

# [By OnlineInterviewQuestions.com](http://OnlineInterviewQuestions.com)

**Rexx** is the acronym for **Restructured Extended Executor**. It is a high-level programming language developed by Mike Cowlishaw to make learning and reading easier. There are multiple applications of Rexx. It is used for a scripting and for processing data. It is also as an internal macro language in some software like THE and ZOC. Rexx is a versatile programming language and can be mixed with various commands to different host environments. It is a very useful language for beginners and the demand for experienced computer professionals is growing exponentially.

This blog covers important **Rexx interview questions** that can help you ace your Rexx interview. If you wish to stand out from the crowd, these interview questions are your best friend.

## **Q1. What is Uni-REXX?**

Uni-REXX is a UNIX implementation of Rexx programming language which offers rich sets of functions that are particularly designed for UNIX environment. Uni-REXX is now available on SUN's Solaris 7/8/9, H/P's HP/UX 10/11, IBM's AIX 4/5, SGI's IRIX 5/6, NCR UNIX, Linux Intel and S/390. It is used for several purposes such as automating system administration tasks. It also helps in development for end-user applications and rapid-prototyping of compiled-language applications.

## **Q2. Enlist the features of Rexx as a programming language.**

Following are the few features of Rexx-

- Rexx language has a very simple syntax format.
- Rexx can support multiple functions, procedures and route commands to several environments.
- It comes with very few artificial limitations and provides crash protection.
- Rexx is based on character strings and dynamic data typing.
- Rexx only supports UNIX style command line parameters.

## **Q3. Does Uni-Rexx give me any capabilities designed specifically for the UNIX environment?**

Rexx offers a rich set of functions other than the standard language features. These functions are designed specifically for the UNIX environment. The functions are categorised in the following groups-

- Environment control- (CHDIR, GETENV, PUTENV, GETCWD, etc.)
- Process management- (\_GETPID, \_FORK, \_WAIT, \_KILL, etc.)
- File and directory management- (\_OPENDIR, \_REaddir, \_STAT, etc.)
- Regular expression processing (\_REGEX)

- Interprocess communications (\_SOCKET, \_CONNECT, \_SEND, \_RECV, etc.)
- System error processing (\_ERRNO, \_SYSERRLIST)
- Configuration management (\_GETUID, \_GETHOSTNAME, etc.)

**Q4. Is it possible to combine Uni-Rexx with a compiled language in any application? What applications interfaces are there?**

there are wide Application Programming Interfaces (APIs) available in uni-Rexx. These APIs help in starting a new Rexx programming language from a compiled language program. It can access Rexx program variables and Rexx program stacks. These APIs also help to define new host command environments for Rexx. If you want to specify user-supplied exits or terminate the Rexx programmes, these APIs can be used. Add user-defined functions called by Rexx programmes with the help of API.

**Q5. What does the sample library of Uni-Rexx consist of?**

In the same library offered by Uni-Rexx, we get a variety of applications such as-

- Illustrations for using API to embed Uni-REXX as a scripting language.
- Utilities for- data calculations, stimulating the “which” command for UNIX, running dbxw debugger on a process, printing on a remote host.
- Functions to- perform a various string search, adds math functions to uni-REXX
- Filter to- remove backspace characters from an input stream, uppercase all characters in an input stream.

**Q6. What is the use of Uni-Rexx?**

Uni-Rexx is a simple language for beginners. It is a high-class versatile programming language that comes with a room of benefits for its users. The uses of Uni-Rexx programming language are-

- It is used for automating system administrative tasks.
- It helps in the development of end-user applications.
- Uni-Rexx can be embedded as system control language in various applications.
- Uni-Rexx supports rapid prototyping of compiled-language applications.

**Q7. Is it possible to port existing applications from the mainframe or Os/2? Are there major changes required to do this?**

Yes, it is possible to port the existing applications from the mainframe or Os/2. Also, you can port with minimum alterations. However, the following modifications are required-

- Modifying references to disk file names.
- Modify the statements in the program that can execute operating system commands. EXECIO and

GLOBALV are two exceptional mainframe commands that can be accessed from within a Uni-REXX program.

- Modifications in parts of the programs that stack data for use by some non-Rexx program.
- Alterations in the stack available to all processes. The alteration here is very basic and a technical support is available to provide guidance and suggestions.

## Q8. What are the functions available in Rexx?

there are some inbuilt functions available in various programming languages. Rexx also comes with certain inbuilt functions. They are-

- Address- the environment where the Rexx command is currently running, the Address function returns the name of the same environment.
- Beep- this function is used to produce a particular sound in the system.
- DataType- if the input is a valid number, it will return the value of "NUM". however, it will return "CHAR" if it is an invalid number. With this function, you can also specify if you want to compare the input value to a NUM or CHAR.
- Form- used for mathematical calculation on the system, this function returns the current setting of "numeric form".
- ERRORTXT- this function will return the Rexx error message associated with 'errorno'. However, the error number needs to be between 0-99. USERID- this function is used to return the current user ID logged into the system.

## Q9. What are the various error conditions seen in Rexx?

There are various error conditions seen in Rexx. They are:

- **ERROR**- when a command sent to the operating system results in an error, this event is raised.
- **FAILURE**- when a command sent to the operating system results in a failure, this event is raised.
- **HALT**- when an operation is dependent on another operation, HALT is raised.
- **NOVALUE**- this is raised when no value is assigned to a variable.
- **NOTREADY**- when an I/O device is not ready to accept any operation, NOTREADY is raised.

## Q10. What are various loop types supported by Rexx?

Rexx supports do loop, do-while loop and do-until loop. Do loop is used to execute a statement for a given number of times. The do-while loop is used to stimulate the simple while loop which is present in other programming languages. Do-until loop exits when the condition is false.

Please Visit [OnlineInterviewquestions.com](http://OnlineInterviewquestions.com) to download more pdfs