

# rdiff 1.0pre DRAFT

Martin Pool

20th July 2004

## 1 Introduction

*rdiff* is a program to compute and apply network deltas. An *rdiff delta* is a delta between binary files, describing how a *basis* (or *old*) file can be automatically edited to produce a *result* (or *new*) file.

Unlike most diff programs, librsync does not require access to both of the files when the diff is computed. Computing a delta requires just a short “signature” of the old file and the complete contents of the new file. The signature contains checksums for blocks of the old file. Using these checksums, rdiff finds matching blocks in the new file, and then computes the delta.

rdiff deltas are usually less compact and also slower to produce than xdeltas or regular text diffs. If it is possible to have both the old and new files present when computing the delta, *xdelta* will generally produce a much smaller file. If the files being compared are plain text, then GNU *diff* is usually a better choice, as the diffs can be viewed by humans and applied as inexact matches.

rdiff comes into its own when it is not convenient to have both files present at the same time. One example of this is that the two files are on separate machines, and you want to transfer only the differences. Another example is when one of the files has been moved to archive or backup media, leaving only its signature.

Symbolically

```
signature(basis-file) -> sig-file  
delta(sig-file, new-file) -> delta-file  
patch(basis-file, delta-file) -> recreated-file
```

rdiff signatures and deltas are binary files in a format specific to rdiff. Signatures consist of a header, followed by a list of checksums for successive fixed-size blocks. Deltas consist of a header followed by an instruction stream, which when executed produces the output file. There are instructions to insert new data specified in the patch, or to copy data from the basis file.

Unlike regular text diffs, rdiff deltas can describe sections of the input file which have been reordered or copied.

Because block checksums are used to find identical sections, rdiff cannot find common sections smaller than one block, and it may not exactly identify common sections near changed sections.

Changes that touch every block of the file, such as changing newlines to CRLF, are likely to cause no blocks to match at all.

rdiff does not deal with file metadata or structure, such as filenames, permissions, or directories. To rdiff, a file is just a stream of bytes. Higher-level tools, such as [rdiff-backup](#) can deal with these issues in a way appropriate to their users.

## 2 Use patterns

A typical application of the rsync algorithm is to transfer a file *A2* from a machine A to a machine B which has a similar file *A1*. This can be done as follows:

1. B generates the rdiff signature of *A1*. Call this *S1*. B sends the signature to A. (The signature is usually much smaller than the file it describes.)
2. A computes the rdiff delta between *S1* and *A2*. Call this delta *D*. A sends the delta to B.
3. B applies the delta to recreate *A2*.

In cases where *A1* and *A2* contain runs of identical bytes, rdiff should give a significant space saving.

## 3 Invoking rdiff

There are three distinct modes of operation: *signature*, *delta* and *patch*. The mode is selected by the first command argument.

### 3.1 signature

```
rdiff [OPTIONS] signature INPUT SIGNATURE
```

**rdiff signature** generates a signature file from an input file. The signature can later be used to generate a delta relative to the old file.

### 3.2 delta

```
rdiff [OPTIONS] delta SIGNATURE NEWFILE DELTA
```

**rdiff delta** reads in a delta describing a basis file. It then calculates and writes a delta delta that transforms the basis into the new file.

### 3.3 patch

rdiff [OPTIONS] patch BASIS DELTA OUTPUT

rdiff applies a delta to a basis file and writes out the result.

rdiff cannot update files in place: the output file must not be the same as the input file.

rdiff does not currently check that the delta is being applied to the correct file. If a delta is applied to the wrong basis file, the results will be garbage.

The basis file must allow random access. This means it must be a regular file rather than a pipe or socket.

### 3.4 Global Options

These options are available for all commands.

- version**            Show program version and copyright.
- help**              Show brief help message.
- statistics**        Show counts of internal operations.
- debug**             Write debugging information to stderr.

Options must be specified before the command name.

## 4 Return Value

**0** Successful completion.

**1** Environmental problems (file not found, invalid options, IO error, etc).

**2** Corrupt signature or delta file.

**3** Internal error or unhandled situation in librsync or rdiff.

## 5 Bugs

Unlike text patches, rdiff deltas can only be usefully applied to the exact basis file that they were generated from. rdiff does not protect against trying to apply a delta to the wrong file, though this will produce garbage output. It may be useful to store a hash of the file to which the digest is meant to be applied.

## 6 Author

rdiff was written by Martin Pool. The original rsync algorithm was discovered by Andrew Tridgell.

Please send questions or bug reports to [librsync-users@lists.sourceforge.net](mailto:librsync-users@lists.sourceforge.net)

This program is part of the [librsync](#) package.