Using Editlists

RIMAGE™

Corporate Headquarters:

Rimage Corporation
7725 Washington Avenue South
Minneapolis, MN 55439
USA

800-553-8312 (toll free US)
Service: +1 952-946-0004 (International)
Fax: +1 952-944-6956

European Headquarters:

Rimage Europe GmbH
Albert-Einstein-Str. 26
63128 Dietzenbach Germany

Tel: +49-(0) 6074-8521-0
Fax: +49-(0) 6074-8521-21

Rimage Corporation reserves the right to make improvements to the equipment and software described in this document at any time without any prior notice. Rimage Corporation reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation of Rimage Corporation to notify any person or organization of such revisions or changes.

This document may contain links to web sites that were current at the time of publication, but may have moved or become inactive since. This document may contain links to sites on the Internet that are owned and operated by third parties. Rimage Corporation is not responsible for the content of any such third-party site.

Contents

Tables

EditList Introduction

The imaging server can create images using either a ParentFolder or an EditList as the source for the files to be imaged. The image order XML file contains a list of one or more ParentFolder or EditList entries and any mix or order. The image order XML is thoroughly described separately from this document. A ParentFolder is simply the path to a folder tree that will be included on the disc, it does not use an EditList. The EditList describes the sources and destinations for the files that will comprise the image. Using an EditList allows for much more flexibility and options than the ParentFolder. The EditList is an XML formatted file with an associated DTD available for validation. It can be created manually by a text editor, by any XML editor, or from code in an application program. The DTD is discussed below and some examples of EditLists are included to help clarify the EditList usage. Some examples are embedded with the element or attribute being discussed and some other are later in this document.

Several of the options in the EditList depend on settings in the image order. For example, the encrypt commands will be ignored unless the encrypt element is in the image order. This affects both PDF and file encryption. The image order DTD should be studied first.

This document describes the EditList version that ships with version 9 of the Rimage software using EditList_1.6.dtd for validation. There is an earlier format of EditList that consisted of simple commands in a text file. This format is still supported and was documented in the past. The XML EditList is the preferred method, further development of the old format ceased several years ago so it does not support any of the newer features of the imager, and is no longer included in this documentation.

It should also be noted that element names, attribute names, and specified strings are case sensitive in XML files. For example, “true” is not the same as “True” and will cause the XML to be rejected.

EditList DTD

The DTD is used to validate the XML contained in an EditList. The DTD is a formal description of the allowed XML format. All of the elements and attributes will be described following the DTD listed below. All Rimage DTD files are found in <RimageSystemFolder>\XML. <RimageSystemFolder> is typically D:\Rimage, but it can be anywhere specified during installation of the Rimage software. The installed DTD file may differ slightly from the version listed below. In particular, this document has the comments removed because complete descriptions of all the elements and attributes are this document, so leaving them in the DTD listing would be redundant. The elements and attributes are hyperlinked to make it easy to find the descriptions.

