News From the REXX Compiler

Klaus Hansjakob
IBM

News from the Compiler

REXX Symposium Boston May 1994

Dr. Klaus Hansjakob

HANSJAKO@VABVM1.VNET.IBM.COM
ATIBMCXP at IBMMAIL

IBM Vienna Software Development Laboratory
Lassallestrasse 1
A-1020 Vienna, Austria
Europe

(+431) 21145-4243

The information contained in this document has not been submitted to any formal IBM test and is distributed on an "As Is" basis without any warranty either expressed or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

In this document, any references made to an IBM licensed program are not intended to state or imply that only IBM's licensed program may be used; any functionally equivalent program may be used instead.

Any performance data contained in this document was determined in a controlled environment, and therefore the results which may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

It is possible that this material may contain references to, or information about IBM products (machines and programs), programming or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming or services in your country.

Agenda

- News from the REXX Compiler
- Packaging an application
  - General considerations
  - DLINK
  - Function packages

Compiler

IBM Compiler and Library for SAA
REXX/370
Release 2

5695-013 5695-014

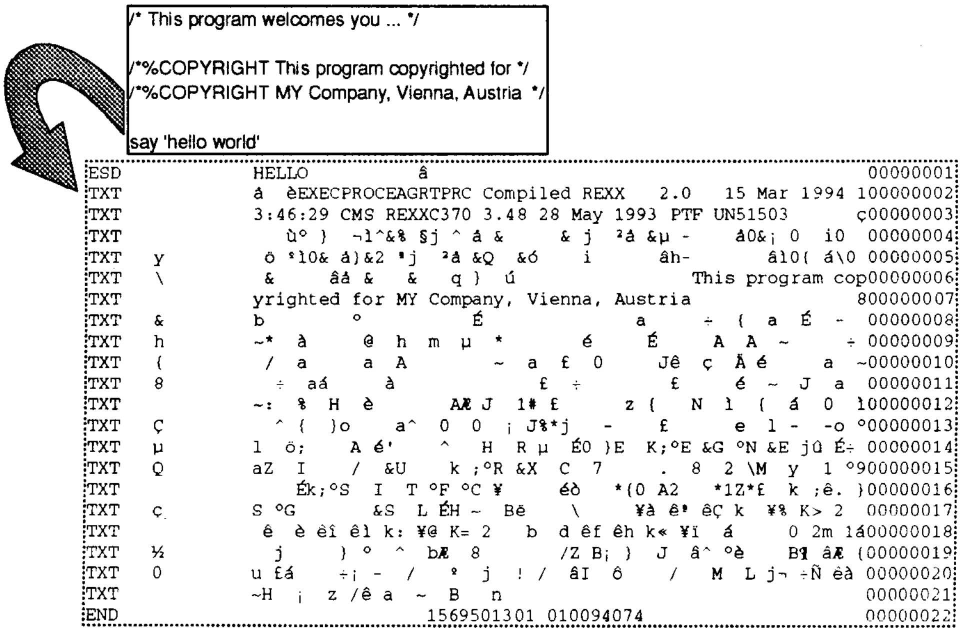
Available for CMS and MVS
Library is part of REXX/VSE

Alternate Library PTFs

	CMS		MVS	
	PTF	APAR	PTF	APAR
Compiler	UN51503 UN51504 JPN	PN48015	UN51833 UN51834 ENU UN51835 JPN	PN48006
Library	No PTF, additional product tape			

Introduction of Copyright, Alternate Library

Copyright



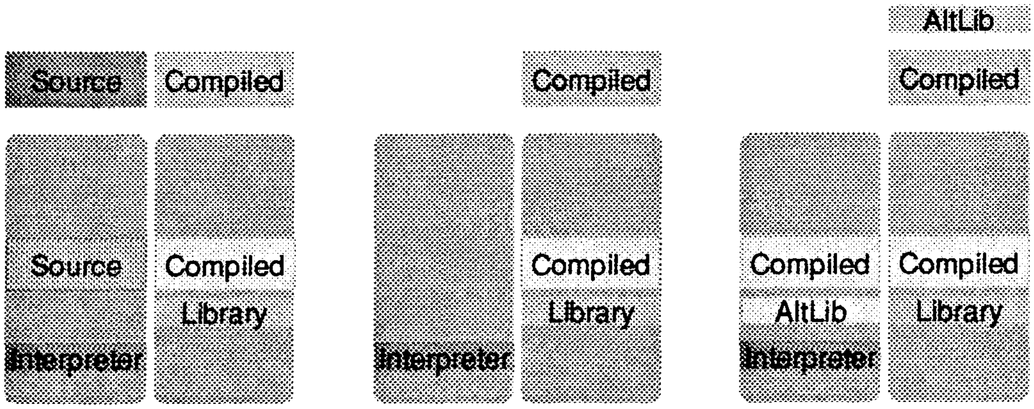
```
/* This program welcomes you ... */

/*%COPYRIGHT This program copyrighted for */
/*%COPYRIGHT MY Company, Vienna, Austria  */

say 'hello world'
```

```
ESD            HELLO          â                                      00000001
TXT            á  èEXECPROCEAGRTPRC Compiled REXX   2.0   15 Mar 1994 100000002
TXT            3:46:29 CMS REXXC370 3.48 28 May 1993 PTF UN51503       ç00000003
TXT               ù° }  ¬1^&% §j ^ á &     & j  ²á &µ -    á0&¡ 0  i0  00000004
TXT     y       ö °10& á)&2 ªj  ²á &Q  &ó    i      âh-      â10( á\0 00000005
TXT     \       &     âá &    &   q }  ú               This program cop00000006
TXT            yrighted for MY Company, Vienna, Austria               800000007
TXT     &      b        °         É          a     ÷  (  a  É  -  00000008
TXT     h      ~*  à     @  h  m  µ  *      é     É     A  A  ~      ÷ 00000009
TXT     {       /  a     a  A          ~  a  £  0     Jê  ç  Ä é     a  ~00000010
TXT     8         ÷  aá      à             £  ÷        £     é  ~  J  a  00000011
TXT            ~:  %  H  è         AÆ J  1# £       z {    N  ì  {  á  0  ì00000012
TXT     Ç          ^  {  }o     a^  0  0  ¡ J%*j   -      £       e  l  -  -o °00000013
TXT     µ      l  ö;    A é'    ^   H  R µ  É0 }E  K;°E &G °N &E jû É÷ 00000014
TXT     Q      aZ  I     /  &U    k ;°R &X  C  7      .  8  2 \M  y  1 °900000015
TXT                Ék;°S  I  T °F °C ¥      éð   *{0 A2   *1Z*£  k ;ê. }00000016
TXT     ç      S °G         &S  L ÉH ~  Bë    \      ¥à ê° êÇ k  ¥% K> 2  00000017
TXT             ê  è êî êl k: ¥@ K= 2   b  d êf êh k« ¥ï  á      0 2m 1á00000018
TXT     ½        j      } °  ^  bÆ  8       /Z B¡ }  J  â^ °è     B¶ âÆ {00000019
TXT     0      u £á     ÷¡  -   /    º  j  ! /  âI  ô      /    M  L j¬ ÷Ñ êà 00000020
TXT            ~H  ¡  z /ê a  ~  B   n                                   00000021
END                           1569501301 010094074                         00000022
```

Alternate Library

- Compile program
  - with ALTERNATE and SOURCELINE
    - use CONDENSE to hide source
    - DLINK does not work
- Distribute Alternate Library
  - without royalties, without paperwork
- Alternate Library
  - is installed on systems without Library
  - invokes interpreter when compiled program is run

