1 Publication Changes:

1.1 z/OS DFSMShsm Managing Your Own Data

SC23-6870-40

1.1.1 HBACKDS

A new keyword is added to the command: EXCLUDEF with an alias of EXCLF. The section explaining this new keyword will be combined with the section explaining the EXCLUDE keyword as the syntax/rules applied to exclude criteria are the same for both keywords. The new section will be updated as follows:

EXCLUDE and EXCLUDEF: Specifying files or directories to be excluded from processing

EXCLUDE | EXCLUDEF are mutually exclusive, optional parameters that are used to specify files and directories that are to be excluded during Unix File backup processing.

EXCLUDE('<*exclude_criteria*>') – specifies a comma-separated list of exclude criteria directly on the command. This criteria is applied to the UNIX files examined during the Unix File backup processing.

The exclude criteria list specified on the command with the EXCLUDE keyword has the following restrictions and rules unique to the keyword:

- The <*exclude_criteria*> list must be specified in single quotes
- The <*exclude_criteria*> list cannot exceed 1023 characters

EXCLUDEF('<*filepath*>') - specifies the UNIX path to a file that contains the exclude criteria to be applied to the UNIX files examined during the Unix File backup processing. The file path must be specified in single quotes and must be an absolute path. Any imbedded single quotes in the filename must be specified with a second single quote, e.g. the filename exclude'criteria.txt would be specified as: 'exclude''criteria.txt'. (optional)

The exclude criteria file specified using the EXCLUDEF keyword has the following restrictions and rules unique to the keyword:

- Line starting with a blank character is a comment.
- Each line can only contain one criterion.
- A single line cannot be longer than 1023 characters.
- Total length of all exclude criteria with added commas to separate them cannot exceed 32767 characters. For 'n' number of criteria, 'n-1' number of COMMA should also be accounted for to calculate the total size.

Exclude criteria, whether specified on the command in a comma separated list using the EXCLUDE keyword or specified within a file using the EXCLUDEF keyword, must conform to the following rules:

- Absolute directory path, beginning and ending with a forward-slash character.

- Partial directory path, ending with a forward-slash character. If the directory name begins with a forward slash character, then it is considered an absolute path. If the directory name does not begin with a forward slash character, it is considered a relative path that begins at the directory requested on the command. A directory name that is a relative path is considered relative to the directory requested on the command only, even when the command requests recursion.
- Filename pattern (does not contain forward slashes). Can contain * or ? wildcards. An asterisk (*) can match with 0 or more characters in a filename. A question mark (?) can match with a single character. Wildcard characters may be escaped with the backslash character if desired.
- A filename pattern can begin with an absolute or relative path, however wildcard characters can only be used in the filename pattern. A directory path containing an asterisk (*) or question mark (?) is considered an invalid pattern.
- If a filename contains special characters that are not to be treated as wildcards or escape characters i.e. an asterisk (*), question mark (?), or backslash (\), these characters must be escaped using a backslash (\). If these characters are not escaped, they will be treated as wildcards or escape characters.
- A directory wildcard may be specified in the directory portion of a criterion by specifying /.../ (<forward slash><period><period><period><forward slash>). A directory wildcard can match with 0 or more directories.
- If a criterion begins with a directory wildcard (/…/) then it is treated as a relative path.
- A comma cannot be used within the exclude criterion.
- A single criterion cannot exceed 1023 characters
- A single forward slash or a single asterisk are treated as invalid patterns.
- A directory path cannot contain consecutive forward slashes.

The example below shows sample criteria in a file specified by the EXCLUDEF keyword. The same criteria could be specified on the command using the EXCLUDE keyword where each criterion is separated by a comma.

Sample criteria

```
 Useful comment (maybe with commas)    <= blank in 1st position - ignored

                         <= empty line - ignored

/a/b/c/                      <= absolute directory

/a/b/c/x1.txt                <= absolute directory with filename

/a/.../d/.../g/syslog1.txt   <= absolute directory with wildcards

/a/.../d/.../g/*log?.txt     <= absolute directory and filename with wildcards

e/f/g/                       <= relative directory

e/f/g/mvs.log                <= relative directory with filename

d/e/f/g/*log?.txt            <= relative directory with filename wildcard

d/.../g/sys*.*               <= relative directory and filename with wildcards
```

```
/.../d/.../g/sys*.*          <= relative directory and filename with wildcards

*.log                        <= filename with wildcards

 *.log2                      <= blank in 1st position makes a comment, ignored

\*abc                   <= escaping wildcards - files with exact name *abc
\\abc                   <= escaping wildcards - files with exact name \abc
```

1.1.2 hbackup

The existing -e option is updated with additional valid exclude criteria:

e - Requests files or directories to be excluded from processing. Filename patterns that are specified in the exclude list can contain any valid filename characters and may contain the question mark (?) and asterisk (*) wildcard characters. The forward slash character is not allowed in the filename pattern.

If the pattern file contains any of the specific ?, *, or \ characters, the backslash character (\) must be used to escape that character to indicate it is not a wildcard. For example, \?, *, or \\).