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT EditList ((BasePathGroup | SrcDst | Options | Filters)*)>
<!ATTLIST EditList
  VolumeGroupPacking (true | false) "false"
>
<!ELEMENT SrcDst EMPTY>
<!ATTLIST SrcDst
  Src CDATA #REQUIRED
  Dst CDATA #IMPLIED
  FileSize CDATA #IMPLIED
  ModifyTime CDATA #IMPLIED
  CreateTime CDATA #IMPLIED
  AccessTime CDATA #IMPLIED
  Hidden (true | false) "false"
  ReadOnly (true | false) "false"
>
<!ELEMENT BasePathGroup (DstGroup+)>
<!ATTLIST BasePathGroup
  BasePath CDATA #REQUIRED
>
<!ELEMENT DstGroup (File*)>
<!ATTLIST DstGroup
  DstPath CDATA #REQUIRED
>
<!ELEMENT File EMPTY>
<!ATTLIST File
  Name CDATA #REQUIRED
  FileSize CDATA #IMPLIED
  ModifyTime CDATA #IMPLIED
  CreateTime CDATA #IMPLIED
  AccessTime CDATA #IMPLIED
  Hidden (true | false) "false"
  ReadOnly (true | false) "false"
>
<!ELEMENT Options EMPTY>
<!ATTLIST Options
  HybridPart (PC | Mac | both) #IMPLIED
  ExpandFolders (true | false) #IMPLIED
  VolumeGroup (start | all | end) #IMPLIED
  SrcCommon CDATA #IMPLIED
  DstCommon CDATA #IMPLIED
  Zip (true | false) #IMPLIED
  EncryptPDF (true | false) #IMPLIED
  TranslateRTFtoPDF (true | false) #IMPLIED
  TranslateTEXTtoPDF (true | false) #IMPLIED
  TranslateHTMLtoPDF (true | false) #IMPLIED
  EncryptFiles (true | false) #IMPLIED
  EncryptFileNames (true | false) #IMPLIED
>
<!ELEMENT Filters EMPTY>
<!ATTLIST Filters
  RegExInclude CDATA #IMPLIED
  RegExExclude CDATA #IMPLIED
  DOSPatternInclude CDATA #IMPLIED
  WildcardInclude CDATA #IMPLIED
  DOSPatternExclude CDATA #IMPLIED
  WildcardExclude CDATA #IMPLIED
  ModTimeBefore CDATA #IMPLIED
  ModTimeAfter CDATA #IMPLIED
>
```

ELEMENT EditList ((BasePathGroup | SrcDst | Options | Filters)*)

A valid EditList consists of a list of 0 or more BasePathGroup, SrcDst, Options, or Filters elements. These may be freely mixed in any order, but it should be noted that the Options and Filters values take effect on any elements following them until changed by another Options or Filters element. For example, this allows some files to be encrypted and others to not be. This Options order dependency is also how VolumeGroups are specified. Later examples will clarify this.

It should also be noted that the order of the elements in the absence of any VolumeGroups does not control the arrangement of files in a multi-volume set. The imager is free to arrange the files in any order it wants. This is typically alphabetical, but this is not guaranteed. Using VolumeGroups is the only way to control the positions of related files in a multi-volume set.

Attribute VolumeGroupPacking (“true” | “false”) “false”

VolumeGroups are described in detail later, but they consist of folders and files that are meant to be kept on a single volume which is part of a multi-volume set. If VolumeGroupPacking is “false” (the default value), then each VolumeGroup will be put on a separate disc. If the value is “true” then the imager will attempt to put as many VolumeGroups as it can on each disc (volume) of the set. Even with packing however note that no VolumeGroup will be split between discs, this is considered an error. VolumeGroupPacking simply allows for multiple groups on a disc if there is room.

Element BasePathGroup (DstGroup+)

This element is included to provide some compatibility with the older EditList command line file. It allows the leading part of a folder path to be specified for a group of other paths containing files and folders to be put on disc. It effectively takes folders and copies them onto discs. An example will help to clarify how this works. It is not as flexible as using the SrcDst elements, but it can sometimes result in smaller EditList files. However, it is not as easy to read and understand as SrcDst and it is recommended that SrcDst be used. The + after the DstGroup indicates that there must be at least one DstGroup specified.

Attribute BasePath CDATA #REQUIRED

The BasePath is a required folder path that will be pre-pended to all the relative DstPath values. It effectively points to the source folder for files and folders that will be placed at the root level of the volume media. If a DstPath is a complete path including a drive then BasePath is ignored. It may be either UNC or drive letter based. Examples of BasePath are:

- C:\user\George\Files
- \\RimageSystem\C_Drive\Rimage\DataFiles

Examples of complete BasePathGroup XML can be found here.

Element DstGroup (File*)

DstGroup contains a list of zero or more (as indicated by *) File entries. It also has the DstPath attribute that is required. If no File entries are listed, then an empty folder will be created using the DstPath value.

Source files are found by combining the BasePath and DstPath. If the ExpandFolders option is false only files at the location BasePath\DstPath will be copied to the media at \DstPath. If ExpandFolders is true both files and folders from BasePath\DstPath will be copied to the media at \DstPath maintaining the hierarchical relationships. There is no limit to the folder depth, unless strict ISO rules are followed which would limit the depth to 8 levels. It is important to note that the path BasePath/DstPath must exist if any File elements are specified.

Attribute DstPath

This must be a folder name contained in BasePath or '\'. It indicates the destination on the optical media. The '\' means to start at the root of the media. It is used as above in combination with the BasePath.

Element File

The File elements contain information about which files to include.