Distributing REXX



Source Compiled

Source / Interpreter — Compiled / Library

Compiled

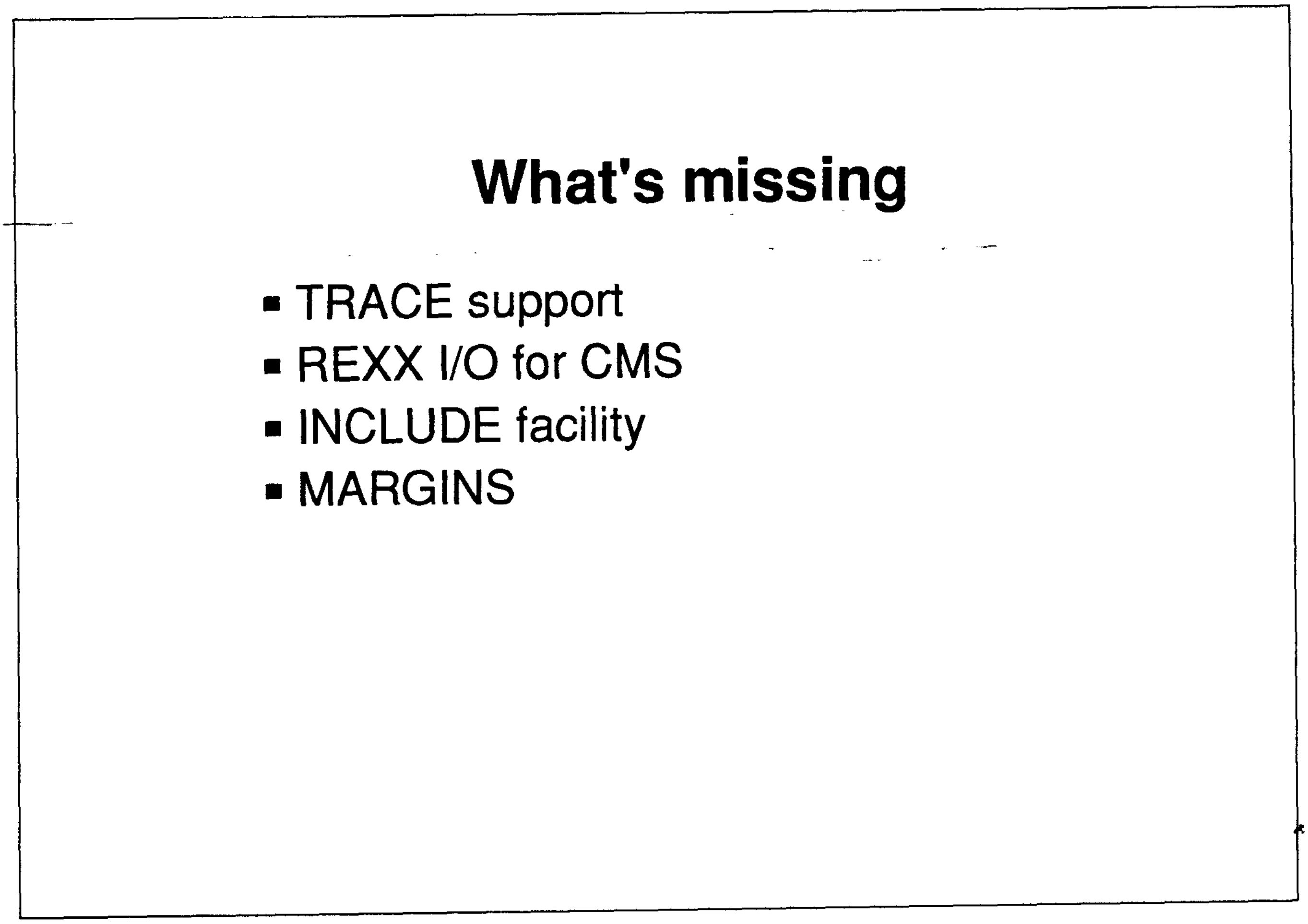
Interpreter — Compiled / Library

AltLib

Compiled

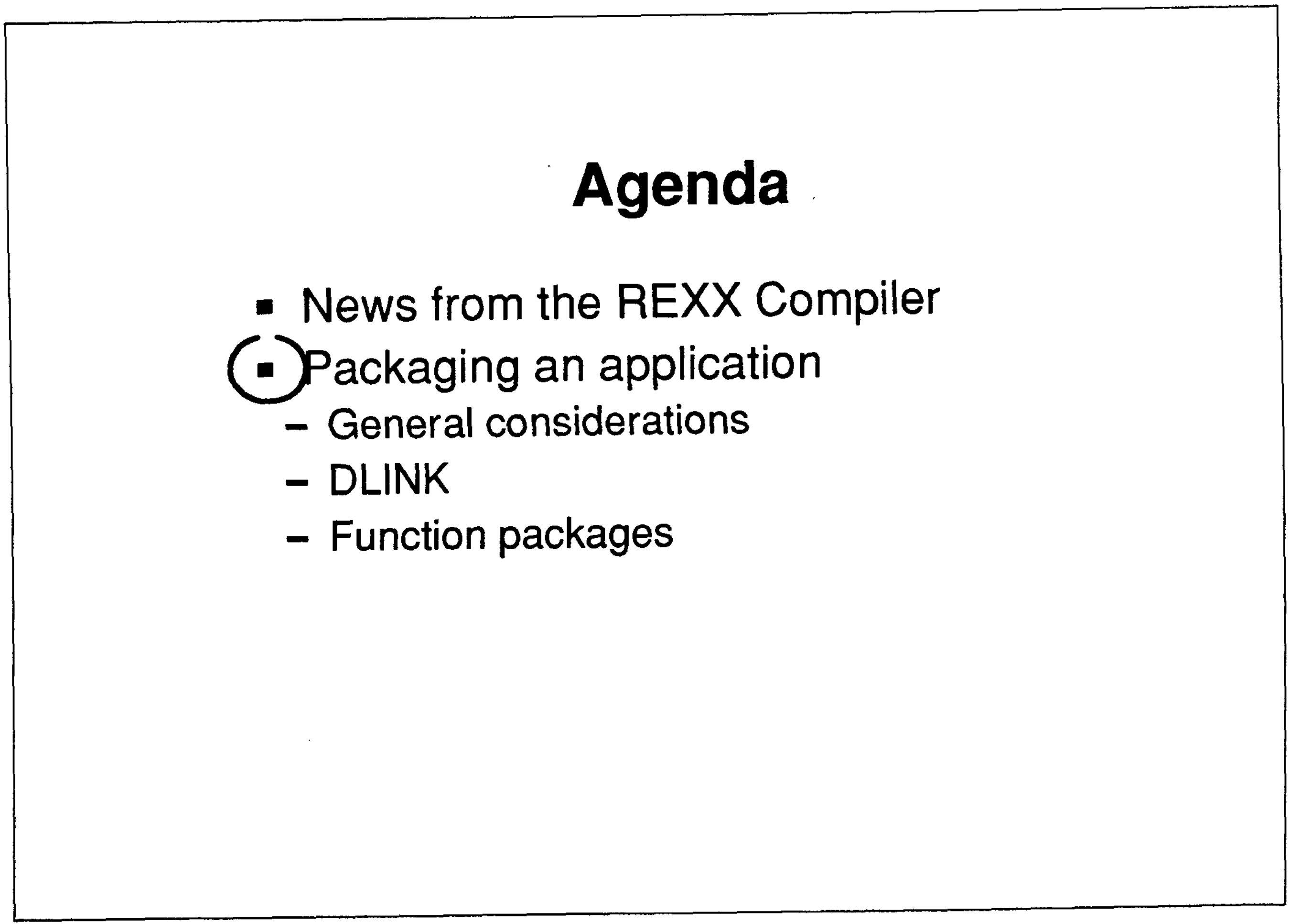
Compiled / AltLib / Interpreter — Compiled / Library

Library optional	required	optional
Source exposed	not exposed	hidden (CONDENSE)
Maintenance problems	no problems	no problems
Packaging few options	all options	some options



What's missing

- TRACE support
- REXX I/O for CMS
- INCLUDE facility
- MARGINS



Agenda

- News from the REXX Compiler
- Packaging an application
  - General considerations
  - DLINK
  - Function packages

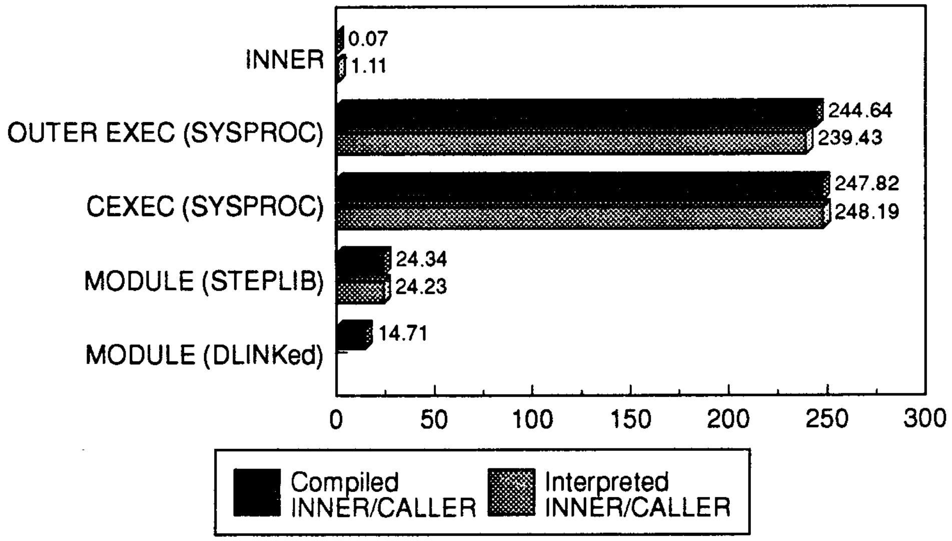
Performance

```
/* REXX Inner */
cvi=sysvar('SYSCPU')
Do i=1 To 1000
  Call inner
End
cve=sysvar('SYSCPU')
Say 'CPU time' cve-cvi
Exit

inner:Procedure
Return
```

