

9) ADDRESSing

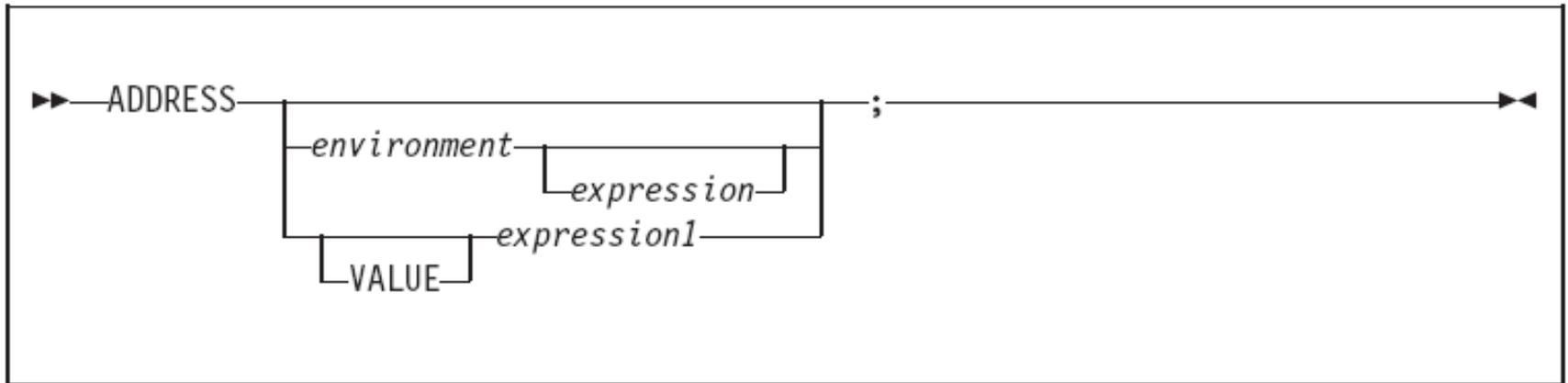
Each line that is not resolved as a REXX code is treated as a command and passed to environment.

By REXX instruction ADDRESS you can specify to which command processor will the line be passed.

This way you can invoke also external subroutine. In this case no result except RC will be returned back.

By TSO command OUTTRAP you can catch the output on the screen and work with it in Your REXX program.

ADDRESS



ADDRESS temporarily or permanently changes the destination of commands. Commands are strings sent to an external environment. You can send commands by specifying clauses consisting of only an expression or by using the ADDRESS instruction.

ADDRESS

For example, in REXX processing, a host command can be:

- A TSO/E command processor, such as ALLOCATE, FREE, or EXEC
- A TSO/E REXX command, such as NEWSTACK or QBUF
- A program that you link to or attach
- An MVS system or subsystem command that you invoke during an extended MCS console session
- An ISPF command or service
- An SAA CPI Communications call or APPC/MVS call

ADDRESS

```
/****** REXX *****/
/* Program */
/* address1 */
/* Description */
/* REXX program to test statements */
/* Author      : Michaelangelo DeParma */
/* Date        : 2nd February 2000. */
/*----- Amendment History -----*/
/*******/
ADDRESS "TSO"
"CLEAR"
"LISTC LEVEL(IULC39)"
```

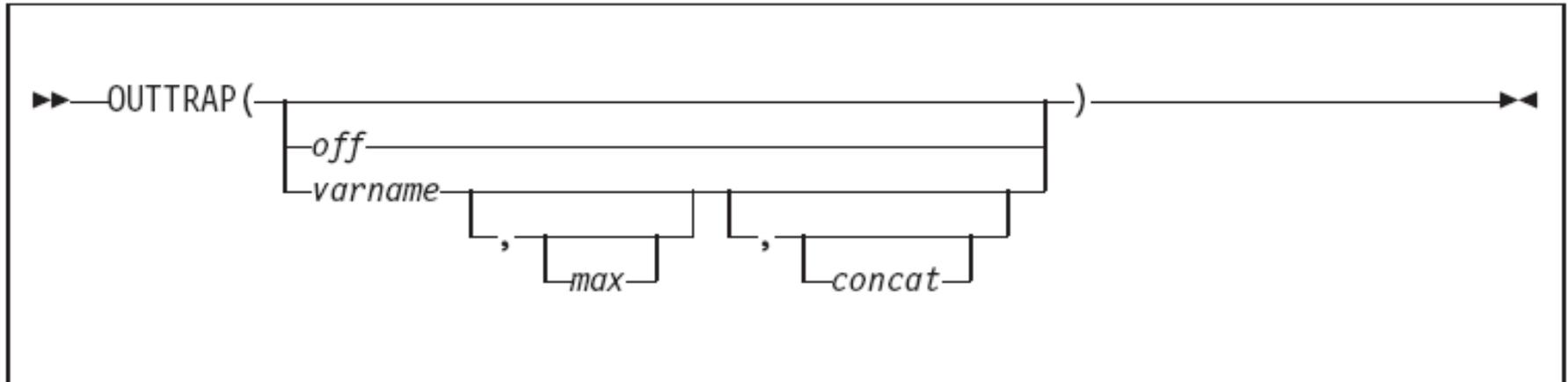
ADDRESS

```
/****** REXX *****/
/* Program */
/* address2 */
/* Description */
/* REXX program to test statements */
/* Author      : Michaelangelo DeParma */
/* Date       : 2nd February 2000. */
/*----- Amendment History -----*/
/*******/
ADDRESS "MVS"
"CLEAR"
ADDRESS "TSO" "LISTC LEVEL(IULC39) "
```