The Name is the only attribute required unless the Rimage Streamer is used. With Streaming the FileSize is also required and the other arguments are recommended, but not required. The Rimage Streamer requires the FileSize in order to transfer the file; it will fail if the size is not supplied. Streaming is documented elsewhere.

Attribute Name CDATA #REQUIRED

Name is either a filename with optional DOS wildcards, or a complete path to a file. A typical entry might be "*.jpg" or "image1.jpg".

Attribute FileSize CDATA #IMPLIED

This is a required argument for Rimage Streaming. It is the file size in bytes.

Attribute ModifyTime CDATA #IMPLIED

This is an optional argument for Rimage Streaming to set the files modified time. The date/time format for all the time fields is CCYY-MM-DD HH:MM:SS.

Attribute CreateTime CDATA #IMPLIED

This is an optional argument for Rimage Streaming to set the files creation time.

Attribute AccessTime CDATA #IMPLIED

This is an optional argument for Rimage Streaming to set the files last access time.

Attribute Hidden (true | false) "false"

This is an optional argument for Rimage Streaming. It sets the file hidden attribute.

Attribute ReadOnly (true | false) "false"

This is an optional argument for Rimage Streaming. It sets the file read only attribute.

Element SrcDst

The SrcDst element describes the relationship of where files come from and where they will be placed on the output media. It is very flexible in that it allows for both files and folders to be listed. As many SrcDst elements as needed may be used. The Options element can by placed anywhere in the list and any options set take effect on following SrcDst elements. This is useful for zipping, PDF, folder expansion, or any other attribute set in the Options element.

SrcDst elements are processed in the order that they are seen in the EditList. However, note that this does not necessarily control the order of files on the media. This is particularly true for any multi-volume/spanning jobs. The files are internally sorted by the Rimage Imager before they are added to the volume or volumes. In general, the imager decides which volume to put files on. The exception to this rule is VolumeGroups, which control which volume a file will be on, but still does not control the order of the files on the volume.

It is highly recommended to use SrcDst instead of DstGroups. It is easier to read and understand.

An example is found here to help clarify the use of SrcDst.

The arguments after Dst, FileSize etc., are only used when the Rimage Streamer is used. FileSize is required in that case, but the rest are optional. FileSize is required because the Rimage Streamer needs the size of the file for transferring it.

Attribute Src CDATA #REQUIRED

The Src attribute is used to specify where a folder or file is found on the hard disk or network drive. Src effectively points to the source for files and folders to be placed on the media. DOS wildcards are allowed if ExpandFolders is false. If a folder is specified it must end with a '\', this is NOT optional!

Attribute Dst CDATA #IMPLIED

Dst contains the media folder or file where the source file or folder will be placed. '\' means the root of the media. "\test\" means in the folder test on the root. Note that a folder MUST end with a '\'. "\test" would imply a file called "test" that is found in the root. Allowing a file for Dst means that files can be renamed.

Src="c:\test\" Dst="\test\" would put all the files from c:\test\ in the \test\ folder on the media. "C:\test\*.*" may be used. *.* is assumed if no specifier after the folder path is found.

Src="c:\test\" would do the same because if no Dst is found, one will be created using the path only part of the Src.

Src="c:\test\*.txt" Dst="\t\" would put only .txt files from c:\test\ in the \t\ folder on the media.

Src="c:\test\abc.txt" Dst="\test\def.txt" would rename the file from abc.txt to def.txt and place it in the indicated \test\ folder on the media.

Attribute FileSize CDATA #IMPLIED

This is the size of the file in bytes. It is only required and used by the Rimage Streamer.

Attribute ModifyTime CDATA #IMPLIED

The file modify time is formatted as CCYY-MM-DD HH:MM:SS for setting the file when the Rimage Streamer is used. If this is not specified then the current date and time will be used.

Attribute CreateTime CDATA #IMPLIED

The file create time is formatted as CCYY-MM-DD HH:MM:SS for setting the file when the Rimage Streamer is used. If this is not specified then the current date and time will be used.

Attribute AccessTime CDATA #IMPLIED

The file access time is formatted as CCYY-MM-DD HH:MM:SS for setting the file when the Rimage Streamer is used. If this is not specified then the current date and time will be used.

Attribute Hidden (true | false) "false"

This is the file hidden attribute set when the Rimage Streamer is used.

Attribute ReadOnly (true | false) "false"

This is the file read-only attribute set when the Rimage Streamer is used.

Element Options

Attributes in Options are used to control special actions of the SrcDst or DstGroups that follow each option element. A common setting is ExpandFolders = “true”.

The Translate…to… options are used to translate files from RTF, TEXT, or HTML into PDF files. The file extensions are used to determine which type of file each one is. EncryptPDF may optionally be combined with any of the Translate... options. The translation is done first, followed by encryption.

Unless specified otherwise default values are empty strings or “false” for logical values.

Attribute HybridPart (PC | Mac | both) #IMPLIED