- An asterisk (*) can be used to match any number of characters within a file name. It can also be used at any position in the pattern.
- A second asterisk may also be used at the end of the pattern to indicate any number of characters following the pattern is acceptable.
- A question mark (?) can be used to represent any single character in the file name. Multiple question mark characters are allowed.
- A directory wildcard may be specified in the directory portion of a criterion by specifying /.../ (<forward slash><period><period><period><forward slash>). A directory wildcard can match with 0 or more directories.
- A filename pattern can begin with an absolute or relative path, however wildcard characters can only be used in the filename pattern. A directory path containing an asterisk (*) or question mark (?) is considered an invalid pattern.

If the requested filename pattern does not include any wildcard characters, then that exact name is used to compare with files that are found in the subdirectories processed.

Directory names that are specified in the exclude list must end in a forward slash character. If the directory name begins with a forward slash character, then it is considered to be an absolute path. If the directory name does not begin with a forward slash character, it is considered to be a relative path that begins at the directory requested on the command.

A maximum of 1023 characters can be specified in the list. Do not use the comma (,) character in filename patterns or directory names.

A new option is added: -E

E - This specifies the UNIX path to a file that contains the exclude criteria to be applied to the UNIX files examined during the Unix File backup processing. The file path can be an absolute path or a path relative to the current working directory. Double quotes may need to be used to avoid additional processing by the UNIX shell.

This option is mutually exclusive with the -e option.

The contents of the exclude criteria files must conform to the following rules:

- Line starting with a blank character is a comment.
- Each line can only contain one criterion.
- A single line cannot be longer than 1023 characters.
- Total length of all exclude criteria with added commas to separate them cannot exceed 32767 characters. For 'n' number of criteria, 'n-1' number of COMMA should also be accounted for to calculate the total size.
- Absolute directory path, beginning and ending with a forward-slash character.
- Partial directory path, ending with a forward-slash character. If the directory name begins with a forward slash character, then it is considered an absolute path. If the directory name does not begin with a forward slash character, it is considered a relative path that begins at the directory requested on the command. A directory name that is a relative path is considered relative to the directory requested on the command only, even when the command requests recursion.
- Filename pattern (does not contain forward slashes). Can contain * or ? wildcards. An asterisk (*) can match with 0 or more characters in a filename. A question mark (?) can match with a single character. Wildcard characters may be escaped with the backslash character if desired.
- A filename pattern can begin with an absolute or relative path however, wildcard characters can only be used in the filename pattern. A directory path containing an asterisk (*) or question mark (?) is considered an invalid pattern.
- If a filename contains special characters that are not to be treated as wildcards or escape characters i.e. an asterisk (*), question mark (?), or backslash (\), these characters must be escaped using a backslash (\). If these characters are not escaped, they will be treated as wildcards or escape characters.
- A directory wildcard may be specified in the directory portion of a criterion by specifying /.../ (<forward slash><period><period><period><forward slash>). A directory wildcard can match with 0 or more directories
- If a criterion begins with a directory wildcard (/…/) then it is treated as a relative path.
- A comma cannot be used within the exclude criterion.
- A single criterion cannot exceed 1023 characters
- A single forward slash or a single asterisk are treated as invalid patterns.
- A directory path cannot contain consecutive forward slashes.

The example below shows sample criteria in a file specified by the EXCLUDEF keyword. The same criteria could be specified on the command using the EXCLUDE keyword where each criterion is separated by a comma.

Sample criteria

```
 Useful comment (maybe with commas)    <= blank in 1st position -
ignored

                           <= empty line - ignored

/a/b/c/                      <= absolute directory

/a/b/c/x1.txt                <= absolute directory with filename

/a/.../d/.../g/syslog1.txt   <= absolute directory with wildcards

/a/.../d/.../g/*log?.txt     <= absolute directory and filename with wildcards

e/f/g/                       <= relative directory

e/f/g/mvs.log                <= relative directory with filename

d/e/f/g/*log?.txt            <= relative directory with filename wildcard

d/.../g/sys*.*               <= relative directory and filename with wildcards

/.../d/.../g/sys*.*          <= relative directory and filename with wildcards

*.log                        <= filename with wildcards

 *.log2                      <= blank in 1st position makes a comment, ignored

\*abc                   <= escaping wildcards - files with exact name *abc
\\abc                   <= escaping wildcards - files with exact name \abc
```

1.1.3 HRECOVER

A new keyword is added to the command: EXCLUDEF with an alias of EXCLF. The section explaining this new keyword will be combined with the section explaining the EXCLUDE keyword as the syntax/rules applied to exclude criteria are the same for both keywords. The new section will be updated as follows.

EXCLUDE and EXCLUDEF: Specifying files or directories to be excluded from processing

EXCLUDE | EXCLUDEF are mutually exclusive, optional parameters that are used to specify files and directories that are to be excluded during Unix File recover processing.

