BREXX/370 V2R2M1 Installation Guide

Document Version 4.0 created on 3. March 2020

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

Abstract

This document covers the installation process of the new of BREXX/370 V2R2M0.
BREXX/370 is provided as-is, please test carefully in test systems only!
BREXX/370 is not the same as IBM's REXX; there are many similarities, but also differences, especially when using MVS specific functions.
The next TK4- Update 9 release contains BREXX/370.

Prerequisites

MVS TK4-

This version of BREXX/370 has been developed and tested within Jürgen Winkelmann's TK4- (update 08). It may work in other versions of MVS but can't be guaranteed.

ISPF support (optional)

BREXX/370 also supports Wally Mclaughlin's version of ISPF and its contained SPF panels.

Recommendations

We strongly recommend testing BREXX/370 in an isolated test system to avoid any impact with your current system. To achieve this, you can easily copy the entire Hercules/MVS directory to another location and install BREXX/370 there.

Preparation of your target MVS38j System

Make sure that your MVS system has a BREXX Alias pointing to a user catalogue defined in the master catalogue. To determine it, run the command:

```
listcat entries('brexx') all
```

The result must look like this:

```
ALIAS --------- BREXX
      IN-CAT --- SYS1.VMASTCAT
      HISTORY
        RELEASE----------------2
      ASSOCIATIONS
        USERCAT--SYS1.UCAT.MVS
```

If BREXX is not defined, add a BREXX Alias accordingly.

If you omit this step, all BREXX data sets are catalogued in the Master Catalog. In this case, it may require the use of the Master Catalog password during the catalogue process. If you are running TK4- you do not see such requests as RAKF is providing the access authorisation of the Master Catalog, which therefore is not password protected. In the default TK4- configuration, only users HERC01 and HERC02 are authorised to update the master catalogue.

BREXX/370 V2R2M1 Installation Guide

Important Notice: The JCLs do not contain a NOTIFY parameter in the JOB card; it is advised to add one with your userid, e.g. **NOTIFY=HERC01**. The next BREXX V2R3M0 will contain a **NOTIFY=&SYSUID** parameter as TK4- then supports &SYSUID.

Installation

Step 0 - Unzip BREXX/370 Installation File

The ZIP installation file consists of several files:

README.pdf	important information to BREXX/370
BREXX370_Installation_Guide_V2R2M0.pdf	this installation document
BREXX370_Users_Guide_V2R2M0.pdf	BREXX/370 User's Guide
BREXX370_Formatted_Screens_Guide_V2R2M0.pdf	Formatted Screens User's Guide
BREXX370_VSAM_Guide_V2R2M0.pdf	VSAM User's Guide
BREXX370_Migration_Notice_V2R2M0.pdf	Migration Notice, new functionality and changes on existing functions
BREXX370_V2R2M0.XMIT	XMIT File containing BREXX modules and Installation JCL

Step 1 - Upload XMIT File

Use the appropriate upload facility of your terminal emulation.

The file created during upload must have **RECFM FB and LRECL 80**. If the DCB does not match, the subsequent unpacking process fails.

Step 2 - Unpack XMIT File

Unpack the XMIT file with an appropriate JCL. If you don't have one you can use the following sample, just cut and paste it in one of your TK4- jcl libraries:

```
//BRXXREC JOB 'XMIT RECEIVE',CLASS=A,MSGCLASS=H
//* ------------------------------------------------------------
//* RECEIVE XMIT FILE AND CREATE DSN OR PDS
//* ------------------------------------------------------------
//RECV370  EXEC PGM=RECV370,REGION=8192K
//RECVLOG  DD SYSOUT=*
//XMITIN   DD  DSN=HERC01.UPLOAD.XMIT,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DSN=&&XMIT2,
//      UNIT=3390,
//      SPACE=(TRK,(300,60)),
//      DISP=(NEW,DELETE,DELETE)
//SYSUT2   DD DSN=BREXX.V2R2M0.INSTALL,
//      UNIT=3390,
//      SPACE=(TRK,(300,60,20)),
//      DISP=(NEW,CATLG,CATLG)
//SYSIN    DD DUMMY
```

HERC01.UPLOAD.XMIT represents the uploaded XMIT File – please change it accordingly to the name you have chosen during the upload process.