This setting is used to control where files go when the Mac/PC (HFS/ISO) hybrid format is used. PC forces files to the ISO side and not the HFS part. Mac puts files in the HFS and not in the ISO part. Both puts files in both parts. The default is “both”.

Attribute ExpandFolders (true | false) #IMPLIED

This option is used to control whether or not folders will be expanded in the following SrcDst or DstGroup elements. The default value is “false”.

Attribute VolumeGroup (start | all | end) #IMPLIED

This option is used in combination with spanning in the image order. If spanning is off, then this setting is ignored. When spanning is on VolumeGroups is used to control how the imager assigns files to volumes. Each group will be kept together in a single volume. An error occurs if the group is too big for a volume. Start is used to indicate the beginning of set of files specified using SrcDst or DstGroup elements. End indicates the end of the group. The special keyword “all” starts a group that will be placed on each volume in addition to any other groups. This group is also terminated with the “end” word. Any files outside of a group will just be treated normally and put on as many volumes as needed.

For VolumeGroupPacking equal to false, the “end” choice effectively finishes the current volume and starts work on the next volume. For VolumeGroupPacking equal to true, “end” does not finish the current volume until there is not enough room for the next volume.

VolumeGroups are ignored unless spanning is enabled in the image order.

Attribute SrcCommon CDATA #IMPLIED

Both SrcCommon and DstCommon are intended to reduce the size of EditLists. SrcCommon is a string that is prepended to each Src value before processing. DstCommon is prepended to each Dst value. This can decrease the size of an EditList that has many long paths that are all the same.

Attribute DstCommon CDATA #IMPLIED

See above.

Attribute Zip (true | false) #IMPLIED

When the image order uses the zip option “Selected Files” this attribute decides which files are considered to be selected. The default value is false. See the image order for all the zip options.

Attribute EncryptPDF (true | false) #IMPLIED

This attribute is used when the image order has the PDF encryption values set. If set to “true” then all “*.pdf” files will be encrypted. Like other Options attributes it stays in effect until it is changed by a later Options value. The image order is documented separately from this document. The default value is false.

Attribute TranslateRTFtoPDF (true | false) #IMPLIED

This option translates any files with extension .RTF into PDF files. The base file name is not changed.

Attribute TranslateTEXTtoPDF (true | false) #IMPLIED

This option translates any files with extension .TXT or .TEXT into PDF files. The base file name is not changed. Since the text file does not have any page size information the PDF settings are US Letter, with 0.5” top and bottom margins, and 0.25” side margins. These settings may be adjustable in a future version.

Attribute TranslateHTMLtoPDF (true | false) #IMPLIED

This option translates any files with extension .HTM or .HTML into PDF files. The base file name is not changed.

Attribute EncryptFiles (true | false) #IMPLIED

This option when combined with encryption being enabled in the image order will cause the selected files to be encrypted with the selected AES algorithm. The Rimage SpanRestore program that is installed on spanned volumes can be used to decrypt the files.

Attribute EncryptFileNames (true | false) #IMPLIED

File names can also be encrypted. This makes it harder to guess the contents of a disc.

Element Filters

This attributes in Filters are used for finer control of the file selections. These attributes are applied only to files, folders are not affected. Filters limit the files that are selected for the output media. Regular expressions or DOS style formats can be used for either including or excluding names. It is also possible to restrict files that have been modified before or after a specified date and time. Filters are applied after any other selections like “*.txt” in a DstGroup or SrcDst have been used.

Both include and exclude expressions may be applied. If both are used then include expressions are evaluated first. It is not recommended to use both DOS and regular expressions.

Attribute RegExInclude CDATA #IMPLIED

This is the regular expression used to specify names that are selected.

Attribute RegExExclude CDATA #IMPLIED

This regular expression is used to prevent names from being used. The RegExInclude expression is evaluated first, and then the exclude one is checked to make sure the name is not to be excluded.

Attribute DOSPatternInclude CDATA #IMPLIED

These expressions use the DOS wildcards ‘*’ and ‘?’. ‘?’ matches and single character while ‘*’ matches anything from the current location to the end of either the name (without extension) or the extension.

Attribute WildcardInclude CDATA #IMPLIED

This is identical to DOSPatternInclude.

Attribute DOSPatternExclude CDATA #IMPLIED

Any files matching the exclude pattern will be rejected.

This is the same as saying: “I do NOT want files included that meet this criteria. For instance, the user can create the filter ‘*.doc’ to exclude all doc files.

DosPatternExclude=”*.doc”

Src="c:\MyFiles\" Dst="\MyFiles\”

Attribute WildcardExclude CDATA #IMPLIED

This is identical to DOSPatternExclude.

Attribute ModTimeBefore CDATA #IMPLIED

Using CCYY-MM-DD HH:MM:SS this will only allow files with a modify date before or including the specified date.

