

File List

1. Overview

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. For example, this capability is important for data integrity validation and to governance issues such as Sarbanes-Oxley. The RepliWeb RDS **File List** feature enables a RDS deployment or replication to automatically create a file containing the above-mentioned information.

File List Concept - A Multi-Steps Process

During the transfer stage a detailed **Transfer Report** is created, containing information about the files that were transferred in the replication process.

Using a text **Template** file, a **File List** file is created on the Controller, combining the information from the **Transfer Report**, with the format defined in a **Template** file. The Template file can be easily created in accordance with a user-defined format.

In the **File List** file, each line represents information pertaining to a single file transferred. Once the File List file has been created, it can then be stored locally on the Controller, e-mailed to multiple recipients, or even executed as a Windows batch file or UNIX script on the RDS Controller.

From an RDS perspective, the order of events is as follows:

1. The user creates a Template file, using environment variables provided by RDS.
2. The job is run with **On Exit – File List** option set. The job creates the log of transferred files (Detailed or Summary Transfer Report).
3. Upon job completion the job uses the Template file to create the File List file where each line represents a file that was transferred.
4. The File List file is emailed or executed on the Controller.

NOTE: The whole process, starting at creating the Template file, until the File List file is stored or executed, is performed exclusively on the Controller.

Basic Example – Customized Report

In this example a customized report is created, listing the attributes of each of the transferred files. The customized report is saved to a file called `c:\attributes.txt`:

1. The user creates a **Template** file, using the environment variable `%s`, which represents the name of the file that was transferred.

The Template file contains one line such as this:

```
attrib "%s" >> C:\attributes.txt
```

NOTE: The Template File is NOT a batch file. It is a text file being used to create a batch file that will then be executed.

2. The job is run with **Detailed or Summary Transfer Report** and **On Exit – File List – Execute** option set.

The **Detailed Transfer Report** looks like:

```
16:35:07 LAN transfer to node <production_srv>
16:35:07 Sending data started at - Thu Aug 05 16:35:07 2004

Source File name : C:\source\expenses.xls
Target File name : D:\target\expenses.xls
File size : 11330 bytes
Transfer ended at : Thu Aug 05 16:35:07 2004
Transfer succeeded

Source File name : C:\source\contract.doc
Target File name : D:\target\C:\source\
File size : 15297 bytes
Transfer ended at : Thu Aug 05 16:35:07 2004
Transfer succeeded

16:35:08 Successfully Copied <10> files
16:35:08 Transfer completed at - Thu Aug 05 16:35:08 2004
```

- 2.1. Upon job completion, the job uses the **Template** file to create the **File List** file, where each line contains information regarding a file that was transferred.

The **File List** looks like:

```
attrib "C:\source\expenses.xls" >> C:\attributes.txt
attrib "C:\source\contract.doc" >> C:\attributes.txt
```

- 2.2. The **File List** file is then executed by RDS on the Controller; each line is executed in sequence. Resulting in the following `C:\attributes.txt` file:

```
___A_ C:\source\expenses.xls  
___A_ C:\source\contract.doc
```

2. Job Configuration

The job configuration consists of the following steps:

1. Set the properties **Transfer Report Style** under the **General** Tab to either **Summary** or **Detailed**. Any other report style will result in an **empty** file list being created.

