USING REXX IN A WINDOWS ENVIRONMENT

This course provides instruction on the use of the Restructured EXtended eXecutor (REXX) language.

Overall Objective

Attendees are PC users who wish to create and use programs written in REXX. A large number of terminal exercises are used to reinforce and emphasise the lecture material. The exercises are organised to allow delegates to progress at their own pace. The course assumes no previous knowledge of REXX.

Audience

The course is designed to run on a PC running the Windows operating system, but can be applied to REXX running on Linux as well. This course is suitable for:

- Users
- Analysts
- System administrators
- Technical support personnel

who have a need to write, understand or maintain REXX programs.

If REXX is to be used on the IBM mainframe using z/OS, OS/390 and TSO environments then attendance of the **Using REXX under TSO** course will be more appropriate.

Prerequisites

Delegates should meet these minimum requirements:

- Some previous programming experience or programming training (in any language) would be useful
- For delegates in the PC (or Linux) environment, familiarity with PC (or Linux) concepts is required
- The ability to create and modify files using Windows *Notepad* or Linux *vim* editor is necessary

Duration

The course is run over 3 to 4 days and consists of classroom lectures and hands-on exercises.

Course objectives

On completion of this course, the delegate will be able to:

- Code REXX instructions with proper syntax
- Construct program logic using REXX control flow instructions
- Invoke and pass parameters to REXX programs
- Code REXX instructions to accomplish file and terminal I/O
- Describe the purpose and use of the data stack
- Write REXX programs that use the data stack
- Use REXX built-in functions to manipulate data
- Create user defined functions and subroutines
- Code REXX instructions that parse complex data strings
- Debug REXX programs using the REXX TRACE facility
- Identify the use of REXX command Environment

Course contents

- Historical background to the REXX language
- Language components: Clauses, Tokens, Labels, Assignments, Instructions, Commands
- Symbols
- Operators: Concatenation, Arithmetic, Comparison, Logical
- REXX built-in Functions
- Parsing strings

USING REXX IN A WINDOWS ENVIRONMENT

Course contents continued

- Decision and Loop instructions
- Data Stack management
- Subroutines and Functions
- The REXX TRACE facility
- External functions
- REXX commands
- File handling commands