Attribute ModTimeAfter CDATA #IMPLIED

This filter would only allow files with a modify date equal to or later than the specified date.

DOS Wildcards

A filter is a string that follows the conventions of DOS wildcarding:

Wildcards are characters that can be used to stand-in for unknown characters in file names. In DOS, wildcard characters can match up with any character that is allowable in a file name. There are two wildcards in DOS: * and ?.

***** matches any combination of allowable characters. The asterisk character, *****, can stand in for any number of characters until the '.' or the end of the extension.

Example: ***.doc** matches any file that ends with .doc.

? will match any single allowable character.

Example: **ab?.txt -** This filter finds any file with a three-letter name, of which the first two letters are **ab**, and with a **txt** extension. So files like abz.txt and ab2.txt are included on the disc.

Example: ***ab?.do? -** This looks in the folder for files that have anywhere from 1 to 5 beginning characters, followed by **ab** followed by one character, and which have an extension of do followed by any one character.

Example: "a*.txt" matches all files starting with 'a' with extension .txt.

Example: "a?.txt" matches files starting with a and having two letter names with extension .txt.

For example, the user chooses a folder named "C:\MyFolder" for inclusion on a CD and this folder contains the following files:

- File1.jpg
- File2.jpg
- File3.cpp
- File4.cpp
- File5.h

Perhaps the user only wants to retrieve files with the ".cpp" or ".h" extension since they are programming language code files that should be backed up to CD. "*.cpp" or "*.h" would work for this.

For more complex name selection it is recommended that regular expressions be used.

Regular Expressions

The regular expression filter strings use a subset of full regular expression pattern matching that is useful for more sophisticated file name matching.

The following table shows the regular expression meta-characters that are supported and how they work.

Meta-character	What it does
.	Matches any single character.
[ ]	Indicates a character class. Matches any character inside the brackets (for example, [abc] matches "a", "b", and "c").
^	If this metacharacter occurs at the start of a character class, it negates the character class. A negated character class matches any character except those inside the brackets (for example, [^abc] matches all characters except "a", "b", and "c"). If ^ is at the beginning of the regular expression, it matches the beginning of the input (for example, ^[abc] will only match input that begins with "a", "b", or "c").
-	In a character class, indicates a range of characters (for example, [0-9] matches any of the digits "0" through "9").
?	Indicates that the preceding expression is optional: it matches once or not at all (for example, [0-9][0-9]? matches "2" and "12").
+	Indicates that the preceding expression matches one or more times (for example, [0-9]+ matches "1", "13", "666", and so on).
*	Indicates that the preceding expression matches zero or more times
??, +?, *?	Non-greedy versions of ?, +, and *. These match as little as possible, unlike the greedy versions which match as much as possible. Example: given the input "<abc><def>", <.*?> matches "<abc>" while <.*> matches "<abc><def>".
$	At the end of a regular expression, this character matches the end of the input. Example: [0-9]$ matches a digit at the end of the input.
|	Alternation operator: separates two expressions, exactly one of which matches (for example, “T|the” matches "The" or "the").
!	Negation operator: the expression following ! does not match the input. Example: a!b matches "a" not followed by "b".
\	Escape character: interpret the next character literally (for example, [0-9]+ matches one or more digits, but [0-9]\+ matches a digit followed by a plus character). Also used for abbreviations (such as \a for any alphanumeric character; see following entries for specifics).
\a	Any alphanumeric character: ([a-zA-Z0-9])
\b	White space (blank): ([\\t])
\c	Any alphabetic character: ([a-zA-Z])
\d	Any decimal digit: ([0-9])
\h	Any hexadecimal digit: ([0-9a-fA-F])
\n	Newline: (\r|(\r?\n))
\q	A quoted string: (\"[^\"]*\")|(\'[^\']*\')
\w	A simple word: ([a-zA-Z]+)
\z	An integer: ([0-9]+)

Table 1 Regular Expression Meta-Characters

Regular expression example using several meta-characters.

^[a-h].*\.c.*$

This expression matches all files that start with the letter ‘a’, or ’b’, or ’c’, or ’d’, or ’e’, or ’f’, or ’g’, or ’h’ with 0 or more following characters and any file extension that starts with ‘c’.

Examples

BasePathGroup

An example combining BasePathGroup and regular expressions filters is shown here. This one includes jpg, tiff, and bmp files. Pay attention to the regular expressions. They are not DOS wildcards, so *.jpg does not do what you might think, in fact it is an illegal regular expression!

The first DstPath will get files from c:\files\images and put them in \images while the second DstPath will get them from c:\files\images\morefiles and put them in \images\morefiles.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE EditList SYSTEM "C:\Rimage\XML\EditList_1.6.dtd">
<EditList>
  <Filters RegExInclude=".+\.(jpg|tiff|bmp)"/>
  <BasePathGroup BasePath="C:\files">
    <DstGroup DstPath="\images">
      <File Name="*.*"/>
    </DstGroup>
    <DstGroup DstPath="\images\morefiles">
      <File Name="*.*"/>
    </DstGroup>
  </BasePathGroup>
</EditList>
```