EXCLUDE('<exclude_criteria>') – specifies a comma-separated list of exclude criteria directly on the command. This criteria is applied to the UNIX files examined during the UNIX File recover processing:

The exclude criteria list specified on the command with EXCLUDE keyword has the following restrictions and rules unique to the keyword:

- The <*exclude_criteria*> list must be specified in single quotes

- The <*exclude_criteria*> list cannot exceed 1023 characters

EXCLUDEF('<*filepath*>') - specifies the UNIX path to a file that contains the exclude criteria to be applied to the UNIX files examined during UNIX File recover processing. The file path must be specified in single-quotes and must be an absolute path. Any imbedded single quotes in the filename must be specified with a second single quote, e.g. the filename exclude'criteria.txt would be specified as: 'exclude''criteria.txt'. (optional)

The exclude criteria file specified using the EXCLUDEF keyword has the following restrictions and rules unique to the keyword:

- Line starting with a blank character is a comment.
- Each line can only contain one criterion.
- A single line cannot be longer than 1023 characters.
- Total length of all exclude criteria with added commas to separate them cannot exceed 32767 characters. For 'n' number of criteria, 'n-1' number of COMMA should also be accounted for to calculate the total size.

Exclude criteria, whether specified on the command in a comma separated list using the EXCLUDE keyword or specified within a file using the EXCLUDEF keyword, must conform to the following rules:

- Absolute directory path, beginning and ending with a forward-slash character.
- Partial directory path, ending with a forward-slash character. If the directory name begins with a forward slash character, then it is considered an absolute path. If the directory name does not begin with a forward slash character, it is considered a relative path that begins at the directory requested on the command. A directory name that is a relative path is considered relative to the directory requested on the command only, even when the command requests recursion.
- Filename pattern (does not contain forward slashes). Can contain * or ? wildcards. An asterisk (*) can match with 0 or more characters in a filename. A question mark (?) can match with a single character. Wildcard characters may be escaped with the backslash character if desired.
- A filename pattern can begin with an absolute or relative path, however wildcard characters can only be used in the filename pattern. A directory path containing an asterisk (*) or question mark (?) is considered an invalid pattern.
- If a filename contains special characters that are not to be treated as wildcards or escape characters i.e. an asterisk (*), question mark (?), or backslash (\), these characters must be escaped using a backslash (\). If these characters are not escaped, they will be treated as wildcards or escape characters.
- A directory wildcard may be specified in the directory portion of a criterion by specifying /.../ (<forward slash><period><period><period><forward slash>). A directory wildcard can match with 0 or more directories.
- If a criterion begins with a directory wildcard (/…/) then it is treated as a relative path.
- A comma cannot be used within the exclude criterion.
- A single criterion cannot exceed 1023 characters
- A single forward slash or a single asterisk are treated as invalid patterns.
- A directory path cannot contain consecutive forward slashes.

The example below shows sample criteria in a file specified by the EXCLUDEF keyword. The same criteria could be specified on the command using the EXCLUDE keyword where each criterion is separated by a comma.

Sample criteria

```
 Useful comment (maybe with commas)    <= blank in 1st position -
ignored

                         <= empty line - ignored

/a/b/c/                      <= absolute directory

/a/b/c/x1.txt                <= absolute directory with filename

/a/.../d/.../g/syslog1.txt   <= absolute directory with wildcards

/a/.../d/.../g/*log?.txt     <= absolute directory and filename with wildcards

e/f/g/                       <= relative directory

e/f/g/mvs.log                <= relative directory with filename

d/e/f/g/*log?.txt            <= relative directory with filename wildcard

d/.../g/sys*.*               <= relative directory and filename with wildcards

/.../d/.../g/sys*.*          <= relative directory and filename with wildcards

*.log                        <= filename with wildcards

 *.log2                      <= blank in 1st position makes a comment, ignored

\*abc                   <= escaping wildcards - files with exact name *abc
\\abc                   <= escaping wildcards - files with exact name \abc
```

1.1.4 Hrecover

The existing -e option is updated with additional valid exclude criteria:

e – Requests files or directories to be excluded from processing. Filename patterns that are specified in the exclude list can contain any valid filename characters and may contain the question mark (?) and asterisk (*) wildcard characters. The forward slash character is not allowed in the filename pattern.

If the pattern file contains any of the specific ?, *, or \ characters, the backslash character (\) must be used to escape that character to indicate it is not a wildcard. For example, \?, *, or \\).

- An asterisk (*) can be used to match any number of characters within a file name. It can also be used at any position in the pattern.
- A second asterisk may also be used at the end of the pattern to indicate any number of characters following the pattern is acceptable.

- A question mark (?) can be used to represent any single character in the file name. Multiple question mark characters are allowed.
- A directory wildcard may be specified in the directory portion of a criterion by specifying /.../ (<forward slash><period><period><period><forward slash>). A directory wildcard can match with 0 or more directories.
- A filename pattern can begin with an absolute or relative path, however wildcard characters can only be used in the filename pattern. A directory path containing an asterisk (*) or question mark (?) is considered an invalid pattern.