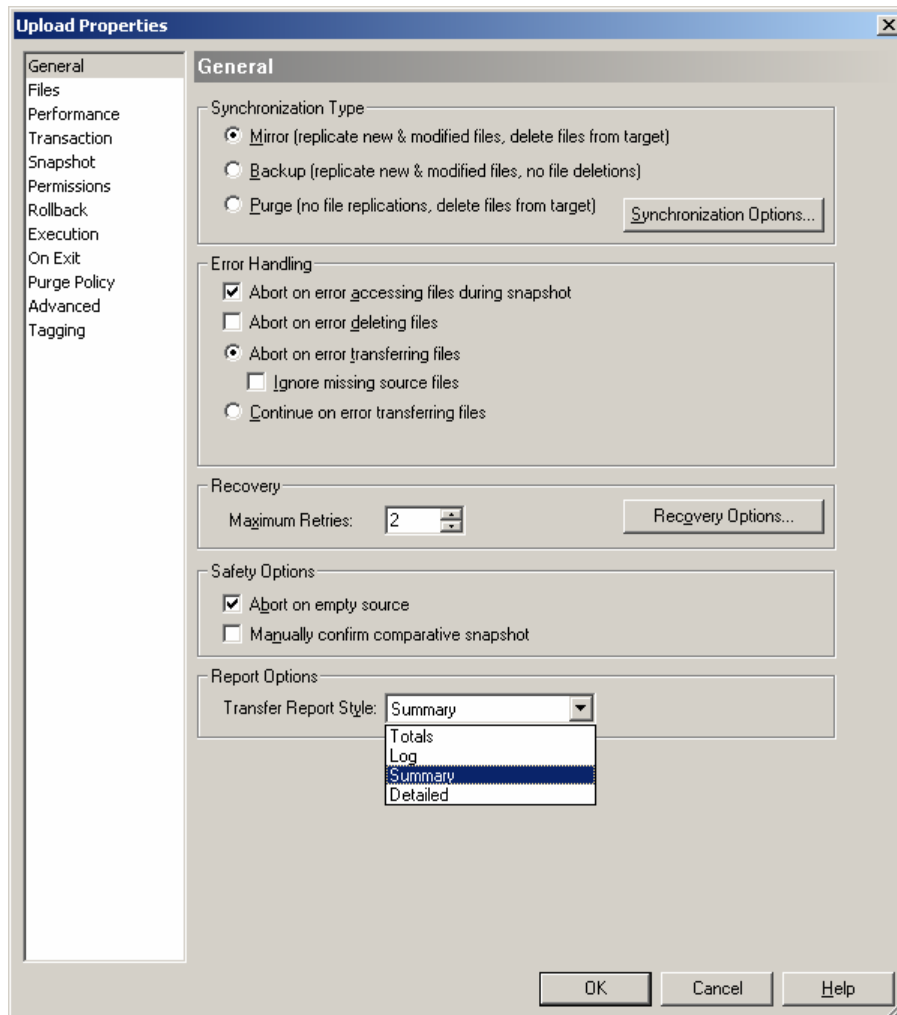


Figure 1 – General Tab – Transfer Report Format

- Specify the File List exit procedure in the **On Exit** tab.

