BREXX/370 V2R5M3 Installation Guide

Document Version 5.3 updated on 19. January 2024

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

Introduction

This document covers the installation process of the new BREXX/370 V2R5M3.
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 and differences, especially when using MVS-specific functions.
Rob Prins' TK5 Update 3 includes the installed version of BREXX/370 V2R5M3.

Prerequisites

MVS TK4-

This version of BREXX/370 was created and tested within Jürgen Winkelmann's TK4 (update 8).

MVS TK5 (update 3)

This version of BREXX/370 was tested on Rob Prins' TK5. It will be included in TK5 (update 03). Therefore, you can skip the installation process unless you update your BREXX installation with a fix release.

Other MVS distributions

It should work with other MVS distributions, but this cannot be verified.
Users should be aware of the following differences:

XMIT RECEIVE STEPLIB DD Statement

It might be necessary to add a STEPLIB DD statement to locate the library containing the RECV370:

```
//RECV370  EXEC PGM=RECV370
//STEPLIB  DD   DSN=library
```

Please add it to the Jobs where needed.

In TK4- and TK5 RECV370 is contained in a system library; therefore, a STEPLIB DD statement is not needed!

REGION SIZE

It may be necessary to lower the REGION size parameter to 4 MB or 6 MB, as MVS may reject the REGION=8192K argument with the warning "REGION UNAVAILABLE, ERROR CODE=20".
TK4- and TK5 support 8MB region sizes.

```
//stepname EXEC PGM=xxxxx,REGION=6144K
```

TSO fullscreen support

BREXX/370 supports and operates effectively in the following TSO full-screen environments:

- Wally McLaughlin's version of ISPF
- Greg Price's REVIEW and RFE (Review Front End)
- Rob Prins' RPF environment

Recommendations

We recommend testing BREXX/370 in an isolated test system to avoid any impact on 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

BREXX Catalogue

In TK5, the required Catalogue entry is already defined in the distribution and can therefore be omitted.

Ensure 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 the BREXX Alias is not defined, add it:

```
//ADDBREXX   EXEC PGM=IDCAMS
//SYSPRINT     DD    SYSOUT=*
//SYSIN        DD    *
  DEFINE ALIAS (NAME(BREXX) RELATE(your-user-catalog))
```

If the submitted job is not running, it might be necessary to enter the password of the master catalogue in the MVS console (in TK4- not needed)

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 is not password protected. In the default TK4- configuration, only users HERC01 and HERC02 are authorised to update the master catalogue.

Important Notice: All JCLs in the installation and sample library contain now a **NOTIFY=&SYSUID** parameter in the JOB card. If the patch, to resolve it during the Submit process by the current user-id, is not applied, you need to change &SYSUID to your user-id, or remove it from the JOB card!
The patch can be found at: http://prycroft6.com.au/vs2mods/index.html#zp60034
This fix is already applied in TK5!

BREXX/370 V2R5M3 Installation Guide

BREXX Catalogue

Make sure that dataset **BREXX.V2R5M3.INSTALL** is not already catalogued from a previous run. It is the recommended dataset name and will be created during the receiving process of RECV370.

If a previous version of this dataset name is still catalogued, the new version ends up as not catalogued: with a ***NOT CATLG 2*** message*!* The Job output is not revealed by a CCOD**.** Any later job which is accessing BREXX.V2R5M3.INSTALL will use the old version of the dataset.

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_V2R5M3.pdf	this installation document
• BREXX370_Users_Guide_V2R5M3.pdf	BREXX/370 User's Guide
• BREXX370_Formatted_Screens_Guide_V2R5M3.pdf	Formatted Screens User's Guide
• BREXX370_VSAM_Guide_V2R5M3.pdf	VSAM User's Guide
• BREXX370_Migration_Notice_V2R5M3.pdf	Migration Notice, new functionality and changes to existing functions
• BREXX370_V2R5M3.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 into one of your 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.V2R5M3.INSTALL,
//    UNIT=3390,
//    SPACE=(TRK,(300,60,20)),
```


created 8.8.19, updated 17.02.24 09:54
Revision 230