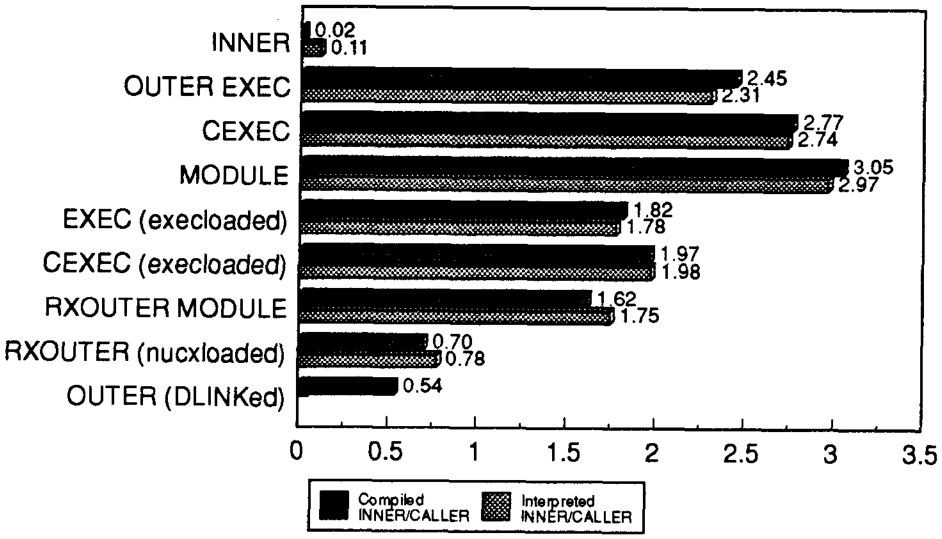
```
/* REXX Caller */
cvi=sysvar('SYSCPU')
Do i=1 to 1000
  Call outer
End
cve=sysvar('SYSCPU')
Say 'CPU time' cve-cvi
Exit
```

```
/* REXX Outer */
Return
```

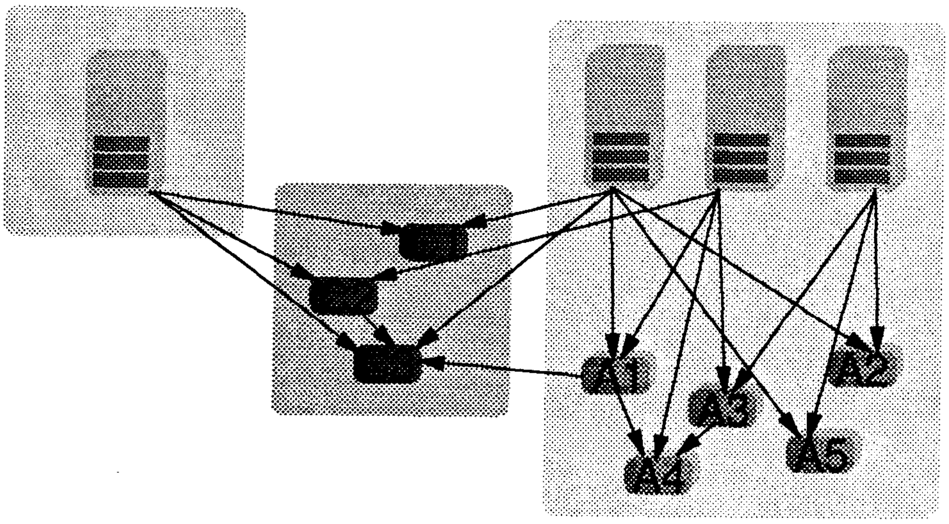
Performance (MVS)



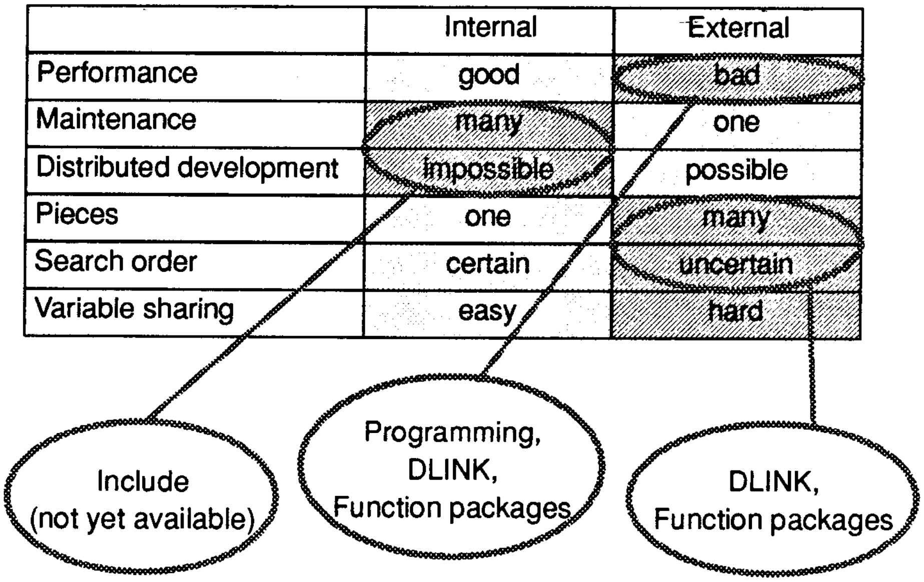
Performance (CMS)



Applications



Internal - External



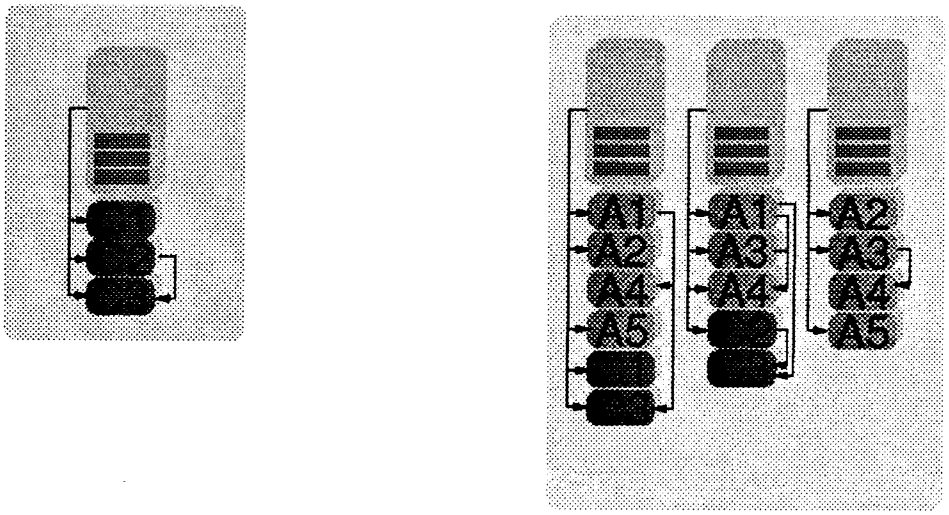
	Internal	External
Performance	good	bad
Maintenance	many	one
Distributed development	impossible	possible
Pieces	one	many
Search order	certain	uncertain
Variable sharing	easy	hard

Include
(not yet available)

Programming,
DLINK,
Function packages

DLINK,
Function packages

DLINKed Applications



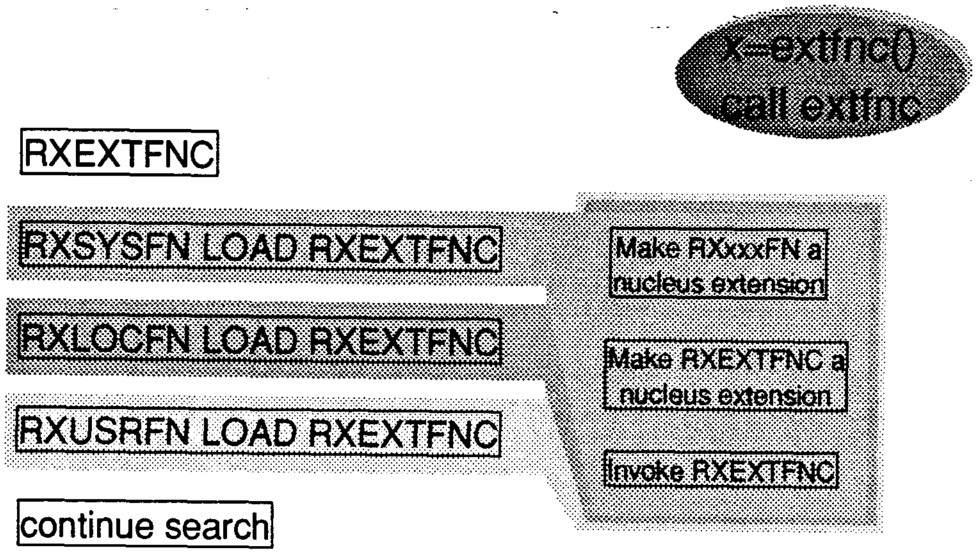
DLINK

- Search overhead zero
- Requires Compiler
- Does not work with Alternate Library