Another example illustrating that the File Elements are optional. If no File elements are found then the filespec *.* will be used. This allows a simple form as shown in the following example. Note that this example also shows the ExpandFolders attribute that will recreate the tree on the media much like using the ParentFolder option in the Image Order. This example would place all of the files in c:\files on the media root folder. All the folders from c:\files and lower would also be created on the media.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE EditList SYSTEM "C:\Rimage\XML\EditList_1.6.dtd">
<EditList>
  <Options ExpandFolders="true"/>
  <BasePathGroup BasePath="C:\files">
    <DstGroup DstPath="\"/>
  </BasePathGroup>
</EditList>
```

SrcDst

This example shows how simple it can be to copy a tree into the image file system. Note that if *.* is not used, for example *.jpg, then only files matching that specification will be copied from all the folders referenced.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE EditList SYSTEM "C:\Rimage\XML\EditList_1.6.dtd">
<EditList>
  <Options ExpandFolders="true"/>
  <SrcDst Src="c:\pictures\*.*" Dst="\"/>
</EditList>
```

This example shows how to copy two folders.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE EditList SYSTEM "C:\Rimage\XML\EditList_1.6.dtd">
```

```
<EditList>
  <SrcDst Src="\\NEWDELL\TestSets\dir1\*.*"Dst="\dir1\"/>
  <Options ExpandFolders="true"/>
  <SrcDst Src=\\NEWDELL\TestSets\dir2\*.txt Dst="\dir2\"/>
</EditList>
```

VolumeGroup

Here are some examples of using VolumeGroup. All files outside of a VolumeGroup are just treated as normal files and will be packed according to normal spanning rules. Note also that the image order must have spanning enabled for VolumeGroup to be used. If spanning is off in the order the VolumeGroup options will simply be ignored.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE EditList SYSTEM "C:\Rimage\XML\EditList_1.6.dtd">
<EditList>
  <Options ExpandFolders="true"/>
  <Options VolumeGroup="start"/>
  <SrcDst Src="\\NEWDELL\TestSets\dir1\*.*"Dst="\dir1\"/>
  <Options VolumeGroup="end"/>
  <SrcDst Src="\\NEWDELL\TestSets\dir2\*.*"Dst="\dir2\"/>
  <SrcDst Src="\\NEWDELL\TestSets\dir3\*.*"Dst="\dir3\"/>
</EditList>
```

