
LinuxandLinuxforzSeries: CGIProgramming withREXX

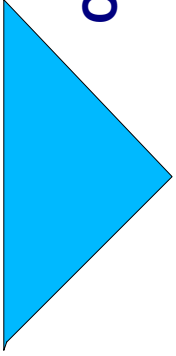
Session8311

Speaker:JanEngelhausen,IBMGermany
REXXDevelopment
eMail:jan.engelhausen@de.ibm.com



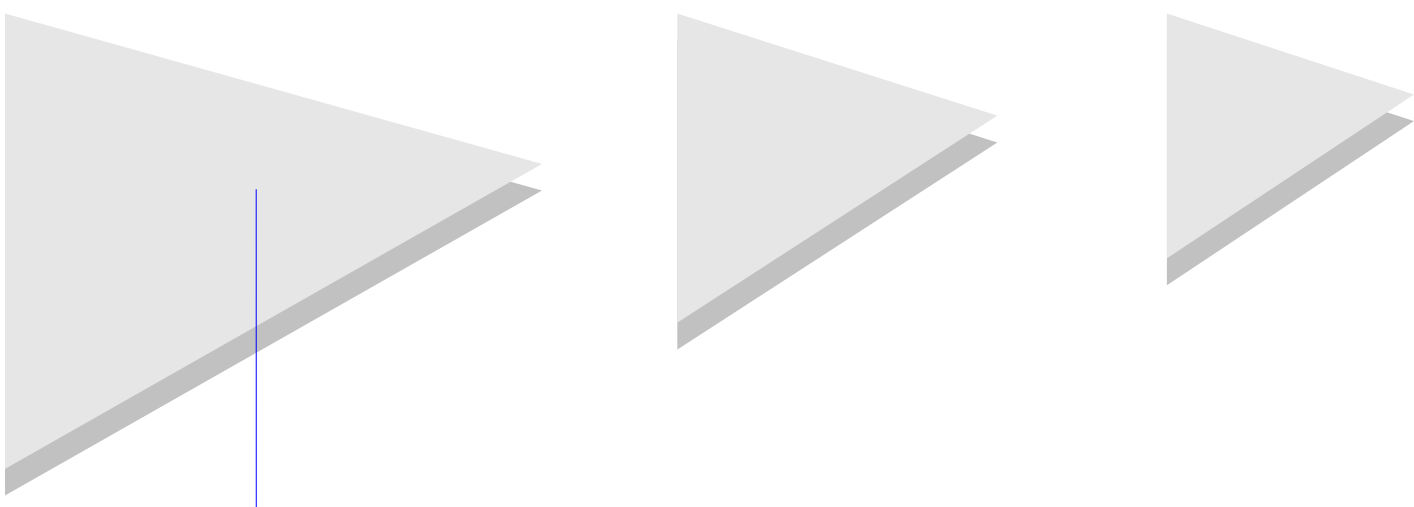
SHARE-03/01/01-LongBeach,USA

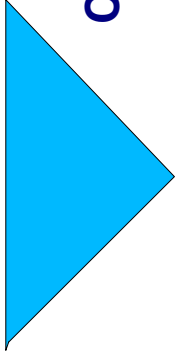




Agenda

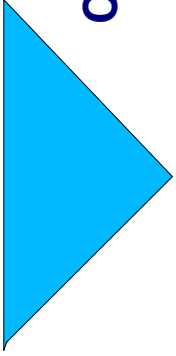
- Intro: Demo of CGI Application
- CGI Fundamentals
- REXX at work
- The REXX Scripting Parts
- The APACHE configuration
- Hints and Tips
- Summary





Demo of WebVisitors' Book

- REXX generates all the web pages you will see.
- it reacts on input data from web forms,
- evaluates the input data,
- stores it into a file ("database").
- REXX creates new web pages depending on the input data.

