete if the post-distribution processing was not completed successfully, even if the distribution itself was. These commands can be run on either the Edge or Center systems, or both.

Commands can be:

- **Edge Pre-Command** – The command will be executed on the Edge prior to the submission of the Job.
- **Edge Post-Command** – The command will be executed on the Edge after a successful execution of the Job.
- **Center Pre-Command** – The command will be executed on the Center prior to the submission of the Job.
- **Center Post-Command** – The command will be executed on the Center after content distribution is successfully completed for a specific Edge.

- **Pre Commands Timeout** – Check here to specify the time to wait (in seconds) until the pre command on the Center and/or the Edge will execute. If the timeout have passed, the execution of the pre-command fails.
- **Post Commands Timeout** – Check here to specify the time to wait (in seconds) until the post command on the Center and/or the Edge will execute. If the timeout have passed, the execution of the pre-command fails.

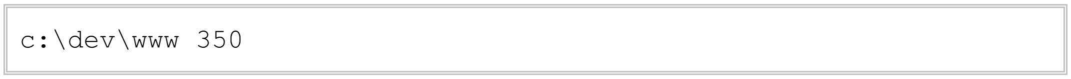
Valid **Command** entries are the full path to any single batch file or script that can be run from a command-line. Valid **Parameters** entries are any switches, qualifiers, etc., that the script or batch file accepts. Multiple parameters must be separated by a single space.

```
<command> <variable1> <variable2>
```

Where `<variable1>` and `<variable2>` from the batch file refer to user-entered values in the **Parameters** field(s).

Example:

If the parameters to be passed to the batch file are the path c:\dev\www and the value 350, the **Parameters** field should appear as follows:



```
c:\dev\www 350
```

Multiple parameters are separated by a space.

Note: The full paths to the commands are always required, as the batch files are run without any environment context, such as system search path, etc. Paths to batch files should always be entered in the native format of the system the command is being run on.

Example:

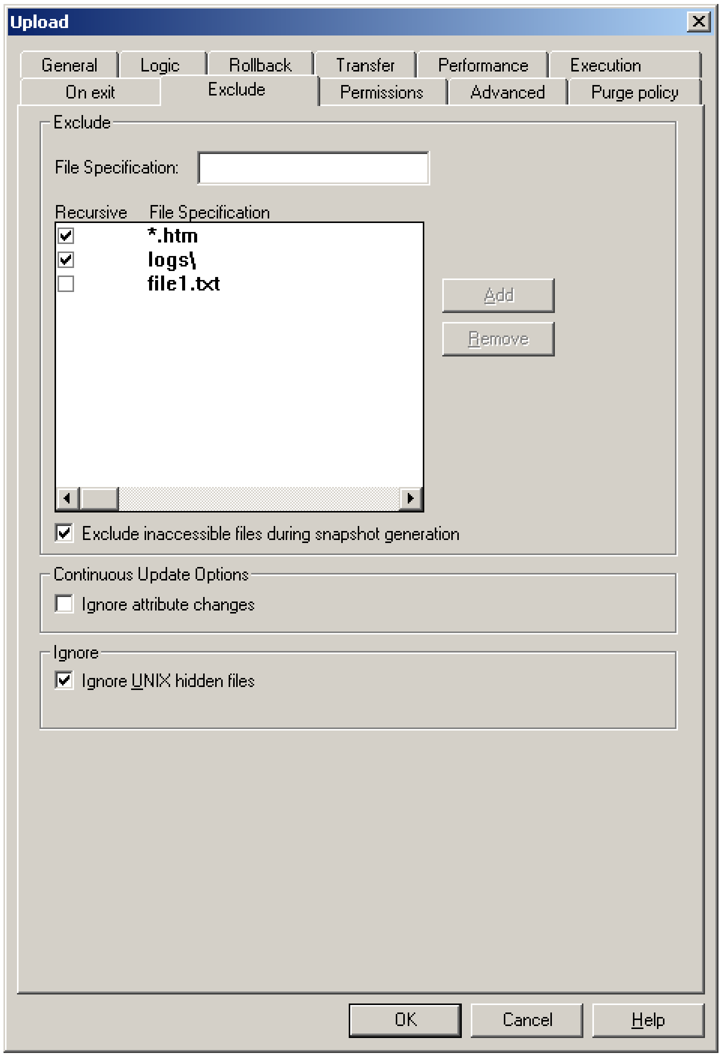
If an Edge post command is being run, and the Edge is a UNIX machine, it should be written as:

```
/scripts/restart.sh
```

Standard output from any command run via **Execution** is saved in R-1 reports. The report name corresponds to the command type. For example, standard output from an **Edge Post Command** would appear in a report titled “Post-Commands: Edge.”

3.3.6. Exclude Tab

NOTE: Use the **Manually Confirm Comparative Snapshot** feature available from the **General Tab**, to view and verify that the appropriate files will be excluded from the replication.



Exclude

File Specification:

Recursive	File Specification
☑	*.htm
☑	logs\
□	file1.txt

☑ Exclude inaccessible files during snapshot generation

Continuous Update Options

□ Ignore attribute changes

Ignore

☑ Ignore UNIX hidden files

Figure 19: File Exclude Tab

- **Exclude**
  - **File Specification** – Specify the files to be excluded from the replication. This can include a specific file name, or files of a specific type (e.g. `index.html or *.exe`). Here too, directory names may be entered to exclude a whole directory from being distributed.

    Examples: `file1.txt, *.htm, file*.*`

```
dir1\*,dir1\subdir1
```

- **Recursive** – Check this option to exclude the file spec from anywhere in the source replication tree.

  In the example shown on the figure above, directories named `logs` and all the files within them, and all `*.htm` files, will be excluded from the replication, regardless of their location on the source tree. The file `file1.txt` will only be excluded from the source directory level.

  For more information and samples on excluding files and directories, please refer to **Appendix K**.

  NOTE: Files that meet the Exclude file specifications will not be transferred, created, or deleted on the source and target systems.

- **Exclude inaccessible files during snapshot generation** – During snapshot generation some of the files may not be accessed (mainly due to permission issues). This will cause the job to fail. Check this option to exclude these files from the replication process and transfer the rest of the replication tree.

  NOTE: During the Snapshot Generation phase, files are accessed for retrieving details only. If the user has access to retrieve details of these files, it will pass this phase successfully. However, if that user does not have access to read or delete these files, the job will fail at the Transfer stage.

  To overcome access problems at the Transfer stage, uncheck the **Abort on error transferring files** at the **Advanced Tab**.

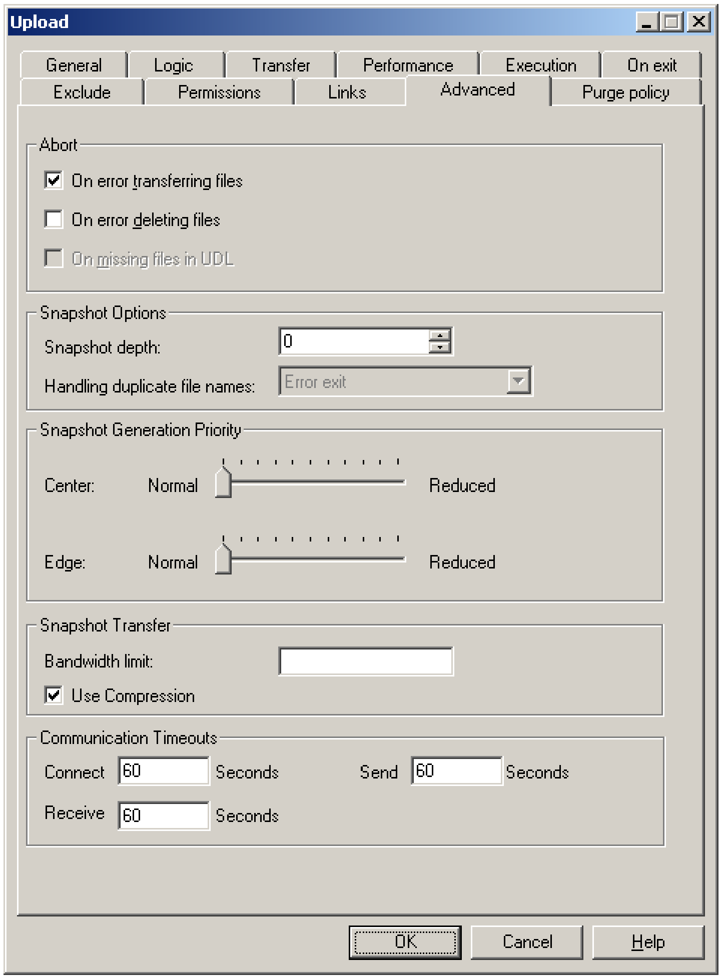
  NOTE: Files that could not be accessed during the snapshot phase, will be listed in the **Exclude** report.

- **Continuous Update Options – Ignore Attribute Changes –** In a Continuous Update job, do not replicate attribute changes if that is the only change that occurred.

- **Ignore UNIX Hidden Files** - R-1 will not transfer UNIX hidden files (i.e. files beginning with a period).

- **Ignore Windows invalid file names** – If the source is UNIX and the target is Windows, UNIX files whose names include invalid characters in Windows will not be transferred. If this option is not selected, and files of that kind exist, the job will fail.

3.3.7. Replication Advanced Tab



Abort

☑ On error transferring files

☐ On error deleting files

☐ On missing files in UDL

Snapshot Options

Snapshot depth: 0

Handling duplicate file names: Error exit

Snapshot Generation Priority

Center: Normal — Reduced

Edge: Normal — Reduced

Snapshot Transfer

Bandwidth limit:

☑ Use Compression

Communication Timeouts

Connect 60 Seconds Send 60 Seconds

Receive 60 Seconds

Figure 20: Advanced Properties Tab

- **Abort** - Specify the conditions in which a job will abort. More than one option may be specified.
    - **On Error Transferring Files** – Abort the job if failure occurred while transferring any of the files to/from the Edge(s).
    - **On Error Deleting Files** – Abort the job if a failure occurred during the deletion of target files during the synchronization process.

- **Snapshot Options** – Specify how the snapshot is created.
    - **Snapshot Depth** – The number of subdirectories to recurs through. The default is 0 (zero), which represents infinite depth.

- **Handling Duplicate File Names** –When R-1 encounters a situation where only one of either the source or target systems is case sensitive (i.e. UNIX), the user must decide how the discrepancies will be handled.

Example:

If the source contains files: File1, file1, and FILE1 and the target are a Windows machine, the target can only have a single "file1", regardless of case.

Select the handling method from the list:

- **Error Exit -** Exit the job with an error.
- **Select Newest** - Transfer the file with the newest modification date.
- **Select Oldest** - Transfer the file with the oldest modification date.
- **Select Lexicographic first** - Transfer the file that is first alphanumerically.
- **Select Lexicographic last** - Transfer the file that is last alphanumerically.

NOTE: Lexicographic order interprets capital letters before lower-case letters.

- **Snapshot Generation Priority** – The snapshot generation stage is a CPU intensive process. On especially large directory trees with a high number of files (hundreds of thousands → millions) this process can potentially leave few CPU resources for other active processes.

  Use the slider bar to reduce the CPU utilization of a given snapshot generation stage. These can be set individually for both Center and Edge. This will lengthen the time required for the snapshot generation, but leave system resources available for other processes.

- **Snapshot Transfer** – The snapshot is being transferred from the Edges to the Center. In case of very large snapshot files (large distribution trees), the snapshot transfer can be optimized.

  - **Bandwidth Limit** - Set bandwidth using an expression.
    See **Appendix B** for a complete explanation on setting Bandwidth limits.

  - **Use Compression** – Compress the snapshot file prior to its transfer to the Center.

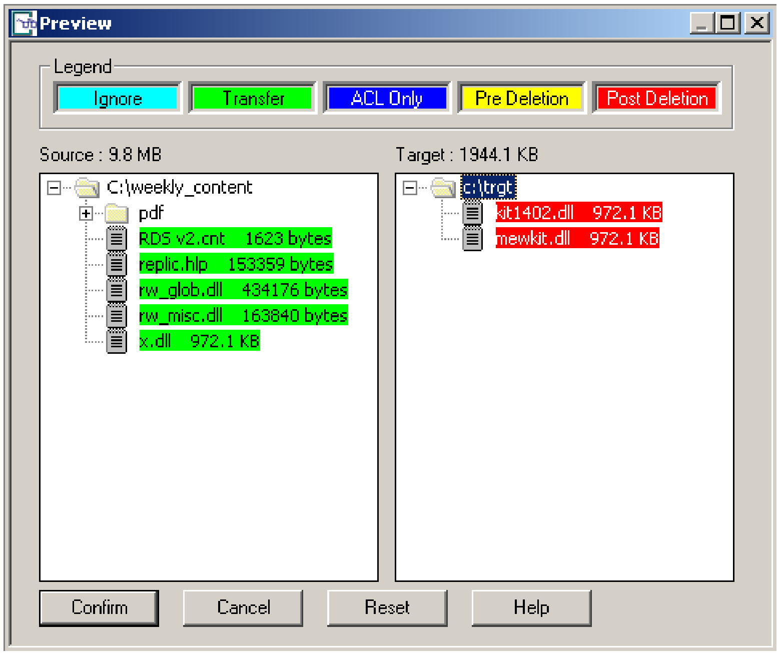
- **Internal Communication Timeouts -** R-1 enables controlling TCP/IP communication timeouts, by using Application Timeouts. Setting these fields will override the operating system TCP timeouts. Value 0 means using the operating system timeouts. Specify here overall R-1 communication timeouts. Timeouts can also be set for each of the transport engine separately for the transfer stage.

- **Connect -** Specify, in seconds, the TCP connect session timeout for general communication within R-1 components. Default is 60 seconds.
- **Send -** Specify, in seconds, the TCP send operation timeout, for general communication within R-1 components. Default is 60 seconds.
- **Receive -** Specify, in seconds, the TCP receive operation timeout, for general communication within R-1 components. Default is 60 seconds.

3.3.8. Replication Preview Window

The preview window enables review of the files that are about to be transferred or deleted during the replication process.

This window is available whenever the user selected to **Manually Confirm Comparative Snapshot.**



Source : 9.8 MB

```
C:\weekly_content
  pdf
  RDS v2.cnt    1623 bytes
  replic.hlp    153359 bytes
  rw_glob.dll    434176 bytes
  rw_misc.dll    163840 bytes
  x.dll    972.1 KB
```

Target : 1944.1 KB

```
c:\trgt
  kit1402.dll    972.1 KB
  newkit.dll    972.1 KB
```

Figure 21: Snapshot Confirmation

The Preview pane shows the source and target directory, and the files located in each of them. The handling of the files during replication depends on the Synchronization Type (Mirror, Backup, Purge).

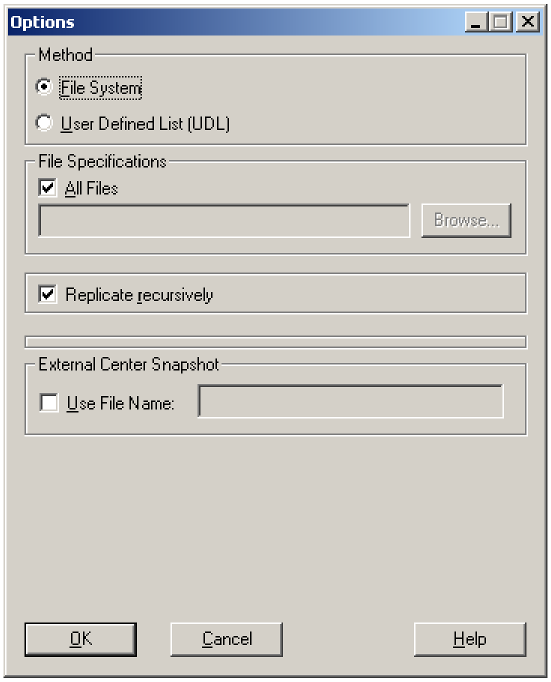
The legend shows how the file will be handled during the replication process:

- **Ignore** – Right click on a file and press ignore to ignore the file during replication.
- **Transfer**– The file will be transferred to the target directory.
- **Permissions** – Only the file’s permission will be transferred.
- **Pre Deletion** – In some cases, files on the target need to be deleted before the transfer takes place. For example, if the source has a file named `content` and the target has a directory named `content`, the target directory needs to be deleted prior the transfer for the transfer to take place.
- **Post Deletion** – The file will be deleted from the target directory after the transfer is complete.

- **Reset All** – Cancel all changes made in the preview list.

- **Cancel** - Cancel all changes made in the preview list and close the Preview window. The job will remain in Preview state until Confirm is selected.
- **Confirm** – Apply changes in the list (if any) and continue the job.

3.3.9. Replication Options Window



Method

● File System

○ User Defined List (UDL)

File Specifications

☑ All Files

☑ Replicate recursively

External Center Snapshot

□ Use File Name:

Figure 22: Snapshot Options Tab

- **Method** – Specify how the snapshot will be created:
    - **File System -** The files to be synchronized will be obtained directly from the directory specified in the **Source Directory** in the Job Definition window, being filtered by the use of file and exclude specifications as applicable.
    - **User defined list (UDL) -** The file list that will be synchronized will be taken from a text file supplied by the user. The file can either be created directly by the user or generated by an outside process. For a full explanation regarding the format and behavior of the User Defined List option, see **Appendix E**.

- **File Specification** – Specify the files to be transferred:
    - **All Files -** By default, R-1 will synchronize the entire directory structure (**Recursively** from the source **Path**) and all of its contents (**All Files**).

      File specification may be used to limit the distribution to a specific file, or files of a specific type. File specification can include wildcards.

      Examples: `file1.txt,.htm,file*.*`

      Entire subdirectories may also be synchronized in the same way. Simply enter the folder name followed by a slash and asterisk (e.g. `dirname\*`).

Examples: `dirname\*`
folder1\subfolder1\subfolder2\*,folder5

NOTE: Files that do not meet the file specifications will not be transferred, created, or deleted on the source and target systems.

- **Replicate Recursively** – Specifies whether to replicate the current directory only or all directories beneath it.

  Specifying an explicit file name in the list will cause the whole directory tree to be built at the target machine. Only the files meeting the file specification are transferred but the complete tree is created.

NOTE: Use the **Manually Confirm Comparative Snapshot** feature available from the **General Tab,** to view and verify that the appropriate files will be transferred in the replication process.

- **External Center Snapshot** – Enter the **Directory** in which the external snapshot files reside.

  In cases where the CST™ and CSM™ processes may take a long time, the user can create the local (Center) snapshot lists. Use this option to specify to R-1 not to perform the Center's snapshot but rather use a previously built one. All files that must reside in the given directory.

  See **Appendix G** for a complete explanation on this feature.

The following fields are enabled only if the Snapshot creation method is UDL:

- **UDL Origin** – If the file list is taken from a UDL, there are four choices for the origin of the User list. These values state where and when the list will be taken from. Although the absolute path of the list will be from the field **UDL File Name**, the system (Console / Center / Edge) must be selected.
  - **Console -** The list will be copied from the system the Console is being run from at the time the job is submitted to the Center. If the job is scheduled, the same list will be used for all subsequent jobs.
  - **Center Submit -** The list will be copied from the Center at the time the job is submitted. This copy will be used as the list for the job. If the job is scheduled, the same list will be used for all subsequent jobs.
  - **Center Runtime -** The list will be taken from the Center at the time the job is actually run. Since a scheduled job spawns an Edge Job to run at each

scheduled time, each scheduled job will use the list as is when it is spawned.

- **UDL File Name** – Enter the name of the UDL text file. A full path is required.

- **Apply UDL to target -** The use of a UDL files on the source machine results in greatly reduced system resources during the snapshot generation stage of the replication. This is because the snapshot is assumed to be the UDL itself. However, on the target machine, a standard snapshot is still issued. When this option is enabled, the UDL is transferred to the target and only the files present in this list are examined for existence and modification. This greatly reduces the associated costs of the target snapshot for large content repositories.

NOTE: For a full description of the UDL feature, please refer to **Appendix F**.

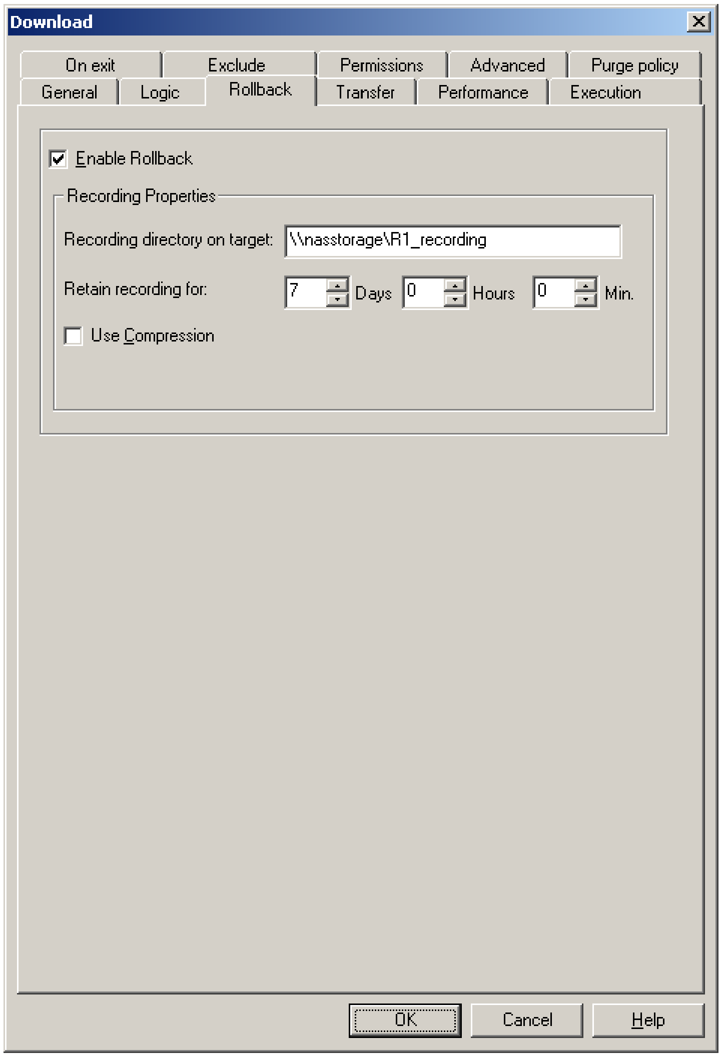
NOTE: The methods in which R-1 deals with any errors in the User Defined List text file can be found in **Appendix F**. Before using this option, please consult the appendix carefully. ***Failure to do so may result in incorrect functionality***.

3.4. Common Job Property Tabs

The following tabs are part of the Replication and Distribution job properties and may be accessed from either job definition property pages.

3.4.1. Rollback Tab

Rollback functionality provides a means of effectively undoing the effects of multiple replications and distributions on a target system to a user-specified ‘point-in-time’. This allows the system administrator to undo the changes carried out by a one-time, scheduled, or continuous update R-1 job.



☑ Enable Rollback

Recording Properties

Recording directory on target: \\nasstorage\R1_recording

Retain recording for: 7 Days 0 Hours 0 Min.

☐ Use Compression

Figure 23: Rollback Definition – Upload / Download / Distribution

Rollback functionality is employed through the ‘recording’ of all replication related changes on a target system, followed by a potential ‘roll back’ of the content at a later time. The recording phase occurs prior to any propagation of changes to a target system and includes the recording of creation, deletion and modification to files and directories (including ownership and permission changes).

NOTE: This tab is only used to define the properties of the recording phase of a job. To actually implement a rollback see **Rollback Job** description in this Guide.

- **Use Rollback –** Check this option to enable rollback recording on the target. Recorded information includes:
  - Files and directories created on the target
  - Files and directories removed from the target
  - Files and directories modified on the target (included modified permission, ownership and Windows ACLs)

  NOTE: Recording only applies to data impacted by a replication or distribution as defined above – it does not record a complete copy of the target directory.

- **Recording Properties –**
  - **Recording Location** – Specify where the recorded data will be stored.

    The recording location may be specified as a path on the target machine (`D:\r1_recordings`) or a network share (`\\nasstorage\R1_recording`)

    Since the recording operation takes place on the target (i.e. an Edge during an upload job or a distribution, or the Center during a download job), the recording directory must be accessible for read and write with the credentials used to access the target file system. This refers to the Edge user defined in an upload or distribution job, or the Center user (i.e. the username used to connect to the Console, or used in the command line) in a download job.

  - **Retain recordings for** – This field defines how long recorded information should be kept for and directly relates to how far back a job can be rolled back. For example, a time of 18 hours will allow the job to undo all replication related changes that took place on the target system for the last 18 hours.

  - **Use Compression** – Enabling this feature will store all recorded data in a compressed form. While preserving disk-space, this feature will increase recording and rollback times.

3.4.2. Setting Permissions Tab

This tab allows setting of permission settings. Available options depend on the operating systems taking part in the replication process (Windows, UNIX). Preserving permissions is only possible when the source and target are of the same nature.

- **Preserve Root Directory Permissions –** Replicate root directory permissions. This option is available both for Windows ←→ Windows and UNIX ←→ UNIX replication.

Windows Permissions [Windows←→ Windows]

- **Replicate Shares [NT ←→ NT] -** Use this option to replicate the directory shares (meaning whether the folder is shared or not).
    - **Disk Shares** – Allows all network users to view this folder (e.g. `myshare`).
    - **Special Shares –** Hidden system shares (e.g. `A$`). Allows administrative personnel to connect to the root directory of a drive.

      For a Windows 2000 Professional computer, only members of the Administrators or Backup Operators group can connect to these shared folders. For a Windows 2000 Server computer, members of the Server Operators group can also connect to these shared folders.

- **Preserving NTFS Attributes [NT ←→ NT] -** When transferring a file, R-1 offers the option of maintaining the file's NTFS permission attributes (ACL/Security Descriptor). If the **Preserve NTFS Attributes** option is selected, the NTFS permissions of the source file will be transferred with the file to the target. If a file of the same name exists on the target, the data and the permissions will be overwritten with those of the source file.

  If the distribution target is a FAT drive, NTFS attributes will not be retained.

  If this option is not selected, the file data will be transferred but will assume the permissions of either a preexisting file of the same name as the file being transferred, or assume the default permissions for new files created by the logged-in user (for the Center, this is the user that connected via the Center Settings window; for the Edge, this is the user whose name and password were entered in the Job Definition window).

  The default setting of this field does **NOT** preserve the NTFS attributes. It is especially important to consider the implications of this when replicating within domains.

NOTE: Both Replicating Shares and Preserving NTFS Attributes require Administrator group privileges on the target system.

- **Preserve Alternative Data Streams [NT ←→ NT]**

  When transferring a file, R-1 offers the option of replicating Macintosh Resource Fork files, represented in NTFS as Data Streams.

NOTE: Replicating Shares, Macintosh Resource files and Preserving NTFS Attributes will only take place when the source and target systems are Windows.

UNIX Permissions [UNIX ←→ UNIX]

- **Preserve UNIX permissions**
  - **SUID** - Replicates the SUID of the files selected for replication
  - **SGID** - Replicates the SGID of the files selected for replication

UNIX Ownership [UNIX ←→ UNIX]

- **Preserve Ownership**
  - **UID** - Replicates User ownership of the files selected for replication
  - **GID** - Replicates Group ownership of the files selected for replication

Linux Extended Attributes [UNIX ←→ UNIX]

- **Preserve Extended attributes –** Enter a list of extended attributes to preserve.

NOTE: Preserving UNIX Attributes will only take place when the source and target systems are UNIX.

UNIX Target

- **Set Created Directories Permission –** Set the value of the directory permissions, 0-777 (Octal method). Permissions will be set to all created sub-directories in the distribution target tree. Directory permissions are set only to directories (and sub-directories) that are actually created during the replication process.

  Check the on creation checkbox to set the directory permissions upon its creation and not at the end of the replication process, to avoid permission problems of not being able to write the files to the directory.

- **Set Transferred Files Permission –** Set the value of the file permissions, 0-777 (Octal method). Permissions will be set to all the files transferred during the distribution process.

- **Set Root Directory Permissions –** Manually set the permissions of the distribution target root directory, 0-777 (Octal method).

  Root permissions are set whether the directory was created as part of the replication or already existed.

NOTE: Setting permission requires appropriate privileges.

NOTE: UNIX permissions (for files and directories) are represented using the octal method. Numbers between 0-777, which are not octal, are invalid.

For example,

```
-rwxrwxrwx = 777  root: read, write, execute
                  group: read, write, execute
                  others: read, write, execute

-rw-r--r-- = 644  root: read, write
                  group: read
                  others: read
```

NOTE: Setting files and directory permissions only affects files that are actually transferred during the replication process and directories that are created during the replication process.

The Preserve Permissions options, affect files or directories that already exist on the target. If the permissions have changed, only the permissions will be transferred during the replication process, although the files are not.

3.4.3. Links Tab [UNIX Only]

This tab allows UNIX Links handling while replicating content. This is only possible when the source and target are UNIX systems.

- **Links to Files**
  - **Ignore -** Do not replicate links to files at all
  - **Create Links [UNIX → UNIX only] -** If a link is encountered on the source, create it as a link on the target.
  - **Create Files -** If a link is encountered on the source, create the file on the target with the name of the link, but the contents of the file the link points to.

- **Links to Directories**
  - **Ignore -** Do not replicate links to directories at all
  - **Create Links [UNIX → UNIX only] -** If a link to a directory is encountered on the source, create it as a link on the target system as well.
  - **Create Directories -** If a link to a directory is encountered on the source, create it as a real directory on the target system, inside the target tree.

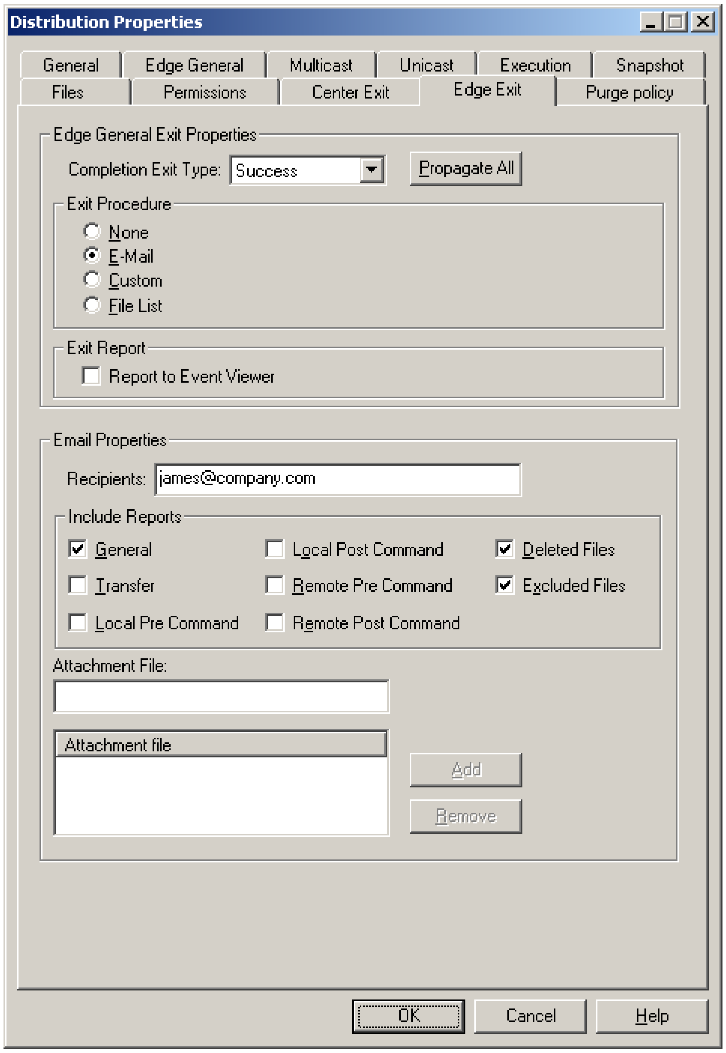
- **Follow symbolic links to directories** – If checked, R-1 will follow on the replication source the directory link and include the files from that linked directory in its snapshot as files that reside in a directory with the link name. Links to directories on the replication target are not followed but rather listed as links.

3.4.4. On Exit / Center Exit / Edge Exit Tabs

The **On Exit** option allows the linking of procedures to the 'exit' status of the job. These procedures will be executed only upon job completion. A failed attempt, that is, a job entering a **Recovery** state does not constitute a completed job.

In the **On Exit / Center Exit** Tab, specify the procedures to be executed on the Center upon completion of the submitted job.

In the **Edge Exit** Tab, specify the procedures to be executed on the Center, upon completion of an Edge job.



Edge General Exit Properties

Completion Exit Type: Success

Exit Procedure

○ None
● E-Mail
○ Custom
○ File List

Exit Report

□ Report to Event Viewer

Email Properties

Recipients: james@company.com

Include Reports

☑ General
□ Transfer
□ Local Pre Command
□ Local Post Command
□ Remote Pre Command
□ Remote Post Command
☑ Deleted Files
☑ Excluded Files

Attachment File:

Attachment file

Figure 24: Center Exit Tab – Email

The **On Exit** feature allows the user to design procedures to be followed at the end of the job.

- **Completion Exit Type** – Select the exit type from the drop down list, to define the notification method for that exit condition.

    - **Abort** - Define operation to perform in case the job aborts.
    - **Success** - Define operation to perform in case the job end successfully.
    - **Error** - Define operation to perform in case the job fails.

- **Exit procedure** – For each completion type, select the method used for notification of job completion by choosing one of the following:

    - **None** - No exit procedures specified.
    - **E-mail** - An email is sent upon completion of the job.
        - **Recipient** – Enter the e-mail address of the e-mail recipient(s). More than one e-mail address may be entered. Multiple e-mail addresses should be separated by commas.

        - **Include Reports**– Select the replication reports to be attached to the E-mail.

          For more information about the available reports refer to **Appendix M – Reports**.

        - **Attachments -** Files may also be attached to e-mails by specifying their full path. More than one file may be attached.

      NOTE: To use the e-mail notification feature be sure that the appropriate SMTP fields were completed during the installation procedure. If these fields were left blank, refer to the **Manage – Email Configuration Tab** to correctly set-up e-mail notification.

    - **Custom -** User defined procedures to be followed on exit from a job. These procedures may require user-defined parameters as well. The **Custom** exit feature allows integration with any other Command Line applications. For detailed instructions please consult the section on Pre / Post distribution commands.

    - **File List –** R-1 enables user to perform various operations on the list of files transferred during a distribution job. The transfer reports are parsed according to a template file.

      For complete information regarding **File List** please refer to **Appendix J**.

      NOTE: To use the File List option, set the properties **Transfer Report Style** under the **Edge General** Tab to either **Summary** or **Detailed**. Any other report style will result in an **empty** file list being created.

  - **Template File –** The input file on the Center for creating the output transferred file list.
  - **Target Directory –** The path on the Center where the output transferred file list is created. Enter the name of an existing directory on the Center, or use the Browse button.

  The output file (the result of the transferred file list based on the template file) may be used in different ways.

  - **None** – The output file is created in the target directory, but no operation is performed on it.
  - **Mail File** – The output file will be emailed to the specified recipient(s). More than one e-mail address may be entered. Multiple e-mail addresses should be separated by commas.
  - **Execute File** – The output file will be executed on the Center.

- **Propagate All** – Use the same specified procedure for all Exit functions.

- **Report to Event Viewer** – For each completion type, report the Job's exit message to the Windows Event Viewer or the UNIX syslog.

  For more information refer to **Appendix I**.

3.4.5. Purge Policy Tab

Configure the Purge Policy depending on the completion status of a job.

The logs that R-1 generates (i.e. General report / Transfer report) can become quite large depending on the size of a given job, the detail level of the reports, and the number of jobs that are actively running, or have already been archived. As these logs can occupy vast amounts of disk space, R-1 is configured to automatically purge them after a given amount of time.

NOTE: When a job has reached its purge age, it will be deleted. A purged job is no longer viewable through either the Console CLI or the Console GUI.

For each of the types of completed jobs **On Success, On Error** and **On Abort**, define the time the job**-** reports should be purged:

- **Use Server Defaults** - For instructions on configuring server defaults, consult refer to the **Manage – Scheduler** Tab.

- **Purge After**- Specify the duration (Days/Hours/Minutes) in which to keep the Job's reports.

- **Never Purge** – Specify to R-1 never to purge a Job.

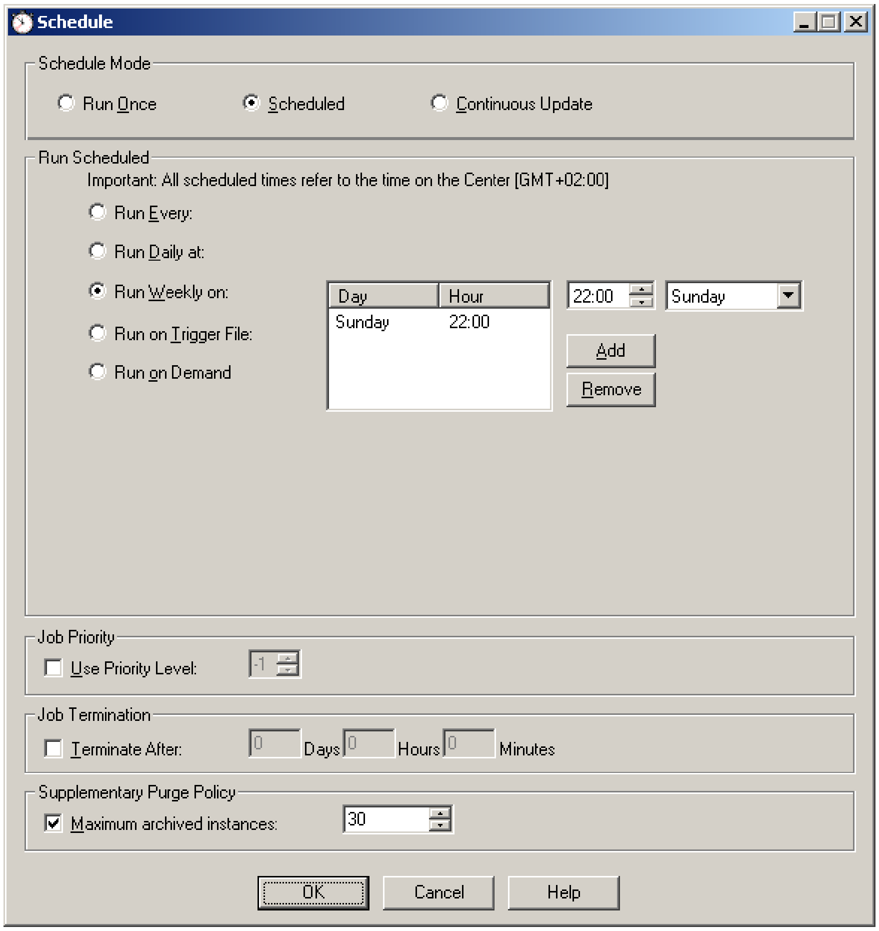
Job reports can also be sent automatically via e-mail using the **On Exit** tab in the job properties**.** This will allow the user to be actively informed of the completion status of a job, and allow the user to keep the reports as long as is required. These reports can similarly be configured to only be sent on Success / Error / or Abort status of a job.

3.5. Scheduling Jobs Window

The **Scheduling** window allows the selection of when a particular R-1 job will run. Jobs may be scheduled to run immediately, continuously, or on a more complex schedule. Job initiation and termination are defined in the R-1 Console and managed by the R-1 Center.

Any **Scheduled** Job will appear with a clock icon in the Console GUI. The job will spawn Edge jobs when the predetermined conditions are met.

Any **Continuous Updates** job will appear with a special icon in the Console GUI. The running instances of that job will appear in the **Details** screen



Schedule

Schedule Mode

○ Run Once ● Scheduled ○ Continuous Update

Run Scheduled

Important: All scheduled times refer to the time on the Center [GMT+02:00]

○ Run Every:

○ Run Daily at:

● Run Weekly on:

○ Run on Trigger File:

○ Run on Demand

Day	Hour
Sunday	22:00

22:00 Sunday

Job Priority

☐ Use Priority Level: -1

Job Termination

☐ Terminate After: 0 Days 0 Hours 0 Minutes

Supplementary Purge Policy

☑ Maximum archived instances: 30

Figure 25: Scheduling a Job

- **Run Once** – Specify whether the job will run immediately or at a later time.
    - **Run Immediately** – The job will run immediately following submission. This is the default behaviour.
    - **Run Once At** - Run a job once on a specific (future) date at a specific time.

- **Scheduled** – Set the job to start based on certain conditions:
    - **Run Every** - Run a job at a specific time interval (days/hours/minutes).

      The scheduling job spawns the first job immediately and starts counting the interval for the next one. The next job will be spawned only after the previous one has finished. It will start either after the specified time has passed, or immediately after the previous job has finished, if it ran longer than the delta specified.

      NOTE: Every spawned Edge job has a different Job ID. The scheduled job itself "runs" forever, until it's aborted by user command.

    - **Run Daily** - Run a job daily at the specified time (hh:mm).
    - **Run Weekly On** - Schedule a job to run on a weekly schedule. Multiple days and times can be configured.
    - **Run on Trigger File** – The job will run when a specific file will be located. The Center continuously scans for the existence of the defined trigger file (e.g. `C:\TriggerDir\trigger1.txt)`. When found, the trigger file is deleted and an Edge job is spawned, performing the Distribution. The user can generate the trigger file manually, by a post-Distribution command issued by another R-1 job, or through an outside process.

      For Distribution jobs, the Trigger file can be located on the Center only. For Upload and Download jobs, the Trigger file location can be: on Center, on Edge, on both Center and Edge, on either Center or Edge.

    - **Run on Demand** – A scheduled job will be initiated, but no replication process will be start until the user selects the **Submit Now** menu option.

      On-demand jobs are similar to Templates, however instead of submitting a job from a template file, the user actually submits a job from a "sample" job. It can be used for easy generation of identical jobs, or simply to have a backup of regularly used jobs for future resubmission

      NOTE: Hitting the **Submit Now** button while an on-demand job is already running, the newly submitted job will initiate only when the previous one completes.

        - **Behalf of Users / Groups** - IT administrators can use this option to prepare a "template" for project managers, who in turn, will submit the job when required. Specify Users and Groups of users who may see and submit this job. These users will issue the **Submit Now** command to initiate the distribution process.

Examples:

Users:	`WEB_Project_Manager`
Domain Users:	`DOMAIN\Domain_User`
Machine-specific users:	`comp_srv\server_user`
Groups:	`Sales_Group` `qa_srv\QA_group`

NOTE: On Demand – on Behalf jobs are visible by the user who submitted the On Demand Job and any user whom the job was run on behalf of (either specifically or as part of a group).

- **Continuous Updates** – Set an event-driven job that replicates changes as soon as they occur on the Source directory.

  For detailed information on event-driven, Continuous Update jobs, refer to **Appendix H**.

