XFIND V1.5

The tool XFIND has been created to perform complex searches with "find" in a z/OS UNIX sysplex sharing environment. Only file systems that are managed locally are used for searching.

- If the file system is a "rwshare" mounted zFS aggregate, the search is done on the zFS-owning system.

- In all other situations searching is done on the USS-owning System.

- This is sufficient and identifies the system where to allow the search regarding the actual file system.

- Important: NFS file systems, as being "remote" file systems, are not searched. Just "local" z/OS file systems are searched. (This behavior could be changed but there is not yet an option.)

- Doing all this avoids unnecessary XCF traffic and overhead. Otherwise, this could end in needing much more time.

- Even in case of zFS file systems that are mounted sysplex-aware and read-write (rwshare) it is recommended to process find commands which result in metadata searches on the zfs owning system as this is fully processed on the owning system. DIO support solves the problem for direct access for file system data on non-owning systems but not for metadata searches.

- You can call the utility using three possibilities.

  (a) It can be executed as a SYSREXX routine, named XFIND, from a console locally or on all systems in parallel.

  (b) It can be started with name "lfind" locally from z/OS UNIX. Using BPXBATCH (and if you need to specify special characters or us it anyway already with exploitation of RXBATCH) you can easily start jobs with system-affinity and address the execution to all systems in parallel as with XFIND.

  (c) There is another version named "lf2stdout", which should be simply defined as a symbolic link pointing to "lfind", that simply writes its output data to STDOUT instead of into a specific UNIX file.

Using XFIND from a console

For demonstration the challenge to list all files that contain at least one space somewhere in the name is chosen. To do so the following command can be entered:

```
route *all,f axr,xfind,t=0 '/ -name "* *" -type f'
```

This starts the System REXX routine on all systems in the sysplex and runs the find command locally for all file systems that are "locally owned". In the end on each system a set of three files is written to the UNIX home directory of the user, that started the XFIND command. Following a sample is shown for system SC63.

XFIND sample output files and data

Type	Perm	Owner	------Size	Filename
_ File	640	HERING	0	XFIND.SC63.error_log
_ File	640	HERING	18973	XFIND.SC63.msg_log
_ File	640	HERING	1225	XFIND.SC63.uss_entries

Sample Msg_Log Data

```
find parm list: "/" "-name" "* *" "-type" "f"

2012-05-06 18:15:45 Starting ...
2012-05-06 18:15:45 Processing SC63 AUTOMNT file system *AMD/u ...
2012-05-06 18:15:45 Skipping SC64 TFS file system /SC64/TMP ...
2012-05-06 18:15:45 Skipping SC70 TFS file system /SC70/TMP ...
```

```
2012-05-06 18:15:45 Processing SC63 TFS file system /SC63/TMP ...
2012-05-06 18:15:45 Skipping SC65 TFS file system /SC65/TMP ...
2012-05-06 18:15:45 Processing SC63 ZFS file system OMVS.ADMR1.HFS ...
2012-05-06 18:15:45 Processing SC63 ZFS file system OMVS.ADMR2.HFS ...
2012-05-06 18:15:45 Skipping SC64 ZFS file system BB38564.SBBOHFS ...
2012-05-06 18:15:45 Processing SC63 ZFS file system OMVS.ADMR4.HFS ...
...
2012-05-06 18:18:30 Processing SC63 HFS file system IWH.V7R1M0.DWC.HFS ...
2012-05-06 18:18:30 Processing SC63 HFS file system OMVS.WTSCPLX2.TOOLS ...
2012-05-06 18:18:30 Processing SC63 HFS file system OMVS.STC.HFS ...
2012-05-06 18:18:30 Stopping ...
2012-05-06 18:18:30 XFIND on SC63    : Processing ended.
```

Usage of "lfind" from within a RXBATCH job

If there is a problem with special characters in the "find" command, the usage of the alternative "lfind" in an RXBATCH job can be the simple solution. This can be a good way for processing in other situations as well. By setting a system affinity the job can be started in parallel on all systems!

In the following example it is requested to show the most important attributes in addition to the names.

```
//RXBATCH  EXEC PGM=IKJEFT01,PARM=RXBATCH
//SYSEXEC  DD DSNAME=&REXXLIB.,DISP=SHR
//STDIN    DD DATA,DLM=##
lfind / -name "* *" -type f -exec ls -E {} ";"
RC=$?; if [ $RC -ne 0 ]; then echo "\nRc=$RC"; exit 1; else exit 0; fi
##
//RXBPARM  DD DATA,DLM=##
 _RXBATCH_SWSU=0        < 0= no switch (default), 1= switch to SU mode
_RXBATCH_LOGIN=0        < 0= no login shell, 1= login shell (default)
   _RXBATCH_SL=1        < 0= use BPXBATSL if in SU mode, 1= use always
_RXBATCH_CP=IBM-273     < code page of STDIN data, default is IBM-1047
##
//STDENV   DD DATA,DLM=##
PATH=/usr/local/bin:/bin
##
//STDOUT   DD SYSOUT=*,LRECL=136,RECFM=VB
//SYSTSIN  DD DUMMY
//SYSTSPRT DD DUMMY    < Use "SYSOUT=*" instead in case of problems
//*STDPARM  DD ...     < Do not add a STDPARM DD statement to this JCL
```

The REXX library is assigned via a job variable. The special code page specific characters in the "lfind" command and in STDIN can be specified as desired based on setting the code page that you use to create the data, for example as shown here by using "_RXBATCH_CP=IBM-273" to say that it is a German code page. The output is provided as done when using XFIND. However, the names start with "lfind" instead of "XFIND".

Usage of "lf2stdout" instead of "lfind"

You can define a symlink named lf2stdout that points to lfind if you want to have all the STDOUT, STDERR and the messages output written to STDOUT and STDERR instead of a file in the user's home directory. This might be the best way to use this function in the meantime as you get all pieces together in the job output. Following is a sample job created for ccsid 1141 and a user with read access to BPX.SUPERUSER in class FACILITY.

```
//UNIX     EXEC PGM=BPXBATSL,PARM='PGM /bin/sh -c $STDIN!$ICONV!su'
//STDIN    DD DATA,DLM=##
lf2stdout /u/hering -level 0 -name "* *" -type f -exec ls -ld {} \;
RC=$?
if [ $RC -ne 0 ]; then echo "\nRc=$RC" >&2; exit 1; else exit 0; fi
##
//STDENV   DD DATA,DLM=##
STDIN=/bin/cat //dd:STDIN
ICONV=iconv -f1141 -t1047
_BPX_BATCH_UMASK=0022
_BPX_SHAREAS=YES
PATH=/usr/local/bin:/bin
##
//STDOUT   DD SYSOUT=*,LRECL=4096,RECFM=VB
//STDERR   DD SYSOUT=*,LRECL=136,RECFM=VB
```

Beneath the resulting hits in STDOUT you may get messages like the following in STDERR.

```
find parm list: "/u/hering" "-level" "0" "-name" "* *" "-type" "f" …

2021-03-22 10:38:50 Starting ...
2021-03-22 10:38:50 Processing SC70 ZFS file system OMVS.HERING.ZFS ...
2021-03-22 10:38:50 Stopping ...
lf2stdout on SC70    : Processing ended.

--
```