```
//     DISP=(NEW,CATLG,CATLG)
//SYSIN     DD DUMMY
```

HERC01.UPLOAD.XMIT	represents the uploaded XMIT File – please change it to the name you chose during the upload process.
BREXX.V2R5M3.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**;

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. Unit type 3390 was the only reliable UNIT that ran in all tested TK4- environments. Other units may sometimes lead to various errors during the unpacking process.

The next steps make use of the unpacked library (in this example BREXX.V2R5M3.INSTALL)

Please run the JCL in the given order (refer to the **Step x** reference in 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		Cleanup: Remove unnecessary installation files	**-> Step 7**
$INSTALL		Install BREXX/370	**-> Step 4**
$CREKEYV	**new**	**Create the Key/Value Database (optional)**	**-> Step 6**
$README		Read me file	
$TESTRX		Test job to verify the BREXX/370 installation	**-> Step 5**
$UNPACK		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	

Activating the new BREXX V2R5M3 Release

The next steps describe how to enable your new BREXX Release. In summary, you must run the following jobs out of the above library in the listed sequence:

$UNPACK	mandatory
$INSTALL	mandatory
$TESTRX	optional, recommended
$CREKEYV	optional recommended
$CLEANUP	optional

See details in the step descriptions

Step 3 - Submit $UNPACK JCL of the unpacked Library

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

! Before submitting the $UNPACK JCL, the XMITLIB parameter must match the dataset name used in the expanded JCL of Step 2.
If you followed the dataset naming recommendations it is:
BREXX.V2R5M3.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.V2R5M3.INSTALL',
//         HLQ='BREXX.V2R5M3',     <-- DO NOT CHANGE ----
//         MEMBER=
```

! If the job does not run and waits, check with option 3.8 the status. It is most likely ”WAITING FOR DATASETS”. The simplest method to resolve this is to LOGOFF and re-LOGON on your TSO session.

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

BREXX.V2R5M3.CMDLIB	REXX commands are directly executable
BREXX. V2R5M3.SAMPLE	REXX Samples scripts
BREXX. V2R5M3.JCL	REXX Job Control
BREXX. V2R5M3.LINKLIB	BREXX Load Modules
BREXX. V2R5M3.APFLLIB	BREXX authorised Load Modules
BREXX. V2R5M3.PROCLIB	BREXX JCL Procedures
BREXX. V2R5M3.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 RC=0. Therefore please carefully check the output of this job.

! **Before you install BREXX, you must decide either on the "normal" BREXX installation or the authorised BREXX installation.**

With the authorised version, you can call from BREXX utilities such as IEBGENER, IEBCOPY, NJE38, etc. which run in authorised mode. This requires that the environment in which you start BREXX is

authorised, meaning Wally Mclaughlin's ISPF, or RFE must be authorised. Plain TSO is already authorised.
Both installations are copied into the same partitioned datasets; they are, therefore, mutually exclusive!

If the standard installation is sufficient, continue with S**tep 4**

If you plan to use the authorised, continue with **Step 4A. In this case,** the MVS authorisation table needs to be updated as well.

Step 4 - Submit $INSTALL JCL for the Standard Installation

The $INSTALL JCL copies the BREXX Members into the appropriate SYS2. Libraries It also removes BREXXSTD from SYS2.LINKLIB as it will be no longer delivered.

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, or else you might see an 0C4.
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.

Continue with STEP 5

Step 4A- Submit $INSTAPF JCL for the Authorised Installation

When you use Rob's TK5, the programs RX, REXX, and BREXX are already authorised by the user mods USERMOD.ZUM0001 and USERMOD.ZUM0014. There is no need for any more activities. You can skip this step and go to STEP 5.

The $INSTPAPF JCL copies the authorised BREXX Members into the appropriate SYS2. Libraries. It also removes BREXXSTD from SYS2.LINKLIB as it will be no longer delivered.

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.

