

BREXX/370 V2R4M0 Software and Migration Notices

Document Version 1.0 created on 25. February 2021

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

Content

This document covers the amendments and changes in the new version of BREXX/370 V2R4M0. For changed functionality, it contains the migration instruction to upgrade from the previous release(s). The installation process is separately described in the **BREXX/370 Installation** document. New features or functions are described in details in:

BREXX370_Users_Guide_V2R4M0.pdf or a separately mentioned document.

All documents are contained in the installation zip file

Upgrade from a previous BREXX/370 Version

Before upgrading to BREXX/370 V2R4M0 backup your system. The easiest way is creating a copy of your TK4- directory containing all of your settings and the DASD volumes. In the case of errors or unwanted behaviour, you can quickly recover to the backup version.

I. Important Changes

If you migrate from an earlier version of BREXX/370 then additionally take a look into the Migration Notices of V2R1M0 and V2R2M0, which are contained in the installation zip file

1. Software Changes requiring actions

1.1. Authorised BREXX Version available

With this release, we ship a standard installation of BREXX as well as an authorised version, which allows to system programs as IEBCOPY, NJE38, etc. The decision about what to install must be made before installation.

1.2. BREXXSTD load module removed

We have straightened the load module structure and removed the BREXXSTD load module from the installation library. If you use JCL with an explicit BREXXSTD call, replace it by BREXX. During the installation process, any existing BREXXSTD module will be removed from SYS2.LINKLIB.

1.3. Call PLI Functions

Example compile jobs for callable PLI Functions can be found in **BREXX. V2R4M0.JCL**:

RXPI calculate PI with 500 digits

BREXX/370 V2R4M0 Software and Migration Notices

RXCUT Return every n.th character of a string

2. Known Problems

2.1. Reading Lines from a sequential dataset

Reading lines of sequential datasets always truncate trailing spaces; this may be an unwanted behaviour for fixed-length datasets. To circumvent this problem, you can use the following method:

If the dataset is allocated via a DD statement:

```
X=LISTDSI('INFILE FILE')
fhandle=OPEN(infile, 'RB')
Record=READ(fhandle, SYSLRECL)
```

If the dataset is used directly:

```
dsn='HERC01.TEMP'
X=LISTDSI('' dsn'')
fhandle=OPEN('' dsn'', 'RB')
Record=READ(fhandle, SYSLRECL)
```

LISTDSI returns the necessary DCB information (SYSLRECL). The OPEN must be performed with OPTION 'RB' which means READ, BINARY. Read uses the record length to create the record.

2.2. BREXX FORMAT Function

The BREXX FORMAT function differs from the standard behaviour of REXX FORMAT:

*FORMAT rounds and formats number with **before** integer digits and **after** decimal places. **expp** accepts the values 1 or 2 (**WARNING different from the ANSI-REXX specification**) where 1 means to use the "G" (General) format of C, and 2 the "E" exponential format of C. Where before+after+1 replaces the place of the total width specifier in C. (**expt** is ignored!)*

After determining the code, we discovered that a complete re-write would be necessary. As the effort does not stand in proportion to the benefit, we decided to leave it as it is.

II. New and amended functionality

1. Functions with changed functionality

There is a major change in every time functions. We have increased the precision of the time format from hundreds of a second to milliseconds in some cases to microseconds. If you use them or rely on the format, please change your REXX scripts accordingly.

BREXX/370 V2R4M0 Software and Migration Notices

| | |
|----------------|-----------------------|
| say TIME('L') | 16:38:03.112765 |
| call wait 100 | now waits 0.1 seconds |
| call wait 5000 | waits 5 seconds |

2. BREXX functions and Commands

This sections contains all new or changed BREXX V2R4M0 functions

DATE(target-date-format,date,input-date-format)

The new date function has now the “used” formats provided by the original REXX. Details in the BREXX User’s Guide

DATETIME(target-format,timestamp,input-format)

Formats are:

| | | |
|---|---------------------------|--------------------------|
| T | is timestamp in seconds | 1615310123 |
| E | timestamp European format | 09/12/2020-11:41:13 |
| U | timestamp US format | 12.09.2020-11:41:13 |
| O | Ordered Time stamp | 2020/12/09-11:41:13 |
| B | Base Time stamp | Wed Dec 09 07:40:45 2020 |

Time('MS'/'US'/'CPU')

Time has gotten new input parameters:

| | |
|-----|---------------------------------------|
| MS | Time of today in seconds.milliseconds |
| US | Time of today in seconds.microseconds |
| CPU | used CPU time in seconds.milliseconds |

LINKMVS(load-module, parms)

LINKPGM(load-module, parms)

Start a load module. Parameters work according to standard conventions.

LOCK('lock-string',<TEST/SHARED/EXCLUSIVE><,timeout>)

UNLOCK('lock-string')

Locks/unlocks a resource to avoid concurrent access to it

TIMESTAMP([day,month,year])

TIMESTAMP returns the unix (epoch) time, seconds since 1. January 1970.

BREXX/370 V2R4M0 Software and Migration Notices

3. Support of Formatted Screens

There is a separate BREXX Formatted Screen document:

BREXX370_Formatted_Screens_V2R4M0.pdf

4. Integration of VSAM I/O

There is a separate BREXX VSAMIO document:

BREXX370_VSAM_V2R4M0.pdf

for details on functions and usage, please take a look into it.