What's Right with Rexx:

Can Rexx Be Used for General Application Development?

Gil Barmwater
(gil_b@bellsouth.net)
Independent Consultant

Outline

- Something different - a discussion
- Background
- Three questions (and, hopefully, answers)
- Some examples
- Other questions from you

Discussion

- "...see if we can determine the suitability of Rexx for general application program development in today's IT environment. "
- Need your input, comments, etc.

Background

- "Rexx is the second most popular programming language..."
  - ► But nobody knows it!
- Reaction of "outsiders" when told you program in Rexx
  - ► "What's Rexx?" or "I heard of that once."
- Comments in newsgroups and discussion lists, even comp.lang.rexx
  - ► Implication that Rexx is somehow not a "serious" programming language

Three Questions/Criticisms

- The ones listed in the abstract:
  - Isn't that a Mainframe language?
  - It's just a scripting language.
  - But Rexx is different!

Question 1 :

- ► Isn't that a Mainframe language?
  - – Originally developed for IBM VM/SP CMS
  - – BUT:
    - Now truly multi-platform, from PDAs to PCs, and intermediate systems to "big iron"
    - Both hardware and operating systems that run Rexx cover a wide range of systems

Question 2:

- Isn't Rexx just a scripting language?
  - Usually means a couple things:
    - It doesn't have a "full" set of programming constructs, i.e. like DOS Batch
    - It is too slow to run any "real" programs

Question 2 (con't.):

- Full set of modern programming constructs?
  - Initial design included concepts like support for structured/modular code and dynamic scoping
  - Object Oriented support further extends ability to use Rexx in multiple development methodologies

Question 2 (con't.):

- Aren't interpreted languages "slow"?
  - ► Yes, slower than compiled code
  - ► BUT, it may not matter:
    - Today's processors are very fast
    - If the program interacts with a user or does file/database I/O, that will be the constraining factor on performance
    - Does it matter if the next screen "prompt" appears in 0.1 sec or 0.001 sec? Or if the report takes 5 sec. to complete instead of 0.05 sec?

Question 3:

- Isn't Rexx "different" somehow?
  - ► Sure, I believe it is BETTER!
  - ► Much quicker cycles through the code/test loop (no build step)
  - ► Much easier to read and understand already written code when it needs modification
  - ► Allows "non-programmers" to solve business problems

Some Examples

- My client's environment
- A "batch" scheduler
- A "control" client
- An object-oriented business model

Other questions/criticisms?

- What comments about using Rexx have you heard that we might address?

Summary

- Rexx is a valid programming language choice for development of application projects of all sizes