ADDRESS "MVS" - Makes the change permanent from the program or until another ADDRESS "ENV" is detected.

ADDRESS "TSO" "COMMAND" will change the environment for the one command only.

OUTTRAP



`OUTTRAP` returns the name of the variable in which trapped output is stored, or if trapping is not in effect, `OUTTRAP` returns the word `off`.

You can use the following arguments to trap lines of command output into compound variables or a series of numbered variables, or to turn trapping off that was previously started.

off specify the word `OFF` to turn trapping off.

varname

the stem of the compound variables or the variable prefix assigned to receive the command output. Compound variables contain a period and allow for indexing, but lists of variables with the same prefix cannot be accessed by an index in a loop.

Note: Do not use “`OFF`” as a variable name.

OUTTRAP

max the maximum number of lines to trap. You can specify a number, an asterisk in quotation marks (“*”), or a blank. If you specify “*” or a blank, all the output is trapped. The default is 999,999,999. If the maximum number of lines are trapped, subsequent lines are not stored in variables.

concat

indicates how output should be trapped. For *concat*, specify one of the following:

- **CONCAT**

indicates that output from commands be trapped in consecutive order until the maximum number of lines is reached. For example, if the first command has three lines of output, they are stored in variables ending in 1, 2, and 3. If the second command has two lines of output, they are stored in variables ending in 4 and 5. The default order for trapping is CONCAT.

- **NOCONCAT**

indicates that output from each command be trapped starting at the variable ending in 1. For example, if the first command has three lines of output, they are stored in variables ending in 1, 2, and 3. If another command has two lines of output, they replace the first command’s output in variables 1 and 2.

OUTTRAP

```
/****** REXX *****/
/* Program */
/* outtrap1 */
/* Description */
/* REXX program to test statements */
/* Author : Michaelangelo DeParma */
/* Date : 2nd February 2000. */
/*----- Amendment History -----*/
/*******/
ADDRESS "TSO"
"CLEAR"
un_used = OUTTRAP("line.", "*", "NOCONCAT")
"LISTC LEVEL(IULC39)"
un_used = OUTTRAP("OFF")
```

OUTTRAP

OUTTRAP is used to retrieve the output from the TSO commands.

As a function a value is returned and is set to a variable (un_used).

The data from the command (LISTC) is stored in an array with a stem variable.

"*" indicates how many lines of data to be retrieved.

"NOCONCAT" ensures that anything in an OUTTRAP before the command is over written.

"OFF" switches off the trap and ensures that no more commands will go to it.

Retrieving the OUTTRAP

```
/* ***** REXX ***** */
/* Program */
/* outtrap1 */
/* Description */
/* REXX program to test statements */
/* Author      : Michaelangelo DeParma */
/* Date       : 2nd February 2000. */
/*----- Amendment History -----*/
/* ***** */
ADDRESS "TSO"
"CLEAR"
un_used = OUTTRAP("line.", "*", "NOCONCAT")
"LISTC LEVEL(IULC39)"
un_used = OUTTRAP("OFF")
DO line_counter = 1 TO line.0
  SAY "Line "||line_counter||" contains : "||line.line_counter
END
```

Retrieving the OUTTRAP

One of the features of the array is that STEM.0 contains the number of elements.

e.g.

line.0 contains how lines are return from the "LISTC" command.

The DO loop is used to retrieve the result.