- **Use Priority Level** – **Priority** is a number between 0-10 to define the Job's priority (where 0 is the lowest priority and 10 is the highest). When a job is submitted, all jobs with lower priority are held. When the job finishes, the jobs are resumed and continue to run. The default priority is 5 (medium priority).

  Uncheck the field to run jobs that will be ignored while calculating priorities. Jobs like that will not cause lower-priority job to hold, and will not be held when jobs with a higher priority are submitted.

  This option is not available for Continuous Update jobs.

- The **Terminate After** field may be used to ensure that the job is terminated if it is not completed within a specified period of time. If **Terminate After** is enabled, the termination timer starts when the job begins.

  This option is not available for continuous update jobs.

- **Maximum archived instances** – For Scheduled and Continuous updates jobs only, specify the maximum number of instances to preserve in the archive directory.

NOTE: Use this option with care, as it may consume a lot of disk space. Keeping 30 instances of a distribution job to 10 Edges will result in keeping 300 jobs.

3.6. Center Management

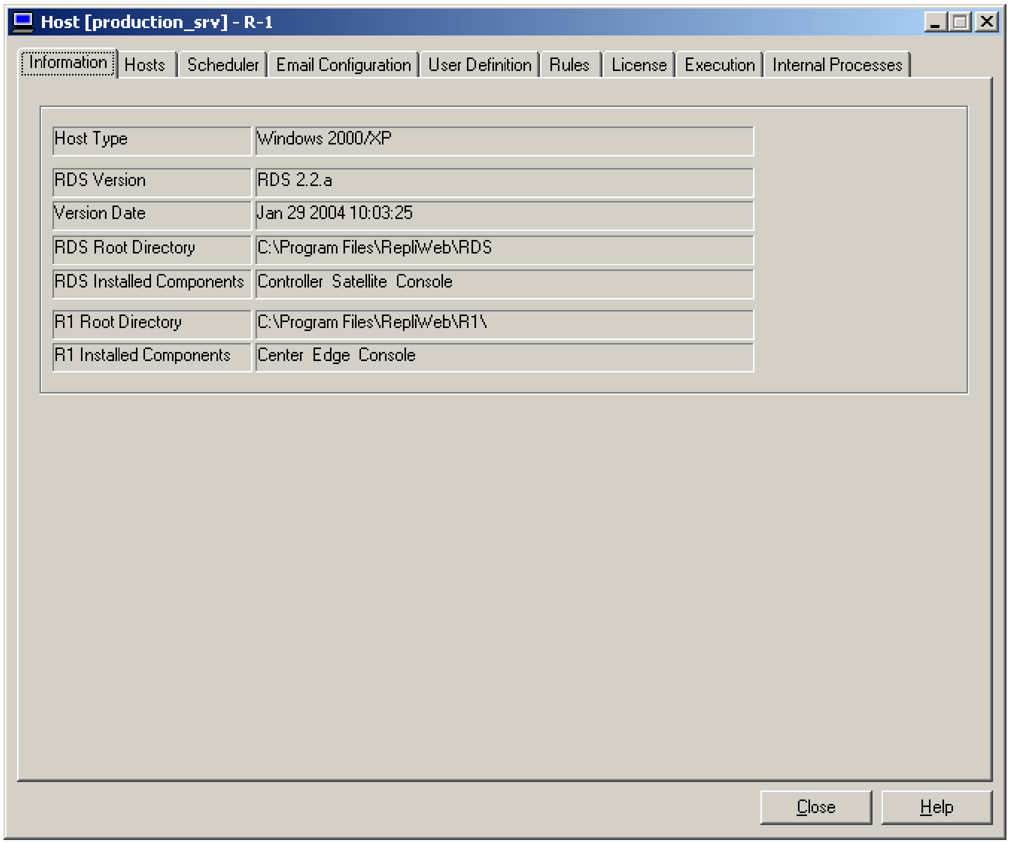
The Manage option is available for members of the Administrator group only. Connected to a UNIX Center, use `root`, connected to a Windows Center, use a member of the Administrator group on the Center.

NOTE: Connecting with an R-1 3.3 (and above) Console to older versions of R-1 – a registry key needs to be added to allow user to Mange an R-1 Center.
On the R-1 Center, add a **String** Registry key called **ManageGroup** under **HKEY_LOCAL_MACHINE\Software\RepliWeb**.
As the value, put the list of names of users who are authorize to Manage that Center.

This option enables changing system parameters, which are not Job specific.

3.6.1. Information Tab

The Information tab displays the build date and version of the various RepliWeb components installed on the Center. In addition, it displays the operating system of the Center.

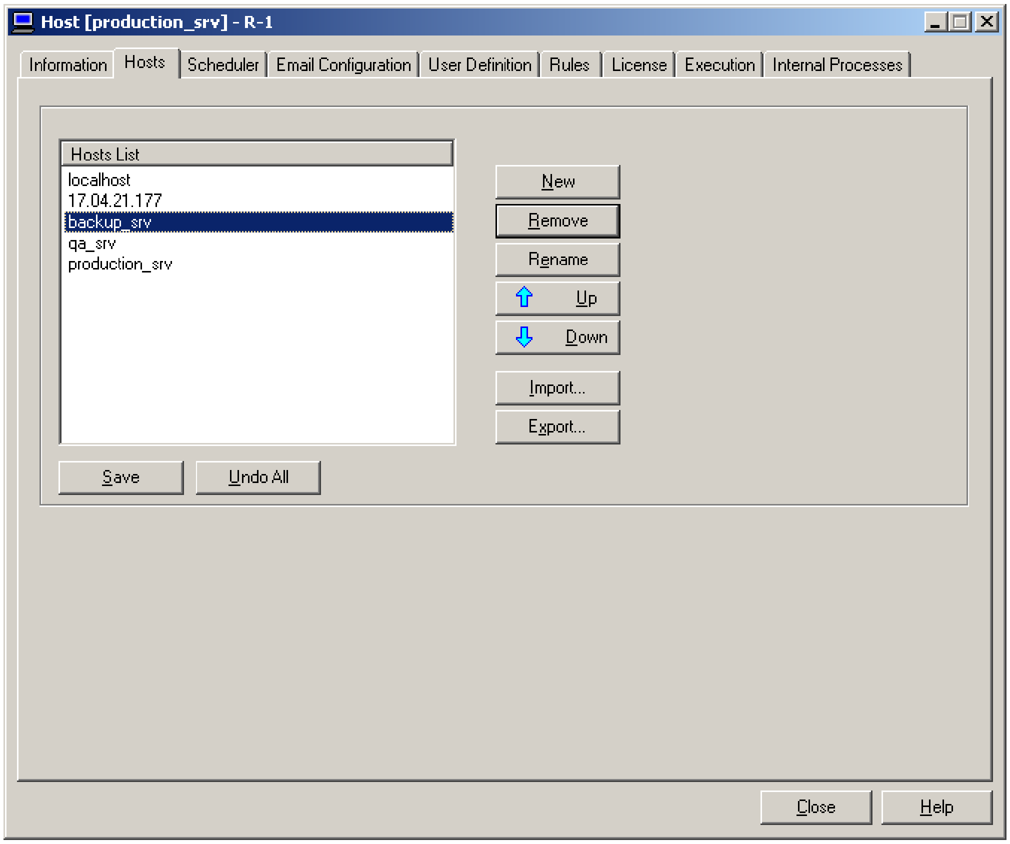


Host Type	Windows 2000/XP
RDS Version	RDS 2.2.a
Version Date	Jan 29 2004 10:03:25
RDS Root Directory	C:\Program Files\RepliWeb\RDS
RDS Installed Components	Controller Satellite Console
R1 Root Directory	C:\Program Files\RepliWeb\R1\
R1 Installed Components	Center Edge Console

Figure 26: Information Tab

3.6.2. Hosts Tab

The Hosts tab allows updating the Hosts list on the managed Center.



Hosts List
localhost
17.04.21.177
backup_srv
qa_srv
production_srv

Figure 27: Managing Hosts

The RepliWeb Hosts list reside on the R-1 Center, listing the hosts the user can replicate between.

This tab allows changing the order of the hosts in the list, and also exporting and importing the list of hosts. This enables easy assignment of the same hosts list in the entire organization.

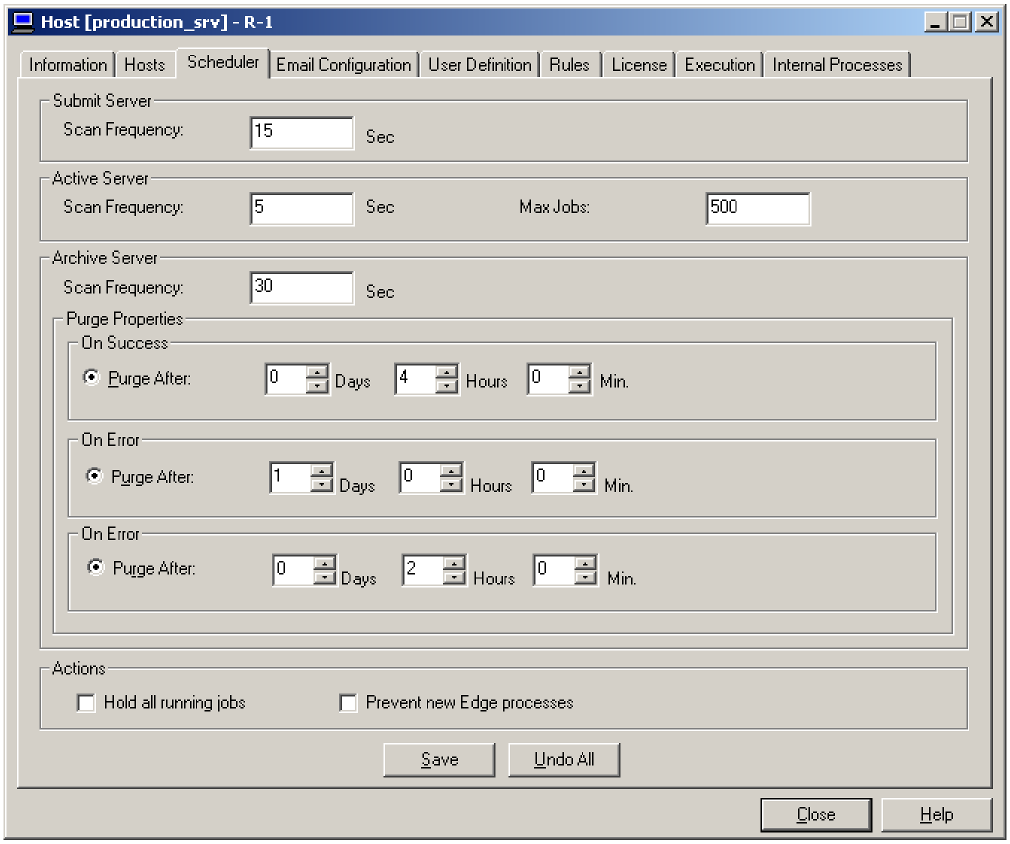
- **New** – Add a Host to the hosts list
- **Remove –** Remove a host from the hosts list
- **Rename –** Rename a host in the hosts list
- **Up –** Move the selected host one level higher in the list.
- **Down –** Move the selected host one level down the list
- **Import –** Import a list of hosts from a file on the Console.
- **Export –** Export the list of hosts to a file on the Console.

NOTE: None of the changes will take effect until the Save button is pressed. Only new jobs will be affected by these changes.

- **Save –** Save the list of hosts to the Hosts file on the Center.
- **Undo All –** Undo changes done to the list. The list of hosts is re-loaded from the Center.

3.6.3. Scheduler Tab

The Scheduler tab enables the user to set the Scheduler properties.



Submit Server
Scan Frequency: 15 Sec

Active Server
Scan Frequency: 5 Sec Max Jobs: 500

Archive Server
Scan Frequency: 30 Sec

Purge Properties

On Success
● Purge After: 0 Days 4 Hours 0 Min.

On Error
● Purge After: 1 Days 0 Hours 0 Min.

On Error
● Purge After: 0 Days 2 Hours 0 Min.

Actions
□ Hold all running jobs □ Prevent new Edge processes

Figure 28: Scheduler Properties

- **Submit Server** – Specify the **Scan Frequency** of the Submit server. This value specifies how frequently the server will check if scheduled jobs need to initiate new jobs.

- **Active Server** – The Active server is responsible for status changes of all running (active) jobs.
  - **Scan Frequency** - Specifies the frequency the Active server will scan the running jobs and determine actions that need to be taken (i.e. move to archive, execute exit command, activate recovery, etc.).

  - **Max Jobs** - Specify the maximum number or simultaneously running jobs. If there's a job that is scheduled to run every minute that distributes data to multiple Edges, and the server is down, when the server is running again, all pending jobs will try to run at once. Limiting the number of jobs, will queue the rest of the jobs until one of the currently running jobs had completed.

- **Archive Server** - The Archive server is responsible for handling completed jobs. Configure here the Purge Policy depending on the completion status of a job.

  NOTE: When a job has reached its purge age, it will be deleted. A purged job is no longer viewable through either the Console CLI or the Console GUI.

  Each R-1 job generates report files that can be used for monitoring and debugging purposes. Since these reports can consume vast amounts of disk space, they are automatically purged after a predetermined amount of time. This means that they will disappear from the Console GUI, and not be visible by a Console CLI **`show`** command.

  The R-1 Scheduler Archive Server can be configured to have different purge policies for different job completion status'. For example, a job that exited with error may want to be kept much longer than a job that completed successfully. The **`Abort`** / **`Error`** / **`Warning`** / **`Success`** Purge Policy fields are in minutes, and will govern how long a job record will be kept after completion.

  By default, job reports will be kept like this:
  - Successful Jobs - 4 hours (240 minutes)
  - Aborted jobs - 2 hours (120 minutes)
  - Failed jobs - 24 hours (1440 minutes)

  - **Scan Frequency** - This value specifies how frequently the server will check if jobs are to be purged.

    As R-1 architecture is designed to be robust and resilient to failures, the purge command is performed in an asynchronous way. When the purge command is issued, the job is "only" marked to be purged, and the actual deletion is performed on a scheduled basis. This means that the user can issue a purge command, reboot the Center server before the jobs are actually deleted and when the server is started the purge operation will still take place.

  - **On Success / Error / Abort** – A different policy can be specified for each completion status.
    - **Purge After**- Specify the duration (Days/Hours/Minutes) in which to keep the Job's reports.

- **Actions –** These two administrative tasks are to be used on the Center or Edge prior to R-1 upgrades, or before other general maintenance activities that require the administrator to stop all processes on the servers.
  - **Hold all jobs** – Put on hold all running and scheduled jobs. This is an administrative task that will put on hold all Replication and Distribution jobs regardless of their originating user.

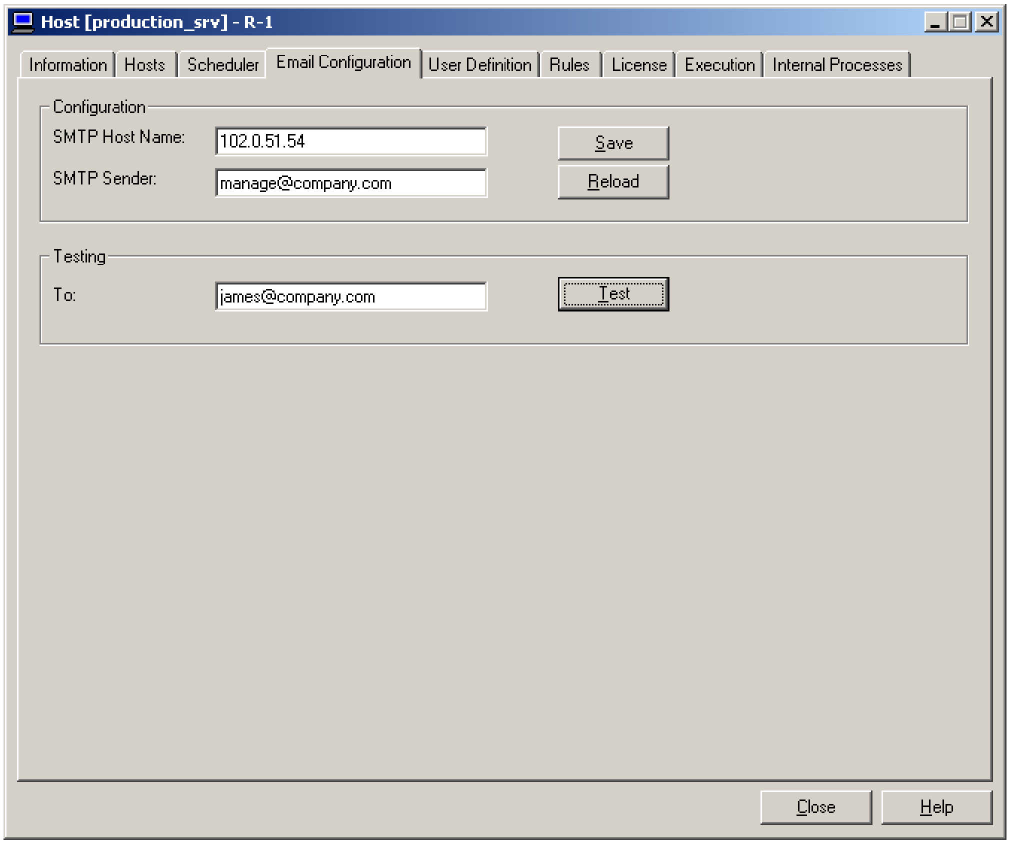
  - **Prevent new Edge processes** – Check this option to prevent new processes on the Edge (cleanup, snapshot, etc.) from being initiated. Note that running R-1 processes will not be stopped and will continue until completion but new processes will not be created (i.e. the Edge will not accept download or upload requests).

NOTE: None of the changes will take effect until the Save button is pressed. Only new jobs will be affected by these changes.

- **Save –** Save changes to the configuration files on the Center.
- **Undo All –** Undo changes done to the configuration files. Values are re-loaded from the Center.

3.6.4. Manage Email Configuration Tab

As R-1 can be setup to send email upon job completion, e-mail configuration parameters should be set on the R-1 Center.



SMTP Host Name: 102.0.51.54

SMTP Sender: manage@company.com

To: james@company.com

Figure 29: Email Configuration

- **SMTP Host Name** – Enter the IP Address or name of the Email server
- **SMTP sender** – Enter the Email address that appears in the *From* field of the mail. It does not necessarily have to be an actual account.
  Format is user@company.com.

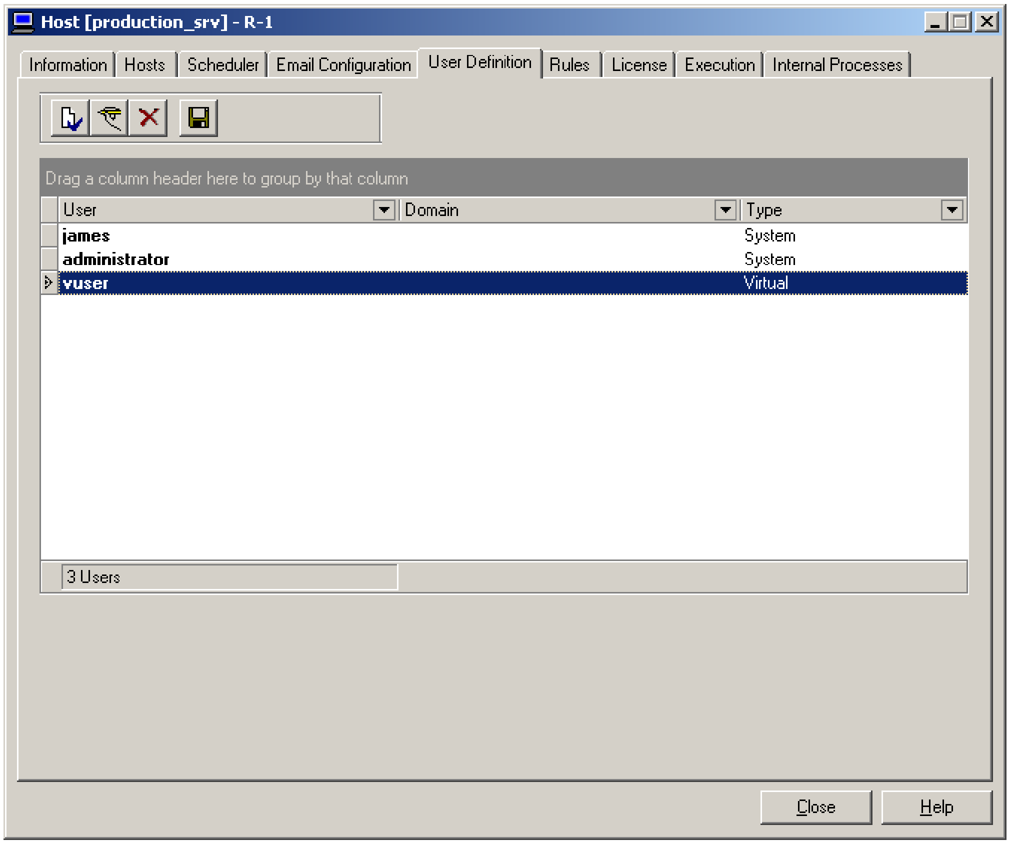
- **Save** – Save email configuration properties on the Center.
- **Reload** – Undo changes and re-load the Center configuration properties.

- **Testing** – Enter the Email address of a **recipient** to test the email configuration settings, and press the **Test** button. Upon successful email-sending, the user will be prompted to save the configuration settings, if those were not already saved.

3.6.5. User Definition Tab

The User Definition tab enables you to define real users and virtual users to convert them to real users. These users may be used for connection between the **Console** and **Center.**

Virtual users are defined on the machine accepting the connection. This way, the machine initiating the connection "knows" only virtual information and only virtual information is sent on the network. Real information never leaves the machine accepting the connection.



Drag a column header here to group by that column

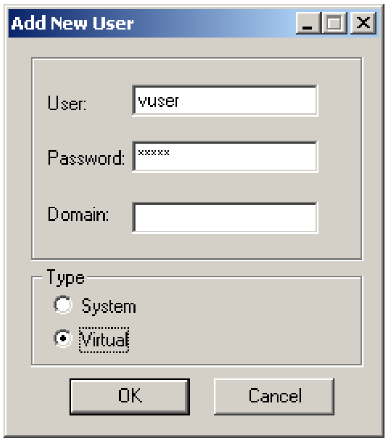
User	Domain	Type
james		System
administrator		System
vuser		Virtual

3 Users

Figure 30: Define Virtual Users

NOTE: In UNIX, you can only define virtual users. No need to define real users to be used in the Rules Tab. In Windows, in order to map a virtual user to a real user, the real user must be defined in the User Definition tab as well.

1. Click on the leftmost button to add a new user.



Add New User

User: vuser

Password: *****

Domain:

Type

○ System

● Virtual

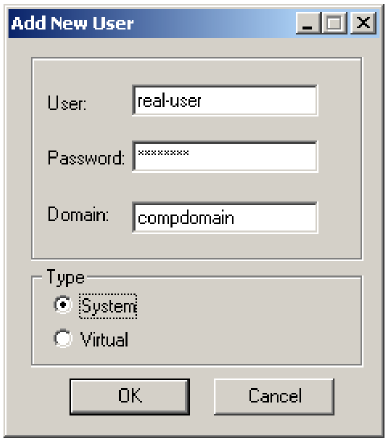
Figure 31: Adding Virtual Users

Add the virtual user and virtual password. Leave the domain field blank.

NOTE: There is no virtual domain. If you put something there it will look for that user in a real domain.

Check on the **Virtual** radio button and click OK.

2. Now add another user, this time the real user that the virtual user is translated to. Check the **System** radio button and fill a real Domain:



Add New User

User: real-user

Password: ********

Domain: compdomain

Type

● System

○ Virtual

Figure 32: Adding System Users

3. Hit the save button.

Updating the Submit Command

Assuming that you have defined virtual users on an **R-1 Edge**, you can go back to your **`submit command`** on the **Console**, and change the user/password combination for the remote connection:

Instead of:

```
      > r1 submit … -user=real-user –password=xxx –domain=compdomain
```

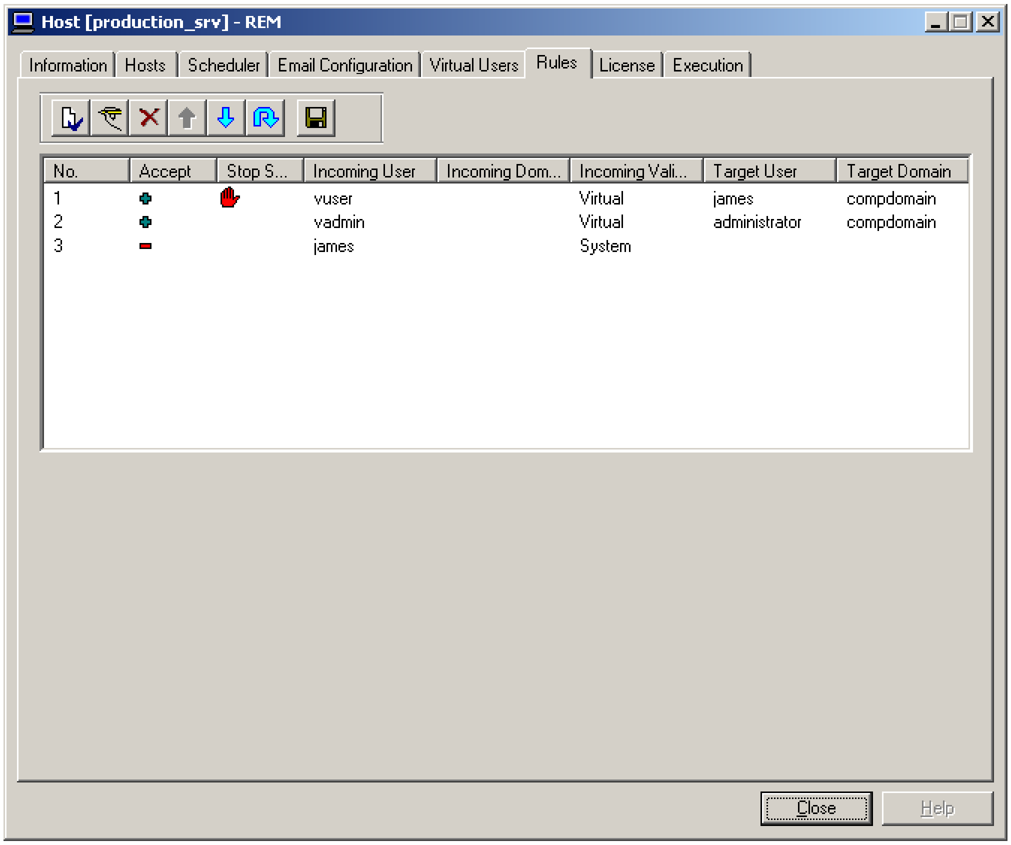
you can now specify:

```
      > r1 submit … -user=vuser –password=vpass   (← no domain)
```

3.6.6. Security Rules Tab

The Security Rules tab enables you to define rules that apply to system and virtual users. These rules apply for connecting from the **Console** to the **Center**.

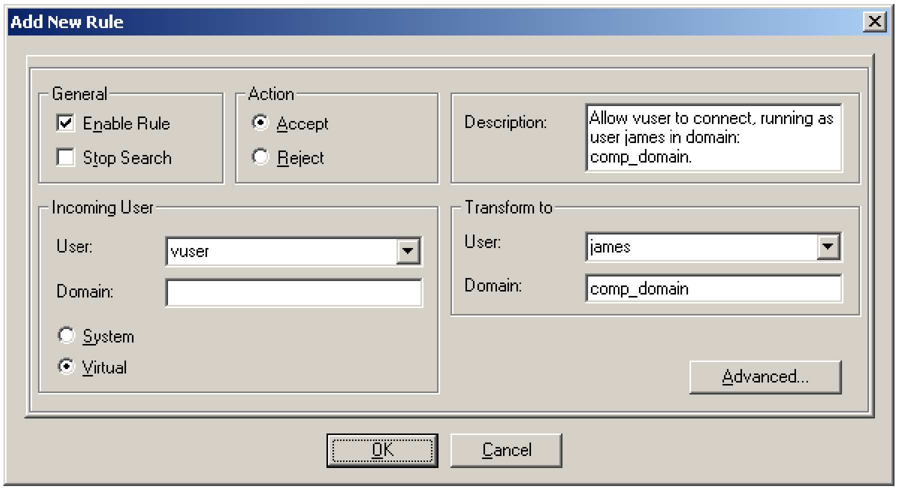
Security rules are defined on the machine accepting the connection, as this is where access to system and virtual users is granted or denied. Here you define the connection between the virtual user and the real user.



No.	Accept	Stop S...	Incoming User	Incoming Dom...	Incoming Vali...	Target User	Target Domain
1	✚	✋	vuser		Virtual	james	compdomain
2	✚		vadmin		Virtual	administrator	compdomain
3	▬		james		System		

Figure 33: Security Rules Tab

1. Click on the leftmost button “New Rule” to add a new rule –



Add New Rule

General
☑ Enable Rule
□ Stop Search

Action
● Accept
○ Reject

Description: Allow vuser to connect, running as user james in domain: comp_domain.

Incoming User
User: vuser
Domain:
○ System
● Virtual

Transform to
User: james
Domain: comp_domain

Figure 34: Adding a Rule

- **Enable Rule** – Specify whether this rule is enabled or not. If this option is not selected the rule is not taken into account while evaluating a user.
- **Stop Search** – Do not check that user using other rules. If the incoming user matches that rule, accept or deny according to that rule.
- **Action** - specify whether the rule is “positive” – approve the user to connect (**Accept**), or “negative” – deny access to that user (**Reject**).
- **Description** – Fill in a description of that rule. This field has no effect on the rule’s functionality.

- **Incoming User** - specify the user name performing the connection. This user name will be translated to a real user on the target machine. Specify whether the incoming user is **Virtual** or not.
  - If the incoming user is a virtual user, you may define a virtual **domain** as well. The domain will be then used in the actual connection command.

NOTE: In UNIX, you can only define virtual users. No need to define real users to be used in the Rules Tab. In Windows, in order to map a virtual user to a real user, the real user must be defined in the User Definition tab as well.

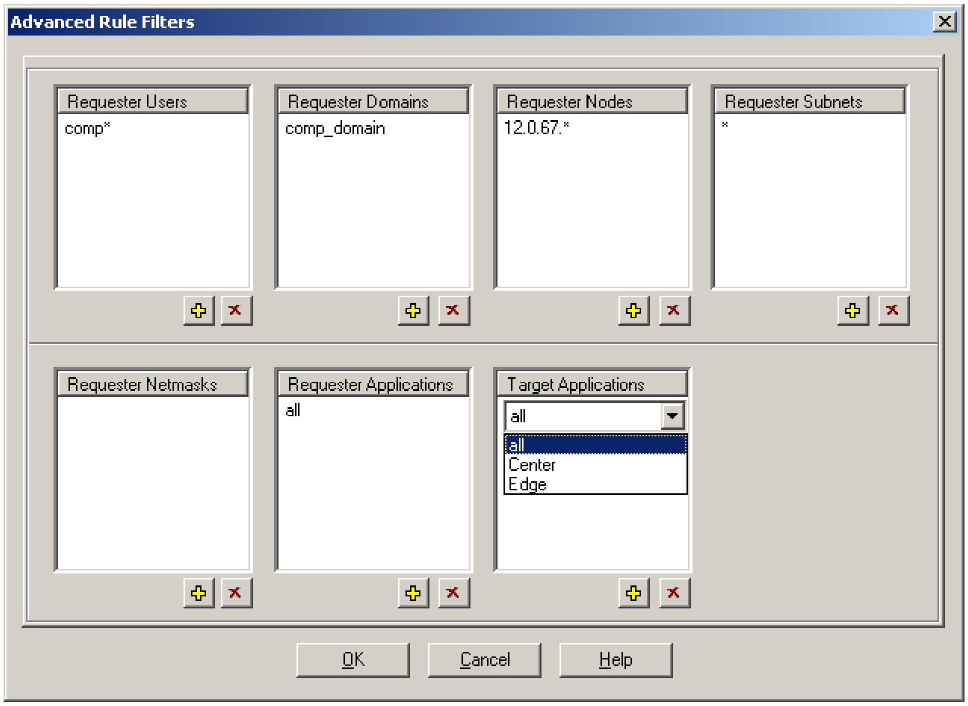
  - In a **Console → Center connection**, the incoming user is the one used in the Console Connection screen.
  - In a **Center → Edge connection**, the incoming user is the one used in the job definition.

- **Transform To**- specify the real target user the **Incoming User** is translated to.
  - If applicable, specify **Domain**.

NOTE: If the virtual user was defined with a domain (a virtual one), you should also specify that domain name here.

This is a basic rule. As we defined only one user, all you have to do is click OK to say that the user "*vuser*", will be examined as a virtual user and translated to the real user "james" from the domain "*comp_domain*".

- **Advanced** – Rules may be farther enhanced to tighten security and limit accessibility to R-1 machines. These features describe the operating system environment from which the connection request is coming from.



Advanced Rule Filters

Requester Users: comp*

Requester Domains: comp_domain

Requester Nodes: 12.0.67.*

Requester Subnets: *

Requester Netmasks:

Requester Applications: all

Target Applications: all

Figure 35: Advanced Rule Parameters

- ○ **Requester Users** - Wildcard and multiple values are supported.
    - ➢ In a **Console → Center connection**, this is the ACTUAL ACCOUNT from which the R-1 command came from, i.e. Windows or UNIX account, not the R-1 Console login.
    - ➢ In a **Center → Edge connection**, this is the user used in the Console login to the Center.
- ○ **Requester Domains** – Wildcard and multiple values are supported.
    - ➢ In a **Console → Center connection**, this is the ACTUAL Windows domain from which the R-1 command came from.

  - ➢ In a **Center → Edge connection**, this is the domain of the user used in the Console login to the Center.

- o **Requester Nodes** – The machine name from which the request comes from. Wildcard and multiple values are supported.

  - ➢ In a **Console → Center connection**, this is the network name of the Console machine.
  - ➢ In a **Center → Edge connection**, this is the network name of the Center.

- o **Requester Subnets** – The subnet from which the request comes from. Wildcard and multiple values are supported. 17.0.84.*- for example.

  - ➢ In a **Console → Center connection**, this is the subnet of the Console machine.
  - ➢ In a **Center → Edge connection**, this is the subnet of the Center.

- o **Requester Netmasks** – The netmask from which the request comes from. Wildcard and multiple values are supported. 255.255.255.0 - for example.

  - ➢ In a **Console → Center connection**, this is the netmask of the Console machine.
  - ➢ In a **Center → Edge connection**, this is the netmask of the Center.

- o **Requester Applications** – The application name from which the request comes from. Multiple values are supported, and ALL can be selected meaning the request can come from any of the listed applications. This field can have the following options:
  - ▪ R-1 Console
  - ▪ Command Line
  - ▪ API
  - ▪ Replication Job
  - ▪ Multicast Distribution

- o **Target Applications** – The target application receiving the connection request. This field can have the following options:
  - ▪ R-1 Center
  - ▪ R-1 Edge

In the screen-shot above, only users that their name begins with "*comp*", from domain "*comp_domain*", and that are connecting from machines whose IP address begins with *12.0.76* are allowed to connect. Other users will not pass that rule and their connection request will be denied.

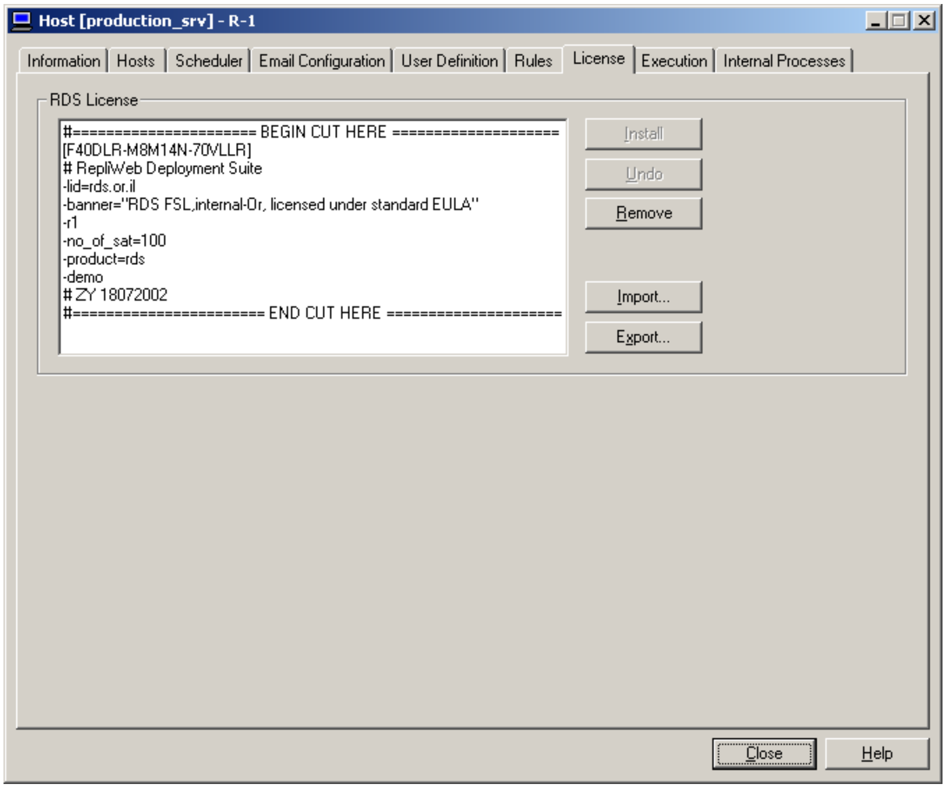
2. Hit the save button

3. Defining a few rules, you can determine the order in which the rules are applied when a user tried to connect to the machine. To update the location of a rule, right-click the rule, and in the floating menu select the **Move** option. A window will open, enabling to move the rule up & down between the existing rules.

3.6.7. License Tab

This window enables managing R-1 Licenses on the Center.

NOTE: Handle R-1 licenses with care. Follow the instruction received from RepliWeb support.



```
#====================== BEGIN CUT HERE =====================
[F40DLR-M8M14N-70VLLR]
# RepliWeb Deployment Suite
-lid=rds.or.il
-banner="RDS FSL,internal-Or, licensed under standard EULA"
-r1
-no_of_sat=100
-product=rds
-demo
# ZY 18072002
#======================== END CUT HERE =====================
```

Figure 36: Licensing

- **Remove** – Remove the license from the remote machine.

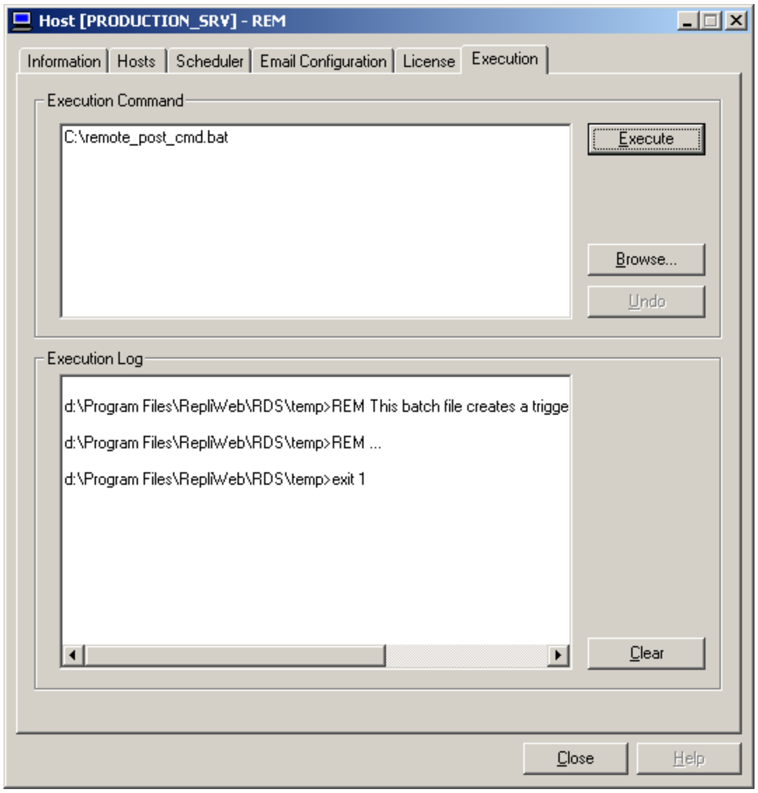
NOTE: R-1 is a licensed application. Removing the license without direct instructions from RepliWeb support will disable the applications.

- **Install** – Save the viewed license on the remote host. Use the license received from Repliweb. Either copy the license from the email and paste it in the license window, or **import** a saved license from the Console machine.
- **Undo –** Undo changes done in the license window, reload license from the remote Host.
- **Export –** Export the license in the window to a file on the Console.

NOTE: R-1 Licenses are digitally signed. Do not make any changes to the text.

3.6.8. Management Execution Tab

The tab enables executing a batch file or an executable on the Center. This screen may be used to test pre/post replication commands that should be run on the Center before or after the replication process.



Execution Command: C:\remote_post_cmd.bat

```
d:\Program Files\RepliWeb\RDS\temp>REM This batch file creates a trigge
d:\Program Files\RepliWeb\RDS\temp>REM ...
d:\Program Files\RepliWeb\RDS\temp>exit 1
```

Figure 37: Remote Execution

- **Execution Command** – Type the name of the file to execute on the Center, or use the **Browse** button to open a file dialog on the remote Center, enabling selection of the file to execute. The executable's output is displayed in the **Log** window.

- **Undo –** Cancel changes done inside the Execution Command text-box.

- **Execute** – Press the button after filling in the name of the file to execute.

NOTE: The file must reside on the remote Center.