If the requested filename pattern does not include any wildcard characters, then that exact name is used to compare with files that are found in the subdirectories processed.

Directory names that are specified in the exclude list must end in a forward slash character. If the directory name begins with a forward slash character, then it is considered to be an absolute path. If the directory name does not begin with a forward slash character, it is considered to be a relative path that begins at the directory requested on the command.

A maximum of 1023 characters can be specified in the list. Do not use the comma (,) character in filename patterns or directory names.

A new option is added: -E

E - This specifies the UNIX path to a file that contains the exclude criteria to be applied to the UNIX files examined during Unix File recover processing. The file path can be an absolute path or a path relative to the current working directory. Double quotes may need to be used to avoid additional processing by the UNIX shell.

This option is mutually exclusive with the -e option.

The contents of the exclude criteria files must conform to the following rules:

- Line starting with a blank character is a comment.
- Each line can only contain one criterion.
- A single line cannot be longer than 1023 characters.
- Total length of all exclude criteria with added commas to separate them cannot exceed 32767 characters. For ‘n’ number of criteria, ‘n-1’ number of COMMA should also be accounted for to calculate the total size.
- Absolute directory path, beginning and ending with a forward-slash character.
- Partial directory path, ending with a forward-slash character. If the directory name begins with a forward slash character, then it is considered an absolute path. If the directory name does not begin with a forward slash character, it is considered a relative path that begins at the directory requested on the command. A directory name that is a relative path is considered relative to the directory requested on the command only, even when the command requests recursion.
- Filename pattern (does not contain forward slashes). Can contain * or ?

wildcards. An asterisk (*) can match with 0 or more characters in a filename. A question mark (?) can match with a single character. Wildcard characters may be escaped with the backslash character if desired.

- A filename pattern can begin with an absolute or relative path, however wildcard characters can only be used in the filename pattern. A directory path containing an asterisk (*) or question mark (?) is considered an invalid pattern.
- If a filename contains special characters that are not to be treated as wildcards or escape characters i.e. an asterisk (*), question mark (?), or backslash (\), these characters must be escaped using a backslash (\). If these characters are not escaped, they will be treated as wildcards or escape characters.
- Hidden directories are represented by /…/ (<forward slash><period><period><period><forward slash>). This syntax functions as a directory wildcard pattern and can match with 0 or more directories.
- If a criterion begins with a hidden directory pattern (/…/) then it is treated as relative path although the first character is a forward slash.
- A comma cannot be used within the exclude criterion.
- A single criterion cannot exceed 1023 characters
- A single forward slash or a single asterisk are treated as invalid patterns.
- A directory path cannot contain consecutive forward slashes.

The example below shows sample criteria in a file specified by the EXCLUDEF keyword. The same criteria could be specified on the command using the EXCLUDE keyword where each criterion is separated by a comma.

Sample criteria

```
 Useful comment (maybe with commas)       <= blank in 1st position -
ignored

                         <= empty line - ignored
/a/b/c/                      <= absolute directory

/a/b/c/x1.txt                <= absolute directory with filename

/a/.../d/.../g/syslog1.txt   <= absolute directory with wildcards

/a/.../d/.../g/*log?.txt     <= absolute directory and filename with wildcards

e/f/g/                       <= relative directory

e/f/g/mvs.log                <= relative directory with filename

d/e/f/g/*log?.txt            <= relative directory with filename wildcard

d/.../g/sys*.*               <= relative directory and filename with wildcards

/.../d/.../g/sys*.*          <= relative directory and filename with wildcards

*.log                        <= filename with wildcards

 *.log2                      <= blank in 1st position makes a comment, ignored
```

```
\*abc                   <= escaping wildcards - files with exact name *abc
\\abc                   <= escaping wildcards - files with exact name \abc
```

1.2 z/OS DFSMShsm Storage Administration

SC23-6871-40

1.2.1 BACKDS

A new keyword is added to the command: EXCLUDEF with an alias of EXCLF. The section explaining this new keyword will be combined with the section explaining the EXCLUDE keyword as the syntax/rules applied to exclude criteria are the same for both keywords. The new section will be updated as follows.

EXCLUDE and EXCLUDEF: Specifying files or directories to be excluded from processing

EXCLUDE | EXCLUDEF are mutually exclusive, optional parameters that are used to specify files and directories that are to be excluded during Unix File backup processing.

EXCLUDE(<exclude_criteria>) – specifies a comma-separated list of exclude criteria directly on the command. This criteria is to be applied to the UNIX files examined during the UNIX File backup processing.

The exclude criteria list specified on the command with EXCLUDE keyword has the following restrictions and rules: unique to the keyword.

