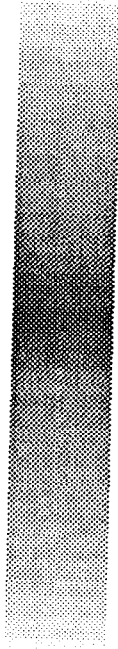# Object REXX—What's New?

Tom Brawn
IBM

Object REXX - What's new?

Tom Brawn
Object REXX Development
IBM Endicott, NY
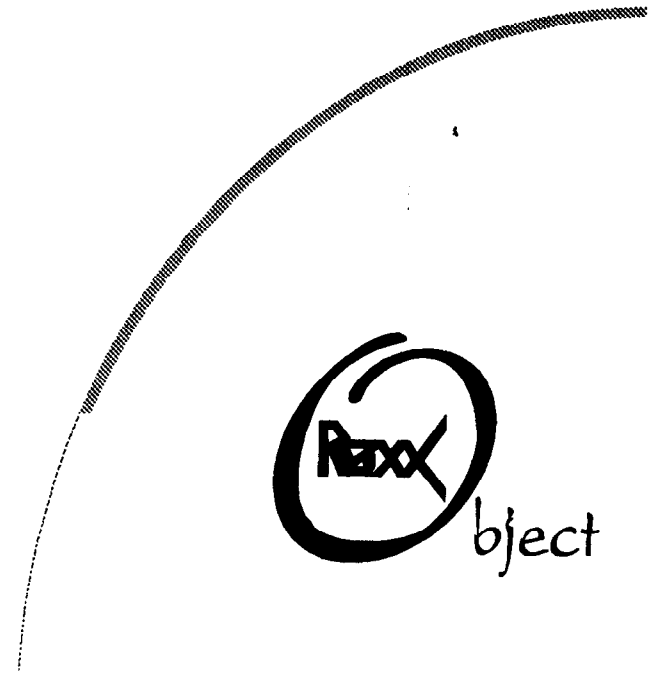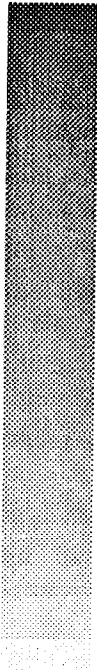Internet id: tombrawn@vnet.ibm.com

Object

Background

- Work began in 1988
- Prototyped since 1989
- Complete rewrite of interpreter
- Language architecture "in progress" and subject to change
- Significant enhancements over the past 18 months
- Limited beta on OS/2 and Windows 4/94

Object

Why Object REXX?

- Remove limitations of current REXX language
- Bring the power of OO programming to REXX
- Bring the usability of REXX to OO programming
- Extend REXX usage
  - windowing,
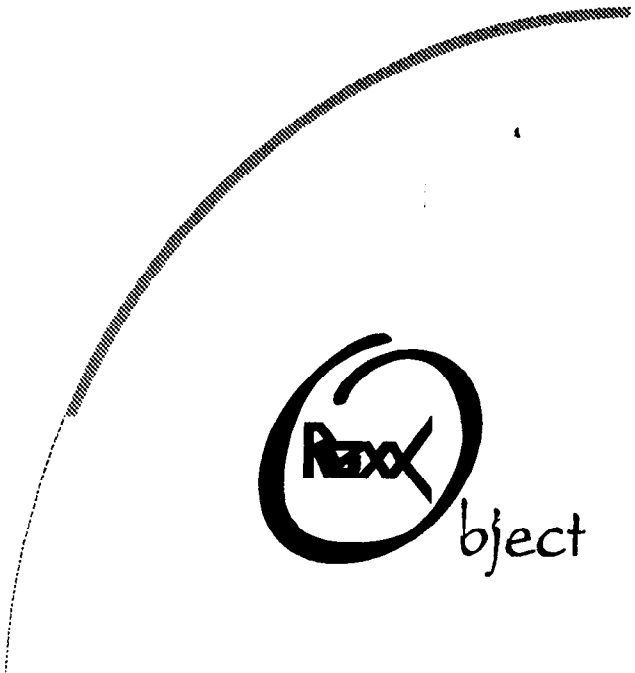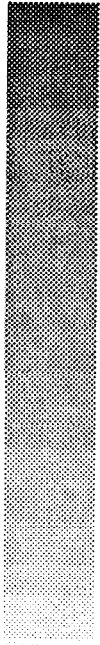
Object

What's New in Object REXX?

- Objects
  - Everything in Object REXX is an object
- Methods
  - Everything that happens in Object REXX is a method
- Messages
  - Everything that happens in Object REXX is caused by a message

Object

New REXX changes...

- Expressions in stems
- Parse enhancements
- Countstr/Changestr functions
- Extended Do syntax
- Date conversions

Expressions in stem references

- Allow expressions on right side of stem .'s
- Use [ ] to contain stem expressions
- Example:

Before...

```
tom =index + 1
chris = index + 2
if employee.tom.chris ...
```

Is written...

```
if employee.[index+1].[index+2] ...
```

Parse enhancements

- parse caseless...
  - Parse template match without regard to case
  - Example:

    ```
    parse caseless value 'Out To Lunch' with first ' to ' last
    say first ==>Out
    say last ==>Lunch
    ```

- parse lower...
  - Translate to lower case, then parse
  - Example:

    ```
    parse lower value 'Out To Lunch' with first ' to ' last
    say first==>out
    say last==>lunch
    ```

countstr/changestr

- countstr(needle, haystack)
  - Returns count of needle in haystack
  - Example:

  countstr('is','This is a test of counstr') ==> 2

- changestr(needle,haystack,new)
  - Returns copy of haystack in which new replaces all occurences of needle
  - Example:

  changestr('1','101100','X') ==> 'X0XX00'
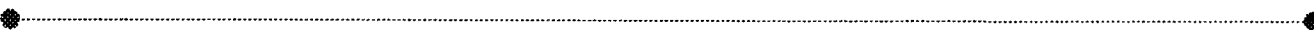
Extended DO

- Adds ability to iterate over stems
- Example:

```
fred.1='Tom';fred.2='Chris';fred.3='Alex';
do tail over fred.
  say fred.tail
end

==>Alex
==>Chris
==>Tom
```

Date Conversions

- Convert date to standard formats
- Example:

```
say date('usa',19931225,'standard')
==>12/25/93

say date('european',19931225,'standard')
==>25/12/93
```