Function Packages

- Commonly used functions
- Early in the search order
- Functions must understand REXX function invocation

Search Order (CMS)



x=extfnc()
call extfnc

RXEXTFNC

RXSYSFN LOAD RXEXTFNC

RXLOCFN LOAD RXEXTFNC

RXUSRFN LOAD RXEXTFNC

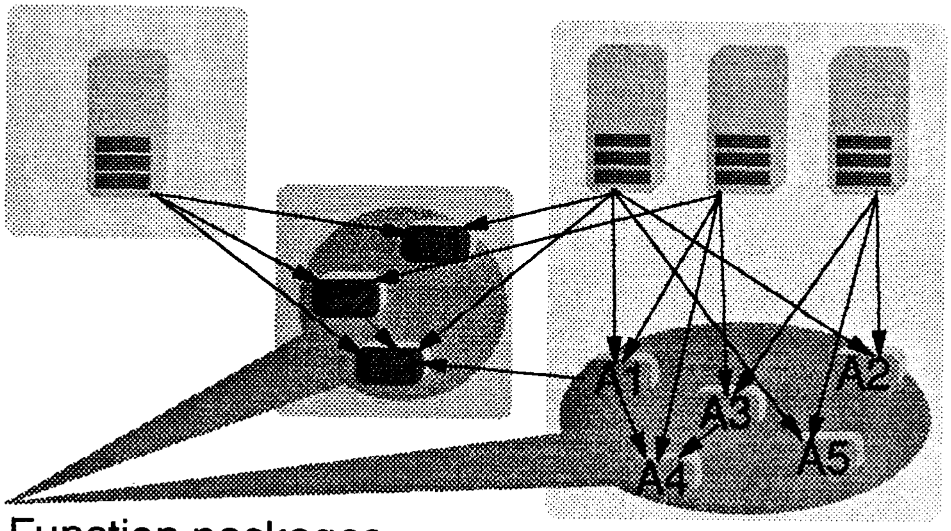
continue search

Make RXxxxFN a nucleus extension

Make RXEXTFNC a nucleus extension

Invoke RXEXTFNC

Applications with Function Packages



Function packages

Function Packages

- First in search order
- Compiler allows to write functions in REXX
- Works with Alternate Library
- May require explicit loading/unloading on CMS
- DLINK may be used when Alternate Library is not used

CMS Function Package Example

- Two files
  - RXUSERFN is function package loader
  - USERFN is function package glue code
- "Glue" code for a function package
  - Use without royalties
  - Allows free naming of function package
    - Requires renaming of files
    - Explicit loading of package and all functions with "RXmyname LOAD"
    - Explained in RXUSERFN header

Necessary Modifications

RXUSERFN ASSEMBLE

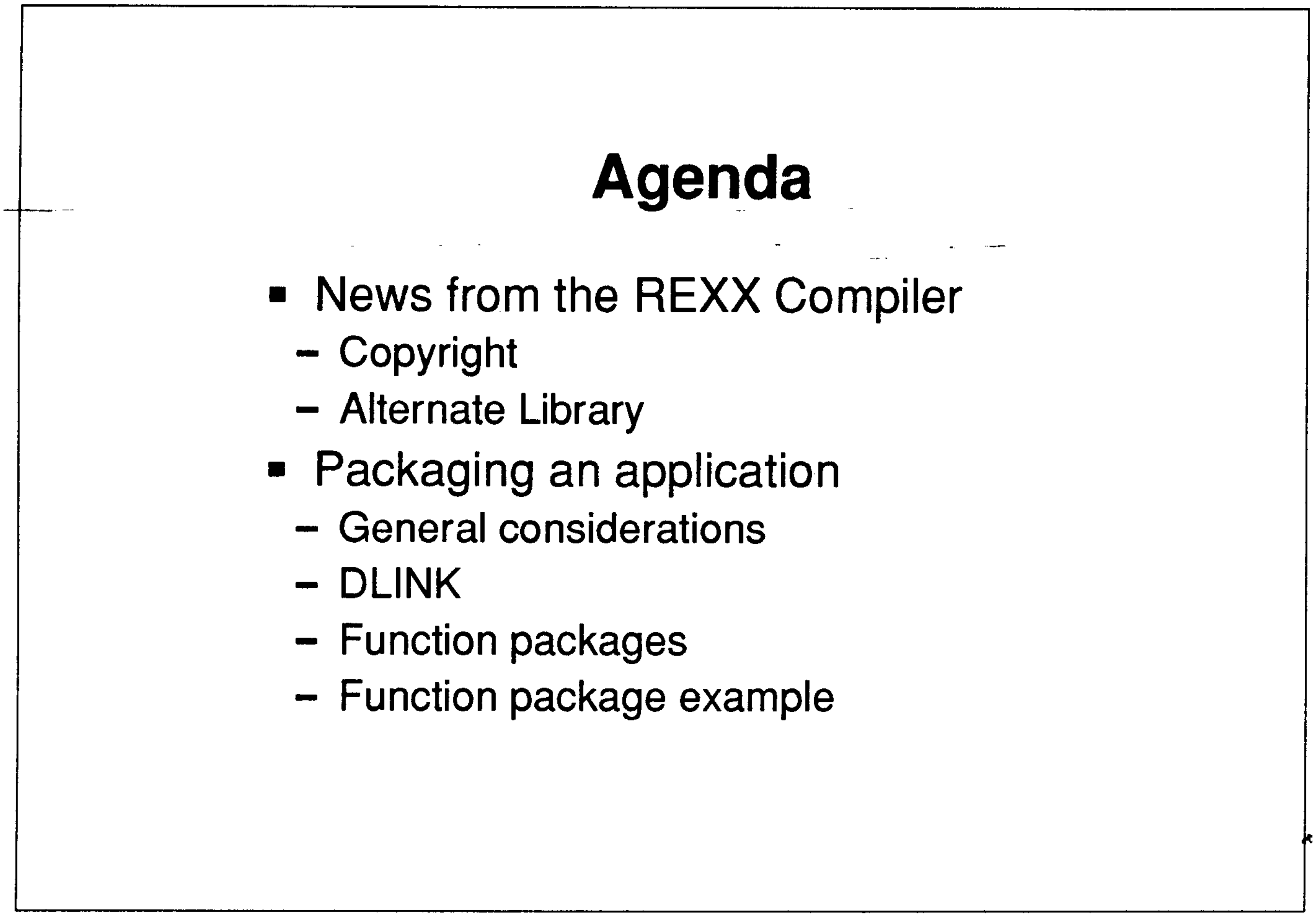
```
&PACKAGE   SETC  'USERFN'        Name of the package to load
&RXPACK    SETC  'RX&PACKAGE'    Name of this program
&CR(1)     SETC  'My Copyright'  Copyright notice
&CR(2)     SETC  'second line'   Copyright notice continued
```

USERFN ASSEMBLE

```
&PACKAGE   SETC  'USERFN'        Name of the package
&CR(1)     SETC  'My Copyright'  Copyright notice
&CR(2)     SETC  'second line'   Copyright notice continued
&FUN(1)    SETC  'USER1'         Name of function
&FUN(2)    SETC  'USER2'         Name of function
&FUN(3)    SETC  'USER3'         Name of function
```

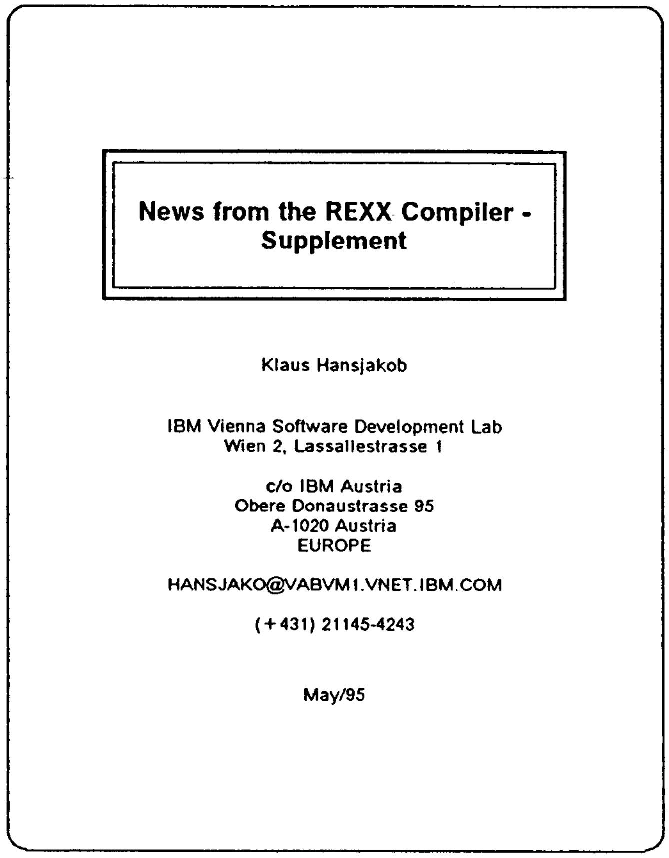
Obtain the Source Code

- Email - write a note to
  - hansjako@vabvm1.vnet.ibm.com
  - ATIBMCXP at IBMMAIL
- Disk - get one
  - as long as supply lasts
  - if you can't use email
  - if you have a way to upload code
- Supplement
  - type in