- The *exclude_criteria* list must be specified in single quotes
- The *exclude_criteria* list cannot exceed 1023 characters

EXCLUDEF(‘<*filepath*>’) - specifies the UNIX path to a file that contains the exclude criteria to be applied to the UNIX files examined during the Unix File backup processing. The file path must be specified in single-quotes and must be an absolute path. Any imbedded single quotes in the filename must be specified with a second single quote, e.g. the filename exclude’criteria.txt would be specified as: ‘exclude’’criteria.txt’. (optional)

The exclude criteria file specified using the EXCLUDEF keyword has the following restrictions and rules unique to the keyword:

- Line starting with a blank character is a comment.
- Each line can only contain one criterion.
- A single line cannot be longer than 1023 characters.
- Total length of all exclude criteria with added commas to separate them cannot exceed 32767 characters. For ‘n’ number of criteria, ‘n-1’ number of COMMA should also be accounted for to calculate the total size.

Exclude criteria, whether specified on the command in a comma separated list using the EXCLUDE keyword or specified within a file using the EXCLUDEF keyword, must conform to the following rules:

- Absolute directory path, beginning and ending with a forward-slash character.
- Partial directory path, ending with a forward-slash character. If the directory name begins with a forward slash character, then it is considered an absolute path. If the directory name does not begin with a forward slash character, it is considered a relative path that begins at the directory requested on the command. A directory name that is a relative path is considered relative to the directory requested on the command only, even when the command requests recursion.
- Filename pattern (does not contain forward slashes). Can contain * or ? wildcards. An asterisk (*) can match with 0 or more characters in a filename. A question mark (?) can match with a single character. Wildcard characters may be escaped with the backslash character if desired.
- A filename pattern can begin with an absolute or relative path, however wildcard characters can only be used in the filename pattern. A directory path containing an asterisk (*) or question mark (?) is considered an invalid pattern.
- If a filename contains special characters that are not to be treated as wildcards or escape characters i.e. an asterisk (*), question mark (?), or backslash (\), these characters must be escaped using a backslash (\). If these characters are not escaped, they will be treated as wildcards or escape characters.
- A directory wildcard may be specified in the directory portion of a criterion by specifying /.../ (<forward slash><period><period><period><forward slash>). A directory wildcard can match with 0 or more directories.
- If a criterion begins with a directory wildcard (/…/) then it is treated as relative path.
- A comma cannot be used within the exclude criterion.
- A single criterion cannot exceed 1023 characters
- A single forward slash or a single asterisk are treated as invalid patterns.
- A directory path cannot contain consecutive forward slashes.

The example below shows sample criteria in a file specified by the EXCLUDEF keyword. The same criteria could be specified on the command using the EXCLUDE keyword where each criterion is separated by a comma.

Sample criteria

```
 Useful comment (maybe with commas)    <= blank in 1st position - ignored

                            <= empty line - ignored

/a/b/c/                      <= absolute directory

/a/b/c/x1.txt                <= absolute directory with filename

/a/.../d/.../g/syslog1.txt   <= absolute directory with wildcards

/a/.../d/.../g/*log?.txt     <= absolute directory and filename with wildcards

e/f/g/                       <= relative directory
```

```
e/f/g/mvs.log                <= relative directory with filename

d/e/f/g/*log?.txt            <= relative directory with filename wildcard

d/.../g/sys*.*               <= relative directory and filename with wildcards

/.../d/.../g/sys*.*          <= relative directory and filename with wildcards

*.log                        <= filename with wildcards

 *.log2                      <= blank in 1st position makes a comment, ignored

\*abc                   <= escaping wildcards - files with exact name *abc
\\abc                   <= escaping wildcards - files with exact name \abc
```

1.2.2 RECOVER

A new keyword is added to the command: EXCLUDEF with an alias of EXCLF. The section explaining this new keyword will be combined with the section explaining the EXCLUDE keyword as the syntax/rules applied to exclude criteria are the same for both keywords. The new section will be updated as follows.

EXCLUDE and EXCLUDEF: Specifying files or directories to be excluded from processing

EXCLUDE | EXCLUDEF are mutually exclusive, optional parameters that are used to specify files and directories that are to be excluded during Unix File recover processing.

EXCLUDE('<exclude_criteria>') – specifies a comma-separated list of exclude criteria directly on the command. This criteria is to be applied to the UNIX files examined during the UNIX File recover processing.

The exclude criteria list specified on the command with EXCLUDE keyword has the following restrictions and rules unique to the keyword.

- The *exclude_criteria* list must be specified in single quotes
- The *exclude_criteria* list cannot exceed 1023 characters

EXCLUDEF('<*filepath*>') - specifies the UNIX path to a file that contains the exclude criteria to be applied to the UNIX files examined during the Unix File recover processing. The file path must be specified in single-quotes and must be an absolute path. Any imbedded single quotes in the filename must be specified with a second single quote, e.g. the filename exclude'criteria.txt would be specified as: 'exclude''criteria.txt'. (optional)

The exclude criteria file specified using the EXCLUDEF keyword has the following restrictions and rules unique to the keyword:

- Line starting with a blank character is a comment.
- Each line can only contain one criterion.
- A single line cannot be longer than 1023 characters.
- Total length of all exclude criteria with added commas to separate them cannot

exceed 32767 characters. For ‘n’ number of criteria, ‘n-1’ number of COMMA should also be accounted for to calculate the total size.