`BREXX.V2R2M0.INSTALL`	is the name of the unpacked library (created during the UNPACK process). It is recommendable to remain with this DSN as it is used in later processes **Make sure there is no previous version of this PDS catalogued**; otherwise, it ends up as not catalogued: with ***NOT CATLG 2*!** Which means it was stored on a volume but could not be added to the catalogue, please remove the PDS manually.

Once the submitted Job has successfully unpacked the XMIT file into the target PDS, proceed with **STEP 3**. The created library contains all JCL to pursue with unpacking and installing.

Important Notice: If you use a different JCL to unpack the XMIT file, use UNIT=3390 in the JCL. The unit type 3390 was the only reliably UNIT that ran in all tested TK4- environments. Other units may sometimes lead to various errors during the unpacking process.

The next steps make usage of the unpacked library (in this example BREXX.V2R2M0.INSTALL)

Please run the JCL in the given order (refer to the **Step x** reference int the table). Submit **Step 3** as the first JCL of the installation sequence. Entries without a Step reference are used from the JCLs as input datasets.

Library content:

$CLEANUP	**4.**	Cleanup: Remove unnecessary installation files	**-> Step 6**
$INSTALL	**2.**	Install BREXX/370	**-> Step 4**
$README		Read me file	
$TESTRX	**3.**	Test job to verify the BREXX/370 installation	**-> Step 5**
$UNPACK	**1.**	Unpack subsequent libraries	**-> Step 3**
BUILD		Contains BREXX/370 Version and date and XMIT date	
CMDLIB		xmit packed command proc	
SAMPLES		xmit packed BREXX commands	
JCL		xmit packed example JCL	
LINKLIB		xmit packed BREXX Load library	
PROCLIB		xmit packed BREXX JCL procedures	
RXINSTDL		Internal CLIST used during Installation	
RXLIB		xmit packed include library	

Step 3 - Submit $UNPACK JCL of the unpacked Library

In the unpacking process, the contained installation files will be expanded into different partitioned datasets.

BREXX/370 V2R2M1 Installation Guide

Before submitting the $UNPACK JCL, the XMITLIB parameter must match the dataset name used in the expand JCL of Step2.

If you followed the recommendation to name it:

BREXX.V2R2M0.INSTALL

no change is required.

```
//BRXXUNP JOB 'XMIT UNPACK',CLASS=A,MSGCLASS=H
//* ------------------------------------------------------------------
//* UNPACK XMIT FILES INTO INSTALL LIBRARIES
//*   *** CHANGE XMITLIB= TO THE EXPANDED XMIT LIBRARY OF INSTALLATION
//* ------------------------------------------------------------------
//*           ---->    CHANGE XMITLIB TO THE UNPACKED XMIT FILE  <----
//*                          XXXXXXXXXXXXX
//*                         X      X      X
//*                        X       X       X
//*                       X        X        X
//*                      X         X         X
//XMITLOAD PROC XMITLIB='BREXX.V2R2M0.INSTALL',
//         HLQ='BREXX.V2R2M0',     <-- DO NOT CHANGE ----
//         MEMBER=
```

If the job does not run and waits check in option 3.8 the status, it is most likely ”WAITING FOR DATASETS”. Simplest method to resolve this is to LOGOFF and re-LOGON to your TSO session.

After completion of the $UNPACK JCL the following new Libraries are available:

BREXX.V2R2M0.CMDLIB	REXX commands directly executable
BREXX. V2R2M0.SAMPLE	REXX Samples scripts
BREXX. V2R2M0.JCL	REXX Job Control
BREXX. V2R2M0.LINKLIB	BREXX Load Module
BREXX. V2R2M0.PROCLIB	BREXX JCL Procedures
BREXX. V2R2M0.RXLIB	BREXX include Libraries

The unpacking process removes any old version of the above libraries, before the creation of the new version. If no old version of these libraries is available, the delete steps end with **RC=4**, as well as the Job ends with **RC=4**. Ignore these errors, if the individual unpack steps return with RC=0. Therefore please carefully check the output of this Job.

Step 4 - Submit $INSTALL JCL of the unpacked Library

The $INSTALL JCL copies the BREXX Members into the appropriate SYS2. Libraries

BREXX.LINKLIB	->	SYS2.LINKLIB
BREXX.PROCLIB	->	SYS2.PROCLIB

All these members are BREXX/370 specific and do not conflict with existing members. Members of the system libraries remain untouched.