Agenda

- News from the REXX Compiler
  - Copyright
  - Alternate Library
- Packaging an application
  - General considerations
  - DLINK
  - Function packages
  - Function package example



News from the REXX Compiler - Supplement

Klaus Hansjakob

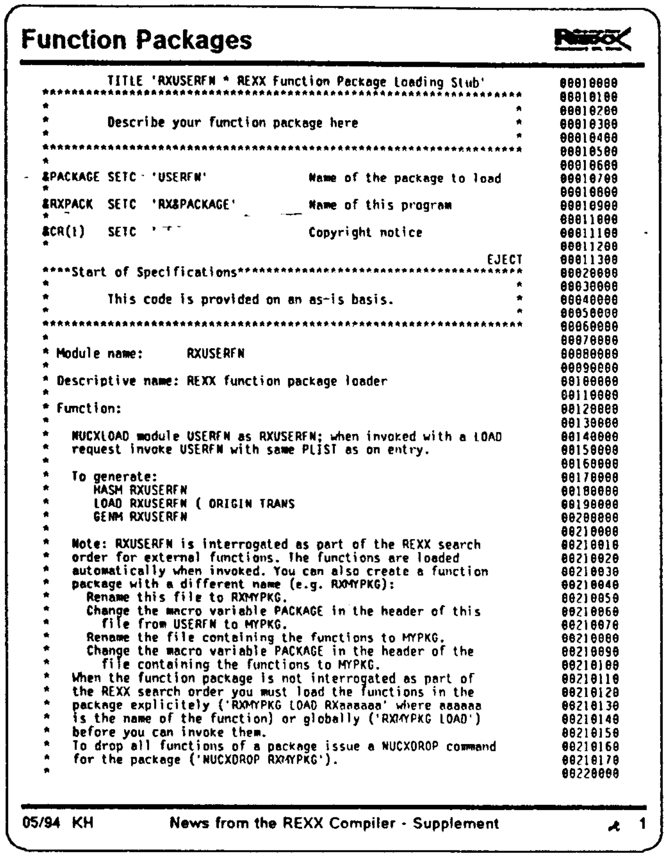
IBM Vienna Software Development Lab
Wien 2, Lassallestrasse 1

c/o IBM Austria
Obere Donaustrasse 95
A-1020 Austria
EUROPE

HANSJAKO@VABVM1.VNET.IBM.COM

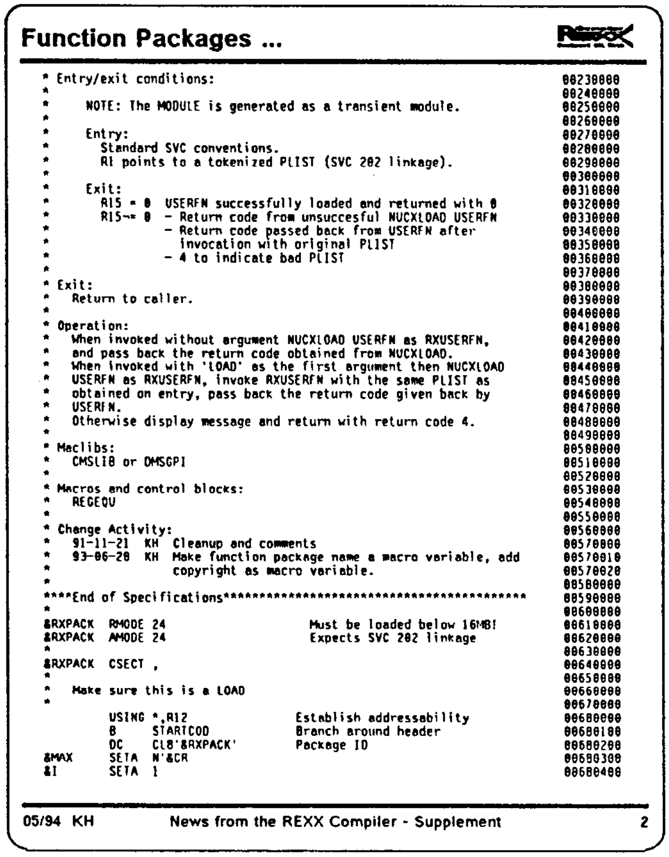
(+431) 21145-4243

May/95



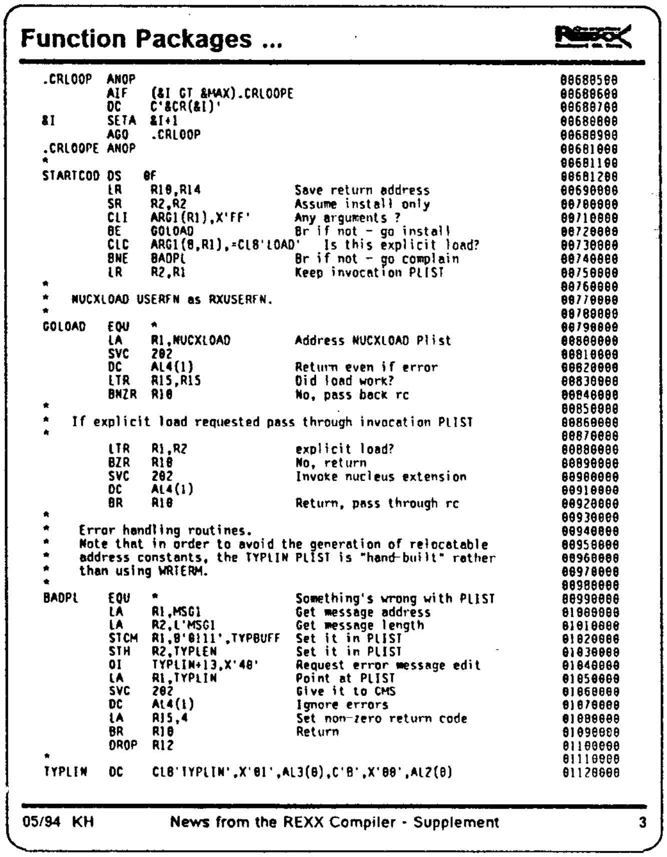
Function Packages

```
         TITLE 'RXUSERFN * REXX Function Package Loading Stub'
**********************************************************************
*                                                                    *
*        Describe your function package here                         *
*                                                                    *
**********************************************************************
*
&PACKAGE SETC  'USERFN'                   Name of the package to load
*
&RXPACK  SETC  'RX&PACKAGE'               Name of this program
*
&CR(1)   SETC  ' '                        Copyright notice
*
                                                                 EJECT
****Start of Specifications*******************************************
*                                                                    *
*        This code is provided on an as-is basis.                    *
*                                                                    *
**********************************************************************
*
* Module name:      RXUSERFN
*
* Descriptive name: REXX function package loader
*
* Function:
*
*   NUCXLOAD module USERFN as RXUSERFN; when invoked with a LOAD
*   request invoke USERFN with same PLIST as on entry.
*
*   To generate:
*      HASM RXUSERFN
*      LOAD RXUSERFN ( ORIGIN TRANS
*      GENM RXUSERFN
*
*   Note: RXUSERFN is interrogated as part of the REXX search
*   order for external functions. The functions are loaded
*   automatically when invoked. You can also create a function
*   package with a different name (e.g. RXMYPKG):
*     Rename this file to RXMYPKG.
*     Change the macro variable PACKAGE in the header of this
*       file from USERFN to MYPKG.
*     Rename the file containing the functions to MYPKG.
*     Change the macro variable PACKAGE in the header of the
*       file containing the functions to MYPKG.
*   When the function package is not interrogated as part of
*   the REXX search order you must load the functions in the
*   package explicitely ('RXMYPKG LOAD RXaaaaaa' where aaaaaa
*   is the name of the function) or globally ('RXMYPKG LOAD')
*   before you can invoke them.
*   To drop all functions of a package issue a NUCXDROP command
*   for the package ('NUCXDROP RXMYPKG').
*
```