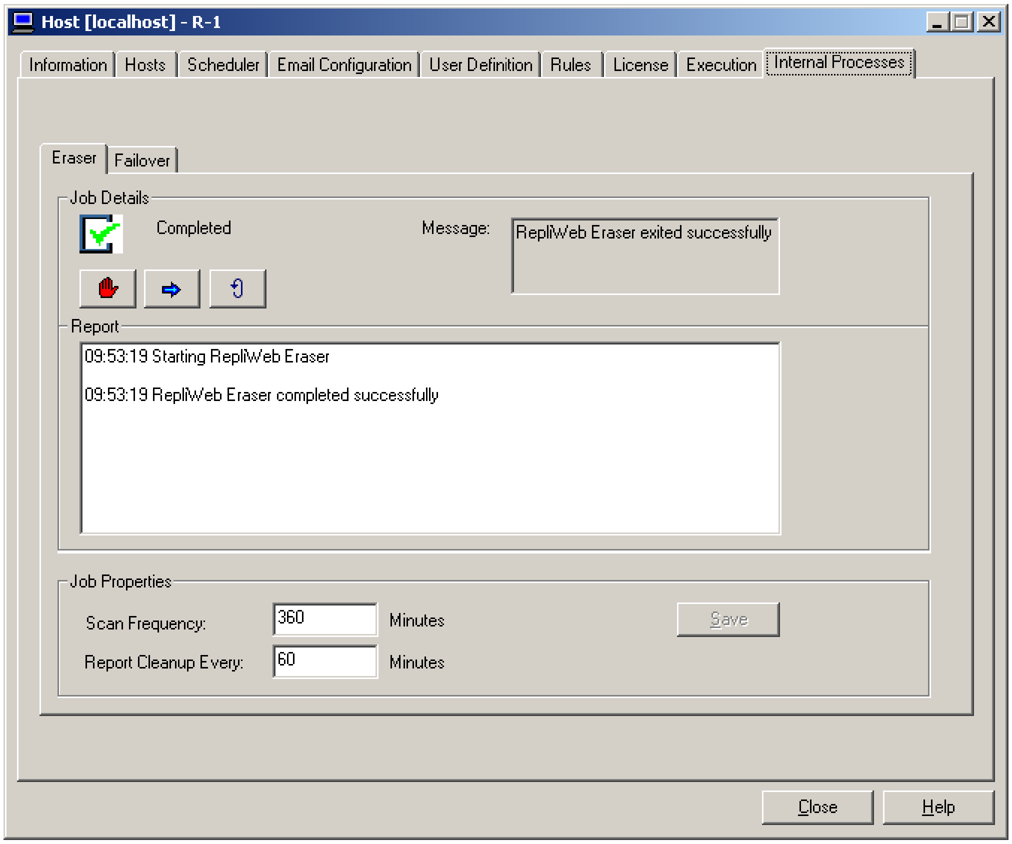
- **Clear** – Clears the text from the Log window.

3.6.9. Manage Internal Processes Tab

R-1 uses internal jobs to perform various system operations.

3.6.9.1. Internal Processes - Eraser

The Eraser job is responsible for purging recorded Rollback and other data that has expired. In this tab the user can monitor the Eraser job, view its report and set its properties. The expiration date of recorded data is set in the **Rollback Job Properties / Purge** tab.



Completed

Message: RepliWeb Eraser exited successfully

```
09:53:19 Starting RepliWeb Eraser
09:53:19 RepliWeb Eraser completed successfully
```

Scan Frequency: 360 Minutes

Report Cleanup Every: 60 Minutes

Figure 38: Internal Process - Eraser

- **Job Details** – View the Eraser job's state, completion message and report. As the Eraser job is an internal process, available operations are **Hold** ✋ and **Continue** ➡ only. Users can also request to refresh the displayed information by clicking **refresh** ↻.

- **Job Properties** –
    - **Scan Frequency -** Specify in minutes how often the Scheduler will submit an Eraser job. This affects how frequently the Eraser will check if expired data needs to be erased.

    - **Report cleanup every –** The Eraser report can be cleaned up at specified intervals. This value is in minutes.

    - **Save** – Save eraser job properties on the Center.

3.6.9.2. Internal Processes - Failover

The Failover process enables R-1 administrators to maintain a Backup (standby) server that will be used as the R-1 Center in case the Primary Server is temporarily shut down.

Failover is an important fault tolerance function of mission-critical systems that rely on constant accessibility. The backup server is automatically and transparently to the user being updated with all R-1 Center files, so when the need arises, it can replace the primary server and act as an R-1 center, running all replication and distribution jobs as before.

NOTE: The Failover Server must contain a complete R-1 installation, installed on **exactly** the same path as the R-1 Center initiating the Failover process. Install at least the Center and Edge components; the Console is optional. This will allow the failover server to act as both an R-1 Center and R-1 Edge for existing and new jobs.

The Failover and Primary server should be of the same platform (Windows ←→ Windows, Linux ←→ Linux, etc.).

The Failover process will fail if these requirements are not met.

The Failover process updates the Failover server with the following R-1 information:

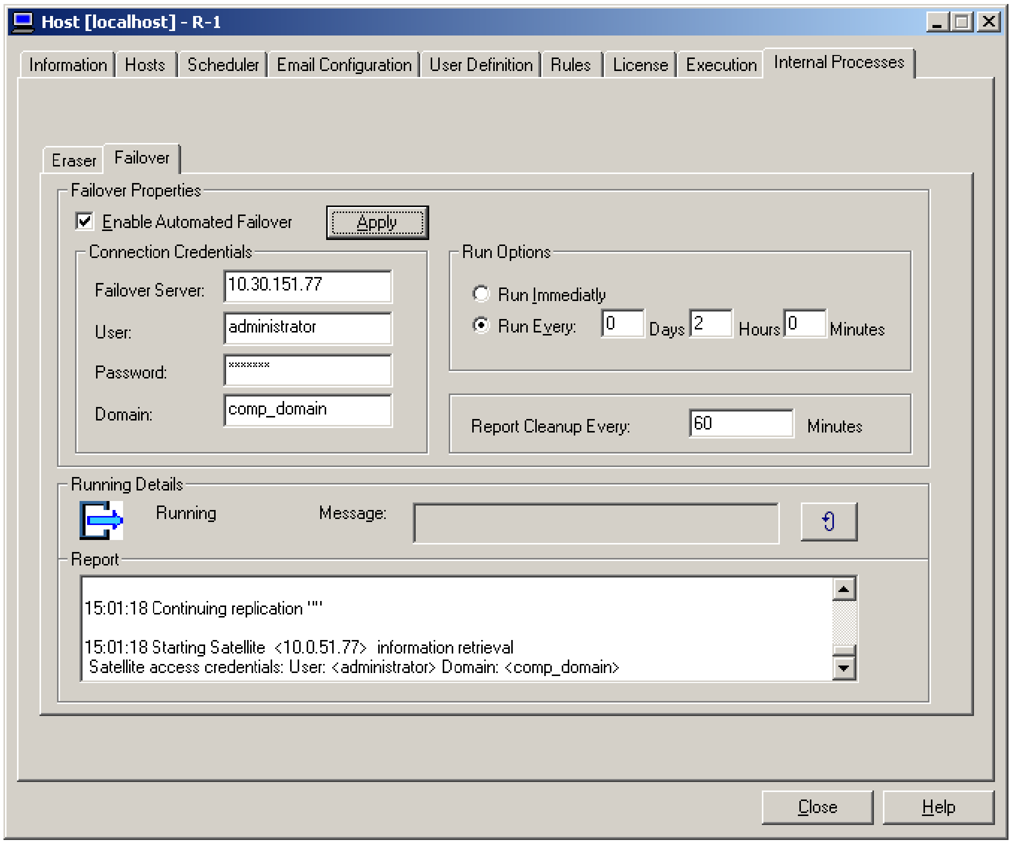
- **Scheduled Jobs** - This includes all continuous update jobs, scheduled jobs, on-demand jobs
- **Templates** – Common and User Specific
- **Containers** – Common and User Specific
- **Configuration Files**
- **Preference Files**

NOTE: The Failover process does NOT copy any content from the Primary server to the Failover server. This is the responsibility of the user/administrator.

Process Description Setup

1. Install R-1 on both Primary and Failover server. Make sure at least the Center and Edge components are installed. Both servers should be installed on EXACTLY the same path on both servers. The Failover and Primary server should be of the same platform (Windows ←→ Windows, Linux ←→ Linux, etc.).
2. When the Failover process is enabled on the Primary server, the R-1 Scheduler service is automatically stopped at the Failover server.
3. The Failover process copies scheduled job definitions, Templates, Containers and all configuration files required for the Failover server to act as a Primary server when needed. The Failover does NOT copy content from the production server to the Failover server.

4. When the Primary server goes down, the **user** should perform the following:
   a. Start the R-1 Scheduler service at the Failover server
   b. The Failover server should get the IP / DNS of the primary server. This should be done according to corporate policy to allow previously scheduled jobs to continue uninterrupted without resubmitting.



Failover Properties

☑ Enable Automated Failover

Connection Credentials

Failover Server: 10.30.151.77
User: administrator
Password: *******
Domain: comp_domain

Run Options

○ Run Immediatly
● Run Every: 0 Days 2 Hours 0 Minutes

Report Cleanup Every: 60 Minutes

Running Details

Running Message:

Report

```
15:01:18 Continuing replication ""

15:01:18 Starting Satellite  <10.0.51.77>  information retrieval
 Satellite access credentials: User: <administrator> Domain: <comp_domain>
```

Figure 39: Internal Process - Failover

- **Failover Properties** – Set the Failover process properties.
  - Press **Apply** to save changed settings. If the Failover is enabled, saved settings take effect immediately.

  - **Enable Automated Failover** – Selecting this option immediately initiates the Failover process.

    To stop the Failover process, un-check the field and press the Apply button.

  - **Connection Credentials** – Enter the credentials that will be used by the Failover process to connect to the Failover Server.
    - **Failover Server** - Enter the host name or IP address of the R-1 Center being used as the Failover Server.

        - **User** – Enter the user name that the Failover Process will use.
        - **Password** - Enter the password for the account specified in the **User** field.
        - **Domain** - If **user** is part of a domain, enter the domain here, if it is not, leave this field blank.

    - **Run Options –** Specify how the Failover process will run:
        - **Run Immediately** - The process will run immediately if the **Enable Automated Failover** is selected.
        - **Run Every** - Run a job at a specific time interval (days / hours / minutes).

    - **Report Cleanup** - The Failover report can be cleaned up at specified intervals. This value is in minutes.

- **Running Details** – View the Failover job's state, completion message and report.

  To refresh the displayed information clicking **refresh** ↻.

3.7. Rollback Job

A **Rollback** job may be submitted for jobs that had the **Use Rollback** option checked in the **Rollback** tab. When the job is completed (successfully or not) using the right mouse menu or the main menu you can select to recover previously recorded data. The Rollback Job allows point-in-time recovery capabilities.

A rollback is similar to an undo function. It enables you to return content to its previously replicated state. Rollbacks are performed on target machine. You cannot roll back content on the source.

For **Scheduled** and **Continuous Updates** jobs, you can rollback to instances of the job. In **Distribution** jobs, you can roll back the complete distribution process (for all Edges), or roll back individual Edges from the **Distribution Details** screen.

Rolling back, (recovering data) is performed by selecting a point-in-time to return to, in addition to various job properties that can be set to define the rollback process. This includes standard job properties like pre/post commands, recovery options, etc.

NOTE: Jobs that use the Rollback functionality can be rolled-back only after the job completes (successfully or not). In order to rollback a job, the job cannot be running. To rollback an instance of a **Scheduled** or **Continuous Updates**, the scheduled job itself needs to be aborted; it is not sufficient that the instance is completed.

3.7.1. Rollback Replication Jobs

Replication jobs (Upload / Download) can be rolled back to recover the target before the replication took place.

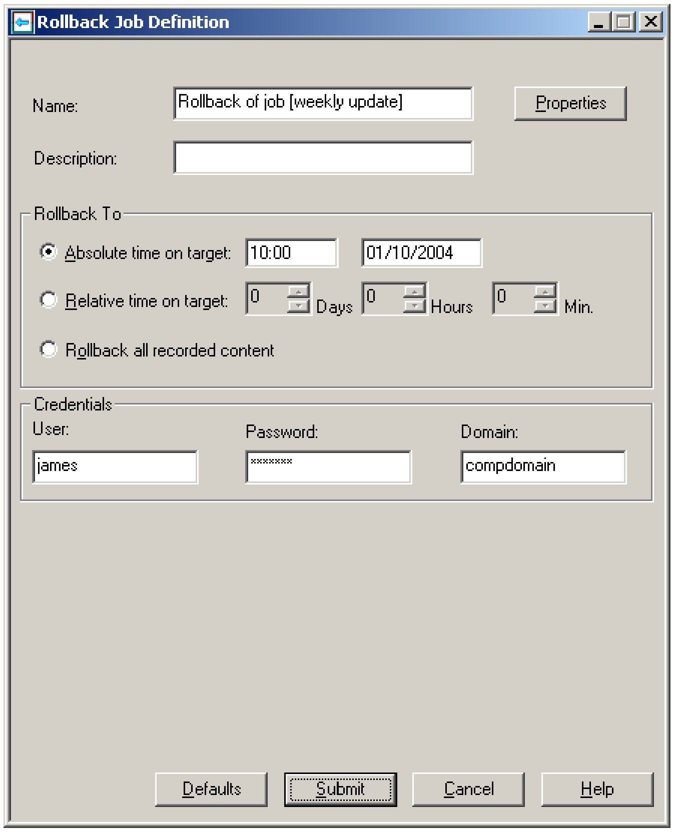
For **Scheduled** and **Continuous Updates** jobs, you can rollback to instances of the job, at a given point-in-time. Note that in that case the Scheduled or the Continuous Updates job itself needs to be aborted and not just the instance.

A job can roll back to a point in time only once. Once recorded data was successfully rolled back, it gets deleted, so that only previous recordings of that job are kept, and can be rolled back to. If the rollback process failed, the recorded data is kept and rollback can be attempted again.

3.7.1.1. Replication Rollback Definition

NOTE: The fields in the windows are automatically filled according to the job being rolled back, even if Rollback defaults were saved.

- **Name** – Enter the name of the Rollback job.
- **Description** – Enter the job's description.



Name: Rollback of job [weekly update]

Description:

Rollback To

● Absolute time on target: 10:00 01/10/2004

○ Relative time on target: 0 Days 0 Hours 0 Min.

○ Rollback all recorded content

Credentials

User: james

Password: *******

Domain: compdomain

Figure 40: Define the Rollback Job

- **Rollback To** – Specify the date and time to recover from. This can be an absolute or a relative time.

NOTE: The time specified is calculated at the Target machine (Edge in case of an Upload job, Center if it's a Download job). It is never the Console time.

  - **Absolute time on target** – Specify the exact time to roll back to, in HH:MM DD/MM/YYYY format (hours:minutes day/month/year).

In case a rollback instance is not found at the exact time specify, R-1 will roll back to the nearest recorded data, without passing the time specified.

  - **Relative time on target** – Specify how long to roll back to, in days / hours / minutes. Here, too, the calculated time to roll back from is calculated on the target machine.

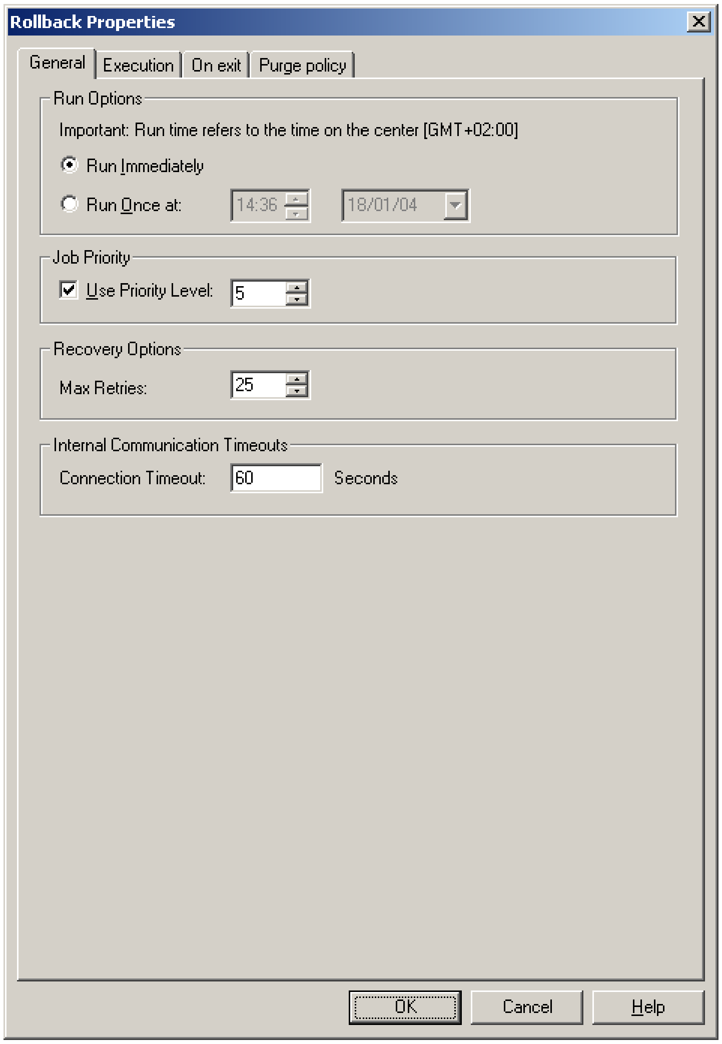
  - **Rollback all recorded content** – Using this option, the rollback process will bring the target directory to its state before the replication took place.

NOTE: For non-scheduled Replication jobs, this is the only option available.

- **Credentials** – Enter the credentials of the Rollback job. By default, these are the credential of the original job being rolled back.
  - **User** – Enter the username of the job performing the rollback.
  - **Password** – Enter the password for the account specified in the **User** field.
  - **Domain** – If the Target platform is Windows, the target Domain may also need to be specified.

NOTE: If you are using different credentials than those used in the original job, make sure there are sufficient access rights to the recording location and the target directory.

3.7.1.2. Replication Rollback General Tab



Run Options

Important: Run time refers to the time on the center [GMT+02:00]

● Run Immediately

○ Run Once at: 14:36 18/01/04

Job Priority

☑ Use Priority Level: 5

Recovery Options

Max Retries: 25

Internal Communication Timeouts

Connection Timeout: 60 Seconds

Figure 41: General Properties of a Rollback Job

- **Scheduling** – Specify whether the Rollback job will run immediately or at a later time.
  - **Run Immediately** – The Rollback job will run immediately following submission.
  - **Run Once At** - Run the Rollback job once on a specific (future) date at a specific time. Default time is the current date and time.

- **Use Priority Level** – **Priority** is a number between 0-10 to define the Rollback Job's priority (where 0 is the lowest priority and 10 is the highest). When the job is submitted, all jobs with lower priority are held. When the job finishes, the jobs are resumed and continue to run. The default priority is 5 (medium priority).

  Uncheck the field to run jobs that will be ignored while calculating priorities. Jobs like that will not cause lower-priority job to hold, and will not be held when jobs with a higher priority are submitted.

- **Recovery Options** – The Rollback process can recover after unexpected errors such as communication line or network faults, user interruption, or even a system crash. In anticipation of these errors, the Center maintains an ongoing record of its progress through each distribution job.

  Like in Replication and Distribution jobs, R-1 obtains all necessary information regarding the failed operation from the record generated, including the original property configurations and the exact point of failure.

  - **Max Retries** - The maximum number of recovery attempts.

NOTE: In case the Rollback job fails, the recorded data is kept, so future Rollback attempts can be made.

- **Internal Communication Timeouts -** R-1 enables controlling TCP/IP communication timeouts, by using Application Timeouts. Setting these fields will override the operating system TCP timeouts. Value 0 means using the operating system timeouts. Specify here overall R-1 communication timeouts. Timeouts can also be set for each of the transport engine separately for the transfer stage.

  - **Connect -** Specify, in seconds, the TCP connect session timeout for general communication within R-1 components. Default is 60 seconds.

3.7.2. Rollback Distribution Jobs

Distribution jobs can be rolled back to recover the target before the distribution process took place.

For **Scheduled** and **Continuous Updates** jobs, you can rollback to instances of the job, at a given point-in-time. Note that in that case, the Scheduled or Continuous Updates job itself needs to be aborted and not just the instance.

A job can roll back to a point in time only once. Once recorded data was successfully rolled back, it gets deleted, so that only previous recordings of that job are kept, and can be rolled back to. If the rollback process failed, the recorded data is kept and rollback can be attempted again.

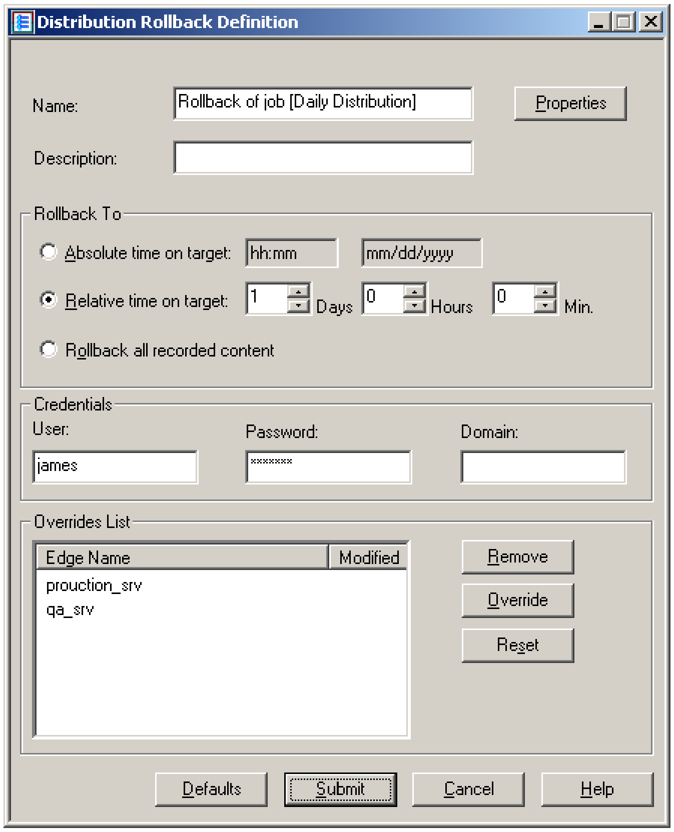
Rolling back a distribution job allows rolling back all Edges or only some of them.

3.7.2.1. Distribution Rollback Definition

NOTE: Some of the fields in the windows are automatically filled according to the job being rolled back, even if Rollback defaults were saved.

Rolling back a distribution job allows rolling back all Edges or some of them. By default, all Edges are rolled back. Edges not requiring rollback can be removed from the Edges list.

- **Name** – Enter the name of the Rollback job.
- **Description** – Enter the job’s description.



Name: Rollback of job [Daily Distribution]

Description:

Rollback To

○ Absolute time on target: hh:mm mm/dd/yyyy

● Relative time on target: 1 Days 0 Hours 0 Min.

○ Rollback all recorded content

Credentials

User: james

Password: *******

Domain:

Overrides List

Edge Name	Modified
prouction_srv	
qa_srv	

Figure 42: Define Distribution Rollback Job

- **Rollback To** – Specify the date and time to recover from. This can be an absolute or a relative time. This will apply to all Edges in the Distribution Rollback job. You cannot specify different times to different Edges.

NOTE: The time specified is calculated at the Target machine (Edge in case of an Upload job, Center if it's a Download job). It is never the Console time.

  - **Absolute time on target** – Specify the exact time to roll back to, in HH:MM DD/MM/YYYY format (hours:minutes day/month/year).

    In case a rollback instance is not found at the exact time specify, R-1 will roll back to the nearest recorded data, without passing the time specified.

  - **Relative time on target** – Specify how long to roll back to, in days / hours / minutes. Here, too, the calculated time to roll back from is calculated on the target machine.

  - **Rollback all recorded content** – Using this option, the rollback process will bring the target directory to its state before the replication took place.

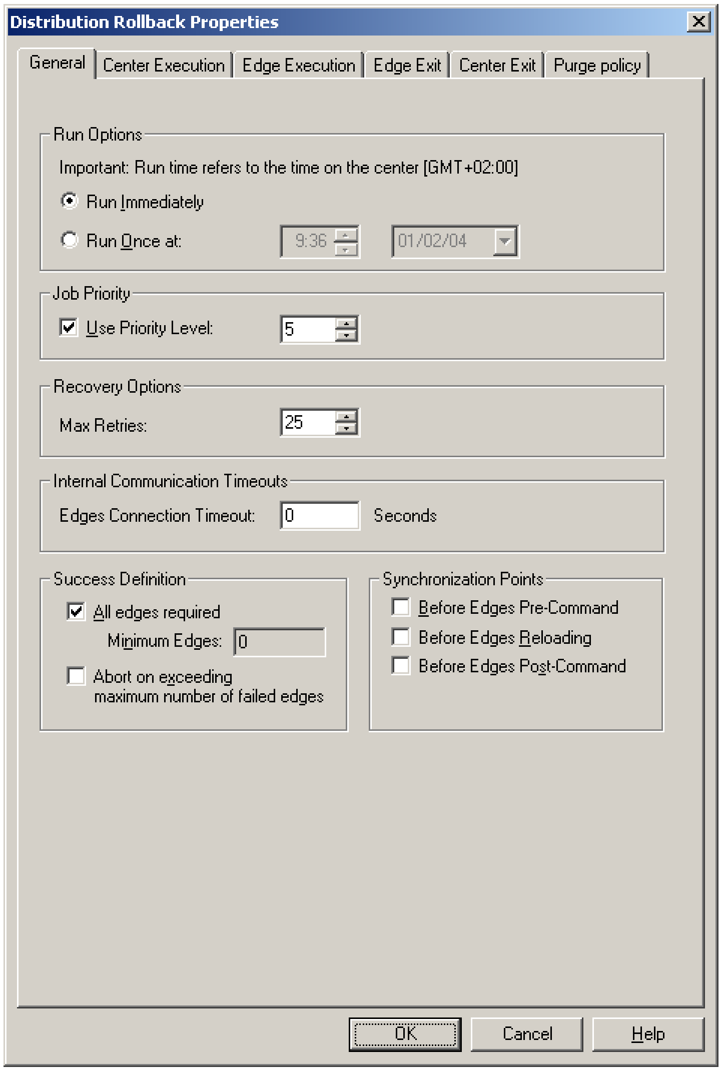
NOTE: For non-scheduled Replication jobs, this is the only option available.

- **Credentials** – Enter the credentials of the Rollback job. By default, the credentials used by each of the Edges are used.
    - **User** – Enter the username of the job performing the rollback.
    - **Password** – Enter the password for the account specified in the **User** field.
    - **Domain** – If the Target platform is Windows, the target Domain may also need to be specified.

NOTE: If you are using different credentials than those used in the original job, make sure there are sufficient access rights to the recording location and the target directory.

3.7.2.2. Distribution Rollback General Tab

The Edge properties set in this tab, are the default Edge settings for the Rollback job. To override these settings for a specific Edge, open the **Edge Override** dialog.



Run Options

Important: Run time refers to the time on the center [GMT+02:00]

● Run Immediately

○ Run Once at: 9:36 01/02/04

Job Priority

☑ Use Priority Level: 5

Recovery Options

Max Retries: 25

Internal Communication Timeouts

Edges Connection Timeout: 0 Seconds

Success Definition

☑ All edges required

Minimum Edges: 0

□ Abort on exceeding maximum number of failed edges

Synchronization Points

□ Before Edges Pre-Command

□ Before Edges Reloading

□ Before Edges Post-Command

Figure 43: Rollback General Tab

- **Scheduling** – Specify whether the Rollback job will run immediately or at a later time.
  - **Run Immediately** – The Rollback job will run immediately following submission.
  - **Run Once At** - Run the Rollback job once on a specific (future) date at a specific time. Default time is the current date and time.

- **Use Priority Level** – **Priority** is a number between 0-10 to define the Rollback Job's priority (where 0 is the lowest priority and 10 is the highest). When the job is submitted, all jobs with lower priority are held. When the job finishes, the jobs are resumed and continue to run. The default priority is 5 (medium priority).

Uncheck the field to run jobs that will be ignored while calculating priorities. Jobs like that will not cause lower-priority job to hold, and will not be held when jobs with a higher priority are submitted.

- **Recovery Options** – The Rollback process can recover after unexpected errors such as communication line or network faults, user interruption, or even a system crash. In anticipation of these errors, the Center maintains an ongoing record of its progress through each distribution job.

  Like in Replication and Distribution jobs, R-1 obtains all necessary information regarding the failed operation from the record generated, including the original property configurations and the exact point of failure.

  - **Max Retries** - The maximum number of recovery attempts.

NOTE: In case the Rollback job fails, the recorded data is kept, so future Rollback attempts can be made.

- **Internal Communication Timeouts -** R-1 enables controlling TCP/IP communication timeouts, by using Application Timeouts. Setting these fields will override the operating system TCP timeouts. Value 0 means using the operating system timeouts. Specify here overall R-1 communication timeouts. Timeouts can also be set for each of the transport engine separately for the transfer stage.

  - **Connect -** Specify, in seconds, the TCP connect session timeout for general communication within R-1 components. Default is 60 seconds.

- **Success Definition** – It may happen that not all Edges are available at the execution time. Define here the number of Edges required for the successful completion of the Rollback Job and when the Distribution Rollback job determined its completion status.

  - **All Edges Required** – Successful distribution to all Edges is required for a successful completion of the Distribution Rollback Job.

  - **Minimum Edges** – If distribution to the minimum number of Edges succeeded, the Rollback Job succeeds.

  - **Abort on exceeding maximum number of failed edges** – This optimization option specifies **when** the Rollback process determines its completion status.

    By default, once an Edge job fails, the Rollback process evaluates its completion status based on the **Minimum Edges** or **All Edges Required** options, as specified by the user. If the conditions are not met, the Rollback job will fail.

When this option is used, if one or more of the Edge jobs fail, and the job will not meet the minimum of number of edges criteria, the Distribution Rollback job will abort immediately all its running Edge jobs.

For example, assume you defined a Rollback job from 10 Edges, and specified the **Success Definition** to minimum of 6 Edge jobs. If 5 Edge jobs fail, it is obvious that the Rollback job will fail, as it will not meet its success definition. Instead of waiting for the other 5 Edge jobs to complete (successfully or not), the Rollback job can abort immediately these 5 Edge jobs and determine its completion status- as failed.

If the success definition requires that all Edge jobs complete successfully, one failed job is sufficient to determine the completion status of the job and the Rollback job will abort immediately.

- **Synchronization Points** – The Center and Edge(s) components may synchronize at certain points of the job execution. All processes will hold, until all Edges reach the specified breakpoint before continuing. More than one breakpoint may be selected:
  - **Before Edges Pre-Commands**
  - **Before Edges Reloading**
  - **Before Edges Post-Commands**

3.7.3. Common Rollback Tabs

The following tabs are used when performing rollback for both Replication and Distribution jobs.

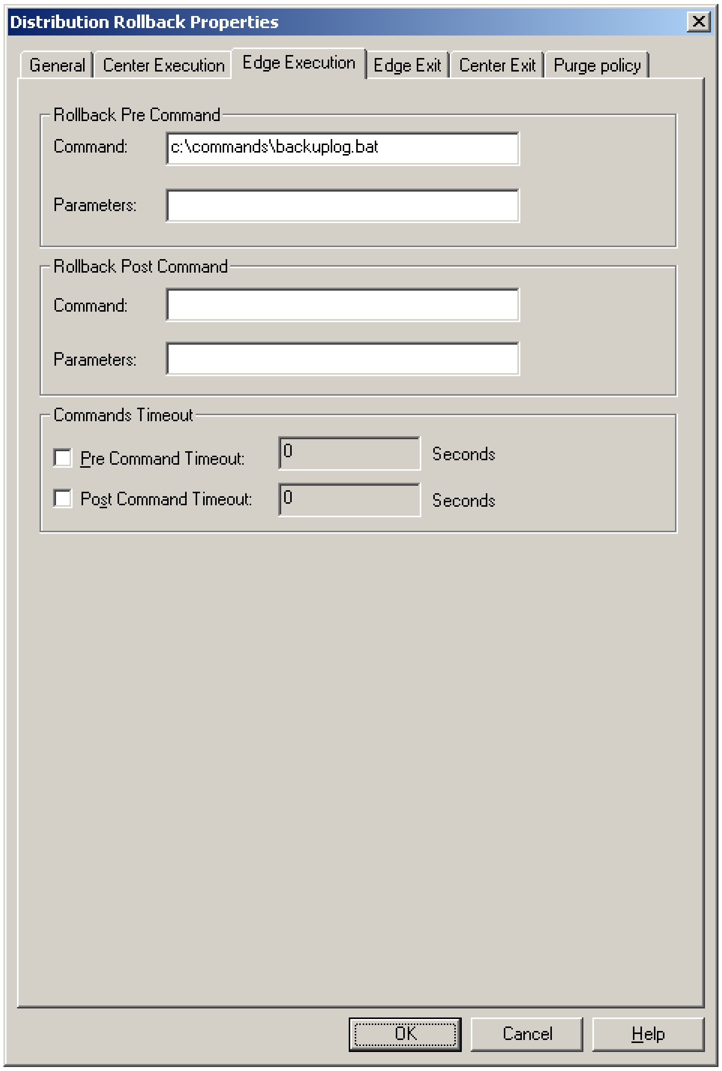
3.7.3.1. Rollback Execution Tab

The execution option enables the user to specify commands to run before / after the Rollback operation. These commands will run on the Target machine.

NOTE: In Upload jobs, the commands will run on the Edge. In Download jobs, the commands will run on the Center.

In the **Center Execution** Tab, specify the pre/post commands that will run on the Center.

In the **Execution / Edge Execution** Tabs, specify the pre/post commands that will run on the Edges.



Rollback Pre Command

Command: c:\commands\backuplog.bat

Parameters:

Rollback Post Command

Command:

Parameters:

Commands Timeout

☐ Pre Command Timeout: 0 Seconds

☐ Post Command Timeout: 0 Seconds

Figure 44: Rollback Pre/Post Commands

- **Rollback pre-command**– The command will be executed prior to the Rollback operation.
    - In a **replication** job, the target can be either the Center or the Edge.
    - In a **distribution** job, specify pre commands that will run on the Center in the Center Execution Tab, and Pre commands that will run on the Edge (target) in the Edge Execution tab.
- **Rollback post command** – The command will be executed after a successful execution of the Job.
    - In a **replication** job, the target can be either the Center or the Edge.
    - In a **distribution** job, specify pre commands that will run on the Center in the Center Execution Tab, and Pre commands that will run on the Edge (target) in the Edge Execution tab.

Valid **Command** entries are the full path to any single batch file or script that can be run from a command-line. Valid **Parameters** entries are any switches, qualifiers, etc., that the script or batch file accepts. Multiple parameters must be separated by a single space.

```
<command> <variable1> <variable2>
```

Where `<variable1>` and `<variable2>` from the batch file refer to user-entered values in the **Parameters** field(s).

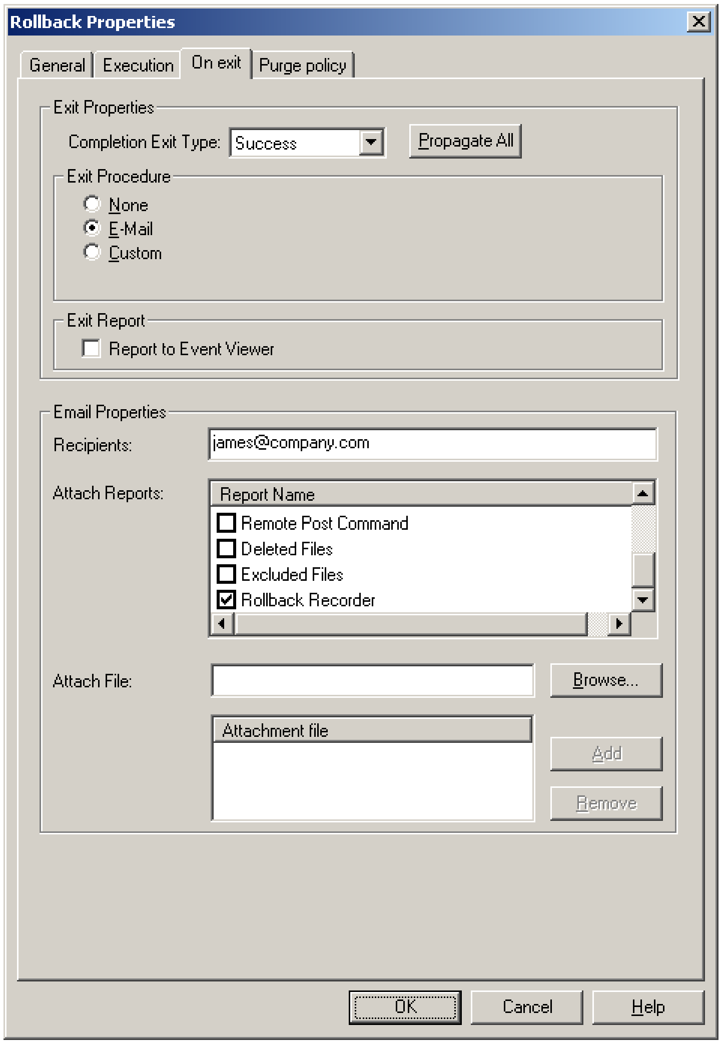
- **Pre Commands Timeout** – Check here to specify the time to wait (in seconds) until the pre command will execute. If the timeout have passed, the execution of the pre-command fails.
- **Post Commands Timeout** – Check here to specify the time to wait (in seconds) until the post command will execute. If the timeout have passed, the execution of the pre-command fails.

3.7.3.2. Rollback On Exit Tab

The **On Exit** option allows the linking of procedures to the 'exit' status of the rollback job. These procedures will be executed only upon the rollback job completion. A failed attempt, that is, a job entering a **Recovery** state does not constitute a completed job.

In the **On Exit / Center Exit** Tab, specify the procedures to be executed on the Center upon completion of the submitted job.

In the **Edge Exit** Tab, specify the procedures to be executed on the Center, upon completion of an Edge job.



Completion Exit Type: Success

○ None
● E-Mail
○ Custom

□ Report to Event Viewer

Recipients: james@company.com

Attach Reports:

Report Name
□ Remote Post Command
□ Deleted Files
□ Excluded Files
☑ Rollback Recorder

Attach File:

Attachment file

Figure 45: Rollback Exit – Email

The **On Exit** feature allows the user to design procedures to be followed at the end of the rollback job.

- **Completion Exit Type** – Select the exit type from the drop down list, to define the notification method for that exit condition.
    - **Abort** - Define operation to perform in case the job aborts.
    - **Success** - Define operation to perform in case the job end successfully.
    - **Error** - Define operation to perform in case the job fails.

- **Exit procedure** – For each completion type, select the method used for notification of job completion by choosing one of the following:
    - **None** - No exit procedures specified.
    - **E-mail** - An email is sent upon completion of the job.
        - **Recipient** – Enter the e-mail address of the e-mail recipient(s). More than one e-mail address may be entered. Multiple e-mail addresses should be separated by commas.
        - **Include Reports**– Select the replication reports to be attached to the E-mail. For more information about the available reports refer to **Appendix M – Reports**.
        - **Attachments -** Files may also be attached to e-mails by specifying their full path. More than one file may be attached.

    NOTE: To use the e-mail notification feature be sure that the appropriate SMTP fields were completed during the installation procedure. If these fields were left blank, refer to the **Manage – Email Configuration Tab** to correctly set-up e-mail notification.

    - **Custom -** User defined procedures to be followed on exit from a job. These procedures may require user-defined parameters as well. The **Custom** exit feature allows integration with any other Command Line applications. For detailed instructions please consult the section on Pre / Post distribution commands.

NOTE: The **File List** option is not available for Rollback jobs.

- **Propagate All** – Use the same specified procedure for all Exit functions.

- **Report to Event Viewer** – For each completion type, report the Job's exit message to the Windows Event Viewer or the UNIX syslog.

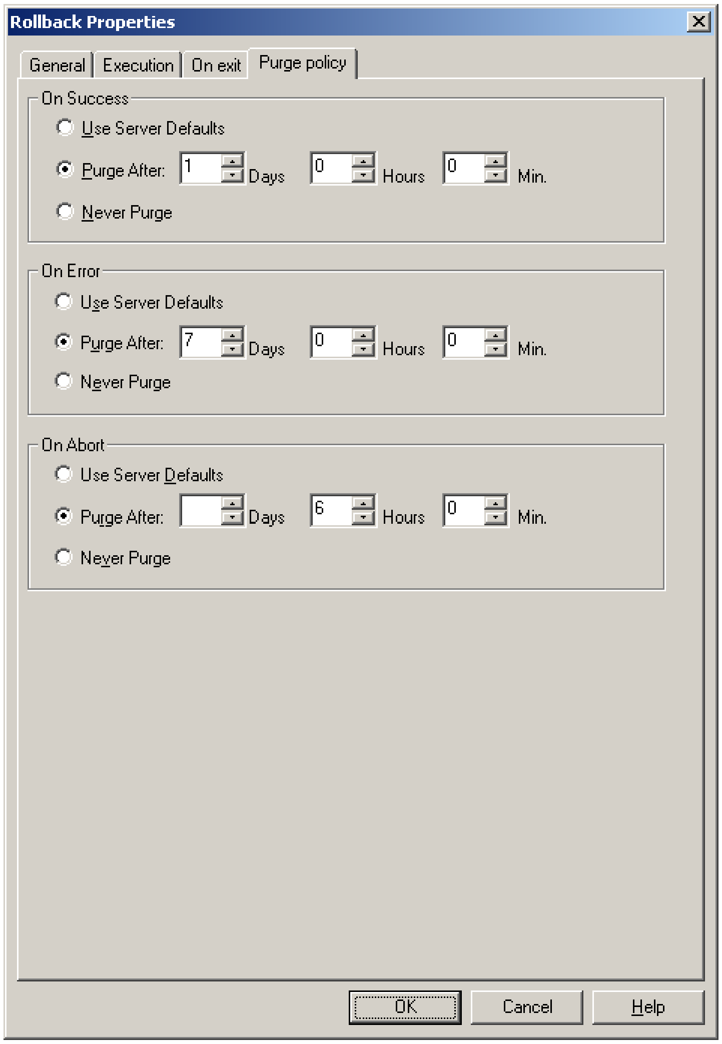
  For more information refer to **Appendix I**.

3.7.3.3. Rollback Purge Tab

Configure the Purge Policy depending on the completion status of the Rollback job.

The logs that R-1 generates (i.e. General report / Transfer report) can become quite large depending on the size of a given job, the detail level of the reports, and the number of jobs that are actively running, or have already been archived. As these logs can occupy vast amounts of disk space, R-1 is configured to automatically purge them after a given amount of time.

NOTE: When a job has reached its purge age, it will be deleted. A purged job is no longer viewable through either the Console CLI or the Console GUI.



On Success

○ Use Server Defaults

● Purge After: 1 Days 0 Hours 0 Min.

○ Never Purge

On Error

○ Use Server Defaults

● Purge After: 7 Days 0 Hours 0 Min.

○ Never Purge

On Abort

○ Use Server Defaults

● Purge After: Days 6 Hours 0 Min.

○ Never Purge

Figure 46: Rollback Purge Policy

For each of the types of completed jobs **On Success, On Error** and **On Abort**, define the time the job**-** reports should be purged:

- **Use Server Defaults** – Use server defaults. For instructions on configuring server defaults, consult refer to the **Manage – Scheduler** Tab.