To authorise the Modules to change the following Modules

- SYS1.UMODSRC(IKJEFTE2)
- SYS1.UMODSRC(IKJEFTE8)

Add the BREXX modules to the sources:

```
         DC     CL8'BREXX    '
```

```
         DC     CL8'REXX     '
         DC     CL8'RX       '
```

To activate the changes submit the Jobs:

- SYS1.UMODCNTL(ZUM0001)
- SYS1.UMODCNTL(ZUM0014)

Restart your MVS

- Shut down your MVS
- Re-IPL your job with the CLPA option
- Shut Down MVS again
- Perform normal IPL

! If you run Wally McLaughlin's ISPF the ISPF libraries must be authorised, otherwise calling a rexx from within ISPF will abend (usually S306).

Step 5 - Submit $TESTRX JCL of the unpacked Library

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

Step 6 - Create the Key/Value Database (optional)

If you want to use Key/Value function in BREXX, modify and submit $CREKEYV. This JCL contains two **CREATE CLUSTER** definitions for the required VSAM datasets. As the K/V Database may contain data from many applications, it is generously sized, adjust it to a size which suits you. It is recommended to allocate it on a separate VOLUME, but not mandatory, insert the one chosen one.

Once it is created you can use it with the Key/Value functions.

! Do not run the JCL if you have already a K/V Database in place, else it will delete all your existing data.

Key/Value Database

```
DEFINE CLUSTER                          -
         (NAME(BREXX.KEYVALUE)          -
          INDEXED                       -
          KEYS(44 0)                    -
          RECORDSIZE(64 8192)           -
          SHAREOPTIONS(2,3)             -
          CYLINDERS(600 50)             -
          VOLUMES(XXXXXX)               -
          UNIQUE                        -
          SPEED)                        -
       DATA                             -
          (NAME(BREXX.KEYVALUE.DATA))   -
       INDEX                            -
          (NAME(BREXX.KEYVALUE.INDEX))
```

Reference Database

```
DEFINE CLUSTER                          -
         (NAME(BREXX.KEYREFS)           -
          INDEXED                       -
```

```
           KEYS(105 0)                      -
           RECORDSIZE(128 512)              -
           SHAREOPTIONS(2,3)                -
           CYLINDERS(250 50)                -
           VOLUMES(XXXXXX)                  -
           UNIQUE                           -
           SPEED)                           -
        DATA                                -
           (NAME(BREXX.KEYREFS.DATA))       -
        INDEX                               -
           (NAME(BREXX.KEYREFS.INDEX))
```

Step 7 - 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.V2R5M3.LINKLIB
BREXX.V2R5M3.PROCLIB

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

Step 8 - ADD BREXX Libraries into the TSO Logon

To run BREXX with its shortcut RX, REXX, and BREXX you must allocate the BREXX libraries into your Logon procedure. There are several ways to achieve this. The easiest and recommended method for **TK4-** users is to add lines into SYS1.CMDPROC**(USRLOGON),**
TK5 users add them into SYS1.CMDPROC**(ISPLOGON),**
other installations may use different libraries.

Locate the line **%STDLOGON**, and 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.V2R5M3.RXLIB') EQ &STR(OK) THEN DO
   FREE FILE(RXLIB)
   ALLOC FILE(RXLIB) +
      DSN('BREXX.V2R5M3.RXLIB') SHR
END
%STDLOGON     <<<  insert  above coding before this CLIST call
```

Using the CLISTs as plain commands, you can either copy them into the user clist or allocate BREXX.V2R5M3.CMDLIB in the appropriate TSO start clist. This may be accomplished in both TK4 and TK5 by including the following part in SYS1.CMDPROC(USRLOGON)::

