Rexx/WS

2014 International Rexx Symposium
Mark Hessling

5 May 2014
Memphis, Tennesee
USA

Rexx/WS

- Enables a Rexx program to function as a Web (HTTP) and Websockets (WS) server
- Based on libwebsockets
  - http://libwebsockets.org
  - Processing via ”callbacks”
- Does not work with ooRexx
  - Requires RexxCallback API

Rexx/WS - Initialisation

- Add protocols – HTTP and/or WS
  - Specify Rexx procedure to call back to
  - Name of protocol (for browser)
  - Maximum message size
- Create context (plumbing for dispatching events)
  - Listening port for server
  - Timeout (ms)

```
rc = WSAddProtocol( !REXXWS.!PROTOCOL_HTTP, 'httpcallback' )
rc = WSAddProtocol( !REXXWS.!PROTOCOL_WS, 'wscallback', 'simple', 5000 )
...
context = WSCreateContext( 8888, 50000 )
```

Rexx/WS – Processing Loop

- Call event dispatcher

```
Do Forever
        rc = WSService( context, 5000 )
        /* do stuff after servicing events or timeout */
End
```

Rexx/WS – Callbacks

- Structure of callback procedures
  - Arguments
    - Context – from WSCreateContext()
    - WSI – Handle to instance of WS client
    - Reason – event type
    - Text – file name (for HTTP) or message payload (for WS)
  - Return code
    - Indicates what libwebsockets should do to client connection

Rexx/WS – HTTP Events

- WS_CONNECTED
  - A client has connected to our HTTP server
- HTTP_CLOSED
  - A client has disconnected from our HTTP server
- HTTP_REQUEST
  - A client has requested a file; HTML, PNG,...
  - WSServeHTTPFile()
    - Send requested file back to client (browser) with correct MIME type

Rexx/WS – WS Events

- WS_CONNECTED
  - A client has connected to our WS server
- WS_CLOSED
  - A client has disconnected from our WS server
- WS_RECEIVE
  - Client has sent a message (text only at this stage)
  - WSWrite()
    - Send (text) message back to client

Rexx/WS - Termination

- WSDeleteContext(context)

Rexx/WS – WS Client

- Rexx/WS can also act as a WS client
  - WSAddProtocl() - WS only
  - WSCreateContext()
    - Port is NO_LISTEN
  - WSService()
    - Callbacks
  - WSDeleteContext()

Rexx/WS – WS Client Events

- WS_CLIENT_CONNECTED
    - Our client has connected to a WS server
- WS_CLOSED
    - Our client has been disconnected from WS server
- WS_CLIENT_RECEIVE
    - Our client has received a message (text only at this stage) from a WS server
    - WSWrite()
        - Send (text) message back to server

Rexx/WS – Demo & Code

- Demo chat server
    - URL to be supplied
- Code walkthrough

Regina and Websockets

- rxstack a Websockets server (Regina v3.8)
- Demo of Javascript talking to rxstack

Questions

?