Please log off and re-login to your TSO session before performing any online testing; this enforces the new loading of modules used during the testing.
In rare situations, the installation of the BREXX Linklib members may create a new dataset extent in SYS2.LINKLIB. In this case, you must also restart your TK4- MVS session.

Step 5 - Submit $TESTRX JCL of the unpacked Library

$TESTRX start a test to verify the installation of BREXX/370. All steps should with RC=0

Step 6 - Submit $CLEANUP JCL of the unpacked Library

The $CLEANUP job removes all unnecessary installation files they are no longer needed, as they were merged into the appropriate SYS2.xxx library

BREXX.V2R2M0.LINKLIB
BREXX.V2R2M0.PROCLIB

You may also wish to remove the uploaded XMIT File, which was used for the first unpack process.

Step 7 - Your Tests

Now it's your turn to test BREXX/370! Please be advised BREXX/370 is not z/OS REXX, so you might miss some functions but find also functions not available in the "original".

Step 8 - ADD BREXX Libraries into TSO Logon

To run BREXX with its shortcut RX, REXX, BREXX you must allocate the BREXX libraries into your Logon procedure. There are several ways to achieve this. The easiest and recommended way is to add lines into in SYS1.CMDPROC**(USRLOGON)**

Locate the line **%STDLOGON**, add the following command sequence before it:

```
/* ALLOCATE SYSEXEC TO SYS2 EXEC   */
IF &SYSDSN('SYS2.EXEC') EQ &STR(OK) THEN DO
   FREE FILE(SYSEXEC)
   ALLOC FILE(SYSEXEC) +
      DSN('SYS2.EXEC') SHR
END

/* ALLOCATE SYSUEXEC TO USER EXECS */
IF &SYSDSN('&SYSUID..EXEC') EQ &STR(OK) THEN DO
   FREE FILE(SYSUEXEC)
   ALLOC FILE(SYSUEXEC) +
      DSN('&SYSUID..EXEC') SHR
END

/* ALLOCATE RXLIB IF PRESENT */
IF &SYSDSN('BREXX.V2R2M0.RXLIB') EQ &STR(OK) THEN DO
   FREE FILE(RXLIB)
   ALLOC FILE(RXLIB) +
      DSN('BREXX.V2R2M0.RXLIB') SHR
END
%STDLOGON     <<<  insert  above coding before this CLIST call
```

The update of the TSO Logon CLIST is a completely manual process! Please take a backup of USRLOGON CLIST first to allow a recovery in case of errors!

Users who upgrade from a previous release of BREXX need to update the logon clist and replace the RXLIB allocation with the current dataset name: **BREXX.V2R2M0.RXLIB**.

Step 9 - Removing old BREXX Libraries (optional)

If you had a previous BREXX/370 version installed and your tests ran successfully, you can remove the libraries of the earlier BREXX version, for example, V2R2M0.

BREXX.V2R2M0.CMDLIB	REXX commands
BREXX. V2R2M0.SAMPLE	REXX Samples scripts
BREXX. V2R2M0.JCL	REXX Job Control
BREXX. V2R2M0.LINKLIB	BREXX Load Modules
BREXX. V2R2M0.PROCLIB	BREXX JCL Procedures
BREXX. V2R2M0.RXLIB	BREXX include Libraries

If you upgrade from the very first BREXX/370 version, you can remove the following libraries:

BREXX.CMDLIB	REXX commands
BREXX.SAMPLE	REXX Samples scripts
BREXX.JCL	REXX Job Control
BREXX.LINKLIB	BREXX Load Modules
BREXX.PROCLIB	BREXX JCL Procedures
BREXX.RXLIB	BREXX include Libraries

Useful functions

There are JCL Procedures delivered, which facilitate the test and execution of REXX scripts.
The installation process merges them into SYS2.PROCLIB.
The delivered RXLIB PDS contains several REXX functions, which are usable as if they were a BREXX internal function. The delivered JCL procedures allocate the RXLIB library, and it is recommended to add it also into the TSO Logon procedures (Step 8).

TSO online

RX rexx-script-name alternatively **REXX rexx-script-name**

Can be used to start REXX script from TSO (option 6). BREXX performs all necessary allocations. It is advised to add a user-specific REXX library, naming convention: **&SYSUID.EXEC** (RECFM=VB, LRECL255). If available, the REXX-script searches path starts from there.