Exclude criteria, whether specified on the command in a comma separated list using the EXCLUDE keyword or specified within a file using the EXCLUDEF keyword, must conform to the following rules:

- Absolute directory path, beginning and ending with a forward-slash character.
- Partial directory path, ending with a forward-slash character. If the directory name begins with a forward slash character, then it is considered an absolute path. If the directory name does not begin with a forward slash character, it is considered a relative path that begins at the directory requested on the command. A directory name that is a relative path is considered relative to the directory requested on the command only, even when the command requests recursion.
- Filename pattern (does not contain forward slashes). Can contain * or ? wildcards. An asterisk (*) can match with 0 or more characters in a filename. A question mark (?) can match with a single character. Wildcard characters may be escaped with the backslash character if desired.
- A filename pattern can begin with an absolute or relative path, however wildcard characters can only be used in the filename pattern. A directory path containing an asterisk (*) or question mark (?) is considered an invalid pattern.
- If a filename contains special characters that are not to be treated as wildcards or escape characters i.e. an asterisk (*), question mark (?), or backslash (\), these characters must be escaped using a backslash (\). If these characters are not escaped, they will be treated as wildcards or escape characters.
- A directory wildcard may be specified in the directory portion of a criterion by specifying /.../ (<forward slash><period><period><period><forward slash>). A directory wildcard can match with 0 or more directories.
- If a criterion begins with a directory wildcard (/…/) then it is treated as relative path.
- A comma cannot be used within the exclude criterion.
- A single criterion cannot exceed 1023 characters
- A single forward slash or a single asterisk are treated as invalid patterns.
- A directory path cannot contain consecutive forward slashes.

The example below shows sample criteria in a file specified by the EXCLUDEF keyword. The same criteria could be specified on the command using the EXCLUDE keyword where each criterion is separated by a comma.

Sample criteria

```
 Useful comment (maybe with commas)    <= blank in 1st position - ignored

                           <= empty line - ignored

                         <= empty line - ignored
/a/b/c/                      <= absolute directory

/a/b/c/x1.txt                <= absolute directory with filename

/a/.../d/.../g/syslog1.txt   <= absolute directory with wildcards
```

```
/a/.../d/.../g/*log?.txt     <= absolute directory and filename with wildcards

e/f/g/                       <= relative directory

e/f/g/mvs.log                <= relative directory with filename

d/e/f/g/*log?.txt            <= relative directory with filename wildcard

d/.../g/sys*.*               <= relative directory and filename with wildcards

/.../d/.../g/sys*.*          <= relative directory and filename with wildcards

*.log                        <= filename with wildcards

 *.log2                      <= blank in 1st position makes a comment, ignored

\*abc                   <= escaping wildcards - files with exact name *abc
\\abc                   <= escaping wildcards - files with exact name \abc
```

1.3 z/OS MVS System Messages Vol 2

SA38-0669-30

1.3.1 Changed messages

1.3.1.1 ARC1334I BACKUP VERSION NOT CREATED

The REAS36 explanation will be complemented with EXCLUDEF keyword and -E option description:

Reascode
Meaning

36 Syntax error in the EXCLUDE list **specified by the EXCLUDE keyword or -e option:**

- length of the filename pattern is higher than 255
- length of absolute directory path s higher than 1023
- length of the relative directory path is higher than 1022
- incorrect using of escape character
- forward slash in the filename pattern
- entry with only single asterisk or forward slash or question mark
- several consecutive slashes in the directory path

Programmer Response:
If EXCLUDE keyword or -e option specified, check the EXCLUDE list correctness, correct it and reissue the request.

If EXCLUDEF keyword or -E option specified, examine the ARC1453I message output, correct the reason of the error and reissue the request.

1.3.1.2 ARC1158I RECOVERY DATA SET/FILE FAILED

The REAS76 explanation will be complemented with EXCLUDEF keyword and -E option description:

Reascode

Meaning

76 Syntax error in the EXCLUDE list **specified by the EXCLUDE keyword or -e option:**

- length of the filename pattern is higher than 255
- length of the relative directory path is higher than 1022
- empty entry in the list
- incorrect using of escape character
- forward slash in the filename pattern
- entry with only single asterisk or forward slash
- several consecutive slashes in the directory path

Programmer Response:

If EXCLUDE keyword or -e option specified, check the EXCLUDE list correctness, correct it and reissue the request.

If EXCLUDEF keyword or -E option specified, examine the ARC1453I message output, correct the reason of the error and reissue the request.

1.3.1.3 ARC1605I COMMAND HAS PARSE ERROR

The REAS58 will be modified to reflect that the file path specified by EXCLUDEF or -E option must not exceed 1023 symbols.

Reas58

Syntax error in the EXCLUDE list:

- Length of the EXCLUDE list is greater than 1023
- Length of the specified EXCLUDEF file path is greater than 1023
- Specified EXCLUDEF file path is a relative path. Specified EXCLUDEF file path must be an absolute path.

