

# AcceleREXX

*The Interpretive REXX Compiler for z/OS and TSO*



- **Faster REXX execs**
- **Protected from unauthorized modification**
- **Without run-time libraries**
- **Supports the complete z/OS REXX language and its extensions**
- **Improves developer productivity**
- **Costs *much* less than IBM's REXX/370™ compiler**

## Overview

AcceleREXX protects your REXX applications, speeds their execution and makes REXX the ideal Rapid Application Development tool:

- REXX offers immediate execution and powerful debugging facilities.
- The companion *REXX Language Xtensions* endow REXX with native SQL support, VSAM access and an expanding Software Development Kit.
- AcceleREXX compiles your REXX applications into secure, high performance load modules.

Enhanced REXX gains enough power to replace COBOL, PL/1, C — even Assembler language!

- Develop complex, 'industrial strength' applications and deploy them as reentrant, shareable load modules
- Compile frequently used execs into LPALIB to further improve I/O and paging performance

- Link compiled execs into REXX Function Packages for the fastest possible access to subroutines and functions
- Write system exit routines for MVS, JES, RACF and vendor supplied software

## Benefits

- Reduce CPU usage and improve performance
- Eliminate the undisciplined modifications that often 'sneak' into production
- Protect sensitive information (the encryption option)
- Conserve disk space (the compression option)
- Streamline application development

## The AcceleREXX Compile Dialog

The AcceleREXX Compiler can be invoked via JCL or through the ISPF dialog panels illustrated in Figure 1 and Figure 2. The REXX execs you select for processing can be compiled into discrete object modules or merged into composite load modules.

```
----- AcceleREXX Compiler -----
Command ==>
RCX020 - Select 1 or more REXX execs for compilation
ISPF library:
Project ==> RLX
Group  ==> SAMPLE      ==>      ==>      ==>
Type   ==> EXEC
Member ==> RNV*          (Blank or pattern for member selection list)

Other partitioned or sequential data set:
Data set name ==>
```

Figure 1: AcceleREXX Compile Panel 1

```
REXX Compile: RLX.SAMPLE.EXEC ----- ROW 1 OF 6
Command ==>                               Scroll ==> HALF

Name      Action  Lib VV.MM  Created      Changed      Size  Init  Mod  ID
s RNVF    1 01 79 02/08/31 03/09/22 15:55 217 135 0 RNX4
s RNVFHELP 1 01 03 02/08/31 03/09/22 15:39 32 35 0 RNX4
s RNVFHLP1 1 01 10 02/09/01 03/09/22 15:42 33 31 0 RNX4
RNVFPAN 1 01 17 02/08/31 03/09/22 15:40 33 35 0 RNX4
RNVFPANS 1 01 08 02/09/01 03/09/22 15:41 33 31 0 RNX4
RNVL 1 01 01 02/08/31 03/09/22 15:43 100 100 0 RNX4
```

Figure 2: AcceleREXX Member Selection List

**For more information**  
**contact the REXX experts at**  
**800 776-0771**  
**[www.relarc.com](http://www.relarc.com)**

## Features

- Load modules run independently of the compiler in both TSO/E and non-TSO address spaces
- REXX execs, TSO CLISTs and compiled applications can invoke one another without restriction
- Supports all documented REXX external interfaces
- Fully supports the RLX implementations of dynamic and static SQL
- Checks REXX syntax and produces symbol cross references
- Removes embedded comments (or retains them for run-time diagnostics)
- Promotes incremental development. Separately compiled execs can be linked into several load modules
- Builds appropriate parameter lists for programs, command processors, REXX external functions, ISPF dialogs, ISPF/Edit macros and system exits — dynamically and automatically

## Prerequisites

AcceleREXX requires the following software or subsequent releases:

- Any release of z/OS or OS/390 whose IBM support is current
- Dynamic SQL support requires RLX/SQL or RLX/TSO
- Static SQL support requires RLX/Compile

Product names are the trademarks or registered trademarks of their respective holders.