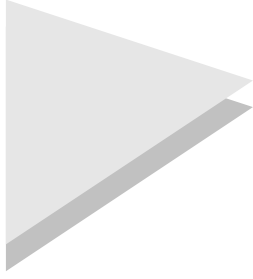
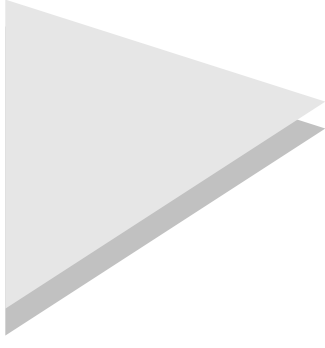


CGI Programming with REXX SHARE2001

REXX at work

Start the demonstration
with a click on

<http://localhost/rxquest>



CGI fundamentals

- CGI stands for "Common Gateway Interface"
- It is used to generate dynamic web content on a server
- If the web server receives a request for a CGI-enabled URL, a script is executed that produces the web page
- A script can obtain possible parameters from a special environment variable or through STDIN

CGI Fundamentals, II

- Once the web server is configured, it invokes a script that has to write the content to STDOUT
- Every output must start with a MIME type specification, almost always:
 - `Content-Type: text/html<linefeed>`
 - `<linefeed>`
 - `<...content follows here...>`
- Parameters are passed depending on the method (GET or POST) used

CGI Fundamentals, III

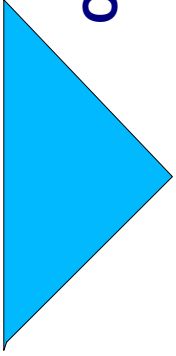
- Using GET in a form results in passing the arguments appended to the url:
 - `<URL>?<field1>=<content>&<field2>=<content>&...`
 - Example: `http://localhost/getParms?name=johndoe`
- Special characters, e.g. `=` and `&` are encoded (see our scripts)
- The `?...string` is available to the script in the environment variable **QUERY_STRING**
- Using POST in a form means that the input data is passed in the STDIN stream
- we will only use the first method, but it is advisable to use POST if you have big data

AsimpleCGI REXXscript

- Letsputtheoryintopractice:

- `#!/usr/bin/rexx`
- `/* my first CGI REXX script: 'first' */`
- `Say 'Content-Type: text/plain' -- MIME type plain text`
- `Say -- one empty line is needed`
- `Say 'Hello from Object REXX'`
- `Say 'Date and time are:' date() time()`
- `Say 'QUERY_STRING:' value('QUERY_STRING','ENVIRONMENT')`