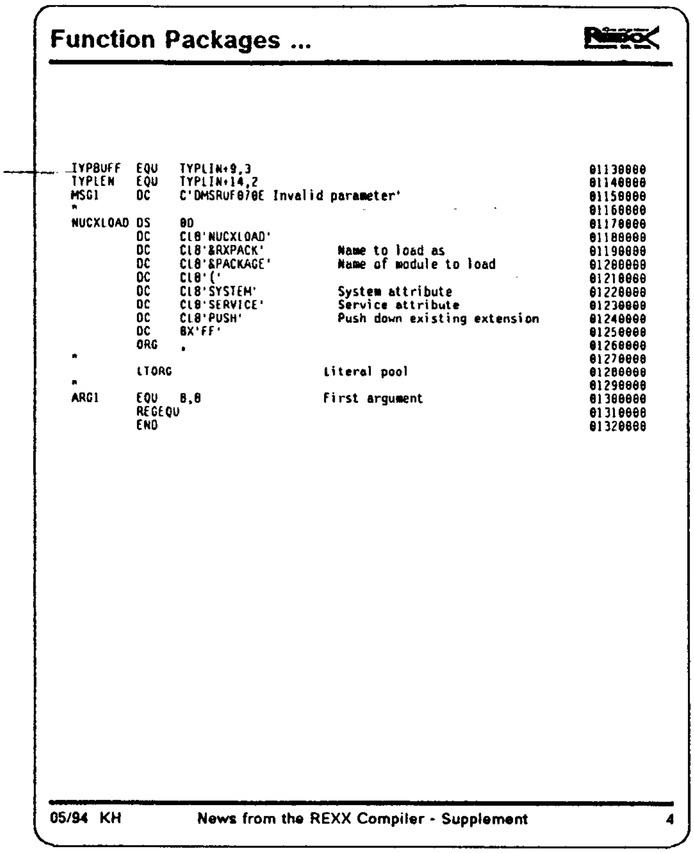
Function Packages ...

```
* Entry/exit conditions:
*
*     NOTE: The MODULE is generated as a transient module.
*
*     Entry:
*       Standard SVC conventions.
*       R1 points to a tokenized PLIST (SVC 202 linkage).
*
*     Exit:
*       R15 = 0  USERFN successfully loaded and returned with 0
*       R15¬= 0  - Return code from unsuccesful NUCXLOAD USERFN
*                - Return code passed back from USERFN after
*                  invocation with original PLIST
*                - 4 to indicate bad PLIST
*
* Exit:
*   Return to caller.
*
* Operation:
*   When invoked without argument NUCXLOAD USERFN as RXUSERFN,
*   and pass back the return code obtained from NUCXLOAD.
*   When invoked with 'LOAD' as the first argument then NUCXLOAD
*   USERFN as RXUSERFN, invoke RXUSERFN with the same PLIST as
*   obtained on entry, pass back the return code given back by
*   USERFN.
*   Otherwise display message and return with return code 4.
*
* Maclibs:
*   CMSLIB or DMSGPI
*
* Macros and control blocks:
*   REGEQU
*
* Change Activity:
*   91-11-21  KH  Cleanup and comments
*   93-06-20  KH  Make function package name a macro variable, add
*                 copyright as macro variable.
*
****End of Specifications*********************************************
*
&RXPACK  RMODE 24                         Must be loaded below 16MB!
&RXPACK  AMODE 24                         Expects SVC 202 linkage
*
&RXPACK  CSECT ,
*
*   Make sure this is a LOAD
*
         USING *,R12                Establish addressability
         B     STARTCOD             Branch around header
         DC    CL8'&RXPACK'         Package ID
&MAX     SETA  N'&CR
&I       SETA  1
```



Function Packages ...

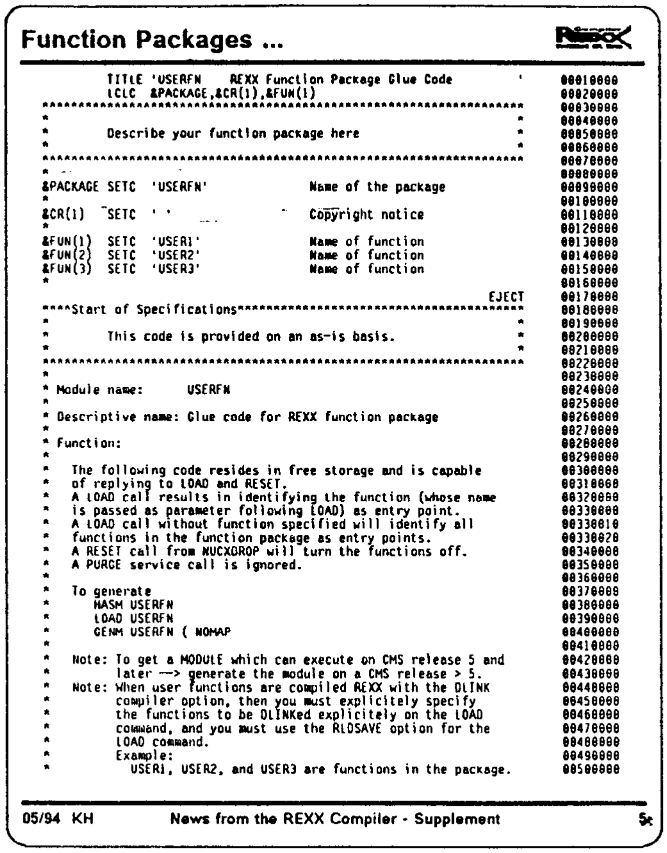
```
.CRLOOP  ANOP
         AIF   (&I GT &MAX).CRLOOPE
         DC    C'&CR(&I)'
&I       SETA  &I+1
         AGO   .CRLOOP
.CRLOOPE ANOP
*
STARTCOD DS    0F
         LR    R10,R14              Save return address
         SR    R2,R2                Assume install only
         CLI   ARG1(R1),X'FF'       Any arguments ?
         BE    GOLOAD               Br if not - go install
         CLC   ARG1(8,R1),=CL8'LOAD'   Is this explicit load?
         BNE   BADPL                Br if not - go complain
         LR    R2,R1                Keep invocation PLIST
*
*   NUCXLOAD USERFN as RXUSERFN.
*
GOLOAD   EQU   *
         LA    R1,NUCXLOAD          Address NUCXLOAD Plist
         SVC   202
         DC    AL4(1)               Return even if error
         LTR   R15,R15              Did load work?
         BNZR  R10                  No, pass back rc
*
*   If explicit load requested pass through invocation PLIST
*
         LTR   R1,R2                explicit load?
         BZR   R10                  No, return
         SVC   202                  Invoke nucleus extension
         DC    AL4(1)
         BR    R10                  Return, pass through rc
*
*    Error handling routines.
*    Note that in order to avoid the generation of relocatable
*    address constants, the TYPLIN PLIST is "hand-built" rather
*    than using WRTERM.
*
BADPL    EQU   *                    Something's wrong with PLIST
         LA    R1,MSG1              Get message address
         LA    R2,L'MSG1            Get message length
         STCM  R1,B'0111',TYPBUFF   Set it in PLIST
         STH   R2,TYPLEN            Set it in PLIST
         OI    TYPLIN+13,X'40'      Request error message edit
         LA    R1,TYPLIN            Point at PLIST
         SVC   202                  Give it to CMS
         DC    AL4(1)               Ignore errors
         LA    R15,4                Set non-zero return code
         BR    R10                  Return
         DROP  R12
*
TYPLIN   DC    CL8'TYPLIN',X'01',AL3(0),C'B',X'00',AL2(0)
```



Function Packages ...

```
IYPBUFF  EQU   TYPLIN+9,3
TYPLEN   EQU   TYPLIN+14,2
MSG1     DC    C'DMSRUF070E Invalid parameter'
*
NUCXLOAD DS    0D
         DC    CL8'NUCXLOAD'
         DC    CL8'&RXPACK'             Name to load as
         DC    CL8'&PACKAGE'            Name of module to load
         DC    CL8'('
         DC    CL8'SYSTEM'              System attribute
         DC    CL8'SERVICE'             Service attribute
         DC    CL8'PUSH'                Push down existing extension
         DC    8X'FF'
         ORG   ,
*
         LTORG                        Literal pool
*
ARG1     EQU   8,8                    First argument
         REGEQU
         END
```

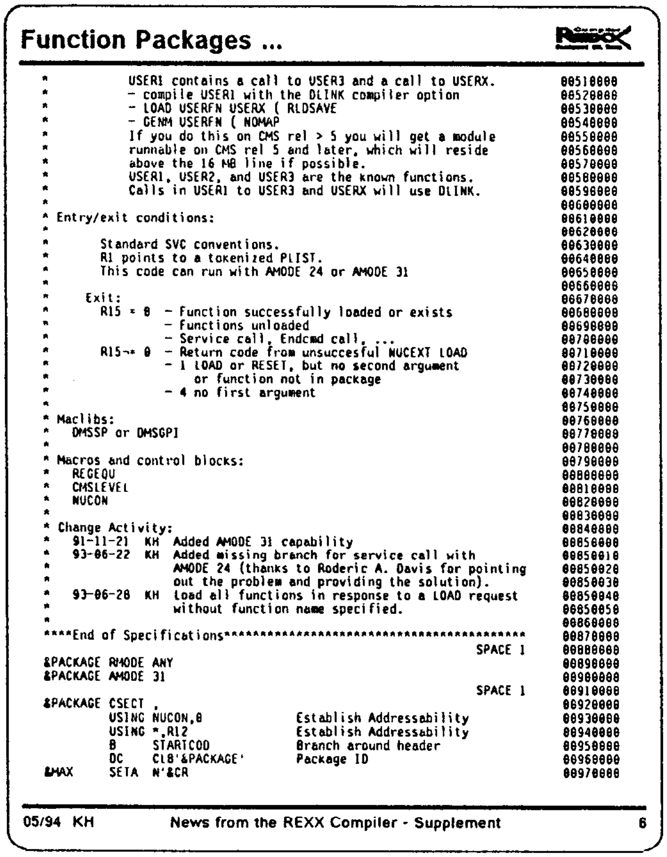
05/94 KH News from the REXX Compiler - Supplement 4