The output is shown on the next page.

```
Line 1 contains : NONVSAM ----- IULC39.DEMOED.ISPPROF
Line 2 contains :          IN-CAT --- ICF.USERID.USERCAT4
Line 3 contains : NONVSAM ----- IULC39.OPSTCMLS
Line 4 contains :          IN-CAT --- ICF.USERID.USERCAT4
***
```

Work section 9.1

- Write a REXX program to show the result of the TSO "LU" command.

```
The result of the LU command is :  
RACF PRODUCT DISABLED: COMMAND ENDED.  
***
```

Work section 9.2

- Write a REXX program to send a message to a TSO user, using the TSO send command.

```
Please enter the message to send:
hello bob
Please enter the TSO ID :
IULC23
hello bob CRONE90
***
```

Work section 9.3

- Write a REXX program to display a userid's datasets name only. (do not include the other entries)

```
Please enter the TSO ID :  
iulc23  
IULC23.CA7.BTIOUT1  
IULC23.CA7.TEXT  
IULC23.CA7.TEXT1  
IULC23.DB2.WORK  
IULC23.DEMOED.ISPPROF  
IULC23.IULC.RULES  
IULC23.JCL.CNTL  
IULC23.OUTPUT  
IULC23.REXX.CLIST  
IULC23.SAREXP.DELCARDS  
IULC23.TEST.PDS  
***
```

Additional Program

- Write a REXX program to display a datasets volume name.

```
Please enter the dataset:  
crone90.crone.rexx  
The dataset is on volume :   DEMS03  
***
```

9) ADDRESSing

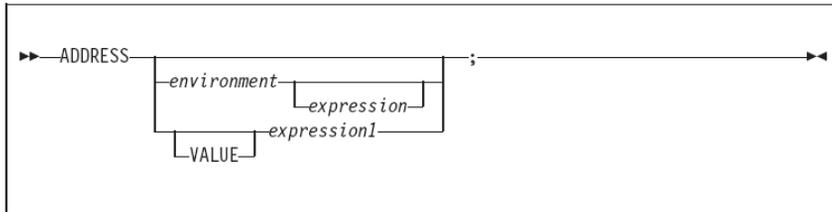
Each line that is not resolved as a REXX code is treated as a command and passed to environment.

By REXX instruction ADDRESS you can specify to which command processor will the line be passed.

This way you can invoke also external subroutine. In this case no result except RC will be returned back.

By TSO command OUTTRAP you can catch the output on the screen and work with it in Your REXX program.

ADDRESS



ADDRESS temporarily or permanently changes the destination of commands. Commands are strings sent to an external environment. You can send commands by specifying clauses consisting of only an expression or by using the ADDRESS instruction.

ADDRESS

For example, in REXX processing, a host command can be:

- A TSO/E command processor, such as ALLOCATE, FREE, or EXEC
- A TSO/E REXX command, such as NEWSTACK or QBUF
- A program that you link to or attach
- An MVS system or subsystem command that you invoke during an extended MCS console session
- An ISPF command or service
- An SAA CPI Communications call or APPC/MVS call

ADDRESS

```
/****** REXX *****/
/* Program */
/* address1 */
/* Description */
/* REXX program to test statements */
/* Author : Michaelangelo DeParma */
/* Date : 2nd February 2000. */
/*----- Amendment History -----*/
/*******/
ADDRESS "TSO"
"CLEAR"
"LISTC LEVEL(IULC39)"
```

4

Copyright ©2006 CA. All rights reserved. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

Here the destination for commands is set to TSO. Examples of another environments: Console, ISPEXEC, MVS,...

Each environment has its own set of commands.

See 'MCOE.REXA.REXX(RX20192)'

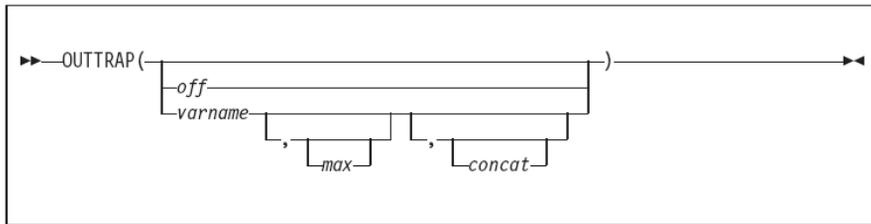
ADDRESS

```
/****** REXX *****/
/* Program */
/* address2 */
/* Description */
/* REXX program to test statements */
/* Author : Michaelangelo DeParma */
/* Date : 2nd February 2000. */
/*----- Amendment History -----*/
/*******/
ADDRESS "MVS"
"CLEAR"
ADDRESS "TSO" "LISTC LEVEL(IULC39)"
```

ADDRESS "MVS" - Makes the change permanent from the program or until another ADDRESS "ENV" is detected.

ADDRESS "TSO" "COMMAND" will change the environment for the one command only.

OUTTRAP



OUTTRAP returns the name of the variable in which trapped output is stored, or if trapping is not in effect, OUTTRAP returns the word off.

You can use the following arguments to trap lines of command output into compound variables or a series of numbered variables, or to turn trapping off that was previously started.

off specify the word OFF to turn trapping off.

varname

the stem of the compound variables or the variable prefix assigned to receive the command output. Compound variables contain a period and allow for indexing, but lists of variables with the same prefix cannot be accessed by an index in a loop.

Note: Do not use "OFF" as a variable name.