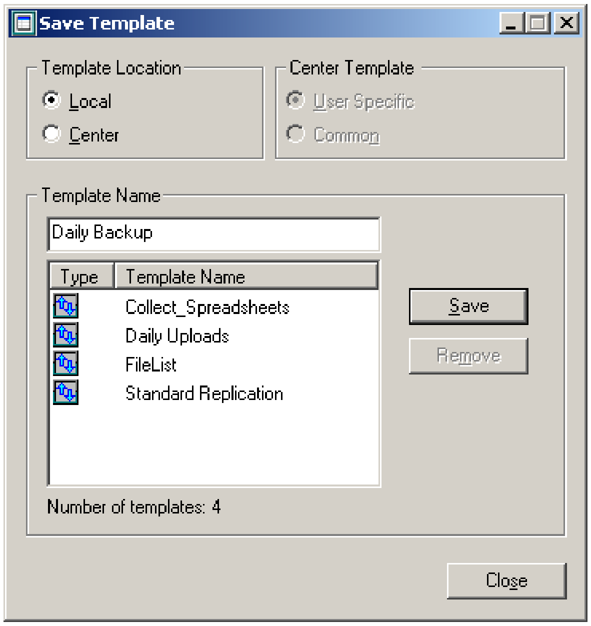
- **Purge After**- Specify the duration (Days/Hours/Minutes) in which to keep the Job's reports.

- **Never Purge** – Specify to R-1 never to purge a Job.

Job reports can also be sent automatically via e-mail using the **On Exit** tab in the job properties**.** This will allow the user to be actively informed of the completion status of a job, and allow the user to keep the reports as long as is required. These reports can similarly be configured to only be sent on Success / Error / or Abort status of a job.

3.8. Using Templates

From the **Job Definition** windows, by clicking on the **Template** button, all field values in the job definition and the associated properties can be saved in a template file. A template file can be used for easy generation of similar jobs, or simply to have a backup of regularly scheduled jobs for future resubmission.



Template Location: ● Local ○ Center

Center Template: ● User Specific ○ Common

Template Name: Daily Backup

Type	Template Name
▨	Collect_Spreadsheets
▨	Daily Uploads
▨	FileList
▨	Standard Replication

Number of templates: 4

Figure 47: Defining Templates

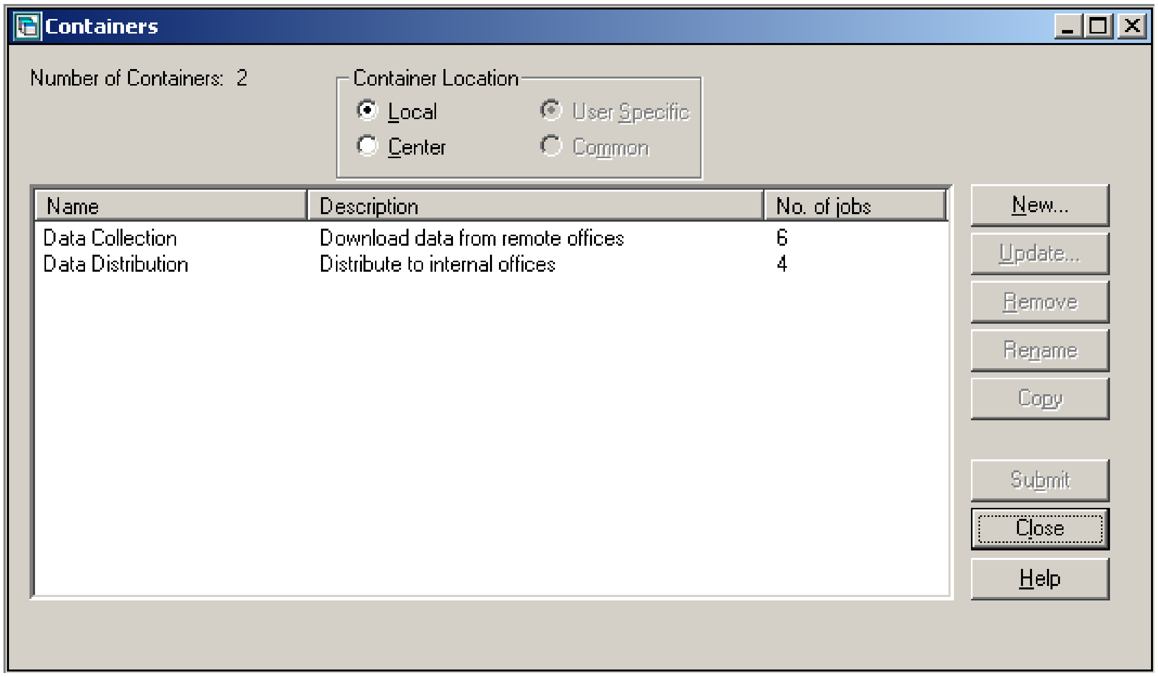
- **Template Location**
  - **Center** indicates that the template should be saved on the Center. Here there are two options:
    - **User Specific** - Templates are accessible only by the **User** that was used when the R-1 Console initially connected to that Center. It is accessible from any Console connecting to the same Center with the same user credentials.
    - **Common**- Templates are available to any user connected to that Center from any Console.
  - **Local** indicates that the templates should be saved on the Console.

NOTE: Saving a template with the **Local** option means it can be accessed no matter which Center that particular Console is connecting to.

A job template may be opened from the **File** menu (File/Template) or by clicking the **Template** button in the Job List window.

3.9. Using Containers

A **Container** is a method used to group multiple jobs in an organized fashion. A Container may be created, removed, renamed, or copied. Pressing **Submit** will immediately submit all jobs in the Container.



Containers

Number of Containers: 2

Container Location

● Local ○ User Specific

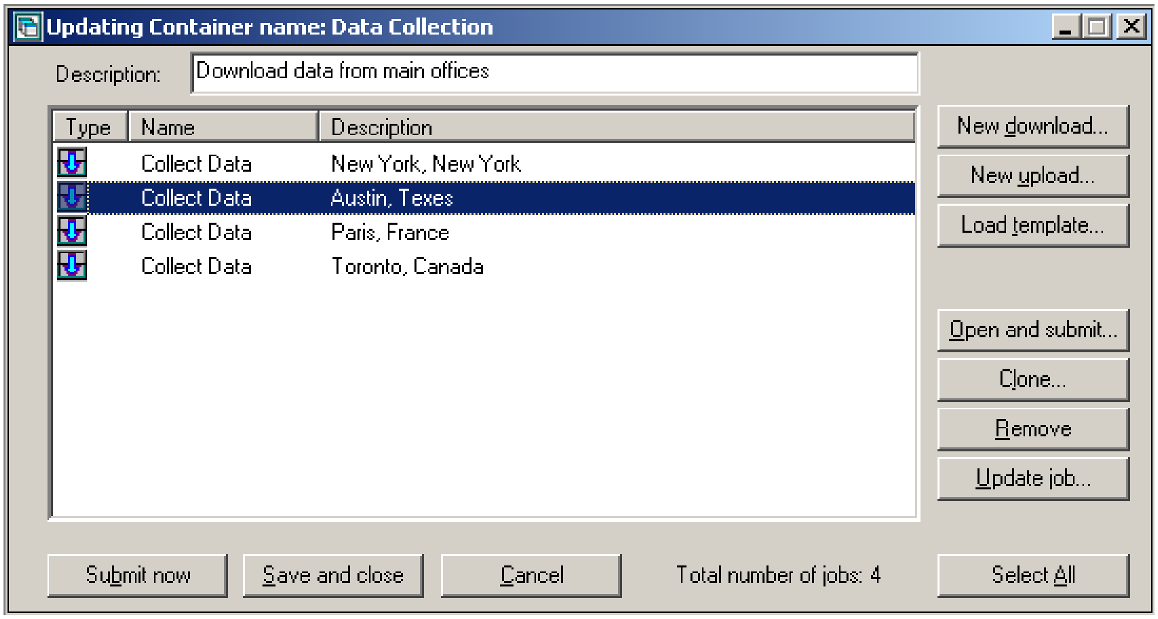
○ Center ○ Common

Name	Description	No. of jobs
Data Collection	Download data from remote offices	6
Data Distribution	Distribute to internal offices	4

Figure 48: Managing Containers

- **Container Location** - As with templates, the Container viewed can be loaded / saved locally (on the Console), or on the Center. If the Container is on the Center it has the option of being globally accessible (**Common** to all users), or only available to the user currently being used to connect to the Center (**User specific**).
- **New** - Upon selecting **New**, the user is prompted for a Container name and description. Once this is completed the user can add jobs to the Container by selecting **New Upload / New Download /** or **Load Template**. This will open their respective job definition or template selection window.

NOTE: Containers can contain only Upload and Download jobs.



Updating Container name: Data Collection

Description: Download data from main offices

Type	Name	Description
⇩	Collect Data	New York, New York
⇩	Collect Data	Austin, Texes
⇩	Collect Data	Paris, France
⇩	Collect Data	Toronto, Canada

Total number of jobs: 4

Figure 49: Defining Containers

When a specific job is selected:

- **Open** – Open the job to make any necessary changes and **Submit** the job immediately.
- **Clone –** Copy the properties of an existing job, make any needed changes, and subsequently add it to the container.
- **Remove** – Remove a job from the container.
- **Update Job –** Edit the Job and place it back in the container.
- **Save -** The container will only be saved upon pressing the **Save** button.
- **Submit Now** - All selected jobs will be immediately submitted.

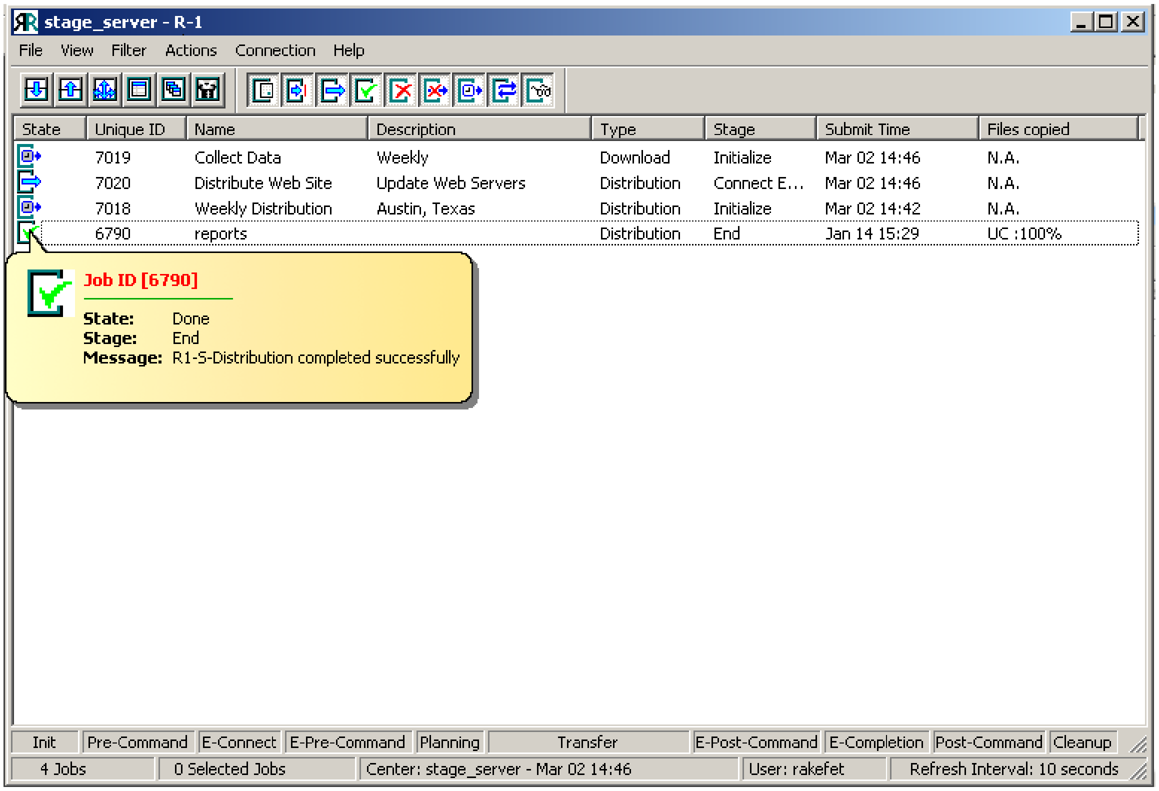
NOTE: Many of the above options become available by selecting a given job or container and right-clicking the mouse.

4. Monitoring and Control

This chapter will outline the Console GUI elements: views, detail windows, menus and toolbars. These functions assist in monitoring Distribution and Replication processes initiated by the Console GUI, the CLI or the R-1 API.

4.1. Console Elements

Job List Window



State	Unique ID	Name	Description	Type	Stage	Submit Time	Files copied
	7019	Collect Data	Weekly	Download	Initialize	Mar 02 14:46	N.A.
	7020	Distribute Web Site	Update Web Servers	Distribution	Connect E...	Mar 02 14:46	N.A.
	7018	Weekly Distribution	Austin, Texas	Distribution	Initialize	Mar 02 14:42	N.A.
✓	6790	reports		Distribution	End	Jan 14 15:29	UC :100%

Job ID [6790]

State: Done
Stage: End
Message: R1-S-Distribution completed successfully

4 Jobs | 0 Selected Jobs | Center: stage_server - Mar 02 14:46 | User: rakefet | Refresh Interval: 10 seconds

Figure 50: The Job List Window

This window displays information about all jobs on a given Center for a particular user. The jobs may be selectively displayed using the available filters. All jobs will remain in the Job List until purged by the R-1 Center or deleted by the initiating user.

The Console can display the following information regarding scheduled, active, or completed jobs. These columns vary depending on individual Console settings, and can be changed in the View/Preference dialog.

Hovering with the mouse above the State column, a tool tip is displayed showing basic information on the job.

- **State** - The current state of the job represented by an icon

Submitted	Job sent to the Center.
Held	Job paused by user. Use the Continue menu item to resume the job.
Running	Job in process.
Rollback	A Rollback job of a Replication job (Upload / Download) in a Running state.
Distribution Rollback	A Rollback job of a Distribution job in a Running state.
Completed	Job successfully completed.
Continuous	Job continuously transfers content, as soon as changes occur.
Scheduled	Job is scheduled for execution.
Failed	Job has exhausted its retries, and has ended in error.
Aborted	Replication / Rollback / Distribution Rollback / Scheduled / Continuous Updates Jobs aborted by user.
Recovering	Job encountered an error and is in recovery mode.
Preview	Job in preview mode, waiting for confirmation.

- **Job ID** - A unique numeric string identifying the job.
- **Name** - The name of the job, if one was provided.
- **Description** - The description of the job, if one was provided.
- **Type** - Indicates whether the synchronization job is distribution, upload or download.

- **Edge**- Edge machine name or IP address.
- **Source Directory** - The job's source directory.
- **Target Directory** - The job's target directory.
- **Stage** - Reflects the progress of the job as it proceeds from Initialization to End
- **Transport Engine** – Displays the transport engine used by that job.
- **Submit Time** – Shows the date and time job was submitted to the Center.
- **Unicast Files** - Number of files to be transferred in Unicast.
- **Files Copied** – Number of files copied.
- **Completion Message** – Reflects the completion status of the job.
- **Scheduled Time** - Date and time job is scheduled to run.

All job entries in the Job List have a context sensitive menu accessible by a right-mouse-click. The options enabled depend on the current state of the particular job.

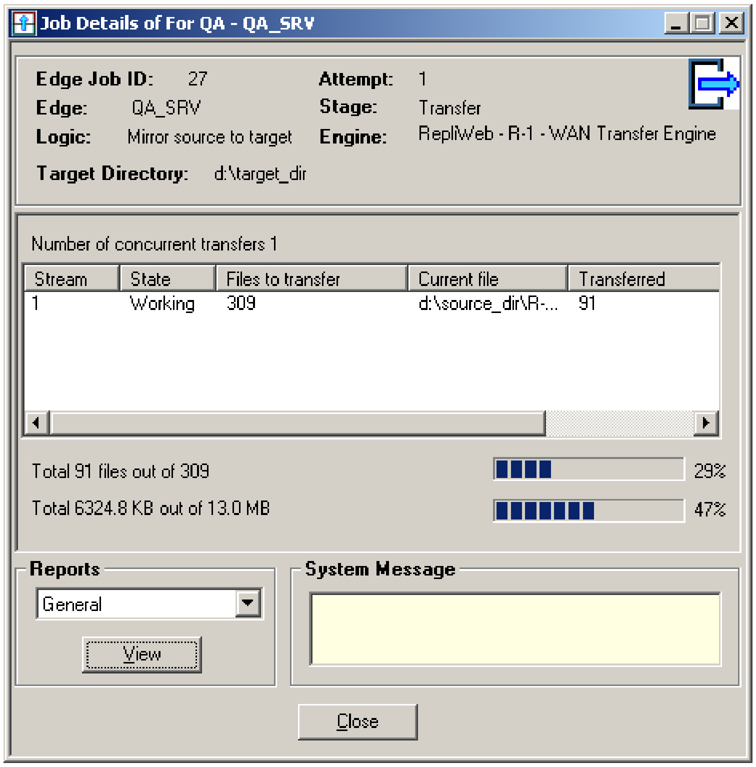
Refer to the **Menus and Toolbars** section for more information about the various elements in this screen.

Job Details

Double click on a job entry in the Job List to open the job's Detail window. The window displays information on the job depending on its current status.

The dialog displayed depends on the type of job selected.

4.1.1. Job Details – Upload / Download / Edge Job



Job Details of For QA - QA_SRV

Edge Job ID: 27
Edge: QA_SRV
Logic: Mirror source to target
Attempt: 1
Stage: Transfer
Engine: RepliWeb - R-1 - WAN Transfer Engine
Target Directory: d:\target_dir

Number of concurrent transfers 1

Stream	State	Files to transfer	Current file	Transferred
1	Working	309	d:\source_dir\R-...	91

Total 91 files out of 309 29%
Total 6324.8 KB out of 13.0 MB 47%

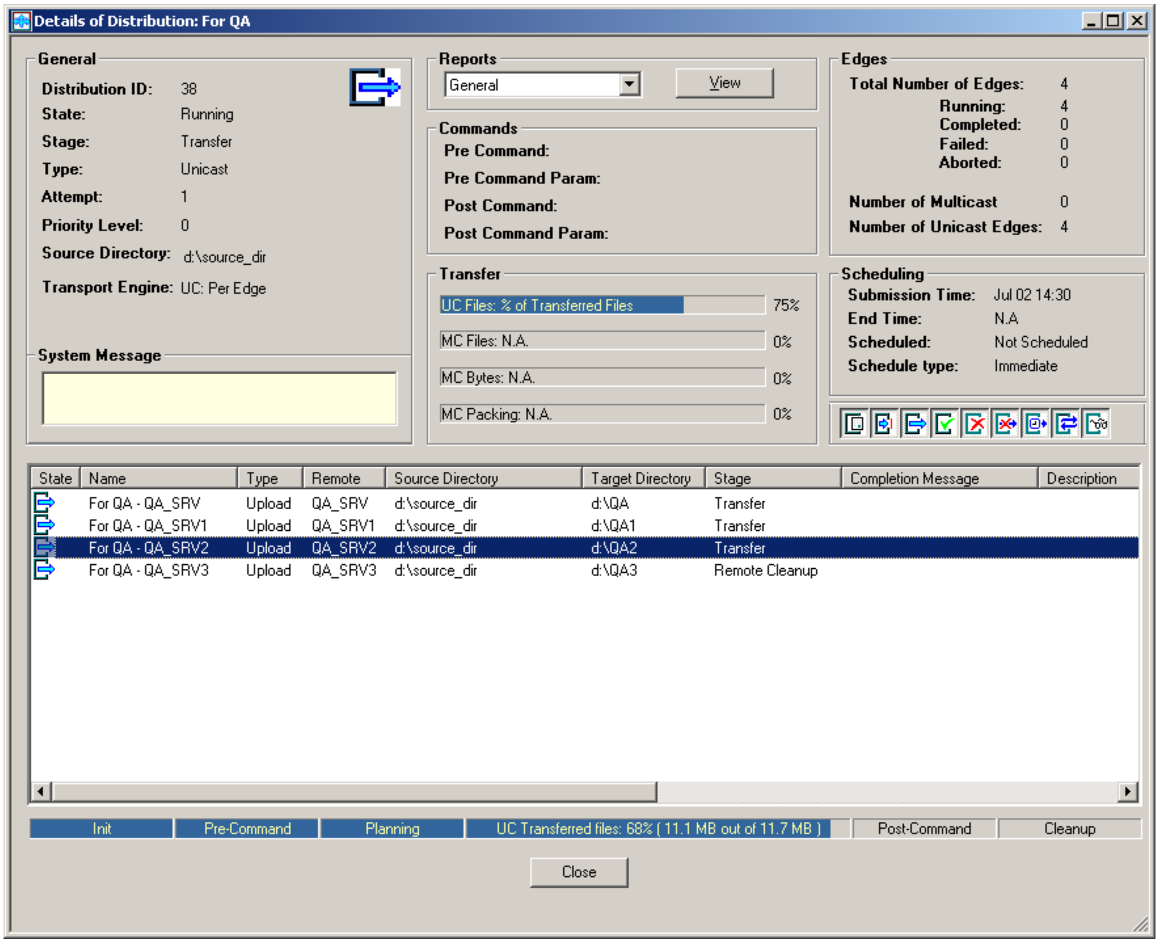
Figure 51: Upload/Download Details Screen

This window displays general information of the job and progress status of transfer streams, pre/post executed commands and transmission statistics.

The job's reports can be accessed by selecting the appropriate report in this window.

For more information about the available reports refer to **Appendix M – Reports**.

4.1.2. Job Details – Distribution Job / Rollback Distribution Job



General

Distribution ID: 38
State: Running
Stage: Transfer
Type: Unicast
Attempt: 1
Priority Level: 0
Source Directory: d:\source_dir
Transport Engine: UC: Per Edge

System Message

Reports

General

Commands

Pre Command:
Pre Command Param:
Post Command:
Post Command Param:

Transfer

UC Files: % of Transferred Files 75%
MC Files: N.A. 0%
MC Bytes: N.A. 0%
MC Packing: N.A. 0%

Edges

Total Number of Edges: 4
Running: 4
Completed: 0
Failed: 0
Aborted: 0

Number of Multicast 0
Number of Unicast Edges: 4

Scheduling

Submission Time: Jul 02 14:30
End Time: N.A
Scheduled: Not Scheduled
Schedule type: Immediate

State	Name	Type	Remote	Source Directory	Target Directory	Stage	Completion Message	Description
	For QA - QA_SRV	Upload	QA_SRV	d:\source_dir	d:\QA	Transfer		
	For QA - QA_SRV1	Upload	QA_SRV1	d:\source_dir	d:\QA1	Transfer		
	For QA - QA_SRV2	Upload	QA_SRV2	d:\source_dir	d:\QA2	Transfer		
	For QA - QA_SRV3	Upload	QA_SRV3	d:\source_dir	d:\QA3	Remote Cleanup		

Init | Pre-Command | Planning | UC Transferred files: 68% (11.1 MB out of 11.7 MB) | Post-Command | Cleanup

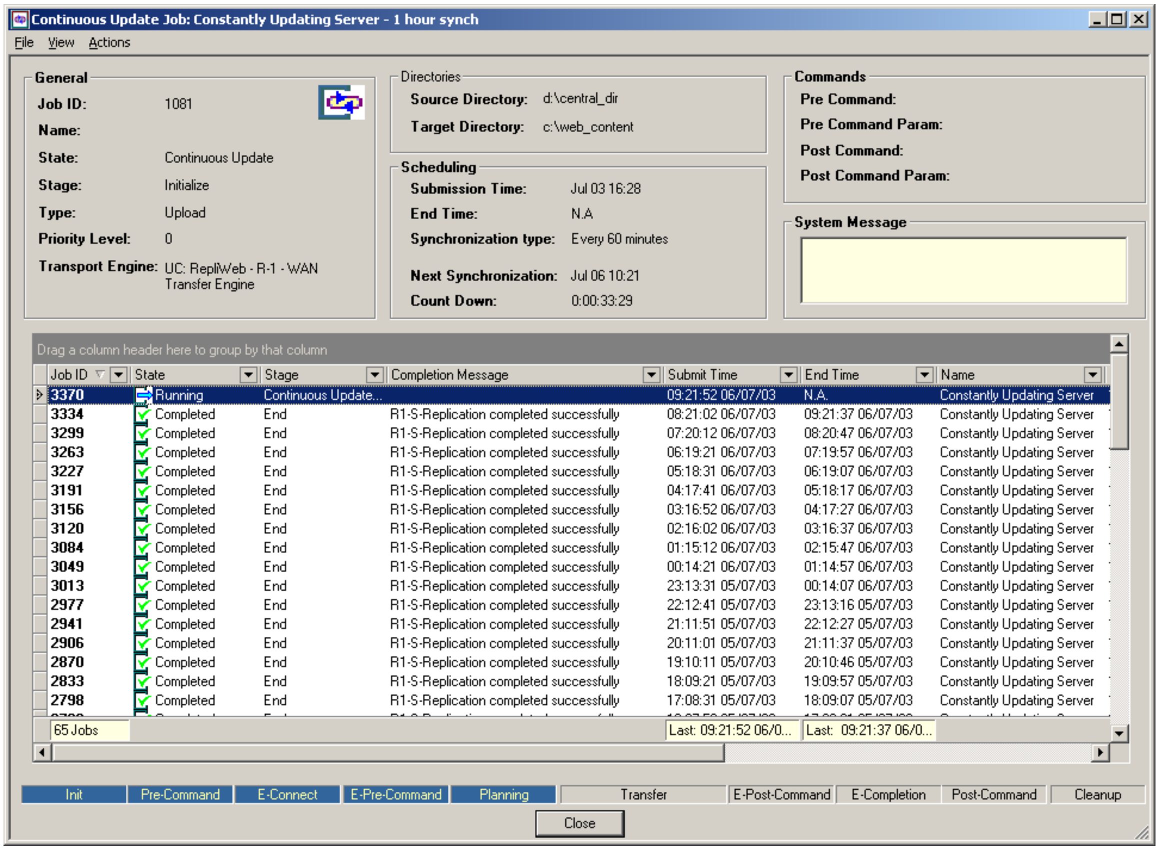
Figure 52: Distribution Details

This window is opened when the job selected in the Job List window is a distribution (one-to-many) job.

The upper part of the screen displays information on the Distribution job. The lower part of the screen displays the list of Jobs representing replication to the specific Edges.

Double click on a job entry to view its details, pre/post command details and progress.

4.1.3. Job Details – Continuous Updates Job


Continuous Update Job: Constantly Updating Server - 1 hour synch
File View Actions

General

Job ID: 1081
Name:
State: Continuous Update
Stage: Initialize
Type: Upload
Priority Level: 0
Transport Engine: UC: RepliWeb - R-1 - WAN Transfer Engine

Directories

Source Directory: d:\central_dir
Target Directory: c:\web_content

Scheduling

Submission Time: Jul 03 16:28
End Time: N.A
Synchronization type: Every 60 minutes
Next Synchronization: Jul 06 10:21
Count Down: 0:00:33:29

Commands

Pre Command:
Pre Command Param:
Post Command:
Post Command Param:

System Message

Drag a column header here to group by that column

Job ID	State	Stage	Completion Message	Submit Time	End Time	Name
3370	Running	Continuous Update...		09:21:52 06/07/03	N.A.	Constantly Updating Server
3334	Completed	End	R1-S-Replication completed successfully	08:21:02 06/07/03	09:21:37 06/07/03	Constantly Updating Server
3299	Completed	End	R1-S-Replication completed successfully	07:20:12 06/07/03	08:20:47 06/07/03	Constantly Updating Server
3263	Completed	End	R1-S-Replication completed successfully	06:19:21 06/07/03	07:19:57 06/07/03	Constantly Updating Server
3227	Completed	End	R1-S-Replication completed successfully	05:18:31 06/07/03	06:19:07 06/07/03	Constantly Updating Server
3191	Completed	End	R1-S-Replication completed successfully	04:17:41 06/07/03	05:18:17 06/07/03	Constantly Updating Server
3156	Completed	End	R1-S-Replication completed successfully	03:16:52 06/07/03	04:17:27 06/07/03	Constantly Updating Server
3120	Completed	End	R1-S-Replication completed successfully	02:16:02 06/07/03	03:16:37 06/07/03	Constantly Updating Server
3084	Completed	End	R1-S-Replication completed successfully	01:15:12 06/07/03	02:15:47 06/07/03	Constantly Updating Server
3049	Completed	End	R1-S-Replication completed successfully	00:14:21 06/07/03	01:14:57 06/07/03	Constantly Updating Server
3013	Completed	End	R1-S-Replication completed successfully	23:13:31 05/07/03	00:14:07 06/07/03	Constantly Updating Server
2977	Completed	End	R1-S-Replication completed successfully	22:12:41 05/07/03	23:13:16 05/07/03	Constantly Updating Server
2941	Completed	End	R1-S-Replication completed successfully	21:11:51 05/07/03	22:12:27 05/07/03	Constantly Updating Server
2906	Completed	End	R1-S-Replication completed successfully	20:11:01 05/07/03	21:11:37 05/07/03	Constantly Updating Server
2870	Completed	End	R1-S-Replication completed successfully	19:10:11 05/07/03	20:10:46 05/07/03	Constantly Updating Server
2833	Completed	End	R1-S-Replication completed successfully	18:09:21 05/07/03	19:09:57 05/07/03	Constantly Updating Server
2798	Completed	End	R1-S-Replication completed successfully	17:08:31 05/07/03	18:09:07 05/07/03	Constantly Updating Server
65 Jobs				Last: 09:21:52 06/0...	Last: 09:21:37 06/0...	

Figure 53: Continuous Updates Details

This window is opened when the job selected in the Job List window is a Continuous Updates job.

The upper part of the screen displays general information on the job. The lower part of the screen displays the list of spawned jobs representing instances of distribution or replication jobs.

Double click on a job entry to view its details, pre/post command details and progress.

4.2. Menus & Toolbars

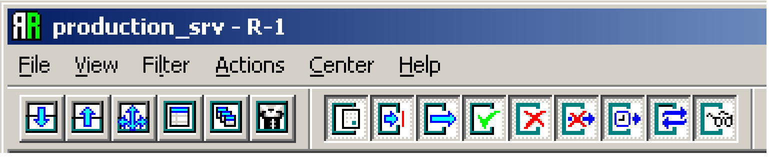


Figure 54: The Menu & Toolbar

The application menus provide access to all R-1 functions.

- **File** - Create a new job (distribution, upload or download), create a new job from a template, use containers, use the wizard and terminate the R-1 Console application.
- **View** - Refresh Job List data, customize the Job List window, including toolbars, layout and field display, set/pause/restart the Job List's refresh rate and set Console properties.
- **Filter** - Apply or remove job display filters based on job state.
- **Actions** - State-sensitive commands that can be performed on individual jobs. The current state of the job determines which options are enabled. These options are only available once one or multiple jobs have been selected.

  Actions include:

  - **Abort** – Abort selected job(s). May be applied to running and held jobs.
  - **Delete** – Delete selected job(s). May be applied to completed jobs only.
  - **Resubmit –** Resubmit a job. May be applied to completed jobs only. Not available to multiple selected jobs.
  - **Hold –** Hold running job(s). May be applied to running or held jobs (to change the resume time).
  - **Continue** – Continue to run held job(s).
  - **Immediate Instance** – Force a scheduled job to spawn a new instance. Not available to multiple selected jobs.
  - **Full Synchronization** – Force a Continuous Update jobs to perform full synchronization.
  - **Submit Now** – Available for On-Demand jobs. The user whose on-behalf the job was submitted can submit the jobs using that option.
  - **Rollback** – Recover recorded information for completed jobs that recorded Rollback information. Not available to multiple selected jobs. May be applied to completed jobs only.

  - **Preview** – Preview and confirm job(s) in Preview state.
  - **Properties** – Open the property pages of the selected job.
  - **Details** – Show the details screen of the selected job. The details screen opened depends on the type of the selected job.

- **Center** - Connect or reconnect to a Center. Members of the Administrators group on the Center can also manage the Center through the Repliweb Enterprise Manager (REM).
- **Help** – Initializes the R-1 help. From the Help menu the user can also access R-1's online knowledge base and support form.

NOTE: It is possible to perform multiple operations on selected concurrently running jobs. Simply right click these jobs in the Job List Window; a context sensitive menu of available job options appears. Click the action(s) to be performed on the jobs selected. Only those actions that can be performed on ALL jobs selected will be highlighted.

Toolbar Buttons

The Distribution and Filter toolbars are analogous to the toolbar functions in most standard Windows applications. As such, the toolbars may be undocked and placed at any place in the main Job List window.

Tool tips appear when the pointer is held over each button.

Listed below are the buttons available on the R-1 Console GUI.

Distribution Buttons

Download	Click this button to define a download job for submission or to save to template.
Upload	Click this button to define and submit an upload job for submission or to save to template.
Distribute	Click this button to define and submit a distribution job for submission or to save to template.
Template	Click this button to load a template.
Container	Click this button to create / manage and submit containers.
Wizard	Click this button to activate the R-1 wizard.

Filter Buttons

Submitted	Click this button to view submitted jobs.
Held	Click this button to view a list of jobs being held.
Running	Click this button to view a list of jobs currently running.

✓	Completed	Click this button to show a list of completed jobs.
✗	Failed	Click this button to view a list of failed jobs.
✗➔	Aborted	Click this button to view aborted jobs.
🕒➔	Scheduled	Click this button to view a list of scheduled jobs.
⇄	Recovering	Click this button to view a list of jobs currently in recovery state.
👓	Preview	Click this button to view jobs in preview mode, and confirm snapshots.
⟳	Continuous Updates	Click this button to view a list of continuous update jobs.

NOTE: The Filter icons are masks to filter the job list displayed.

4.3. The Job Flow and Center Status Bars

The Job Flow and Center Status bars, located at the bottom of the Console, present critical information in an easy to understand, at a glance format.

Job Flow

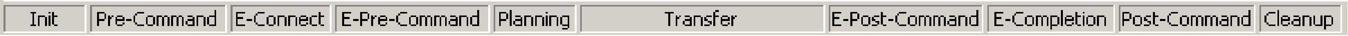
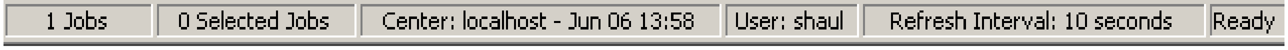


Figure 55: The Job Flow Progress Bar

The **Job Flow** section of the Job List window shows the process stage of the selected job.

The progress through each of the stages in a job's flow: Initialization, Pre-Commands, Planning, Transfer, Post-Commands and Cleanup are clearly displayed (See **Chapter 2** for additional information on the various stages of the **Distribution Process**.)

Center Status



1 Jobs | 0 Selected Jobs | Center: localhost - Jun 06 13:58 | User: shaul | Refresh Interval: 10 seconds | Ready

Figure 56: The Center Status Bar

The bottom margin of the Console shows the Center Status. Basic connection information appears, including:

- **Jobs** – Displays the number of jobs in the Job List window.
- **Selected Jobs** – Displays the number of jobs selected in the Job List window.
- **Center** - Name or IP address of the R-1 Center as entered in the **Center** field when establishing the connection, and the Center local time.
- **User** - Current user, as entered in the **User** field when establishing the connection.
- **Domain** - Current domain, as entered in the **Domain** field when establishing the connection.

NOTE: The **Domain** setting is optional as it is relevant only when connecting to a Windows Center. If using a Windows Center, the current User Name and Password settings will be validated as the specified Domain accounts. If omitted, the User Name and Password are specified as local Center accounts.

- **Refresh Properties** - The performance and frequency of the updates may be controlled from the **View / Preference** menu.
  - **Rate** - Displayed in seconds. This is the frequency that the Console uses to query the Center for information.
  - **State** - Either **Auto-Refresh** or **Paused**. In a Paused state, the data displayed in the Console may be manually updated by pressing F5.

4.4. View Preferences Dialog

General Tab

- **Main Refresh Rate** - Set how often the job window is refreshed, updating the individual job characteristics (i.e. State / Files copied etc…)
- **Show Center Settings At Startup** – This checkbox is only valid when full Center settings have been saved (i.e. Center / User / Password / Domain). Having this box unchecked will result in the Console automatically using the saved information, bypassing the initial Center Settings screen. To connect to a different Center in the future, choose the Connection/Connect menu option.
- **Show Tool Tips** – Specify whether to show the Job tool tip while in the Job List window, and for how long. The tool tip displays basic information on the job, if the mouse is hovering above the State column for a few seconds.

Columns Tab

This tab allows the user to choose which columns will be viewable in the main Console window for a specific job and also which columns will be available for an Edge in the Job Distribution details list.

Locations Tab

This tab allows the user to choose where all job and template defaults will be obtained.

Job Defaults and **Console Preferences** can be saved:

- **Local** – Setting are saved on Console, and default will only apply to the current Console system. If a job is defined, and the Default button (in the main job definition window) is pressed, all fields and options will be saved to the default job template.
- **Center** – Each Console connected to this Center will use this default.

Console preferences include the viewable columns and refresh rate.

The template and container defaults can also be Local or Center. If Center is selected, they can be either **User Specific** or **Common** (for use by any user).

5. Creating a Distribution Job: CLI

Using the R-1 Command Line Interface (CLI) allows the user to submit, monitor, and control distribution jobs. R-1 affords the flexibility to choose between running the CLI on either a Windows or UNIX machine. The distribution can then be run without regard to the location of the Center and Edge components.

This chapter will outline all commands and qualifiers that are part of the R-1 Command Line Interface.

5.1. Introduction

The Console CLI resides in the `~/repliweb/r1/console/bin` directory on a UNIX installation, and `~\RepliWeb\R1\Console\Bin` on a Windows installation.

The appropriate PATH should be added to the relevant systems so that the Console CLI can be run from any directory on the system.

The syntax of the **`r1 command`** is:

```
r1 <command> [-qualifier=value] [-switch] [-noswitch]
```

The **`<command>`** verb defines the requested action (**`submit`**, **`purge`**, etc.).

NOTE: All CLI commands should be written in one line.

The CLI employs several qualifiers providing all the necessary information to precisely define the outcome of the command. This can range from the standard job definition information such as the Center, Edges, source and target directories to advanced distribution options.

If a qualifier does take a value, it should be followed by an equal sign and the value. If the value contains any spaces or wildcards, the value should be properly escaped and encapsulated in quotation marks. This will be made clear in examples accompanying the text.

A qualifier that does not take a value is also called a **switch**. All switches can be negated by adding the prefix `-no` to the qualifier name.

Switch Example:

```
> r1 submit … -tree_recurse
```

Here, the replication should recurse through the entire source tree structure. In the following example, however, only the root directory and the files within it will be replicated.

```
> r1 submit … -notree_recurse
```

Simple Examples:

- To list all jobs for user “root” on a Center called `stage_server`, the following command should be used:

```
> r1 show -center=stage_server -center_user=root
-center_password=root_password
```

- To get the list of completed jobs from the same Center performing, use the following command:

```
> r1 show -center=stage_server -center_user=root
-center_password=root_password -state=completed
```

The **`set`** command may be used to specify a default Center connection. The Center must be specified in each R-1 command. Using **`set`**, the user can avoid having to specify the Center connection qualifiers in subsequent commands. The connection information stored is user specific. In the following example, the **`show`** is used to display all jobs on the Center `stage_server`. The **`set`** command is issued first so that the Center information need not be re-entered.

The **`set`** command specifies the default user name and password to be used on the Center `(stage_user).`

The **`submit`** command demonstrates how to submit a distribution job to run on the `stage_server` Center replicating the directory tree `/video01` from `stage_server` to the directory /stream on `live_server[1-3].`

The user and password specified in the **`submit`** command will be used as access credentials to `live_server` by the distribution process itself. The distribution job run on the `stage_server` will function under the context of the user `stage_user` (the default user set by the previous **`set`** command).

```
>r1 set -center=stage_server-center_user=stage_user
-center_password=PASS
RWAPI-S-SETS, successfully set default center information

>r1 show
RWAPI-E-NOJOB, no jobs matching query on center "stage_server"

>r1 submit -source_directory=/video01 -target_directory=/streams
-user=live_user -password=PASS live_server1,live_server2,
live_server3
RWAPI-S-SUS, job <22> successfully submitted
```

5.2. R-1 Commands

All R-1 commands support the use of qualifiers. Qualifiers allow the user to control the command's outcome and behavior.

As some of the commands use mutual groups of qualifiers, these qualifiers will be described in this chapter prior to the command's description. Each command description will list the qualifiers used by that command.

Commands list

- General Commands
  - **`Help`**
  - **`Information`**
  - **`Set`**
  - **`Scramble`**
  - **`Manage`**
- Job Definition Commands
  - **`Submit`**
  - **`Demand_Submit`**
  - **`Resubmit`**
  - **`Replication_Rollback`**
  - **`Distribution_Rollback`**
- Control Commands
  - **`Purge`**
  - **`Delete`**
  - **`Hold`**
  - **`Resume`**
  - **`Abort`**
- Monitoring Commands
  - **`Show`**

5.2.1. Cross Platform Formats and Escaping

As R-1 is a cross-platform system, there may be cases where adjustments need to be done to overcome the syntax differences between Windows and UNIX. As the Console, Center and Edge may reside not only on different machines but also on different operating systems, it is important that each component will receive the commands in the native format of the systems it resides on.

This is relevant for qualifiers dealing with directory and file names, and also for qualifiers like `-job_name` which may include blanks.

For source directory qualifiers, the format must be in the native format of the source operating system (in a distribution job, that will be the Center). For target directory qualifiers, like `-target_directory`, the format will be that of the target operating system (the Edge, in a distribution job).

In Windows, escaping needs to be performed only when there are blanks within the expression. In UNIX, escaping include doubling \ and adding \ before “.

Examples:

If the command is issued from the R-1 Console CLI residing on a Windows machine, and the Edge is also a Windows machine, escape the directory name only if it includes blanks.

```
>r1 submit … -source_directory="D:\SourceContent\New Content"
```

If the command is issued from the R-1 Console CLI residing on a UNIX machine, and the Center is on a Windows machine, the expression must be properly escaped.

```
>r1 … -distribution_name=\\\"Collect Data\\\"
```

If the command is issued from the R-1 Console CLI residing on a Windows machine, and the Edge is on a UNIX machine, the directory name must be in UNIX format.

```
>r1 submit … -target_directory=/TargetContent/New
```

5.3. General Qualifiers

5.3.1. Output Control Qualifiers

Output control qualifiers are used to control the redirection of the command output. By default, the command output is written to the standard out.

`-output=`**`output_file_name`**

> The command's output will be written to **`output_file_name`**. This should be a full path to avoid the file being placed in the job temporary directory. Omitting a path will not place the output file in the current directory.

`-output_mode=`**`new|append`**

> This qualifier must be used in conjunction with the `-output` qualifier.
> **`new`** - Specifies that a new file will be created. If an output file with the same name exists, it will be overwritten.
> **`append`** - Indicates that the output of the command is to be appended to the specified output file. If the output file does not exist, it will be created.