Function Packages ...

```
         TITLE 'USERFN    REXX Function Package Glue Code              '
         LCLC  &PACKAGE,&CR(1),&FUN(1)
***********************************************************************
*                                                                     *
*        Describe your function package here                         *
*                                                                     *
***********************************************************************
*
&PACKAGE SETC  'USERFN'                 Name of the package
*
&CR(1)   SETC  ' '                      Copyright notice
*
&FUN(1)  SETC  'USER1'                  Name of function
&FUN(2)  SETC  'USER2'                  Name of function
&FUN(3)  SETC  'USER3'                  Name of function
*
                                                                  EJECT
****Start of Specifications********************************************
*                                                                     *
*        This code is provided on an as-is basis.                    *
*                                                                     *
***********************************************************************
*
* Module name:      USERFN
*
* Descriptive name: Glue code for REXX function package
*
* Function:
*
*   The following code resides in free storage and is capable
*   of replying to LOAD and RESET.
*   A LOAD call results in identifying the function (whose name
*   is passed as parameter following LOAD) as entry point.
*   A LOAD call without function specified will identify all
*   functions in the function package as entry points.
*   A RESET call from NUCXDROP will turn the functions off.
*   A PURGE service call is ignored.
*
*   To generate
*      HASM USERFN
*      LOAD USERFN
*      GENM USERFN ( NOMAP
*
*   Note: To get a MODULE which can execute on CMS release 5 and
*         later --> generate the module on a CMS release > 5.
*   Note: When user functions are compiled REXX with the DLINK
*         compiler option, then you must explicitely specify
*         the functions to be DLINKed explicitely on the LOAD
*         command, and you must use the RLDSAVE option for the
*         LOAD command.
*         Example:
*           USER1, USER2, and USER3 are functions in the package.
```

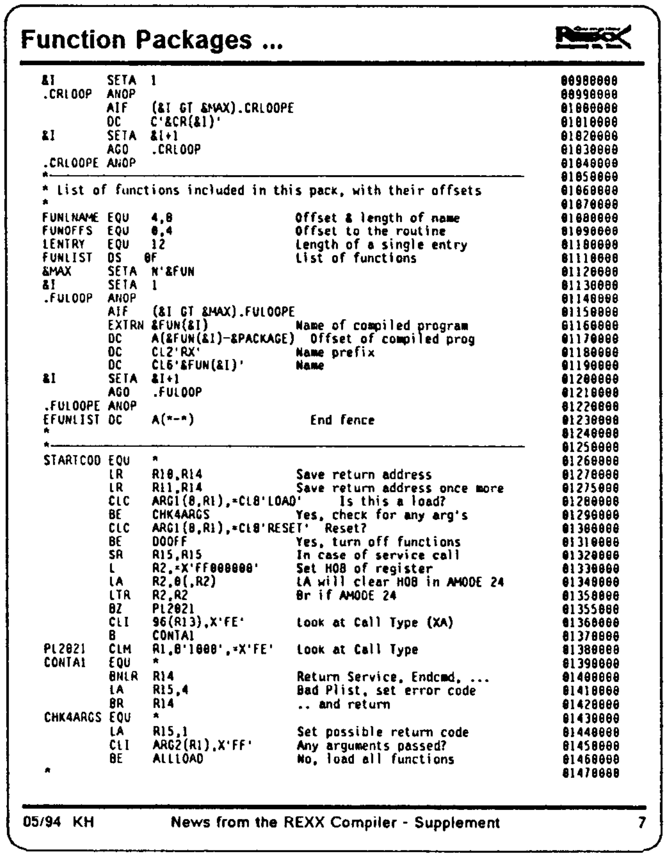
05/94 KH News from the REXX Compiler - Supplement 5



Function Packages ...

```
*          USER1 contains a call to USER3 and a call to USERX.
*          - compile USER1 with the DLINK compiler option
*          - LOAD USERFN USERX ( RLDSAVE
*          - GENM USERFN ( NOMAP
*          If you do this on CMS rel > 5 you will get a module
*          runnable on CMS rel 5 and later, which will reside
*          above the 16 MB line if possible.
*          USER1, USER2, and USER3 are the known functions.
*          Calls in USER1 to USER3 and USERX will use DLINK.
*
* Entry/exit conditions:
*
*      Standard SVC conventions.
*      R1 points to a tokenized PLIST.
*      This code can run with AMODE 24 or AMODE 31
*
*     Exit:
*      R15 = 0  - Function successfully loaded or exists
*               - Functions unloaded
*               - Service call, Endcmd call, ...
*      R15¬= 0  - Return code from unsuccesful NUCEXT LOAD
*               - 1 LOAD or RESET, but no second argument
*                   or function not in package
*               - 4 no first argument
*
* Maclibs:
*   DMSSP or DMSGPI
*
* Macros and control blocks:
*   REGEQU
*   CMSLEVEL
*   NUCON
*
* Change Activity:
*   91-11-21  KH  Added AMODE 31 capability
*   93-06-22  KH  Added missing branch for service call with
*                 AMODE 24 (thanks to Roderic A. Davis for pointing
*                 out the problem and providing the solution).
*   93-06-28  KH  Load all functions in response to a LOAD request
*                 without function name specified.
*
****End of Specifications**********************************************
                                                                SPACE 1
&PACKAGE RMODE ANY
&PACKAGE AMODE 31
                                                                SPACE 1
&PACKAGE CSECT ,
         USING NUCON,0              Establish Addressability
         USING *,R12                Establish Addressability
         B     STARTCOD             Branch around header
         DC    CL8'&PACKAGE'        Package ID
&MAX     SETA  N'&CR
```

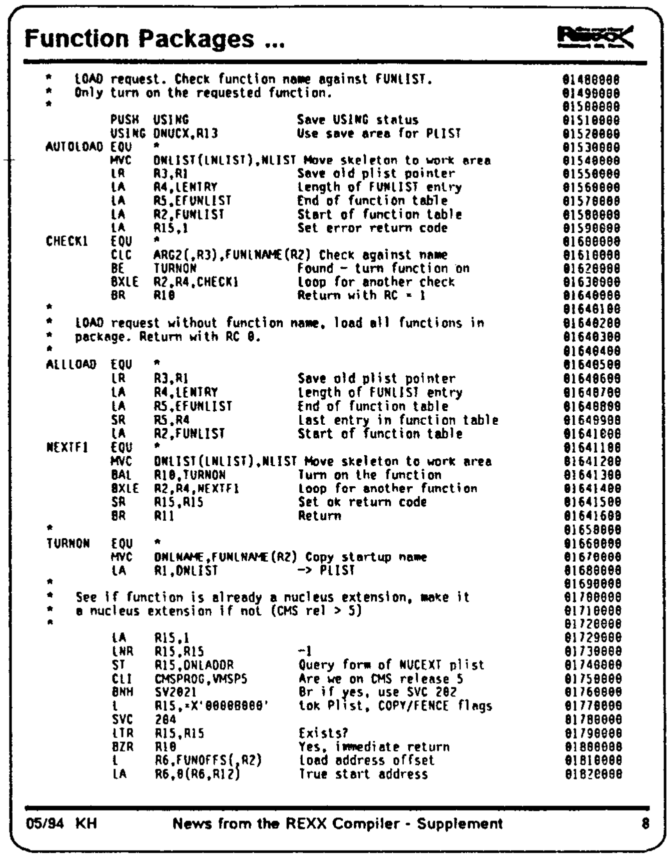
05/94 KH News from the REXX Compiler - Supplement 6



Function Packages ...