If more than one group is declared each group will be on its own volume unless VolumeGroupPacking is true. This example will create two volumes if VolumeGroupPacking is false. If VolumeGroupPacking is true it will produce one or two volumes depending on the size of the groups.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE EditList SYSTEM "C:\Rimage\XML\EditList_1.6.dtd">
<EditList>
  <Options ExpandFolders="true"/>
  <Options VolumeGroup="start"/>
  <SrcDst Src="\\NEWDELL\TestSets\dir1\*.*"Dst="\dir1\"/>
  <Options VolumeGroup="end"/>
  <Options VolumeGroup="start"/>
  <SrcDst Src="\\NEWDELL\TestSets\dir2\*.*"Dst="\dir2\"/>
  <Options VolumeGroup="end"/>
</EditList>
```

Here is an example showing how to place files on each copy of a spanned volume set. All of the files in folder dir1 will be on each volume of the set. The dir2 files will only be on one volume since they are outside of the “all” and “end” pair.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE EditList SYSTEM "C:\Rimage\XML\EditList_1.6.dtd">
<EditList>
  <Options ExpandFolders="true"/>
  <Options VolumeGroup="all"/>
  <SrcDst Src="\\NEWDELL\TestSets\dir1\*.*"Dst="\dir1\"/>
  <Options VolumeGroup="end"/>
  <SrcDst Src="\\NEWDELL\TestSets\dir2\*.*"Dst="\dir2\"/>
</EditList>
```