`-output_style=`**`tagged`**

> This qualifier describes how the output of the command issued is presented. **`tagged`** specifies that the output will be tagged in XML format for future parsing. This option should be used by applications that will need to parse the output of R-1 commands.

`-silent`

> This qualifier prevents any output from being sent to the standard output (by default the screen.) It should be used by calls to the R-1 API (Application Program Interface).

5.3.2. Center Connection Qualifiers

With the distributed architecture of R-1, commands issued on the Console machine initiate, monitor and control distribution tasks on a Center. The Center can be on the same machine as the Console or can reside on a separate machine. In any case, with each R-1 command, Center connection information must be provided. This information is used to define the Center executing the command and its credentials.

`-center=`**`center_id`**

> `-center` defines the Center to which the command is sent. **`center_id`** can be either a hostname or IP address.

`-center_user=`**`user_name`**

> **`user_name`** indicates the context under which the connection to the Center takes place. Any actions taking place on the Center will run under the **`user_name`**

context. The control and monitor commands will affect only those jobs submitted by this **`user_name.`** **`user_name`** must be a valid user on the Center system for the connection to succeed, as the **`user_name`** credentials (user name and password) require validation by the Center.

`-center_password=`**`password`**

`password` represents **`user_name`**'s password, to be verified as connection credentials by the Center. **`password`** is provided as plain text. To use this qualifier in scripts, we recommend that the password be scrambled. This is accomplished using the **`scramble`** command.

`-scrambled_center_password=`**`scrambled_password`**

To replace the `-center_password` qualifier to use a password scrambled with the **`scramble`** command instead of a clear-text password.

NOTE: The scramble command needs be used only once to generate a scrambled password.

`-center_domain=`**`domain`**

When **`user_name`** is part of a Windows domain, this qualifier must be added. If **`user_name`** is not part of a Windows domain (i.e. a UNIX user, or a local user on a Windows machine), this qualifier can be omitted.

`-center_ssl|-nocenter_ssl`

Specify to R-1 that all communication to the Center will be over SSL.

For explanation about R-1 security, see the **Overview Chapter**.

5.4. General Commands

5.4.1. Help Command

The **`help`** command displays help text about the syntax and qualifiers of the command line interface (CLI).

Help Qualifiers

⇒ `Output Control qualifiers`

Example:

To extract the command line help text into a text file “`c:\r1.txt`”:

```
> r1 help -output=c:\r1.txt
```

5.4.2. Information Command

To display software version data, use the **`information`** command. The Console component version is always displayed. If the Center and/or Edge connection information is provided with the command (or previously with the **`set`** command), the Center and Edge versions will also be displayed.

Information Qualifiers

⇒ `Output Control qualifiers`
⇒ `Center Connection qualifiers`

`-edge=`**`edge_name`**

Specify the name of the Edge to display information on. **`edge_name`** can be a computer name or IP address.

`-edge_user=`**`user_on_edge`**

The `-edge_user` qualifier identifies the user all actions on the Edge will be performed under.

`-edge_password=`**`password_on_edge`**

Use the `-edge_password` qualifier to enter **`user_on_edge`**'s password.

`-edge_domain=`**`domain_name`**

Specify the Edge's domain. When **`user_on_edge`** is part of a Windows domain, this qualifier must be added. If **`user_on_edge`** is not part of a Windows domain (i.e. a UNIX user, or a local user on a Windows machine), this qualifier can be omitted.

Examples:

When no information is provided:

```
>r1 information
Console information:
   Version : R-1 2.2
   Build Date : May  5 2001 09:41:04
```

When Center information is provided:

```
>r1 information -center=stage_server -center_user=root
-center_password=PASS
Console information:
   Version : R-1 3.2
   Build Date : May  5 2001 09:41:04

Center "stage_server" information:
   Version : R-1 2.0
   Build Date : May  3 2001 19:41:18
```

When Center and Edge information is provided:

```
>r1 information -center=stage_server -center_user=root
-center_password=password -edge=web1 -edge_user=username
-edge_password=password
Console information:
   Version : R-1 3.2
   Build Date : May  5 2001 09:41:04

Center "stage_server" information:
   Version : R-1 2.2
   Build Date : May  3 2001 19:41:18

Edge "web1" information:
   Version : R-1 2.0
   Build Date : May  3 2001 19:41:18
```

5.4.3. Set Command

Using the **`set`** command saves the User's Center settings as the default.
If the **`set`** command is **not** used to define a User's Center Settings, one or more of the Center Connection qualifiers must be added to every command line issued:

Set Qualifiers

⇒ `Output Control qualifiers`
⇒ `Center Connection qualifiers`

Example:

```
>r1 set -center=stage_server -center_user=stage_user
-center_password=stage_password
RWAPI-S-SETS, successfully set default center information
```

5.4.4. Scramble Command

Use the **`scramble`** command to generate hashed passwords. The hashed passwords can then be safely used in scripting instead of the clear text password.

Scramble Qualifiers

⇒ `Output Control qualifiers`

`-password=`**`input_password`**

`input_password` specifies the clear text password to be encrypted.

Example:

```
>r1 scramble -password=guest
Your encrypted password is: L9CFMSD294ACQIBSSR7ONCKJ80
```

NOTE: The scramble command needs be used only once to generate a scrambled password.

5.4.5. Manage Command

Use the **`manage`** command to execute scripts and executables on the remote server.

Manage Qualifiers

⇒ Output Control qualifiers
⇒ Center Connection Qualifiers

-command=**command_and_parameters**

Specify the command name and parameters to be executed on the remote server.

-viewer=**view_application**

This qualifier will pipe the output from the executed script directly into another application.

Example:

```
>r1 manage -command=c:\dir_inetpub.bat
d:\Program Files\RepliWeb\R1\temp>REM This batch files lists
directory inetpub

d:\Program Files\RepliWeb\R1\temp>dir c:\inetpub
 Volume in drive C has no label.
 Volume Serial Number is 60AA-11C9

 Directory of c:\inetpub

16/10/2002  11:50a      <DIR>          .
16/10/2002  11:50a      <DIR>          ..
16/10/2002  11:48a      <DIR>          AdminScripts
16/10/2002  11:50a      <DIR>          ftproot
16/10/2002  11:49a      <DIR>          iissamples
16/10/2002  11:50a      <DIR>          mailroot
16/10/2002  11:49a      <DIR>          Scripts
16/10/2002  11:50a      <DIR>          webpub
21/10/2002  11:46a      <DIR>          wwwroot
               0 File(s)              0 bytes
               9 Dir(s)     318,947,328 bytes free
```

5.5. Job Definition Commands

Job definition commands are commands that are used to define jobs for either immediate, or future submission. Each job has individual settings from its scheduled run time, target Edges, source and target directories, and specific Edge definitions. These job definitions can be taken as qualifiers when the job is submitted, from a job previously submitted, or from a job template previously defined. These possibilities are reflected in the **submit** command.

5.5.1. Edge Connection Qualifiers

Edge connection information must be provided to allow the Center to access the various Edges. This information is used to define the Edge executing the command and its credentials.

-user=**user_on_edge**

The -user qualifier identifies the user all actions on the Edge will be performed under. All processes running on the Edge during the distribution job, including the security context under which any content is read/written, will run under this user's security context.

-password=**password_on_edge**

Use the -password qualifier to enter -user's password.

-domain=**domain_name**

Specify the Edge's domain. When **user_on_edge** is part of a Windows domain, this qualifier must be added. If **user_on_edge** is not part of a Windows domain (i.e. a UNIX user, or a local user on a Windows machine), this qualifier can be omitted.

NOTE: All Edge Connection qualifiers are positional, and can affect all or some of the Edges in the specified command depending on their position in the command line.

5.5.2. Job Definition Qualifiers

-distribution_name=**distribution_job_name**

The -distribution_name qualifier defines the name of the distribution job.

Example:

```
>r1 submit … -distribution_name=\"Collect Data\"
```

-description=**distribution_job_description**

The -description qualifier provides a description of the distribution job.

Example:

```
>r1 submit … -description="Pull content from Austin, Texas"
```

The two qualifiers noted above are used for administrative and/or cosmetic purposes. Monitoring can also be queried by distribution name. Each job is issued a unique ID (UID), instead of having to query the Center for the UID of a particular job, the job can simply be referred to by its name and description. This is valuable for those applications that rely on their own naming conventions, thus saving the user from having to initially query a Center to determine the unique ID of a submitted job.

`-source_directory=`**`distribution_source_directory`**

> **`distribution_source_directory`** indicates the source directory of the distribution, it should be in the native format of the source operating system.

`-target_directory=`**`distribution_target_directory`**

> **`distribution_target_directory`** indicates the target directory for the distribution, it should be in the native format of the target operating system.

Example:

```
>r1 submit -source_directory=C:\updatedContent
-target_directory=D:\WWWContent -user=user_name
-password=password web1,web2,web3
```

The above command submits a distribution of the source directory `C:\updatedContent` from the default Center to the target directory `-target=D:\WWWContent` on nodes `web1`, `web2`, and `web3`. Access to all target nodes is done with the user *`user_name`* of these nodes.

NOTE: Source and target directories must be defined in their respective local formats. For an explanation on file formats see **Cross Platform Formats and Escaping** description in **Chapter 5**.

NOTE: The qualifiers `-user`,`-password`,`-domain`, and `-target_directory` are positional. If you specify them before the target nodes list, they apply to all nodes. If specified for a specific node, they apply to that node only.

Examples:

```
>r1 submit -source_directory=C:\updatedContent
-target_directory=D:\WWWContent -user=user_name
-password=password web1,web2
-target_directory=E:\WEB2Content,web3
```

Distribution to nodes `web1` and `web3` will be to target directory `D:\WWWContent` and distribution to node `web2` will be to target directory `E:\WEB2Content` .

```
>r1 submit web1,web2, web3 -source_directory=C:\updatedContent
-target_directory=D:\WWWContent -user=user_name
-password=password
-target_directory=E:\WEB2Content
```

Distribution to node `web3` will be to target directory `D:\WWWContent.` Distribution to nodes `web1` and `web2` will fail, as there is no `-target_directory` specification for these nodes.

NOTE: To avoid confusion in the command line, both in Edges' qualifiers order and cross platform file format, it is recommended to use the **`edge_default`** and **`edge_override`** commands.

Example:

Instead of using:

```
>r1 submit -source_directory=C:\updatedContent
-target_directory=D:\WWWContent -user=user_name
-password=password web1,web2
-target_directory=E:\WEB2Content,web3
```

It is better to use:

```
>r1 submit -source_directory=C:\updatedContent
-modifier=c:\mod.txt web1,web2, web3
```

The file mod.txt will include:

```
edge_default set -target_directory=D:\WWWContent
-user=user_name -password=password

edge_override web1 mod <any change you want>
edge_override web3 mod <any change you want>

# Overriding web2 target directory
edge_override web2 mod –target_directory=E:\WEB2Content
```

5.5.3. Distribution Qualifiers

These qualifiers control the logic of the distribution process.

`-distribution_type=`**`unicast|multicast`**

Defines the distribution type:
`unicast` - Indicates that all Edges will be accesses using TCP Unicast only.
`multicast` - Indicates that Multicast will be used. If Edge content cannot be distributed with Multicast, the distribution to that Edge will fail.

`-continue_on_empty_source|-nocontinue_on_empty_source`

If the source directory of the distribution is empty, **ALL** files on the target run the risk of being deleted. For this reason, an empty source list will instigate a job failure.

Use this option to make sure the Job will not proceed in case of an empty source directory.

`-unicast`

`-unicast` is a Positional qualifier. When specified before an Edge (in the Edges list), it indicates to the Center to transfer content to that Edge using Unicast only. (This will override the `-distribution_type` if specified for the complete Edges list.)

`-use_central_lists=`**`snapshot_files_directory`**

In cases where the CST™ and CSM™ processes may take a long time, the user can create the local (Center) snapshot lists. This qualifier will specify to R-1 not to perform the Center's snapshot but rather use a previously built one. The files that the user built will reside in the **`snapshot_files_directory`**.

See **Appendix E** for a complete explanation on this feature.

-cross_platform_indication=**auto_detect|windows_only|unix_only|cross_platform**

By default R-1 queries all Edges for their Operating System to decide what kind of Snapshot it should create. This may cause the job to hold, as the Snapshot phase will not be completed without information from all Edges. Using this qualifier, the user can specify what kind of snapshot to create, so the Edges will not be queried.

windows_only or **unix_only**: Use when all Edges are of the same Operating System as the Center. This will result in an enhanced snapshot using all file system qualifiers (including permissions).
cross_platform: R-1 will create a snapshot containing only information which is common to both Operating Systems. Permissions will not be transferred.
auto_detect: Default. R-1 will query all Edges for their Operating System and will build the Snapshot accordingly.

Controlling the definition of distribution success:

-all_edges_required|-noall_edges_required

By default it is required that the distribution will complete for all Edges. If that is not the case, -noall_edges_required should be specified.

-min_nodes=**n**

If -noall_edges_required is specified, **n** specifies the minimum number of Edges that successful completion for them defines the success of the distribution job.

-min_nodes_optimization|-nomin_nodes_optimization

The distribution job determined its completion status depending on the -all_edges_required and -min_nodes qualifiers.

This switch specifies **when** the distribution process determines its completion status.

- -nomin_nodes_optimization: completion status is determined upon completion of **ALL** Edge jobs.
- -min_nodes_optimization: (the default) completion status is determined during Edge jobs execution.

5.5.4. Synchronization Qualifiers

In addition to pre and post commands of the Center, each Edge distribution can have its own specific pre and post commands. When these commands need to be synchronized between the different Edges, the following qualifiers should be used:

`-sync_before_pre_command|-nosync_before_pre_command`

Edge pre commands will not start before all Edges are ready to run their pre commands.

`-sync_after_pre_command|-nosync_after_pre_command`

Edge will not continue to their transfer stage before all Edges have completed their pre commands.

`-sync_before_post_command|-nosync_before_post_command`

Edge post commands will not start before all Edges are ready to run their post commands.

5.5.5. Multicast Qualifiers

Controlling Multicast distribution parameters (ignored for Unicast distribution).

For a complete explanation regarding Multicast Transfer see **Distribution Features – Multicast** in the **Overview** chapter.

`-mc_engine=`**`rwmc`**

Select the Multicast transport engine. This will enable setting specific engine properties using the corresponding qualifiers.

`rwmc` – Use the RepliWeb multicast engine. In addition to general Multicast qualifiers, use with R-1 Multicast Engine Qualifiers.

`-mc_min_quorum=`**`n`**

The minimum number of Edges required for starting the Multicast stage of the distribution job. A job is ready to be a part of the Multicast transfer once it has completed its CSM™. The distribution job will wait for a minimum number of jobs to reach that stage before calculating which files should be sent with Multicast and which with Unicast.

`-unicast_edge_failover|-nounicast_edge_failover`

If an Edge did not meet the Multicast Quorum in time, it will continue on its own using Unicast. Also, an Edge that was part of the Multicast transfer but failed to receive content will continue and recover using Unicast.

This qualifier will take effect only if the `-mc_min_quorum` was not reached and `-distribution_type` is **`multicast.`**

`-unicast_file_failover|-nounicast_file_failover`

A file that was part of the Multicast transfer but failed during transmission will continue and recover using Unicast.

This qualifier will take effect only if the `-mc_file_quorum` was not reached and `-distribution_type` is **`multicast.`**

`-mc_quorum_max_wait=`**`n`** **`in seconds`**

Specifies the time in seconds the job waits for the Multicast Quorum to assemble.

`-mc_maximize_time=`**`n`** **`in %`**

Once Multicast Quorum is reached, defines, the job will wait to allow other Edges to reach the Multicast stage. This is in percent of the amount of time it took the quorum to assemble.

`-mc_file_quorum=`**`n`**

The minimum number of Edges to which a file has to be sent for Multicast transmission. Files that should be sent to fewer Edges than specified are not sent with Multicast but with Unicast, if `-unicast_file_failover` is specified.

`-mc_min_file_size=`**`n`** **`in KB`**

The minimum size of a file for Multicast transmission. Smaller files are not sent with Multicast but with Unicast, if `-unicast_file_failover` is specified.

`-mc_max_file_size=`**`n`** **`in KB`**

The maximum size of a file for Multicast transmission. Larger files are not sent with Multicast but with Unicast, if `-unicast_file_failover` is specified.

`-mc_buffer_size=`**`n`** **`in bytes`**

`n` specifies the buffer size for Multicast transmission.

`-mc_pack_files|-nomc_pack_files`

Specify whether to use file packing while multicasting data.

When the source directory contains many files, it is recommended to pack them into larger packages for better performance, as the overhead of multicasting a file take longer than actually transmitting that file. Multicasting just a few large files it is recommended not to use file packing.

`-mc_max_pack_size=`**`n`** **`in MB`**

Determine the maximum total files size to pack in one package.

`-mc_max_pack_buffer_size=`**`n`** **`in Bytes`**

Each package is splitted into buffers. **`n`** specifies the maximum size of each buffer, in Bytes.

`-mc_pack_compress|-nomc_pack_compress`

Use this switch to compress the files while packing. This will allow more files to be sent in one package.

`-mc_packet_retry_factor=`**`n`**

Determine the number of times a packet should be sent in Multicast to verify transmission.

5.5.6. R-1 Multicast Engine Qualifiers

`-mc_group=`**`mc_group_address`**

Determine the target address of the Multicast group. Format is IP Address.

`-mc_port=`**`n`**

`n` specifies the port on the Center to be used for Multicast transmission.

`-mc_ttl=`**`n`**

`n` specifies the number of hops a buffer can do before Multicast transmission fails. Default is 32.

`-mc_send_repetitions=`**`n`**

`n` specifies the number of times a file should be sent in Multicast transmission.

`-mc_max_transfer_rate=`**`n`** **`in MB|Mb|KB|Kb / seconds`**

Determine the maximum Multicast transfer rate.
`n` represents an absolute numeric value for the maximum Multicast transfer rate. It should be followed by **`MB|Mb|KB|Kb.`**

An expression can also be entered using time frames. See **Appendix B** for Time Frames expressions.

Controlling the collection of a Multicast distribution parameters in the Edge:

`-mc_server_timeout=`**`n`** **`in seconds`**

`n` specifies the maximum number of seconds an Edge will wait to receive data from the Center in Multicast transmission.

5.5.7. Recovery Qualifiers

These qualifiers determine the number of times a job should attempt to recover before exiting with a failure, as well as how long the job should wait between retries.

`-max_retries=`**`n`**

The maximum number of recovery tries before failure.

`-base_interval=`**`n`** **`in seconds`**

The interval between retries.

`-retry_factor=`**`n`** **`in percent`**

An incremental increasing factor used to geometrically increase the interval length between attempts. For example, with `-base_interval`=100 and `-retry_factor`=10.

`-max_interval=`**`n`** **`in seconds`**

The maximum period of time to which the `-base_interval` can grow to using `-retry_factor`.

5.5.8. Execution Qualifiers

Data distribution is often part of a wider operation. In that event, pre or post processing are needed and become an integral part of the distribution operation. The built in recovery mechanism considers an operation incomplete if the post-synchronization processing was not completed successfully, even if the distribution itself did.

Commands can be run prior (pre) or after (post) a distribution on the Center machine.

The commands include an absolute path to the batch file or script, and any parameters or switches should be passed to the command using the appropriate qualifier.

`-pre_center_command=`**`command_to_run`**

R-1 will execute the **`command_to_run`** on the Center prior to the Job submission.

`-pre_center_parameters=`**`command_parameters`**

`command_parameters` is the text to be passed as parameters to the `-pre_center_command`.

`-post_center_command=`**`command_to_run`**

R-1 will execute the **`command_to_run`** on the Center after a successful execution of the Job.

`-post_center_parameters=`**`command_parameters`**

`command_parameters` is the text to be passed as parameters to the `-post_center_command`.

5.5.9. Exit Procedure Qualifiers

These are procedures that will be performed on completion of a job. These procedures include running specified "on-exit" commands with parameters and e-mail notification.

NOTE: Exit is not executed between recovery attempts. It is executed only if this job will never run again.

`-success_exit_procedure=`**`exit_procedure_name`**

The batch file or script to run on successful completion of a job.

`-success_exit_parameters`**`=exit_procedure_parameters`**

The parameters to pass to the `-success_exit_procedure`.

`-success_email_recipients=`**`email_address`**

The e-mail address of the person(s) to be notified in case of successful job completion. Format is: name@company.com. It can also be a list separated by commas.

`-success_user_attachment=`**`user_files_to_be_attached`**

The full path to any file to be attached to the e-mail.

`-success_email_include_reports=\"`**`general,transfer,pre_distribution, post_distribution\"`**

The job reports to include in the e-mail. For more information about the available reports refer to **Appendix M – Reports**.

`-abort_exit_procedure=`**`exit_procedure_name`**

The batch file or script to run if the job was aborted.

`-abort_exit_parameters=`**`exit_procedure_parameters`**

The parameters to pass to the `-abort_exit_procedure`.

`-abort_email_recipients=`**`email_address`**

The e-mail address of the person(s) to be notified in case the job was aborted. Format is: name@company.com. It can also be a list separated by commas.

`-abort_user_attachment=`**`user_files_to_be_attached`**

The full path to any file to be attached to the e-mail.

`-abort_email_include_reports=\"`**`general,transfer,pre_distribution, post_distribution\"`**

The job reports to include in the e-mail.

-error_exit_procedure=**exit_procedure_name**

The batch file or script to run upon the job completing in error.

-error_exit_parameters=**exit_procedure_parameters**

The parameters to pass to the -error_exit_procedure.

-error_email_recipients=**email_address**

The e-mail address of the person(s) to be notified in case the job completed in error after all retries. Format is: name@company.com. It can also be a list separated by commas.

-error_user_attachment=**user_files_to_be_attached**

The full path to any file to be attached to the e-mail.

-error_email_include_reports=\”**general,transfer,pre_distribution, post_distribution\”**

The job reports to include in the e-mail.

-event_report_on=**success,abort,error**

For each completion type, specify whether to report the Job’s exit message to the Windows Event Viewer or the UNIX syslog.

For more information refer to **Appendix I**.

NOTE: For e-mail to be sent as part of the Exit procedure, in the -success_exit_procedure, -abort_exit_procedure and -error_exit_procedure qualifiers the **exit_procedure_name** must be **__MAIL__.**

Only in that case, e-mail will be sent according to the appropriate Job status and the e-mail qualifiers (recipient, attachments, reports.)

Example:

```
>r1 submit … -success_exit_procedure=__MAIL__
-success_email_recipients= manager@company.com,teamleader@company.com
-success_user_attachment=c:\user.doc
```

The following qualifiers are used to operate on the list of transferred files. The transfer reports are parsed according to a template file. The output file may then be executed or email.

For complete information regarding **File List** please refer to **Appendix J**.

NOTE: For file list parsing to be performed, in the `-success_exit_procedure`, `-abort_exit_procedure` and `-error_exit_procedure` qualifiers the **`exit_procedure_name`** must be **__FULL_FILE_LIST__.**

Only in that case, the transfer log will be parsed according to the template file.

`-success_file_list_action=`**`none|email|execute`**

Specify whether the output list should be emailed or executed, upon successful completion of the job.

`-success_file_list_template_file=`**`success_file_list_template_name`**

This qualifier specifies the input file on the Center used for formatting the output list, upon successful completion of the job. This qualifier must be specified if the success custom exit procedure is __FULL_FILE_LIST__.

`-success_file_list_target_directory=`**`success_file_list_target_directory`**

This qualifier specifies the target directory in which the output file will reside, upon successful completion of the job. This qualifier must be specified if the success custom exit procedure is __FULL_FILE_LIST__.
This qualifier must be specified if `-success_file_list_action` is **`none.`**

`-success_file_list_recipients=`**`success_file_list_email_address`**

The e-mail address of the person(s) the file list will be sent to, upon successful completion of the job. Format is: name@company.com. It can also be a list separated by commas.
This qualifier must be specified if `-success_file_list_action` is **`email`**.

`-abort_file_list_action=`**`none|email|execute`**

Specify whether the output list should be emailed or executed in case the job is aborted.

`-abort_file_list_template_file=`**`abort_file_list_template_name`**

This qualifier specifies the input file on the Center used for formatting the output list. This qualifier must be specified if the abort custom exit procedure is __FULL_FILE_LIST__.

`-abort_file_list_target_directory=`**`abort_file_list_target_directory`**

> This qualifier specifies the target directory in which the output file will reside. This qualifier must be specified if the abort custom exit procedure is __FULL_FILE_LIST__.
> This qualifier must be specified if `-abort_file_list_action` is **`none.`**

`-abort_file_list_recipients=`**`abort_file_list_email_address`**

> The e-mail address of the person(s) the file list will be sent to. Format is: name@company.com. It can also be a list separated by commas.
> This qualifier must be specified if `-abort_file_list_action` is **`email`**.

`-error_file_list_action=`**`none|email|execute`**

> Specify whether the output list should be emailed or executed in case the job fails.

`-error_file_list_template_file=`**`error_file_list_template_name`**

> This qualifier specifies the input file on the Center used for formatting the output list. This qualifier must be specified if the error custom exit procedure is __FULL_FILE_LIST__.

`-error_file_list_target_directory=`**`error_file_list_target_directory`**

> This qualifier specifies the target directory in which the output file will reside. This qualifier must be specified if the error custom exit procedure is __FULL_FILE_LIST__.
> This qualifier must be specified if `-error_file_list_action` is **`none.`**

`-error_file_list_recipients=`**`error_file_list_email_address`**

> The e-mail address of the person(s) the file list will be sent to. Format is: name@company.com. It can also be a list separated by commas.
> This qualifier must be specified if `-error_file_list_action` is **`email`**.

5.5.10. Scheduling Qualifiers

These qualifiers enable to schedule jobs in various ways. Jobs may be scheduled to run immediately, or on a more complex schedule.

Any scheduled Job will appear with a clock icon in the Console GUI, and as a scheduled job in the **`show`** command. The job will spawn Edge or Distribution jobs when the predetermined conditions are met.

The scheduled job spawns the first job immediately and starts counting the interval for the next one. The next job will be spawned only after the previous one has finished. It will start either after the specified time has passed, or immediately after the previous job has finished, if it ran longer than the delta specified.

Every spawned job has a different Job ID. The scheduled job itself “runs” forever, until it’s aborted by user command.

`-run_option=`**`immediate|once|every|daily|weekly|triggerfile|filespec|on_demand|continuous_update`**

Defines when the job is scheduled to run:
`immediate` - Run immediately.
`once` - For a job to be run once at a later time.
`every` - For a job to run at a certain interval.
`daily` - For a job to run once daily.
`weekly` - For a job to be run on multiple days and times on a weekly cycle.
`triggerfile` - For a job that will be triggered by the creation of a file.
`on_demand` – For a job that will be triggered by issuing the **`demand_submit`** command. A scheduled job will be displayed, but no replication process will be initiated until the **`demand_submit`** command is issued.
`continuous_update` – For an event-driven job that replicates changes as soon as they occur.

`-trigger_file=`**`trigger_file_name`**

The full path to the trigger file residing on the Center. Only used when the `-run_option=`**`triggerfile.`**

For Distribution jobs, the Trigger file can be located on the Center only.
For Upload and Download jobs, the Trigger file location can be: on Center, on Edge, on both Center and Edge, on either Center or Edge. Upload job trigger file qualifiers can be specified in the Modifier file only, specified in **Appendix F**.

`-once_time=`**`schedule_date`**

Specifies the precise date and time the job should run. Only used when the `-run_option=`**`once.`** Format supports most standard time formats like: “ddmmyyyyhhmm”. For a complete list of supported time formats please refer to **Appendix B – Absolute Time Expressions .**

The scheduling job spawns the first job immediately and starts counting the interval for the next one. The next job will be spawned only after the previous one has finished. It will start either after the specified time has passed, or immediately after the previous job has finished, if it ran longer than the delta specified.

`-every=`**`n`** **`in minutes`**

The job should be run every **`n`** minutes.

`-daily=`**`schedule_time`**

Run daily at **`schedule_time`**. Format is *hhmm*.

`-weekly=`**`weekly_expression`**

Run weekly at **`weekly_expression.`**
Format is: \”w:00 h:12 m:04\” or (w:00 h:12 m:04)

Example:

The job will run every week on:
Sunday 16:25 (04:25 PM)
Tuesday 10:25(10:25 AM)

```
>r1 submit … -run_option=weekly
-weekly=\"(w:00 h:16 m:25) (w:02 h:10 m:25)\"
```

`-terminate_after=`**`n`**   **`in minutes`**

If the job has not completed after **`n`** minutes, it will abort.

`-priority=`**`n`**

-priority is a number between 0-10 that defines the Job’s priority (where 0 is the lowest priority and 10 is the highest). When a job is submitted, all jobs with lower priority are held. When the job finishes, the jobs are resumed and continue to run. The default priority is 5 (medium priority).

To run jobs that will be ignored while calculating priorities, set **`n`** to -1. Jobs like that will not cause lower-priority job to hold, and will not be held when jobs with a higher priority are submitted.

The following qualifiers should be used only with `-run_option=`**`continuous_update.`** Specify the interval in which the job rescans both source and target machines to determine what needs to be replicated, to make sure both source and target are synchronized

NOTE: A continuous update job may **not** be used with the `-priority` and `–terminate_after` qualifiers.

For detailed information on event-driven, Continuous Update jobs, refer to **Appendix H**.

`-synchronize_every=`**`n`** **`in minutes`**

To be used only with `-run_option=`**`continuous_update.`** This qualifier specifies how often (in minutes) the job should synchronize, completing all exit and post replication procedures, returning to the pre distribution processes and the snapshot phase.

`-synchronize_daily=`**`schedule_time`**

To be used only with `-run_option=`**`continuous_update.`** Specify the job to synchronize the source and target directories daily at a specified time. Format is *hhmm*.

`-max_synchronize_delay=`**`n`** **`in seconds`**

To be used only with `-run_option=`**`continuous_update.`** Specify in seconds, how long to wait for a distribution job to exit before full synchronization is about to take place. In case of a lengthy post command or other operation, if the job is about to complete, the Continuous Update job will wait the time specified before aborting the job and performing the full synchronization.

`-max_archived_instances=`**`n`**

For **Scheduled** and **Continuous Updates** jobs only, specify the maximum number of instances to preserve in the archive directory.

NOTE: Use this option with care, as it may consume a lot of disk space. Keeping 30 instances of a distribution job to 10 Edges will result in keeping 300 jobs.

5.5.11. Purge Qualifiers

Configure the Purge Policy depending on the completion status of a job.

The logs that R-1 generates (i.e. General report / Transfer report) can become quite large depending on the size of a given job, the detail level of the reports, and the number of jobs that are actively running, or have already been archived. As these logs can occupy vast amounts of disk space, R-1 is configured to automatically purge them after a given amount of time.

NOTE: When a job has reached its purge age, it will be deleted. A purged job is no longer viewable through either the Console CLI or the Console GUI.

The following qualifiers override the Center defaults. Defaults are located in the Center configuration file.

`-purge_on_success=`**`n`** **`in minutes`**

A job will automatically be purged upon successful completion after **n** minutes. To use the server defaults enter 0 (zero).

`-purge_on_error=`**`n`** **`in minutes`**

A failed job will automatically be purged after **n** minutes. To use the server defaults enter 0 (zero).

`-purge_on_abort=`**`n`** **`in minutes`**

An aborted job will automatically be purged after **n** minutes. To use the server defaults enter 0 (zero).

5.5.12. Input Template Qualifiers

The following qualifiers are used for employing templates in the submit command.

NOTE: Using a replication template (upload or download job), `-type=`**`replication`** must be specified.

`-in_template=`**`template-name`**

Load the job qualifiers from a template previously saved using the GUI. Specify the Template name.

`-in_template_location=`**`local|center_user|center_common`**

Specify the location of the template.

`local` – The template is loaded from the Console machine. To access a template stored locally, it must have been saved locally using the template command

`center_user` – The template was originally stored on the Center machine by the same **`center_user`** that is defined in the current command and is therefore only available for this user.

`center_common` – The template is stored on the Center machine and is accessible to all users connecting to the Center. To access this template it must have been stored using the **`center_common`** option.

Example:

The following command is used to submit a Distribution template, located on the Center for everyone to use, while changing the Credentials used for the Edges:

```
>r1 submit -in_template=weekly_distribution
-in_template_location=center_common

R1API-S-SUS, job <4121> successfully submitted
```

The Template definition already includes all Center credentials and Distribution properties like Transport Engine. The template may include override information for some of the Edges, like Exit procedures.

NOTE: The Template Edges list may be overridden. If the command line includes an Edge list that does not match the Edge list in the Template, the command line list is being used.

Example:

If the template refers to edges `edge1`, `edge2` and `edge3`, the following command will submit only to edges `web1` and `web2`:

```
>r1 submit -in_template=weekly_distribution
-in_template_location=center_common
-unicast_engine=rw_lan web1,web2
R1API-S-SUS, job <4122> successfully submitted
```

Example:

If the Template contains `web1` and `web2` with different Target Directories for each of the edges, and the submit command has different Credentials for these edges; the resulting job will have different target directories **and** different credentials for each of the edges.

```
>r1 submit -in_template=weekly_distribution
-in_template_location=center_common
web3 -user=username3 -password=***
R1API-S-SUS, job <4123> successfully submitted
```

Edge `web3` will have different credentials but will use the default target directory specified in the Template.

Scripting Example:

Using previously saved Templates with the CLI is good practice when using scripts. That way generic scripts may be written, and when the need arises, only variable definitions need to be changed and not the whole command.

The following script is an example that allows to easily change the Edges List, Credentials and Target directory, without changing the script or the command line itself.

1. Save a generic template with all required defaults (transport engine, source directory, exit procedures, etc.).

2. Create a generic script, looking like that:

```
REM Define variables to be used by the r1 submit command
set edge_list=edge1,edge2,edge3
set user=edge_user
set password=edge_password
set domain=mydomain
set target_dir=d:\weekly_target

REM The r1 submit command
r1 submit -in_template=generic -in_template_location=local
edge_list -user=%user% -password=%password% -domain=%domain%
-target_directory=%target_dir%
```

3. Now, when you need to change the edges list, all you need to do is change the line setting the edges list.

The script can be enhanced to accept parameters. It will then use the following commands at the beginning of the script:

```
set edge_list=%1
set user=%2
set password=%3
set domain=%4
set target_dir=%5
```

5.5.13. On Demand Qualifiers

-behalf_users=**users_list**

Use this qualifier to list the users that may actually view and submit the job. Other users will not see this job. User names should be separated by a comma.

Valid user names may be:

Users:	`WEB_Project_Manager`
Domain Users:	`DOMAIN\Domain_User`
Machine-specific users:	`comp_srv\server_user`

`-behalf_groups=`**`groups_list`**

Use this qualifier to list the groups of users that may actually view and submit the job. Other users will not see this job. User names should be separated by a comma.

Valid group names may be:

```
Sales_Group
qa_srv\QA_group
```

NOTE: On Demand – on behalf jobs are visible by the user who submitted the On Demand Job and any user whom the job was run on behalf of (either specifically or as part of a group).

Example:

The following command is used to submit the “on-demand” job. No distribution process is initiated, but the job is visible using the show command.

```
>r1 submit -center=34.0.25.71 -center_user=rwroot
-center_pass=***
-replication_name=\”On Demand Restore\” -edge=152.163.12.2
-user=user_name -scrambled_pass=00U6HJ8MPCLIBSSR7ONCKJ80
-source_dir=/usr/backup -target_dir=d:\data\content
-run_option=on_demand
-behalf_users=comp_srv\server_user,WEB_Project_Manager
-behalf_groups=Sales_Group
R1API-S-SUS, job <4121> successfully submitted
```

5.5.14. Submit Command

The `Submit` command submits a defined distribution job to the Center for distribution along with all of its relevant qualifiers.

Submit Qualifiers

```
⇒ Output Control qualifiers
⇒ Center Connection qualifiers
⇒ Edge Connection qualifiers
⇒ Job definition qualifiers
⇒ Scheduling qualifiers
⇒ Distribution qualifiers
⇒ Synchronization qualifiers
⇒ Recovery qualifiers
⇒ Execution qualifiers
⇒ Exit Procedure qualifiers
⇒ Purge qualifiers
⇒ Rollback definition qualifiers
```

`-type=`**`distribution|replication`**

Specifies the type of the job to be submitted:
`distribution` – A one-to-many distribution in which content is always updated from the Center to the Edge(s). This is the default.
`replication` – A one-to-one replication job in which content can be downloaded from the Center to the Edge or uploaded from the Edge to the Center.

In addition to the above listed qualifiers, the specific performance of several or all Edges can be controlled using a `-modifier` file. This file contains commands to specify default behavior for all Edges and override that behavior for one or several Edges.

NOTE: Qualifiers that are listed in **Modifier Commands & Qualifiers - Appendix F** can be used **ONLY** in the modifier file, and may not be used as qualifiers in the **`submit`** command line itself.

`-modifier=`**`file_name`**

Specifies a file that contains commands that change the distribution properties for all Edges or for a specific Edge.

Use this qualifier to specify Edge distribution parameters in an organized way. Instead of listing all qualifiers over and over again each time the **`submit`** command is issued (like Connection qualifiers), use the **`edge_default`** command to specify the default distribution properties for all Edges.

For Edges that differ in some or all qualifiers than the default qualifiers, use the **`edge_override`** command to specify the Edge's distinct distribution properties.

The **`edge_default`** command is used to define default distribution properties for all Edges.

The **`edge_override`** command is used to override default or command properties for some (or all) Edges.

Use the **`pre_distribution_download`** command to specify the qualifiers of the download job that should take place prior to the distribution process.

NOTE: See **Appendix F** for the complete explanation of the commands and the list of qualifiers that can be used in the `-modifier` file commands.

The `-modifier` file format is the following:

```
edge_default set [-qualifiers]
{blank lines}
edge_override edge1[,edge2,,] mod [-qualifiers]
pre_distribution_download set [-qualifiers]

# A comment line
```

Example :

```
> r1 submit [-qualifiers] web1,web2,japan1,japan2,us1,us2
-modifier=mod.txt
```

The file `mod.txt` will look like:

```
# specify using LAN engine for all edges
edge_default set –unicast_engine=rw_lan
# specify that the transfer to Japan1 and Japan2 nodes will use
# Transport and compression and SSL
edge_override japan1,japan2 mod –unicast_engine=rw_wan -compression=ap
-ssl
```

5.5.15. Demand_Submit Command

This command is used to immediately submit an on-demand job. The command can be issued for every job that has been submitted with the `-run_option=`**`on_demand`** qualifier.

On-demand jobs are similar to Templates, however instead of submitting a job from a template file, the user actually submits a job from a “sample” job. It can be used for easy generation of identical jobs, or simply to have a backup of regularly used jobs for future resubmission.

IT administrators can use the **Run On Behalf** option to prepare a “template” for project managers, who in turn, will submit the job when required. Specify Users and Groups of users who may see and submit this job. These users, in turn, will issue the **Submit Now** command to initiate the distribution process.

NOTE: Issuing a **`demand_submit`** command while an on-demand job is already running for that job, the newly submitted job will initiate only when the previous one completes.

Demand_Submit Qualifiers

⇒ Output Control Qualifiers
⇒ Center Connection Qualifiers

`-job=`**`n`**

n specifies the job to be immediately submitted.

`-name=`**`job_name`**

Submit all **on demand** jobs that match the **`job_name`** criteria. This may include wildcards.

Example:

The following command is used to submit the **on demand** job. No distribution process is initiated, but the job is visible using the show command.

```
>r1 submit -center=34.0.25.71 -center_user=rwroot
-center_pass=***
-distribution_name=\"On Demand Restore\" -edge=152.163.12.2
-user=user_name -scrambled_pass=00U6HJ8MPCLIBSSR7ONCKJ80
```

```
-source_dir=/usr/backup -target_dir=d:\data\content
-run_option=on_demand
-behalf_users=comp_srv\server_user,WEB_Project_Manager
-behalf_groups=Sales_Group
R1API-S-SUS, job <4121> successfully submitted
```

The following command will immediately submit job 4121 with exactly the same qualifiers. This time, a distribution job is initiated.

```
>r1 demand_submit -job=52 -center=34.0.25.71
-center_user=rwroot -center_pass=***
R1API-S-RS, successfully submitted "on-demand job" from job <52>
```

5.5.16. Resubmit Command

Used to resubmit a job that has already been submitted. The command can be issued on completed jobs only as long as the job has not been automatically purged by R-1 or the user. Only completed jobs can be resubmitted.

Adding any additional qualifiers to the command will either add the qualifier to the new job, or override previously set qualifiers.

Resubmit Qualifiers

⇒ `Output Control qualifiers`
⇒ `Center Connection qualifiers`
⇒ `Edge Connection qualifiers`
⇒ `Job definition qualifiers`
⇒ `Scheduling qualifiers`
⇒ `Distribution qualifiers`
⇒ `Synchronization qualifiers`
⇒ `Recovery qualifiers`
⇒ `Execution qualifiers`
⇒ `Exit Procedure qualifiers`
⇒ `Purge qualifiers`
⇒ `Rollback Definition qualifiers`

`-job_id=`**`n`**

`n` specifies the job to be re-submitted. The job must be completed.

`-type=`**`distribution|replication`**

Specifies the type of the job to be submitted:
`distribution` – A one-to-many distribution in which content is always updated from the Center to the Edge(s). This is the default.
`replication` – A one-to-one replication job in which content can be downloaded from the Center to the Edge or uploaded from the Edge to the Center.

NOTE: The job's type must be identical to the job being resubmitted. If the resubmitted job was a replication job, `-type=`**`replication`** must be used. If the resubmitted job was a distribution job, `-type=`**`distribution`** must be used.

In addition to the above listed qualifiers, the specific performance of several or all Edges can be controlled using a `-modifier` file. This file contains commands to specify default behavior for all Edges and override that behavior for one or several Edges.

NOTE: Qualifiers that are listed in **Modifier Commands & Qualifiers - Appendix F** can be used **ONLY** in the modifier file, and may not be used as qualifiers in the **`submit`** command line itself.

`-modifier=`**`file_name`**

Specifies a file that contains commands that change the distribution properties for all Edges or for a specific Edge.

NOTE: See **Appendix F** for the complete explanation of the commands and the list of qualifiers that can be used in the `-modifier` file commands.

Example:

The following command will resubmit job 927 with exactly the same qualifiers it was originally submitted with:

```
>rds resubmit -job=927
RWAPI-S-RS, job <1025> successfully resubmitted
```

Override Example:

Any of the original job's properties can be overridden using the resubmit command. The following command will resubmit job 989 with the originally submitted job, overriding the credentials and the edges list:

```
>rds resubmit -job=989 edge1, edge2, edge3
-user=user_name -passwrod=password
RWAPI-S-RS, job <1180> successfully resubmitted
```

5.5.17. Replication_Rollback Command

This command is used to rollback a Replication Job. A **Rollback** job may be submitted for jobs that had the **Use Rollback** option checked in the `submit` command. When the job is completed (successfully or not) you can select to recover previously recorded data. The Rollback Job allows point-in-time recovery capabilities.