- Let'sgoto<http://localhost/orxwww/firsttoseewhatitdoes>

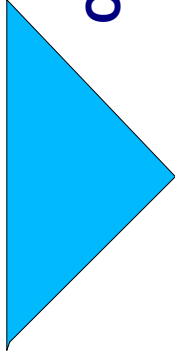


REXX at work: The scripts

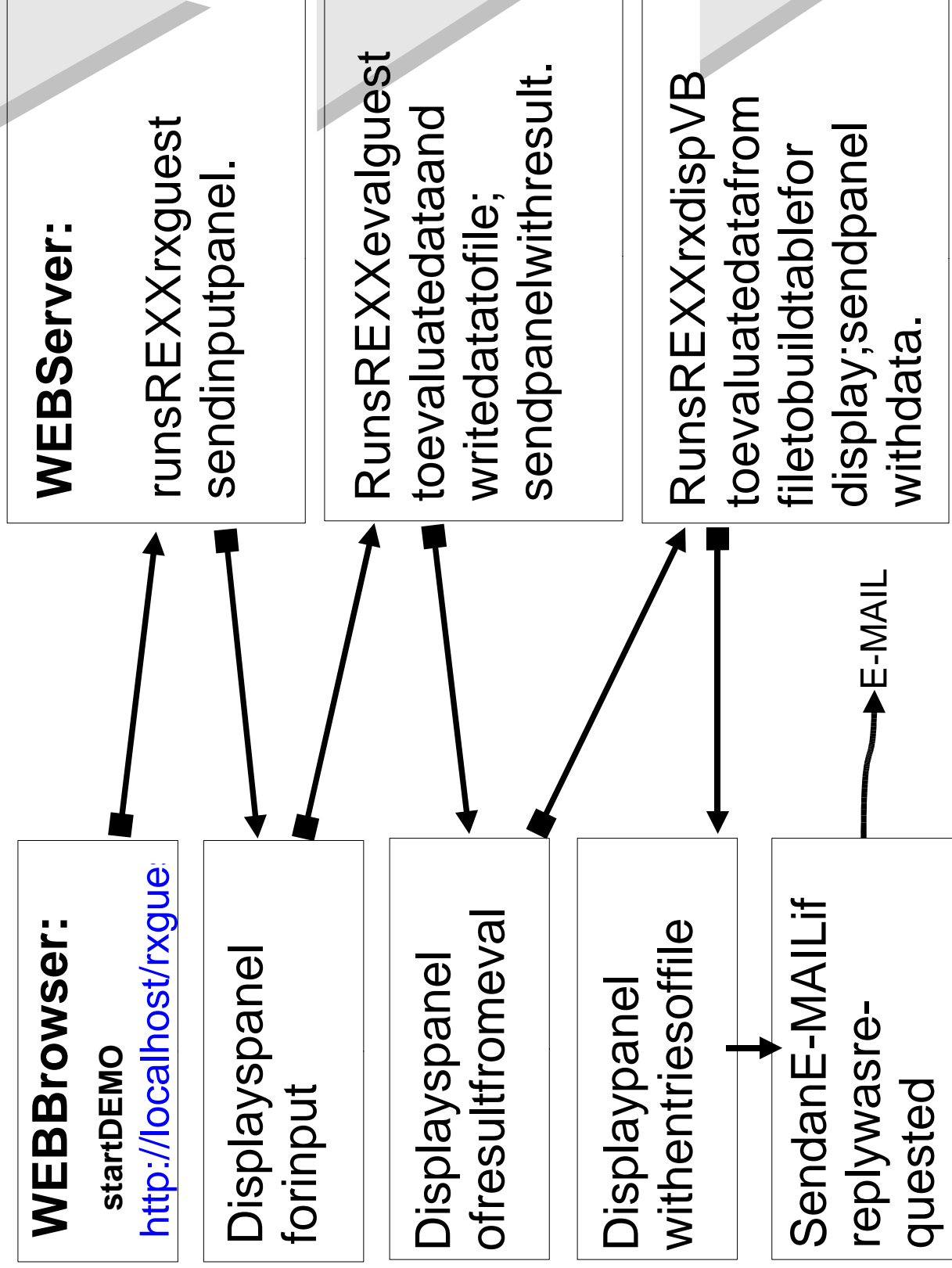
- The REXX script *rxquest* generates the input panel
 - The Apache web server runs the REXX script and
 - sends the script output to the requester
- The input of the user is sent from the browser to the server
 - It runs the REXX script *evalquest*
 - The script checks the input and according to what was given sends data back to the requester (through the server)

Rexxatwork:Thescripts

- Thenextpaneldisplays
 - whetheranentryintotheVisitors'Bookhasbeenmade and
 - whatpositionnumberithasgot
- Thelast10entriesoftheVisitors'Bookcanbedisplayed.
 - Ifthisisdesired,theREXXscript **rxdispVB**isrun
 - Itselectsandprepares the data from the Visitors' Book and
 - theserver sendsthe output of the script to the requesting browser

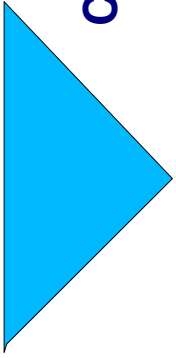


CGI Programming with REXX SHARE2001



REXX at work

- The REXX interpreter is activated through the first line in the script **#! /usr/bin/rexx**.
- The REXX script must be owned by user “nobody” and readable and executable.
- The REXX script must be in the **cgi-bin** directory defined in the Apache server's configuration file (see next chapter).
- REXX issues the HTML tags with the **say** function to STDOUT and the Apache server sends them to the requester.
- REXX evaluates the global variable **QUERY_STRING** to get the input parameters.
 - ➔ Therefore the method GET is used to pass/retrieve data



TheREXX ScriptingParts

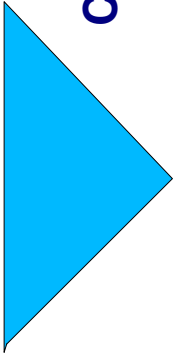
Seethe3REXXscripts: rxquest, evalguest, rxdispvB

```
/* standard header for URL */
```

```
say "Content-type: text/html"  
say
```

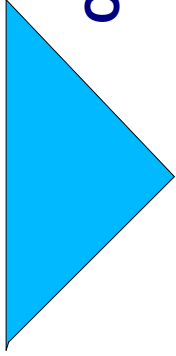
```
/* If double quotes are needed for a  
parameter definition of a HTML tag, use  
the single quote for the string  
delimiter. */
```

```
say '<img align='right' src='/orxok.gif'>'
```