```
FREE FILE(SYSPROC)
ALLOC FILE(SYSPROC) +
  DSN('&SYSUID..CMDPROC','SYS1.CMDPROC','SYS2.CMDPROC', -
      'SYS2.REVIEW.CLIB','BREXX.V2R5M3.CMDLIB') SHR
```

The update of the TSO Logon CLIST is an entirely 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.V2R5M3.RXLIB**.

Step 9 - Your Tests

! It is advised to **LOGOFF** and **LOGON** again to your TSO system to make sure that the newly installed modules become active. If you do not restart, you might receive errors.

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 10- Remove 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, V2R5M0. Make sure you save your private changes.

BREXX.V2R5M0.INSTALL	Installation library
BREXX.V2R5M0.CMDLIB	REXX commands
BREXX. V2R5M0.SAMPLES	REXX Samples scripts
BREXX. V2R5M0.JCL	REXX Job Control
BREXX. V2R5M0.LINKLIB	BREXX Load Modules
BREXX. V2R5M0.APFLLIB	BREXX Load Modules (authorized)
BREXX. V2R5M0.PROCLIB	BREXX JCL Procedures
BREXX. V2R5M0.RXLIB	BREXX include Libraries

You can also remove all older installation libraries, after saving your private changes:

BREXX.V2RrMm.CMDLIB	REXX commands
BREXX. V2RrMm.SAMPLE	REXX Samples scripts
BREXX. V2RrMm.JCL	REXX Job Control
BREXX. V2RrMm.LINKLIB	BREXX Load Modules
BREXX. V2RrMm.PROCLIB	BREXX JCL Procedures
BREXX. V2RrMm.RXLIB	BREXX include Libraries

V2RrMm is the notation of a previous release, e.g. V2R5M1, V2R5M0, V2R4M0, etc.

Additional Settings (optional)

If you want to communicate with the control program of the host system (either Hercules or VM) you can do so, by running an:

ADDRESS COMMAND 'CP cp-parameter …'

For VM you need to use a valid CP command.

Example: **ADDRESS COMMAND 'CP QUERY TIME'**

If your system is running within Hercules your CP commands are routed to Hercules and need to be Hercules commands.

Example: **ADDRESS COMMAND 'CP DEVLIST'**

To communicate with Hercules you need to enable the DIAG8 commands

DIAG8CMD ENABLE in the Hercules console.

In TK4 and TK5 systems it is already enabled. If it is not enabled and you run an

ADDRESS COMMAND "CP command"

BREXX will abend typically with an 0C6.

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 the 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 that allows you to run REXX Scripts in a TSO Batch environment.
The Procedure performs all 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 rexx script
Additionally, you can add a P='input-parameters' JCL Parameter field, if you rexx receives input parameters.

TSO CLISTs

To use the Clists of BREXX.V2R5M3.CMDLIB without an EXEC command, the library must be allocated to TSO, alternatively, you can copy the members to an allocated library (e.g. SYS2.CMDPROC)-

Plain Batch (start REXX JCL Procedure)

There is a JCL Procedure defined that 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 that cover the following areas:

- Basic functionality in Members starting with '$'
- FSS samples, starting with '#'

- VSAM samples beginning with '@'
- All other scripts either breach the above rules or are original samples from Vasilis Vlachoudis' BREXX installation.

BREXX/370 Hints

- Make sure your REXX files do not have line numbers! They are not wiped away by BREXX/370 and are thus considered script content. This causes mistakes during interpretation, and occasionally even system abends! To disable line numbering and delete existing numbers, use UNNUM as a primary command in the RFE or RPF Editor.
- 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

All BREXX/370 documentation is contained in the installation zip file. The original BREXX documentation can be found at:

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

Credits

- BREXX has been developed by Vasilis Vlachoudis, who made it publicly available as freeware for non-commercial purposes.
- Jason Winter's JCC Compiler to compile 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 significant 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 the EXECIO code
- The NJE38DIR load module was extracted out of Bob Polmanter's NJE38 V2 modules
- The VTOC module was contributed by David Cartwright (CBT File 587)

- We wish to thank the following persons for patiently answering our questions and for their support and advice:
  Vasilis Vlachoudis, Jürgen Winkelmann, Jason Winter, Wally Mclaughlin and Greg Price, Rob Prins, Bob Polmanter, Steve Scott, and many others!

BREXX/370 Source Code

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

https://github.com/mvslovers/brexx370/