Programmer response:

Correct the EXCLUDE list specified by EXCLUDE (-e) option or the exclude file path specified by EXCLUDEF (-E) option and reissue the request.

1.3.2 New messages

1.3.2.1 ARC1453I

ARC1453I ERROR IN EXCLUDE LIST FILE *filepath*, *text*

Explanation

An error has occurred during processing exclude list file *filepath*. The *text* describes the reason of the error:

- INVALID EXCLUDEF CRITERIA, LINE# nnnnnnnn
  Syntax error in one or more criteria inside EXCLUDEF file
- INVALID FILE PATH
  filepath is not an absolute path or has trailing slash.
- FILE NOT FOUND
  filepath is not found
- NOT REGULAR FILE
  filepath is not a Regular file type (for example – it could be a Directory).
- NOT AUTHORIZED
  The user is not authorized to read the file specified by *filepath*. This message is accompanied by the message ARC1451I.
- UNIX OR RACF SYSCALL FAILED
  This message is accompanied by the messages ARC1449I or ARC1450I.
- *service_name* RETCODE=*xxxxxxxx* RSNCODE=*xxxxxxxx*
  z/OS UNIX callable service *service_name* returned an unexpected error.
- INSUFFICIENT STORAGE
  Temporary storage shortage.
- TOO LONG EXCLUDE LIST
  Total length of all exclude criteria with added commas to separate them exceeds 32767 characters.
- COMMA IN LINE *xxxxxxxx*
  A comma is used within the exclude criterion.
- TOO LONG LINE *xxxxxxxx*
  A single line is longer than 1023 characters.
- INVALID EXCLUDEF CRITERIA, LINE# *xxxxxxxx*
  Invalid EXCLUDE criterion is specified.
- BPX1RPH RETCODE=xx RSNCODE=yy
  Problem in getting the real filepath by BPX1RPH.
- BPX1OPN RETCODE=xx RSNCODE=yy
  Problem in opening the EXCLUDEF file
- BPX1RED RETCODE=xx RSNCODE=yy
  Problem in reading the EXCLUDEF file
- BPX1LSK RETCODE=xx RSNCODE=yy
  Problem in changing the file offset while processing a buffer read earlier.
- BPX1CLO RETCODE=xx RSNCODE=yy
  Problem in closing the EXCLUDEF file

System action

DFSMShsm rejects the command with this exclude list file.

User response.
Check and modify the file *filepath* with exclude list.

Programmer response
None.

Source
DFSMShsm

1.4 z/OS DFSMShsm Diagnosis

GC52-1387-40

1.4.1 Using Patches For Problem Determination

Chapter 9. Using patches for problem determination is updated to include a new section titled “Record all EXCLUDEF criteria in the PDA trace dataset during UNIX file backup/recovery” after the section titled “Diagnosing errors when accessing the cloud”. An update to the Chapter 9. Introduction is also made to list the new section.

Problem determination patches

The problem determination patches are designed to aid users in diagnosing problems. Some of the patches can appropriately be added as startup members, while others are appropriate only for temporary usage.

The following table shows brief descriptions and the associated procedures for all the DFSMShsm supported problem determination patches.

.

.

.

Analyzing the CELL POOL free chain page 95	“Analyzing the CELL POOL free chain” on
Record all EXCLUDEF criteria on trace dataset during Unix file backup/recovery	**“Record all EXCLUDEF criteria on trace dataset during Unix file backup/recovery” on page 97.**

Record all EXCLUDEF criteria in the PDA trace dataset during Unix file backup/recovery.

During Unix file backup or recover processing, the patch described below can be enabled in order to capture all criteria read from an EXCLUDEF file. This patch records all criteria in the

PDA trace datasets before performing any validation of the criteria. (This records all criteria as found in the EXCLUDEF file).

The following PATCH command turns this functionality ON:

```
PATCH .ARCCVT.+02E BITS(....1...)
```

The following PATCH command turns this functionality OFF:

```
PATCH .ARCCVT.+02E BITS(....0...)
```

1.4.2 Entries That Pass Error Codes to ARCERP

Table 3. Entries That Pass Error Codes to ARCERP is updated with an entry for ARCZUXLT in between ARCZUPIN and ARCZVALB.

Table 3. Entries That Pass Error Codes to ARCERP

Error Code	Process Level	Description
ARCZUPIN		
4*nn*	Debug	Parameter list is not valid
ARCZUXLT		
40n	**Snap**	**Parameter list is not valid**
41n	**Debug**	**Internal error**
ARCZVALB		
16	Fatal	I/O error in CDS.
nn	Fatal	Return code from ARCZVSAM.
400	Debug	Parameter list errors.

1.5 z/OS DFSMShsm Data Areas

GC14-7504-50

1.5.1 MCVT—Management Communication Vector Table

The following table is updated to include a new patch bit in MCVT for enabling/disabling EXCLUDE/EXCLUDEF criteria tracing.