A rollback is similar to an undo function. It enables you to return content to its previously replicated state. Rollbacks are performed on target machine. You cannot roll back content on the source.

For **Scheduled** and **Continuous Updates** jobs, you can rollback to instances of the job using the **`replication_rollback`** command. In **Distribution** jobs, you can roll back the complete distribution process (for all Edges) using the **`distribution_rollback`** command, or roll back individual Edges using the **`replication_rollback`** command.

Rolling back, (recovering data) is performed by selecting a point-in-time to return to, in addition to various job properties that can be set to define the rollback process. This includes standard job properties like pre/post commands, recovery options, etc.

NOTE: Jobs that use the Rollback functionality can be rolled-back only after the job completes (successfully or not). In order to rollback a job, the job cannot be running. To rollback an instance of a **Scheduled** or **Continuous Updates**, the scheduled job itself needs to be aborted; it is not sufficient that the instance is completed.

Replication_Rollback Qualifiers

⇒ `Output Control Qualifiers`
⇒ `Center Connection Qualifiers`
⇒ `Edge Connection qualifiers`
⇒ `Execution Qualifiers`
⇒ `Exit Procedure qualifiers`

`-job=`**`job_id`**

Use this qualifier to specify the job that needs to be rolled back. Note that you can only roll back completed jobs.

`-replication_name=`**`rollback_job_name`**

The `-replication_name` qualifier defines the name of the replication rollback job.

`-description=`**`rollback_job_description`**

The `-description` qualifier provides a description of the replication rollback job.

`-rollback_to_time=`**`time_expression`**

Specify the date and time to recover from. This can be an absolute or a relative time.

NOTE: The time specified is calculated at the Target machine (Edge in case of an Upload job, Center if it's a Download job). It is never the Console time.

- **Absolute time on target** – Specify the exact time to roll back to. Format supports most standard time formats like: "ddmmyyyyhhmm". For a complete list of supported time formats please refer to **Appendix B – Absolute Time Expressions** .

  In case a rollback instance is not found at the exact time specify, R-1 will roll back to the nearest recorded data, without passing the time specified.

- **Relative time on target** – Specify how long to roll back to, in days / hours / minutes. Here, too, the calculated time to roll back from is calculated on the target machine. For relative time format, please refer to **Appendix B – Relative Time Expressions** .

- **Rollback all recorded content** – Using this option, the rollback process will bring the target directory to its state before the replication took place. To use this option, omit the `-rollback_to_time` qualifier from the command.

  NOTE: For non-scheduled Replication jobs, this is the only option available.

Absolute Time Example:

The following command is used to Rollback a scheduled upload job to an absolute time. Data will be rolled back to 10:30 PM according to the Target machine (in this case the Edge):

```
>r1 replication_rollback -job=3212 -replication_name="Rolling
back" -description="Rolling back due to…" -center=center_name
-center_user=user_name -center_password=password
-user=user_name -password=password
-rollback_to_time=2230
R1API-S-SUS, job <4121> successfully submitted
```

NOTE: If you are using different credentials than those used in the original job, make sure there are sufficient access rights to the recording location and the target directory.

Relative Time Example:

The following command is used to Rollback a scheduled Upload job to a relative time. Data will be rolled back to its state 4 hours ago:

```
>r1 replication_rollback -job=212 -replication_name=”Rolling
back” -description=”Rolling back due to…” –center=center_name
-center_user=user_name –center_password=password
-user=user_name -password=password
-rollback_to_time=”-4 hours”
R1API-S-SUS, job <421> successfully submitted
```

5.5.18. Distribution_Rollback Command

This command is used to rollback a Distribution Job, including all Edge jobs belonging to it. A **Rollback** job may be submitted for jobs that had the **Use Rollback** option checked in the **`submit`** command. When the job is completed (successfully or not) you can select to recover previously recorded data. The Rollback Job allows point-in-time recovery capabilities.

A rollback is similar to an undo function. It enables you to return content to its previously replicated state. Rollbacks are performed on target machine. You cannot roll back content on the source.

In **Distribution** jobs, you can roll back the complete distribution process (for all Edges) using the **`distribution_rollback`** command, or roll back individual Edges using the **`replication_rollback`** command. For **Scheduled** and **Continuous Updates** jobs, you can rollback to instances of the job using the **`replication_rollback`** command.

Rolling back, (recovering data) is performed by selecting a point-in-time to return to, in addition to various job properties that can be set to define the rollback process. This includes standard job properties like pre/post commands, recovery options, etc.

NOTE: Jobs that use the Rollback functionality can be rolled-back only after the job completes (successfully or not). In order to rollback a job, the job cannot be running. To rollback an instance of a **Scheduled** or **Continuous Updates**, the scheduled job itself needs to be aborted; it is not sufficient that the instance is completed.

Distribution_Rollback Qualifiers

⇒ `Output Control qualifiers`
⇒ `Center Connection qualifiers`
⇒ `Edge Connection qualifiers`
⇒ `Synchronization qualifiers`
⇒ `Execution qualifiers`
⇒ `Exit Procedure qualifiers`

`-job=`**`job_id`**

Use this qualifier to specify the job that needs to be rolled back. Note that you can only roll back completed jobs.

`-distribution_name=`**`rollback_job_name`**

The `-distribution_name` qualifier defines the name of the distribution rollback job.

`-description=`**`distribution_job_description`**

The `-description` qualifier provides a description of the distribution rollback job.

`-rollback_to_time=`**`time_expression`**

Specify the date and time to recover from. This can be an absolute or a relative time.

NOTE: The time specified is calculated at the Target machine (Edge in case of an Upload job, Center if it's a Download job). It is never the Console time.

- **Absolute time on target** – Specify the exact time to roll back to. Format supports most standard time formats like: "ddmmyyyyhhmm". For a complete list of supported time formats please refer to **Appendix B – Absolute Time Expressions .**

  In case a rollback instance is not found at the exact time specify, R-1 will roll back to the nearest recorded data, without passing the time specified.

- **Relative time on target** – Specify how long to roll back to, in days / hours / minutes. Here, too, the calculated time to roll back from is calculated on the target machine. For relative time format, please refer to **Appendix B – Relative Time Expressions** .

- **Rollback all recorded content** – Using this option, the rollback process will bring the target directory to its state before the replication took place. To use this option, omit the `-rollback_to_time` qualifier from the command.

  NOTE: For non-scheduled distribution jobs, this is the only option available.

Absolute Time Example:

The following command is used to Rollback a distribution job to an absolute time. Data will be rolled back to 10:30 PM according to the Target machines (in this case the Edges):

```
>r1 distribution_rollback -job=3212 -distribution_name=”Rolling
back” -description=”Rolling back due to…” -center=center_name
-center_user=user_name –center_password=password
-user=user_name -password=password
-rollback_to_time=2230
R1API-S-SUS, job <4121> successfully submitted
```

NOTE: If you are using different credentials than those used in the original job, make sure there are sufficient access rights to the recording location and the target directory.

Relative Time Example:

The following command is used to Rollback a distribution job to a relative time. Data will be rolled back to its state 4 hours ago:

```
>r1 distribution_rollback -job=212 -distribution_name=”Rolling
back” -description=”Rolling back due to…” -center=center_name
-center_user=user_name –center_password=password
-user=user_name -password=password
-rollback_to_time=”-4 hours”
R1API-S-SUS, job <421> successfully submitted
```

5.6. Control Commands

R-1 Control commands enable manipulation over jobs.

5.6.1. Single Job Selection Qualifiers

`-job=`**n**

Select the job with the specified unique ID.

5.6.2. *Purge Command*

The **`purge`** command purges and clears completed jobs. This includes jobs completed successfully, aborted by user, or jobs that have failed. The command can be applied to multiple selected jobs.

Purge Qualifiers

⇒ `Output Control qualifiers`
⇒ `Center Connection qualifiers`
⇒ `Edge Connection qualifiers`
⇒ `Single Job Selection qualifiers`

`-confirm|-noconfirm`

Prompt the user to confirm a job's purge in **`y/n/a/q`** format:
y – Yes, purge the current job.
n – No, cancel purge of the current job.
a – All, purge all selected jobs.
q – Quit, cancel the purge operation.

`-log|-nolog`

If `-log` switch is specified, R-1 will write a log line for each purged job. The log line is written to the output specified by the Output Qualifiers.

`-state=`**`completed,failed,interrupted,aborted`**

Purge all jobs in the specified state.

`-name=`**`job_name`**

Purge all jobs that match the **`job_name`** criteria. This may include wildcards.

Example:

This example will purge all jobs that are in the “aborted” state:

```
>r1 purge -state=aborted -confirm
REBLIWEB-I-PURGE: job ID <3> (y/n/a/q)?  [y] y
REBLIWEB-I-PURGE: job ID <14> (y/n/a/q)?  [y] n
REBLIWEB-I-PURGE: job ID <75> (y/n/a/q)?  [y] n
Purged <1> jobs out of <1> requested
R1API-S-PJS, purge command successfully deleted <1> jobs
```

5.6.3. Delete Command

The **`delete`** command deletes a specific job listings/info from a Center. This command handles one job at a time. This command can clear completed and scheduled job.

Delete Qualifiers

⇒ Output Control qualifiers
⇒ Center Connection qualifiers
⇒ Edge Connection qualifiers
⇒ Single Job Selection qualifiers

`-edge_id=`**`n`**

Delete a completed Edge job.

Example:

```
>r1 delete -job=585
R1API-S-DJS, job <585> successfully deleted
```

5.6.4. Hold Command

The **`hold`** command can handle a specific job on the Center or on the Edge. This command handles one job at a time. Held jobs will hold indefinitely unless a resume time is specified, or the **`resume`** command is issued.

Hold Qualifiers

- ⇒ `Output Control qualifiers`
- ⇒ `Center Connection qualifiers`
- ⇒ `Edge Connection qualifiers`
- ⇒ `Single Job Selection qualifiers`

`-edge_id=`**`n`**

Hold an Edge job.

`-hold_until=`**`hold_date`**

The **`hold_date`** specifies when to resume the job from its held state. For a complete list of supported time formats please refer to **Appendix B – Absolute Time Expressions .**

Example:

```
>r1 hold -job=3307 -hold_until=130820051200
R1API-S-HJS, job <3307> successfully held
```

5.6.5. Resume Command

Used to resume jobs that are currently on hold. This command handles one job at a time.

Resume qualifiers

⇒ Output Control qualifiers
⇒ Center Connection qualifiers
⇒ Edge Connection qualifiers
⇒ Single Job Selection qualifiers

`-edge_id=`**n**

Resume a held Edge job.

Example:

```
>r1 resume -job=5
R1API-S-RJS, job <5> successfully resumed
```

5.6.6. Abort Command

Used to abort any scheduled or running jobs. This command handles one job at a time.

Abort qualifiers

- ⇒ `Output Control qualifiers`
- ⇒ `Center Connection qualifiers`
- ⇒ `Edge Connection qualifiers`
- ⇒ `Single Job Selection qualifiers`

`-edge_id=`**`n`**

Abort an Edge job.

`-reason=`**`abort_reason_text`**

The **`abort_reason_text`** will be displayed as the Completion Message of the job.

Example:

```
>r1 abort -job=1974
R1API-S-AJS, job <1974> successfully aborted
```

5.7. Monitoring Commands

5.7.1. Show Command

The **show** command is used for querying and monitoring R-1 jobs. As with any other command, the Center and user context must be defined either in the command line or from a previously saved user profile.

Show Qualifiers

⇒ Output Control qualifiers
⇒ Center Connection qualifiers

The following qualifiers allow selection of multiple jobs:

-state=**submitted,running,completed,failed,aborted,scheduled,hold, recovering**

Show all jobs in the specified state.

-job=**n**

Show the job with the specified unique ID.

-name=**job_name**

Show all jobs that match the **job_name** criteria. This may include wildcards.

-edge_id=**n**

When used with -query=**full**, includes information on the specified Edge's jobs.

NOTE: The report will include **ONLY** the Edge job information, without the Distribution Job's details.

-edge_list=**edge_id_list** **\"n,n,n\"**

Retrieve information on the specified Edge's jobs.

NOTE: -edge_list cannot be used with the -edge_id or -job qualifiers.

-edge=**edge_name**

When used with -query=**full**, includes information on the specified Edge's jobs in addition to the Distribution job's details.

-edge_state=\"**submitted,running,completed,failed,aborted,scheduled, hold,recovering\"**

When used with -query=**full**, includes information on Edges' jobs in the specified state, in addition to the Distribution job's details.

The following qualifiers specify the information to be displayed:

`-query=`**`basic|detailed|full`**

Specifies the information to be displayed on the job(s) specified in any of the Job Selection qualifiers **(`-job`**, **`-job_name`**, etc.)

`basic` - Show job basic information.
`detailed` - Show more detailed information.
`full` Show **`detailed`** information on the Center job and information on the Edge(s) jobs. The information on the Edge(s) jobs is dependant on the -edge_query qualifier.

Example:

With `-query`=**`basic`**, the command output will look like that:

```
------------------------------------------------------------
Distribution : 33
------------
Name                       : ImmediateJob
State                      : RUN
Stage                      : E-INFO
Central Source Dir         : g:\111
Transport Engine           : RepliWeb
Number of Edges            : 1 - 1 running, 0 completed, 0
aborted, 0 failed
Number of Multicast Edges : 0
Number of Unicast Edges    : 1
------------------------------------------------------------
R1API-S-SHS, successfully displayed <1> jobs
```

With -query=**detailed**, the command output will look like that:

```
-------------------------------------------------------------
Distribution : 33
------------
Name                       : ImmediateJob
State                      : RUN
Stage                      : E-INFO
Central Source Dir         : g:\111
Transport Engine           : RepliWeb
Number of Edges            : 1 - 1 running, 0 completed, 0
aborted, 0 failed
Number of Multicast Edges : 0
Number of Unicast Edges    : 1
Submitted                  : Tue Jun 18 10:50:34 2002
Scheduled                  : Not Scheduled
Schedule type              : Immediate
Edge                       : localhost
-------------------------------------------------------------
R1API-S-SHS, successfully displayed <1> jobs
```

With -query=**full**, the command output will look like that:

```
-------------------------------------------------------------
Distribution : 33
------------
Name                       : ImmediateJob
State                      : RUN
Stage                      : C-SNAPSHOT
Central Source Dir         : g:\111
Transport Engine           : RepliWeb
Number of Edges            : 1 - 1 running, 0 completed, 0
aborted, 0 failed
Number of Multicast Edges : 0
Number of Unicast Edges    : 1
Submitted                  : Tue Jun 18 10:35:06 2002
Scheduled                  : Not Scheduled
Schedule type              : Immediate
Edge                       : localhost
======================================================
Edge [localhost] : Job ID [34]
==============================
State                      : RUN
Stage                      : CRE SAT DIR
State message              :
Distribution               : 33
Destination directory      : g:\core_test
Total unicast copied       : 0 out of 0 files
======================================================
-------------------------------------------------------------
R1API-S-SHS, successfully displayed <1> jobs
```

-report=\”**general,transfer,pre_distribution,post_distribution\”**

Will display the specified report.

-edge_query=**basic|detailed**

When used with -query=**full**, includes information on the specified Edge’s jobs.
basic - Show job basic information.
detailed - Show more detailed information.

-edge_report=\”**general,transfer,pre_edge,post_edge,pre_center,
post_center,deleted_files,excluded_files\”**

Will display the specified report on the Edge. Accompanied with -edge_id the report will include ONLY the Edge’s report.

-viewer=**view_application**

This qualifier will pipe the output from the show command directly into another application.

Examples:

Show all jobs running on the Center:

```
>r1 show
```

Show detailed information about job’s status:

```
> r1 show -job=123 -query=detailed
```

Show detailed information about job’s status and all its Edges jobs that are in running state.

```
> r1 show -job=123 -query=full -edge_state=running
```

Appendices

Appendix A
Selecting a Compression Algorithm

Optimizing Compression to Accelerate Transfer Speed

If minimizing file transfer time is a major consideration, choose a compression algorithm using following criteria:

Generally, **Power** compression yields a higher compression ratio than other compression methods. However, **Power** compression takes a relatively long time per packet and, consequently, the overall transfer may take more time. This is likely if the CPU is slow, or if high-speed communications lines are being used. If so, try the **AP** or **Normal** methods.

File transfer time will probably be poor when compressing over LANs, although network load problems will improve. Unless there are bottleneck problems due to network overload, specify **None** as the compression method on LANs or other high-speed networks.

As a rule of thumb, compression will accelerate transfer on slow communication lines, i.e., when a transfer rate of up to 256 Kbps is allocated for a single distribution operation. At higher transfer rates, compression will slow down the transfer, although it will always reduce the volume of transferred data (unless the files were previously compressed). For example, T1 lines are used for many types of communication, including data communications, telephone, etc., assuming a slice of 512Kbps is allocated for TCP/IP on a T1 line. If a single operation is using the entire 512Kbps, throughput time will be reduced if compression is used. However, if two concurrent distribution operations are initiated on such a line, each operation has an effective throughput of 256Kbps, and compression will accelerate both of them.

File Types

The compression ratio also depends on the type of data to be transferred, such as text files, executable files, etc. Certain compression methods are particularly appropriate or inappropriate for specific data types:

- **Power** provides the highest degree of compression on executable files.
- **AP** usually yields the best compression ratio on data files.

Trying to compress already compressed files will result in high CPU usage but will not yield smaller files. It is recommended to avoid compression of the following file types:

```
.Z,.zip,.arj,.rar,.lzh,.arc,.tgz,.gz,.taz,.gif,.tif,.sit,.mpg,.mp
eg,.jpg
```

Users who are uncertain about the optimal compression method are advised to experiment with the different methods under typical system and network load conditions, using typical files.

Optimizing Compression to Reduce Network Load

Under certain circumstances, such as those described below, it may be appropriate to reduce the volume of data being transferred, regardless of the transfer speed:

- Using packet switching networks, public networks, or Satellite communication networks that bill according to the volume of data transferred and not the amount of time that the line is in use.
- Performing massive data transfer operations that overload the network, causing poor response time for other network users or applications.

Users wishing to reduce network load should select the compression method yielding the highest degree of compression for the specific files to be replicated, as explained above.

Appendix B
Bandwidth Time Frame Expressions & Time Formats

Bandwidth Limit

Absolute Bandwidth Limit

This represents an absolute numeric value for the maximum bandwidth usable by the transfer, and specified as a number (the number can be a decimal fraction, such as 7.6 or 9.45) followed by two letters to denote the units used: K (for kilo) or M (for mega) followed by B (for bytes) or b (for bits):

nnnn.nnnn[K or M][B or b]

Examples:

256 KB = 256 kilobytes per second
1.6Mb = 1.6 megabits per second

The default is bits per second, so:

256 = 256 bits per second
256K = 256 kilobits per second

Relative Bandwidth Limit

This value defines the maximum percentage of available bandwidth available to the transfer. A relative bandwidth limit varies in relation to the line's current load. It is specified as an integer between 1-100 followed by the percentage symbol **%**:

nn%

Time Expressions

A bandwidth expression is used for applying varying bandwidth limitations to specific time frames.

The bandwidth qualifier accepts a list of this style (multiple bandwidth limit values with corresponding time frame expressions) allowing specification of different bandwidths applicable to different time frames within a single transfer operation.

Example:

A single large-scale operation will use no more than 25% of the line between 8:00 (8AM) to 17:00 (5 PM). After 17:00 a less restrictive limit is to be implemented, permitting the process to use no more than 85% of the line.

The way to write this is:

```
25% h:08-17, 85% h:17-24
```

Or:

```
100KB h:08-22, 400KB h:22-24
```

Absolute Time Expressions

R-1 uses the following formats to specify dates and times:

- **For time:**
  - *hhmm* - hours (two digits) and minutes (two digits).

    The current date (today) is assumed.

- **For date:**
  - *ddMMyy* - day of the month (two digits), month (two digits) and year (two digits).
  - *ddMMyyyy* - day of the month (two digits), month (two digits) and year (four digits).

    The time is assumed to be midnight on the specified date.

- **For time and date:**
  Various combinations of date and time expressions, specifying an absolute date and time:
  - *ddMMyyhhmm*
    ddMMyyyyhhmm
    ddMMyyyyhhmmss – specify seconds ad well.

The **today** keyword can also be used. If no time is specified with a qualifier requiring a time expression, the time is assumed to be midnight and the date is assumed to be today.

Relative Time Expressions

Sometimes it is more useful to specify a time expression as relative to the current time rather than in absolute terms.

Relative time expressions are specified as values that will be added or subtracted from the current time to obtain the final value. They consist of one or more terms composed of a sign (+ or -), a numeric value and a keyword indicating the units. The following units can be used:
years, months, weeks, days, minutes, hours or **seconds**.

A space must be included between the value and the units keyword, so a relative time expression must always be enclosed in quote marks.

If no sign is included in a term, the default sign is +. Each term in an expression is added or subtracted (according to its sign) from the current time when the expression is evaluated (during execution).

Examples:

Anything over six months old:

```
"-6 months"
```

Anything that is dated 3 weeks, 4 days and 7 hours before the current time:

```
"-3 weeks -4 days -7 hours"
```

NOTE: If a sign is not used specifically, the term has a sign of +, so that:

```
"-5 weeks 4 days"
```

Means five weeks less 4 days ago and not 5 weeks and 4 days ago.

It is the same as the expression:

```
"-5 weeks +4 days"
```

Or the expression:

```
"-4 weeks -3 days"
```

Setting Time Frames

Certain qualifiers can accept **Time Frame Expressions** as values. These expressions specify when the qualifier should be in effect. A simple **Time Frame Expression** includes a range of times and a character identifying their type, such as hours, months, or days of the week. A "range" may be specified which includes only a single value, thus the user may omit either the upper or lower limit, specifying that the qualifier should take effect only until a certain time or from a certain time. Alternatively, a list of values instead of a range can also be used.

Valid time frame types include:

y - years
M - months (valid values range from 1 to 12)
d - days of the month (valid values range from 1 to 31)
w - days of the week (valid values range from 1 to 7)
h - hours (valid values range from 1 to 24)
m - minutes (valid values range from 1 to 60)
s - seconds (valid values range from 1 to 60)
t - time; uses 2, 4, 6 or 10-digit format and can replace h, m and s:
    2 digit format = hours;
    4 digit format = hours & minutes;
    6 digit format = hours, minutes & seconds;
    10 digit format = day of the month, month, hours, minutes, seconds.

Simple **Time Frame Expressions** can be combined into compound expressions by stringing them together.

- When an ampersand (&) mark is placed between two expressions, both expressions are taken together (this is equivalent to a logical AND operation).
- If there is nothing between the two expressions, then one or the other apply (this is equivalent to a logical OR operation).
- Parentheses () can be used to enclose expressions, indicating that the expression in the parentheses is a single expression.
- An expression can also be defined as an exception. Adding an exclamation mark (!) before a given expression, specifies any time except that within the time frame described by that expression (this is equivalent to a logical NOT operation).

If an hours range is specified with an upper limit that is smaller then the lower limit, for example, h:21-6, the upper limit is assumed to be on the next day, so that h:21-6 is interpreted as a time frame starting at 9 PM in the evening and ending at 6 AM the following morning.

Examples:

Monday through Friday 8 AM-5 PM:

```
(w:2-6)&(h:8-17)
```

Same as above but excluding both New Year's Day and Christmas:

```
(w:2-6)&(h:8-17)&!((M:1)&(d:1) (M:12)&(d:25))
```

Every other day from 9pm and on:

```
(w:2 4 6)&(t:2100-)
```

Every day before 8am and after 5pm:

```
(h:-8 17-)
```

Appendix C
Qualifiers

Below is a list of qualifiers available with the Command Line Interface (CLI). The complete description of the commands and qualifiers may be found in **Chapter 5, Creating a Distribution Job**.

```
r1 <command> [-qualifiers] [parameter]
```

NOTE: It is **EXTREMELY** important to remember that all path locations and commands **MUST** be written in a native format, i.e., Windows commands and path names written in a Windows format; UNIX commands and path names using UNIX format. This is especially crucial when replicating cross-platform jobs.

See **Cross Platform Formats and Escaping** description in **Chapter 5**.

Common Qualifiers

Output Control Qualifiers

```
-output=output_file_name
-output_mode=new|append
-output_style=tagged
-silent
```

Center Connection Qualifiers

```
-center=center_id
-center_user=user_name
-center_password=password
-scrambled_center_password=scrambled_password
-center_domain=domain
-center_ssl|-nocenter_ssl
```

Help Command

⇒ Output Control qualifiers

Information Command

```
⇒ Output Control qualifiers
⇒ Center Connection qualifiers
```

-edge=**edge_name**

-edge_user=**user_on_edge**

-edge_password=**password_on_edge**

-edge_domain=**domain_name**

Set Command

```
⇒ Output Control qualifiers
⇒ Center Connection qualifiers
```

Scramble Command

```
⇒ Output Control qualifiers
```

-password=**input_password**

Submit Command

```
⇒ Output Control qualifiers
⇒ Center Connection qualifiers
```

-modifier=**file_name**

edge_override
edge_default
pre_distribution_download

NOTE: See **Appendix F** for the complete explanation of the commands and the list of qualifiers that can be used in the modifier file commands.

-type=**distribution|replication**

Edge Connection Qualifiers

-user=**user_on_edge**

-password=**password_on_edge**

-domain=**domain_name**

Job Definition Qualifiers

-distribution_name=**distribution_job_name**

-description=**distribution_job_description**

-source_directory=**distribution_source_directory**

-target_directory=**distribution_target_directory**

NOTE: The qualifiers -user,-password,-domain, and -target_directory are positional. If you specify them before the target nodes list, they apply to all nodes. If specified for a specific node, they apply to that node only.

Distribution Qualifiers

-distribution_type=**unicast|multicast**

-continue_on_empty_source|-nocontinue_on_empty_source

-unicast

-use_central_lists=**snapshot_files_directory**

-cross_platform_indication=**auto_detect|windows_only|unix_only|
cross_platform**

Controlling Success

-all_edges_required|-noall_edges_required

-min_nodes=**n**

-min_nodes_optimization|-nomin_nodes_optimization

Multicast

-mc_engine=**rwmc|rwdf**

-mc_min_quorum=**n**

-unicast_edge_failover|-nounicast_edge_failover

-unicast_file_failover|-nounicast_file_failover

-mc_quorum_max_wait=**n** **in seconds**

-mc_maximize_time=**n** **in %**

-mc_file_quorum=**n**

-mc_min_file_size=**n** **in KB**

-mc_max_file_size=**n** **in KB**

-mc_buffer_size=**n** **in bytes**

-mc_pack_files|-nomc_pack_files

-mc_max_pack_size=n **in MB**

-mc_max_pack_buffer_size=n **in Bytes**

-mc_pack_compress|-nomc_pack_compress

-mc_packet_retry_factor=**n**

R-1 Multicast Engine

-mc_group=**mc_group_address**

-mc_port=**n**

-mc_ttl=**n**

-mc_mc_send_repetitions=**n**

-mc_max_transfer_rate=**n** **in MB|Mb|KB|Kb / seconds**

Multicast in the Edge

-mc_server_timeout=**n** **in seconds**

Synchronization Qualifiers

-sync_before_pre_command|-nosync_before_pre_command

-sync_after_pre_command|-nosync_after_pre_command

-sync_before_post_command|-nosync_before_post_command

Recovery Qualifiers

-max_retries=**n**

-base_interval=**n** **in seconds**

-retry_factor=**n** **in percent**

-max_interval=**n** **in seconds**

Execution Qualifiers

-pre_center_command=**command_to_run**

-pre_center_parameters=**command_parameters**

-post_center_command=**command_to_run**

-post_center_parameters=**command_parameters**

Exit Procedure Qualifiers

NOTE: Exit is not executed between recovery attempts. It is executed only if this job will never run again.

-success_exit_procedure=**exit_procedure_name**

-success_exit_parameters**=exit_procedure_parameters**

-success_email_recipients=**email_address**

-success_user_attachment=**user_files_to_be_attached**

-success_email_include_reports=\"**general,transfer,pre_distribution, post_distribution\"**

-abort_exit_procedure=**exit_procedure_name**

-abort_exit_parameters=**exit_procedure_parameters**

-abort_email_recipients=**email_address**

-abort_user_attachment=**user_files_to_be_attached**

-abort_email_include_reports=\"**general,transfer,pre_distribution, post_distribution\"**

-error_exit_procedure=**exit_procedure_name**

-error_exit_parameters=**exit_procedure_parameters**

-error_email_recipients=**email_address**

-error_user_attachment=**user_files_to_be_attached**

-error_email_include_reports=\"**general,transfer,pre_distribution, post_distribution\"**

-event_report_on=**success,abort,error**

NOTE: For e-mail to be sent as part of the Exit procedure, in the -success_exit_procedure, -abort_exit_procedure and -error_exit_procedure qualifiers the **exit_procedure_name** must be **__MAIL__.**

Only in that case, e-mail will be sent according to the appropriate Job status and the e-mail qualifiers (recipient, attachments, reports.)

The following qualifiers are used to operate on the list of transferred files. The transfer reports are parsed according to a template file. The output file may then be executed or email.

NOTE: For file list parsing to be performed, in the -success_exit_procedure, -abort_exit_procedure and -error_exit_procedure qualifiers the **exit_procedure_name** must be **__FULL_FILE_LIST__.**

Only in that case, the transfer log will be parsed according to the template file.

```
-success_file_list_template_file=success_file_list_template_name
-success_file_list_target_directory=success_file_list_target_directory
-success_file_list_action=none|email|execute
-success_file_list_recipients=success_file_list_email_address
-abort_file_list_template_file=abort_file_list_template_name
-abort_file_list_target_directory=abort_file_list_target_directory
-abort_file_list_action=none|email|execute
-abort_file_list_recipients=abort_file_list_email_address
-error_file_list_template_file=error_file_list_template_name
-error_file_list_target_directory=error_file_list_target_directory
-error_file_list_action=none|email|execute
-error_file_list_recipients=error_file_list_email_address
```

Scheduling Qualifiers

```
-run_option=immediate|once|every|daily|weekly|triggerfile|filespec|
on_demand|continuous_update
-trigger_file=trigger_file_name
-once_time=schedule_date
-every=n                      in minutes
-daily=schedule_time
-weekly=weekly_expression
-terminate_after=n            in minutes
-Priority=n
-synchronize_every=n          in minutes
-synchronize_daily=schedule_time
-max_synchronize_delay=n      in seconds
-max_archived_instances=n
```

Purge Qualifiers

```
-purge_on_success=n           in minutes
-purge_on_error=n             in minutes
-purge_on_abort=n             in minutes
```

Input Template Qualifiers

```
-in_template=template-name
-in_template_location=local|center_user|center_common
```

On Demand Qualifiers

-behalf_users=**users_list**

-behalf_groups=**groups_list**

Edge_Override & Edge_Default Commands

NOTE: Qualifiers that are listed as Edge Default and Edge Modifier qualifiers, can be used **ONLY** in the modifier file, and may not be used as qualifiers in the **submit** command line itself.

Edge Default Modifiers Qualifiers

NOTE: The following qualifiers can be specified in the modifier file with the **edge_default** command ONLY!

⇒ Scheduling Qualifiers
⇒ Purge Qualifiers

File Qualifiers

-exclude=**file_type**

-recursive_exclude=**file_type**

-snapshot_confirm|-nosnapshot_confirm

-all_files|-noallfiles

-file_specs=**file_type**

-ignore_attribute_changes |-noignore_attribute_changes

UDL Qualifiers

-user_list=**file_name**

-list_source=**console|controller_runtime|contoller_submit**

-use_user_list_on_target|-nouse_user_list_on_target

Links Qualifiers

-links=**link|file|ignore**

-dir_links=**ignore|link|directory**

-follow_links|-nofollow_links

Permission Qualifiers

-preserve_ntfs_attributes|-nopreserve_ntfs_attributes

-preserve_alternative_data_streams=**none|mac_resources**

-preserve_root_permission|-nopreserve_root_permission

UNIX machines only

-preserve_permission=**protection,suid,sgid**

-directory_permissions=**dir_permission**

-directory_permissions_on_creation|-nodirectory_permissions_on_creation

-copy_dir_permissions_on_creation|-nocopy_dir_permissions_on_creation

-root_permissions=**root_permission**

-files_permissions=**file_permission**

-preserve_ownership=**gid|uid**

-extended_attributes=**LINUX_Descriptor**

Edge Default and Edge Override Modifiers Qualifiers

NOTE: The following qualifiers can be specified either with the **edge_default** or with the **edge_override** commands:

-scrambled_password=**hashed_password_on_edge**

Performance\Throughput Qualifiers

-tree_recurse|-notree_recurse

-depth_limit=**n**

-use_snapshot_compression

-snapshot_bandwidth=**bandwidth_expression**

-user_list=**file_name**

-list_source=**center_runtime**

-abort_on_missing_file|-noabort_on_missing_file

-duplicate_files_handling**=error_exit|newest|oldest|
lexicographic_first|lexicographic_last**

-abort_on_delete_error|-noabort_on_delete_error

-abort_on_makelist_error|-noabort_on_makelist_error

-abort_on_transfer_error|-noabort_on_transfer_error

-reduce_controller_snapshot_resources=**n**

```
-target_directory=target_directory_name
-delete_before_transfer|-nodelete_before_transfer
_purge_files_only|-nopurge_files_only
```

Recovery

```
-max_retries=n
-base_interval=n              in seconds
-retry_factor=n               in percent
-max_interval=n               in seconds
```

Reporting Qualifiers

```
-report_style=total|logs|summary|detailed
```

Replication Level Time Tolerance Qualifiers

```
-logic=mirror|backup|purge
-minimal_purge_age=n                      in seconds
-tree_valid_for=tree_valid_time           in HHMM
-cmds_valid_for=cmd_valid_time            in HHMM
```

Transfer Integrity Qualifiers

```
-compare_after_transfer|-nocompare_after_transfer
-ssl|nossl
-overwrite_after_completion|-nooverwrite_after_completion
-satellite_transfer_temp_dir=temp_directory_name
-rename_hook=hook_filename
-rename_hook_param=hook_parameters
-copy_shares|-nocopyshares
-ignore_hidden|-noignore_hidden
-ignore_windows_invalid_file_names|-noignore_windows_invalid_file_names
-file_time_tolerance=n        in seconds
-reduce_satellite_snapshot_resources=n
-compression=none|normal|ap|power
-dont_compress=file_name
```

Transfer Qualifiers

```
-bandwidth=n                  in MB|Mb|KB|Kb|%
-transfer_streams=n
-differential_transfer|-nodifferential_transfer
-differential_min_size=n      in bytes
-unicast_engine=rw_lan|rw_wan|df
```

Execution Qualifiers for a Specific Edge Distribution

```
-pre_controller_command=command_to_run
-pre_controller_parameters=command_parameters
-pre_satellite_command=command_to_run
-pre_satellite_parameters=command_parameters
-post_controller_command=command_to_run
-post_controller_parameters=command_parameters
-post_satellite_command=command_to_run
-post_satellite_parameters=command_parameters
-pre_command_timeout=n        in seconds
-post_command_timeout=n       in seconds
```

Wait Settings Qualifiers

```
-wait
-wait_interval=n              in seconds
-wait_timeout=n               in minutes
```

TCP Communication Qualifiers

```
-connect_timeout=n            in seconds
-send_timeout=n               in seconds
-receive_timeout=n            in seconds
-lan_connect_timeout=n        in seconds
-lan_send_timeout=n           in seconds
-lan_receive_timeout=n        in seconds
-wan_connect_timeout=n        in seconds
-wan_send_timeout=n           in seconds
-wan_receive_timeout=n        in seconds
```

Rollback Definition Qualifiers

```
-rb_enable|norb_enable
-rb_location=recording_folder
-rb_retain=n                  in minutes
-rb_compress|-norb_compress
```

Demand_Submit Command

```
   ⇒ Output Control qualifiers
   ⇒ Center Connection qualifiers
```

```
-job=n
-name=job_name
```

Resubmit Command

```
   ⇒ Output Control qualifiers
   ⇒ Center Connection qualifiers
   ⇒ Edge Connection qualifiers
   ⇒ Job definition qualifiers
   ⇒ Scheduling qualifiers
   ⇒ Distribution qualifiers
   ⇒ Synchronization qualifiers
   ⇒ Recovery qualifiers
   ⇒ Execution qualifiers
   ⇒ Exit Procedure qualifiers
   ⇒ Purge qualifiers
```

```
-job_id=n
-modifier=file_name
      edge_override
      edge_default
      pre_distribution_download
```

NOTE: See **Appendix F** for the complete explanation of the commands and the list of qualifiers that can be used in the modifier file commands.

```
-type=distribution|replication
```

Replication_Rollback Command

```
⇒ Output Control Qualifiers
⇒ Center Connection Qualifiers
⇒ Edge Connection qualifiers
⇒ Execution Qualifiers
⇒ Exit Procedure qualifiers
```

-job=**job_id**

-replication_name=**rollback_job_name**

-description=**rollback_job_description**

-rollback_to_time=**time_expression**

Distribution_Rollback Command

```
⇒ Output Control Qualifiers
⇒ Center Connection Qualifiers
⇒ Edge Connection qualifiers
⇒ Execution Qualifiers
⇒ Exit Procedure qualifiers
```

-job=**job_id**

-distribution_name=**rollback_job_name**

-description=**rollback_job_description**

-rollback_to_time=**time_expression**

Purge Command

```
⇒ Output Control qualifiers
⇒ Center Connection qualifiers
⇒ Edge Connection qualifiers
```

-confirm|-noconfirm

-log|-nolog

-state=**completed,failed,interrupted,aborted**

-name=**job_name**

Single Job Selection Qualifiers

-job=**n**

Delete Command

⇒ Output Control qualifiers

⇒ Center Connection qualifiers

⇒ Edge Connection qualifiers

⇒ Single Job Selection qualifiers

-edge_id=**n**

Hold Command

⇒ Output Control qualifiers

⇒ Center Connection qualifiers

⇒ Edge Connection qualifiers

⇒ Single Job Selection qualifiers

-edge_id=**n**

-hold_until=**hold_date**

Resume Command

⇒ Output Control qualifiers

⇒ Center Connection qualifiers

⇒ Edge Connection qualifiers

⇒ Single Job Selection qualifiers

-edge_id=**n**

Abort Command

⇒ Output Control qualifiers

⇒ Center Connection qualifiers

⇒ Edge Connection qualifiers

⇒ Single Job Selection qualifiers

-edge_id=**n**

-reason=**abort_reason_text**

Show Command

⇒ Output Control qualifiers

⇒ Center Connection qualifiers

Multiple Job Selection

-state=**submitted,running,completed,failed,aborted,scheduled,hold, recovering**

-job=**n**

-name=**job_name**

-edge_id=**n**

-edge_list=**edge_id_list**   **\"n,n,n\"**

NOTE: -edge_list cannot be used with the -edge_id or -job qualifiers.

-edge=**edge_name**

-edge_state=\"**submitted,running,completed,failed,aborted,scheduled, hold,recovering\"**

Reporting

-query=**basic|detailed|full**

-report=\"**general,transfer,pre_distribution,post_distribution\"**

-edge_query=**basic|detailed**

-edge_report=\"**general,transfer,pre_edge,post_edge,pre_center, post_center,deleted_files,excluded_files\"**

-viewer=**view_application**

Appendix D
Center Administration

Configuring R-1 To Work Through Firewalls

To access a Center separated from a Console by a firewall, registered TCP port 2837 must be configured to accept an inbound session from the Console to the Center.

NOTE: If the Console and Center are on the same machine, no firewall ports need to be opened.

The file transfer itself (Center ←→ Edge) is performed over registered TCP ports 2837 and 5745, depending on the selected transport method (5745 for advanced WAN transport, 2837 for LAN transport). Since the Center initiates every communication, only an outbound session from the Center to the Edge on ports 2837 and 5745 must be opened. This remains true regardless of the direction of the data flow, i.e. a download job pulling files from an Edge to a Center is still a single outbound TCP session from the Center to the Edge.

The exact procedure for opening this port is firewall dependant and should be included in your firewall documentation.

In a more complex topology one needs only ensure that the respective ports are opened for Center ←→ Edge and Center ←→ Console communications.

Appendix E
User Defined Lists (UDL)

Using User Defined Lists (UDL)

Using UDL is intended for users required to specify which specific files from the source tree will be distributed. The UDL is a text file supplied by the user containing the names of these files (relative to the replication source root). The file can either be edited by the user or generated by an outside process.

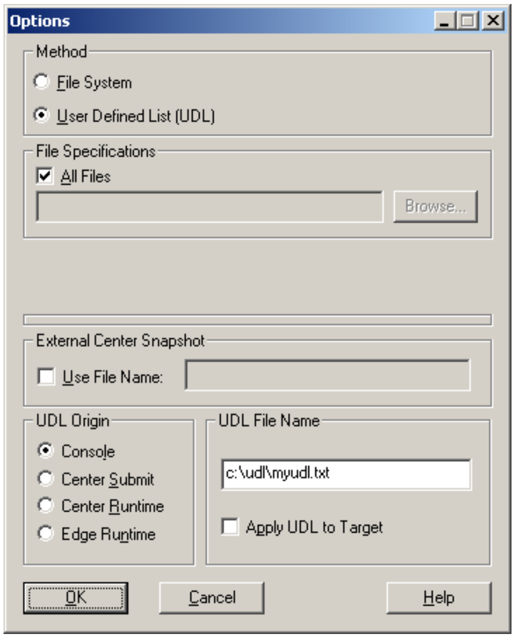
- Each hierarchic level (files and directories that are under the same directory) should be sorted in lexicographic order (doesn't matter if it is directory or a file).
- Path should be relative to the source directory, no need to put '/' or '\' in the beginning of the path.
- Each line in the list represents a single file or directory. Wildcards cannot be used.
- Each file and directory should be specified only once in the list.
- Only files to be transferred (if different from the target) should be specified in the list.
- The directories that should be included in the list are only those that their sub tree contains a file that should be transferred.
- A directory should be specified in the list only after all its files and subdirectories.
- Using of '\' or '/' must be according to source file system type.

UDL Scope

The UDL can be applied either to the source or to both source and targets.

If the UDL is applied to the source, it is used only on the source tree. A regular snapshot is done on the target tree. Using this with the Mirror logic will cause the entire target to become the mirror of the UDL. This means that files that are not in the UDL will be removed from the target and files that are part of the UDL will be synchronized. This is good when you want to use the UDL to override the entire target.

If the UDL is applied to both source and target, only the files that are part of the UDL are synchronized. The UDL acts in this case as a 'filter' and target files that are not part of the UDL are not touched in any way.



Options

Method

○ File System

● User Defined List (UDL)

File Specifications

☑ All Files

Browse...