The REXX library search sequence is:

1.	SYSUEXEC	typically &SYSUID.EXEC
2.	SYSUPROC	(optional)
3.	SYSEXEC	(optional)
4.	SYSPROC.	(optional)

At least one of these libraries needs to be pre-allocated during the TSO logon process. It is not mandatory to have all of them allocated. It depends on your planned REXX development environment. The allocations may consist of concatenated datasets.

RX 'dataset-name(rexx-script-name)' or REXX 'dataset(rexx-script-name)'

Alternatively, you can specify a fully qualified dataset-name

TSO Batch (start REXX JCL Procedure)

There is a JCL Procedure defined which allows you to run REXX Scripts in a TSO Batch environment.
The Procedure performs al necessary BREXX and TSO allocations.
Some ADDRESS TSO commands as ALLOC/FREE are supported

```
//DATETEST JOB CLASS=A,MSGCLASS=H,REGION=8192K,NOTIFY=&SYSUID
//*
//* ------------------------------------------------------------------*
//* TEST REXX DATE AS TSO BATCH
//* ------------------------------------------------------------------*
//REXX EXEC RXTSO,EXEC='DATE#T',SLIB='BREXX.SAMPLES'
```

EXEC defines the rexx to run
SLIB defines the library containing the rrexx
Additionally you can add a ,P='input-parameters' JCL Parameter field, if you rexx receives input parameters.

BREXX/370 V2R2M1 Installation Guide

Plain Batch (start REXX JCL Procedure)

There is a JCL Procedure defined which allows you to run REXX Scripts in a plain Batch environment.
The Procedure performs all necessary BREXX allocations
ADDRESS TSO commands are not supported here!

```
//DATETEST JOB CLASS=A,MSGCLASS=H,REGION=8192K,NOTIFY=&SYSUID
//*
//* -------------------------------------------------------------------*
//* TEST REXX DATE AS TSO BATCH
//* -------------------------------------------------------------------*
//REXX EXEC RXBATCH,EXEC='ETIME#T',SLIB='BREXX.SAMPLES'
```

EXEC defines the rexx to run
SLIB defines the library containing the rexx
Additionally you can add a ‚P='input-parameters' JCL Parameter field, if your rexx receives input parameters.

BREXX/370 Sample Library

The Library BREXX.SAMPLES contains a variety of REXX scripts which cover the following areas:

- Basic functionality in Members starting with '$'
- FSS samples, starting with '#'
- VSAM samples starting with '@'
- All other scripts are original samples delivered with Vasilis Vlachoudis BREXX installation.

BREXX/370 Hints

- Please make sure that your REXX files do not contain line numbering! They are not wiped out by BREXX/370 and therefore treated as the content of the script. This lead to errors during interpretation, sometimes even system abends! Use **UNNUM** as a primary command in the RFE editor to switch line numbering off and remove existing numbers.
- If the BREXX/370 call leads to an S106 Abend, the most likely reason is the creation of a new extent in SYS2.LINKLIB during the installation process. Its size and number of extents are loaded during IPL and kept while MVS is up and running. The creation of new extents will therefore not be discovered.
    - You can either re-IPL your system or better
    - REORG SYS2.LINKLIB with IEBCOPY

BREXX Documentation

You find the BREXX documentation at:

https://ftp.gwdg.de/pub/languages/rexx/brexx/html/rx.html

BREXX/370 V2R2M1 Installation Guide

Credits

- BREXX has been developed by Vasilis Vlachoudis, who made it publicly available as freeware for non-commercial purposes.
- Jason Winter's JCC Compiler compiled BREXX
- JCC and the JCC-Library are owned and maintained by him. While not being freeware, Jason allows non-commercial usage and distribution of Software created using JCC through a relaxed license, as long as the complete source code always accompanies those distributions.
- Vasilis and Jason explicitly consented to make the JCC based version of BREXX available on TK4-. Thanks to both for their valuable contribution to the TK4- MVS 3.8j Tur(n)key system.
- The VSAM Interface is based on Steve Scott's VSAM API.
- The FSS Part is based on Tommy sprinkle's FSS - TSO Full-Screen Services
- Daniel Gaeta contributed his EXECIO implementation.
- We wish to thank the following persons for patiently answering our questions and their support and advice:
  Vasilis Vlachoudis, Jürgen Winkelmann, Jason Winter, Wally Mclaughlin and Greg Price, Steve Scott and many others!

BREXX/370 Source Code

The BREXX/370 Source Code can be found and downloaded at:

https://github.com/mgrossmann/BREXX/370