Table 111. MCVT—Management Communication Vector Table

Offsets				
Decimal (Hex)	**Type**	**Length**	**Name**	**Description**
46 (2E)	BITSTRING	2	MCVTRACE	The next 2 bytes contain the trace options:
	1...		MCVTRCAT	When set to 1, changes to the control data sets are Traced in the log data set.
	.1..		MCVTFTRC	When set to 1, the problem determination aid Trace function is active.

..1.	MCVTBDSMF	Trace ARCBDSMF calls.
...1	MCVTZMASK	Trace ARCZMASK calls.
.... 1...	**MCVTUXLT**	**Trace criteria on EXCLUDEF file.**

1.5.2 MCVT Data Area Cross-Reference Table

The following table is updated to include a new patch bit in MCVT for enabling/disabling EXCLUDE/EXCLUDEF criteria tracing.

Table 113. MCVT Data Area Cross-Reference Table (continued)

Name	Hex offset	Hex Value	Struct Level
MCVTUSR3	3E0		2
MCVTUSR4	3E4		2
MCVTUUDT	1D0		2
MCVTUXLT	**2E**	**08**	**3**
MCVTVACH	270		2
MCVTVACT	274		2

1.5.3 MWE—Extension for Data Set Backup Enhancements (MWE—Extension for Data Set Backup Enhancements (MWE—Management Work Element

The following table, “Table 136. MWE—Additional UNIX data stored in MWEDSN area”, is updated to include EXCLUDE/EXCLUDEF fields.

Table 136. MWE—Additional UNIX data stored in MWEDSN area

Offsets Decimal	(Hex)	Type	Length	Name	Description
148	(94)		44	MWE_UNIX_DATA	UNIX request specifics
148	(94)	CHARACTER	1	MWE_UNIX	Set to / for UNIX files
149	(95)	CHARACTER	1	MWE_UNIX_CONFLICT	Indicator of file/directory
150	(96)	CHARACTER	2	*	Reserved - always 0, no conflict with MWESGL
152	(98)	CHARACTER	12	*	Reserved
164	(A4)	BITSTRING	4	MWE_UNIX_FLAGS	Flags for UNIX
		1...		MWE_UF_REG	Spawned MWE registered
		.x..		*	Reserved
		..1.		MWE_UF_EXCLD_REQ	Request with EXCLUDE list
		...1		MWE_UF_EXCLD_PASSED	ON = at least one entry passed EXCLUDE filter
		.... xx..		*	Reserved
		.... ..1.		MWE_UF_EXCLDF_REQ	EXCLUDE file specified
		x		*	Reserved
168	(A8)	CHARACTER	4	*	Reserved
172	(AC)	PTR	4	MWE_EXCLD_TABL@	Pointer to offset table
176	(B0)	PTR	4	MWE_EXCLD_AREA@	Pointer to EXCLUDE area
180	(B4)	CHARACTER	4	*	Reserved
184	(B8)	FIXED	4	MWE_EXCLUDE_OFF	EXCLUDE structure offset
188	(BC)	CHARACTER	4	*	Reserved

A New table is added to document the EXCLUDE Struct with a note explaining the offset. This struct is present when a backup or recovery request is made with the EXCLUDE (-e) or EXCLUDEF (-E) keywords (options).

Table 141. MWE—EXCLUDE Struct

Offsets Decimal	(Hex)	Type	Length	Name	Description
		STRUCTURE		MWE_EXCLUDE_STRUCT	EXCLUDE patterns list/file structure
		FIXED	2	MWE_EXCLUDE_LEN	Length of Exclude Criteria list or

			length of path to file containing exclude criteria
CHARACTER	*	MWE_EXCLUDE_LIST	Exclude criteria list or path to file containing exclude criteria

Note: offset of this structure is MWE_EXCLUDE_OFF

1.5.4 MWE — Cross-Reference Table

The following table is updated to include the EXCLUDE/EXCLUDEF fields and the note at the bottom explaining the offsets for MWE_EXCLUDE_STRUCT, MWE_EXCLUDE_LEN, and MWE_EXCLUDE_LIST

Table 142. MWE — Cross-Reference Table (continued)

Name	**Hex offset**	**Hex Value**	**Struct Level**
MWE_END	120		2
MWE_EXCLD_AREA@	B0		3
MWE_EXCLD_TABL@	AC		3
MWE_EXCLUDE_LEN*			3
MWE_EXCLUDE_LIST*			3
MWE_EXCLUDE_OFF	B8		3
MWE_EXCLUDE_STRUCT*			1
MWE_F_UNIX	84	04	4
.			
.			
.			
MWE_TIME	248		3
MWE_UF_EXCLD_PASSED	A4	10	3
MWE_UF_EXCLD_REQ	A4	20	3
MWE_UF_EXCLDF_REQ	A4	02	3
MWE_UF_REG	A4	80	3

* The offset for MWE_EXCLUDE_STRUCT is variable and is the value specified by MWE_EXCLUDE_OFF. The offset values for MWE_EXCLUDE_LIST and MWE_EXCLUDE_LEN are relative to the offset for MWE_EXCLUDE_STRUCT. See Table 141 for details.