External Center Snapshot

□ Use File Name:

UDL Origin

● Console

○ Center Submit

○ Center Runtime

○ Edge Runtime

UDL File Name

c:\udl\myudl.txt

□ Apply UDL to Target

Figure 57: Options Tab – Using UDL

In Distribution jobs, UDL values are specified in the **Job Properties / Snapshot Tab**.

- **UDL Origin** – If the file list is taken from a UDL, there are four choices for the origin of the User list. These values state where and when the list will be taken from. Although the absolute path of the list will be from the field **UDL File Name**, the system (Console / Center / Edge) must be selected.
  - **Console -** The list will be copied from the system the Console is being run from at the time the job is submitted to the Controller. If the job is scheduled, the same list will be used for all subsequent jobs.
  - **Center Submit -** The list will be taken from the Center at the time the job is submitted. Any changes made to the list between the time the job is submitted, and the time the job is actually run will be ignored. If the job is scheduled, the same list will be used for all subsequent jobs.
  - **Center Runtime -** The list will be taken from the Center at the time the job actually runs. Since a scheduled job spawns a Satellite Job to run at each scheduled time, each scheduled job will use the list as is when it is spawned.
  - **Edge Runtime -** The list will be taken from the Edge at the time the job actually runs. This option is available for Replication (Upload / Download) Jobs only.

- **UDL File Name** – Specify the name (including full path) of the UDL file to be used. The file must reside on the Console, Center or Edge, according to the UDL Origin specified.

- **Apply UDL to Target** – The use of a UDL files on the source machine results in greatly reduced system resources during the snapshot generation stage of the replication. This is because the snapshot is assumed to be the UDL itself. However, on the target machine, a standard snapshot is still issued. When this option is enabled, the UDL is transferred to the target and only the files present in this list are examined for existence and modification. This greatly reduces the associated costs of the target snapshot for large content repositories.

The UDL is like an enhanced include/exclude list; selecting to apply UDL to target, will skip the scanning of the target directory and apply the list instead. That way, only the files listed in the UDL are handled during the replication phase, all other files on target stay intact, as they are excluded from the replication. Using the UDL on the Replication Source only, the target directory is scanned, all UDL files are transferred, and the Target is changed according to the Replication Logic selected.

Creating a UDL

To create a UDL, edit a .txt file.

The list must contain the names and extensions of all files that you wish to distribute. To distribute files located within subdirectories of the relative root, you must specify the location of the requested file beginning at the relative root and including all folders in which the file resides.

Example:

To replicate the following files:

```
a.a
b.b
subdir1/a.1
subdir1/a.2
subdir1/subsub/c.c
subdir2/b.1
```

The list should read:

```
a.a
b.b
subdir1/a.1
subdir1/a.2
subdir1/subsub/c.c
subdir1/subsub
subdir1
subdir2/b.1
subdir2
```

NOTE:

- The list must be in increasing lexicographic order.
- Any subdirectories must be specified immediately after the files contained therein.
- All paths are relative to the path entered in the **`Path`** field.

CLI UDL Qualifiers

The UDL options are controlled by qualifiers of the **`edge_default`** command. These qualifiers cannot be specified at the **`edge_override`** command. All qualifiers can also be set using the Console GUI.

`-user_list=`**`file_name`**

`file_name` is the name of the UDL file name.

`-list_source=`**`console|controller_runtime|contoller_submit`**

The UDL can be prepared on the Console machine, and passed at the time of submit or it can be specified at the time of submit to be taken from the Center. There are three options for the origin of the `-user_list`. These values state where and when the list will be taken from.

`console` - The list will be taken from the Console machine at submit time.
`controller_submit` - The list will be taken from the Center at the time the job is submitted. Any changes made to the list between the time the job is submitted, and the time the job is actually run will be ignored.
`controller_runtime` - The list will be taken from the Center at the time the job actually runs.

`satellite_runtime` The user list will be taken from the Edge machine at runtime. This option is available only for Replication Upload and Download jobs (`-type=`**`replication`**).

`-use_user_list_on_target|-nouse_user_list_on_target`

Applying UDL to target machine. Specify whether the UDL is replacing the Tree Snapshot on Source only or also on Target. The UDL is like an enhanced include/exclude list; selecting to apply UDL to target, will skip the scanning of the target directory and apply the list instead. That way, only the files listed in the UDL are handled during the replication phase, all other files on target stay intact, as they are excluded from the replication. Using the UDL on the Replication Source only, the target directory is scanned, all UDL files are transferred, and the Target is changed according to the Replication Logic selected.

`-abort_on_missing_file|noabort_on_missing_file`

Handling missing files, specified in the UDL. By default, if R-1 tries to transfer a file specified in the list that cannot be found, it will skip the file and exit successfully. Using `-abort_on_missing_file` will have R-1 abort the job, exiting with an error.

Appendix F
Modifier Commands & Qualifiers

The **`Submit`** command submits a defined distribution job to the Center for distribution along with all of its relevant qualifiers.

The specific performance of several or all Edges can be controlled with a modifier file. This is a file that contains commands to specify default behavior for all Edges and override that behavior for one or several Edges.

This chapter will list the commands and qualifiers that may be used within that file.

The file format is the following:

```
# A comment line
{A blank line}
edge_default set [-qualifiers]
edge_override edge1[,edge2,,] mod [-qualifiers]
pre_distribution_download set [-qualifiers]
```

NOTE: Qualifiers that are listed as Edge Default and Edge Modifier qualifiers, can be used **ONLY** in the modifier file, and may not be used as qualifiers in the **`submit`** command line itself.

Modifier Commands

Edge_Default Command

Use the **`edge_default`** command to set default behavior for all Edges.

```
edge_default set [-qualifiers] [-switch]
```

Edge_Override Command

Use the **`edge_override`** command to override specific behaviors for some or all the Edges.

```
edge_override edge1[,edge2,,] mod [-qualifiers] [-switch]
```

Pre_Distribution_Download Command

Use the **`pre_distribution_download`** command to specify the qualifiers of the download job that should take place prior to the distribution process. Only one job can be defined as a Pre-Distribution Download Job.

NOTE: The command may be used only once in the modifier file.

`Pre_distribution_download set [-download] [-qualifiers] [-switch]`

The command accepts all Modifier Qualifiers. There are no limits on the qualifiers, as long as the result is a valid Download job.

NOTE: The qualifier `-download` is mandatory.

Example:

```
pre_distribution_download set –unicast_engine=rw_wan
-source_directory=srcdir -target_directory=updatecontent
-satellite=comp_server -user=username -pass=password -download
```

Modifier Qualifiers

Edge Default Modifiers Qualifiers

NOTE: The following qualifiers can be specified in the modifier file with the **`edge_deault`** command **ONLY**!

File Qualifiers

For more information and samples on excluding files and directories, please refer to **Appendix K**.

`-exclude=`**`file_type`**

File types that match the criteria in **`file_type`** should be completely ignored during distribution.

`-recursive_exclude=`**`recursive_file_type`**

The following file types should be recursively ignored during distribution. This means any files or directories matching the **`recursive_file_type`** criteria will not be deleted or transferred.

Example:

This example will exclude the `*.logs` file type from the relative root of distribution as well as the file `active.log`. Distribution will also ignore all directories beginning with `__vti__`. Notice that entire directories to be excluded must end with a backslash. Since this command is being run from the CLI, it must be properly escaped.

```
-exclude=\"*.logs,active.log\"
-recursive_exclude=\"__vti__*\\\\"
```

`-snapshot_confirm|-nosnapshot_confirm`

Specify to pause the replication job after the planning stage. The job will enter a Preview state. To preview and change the files that are about to be transferred and deleted during the replication phase and resume the job, use the Console GUI.

`-all_files|-noall_files`

Synchronize all files during the distribution process.

`-file_specs=`**`file_type`**

Synchronize only the following types of files specified in **`file_type`**. These can include individual files, wildcards, or entire subdirectories. Must be used with `–noall_files`.

Example:

```
Edge_default set … -noall_files
-file_specs=\”subDir1\\*,*.doc,budget.xls\”
```

NOTE: `-all_files` and `-file_specs` are mutually exclusive. Also, when a file specification is used to limit the files to be distributed, R-1 examines only those files of the type specified. Files that do not meet the file specifications will not be transferred, created, or deleted on the source and target systems.

`-ignore_attribute_changes |-noignore_attribute_changes`

In a Continuous Update job, do not replicate attribute changes if that is the only change that occurred. To be used only with `-run_option=`**`continuous_update.`**

UDL Qualifiers

`-user_list=`**`file_name`**

`file_name` is the name of the UDL file name.

`-list_source=`**`console|controller_runtime|contoller_submit`**

The UDL can be prepared on the Console machine, and passed at the time of submit or it can be specified at the time of submit to be taken from the Center. There are three options for the origin of the `-user_list`. These values state where and when the list will be taken from.

`console` - The list will be taken from the Console machine at submit time.
`controller_submit` - The list will be taken from the Center at the time the job is submitted. Any changes made to the list between the time the job is submitted, and the time the job is actually run will be ignored.
`controller_runtime` - The list will be taken from the Center at the time the job actually runs.

`-use_user_list_on_target|-nouse_user_list_on_target`

Applying UDL to target machine. Specify whether the UDL is replacing the Tree Snapshot on Source only or also on Target. The UDL is like an enhanced include/exclude list; selecting to apply UDL to target, will skip the scanning of the target directory and apply the list instead. That way, only the files listed in the UDL are handled during the replication phase, all other files on target stay intact, as they are excluded from the replication. Using the UDL on the Replication Source only, the target directory is scanned, all UDL files are transferred, and the Target is changed according to the Replication Logic selected.

Links Qualifiers

Use the following qualifiers when the source directory is on a UNIX machine only.

`-links=`**`link|file|ignore`**

This qualifier is only applicable when the source is a UNIX machine.

`link` creates a link on the target [identical to the link on the source]
`file` creates the file the link on the source points to on the target.
`ignore` ignores all links in transfer.

`-dir_links=`**`ignore|link|directory`**

The `-dir_links` qualifier deals with links to directories. This qualifier is only applicable when the source is a UNIX machine.

`link` - Creates a link to the directory on the target machine.
`directory` - Creates a real directory on the target.
`ignore` - Ignores links to directories.

`-follow_links|-nofollow_links`

`-follow_links` relates to the snapshot generation phase. It determines whether R-1 follows links when building its source tree snapshot. R-1 will follow on the replication source the directory link and include the files from that linked directory in its snapshot as files that reside in a directory with the link name. Links to directories on the replication target are not followed but rather listed as links. The default is to not follow the directory links.

Example:

If one uses `-follow_links` and the source tree - `/docs/tree` has a link `/docs/tree/sub1/sub2/linkToTmp` which points to `/tmp` (a root completely outside of your tree), the snapshot result will include the entire content of `/tmp under /docs/tree/sub1/sub2/linkToTmp` (`/tmp/a.a` will appear in your snapshot as `/docs/tree/sub1/sub2/linkToTmp/a.a`).

If there is a link to a file in your tree `/qa/tree/sub1/file_link` which points to `/tmp/b.b` it will appear on your list `as /qa/tree/sub1/file_link` with an indication that it is a link and will be included regardless of the -follow qualifier.

Permission Qualifiers

Available options depend on the operating systems taking part in the replication process (Windows, UNIX). Preserving permissions is only possible when the source and target are of the same nature.

`-preserve_ntfs_attributes|-nopreserve_ntfs_attributes`

The `-preserve_ntfs_attributes` qualifier will preserve all NTFS permissions. If a file of the same name exists on the target, the data and the permissions will be overwritten with those of the source file. Naturally, if the distribution target is a FAT drive, NTFS attributes will not be retained.

If this option is not selected, the file data will be transferred but will assume the permissions of either a preexisting file of the same name as the file being transferred, or assume the default permissions for new files created by the logged-in user (for the Center, this is the user that connected via the Center Settings window; for the Edge, this is the user whose name and password were entered in the Job Definition window).

The default setting of this field does not preserve the NTFS attributes. It is especially important to consider the implications of this when distributing within domains.

-preserve_alternative_data_streams=**none|mac_resources**

Use this qualifier to specify whether to replicate Macintosh fork resource files.
none – Replicate only standard NTFS files.
mac_resources – Replicate standard NTFS files and their corresponding Macintosh resource fork.

-preserve_root_permission|-nopreserve_root_permission

Use this qualifier to replicate root directory permissions.

The following qualifiers are for use with distribution jobs performed between UNIX machines only:

-preserve_permission**=\"protection,suid,sgid\"**

protection - Specify to replicate UNIX permissions of the replicated files (read, write, execute).
suid – Replicate the Set UID bit.
sgid - Replicate the Set GID bit.

-directory_permissions=**dir_permission**

Set the value of UNIX directory permissions (0-777 octal). Permissions will be set to all created sub-directories in the distribution target tree.

Directory permissions are set only to directories (and sub-directories) that are actually created during the replication process.

-directory_permissions_on_creation|-nodirectory_permissions_on_creation

A switch to determine that the directory permissions will be created for each directory upon its creation, and not at the end of the replication process.

-copy_dir_permissions_on_creation|-nocopy_dir_permissions_on_creation

A switch to determine that the directory permissions will be copied for each directory upon its creation, and not at the end of the replication process, to avoid permission problems of not being able to write the files to the directory.
The default is -copy_dir_permissions_on_creation.

-root_permissions=**root_permission**

Manually set the permissions of the distribution target root directory, 0-777 (Octal method). Root permissions are set whether the directory was created as part of the replication or already existed.

-files_permissions=**file_permission**

Set the value of UNIX files permissions (0-777 octal). Permissions will be set to all the files transferred during the distribution process.

`-preserve_ownership=`**`gid|uid`**

`uid` - Saves User ownership of the files selected for distribution.
`gid` - Saves Group ownership of the files selected for distribution.

`-extended_attributes=`**`LINUX_Descriptor`**

Replicate LINUX extended attributes.

Edge Default and Edge Override Modifiers Qualifiers

NOTE: The following qualifiers can be specified either with the **`edge_default`** or with the **`edge_override`** commands:

`-scrambled_password=`**`hashed_password_on_edge`**

Replace `the -password` qualifier if a scrambled password is used.
`hashed_password_on_edge` is generated using the **`scramble`** command.

Performance\Throughput Qualifiers

`-tree_recurse|-notree_recurse`

A switch, specifying whether distribution should recurse through the entire source tree structure, or should only distribute the root directory specified without performing recursion.

`-depth_limit=`**`n`**

The number of subdirectories to recurse through. The default is 0 (zero), which represents infinite depth.

`-use_snapshot_compression`

A switch specifying whether to compress the snapshot before transfer from the Edge(s) to the Center. Use when dealing with large replication tress that result in large snapshot files.

`-snapshot_bandwidth=`**`bandwidth_expression`**

Specify a bandwidth control expression to be used in the transfer of the snapshot file. Refer to **Appendix B** for guidelines on bandwidth expressions.

`-user_list=`**`file_name`**

By default, the files to be distributed will be obtained directly from the directory specified as the source directory, being filtered by the use of file and exclude specifications. By using `-user_list` the files that will be distributed will be taken from a text file supplied by the user. **`file_name`** should represent a full path. The file can either be created directly by the user or generated by an outside process. This qualifier must be used in conjunction with the `-list_source` qualifier.

For a full explanation regarding the format and behavior of the User Defined List (UDL) option, see **Appendix E**. This qualifier must be used in conjunction with the `-list_source` qualifier.

`-list_source=`**`center_runtime`**

Although the path to the user list is defined with the `-list_source` qualifier, the source system and time at which the list is taken is not. **`center_runtime`** will take the list from the Center at the time the job is actually run.

`-abort_on_missing_file|-noabort_on_missing_file`

If a user list is being used and R-1 cannot find a file specified in the list, it will move on to the next file. Using `-abort_on_missing_file` will have R-1 abort the job with error status.

NOTE: The methods in which R-1 deals with any errors in the User Defined List (UDL) text file can be found in **Appendix E**. Before using this option, please consult the appendix carefully. Failure to do so may result in incorrect functionality.

`-duplicate_files_handling=`**`error_exit|newest|oldest|`**

`lexicographic_first|lexicographic_last`

When R-1 encounters a situation where a source or target system is case sensitive (i.e. UNIX), the User must decide how it will handle discrepancies:

`error_exit` - Exit the job with an error (default).
`newest` - Transfer the file with the newest modification date.
`oldest` - Transfer the file with the oldest modification date.
`lexicographic_first` - Transfer the file that is first alphabetically.
`lexicographic_last` - Transfer the file first is last alphabetically.

NOTE: Lexicographic order interprets capital letters before lower-case letters.

Example:

If the source contains files: File1, file1, and FILE1 and the target is a Windows machine, the target can only have a single “file1”, regardless of case.

`-abort_on_delete_error|-noabort_on_delete_error`

If there is an error deleting any files on the target machine, abort the job in error. This can happen due to insufficient permissions, or if a file that R-1 has marked for deletion has been manually deleted by the user or by an external process. .

`-abort_on_makelist_error|-noabort_on_makelist_error`

If there is an error creating the snapshot on the source or target machine the job will abort in error. This error can appear if the source does not exist, the user name the job is running under does not have the appropriate permissions to access the directories specified in the job definition, or there is not enough free temporary disk space within the R-1 installation directory.

In the case of UNIX file systems, an error will also be created if there are non-file objects such as sockets within the directory being scanned.

NOTE: During the Snapshot Generation phase, files are accessed for retrieving details only. If the user has access to retrieve details of these files, it will pass this phase successfully. However, if that user does not have access to read or delete these files, the job will fail at the Transfer stage. To overcome access problems at the Transfer stage, use the `-abort_on_tranfer_error` qualifier.

`-abort_on_transfer_error|-noabort_on_transfer_error`

If there is an error in the actual transferring of files, or the target has run out of free disk space, abort the job in error. For example, if a file has been altered (deleted / renamed) between the time the snapshot was generated and the actual file transfer.

`-controller_snapshot_resources=`**`n`**

`n` can take a value from 0 to 10. The Snapshot generation stage can be quite CPU intensive. To reduce the amount of CPU usage on the Center during the Snapshot phase, increase the value of this qualifier.

`-target_directory=`**`target_directory_name`**

`target_directory_name` indicates the target directory for the distribution, it should be in the native format of the target operating system. For an explanation on file formats see **Cross Platform Formats and Escaping** description in **Chapter 5**.

`-delete_before_transfer|-nodelete_before_transfer`

Use `-delete_before_transfer` when transferring large files and there are also many files to delete from the target directory. This will prevent the replication from failing due to lack of disk space on the target machine.

The default is -nodelete_before_transfer, meaning files removed from the source directory will be deleted from the target directory AFTER transferring changed files.

`_purge_files_only|-nopurge_files_only`

Use `-purge_files_only` when you want the replication process to remove files deleted from the source directory at the target directory, but to preserve the directory structure on the target.

The default is -no`purge_files_only`, meaning that files and directories removed from the source will also be removed from the target.

This qualifier applies only when using **mirror** and **purge** logic.

Edge Recovery Qualifiers

These qualifiers determine the number of times a job should attempt to recover before exiting with a failure, as well as how long the job should wait between retries.

`-max_retries=`**n**

The maximum number of recovery tries before failure.

`-base_interval=`**n** **in seconds**

The interval between retries.

`-retry_factor=`**n** **in percent**

An incremental increasing factor used to geometrically increase the interval length between attempts. For example, with `-base_interval`=100 and `-retry_factor`=10.

`-max_interval=`**n** **in seconds**

The maximum period of time to which the `-base_interval` can grow to using `-retry_factor`.

Reporting Qualifiers

For more information about the available reports refer to **Appendix M – Reports**.

`-report_style=`**total|logs|summary|detailed**

Define the level of detail in the Transfer report. This report logs information regarding the file transfer itself.

total - Displays the least amount of job detail. Information includes job start/end time and quantity of files transferred.
log - This style includes all information from **total** as well as the source/target (including path) filename and the file size.
summary - All information from **log** and **total** styles is included, as well as the start and end times of any files transferred.
detailed - All of the above information as well as block level transfer details per file transferred.

This report file may be very large. If the job includes the transfer of many files, or the Center has a limited amount of free space available, it is recommended that **Log** or **Total** styles be used.

Replication Level Time Tolerance Qualifiers

Specify the content synchronization logic.

`-logic=`**`mirror|backup|purge`**

mirror - At the end of the synchronization process, the Source and Target directories are exactly the same.

- Files that exist on source but do not exist on target are copied to target.
- Files that exist on both source and target – with any differing properties (size / modification date) - are copied from the source to the target.
- Files that do not exist on source but exist on target are deleted from target if they are older than the setting set in the `-minimal_purge_age` qualifier.
- Files existing on both source and target having differing NTFS permissions only are not transferred, however the NTFS permissions may be replicated by putting a check mark in the preserve/detect NTFS checkbox.
- Files that exist on both source and target that are identical will **NOT** be copied.

backup - Files from source are copied to target but no deletions are made on target machine.

- Files that exist on source but do not exist on target are copied to target.
- Files that exist on both source and target are copied from the source to the target only if they have a more recent creation / modification date.
- Files that do not exist on source but exist on target are left unchanged.
- **Important for Windows Target:** Mismatch - a file (directory) on the source has the same name and path as a directory (file) on the target - results in deletion of the file (directory) on the target and the creation of the directory (file) from the source to the target.

purge - Files/directories that exist on target but do not exist on source are deleted from the target. This logic involves no transfer of files from the source to the target.

If this option is selected, an additional option can be enabled to only delete files on the target that are older than a preset parameter.

NOTE: Using a **Continuous Updates** job, if a directory is renamed, it is purged (deleted) from the target directory.

`-minimal_purge_age=`**`n`**   **`in seconds`**

If specified, R-1 will not purge files that were created or modified within the past **n** seconds.

These qualifiers determine the lifespan of the R-1 Snapshots and/or the pre-distribution commands. A job recovers based on the information gathered during the initial data deployment attempt. The job relies on the initial Comparative Snapshot generated and perhaps even the pre-distribution command. If a significant amount of time has elapsed between the recovery attempt and the initial run, the Comparative Snapshot and pre-distribution command may no longer be 'relevant'. The following qualifiers define how long a Snapshot and pre-distribution commands are valid.

`-tree_valid_for=`**`tree_valid_time`**   **`in HHMM`**

The value entered in this qualifier indicates the valid life of the snapshot of the directory tree. If the threshold has been surpassed, a new directory tree Snapshot is obtained on each recovery attempt, and the job will use the new Snapshot.

`-cmds_valid_for=`**`cmd_valid_time`**   **`in HHMM`**

The value entered in this qualifier indicates the valid life of the pre-replication command. If the threshold has been surpassed, the command will be re-executed, and the job processed.

Transfer Integrity Qualifiers

`-compare_after_transfer|-nocompare_after_transfer`

R-1 will analyze each distributed target file to ensure that it matches the source. This prevents problems in data integrity caused by external processes, such as another user or application altering either source or target data during distribution. When comparing files on the Center and Edge nodes, a CRC (Cyclic Redundancy Check) calculation on each file is performed and the results are compared. Only the results of the calculation are transmitted for purposes of comparison, so the comparison itself is immediate. Enabling this option can significantly decrease performance.

`-ssl|-nossl`

Specify to R-1 that all communication with the Edges will be over SSL.

For explanation about R-1 security, see the **Overview Chapter**.

`-overwrite_after_completion|-nooverwrite_after_completion`

If a user attempts to access a file that is in the process of being transferred, or if the distribution fails in the middle of a file transmission, only part of the file will exist on the target therefore users may access corrupted data. Using this qualifier, the file will be transferred to a temporary location on the target, leaving the original file on the target intact. Once the entire file has been transferred to the target, the original file is overwritten, and the temporary file removed. Use this qualifier to ensure that users will only access a complete file.

NOTE: Overwrite After Completion of Transfer feature is activated automatically by R-1 using Multicast or the LAN transfer engine. To override the Rename procedure or the default temporary directory, use the `-rename_hook` and `-satellite_transfer_temp_dir` qualifiers.

`-satellite_transfer_temp_dir=`**`temp_directory_name`**

By default, R-1 will use its installation directory as the temporary directory. Use **`temp_directory_name`** to define the temporary directory on the Edge. This should be specified using the native file format of the respective system. For an explanation on file formats see **Cross Platform Formats and Escaping** description in **Chapter 5**.

`-rename_hook=`**`hook_filename`**

Specify the name of the rename hook script. For the Rename Hook option to work, `-overwrite_after_completion` must be specified too.

`-rename_hook_param=`**`hook_parameters`**

Specify the parameters to be passed to the rename hook script.

`-copy_shares|-nocopy_shares`

Used to replicate shares when both source and target are NTFS file systems. When this qualifier is specified, shares defined on directories in the source tree will be replicated to the target.

`-ignore_hidden|-noignore_hidden`

Using the `-noignore_hidden` switch will replicate UNIX hidden files (i.e. files beginning with a period) during file transfer. The default is `-ignore_hidden`, meaning that hidden files will not be transferred.

`-ignore_windows_invalid_file_names|-noignore_windows_invalid_file_names`

If the source is UNIX and the target is Windows, UNIX files whose names include invalid characters in Windows will not be transferred. If this option is not selected, and files of that kind exist, the job will fail.

`-file_time_tolerance=`**`n`** **`in seconds`**

The distribution process may be fine tuned by adjusting the age threshold used by R-1 to decide that a target file is too old and requires replacement with a current version. This parameter sets a threshold used by R-1 to compare the modification date/time of similar files on the source and target.

One application for the file time tolerance parameter would be to allow for the difference in time stamping when replicating between FAT and NTFS drives. In this case, a value of two (2) seconds would avoid the possibility of unnecessary file transfers.

`-reduce_satellite_snapshot_resources=`**`n`**

`n` can take a value from 0 to 10. The Snapshot generation stage can be quite CPU intensive. To reduce the amount of CPU usage on the Center during the Snapshot phase, increase the value of this qualifier.

`-compression=`**`none|normal|ap|power`**

`none` - Specify **`none`** as your compression method on LANs or other high-speed networks.
`normal/ap` - Use **`normal`** or **`ap`** if your CPU is slow, or if high-speed communications lines are employed. **`ap`** usually yields the best compression ratio on data files.
`power` - Compression yields a higher compression ratio than other compression methods. **`power`** provides the highest degree of compression on executable files.

`-dont_compress=`**`file_name`**

`file_name` specifies the file types that should not be compressed by R-1. These can include individual files, wildcards, or entire subdirectories.

Transfer Qualifiers

`-bandwidth=`**`n`** **`in MB|Mb|KB|Kb|%`**

`n` represents an absolute numeric value for the maximum bandwidth usable by the transfer. It should be followed by **`MB|Mb|KB|Kb|%`** to identify whether the bandwidth control will be absolute, or relative.

A complete explanation on Relative and Absolute Bandwidth may be found in the **Overview chapter**. An expression can also be entered using time frames, by following the guidelines in **Appendix B**.

`-transfer_streams=`**`n`**

Use `-transfer_streams` when distributing over VPN or over a high-speed LAN. Transfer performance can be boosted when performing simultaneous parallel transfers of partial file lists.

n specifies the number of parallel transfer operations required and can take a value of 0-10. That is, if the distribution comparative snapshot indicates a need to transfer x files, that list will be broken internally into **n** smaller lists that will be transferred in parallel.

`-differential_transfer|-nodifferential_transfer`

Tells R-1 to use the differential transfer option. R-1 will only transfer the changes in a given file during transfer, down to the block level.

`-differential_min_size=`**`n`** **`in bytes`**

Applying differential transfer to small files can actually increase the total replication time. **n** represents the minimum size of a file R-1 should apply differential transfer to. There is no advantage at all in using the Differential Transfer option with files smaller than 64K.

`-unicast_engine=`**`rw_lan|rw_wan`**

Select the Unicast transport engine.
`rw_lan` – Use a transfer engine optimized for LAN environment.
`rw_wan` – Use a transfer engine optimized for WAN environment

NOTE: Every engine has its own set of parameters n qualifiers. R-1 will ignore parameters of the engines not currently in use during the transport process.

Execution Qualifiers for a Specific Edge Distribution

Data distribution is often part of a wider operation. In that event, pre or post processing are needed and become an integral part of the distribution operation. The built in recovery mechanism considers an operation incomplete if the post-synchronization processing was not completed successfully, even if the distribution itself did. Commands can be run prior (pre) or after (post) a distribution on the Center and Edge machines respectively.

The commands should be an absolute path to the batch file or script, and any parameters or switches should be passed to the command using the appropriate qualifier.

`-pre_controller_command=`**`command_to_run`**

R-1 will execute the **`command_to_run`** on the Center prior to the Job submission.

`-pre_controller_parameters=`**`command_parameters`**

`command_parameters` is the text to be passed as parameters to the `-pre_controller_command`.

`-pre_satellite_command=`**`command_to_run`**

R-1 will execute the **`command_to_run`** on the Edge prior to the Job submission.

`-pre_satellite_parameters=`**`command_parameters`**

`command_parameters` is the text to be passed as parameters to the `-pre_satellite_command`.

`-post_controller_command=`**`command_to_run`**

R-1 will execute the **`command_to_run`** on the Center after a successful execution of the Job.

`-post_controller_parameters=`**`command_parameters`**

`command_parameters` is the text to be passed as parameters to the `-post_controller_command`.

`-post_satellite_command=`**`command_to_run`**

R-1 will execute the **`command_to_run`** on the Edge after a successful execution of the Job.

`-post_satellite_parameters=`**`command_parameters`**

`command_parameters` is the text to be passed as parameters to the `-post_satellite_command`.

`-pre_command_timeout=`**`n`** **`in seconds`**

`n` specifies the time to wait until the pre command on the Center and/or the Edge(s) will execute. If the timeout have passed, the execution of the pre-command fails. 0 means no timeout is specifies.

`-post_command_timeout=`**`n`** **`in seconds`**

`n` specifies the time to wait until the post command on the Center and/or the Edge(s) will execute. If the timeout have passed, the execution of the pre-command fails. 0 means no timeout is specifies.

Exit Procedure Qualifiers

All Distribution Exit Procedure Qualifiers listed in the CLI chapter may be used here. There are different reports that may be used while completing an Edge job, these are listed below. For more information about the available reports refer to **Appendix M – Reports**.

```
-success_email_include_reports=\”general,transfer,pre_edge,post_edge,
pre_center,post_center,deleted_files,excluded_files\”
```

The job reports to include in the e-mail upon successful completion of a job.

```
-abort_email_include_reports=\”general,transfer,pre_edge,post_edge,
pre_center,post_center,deleted_files,excluded_files\”
```

The job reports to include in the e-mail, when the job is aborted.

```
-error_email_include_reports=\”general,transfer,pre_edge,post_edge,
pre_center,post_center,deleted_files,excluded_files\”
```

The job reports to include in the e-mail, when the job has failed.

Wait Settings Qualifiers

These qualifiers deal with when R-1 returns control to the user. Use these qualifiers while using the CLI from scripts.

```
-wait
```

R-1 will not return control until job has completed.

```
-wait_interval=n               in seconds
```

R-1 will check if the job has completed every **n** seconds. Default is 5 seconds.

```
-wait_timeout=n                in minutes
```

If the job has not completed after **n** seconds, return control to the user anyway. This in no way affects the job. **n** is in minutes, with default 60.

TCP Communication Qualifiers

These qualifiers enable controlling TCP/IP communication timeouts, by using Application Timeouts. Setting these qualifiers will override the operating system TCP timeouts. Value 0 means using the operating system timeouts.

Timeouts can be set for each engine separately as well as for overall R-1 communication.

`-lan_connect_timeout=`**`n`** **in seconds**

Specify, in seconds, the TCP connect session timeout to be used during transfer using the LAN engine. Default is 0, using Operating system defaults.

`-lan_send_timeout=`**`n`** **in seconds**

Specify, in seconds, the TCP send operation timeout to be used during transfer using the LAN engine. Default is 0, using Operating system defaults.

`-lan_receive_timeout=`**`n`** **in seconds**

Specify, in seconds, the TCP receive operation timeout to be used during transfer using the LAN engine. Default is 0, using Operating system defaults.

`-wan_connect_timeout=`**`n`** **in seconds**

Specify, in seconds, the TCP connect session timeout to be used during transfer using the WAN engine. Default is 60 seconds.

`-wan_send_timeout=`**`n`** **in seconds**

Specify, in seconds, the TCP send operation timeout to be used during transfer using the WAN engine. Default is 60 seconds.

`-wan_receive_timeout=`**`n`** **in seconds**

Specify, in seconds, the TCP receive operation timeout to be used during transfer using the WAN engine. Default is 60 seconds.

`-connect_timeout=`**`n`** **in seconds**

Specify, in seconds, the TCP connect session timeout for general communication within R-1 components. Default is 60 seconds.

`-send_timeout=`**`n`** **in seconds**

Specify, in seconds, the TCP send operation timeout, for general communication within R-1 components. Default is 60 seconds.

`-receive_timeout=`**`n`** **in seconds**

Specify, in seconds, the TCP receive operation timeout, for general communication within R-1 components. Default is 60 seconds.

Scheduling Qualifiers

These qualifiers enable to schedule jobs in various ways. Distribution Scheduling qualifiers are specified in the **CLI Chapter – Scheduling Qualifiers**.

Additional options regarding trigger file are available for Upload and Download jobs only. For Distribution jobs, the Trigger file can be located on the Center only.

For Upload and Download jobs, the Trigger file location can be: on Center, on Edge, on both Center and Edge, on either Center or Edge.

`-run_option=`**`immediate|once|every|daily|weekly|triggerfile|sat_triggerfile|cont_or_sat_triggerfile|cont_and_sat_triggerfile`**

Defines when the job is scheduled to run.
`immediate` – Run immediately
`once -` For a job to be run once at a later time
`every -` For a job to run at a certain interval
`daily -` For a job to run once daily
`weekly -` For a job to be run on multiple days and times on a weekly cycle
`triggerfile` – For a job that will be triggered by the creation of a file on the Center.
`sat_triggerfile` – For a job that will be triggered by the creation of a file on the Edge. Used with the qualifier `-sat_trigger_file`.
`cont_or_sat_triggerfile` – For a job that will be triggered by the creation of a file on the Center or on the Edge. Used with the qualifiers `–sat_trigger_file` and `–trigger_file`.
`cont_and_sat_triggerfile` – For a job that will be triggered by the creation of a file on the Center and on the Edge. Used with the qualifiers `–sat_trigger_file` and `–trigger_file`.

`-trigger_file=`**`trigger_file_name`**

The full path to the trigger file residing on the Center. Used with –run_option=**triggerfile**, **cont_or_sat_triggerfile** or **cont_and_sat_triggerfile.**

`-sat_trigger_file=`**`sat_trigger_file_name`**

The full path to the trigger file residing on the Edge. Used with –run_option=**sat_triggerfile**, **cont_or_sat_triggerfile** or **cont_and_sat_triggerfile.**

`-sat_check_interval=`**`n          in seconds`**

Specify the time interval to check whether the trigger file is on the Edge. Only used when –run_option=**sat_triggerfile**, **cont_or_sat_triggerfile** or **cont_and_sat_triggerfile.**

Example:

The job will run when files exist on both Center and Edge.

```
>r1 submit … -run_option=cont_and_sat_triggerfile
-sat_trigger_file=c:\target\satellite_trigger.txt
-trigger_file=d:\source\controller_trigger.txt
```

Rollback Definition Qualifiers

Rollback functionality is employed through the ‘recording’ of all replication related changes on a target system, followed by a potential ‘roll back’ of the content at a later time. The recording phase occurs prior to any propagation of changes to a target system and includes the recording of creation, deletion and modification to files and directories (including ownership and permission changes).

The following qualifiers are used to define the properties of the recording phase of a job. To actually activate a rollback job, see the **Rollback Job Qualifiers** description below.

NOTE: Submitting a Distribution job, the Rollback Definition qualifiers may be used in the Modifier file only and not in the submit command itself.

`-rb_enable|norb_enable`

Use this qualifier to enable rollback recording on the target.
Recorded information includes:

- Files and directories created on the target
- Files and directories removed from the target
- Files and directories modified on the target (included modified permission, ownership and Windows ACLs)

NOTE: Recording only applies to data impacted by a replication or distribution as defined above – it does not record a complete copy of the target directory.

`-rb_location=`**`recording_folder`**

Specify where the recorded data will be stored. The recording location may be specified as a path on the target machine (`D:\r1_recordings`) or a network share (`\\nasstorage\R1_recording`)

Since the recording operation takes place on the target (i.e. an Edge during an upload job or a distribution, or the Center during a download job), the recording directory must be accessible for read and write with the credentials used to access the target file system. This refers to the Edge user defined in an upload or distribution job, or the Center user (i.e. the username used to connect to the Console, or used in the command line) in a download job.

This qualifier must be specified when using the `-rb_enable` qualifier.

`-rb_retain=`**`n`** **`in minutes`**

Specify in minutes how long recorded information should be kept for and directly relates to how far back a job can be rolled back. For example, a time of 18 hours will allow the job to undo all replication related changes that took place on the target system for the last 18 hours. Default is 1 week.

`-rb_compress|-norb_compress`

Using `-rb_compress` will store all recorded data in a compressed form. While preserving disk-space, this feature will increase recording and rollback times. The default is `-norb_compress`.

Replication Example

The following command is used to submit a Replication job that records Rollback information on a network share drive.

```
>r1 submit -center=34.0.25.71 -center_user=rwroot
-center_pass=***
-replication_name=\"Web Site Update\" -edge=152.163.12.2
-user=user_name -password=password
-source_dir=/usr/backup -target_dir=d:\data\content
-rb_enable
-rb_location=\\nasstorage\R1_recording
R1API-S-SUS, job <4155> successfully submitted
```

Distribution Example

The following command is used to submit a Distribution job that records Rollback information on a network share drive; each Edge's data is recorded on a different location.

NOTE: Submitting a Distribution job, the Rollback Definition qualifiers may be used in the Modifier file only and not in the submit command itself.

```
>r1 submit -center=34.0.25.71 -center_user=rwroot
-center_pass=***
-distribution_name=\"Sites Update\"
-user=user_name -password=password
-source_dir=d:\web_site\content -target_dir=e:\data\content
-mod=c:\webmod.txt
website1,website2,website3
>edge_default set -rb_enable -rb_retain=600
-rb_location=\\nasstorage\recording

>edge_override website1 mod -rb_location=\\nasstorage\recording\web1

>edge_override website2 mod -rb_location=\\nasstorage\recording\web2

>edge_override website3 mod -rb_location=\\nasstorage\recording\web3
R1API-S-SUS, job <4157> successfully submitted
```

The file `c:\webmod.txt` will look like:

```
# Enable Rollback to all Edges
edge_default set -rb_enable -rb_retain=600 -rb_location=\\nasstorage\recording
# Specify a different recording folder to each Edge
edge_override website1 mod -rb_location=\\nasstorage\recording\web1
edge_override website2 mod -rb_location=\\nasstorage\recording\web2
edge_override website3 mod -rb_location=\\nasstorage\recording\web3
```

Appendix G
External Center Snapshot

In cases where the CST™ and CSM™ processes may take a long time, the user can create the local (Center) snapshot lists. The user can specify to R-1 not to perform the Center's snapshot but rather use a previously built one, giving a directory name.

The directory must include ALL snapshots needed for that R-1 job. It is the user responsibility to make ALL needed lists (notice that each R-1 job may require different lists).

The snapshot lists file names are:

```
center_ntfs4_list.rpl
center_ntfs5_list.rpl
center_unix_list.rpl
center_cp_list.rpl
```

R-1 will check that the directory exists, and that ALL needed lists (can be one or more) in that directory are there.

If it fails to find the directory or any of the lists - the job aborts. Note that a partial file list is not enough. All lists should exist in the directory given by the user.

If all needed lists exist, R-1 will process the job without making new local snapshots.

NOTE: Use this option cautiously. The job will abort if not all local snapshot files exist.

To set this option, use the **Distribution Properties / Snapshot** tab in the R-1 Console GUI, or use the R-1 CLI -`use_central_lists` qualifier.

Appendix H
Continuous Updates

R-1's Continuous Updates jobs provide event-driven replications and distribution processes.

While submitting a Continuous Updates job, a distribution 'father' job is created whose purpose is to spawn and monitor 'children' distributions that perform the data distribution itself. The children are standard R-1 distribution or replication jobs with the significant addition of the RepliWeb Event Driven Agent (REDA). The REDA is an event driven process - not a device driver - that detects, collects, and filters the events on the source data that are relevant to each of the distribution targets – subsequently directing how R-1 responds to the events.

NOTE: Every spawned job has a different Job ID. The continuous update job itself "runs" forever, until it's aborted by user command.

Distribution Process / Stages

Continuous Updates jobs differ from other scheduled R-1 replication or distribution jobs in that they run forever, while at specified intervals, they spawn a job, performing the complete cycle every job performs.

The source machine sends instruction to the target(s) requesting a snapshot of a given directory, while in turn building its own. The respective machines do all of the state processing locally, without use of the network or any conversational protocols. Once the targets have returned snapshots to the source, the source machine then builds a matrix of which objects need to go to which machines, and which need to be deleted – and subsequently performs the necessary operations.

NOTE: In Continuous Update file deletion is made always before transfer, so any files deleted from the source or target machines will be logged in the Pre-Transfer-Deletion report.

NOET: Using a Continuous Updates job, if a directory is renamed, it is purged (deleted) from the target directory.

From the moment the CSM generation is instigated, the REDA begins detecting and collecting any relevant changes made to the source directory (relevant with respect to the source file and 'exclude' specifications). The REDA then continuously instructs the R-1 distribution to propagate the relevant file / directory creation and deletion events to the target systems. The continuous update distribution will autonomously force a full synchronization on a pre-determined periodic basis as well as under special circumstances (for example, if a flood of events results in a 'dropped' event).

The forced synchronization ensures that at fixed intervals, any damaged, missing, or corrupt files on the target are corrected according to the latest source data, in addition to the execution of all distribution level tasks.
At full synchronization, the current instance is performing all On-exit commands, e-mail reporting, and post commands (if specified), then a new instance is spawned, going through the phases of the job: pre-distribution processing (pre commands, pre distribution download) and planning. The running instance then waits for events to act upon, until the next synchronization.