OUTTRAP

max the maximum number of lines to trap. You can specify a number, an asterisk in quotation marks ("*"), or a blank. If you specify "*" or a blank, all the output is trapped. The default is 999,999,999. If the maximum number of lines are trapped, subsequent lines are not stored in variables.

concat indicates how output should be trapped. For *concat*, specify one of the following:

- **CONCAT**
indicates that output from commands be trapped in consecutive order until the maximum number of lines is reached. For example, if the first command has three lines of output, they are stored in variables ending in 1, 2, and 3. If the second command has two lines of output, they are stored in variables ending in 4 and 5. The default order for trapping is **CONCAT**.
- **NOCONCAT**
indicates that output from each command be trapped starting at the variable ending in 1. For example, if the first command has three lines of output, they are stored in variables ending in 1, 2, and 3. If another command has two lines of output, they replace the first command's output in variables 1 and 2.

OUTTRAP

```
/****** REXX ******/
/* Program */
/* outtrap1 */
/* Description */
/* REXX program to test statements */
/* Author : Michaelangelo DeParma */
/* Date : 2nd February 2000. */
/*----- Amendment History -----*/
/*******/
ADDRESS "TSO"
"CLEAR"
un_used = OUTTRAP("line.", "*", "NOCONCAT")
"LISTC LEVEL(IULC39)"
un_used = OUTTRAP("OFF")
```

Write it and test it.

See 'MCOE.REXA.REXX(RX20194)'

OUTTRAP

OUTTRAP is used to retrieve the output from the TSO commands.

As a function a value is returned and is set to a variable (un_used).

The data from the command (LISTC) is stored in an array with a stem variable.

"*" indicates how many lines of data to be retrieved.

"NOCONCAT" ensures that anything in an OUTTRAP before the command is over written.

"OFF" switches off the trap and ensures that no more commands will go to it.

Retrieving the OUTTRAP

```
/****** REXX *****/
/* Program */
/* outtrap1 */
/* Description */
/* REXX program to test statements */
/* Author : Michaelangelo DeParma */
/* Date : 2nd February 2000. */
/*----- Amendment History -----*/
/*******/
ADDRESS "TSO"
"CLEAR"
un_used = OUTTRAP("line.", "", "NOCONCAT")
"LISTC LEVEL(IULC39)"
un_used = OUTTRAP("OFF")
DO line_counter = 1 TO line.0
SAY "Line "||line_counter||" contains : "||line.line_counter
END
```

Write it and test it.

Retrieving the OUTTRAP

One of the features of the array is that STEM.0 contains the number of elements.

e.g.

line.0 contains how lines are return from the "LISTC" command.

The DO loop is used to retrieve the result.

The output is shown on the next page.

```
Line 1 contains : NONVSAM ----- IULC39.DEMOED.ISPPROF
Line 2 contains :      IN-CAT --- ICF.USERID.USERCAT4
Line 3 contains : NONVSAM ----- IULC39.OPSTCMLS
Line 4 contains :      IN-CAT --- ICF.USERID.USERCAT4
***
```

See 'MCOE.REXA.REXX(RX20195)'

Work section 9.1

- Write a REXX program to show the result of the TSO "LU" command.

```
The result of the LU command is :  
RACF PRODUCT DISABLED: COMMAND ENDED.  
***
```

Write it and test it. Use TSO command "TIME" instead of "LU". Do not use OUTTRAP for this.

Work section 9.2

- Write a REXX program to send a message to a TSO user, using the TSO send command.

```
Please enter the message to send:
hello bob
Please enter the TSO ID :
IULC23
hello bob CRONE90
***
```

Write it and test it. Use TSO command: "send "TSOMES"
user("pmfID") logon"

The user who is not logged on cannot receive the message.

Work section 9.3

- Write a REXX program to display a userid's datasets name only. (do not include the other entries)

```
Please enter the TSO ID :
iulc23
IULC23.CA7.BTIOUT1
IULC23.CA7.TEXT
IULC23.CA7.TEXT1
IULC23.DB2.WORK
IULC23.DEMOED.ISPPROF
IULC23.IULC.RULES
IULC23.JCL.CNTL
IULC23.OUTPUT
IULC23.REXX.CLIST
IULC23.SAREXP.DELCARDS
IULC23.TEST.PDS
***
```

Write it and test it. Use TSO command: "listcat level("pmfID")"

Additional Program

- Write a REXX program to display a datasets volume name.

```
Please enter the dataset:  
crone90.crone.rexx  
The dataset is on volume :   DEMS03  
***
```

This course has been prepared by Milos Forman for MCoE needs only!

15 Copyright ©2006 CA. All rights reserved. All trademarks, trade names, services marks and logos referenced herein belong to their respective companies.

Write it and test it. Use TSO command: "listds"