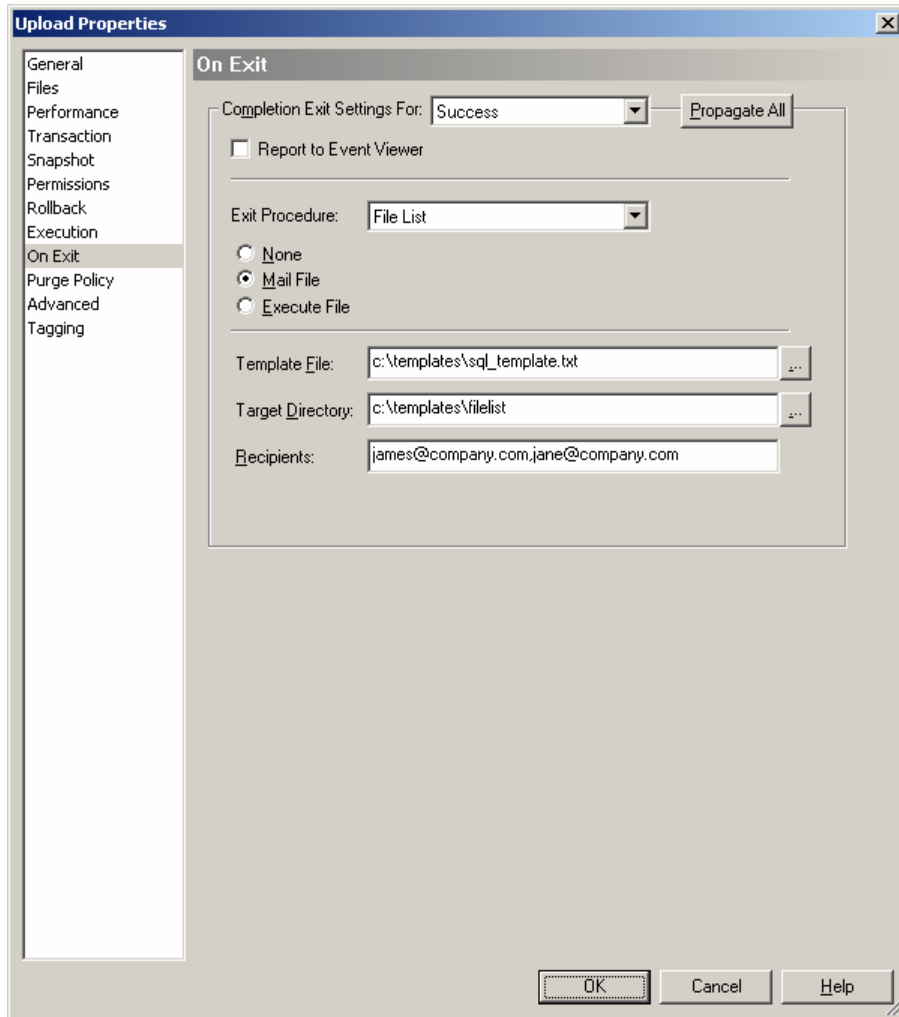


Figure 2 – On Exit – File List

2. Specify the File List properties

- **None** – Creates the file list on the Controller
 - Template File** – Mandatory.
The full path to a custom template must be defined. Templates are explained in detail in the following section.
 - Target Directory** – Mandatory.
Specifies the directory on the Controller 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 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 Controller. Specifies the directory on the Controller 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 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 Controller, 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 Controller. Specifies the directory on the Controller 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 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.

3. Template File

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. For example, this capability is important for data integrity validation and to governance issues such as Sarbanes-Oxley. The RepliWeb RDS **File List** feature enables an RDS replications to automatically create a file containing the above-mentioned information.

File List Concept - A Multi-Steps Process

During the transfer stage a detailed **Transfer Report** is created, containing information about the files that were transferred in the replication process.

Using a text **Template** file, a **File List** file is created on the Controller, combining the information from the **Transfer Report**, with the format defined in a **Template** file. The Template file can be easily created in accordance with a user-defined format.

In the **File List** file, each line represents information pertaining to a single file transferred. Once the File List file has been created, it can then be stored locally on the Controller, e-mailed to multiple recipients, or even executed as a Windows batch file or UNIX script on the RDS Controller.

From an RDS perspective, the order of events is as follows:

1. The user creates a Template file, using environment variables provided by RDS.
2. The job is run with **On Exit – File List** option set. The job creates the log of transferred files (Detailed or Summary Transfer Report).
3. Upon job completion the job uses the Template file to create the File List file where each line represents a file that was transferred.
4. The File List file is emailed or executed on the Controller.

NOTE: The whole process, starting at creating the Template file, until the File List file is stored or executed, is performed exclusively on the Controller.

NOTE: Set the properties **Transfer Report Style** under the **General** Tab to either **Summary** or **Detailed**. Any other report style will result in an **empty** file list being created.

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 variables listed below. 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:

RW_JOB_UNIQUE_ID	Job ID
RW_JOB_NAME	Job Name
RW_JOB_COMPLETION_STATUS	The job's completion status: SUCCESS ABORT ERROR

%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: In UNIX, use $\$x$ while in Windows, use $\%x$

3.1. Examples of Template Files

The following chapter gives multiple examples of Template files that can be used in RDS.

Customized Report

The following template file can be used to create a different report than the Transfer Report.

Windows:



```
Source:<[%c]%s> Target:<[%n]%d> Size(bytes):<%z> Transfer Ended at:<%t>
```

UNIX:



```
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>
```

UNIX File List option including Parameters

The following template file can be used on UNIX using RDS environment variables.



```
HEADER  
BODY  
file %s  
FOOTER  
comp_status $RW_JOB_COMPLETION_STATUS  
job_name $RW_JOB_NAME
```

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 Controller, 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 Controller, 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
```

Executing scripts 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,%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,Tue Feb 17 10 51 35 2004,2004-02-17 10:51:35
srv,target_srv,4320,Tue Feb 17 10 51 35 2004,2004-02-17 10:51:35
srv,target_srv,126344,Tue Feb 17 10 51 35 2004,2004-02-17 10:51:35
srv,target_srv,4320,Tue Feb 17 10 51 35 2004,2004-02-17 10:53:40
End of Job 277
```

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, Transfer_Completion_Time, SQL_Time_Format_Transfer
_Completion_Time)
VALUES ('%c', '%n', '%z', '%s', '%d', '%t', '%q')
```

###

For any additional information, please contact us at support.repliweb.com