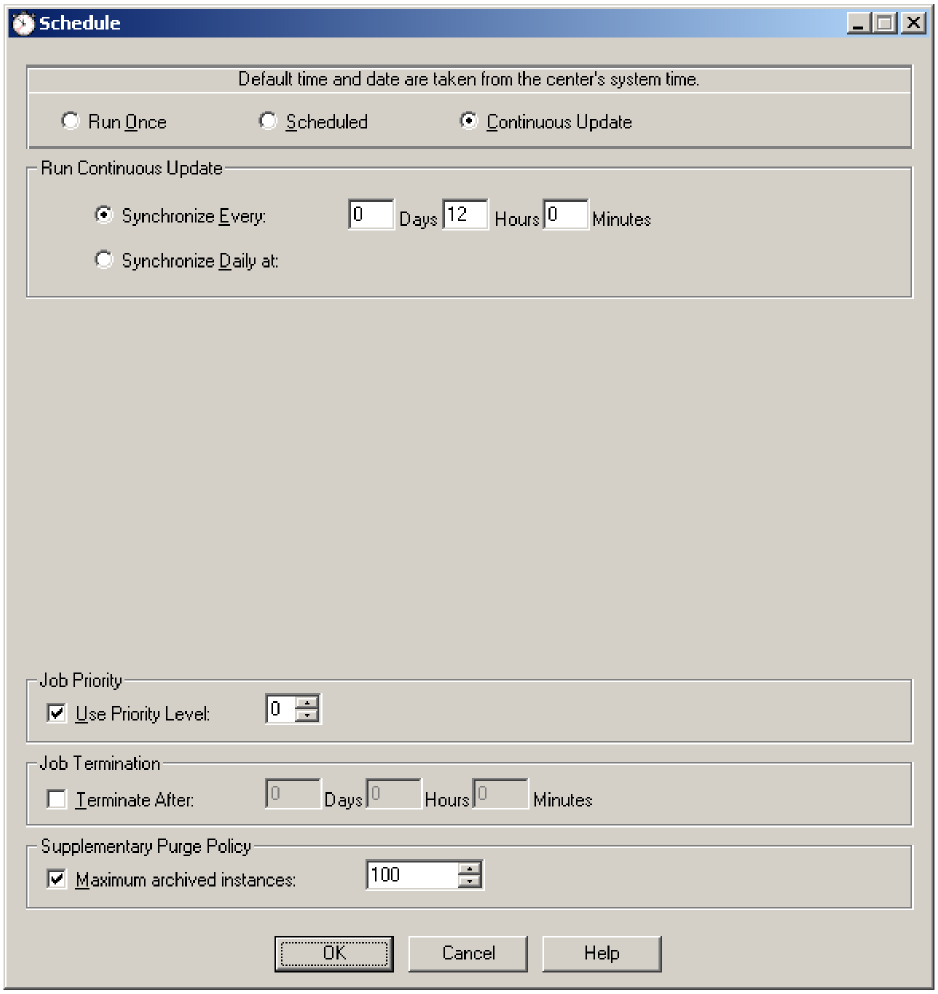
- **Immediate Synchronization** – While in the Job List window, the user may force full synchronization to a Continuous Update job, using the menu.

Job Configuration

To make sure source and target directories are always synchronized, synchronization should take place at specified intervals. The synchronization phase actually performs all exit procedures of a job, and repeats the snapshot phase, capturing the differences between the source and target directories. That way, if any events were missed, the synchronization phase will verify the both systems hold the same information.

NOTE: The **Manually confirm Comparative Snapshot** option is unavailable for **Continuous Updates** jobs.

From the **Job Definition** window (Upload or Distribution), press the **Schedule** button. Then select the **Continuous Updates** option.



Schedule

Default time and date are taken from the center's system time.

○ Run Once ○ Scheduled ● Continuous Update

Run Continuous Update

● Synchronize Every: 0 Days 12 Hours 0 Minutes

○ Synchronize Daily at:

Job Priority

☑ Use Priority Level: 0

Job Termination

□ Terminate After: 0 Days 0 Hours 0 Minutes

Supplementary Purge Policy

☑ Maximum archived instances: 100

Figure 58: Continuous Updates Definition

- **Continuous Updates** – Set an automatic job that replicates changes as soon as they occur.
    - **Synchronize Every** – Specify the interval in which the job rescans both source and target machines to determine what needs to be replicated, to make sure both source and target are synchronized (days/hours/minutes).
    - **Synchronize Daily** – Specify the job to synchronize the source and target directories daily at a specified time.
    - **Max Synchronize Delay** – Specify in seconds, how long to wait for a distribution job to exit before full synchronization is about to take place. In case of a lengthy post command or other operation, if the job is about to

complete, the Continuous Update job will wait the time specified before aborting the job and performing the full synchronization.

Appendix I
Exit Messages

R-1 has the ability to report events to the Microsoft Windows Event Viewer and the UNIX syslog. The current implementation issues reports upon the completion of an R-1 distribution or replication job. Job completion means the job has completed successfully, has been aborted by the user, or has exhausted all retry attempts and failed.

Success Exit Messages

R1-S-Distribution completed successfully

Information Exit Messages

R1-I-Waiting for wakeup conditions
R1-I-Holding for Comparative Snapshot confirmation
R1-I-Source is empty - holding for user confirmation

Abort Exit Messages

R1-W-Aborted by user command

Failure Exit Messages

R1-E-Failed at Center pre distribution download
R1-E-Failed at Center pre commands
R1-E-Failed at Center post commands
R1-E-Failed in Edges pre commands
R1-E-Failed in Edges post commands
R1-E-Failed to submit Edges logic
R1-E-Failed in Edges gathering remote information
R1-E-Failed to generate local snapshot
R1-E-Failed to control Edges operation
R1-E-Failed to gather multicast quorum
R1-E-Failed to maximize multicast group
R1-E-Failed to create multicast CSM
R1-E-Failed to multicast files
R1-E-Failed to check unicast progress
R1-E-Failed to wait for Edges completion
R1-E-Failed to set last break point
R1-E-Failed to set wakeup condition
R1-E-RepliWeb not installed
R1-E-Failed to connect using dial-up

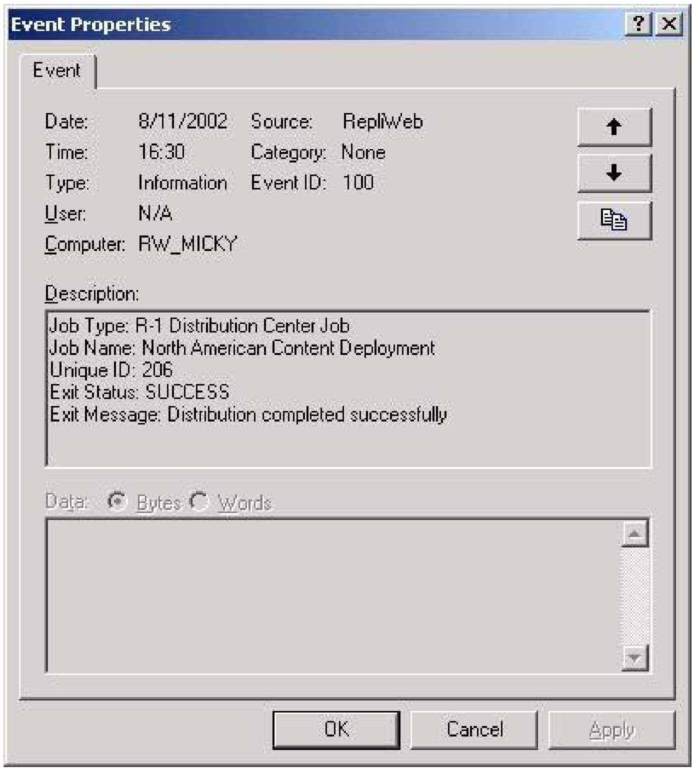
R1-E-Failed to disconnect dial-up
R1-E-Failed to get remote node information
R1-E-Failed in paths check
R1-E-Failed to convert remote node information
R1-E-Failed to check break point
R1-E-Failed at license validation
R1-E-Failed to execute pre replication commands
R1-E-Failed to execute post replication commands
R1-E-Failed to generate snapshots
R1-E-Failed to compare snapshots
R1-E-Failed to transform comparative snapshot
R1-E-Failed at pre transfer deletes
R1-E-Failed at post transfer deletes
R1-E-Failed to transfer files
R1-E-Failed to cleanup Node temporary files
R1-E-Failed to create Node directories
R1-E-Unknown state of replication job #

Internal Failure Messages

R1INT-F-Didn't find configuration file in working directory
R1INT-F-Failed to allocate configuration file handle
R1INT-F-Failed to set lock on configuration file
R1INT-F-Failed to set configuration file table
R1INT-F-Failed to read configuration file
R1INT-F-Failed to get configuration file field
R1INT-F-Failed to set configuration file field
R1INT-F-Failed to set configuration file section
R1INT-F-Failed to update configuration file
R1INT-F-Failed to get job state from configuration file
R1INT-F-Failed to initialize configuration file
R1INT-F-Failed at initialization
R1INT-F-Missing parameters (working directory)
R1INT-F-Missing parameters (job id)
R1INT-F-Couldn't set local job directory and multicast information directory
R1INT-F-Failed to initialize cleanup file
R1INT-F-Failed to initialize report file
R1INT-F-Failed to initialize debug file
R1INT-F-Failed to initialize network module
R1INT-F-Unable to get RepliWeb server name
R1INT-F-Failed to allocate schedule handle
R1INT-F-Failed to release cores after hold
R1INT-F-Failed to create cores hold file
R1INT-F-Holding by server instruction
R1INT-F-Failed to initialize the distribution manager job
R1INT-F-Failed at distribution manager validity check

R1INT-F-Failed to clean R-1 working directory
R1INT-F-Distribution stopped at debug check point
R1INT-F-Unable to set default FASTCopy qualifiers
R1INT-F-Unable to set default FASTCopy transfer qualifiers
R1INT-F-Unable to allocate sa handle
R1INT-F-Failed to write to hold file

Microsoft Event Viewer



Date: 8/11/2002 Source: RepliWeb
Time: 16:30 Category: None
Type: Information Event ID: 100
User: N/A
Computer: RW_MICKY

Description:
Job Type: R-1 Distribution Center Job
Job Name: North American Content Deployment
Unique ID: 206
Exit Status: SUCCESS
Exit Message: Distribution completed successfully

Figure 59: Windows Event Viewer

Events from R-1 will show under **Source = RepliWeb**
The event description consists of the following:

- **Job Type**
  - R-1 Distribution Center Job
  - R-1 Replication Job
  - R-1 Distribution Edge Job
- **Job Name -** Defined upon job submission
- **Unique ID** - R-1 Job ID
- **Exit Status**
  - SUCCESS [Leads to “Information” event]
  - ABORT [Leads to “Warning” event]

- ERROR [Leads to "Error" event]
- WARNING [Leads to "Error" event]

Event ID List

Event ID	Job Type	Exit Status
100	R-1 Distribution Center Job	SUCCESS
110	R-1 Distribution Center Job	ABORT
120	R-1 Distribution Center Job	ERROR
125	R-1 Distribution Center Job	WARNING*
130	R-1 Replication Job	SUCCESS
140	R-1 Replication Job	ABORT
150	R-1 Replication Job	ERROR
155	R-1 Replication Job	WARNING*
160	R-1 Distribution Edge Job	SUCCESS
170	R-1 Distribution Edge Job	ABORT
180	R-1 Distribution Edge Job	ERROR
185	R-1 Distribution Edge Job	WARNING*

*A job that has completed successfully, but has a failed the "On-exit" command will appear in the event viewer as a "Warning".

UNIX syslog

The UNIX syslog configuration is system specific; configuration information is usually found in `/etc/syslog.conf`. When R-1 is configured to notify the syslog of replication events, the events are sent in XML format for simplified parsing.

The events are passed with syslog_priority LOG_INFO for successful job completion, LOG_WARNING on a user aborted job, and LOG_ERROR on a job completed with error.

A job that has completed successfully, but has a failed the "On-exit" command will appear in the event viewer as a "Warning".

The event is constructed in the following form (without newline / tab formatting)

```
<RepliWeb R-1 Message>
      <Job Type>
            %s
      </Job Type>
      <Job Name>
            %s
      </Job Name>
      <Unique ID>
            %s
      </Unique ID>
      <Status>
            %s
      </Status>
      <Exit Message>
            %s
      </Exit Message>
</RepliWeb R-1 Message>
```

The values for each XML field are identical to those described above in the Windows Event Viewer.

Appendix J
File List

A critical aspect of enterprise level data deployment is the ability to easily determine exactly which files were transferred to which systems, and the details associated with the transfer itself. The RepliWeb R-1 **File List** feature enables a R-1 deployment or replication to automatically create a file containing the above-mentioned information.

A 2-Steps Process

During the transfer stage a log is created, each line representing information pertaining to a single file transferred. The file itself has a default format (described in this appendix), but can be easily created in accordance with a user-defined format by employing a template. Once the file has been created, it can then be stored locally on the Center, e-mailed to multiple recipients, or even executed as a Windows batch file or UNIX script. Some examples of the former, as well as detailed directions in setting up the File List feature are described below.

Job Configuration

The job configuration consists of the following steps:

1. Set the properties **Transfer Report Style** under the **Edge General** Tab to either **Summary** or **Detailed**. Any other report style will result in an **empty** file list being created.

2. Specify the File List exit procedure in the respective **On Exit tab** – this varies depending on the type of job scheduled and should be enabled in all tabs that apply.

    - In an **Upload** / **Download** job –The File List exit procedure should be enabled in the **On Exit** tab

    - In a **Distribution** job using **Unicast** – The file list is created for each edge containing the files unicast. The File List exit procedure should be enabled in the **Edge Exit** tab.

    - In a **Distribution** job using **Multicast** – The file list will be created for all files Multicast by the distribution. The File List exit procedure should be enabled in the **Center Exit** tab.

- In **Distribution** jobs using **Multicast** with **Edge** / **File Level Complement** – Define the File List exit procedure **BOTH** in the **Center Exit** tab (for multicast file list) and the **Edge Exit** tab (for the files transferred using Unicast).

3. Specify the File List properties
   - **None** – Creates the file list on the Center
     a. **Template File** – Mandatory.
     The full path to a custom template must be defined. Templates are explained in detail in the following section.

     b. **Target Directory** – Mandatory.
     Specifies the directory on the Center where the file list will be created. Do not specify a file name, as the file list will be created as **transfer_*JOBID*.log**, where *JOBID* is the unique id of the distribution or replication job.

   - **Mail File List** – Mails the file list to the specified recipients
     a. **Template File** – Mandatory.
     The full path to a custom template must be defined. Templates are explained in detail in the following section.

     b. **Target Directory** – Optional.
     Use this feature to keep a copy of the File List on the Center. Specifies the directory on the Center where the file list will be created. Do not specify a file name, as the file list will be created as **transfer_*JOBID*.log**. *JOBID* is the unique id of the distribution or replication job.

     c. **Recipients** – Mandatory.
     Specify the e-mail address(es) of those receiving the file list. Multiple email addresses should be separated by a comma.
     e.g. user1@company.com,user2@company.com

   - **Execute File List** – Creates a Windows batch file or UNIX script based on the specified template file. Since each line in the file list represents a unique file that was transferred, when the batch file / script is executed on the center, each line is executed in sequence. Some examples of using the **Execute File List** option are provided below.
     a. **Template File** – Mandatory.
     The full path to a custom template must be defined. Templates are explained in detail in the following section.

     b. **Target Directory** – Optional.
     Use this feature to keep a copy of the File List batch file or script on the Center. Specifies the directory on the Center where the file list will be created. Do not specify a file name, as the file list will be created as

transfer_*JOBID*.bat on windows, and **transfer_*JOBID*.sh** on UNIX. *JOBID* is the unique id of the distribution or replication job.

For the Execute option, it is ideal to leave this field empty and use the default Job directory. That way the intermediate file will be purged along with the job itself.

Using a Template File

The template file dictates the format of the File List that will be created or executed (as explained above).

Creating a Template File

The template file format is as follows:

HEADER
multiple lines written to the file once
BODY
one line written to the file for each file transferred
FOOTER
multiple lines written to the file once

1. Section names are case sensitive
2. HEADER section may contain multiple lines.
3. HEADER section is optional. If omitted, there's no need to write the BODY header. The first line in the file will be regarded as the template line.
4. If BODY is specified, FOOTER must exist (but can be empty).
5. BODY may contain only one (1) line. Since each line in the file list represents a unique file transferred, the line in the template file is a representation of what each line in the file list will look like.
6. The template file line can contain any text, plus the following variables. When the file list is created after a deployment / replication, the variables are replaced on a line-by-line basis with the information about the given file.
7. Template line may use pre-defined variables for information about the transferred file:

```
%j job name
%c source hostname
%n destination node/hostname
%z file size (bytes)
%s source file specification
%d target file specification
%t file transfer completion time
%q sql time format file transfer completion time
```

NOTE: `%n` & `%d` are **NOT** supported in **Multicast** file list(s) as there are multiple values for the destination parameters

Environment Variables

R-1 provides environment variables to be used while handling a distribution job. These variables may be used in the Post Commands or the File List batch files.

RW_JOB_UNIQUE_ID=NNN	Job ID
RW_JOB_NAME=	Job Name
RW_JOB_COMPLETION_STATUS	The job’s completion status: SUCCESS|ABORT|ERROR

Examples of Template Files

The following template file is automatically used for unicast transfer if the **Template File** field is left blank:

```
Source:<[%c]%s> Target:<[%n]%d> Size(bytes):<%z> Transfer Ended at:<%t>
```

A file list created with this template would then take on the following format:

```
Source:<[PAUL]C:\source\expenses.xls>
Target:<[RINGO]C:\target\expenses.xls> Size(bytes):<147456> Transfer
Ended at:<Mon Jun 30 14 45 05 2003>
Source:<[PAUL]C:\source\contract.doc>
Target:<[RINGO]C:\target\contract.doc> Size(bytes):<1247458> Transfer
Ended at:<Mon Jun 30 14 47 12 2003>
```

Examples of Execute Template Files

An execute template file takes on the same form, and uses the same variables as a normal template file, but must contain commands that can be executed from a command-prompt / shell.

The following are examples of execute template files that perform a given task:

Delete source files after transfer

```
del “%s”
```

Would create the following batch file on the Center, which would then be executed when the job is complete:

```
del “C:\source\expenses.xls”
del “C:\source\contract.doc”
```

Command

In this example the attributes of each of the transferred files and append them to a file:

```
attrib “%s” >> C:\attributes.txt
```

Would create the following batch file on the Center, which would then be executed when the job is complete:

```
attrib “C:\source\expenses.xls” >> C:\attributes.txt
attrib “C:\source\contract.doc” >> C:\attributes.txt
```

Formatted HTML List

In this example, the file list takes on the following form:

```
<li><a href=%s>%s</a> Transferred at %t
```

Resulting in the following file that could be viewed in a web browser:

```
<li><a
href=“C:\source\expenses.xls”>“C:\source\expenses.xls”</a>
Transferred at Mon Jun 30 14 45 05 2003
<li><a
href=“C:\source\expenses.xls”>“C:\source\expenses.xls”</a>
Transferred at Mon Jun 30 14 47 12 2003
```

Post Command and File List

To even enhance the user’s power at the exit stage, this file can also include header and footer that are in free format, allowing calling other batch file or contain executable commands. This allows both File List and Post Commands as the exit procedure.

The File List Template can then look like this:

```
HEADER
echo Start of Job %RW_JOB_UNIQUE_ID% %RW_JOB_NAME%>> c:\lists\file_list.txt
echo Completion Status: %RW_JOB_COMPLETION_STATUS% >> c:\lists\file_list.txt
d:\repliwebbatchfiles\checkstatus.bat
BODY
echo %c,%n,%z,%s,%d,%t,%q >> c:\lists\file_list.txt
FOOTER
echo End of Job %RW_JOB_UNIQUE_ID%>> c:\lists\file_list.txt
```

As part of the replication process, at the exit stage, a batch file is created at the job directory, based on the template file. Then the batch file is executed, writing the Job ID, Name and Completion Status using Environment variables, handling the job's completion status (`d:\repliwebbatchfiles\checkstatus.bat`) and writing the information about the transferred files to the log file (`d:\lists\file_list.txt`). At the end of the list a line will be added marking the end of the list using the Environment Variable holding the Job ID.

The output file (`c:\lists\file_list.txt`) will then look like this:

```
Start of Job 277 Weekly Update
Completion Status: SUCCESS
srv,target_srv,1263,d:\source\record.info,d:\target\log\record.info,Tue
Feb 17 10 51 35 2004,2004-02-17 10:51:35
srv,target_srv,4320,d:\source\2\1\record.pack,d:\target\log\record.pack
,Tue Feb 17 10 51 35 2004,2004-02-17 10:51:35
srv,target_srv,126344,d:\source\2\2\record.info,d:\target\log\2\record.
info,Tue Feb 17 10 51 35 2004,2004-02-17 10:51:35
srv,target_srv,4320,D:\dmgr_source\2\2\record.pack,d:\target\log\record
.pack,Tue Feb 17 10 51 35 2004,2004-02-17 10:53:40
End of Job 277
```

Other Examples of Template File Use

Insert file transfer statistics into a SQL Database

By using the following template file, in addition to the Mail File List feature, file transfer statistics can be inserted into a SQL Database.

The e-mail account is POP'ed by the SQL server, and the lines in the file list represent 'insert' commands into the SQL Database.

```
INSERT INTO File_list
(Source_Hostname,Destination_Hostname,Filesize,Source_Filespec,
Target_Filespec,Tranfer_Completion_Time,SQL_Time_Format_Transfer
_Completion_Time)
VALUES ('%c', '%n', '%z','%s', '%d', '%t', '%q')
```

Appendix K
Excluding Files and Directories

A given distribution process may require the exclusion of certain files or directories. The most common scenario includes temporary and log files that are system specific and should not be distributed to the target nodes. R-1 enables the exclusion of files and directories from the distribution process.

The exclude specification can include:

- Files & Directories– wildcards may be used
- UNIX hidden files - Exclude files beginning with a period
- Ignore Windows invalid file names – If the source is UNIX and the target is Windows, UNIX files whose names include invalid characters in Windows will not be transferred. (If this option is not selected, and files of that kind exist, the job will fail.)

Please note, that the Job properties may specify which files (using file names, folder names or file types) are to be transferred. In that case, the **File Specification** is calculated together with the **Exclude** files options.

For example, if File Specification = `*.exe` and Exclude Files = `abc.exe`, then all `*.exe` files **BUT** `abc.exe` will be transferred.

NOTE: Files can be excluded from distribution to all Edge(s). They cannot be excluded from distribution to some of the Edge(s) within a distribution job.

To ensure that the exclude specification defined will perform as expected, it is advised to employ the **Manually confirm comparative snapshot** feature for the first distribution. For a complete description of this feature, consult the User's Guide or online help through the R-1 Console.

The exclude specifications are best illustrated by means of example. The following examples build in complexity from a relatively simple exclude specification, to a more complicated one.

NOTE: Using * as the exclude specification will result in all files in the source directory being excluded from the replication.

1. Exclude Examples

The following samples illustrate various exclusion scenarios.

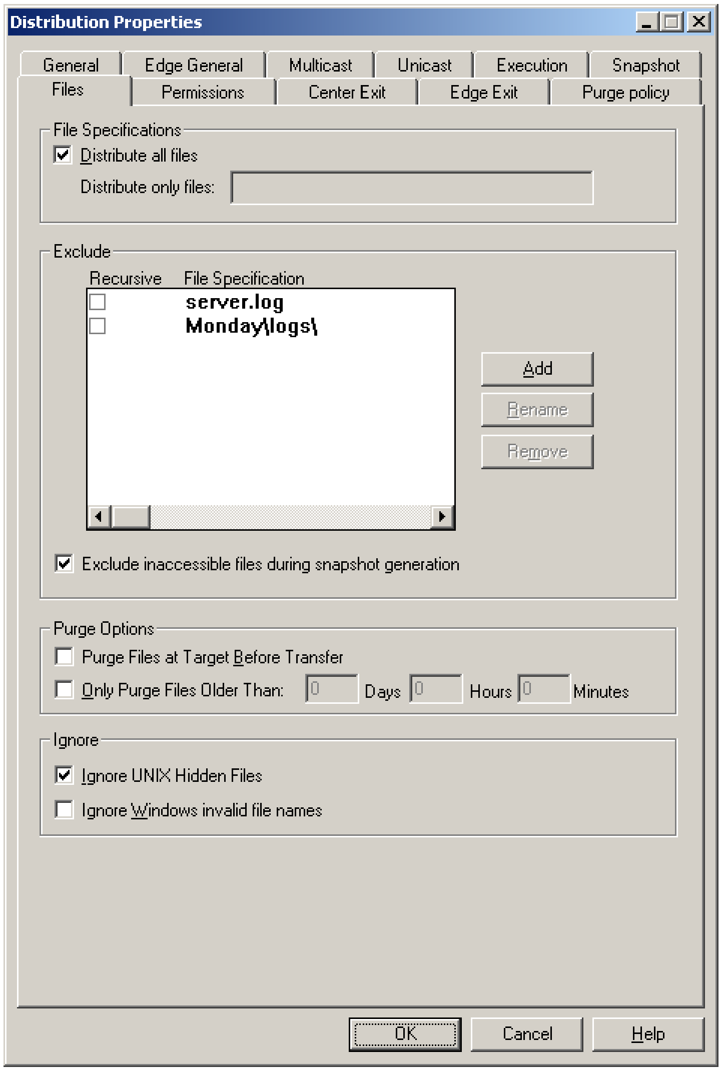
Please note the following:

- Folder names are always relative to the source path.
- To exclude first level folders from the source just enter in the exclude field: `foldername\`
- To exclude lower folders put the entire path relative to the source in the field: `foldername\subfolder\`
- Excluding a directory means, "exclude the directory and all files and folders within that directory".
- By checking the **Recursive** option, you exclude all such folder names from within the source tree, and NOT all files within an excluded directory.

1.1. Simple Files and Directories Exclusion

Source Directory: `C:\Master_data`
Would like to exclude:

- `C:\Master_data\server.log`
- `C:\Master_data\Monday\logs` directory and everything inside it



File Specifications
☑ Distribute all files
Distribute only files:

Exclude

Recursive	File Specification
□	server.log
□	Monday\logs\

☑ Exclude inaccessible files during snapshot generation

Purge Options
□ Purge Files at Target Before Transfer
□ Only Purge Files Older Than: 0 Days 0 Hours 0 Minutes

Ignore
☑ Ignore UNIX Hidden Files
□ Ignore Windows invalid file names

Figure 60: Excluding Files & Directories

Notice:

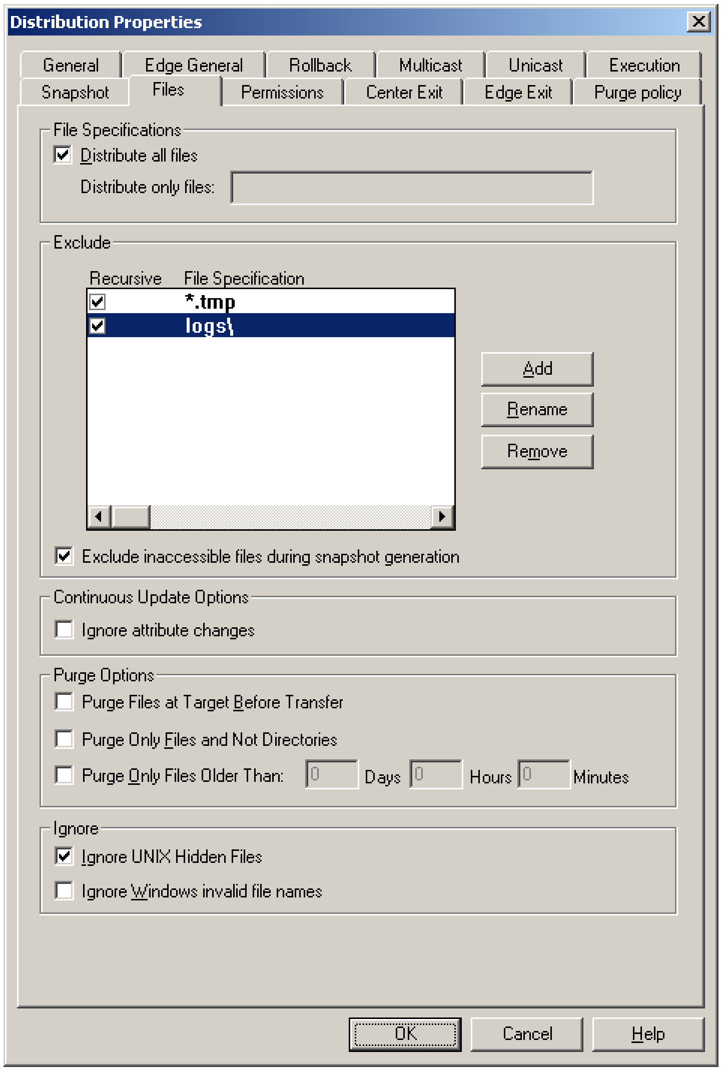
- Exclude specifications are relative to the **job defined source directory**
- A slash must be placed after a directory. The slash is system specific and relative to the **source system Operating System** (i.e. `Monday\logs\` on a Windows source and `Monday/logs/` on a UNIX source)
- The **Recursive** checkbox will be explored in the subsequent examples. It **does not** refer to whether the files and directories within `C:\Master_data\Monday\logs`

will also be excluded – they always are regardless of whether the **Recursive** box is checked or not.

1.2. Recursive Files and Directories Exclusion

Source Directory: `C:\Master_data`
Would like to exclude:

- All files with a `tmp` extension throughout the source directory tree (i.e. `*.tmp`)
- All `logs` directories within `C:\Master_data` and everything inside them



File Specifications

☑ Distribute all files

Distribute only files:

Exclude

Recursive	File Specification
☑	*.tmp
☑	logs\

☑ Exclude inaccessible files during snapshot generation

Continuous Update Options

☐ Ignore attribute changes

Purge Options

☐ Purge Files at Target Before Transfer

☐ Purge Only Files and Not Directories

☐ Purge Only Files Older Than: 0 Days 0 Hours 0 Minutes

Ignore

☑ Ignore UNIX Hidden Files

☐ Ignore Windows invalid file names

Figure 61: Recursive Exclude

Notice:

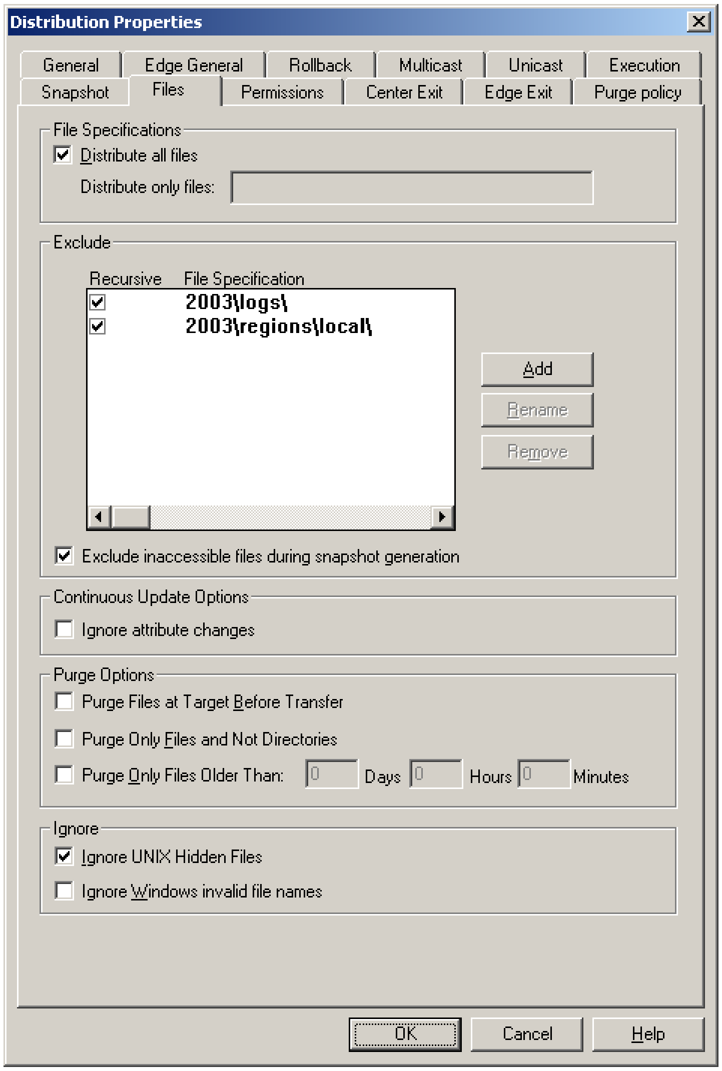
- All *.tmp files in `C:\Master_data` and any of its sub-directories will be excluded
- All `logs` directories in `C:\Master_data` and any of its sub-directories will be excluded

- By checking the **Recursive** option, you exclude all such folder names from within the source tree.

1.3. Recursive Multi-Level Directories Exclusion

Source Directory: `C:\Master_data`
Would like to exclude:

- All `logs` directories within `C:\Master_data\2003` and any of its subdirectories, and everything inside those `logs` directories
- All `local` directories within `C:\Master_data\2003\Regional` and any of its subdirectories, and everything inside those `local` directories



File Specifications

☑ Distribute all files

Distribute only files:

Exclude

Recursive	File Specification
☑	2003\logs\
☑	2003\regions\local\

☑ Exclude inaccessible files during snapshot generation

Continuous Update Options

□ Ignore attribute changes

Purge Options

□ Purge Files at Target Before Transfer

□ Purge Only Files and Not Directories

□ Purge Only Files Older Than: 0 Days 0 Hours 0 Minutes

Ignore

☑ Ignore UNIX Hidden Files

□ Ignore Windows invalid file names

Figure 62: Recursive – Multi Level Exclude

Notice:

- The first exclude specification will exclude all `logs` directories that fall inside `C:\Master_Data\2003`. For example, `C:\Master_Data\2003\logs` **will** be excluded, as well as `C:\Master_Data\2003\Regional\Toronto\logs`, but `C:\Master_Data\logs` **will not** be excluded.
- The second exclude specification will exclude all `local` directories that fall inside `C:\Master_Data\2003\Regional`.

  For example, `C:\Master_Data\2003\Regional\local` **will** be excluded, as well as `C:\Master_Data\2003\Regional\Toronto\intranet\local`, but `C:\Master_Data\local` **will not** be excluded.

1.4. Excluding Files in a Distribution Job

The sample job requires that:

1. `.gif` files should not be transferred
2. Unique systems files should not be transferred `(win.ini)`
3. User specific folders should not be transferred.

The job properties will specify:

1. Recursively excluded `*.gif`
2. Exclude `win.ini`
3. Exclude sub folder `james` and all its files and sub directories.

The following table shows the source and target directory trees, before and after the replication process.

Source Tree	Target Tree Before Replication	Target Tree After Replication
❑ Users o a.exe (07-jan-03) o win.ini (01-mar-02) o a.jpg o b.jpg o a.gif o b.gif o Common_Files • b.exe • x.jpg • x.gif o dll_folder • a.dll • b.dll • c.dll o james • my_docs o a.doc o b.doc • c.gif • d.gif	❑ Users o a.exe (01-jan-03) o win.ini (01-jan-03) o Common_Files • b.exe o dll_folder • a.dll • b.dll • c.dll • d.dll o james • a.txt • b.txt • c.txt	❑ Users o a.exe (07-jan-03) o win.ini (01-jan-03) o a.jpg o b.jpg o Common_Files • b.exe • x.jpg o dll_folder • a.dll • b.dll • c.dll • d.dll* o james • a.txt • b.txt • c.txt

The **Target Tree** changes as a result of the replication job. Some files are replaced, some files and directories stay intact.

- ❑ Updated
  - `a.exe` is updated as the file on the target machine is older than the one in the source machine.
- ❑ Unchanged
  - `win.ini` is unchanged as it was excluded from the replication.
  - `james` directory and all its tree is unchanged as the whole source directory `james` was excluded from the replication.
- ❑ New
  - *.jpg files were copied to the target machine. No `.gif` files were transferred as they were excluded recursively from the replication.
- ❑ *Deleted
  - This file is deleted only if the replication logic is **Mirror**. Using Mirror logic, at the end of the synchronization process, files that do not exist on source but exist on target are deleted from target. Using **Backup** logic the file will remain intact.

NOTE: The different target directories on the different Edges may look different after the distribution job is complete, depending on their tree before the replication. But, the excluded files are excluded from the complete job, and these files will not be transferred to any of the target Edge(s).

2. Excluding Files and Directories Using the CLI

`-exclude=`**`file_type`**

File types that match the criteria in **`file_type`** should be completely ignored during the replication process.

`-recursive_exclude=`**`recursive_file_type`**

The following file types should be recursively ignored during the replication. This means any files or directories matching this criteria will not be deleted or transferred.

Example

This example will exclude the `.logs` file type from the relative root of replication as well as the `file active.log`. The replication will also ignore all directories beginning with `__vti__`.

NOTE: Entire directories to be excluded must end with a backslash.

Since this command is being run from the CLI, it must be properly escaped.

```
>r1 submit …
-exclude=\”*.logs,active.log\”
–recursive_exclude=\”__vti__*\\\\\”
```

NOTE: There are no spaces (blanks) after the commas.

3. Automatic Excludes…

R-1 may exclude files automatically during the Distribution process.

1. Files that could not be accessed during the snapshot generation phase. These files will be listed in the **Exclude** report.

2. During the Snapshot Generation phase, files are accessed for retrieving details only. If the user has access to retrieve details of these files, it will pass this phase successfully. However, if that user does not have access to read or delete these files, the job will fail at the Transfer stage.

   To overcome access problems at the Transfer stage, uncheck the **Abort on error transferring files** at the **Edge General Tab**.

Appendix L
Services Control

The R-1 Services (Server & Scheduler) should be activated before a server can be used as R-1 Center.

On a **Windows** installation, the R-1 processes are running as services, which are started automatically on system startup. Using the Start Menu / Control Panel / Services, make sure that the RDS Server and RDS Scheduler are running.

On a **UNIX** system, the system administrator should add the activation of the background processes to the system startup. R-1 installation does not do that automatically as each system is configured differently according to corporate conventions.

The command that should be added to the UNIX startup is:

```
      /usr/bin/repliweb_scheduler
```

If this command is issued from "root" account, the processes will keep running after you quit.

There are 4 processes that start in the background and perform the function of the scheduler.

As a general policy, it is recommended that the startup of the R-1 scheduler will be added to a later stage of the system initiation and rather than an early stage.

To start the scheduler automatically at system start up, the **repliweb_scheduler** command needs to be added to one of the startup files like `init.d, rc.d` or other.

There is no standard policy, not something RepliWeb can add to the install; each UNIX system administrator is free to do it in a different way. It is an administrator decision.

Below please find a sample on how to add the scheduler to start automatically at system start up:

In `/etc/rc3.d` create file named `S100repliweb` that contains the following script:

```
#!/sbin/sh
#
#
case "$1" in

'start')
      echo "Starting RDS RepliWeb Scheduler"
      rm <RDS installation directory>/config/active.pid
      rm <RDS installation directory>/config/api_server.pid
      rm <RDS installation directory>/config/archive.pid
      rm <RDS installation directory>/config/submit.pid
      /usr/bin/repliweb_scheduler
      ;;
'stop')
      echo "Stopping RDS RepliWeb Scheduler"
      rm <RDS installation directory>/config/active.pid
      rm <RDS installation directory>/config/api_server.pid
      rm <RDS installation directory>/config/archive.pid
      rm <RDS installation directory>/config/submit.pid
      /usr/bin/repliweb_scheduler stop
      ;;
*)
      echo "Start or Stop RDS RepliWeb Scheduler"
      echo "Usage: $0 {start | stop }"
      exit 1
      ;;
esac
exit 0
```

In order to check that the processes are up and running, run:

```
      #ps -x | grep sched_dll
```

You should see the following processes:

/usr/rds/controller/scheduler/bin/sched_dll **RepliWeb Scheduler API Server**
/usr/rds/controller/scheduler/bin/sched_dll **RepliWeb Submit Server**
/usr/rds/controller/scheduler/bin/sched_dll **RepliWeb Archive Server**
/usr/rds/controller/scheduler/bin/sched_dll **RepliWeb Active Server**

Appendix M
Reports

Each R-1 job generates report files that can be used for monitoring and debugging purposes. Since these reports can consume vast amounts of disk space, they are automatically purged after a predetermined amount of time. This means that they will disappear from the Console GUI, and not be visible by a Console CLI `show` command.

NOTE: When a job has reached its purge age, it will be deleted. A purged job is no longer viewable through either the Console CLI or the Console GUI.

The job's reports can be accessed by selecting the appropriate report in **Job List** and **Job Details** windows. Reports are also accessible using CLI show command. Reports can be sent via email at the end of the distribution process.

Report Types

R-1 generates various reports during the different stages of the distribution process. Note that if a stage was not performed (i.e. Post Command was not executed) the corresponding report will not be generated.

- **General** – Contains job information and properties (like Source and Target directories), and specifies the stages the distribution process is going through (e.g. license validation, transfer, post command, etc.)
- **Transfer** – Provides information about the transfer progress. See below the various styles this report can have. This report is available only if transfer actually took place during the distribution job.
- **Pre Distribution Command** – Logs the batch file executed on the Center before the distribution process started. This report is available only if Pre Distribution command was actually executed.
- **Post Distribution Command -** Logs the batch file executed on the Center after the distribution process completed. This report is available only if Post Distribution command was actually executed.
- **Pre Center Command -** Logs the batch file executed on the Center before the distribution to a specific Edge started. This report is available only if Center Pre command was actually executed.
- **Post Center Command -** Logs the batch file executed on the Center after the distribution to a specific Edge completed. This report is available only if Center Post command was actually executed.
- **Pre Edge Command -** Logs the batch file executed on the Edge before the distribution to this Edge started. This report is available only if Edge Pre command was actually executed.

- **Post Edge Command -** Logs the batch file executed on the Edge after the distribution to this Edge completed. This report is available only if Edge Post command was actually executed.
- **Deleted Files** – Lists the files that were deleted from the target directory during the distribution process (before and after actual transfer took place)
- **Excluded Files –** Lists the files that were excluded from the distribution process. File could be excluded because the user selected to exclude them (in the Files – Exclude tab, or because the user has selected to ignore inaccessible files – these files will also be listed in this report.
- **Rollback Recorder –** Lists the files recorded in a job that keeps rollback data. This report is available only if the Rollback option was turned on and files were about to delete or change during the transfer stage.
- **Rollback Reloader –** Lists the files recovered during a Rollback job. This report is available only in Rollback jobs.

Transfer Report Styles

Report Style defines the level of detail in the **Transfer** report. This report logs information regarding the file transfer itself.

NOTE: To use the **File List** option, set the properties **Transfer Report Style** under the **Edge General** Tab to either **Summary** or **Detailed**. Any other report style will result in an **EMPTY** file list being created.

- **Totals -** Displays the least amount of job detail. Information includes job start/end time and quantity of files transferred.
- **Log -** This style includes all information from **Totals** as well as the source/target (including path) filename and the file size.
- **Summary -** All information from **Log** and **Totals** styles is included, as well as the start and end times of any files transferred.
- **Detailed** - All of the above information as well as block level transfer details per file transferred.

NOTE: This report file may be very large. If the job includes the transfer of many files, or the Center has a limited amount of free space available, it is recommended that **Log** or **Totals** styles be used.

Index

Notes