The ooRexx Incubator...Hatching Future Extensions

Rick McGuire
2008 Rexx Symposium

What is an incubator?

in·cu·ba·tor
Pronunciation:
\ˈiŋ-kyə-ˌbā-tər, ˈin-\
Function:
noun
Date:
1857

: one that incubates: as a: an apparatus by which eggs are hatched artificially b: an apparatus with a chamber used to provide controlled environmental conditions especially for the cultivation of microorganisms or the care and protection of premature or sick babies c: an organization or place that aids the development of new business ventures especially by providing low-cost commercial space, management assistance, or shared services

What is the Rexx incubator

- A place where new potential ooRexx additions can be nurtured and grown
  - Brought up to project coding standards
  - Played with to understand what function needs to be added
  - Act as a demonstration of what can be done
- Not every project in the incubator will end up in the base product
  - But we hope many will

Where is the incubator?

- The incubator is a special section of the ooRexx code tree
- SVN checkout:
  http://oorexx.svn.sourceforge.net/svnroot/oorexx/incubator
- Direct browsing:
  http://oorexx.svn.sourceforge.net/viewvc/oorexx/incubator/

Many projects current in the incubator

- Most (currently ALL) written entirely in ooRexx
- Contributed from a variety of sources
- Some of these, you've already heard about this week:
  - David Ashley's Internet classes
  - ooRexxTry
  - DecimalFormat

CSVStream

- Contributed by Jon Wolfers
- A wrapper class that allows read and writing of CSV (Comment Separated Value) files
  - Handles encoding of the values, including escaping of literals
  - Allows access to header values and data
  - Gives direct access to “rows” of data

orx-irc

- Contributed by Moritz Hoffmann
- Object-oriented wrapper for the IRC (Internet Relay Chat) protocol.
  - Only a library, so programmers must enhance it. Many different possible use cases: chat clients, bots etc.
  - Wraps each IRC entity into objects: Servers, users, channels
  - Supports invocation of IRC functions as object methods that block until the whole result has been received.

rexxdoc

- Contributed by Moritz Hoffmann
- (oo)Rexx source code documentation application.
  - Similar to JavaDoc: Comments starting with /** are recognized as rexxdoc comments.
  - Consists of a parser and front end.
    - Parser understands most Rexx code and transforms it into parse-tree.
    - Front end converts tree information info documentation.
    - Alternate front ends can render information in different formats.

Dynamic ClassLoader

- Contributed by Moritz Hoffmann
- Allows loading any Rexx class dynamically without naming its source file
  - Layered architecture to allow a pluggable back end tree.
  - Allows packaging of multiple source files together in TAR or ZIP archives
  - Allows loading classes under different names that their original name to avoid name clashes.
  - Future version may support network class loaders

```
::REQUIRES "test~testpackage(someclass as somecl, otherclass)"
```

Enumeration

- Contributed by Moritz Hoffmann
- Basic enumeration class
  - Define an enum type:

    ```
    ::Class Colors Subclass Enumeration -- The type
    ::Class Red Subclass Colors -- An item
    ::Class Blue Subclass Colors ...
    ```
  - Obtaining an item:

    ```
    item = .Colours~Red
    ```
  - Operations common to all items are to be defined in the enumeration type
  - Implementation replaces equality operators

    ```
     Colour~red = item yields .true (see above)
    ```

Serialization

- Contributed by Moritz Hoffmann
- Object Serialization/Deserialization framework.
  - Allows any kind of Object to be serialized to a stream.
  - Restoring a serialized object restores all references. The new object is a deep-copy of the original.
  - Enhanced stream class exists to read and write objects.
  - Circular references and special characters in strings are no problem.
  - Serialization either automatic or via special methods.

altstream

- Contributed by Rick McGuire
- Alternate stream implementations
  - Not backed by an actual I/O stream
  - Input/Output to/from a Rexx Array
  - Output to a mutablebuffer
  - Input from a string value
  - An alternate stream can be plugged into any situation where a stream object can be used

regular expressions

- Contributed by Rick McGuire
- New framework for regular expression matching
  - Loosely based on Java regex classes
    - Separate Pattern and Matcher classes
  - Pattern compiler is separate from the pattern engine
    - Allows any regular expression syntax to be supported
  - Regex replacement will eventually be supported
  - Written to allow custom pattern types to be added
  - Libraries of common pattern matchers possible

xml

- Contributed by David Ashley
- XML Framework
  - Currently just a SAX-like XML parser
  - Beginning of a DOM (Document Object Model) API
  - Will eventually expand to having a DOM parser
  - Eventual target is to have full XPATH query support

pipes

- Contributed by Rick McGuire
- A CMS pipelines-like framework
  - Currently just a sample that ships with ooRexx
  - Would like to make it fully supported class library
  - Many new stages added since last year
  - Remodeled after the Java pipelines package

???????

- Contributed by [*your name here*]
- Neat class for ???????
  - New contributions always welcome!
  - We need more people involved with the ooRexx project
  - Assistance with other items in the incubator also welcome