The APACHE configuration

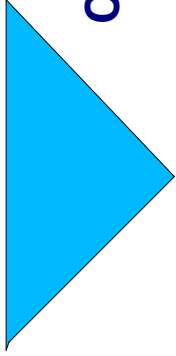
- Definitions are in the configuration file:
httpd.conf (`indirectory /etc/httpd/conf`)
- Define internal user nobody and group nobody
 - ✓ User nobody
 - ✓ Group nobody
- Define the alias for the REXX scripts
 - ✓ `ScriptAlias /orxwww/ "/home/httpd/cgi-bin/orxwww/"`
 - ✓ `ScriptAlias /rxquest "/home/httpd/cgi-bin/orxwww/rxquest"`
- Define the `ServerName` parameter



The APA CHE configuration

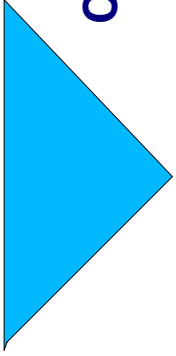
- Add the following definition to the configuration file:

```
<Directory "/home/httpd/cgi-bin/orxwww">  
    AllowOverride None  
    Option ExecCGI  
    Order allow,deny  
    Allow from all  
</Directory>
```



Hints and Tips

- The **shellprocessor** (e.g.:bash) will not work if the REXX scripts contain CR/LF as line delimiters. This can happen if the files are copied from Windows(TM) or OS/2(TM).
- Install the RPM image of Object REXX to get links already setup.
- Use **#! /usr/bin/rexx** NOT **#!/usr/local/orc/bin/rexx** in your REXX scripts (because the link will not be changed).
- Get the Apache configuration in one file. This can be achieved by commenting out the following lines:
 - A) `#ResourceConfig/srm.conf`
 - B) `#AccessConfig/access.conf`



Hints and Tips

- In case of problems see the APACHE's log file: **error_log** in the directory: **/var/log/httpd** (defined in Apache's configuration file).
- Also the “tracer” output and errors from the REXX scripts are logged in the **error_log** file.
- The guesteval screen shows all variables sent by the requester if &debug=envis added at the end of the Location string (browser) and the >enter< key is pressed.
- A REXX script can be called in a window on the server system and debugged.

Hints and Tips

- Do not allow user to access the root directory for security reasons. Specify the following in the `APACHE` configuration:

```
Change:  <Directory /usr/doc>
         Options FollowSymLinks Indexes
         +Includes AllowOverride None
       </Directory>
```

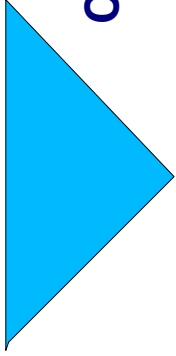
```
To:      <Directory /usr/doc>
         order deny,allow
         deny from all
         allow from localhost
         Options Indexes FollowSymLinks
         +Includes AllowOverride None
       </Directory>
```

- `User nobody` must not log on to the system.
- `User nobody` must have only group permission of `nobody`.
- `User nobody` stores the database of `fvbook.txt` file in the sample.
- `User nobody` must have very limited permission.

Hints and Tips

- Reduce the use of the SAY instruction by issuing more than one line of a HTML string:
- This will structure your REXX script and accelerate the issuing of the HTML page
(The comma at the end of a line is for continuation)

say "<head> <title> Object REXX visitors' book</title> </head>","
"<h1>Visitors' book of Object REXX on server 'srv_name':</h1>","
"
 Date: "date()"; Time: "time()";



Summary

You will like writing REXX CGI scripts because:

- * You now know how to set up the Apache web server
- * REXX is an easy-to-use scripting language
- * REXX is powerful in handling strings - needed for CGI!
- * Changes are debugged and tested quickly
- * Our sample scripts can be obtained from our website

Get our samples and try for yourself,

 **It is easy to write CGI scripts with
• Object REXX!**

ObjectREXX-Furtherinformation

- Documentation
 - Availableonline(+Samples)&inPDF
- Internethomepages:
 - REXXlanguage:
<http://www2.hursley.ibm.com/rexx>
 - ObjectREXX:
<http://www.ibm.com/software/ad/obj-rexx>
- E-Mailyour questionsto:
 - orexx@vnet.ibm.com