

BREXX/370 V2R2M0 Software Notices

Document Version 1.0 created on 6. March 2019

Authors: Peter Jacob (pej), Mike Großmann (mig)

Abstract

This document covers the changes in the new version, migration instruction to upgrade from the previous Release. The installation process is separately described in the BREXX/370 Installation document.

Upgrade from a previous BREXX/370 Version

Before upgrading to BREXX/370 V2 R1.0 backup your system. The easiest way is creating a copy of your TK4- directory containing all of your settings DASD volumes. In the cases of errors or unwanted behaviour, you can easily recover to the backup version.

Important Changes

Due to the extended calling functionality in the new version of BREXX/370 an import of required REXX scripts is no longer necessary. For this reason, all pre-defined import libraries have been removed from the JCL Procedures RXTSO and RXBATCH. The installation will update them in SYS2.PROCLIB. For similar reasons the CLISTs RX and REXX are no longer necessary and will be therefore removed from SYS2.COMDPROC, there will be an RX and REXX member in SYS2.LINKLIB which replaces the CLIST version.

Important Notice

If you made changes or extensions in the PROCLIB Member RXTSO and/or RXBATCH save the changes to re-introduce them in the newly installed version

If you made changes or extensions in the COMDPROC Members RX and/or REXX save them to incorporate them in a newly created function of your own.

Libraries

The following BREXX libraries are necessary for running REXX:

```
BREXX.JCL
BREXX.SAMPLES
BREXX.SAMPLIB      *** new with this version ***
BREXX.RXLIB
```

They will be delivered and created during the installation process, existing libraries will be overwritten!

Important Notice

If you made changes or added your own entries in one of these libraries, save them before beginning with the installation process!

Calling external REXX Scripts or Functions

It is now possible to call external REXX scripts, either by:

CALL your-script parm1,parm2... or a function call

BREXX/370 V2R2M0 Software Notices

Value=your-script(parm1,parm2,...)

The called script will be sought by the following sequence:

- Internal subprocedure or label (contained in the running script)
- current library (where the calling REXX is originated)
- BREXX.RXLIB

Important Notice

The variable scope slightly differs from IBM's REXX implementation. In IBM's implementation a call to an external REXX script, the called REXX is always treated as a procedure without access to the caller's variable pool. BREXX can access and update the caller's pool. If a PROCEDURE is used in the called BREXX script, variables can be made available by the EXPOSE statement.

Software Changes requiring actions

The STORAGE function has been changed to become compatible with IBM's z/OS REXX STORAGE version.

In IBM's REXX, the storage address must be in hex format, in BREXX it was decimal. With this version, we match with IBM's specification and allow only hex notation.

Important Notice

If you have created REXX scripts using the STORAGE function and dislike to update all of them, you can replace them by the BSTORAGE function, which still works with decimal addresses. BSTORAGE is a REXX function being part of the BREXX: RXLIB library.

New Functionality

BREXX functions coded in REXX

ABEND(abend-code)

Terminate program with Abend-Code, produces an SYSUDUMP

USERID()

Signed in UserId (available in Batch and Online)

WAIT(wait-time)

Stops REXX script for some time, wait-time is in hundreds of a second

WTO(console-message)

Write a message to the operator's console

SYSVAR(request-type)

a TSO-only function to retrieve certain TSO runtime information
request-type:

SYSENV	FORE/BACK	Foreground TSO / Batch TSO
SYSISPF	ACTIVE/NOT ACTIVE	
SYSREF	TSO Prefix (only available in Foreground TSO)	
SYSUID	Userid	

BREXX/370 V2R2M0 Software Notices

Brex has the capability code new functions or command in REXX. They are transparent and will be called in the same way as basic BREXX function.

Overview

BSTORAGE(...)

Storage command in the original BREXX decimal implementation

LISTALC()

Lists all allocated Datasets in this session or region

BRXMSG(...)

Standard message module to display a message in a formatted way, examples

```
rc=brxmsg( 10,'I','Program has been started')
rc=brxmsg(100,'E','Value is not Numeric')
rc=brxmsg(200,'W','Value missing, default used')
rc=brxmsg(999,'C','Division will fail as divisor is zero')
```

will return:

```
BRX0010I  PROGRAM HAS BEEN STARTED
BRX0100E  VALUE IS NOT NUMERIC
BRX0200W  VALUE MISSING, DEFAULT USED
BRX0999C  DIVISION WILL FAIL AS DIVISOR IS ZERO
```

DAYSBETW(date1,date-2,...)

Return days between 2 dates

JOBINFO(request-type)

Return information about currently running job or TSO session.

Request-type

JOBNAME	returns job name
JOBNUMBER	returns job number
STEPNAME	returns step name
PROGRAMNAME	returns running program name

LINKMVS(load-module, parms)

Starts a load module. Parameters (if any) will send in the format JCL would pass it.

MVSCBS()

Allows addressing of some MVS control blocks. This function must be imported, then the following functions can be used:

Cvt(), Tcb(), Ascb(), Tiot(), Jscb(), Rmct(), Asxb(), Acee(), Ecv(), Smca()

PDSDIR(dsn)

Return directory entries in a stem variable

RXDATE(...)

Return and optionally convert dates in certain formats

RXDSINFO(dsn/dd-name,options)

Return dsn or dd-name attributes

BREXX/370 V2R2M0 Software Notices

[RXDYNALC\(...\)](#)

Allows dynamic allocations of datasets or output

[RXSORT\(...\)](#)

Sorts a stemvariable with different sort algorithms

[SEC2TIME\(seconds\)](#)

Converts an amount of seconds into the format [days]hh:mm:ss

[SORTCOPY\(stem-variable\)](#)

Copies any stem variable into the stem SORTIN, which can be used by RXSORT

[STEMCOPY\(source-stem-variable,target-stem-variable\)](#)

Copies any stem variable into another stem variable

[TODAY\(\)](#)

Returns today's date in certain formats

For details of these functions refer to the BREXX/370 User's Guide.