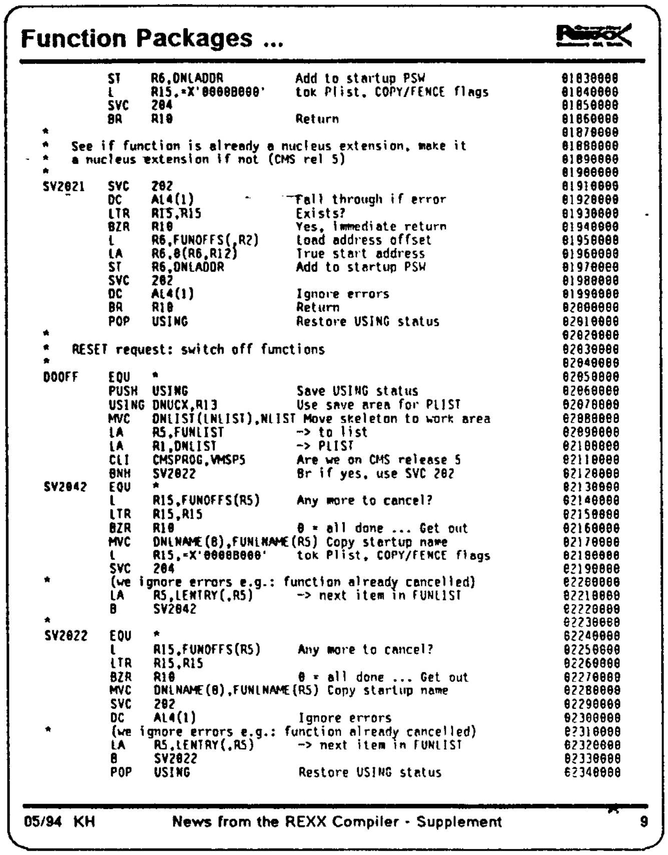
```
&I       SETA  1
.CRLOOP  ANOP
         AIF   (&I GT &MAX).CRLOOPE
         DC    C'&CR(&I)'
&I       SETA  &I+1
         AGO   .CRLOOP
.CRLOOPE ANOP
*----------------------------------------------------------------------
* List of functions included in this pack, with their offsets
*
FUNLNAME EQU   4,8                    Offset & length of name
FUNOFFS  EQU   0,4                    Offset to the routine
LENTRY   EQU   12                     Length of a single entry
FUNLIST  DS   0F                      List of functions
&MAX     SETA  N'&FUN
&I       SETA  1
.FULOOP  ANOP
         AIF   (&I GT &MAX).FULOOPE
         EXTRN &FUN(&I)               Name of compiled program
         DC    A(&FUN(&I)-&PACKAGE)  Offset of compiled prog
         DC    CL2'RX'                Name prefix
         DC    CL6'&FUN(&I)'          Name
&I       SETA  &I+1
         AGO   .FULOOP
.FULOOPE ANOP
EFUNLIST DC    A(*-*)                   End fence
*
*----------------------------------------------------------------------
STARTCOD EQU   *
         LR    R10,R14                Save return address
         LR    R11,R14                Save return address once more
         CLC   ARG1(8,R1),=CL8'LOAD'      Is this a load?
         BE    CHK4ARGS               Yes, check for any arg's
         CLC   ARG1(8,R1),=CL8'RESET'  Reset?
         BE    DOOFF                  Yes, turn off functions
         SR    R15,R15                In case of service call
         L     R2,=X'FF000000'        Set HOB of register
         LA    R2,0(,R2)              LA will clear HOB in AMODE 24
         LTR   R2,R2                  Br if AMODE 24
         BZ    PL2021
         CLI   96(R13),X'FE'          Look at Call Type (XA)
         B     CONTA1
PL2021   CLM   R1,B'1000',=X'FE'      Look at Call Type
CONTA1   EQU   *
         BNLR  R14                    Return Service, Endcmd, ...
         LA    R15,4                  Bad Plist, set error code
         BR    R14                    .. and return
CHK4ARGS EQU   *
         LA    R15,1                  Set possible return code
         CLI   ARG2(R1),X'FF'         Any arguments passed?
         BE    ALLLOAD                No, load all functions
*
```

05/94 KH News from the REXX Compiler - Supplement 7



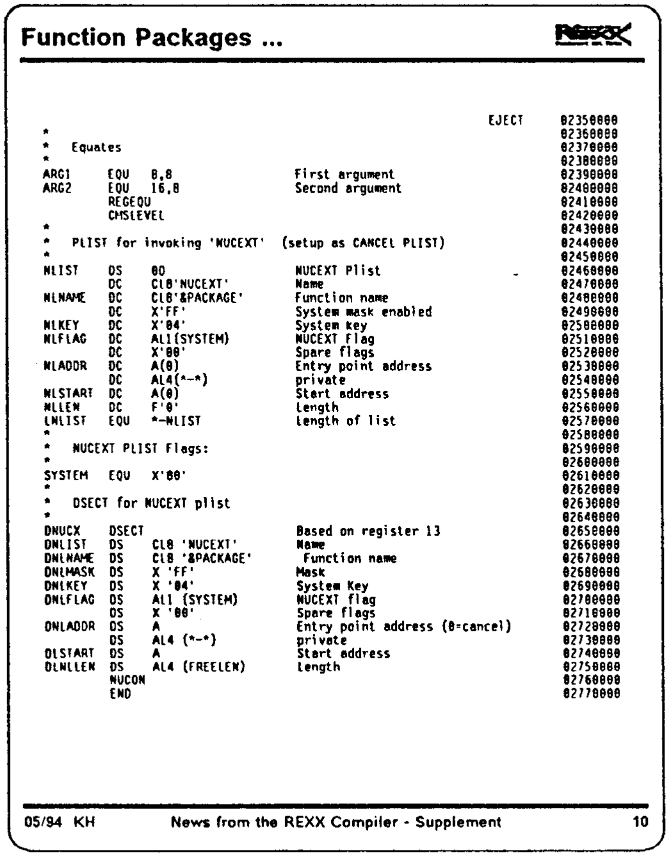
Function Packages ...

```
*   LOAD request. Check function name against FUNLIST.
*   Only turn on the requested function.
*
         PUSH  USING                  Save USING status
         USING DNUCX,R13              Use save area for PLIST
AUTOLOAD EQU   *
         MVC   DNLIST(LNLIST),NLIST Move skeleton to work area
         LR    R3,R1                  Save old plist pointer
         LA    R4,LENTRY              Length of FUNLIST entry
         LA    R5,EFUNLIST            End of function table
         LA    R2,FUNLIST             Start of function table
         LA    R15,1                  Set error return code
CHECK1   EQU   *
         CLC   ARG2(,R3),FUNLNAME(R2) Check against name
         BE    TURNON                 Found - turn function on
         BXLE  R2,R4,CHECK1           Loop for another check
         BR    R10                    Return with RC = 1
*
*   LOAD request without function name, load all functions in
*   package. Return with RC 0.
*
ALLLOAD  EQU   *
         LR    R3,R1                  Save old plist pointer
         LA    R4,LENTRY              Length of FUNLIST entry
         LA    R5,EFUNLIST            End of function table
         SR    R5,R4                  Last entry in function table
         LA    R2,FUNLIST             Start of function table
NEXTF1   EQU   *
         MVC   DNLIST(LNLIST),NLIST Move skeleton to work area
         BAL   R10,TURNON             Turn on the function
         BXLE  R2,R4,NEXTF1           Loop for another function
         SR    R15,R15                Set ok return code
         BR    R11                    Return
*
TURNON   EQU   *
         MVC   DNLNAME,FUNLNAME(R2) Copy startup name
         LA    R1,DNLIST              -> PLIST
*
*   See if function is already a nucleus extension, make it
*   a nucleus extension if not (CMS rel > 5)
*
         LA    R15,1
         LNR   R15,R15                -1
         ST    R15,DNLADDR            Query form of NUCEXT plist
         CLI   CMSPROG,VMSP5          Are we on CMS release 5
         BNH   SV2021                 Br if yes, use SVC 202
         L     R15,=X'00000000'       tok Plist, COPY/FENCE flags
         SVC   204
         LTR   R15,R15                Exists?
         BZR   R10                    Yes, immediate return
         L     R6,FUNOFFS(,R2)        Load address offset
         LA    R6,0(R6,R12)           True start address
```



Function Packages ...

```
         ST    R6,DNLADDR             Add to startup PSW
         L     R15,=X'00000000'       tok Plist, COPY/FENCE flags
         SVC   204
         BR    R10                    Return
*
*   See if function is already a nucleus extension, make it
*   a nucleus extension if not (CMS rel 5)
*
SV2021   SVC   202
         DC    AL4(1)                 Fall through if error
         LTR   R15,R15                Exists?
         BZR   R10                    Yes, immediate return
         L     R6,FUNOFFS(,R2)        Load address offset
         LA    R6,0(R6,R12)           True start address
         ST    R6,DNLADDR             Add to startup PSW
         SVC   202
         DC    AL4(1)                 Ignore errors
         BR    R10                    Return
         POP   USING                  Restore USING status
*
*   RESET request: switch off functions
*
DOOFF    EQU   *
         PUSH  USING                  Save USING status
         USING DNUCX,R13              Use save area for PLIST
         MVC   DNLIST(LNLIST),NLIST Move skeleton to work area
         LA    R5,FUNLIST             -> to list
         LA    R1,DNLIST              -> PLIST
         CLI   CMSPROG,VMSP5          Are we on CMS release 5
         BNH   SV2022                 Br if yes, use SVC 202
SV2042   EQU   *
         L     R15,FUNOFFS(R5)        Any more to cancel?
         LTR   R15,R15
         BZR   R10                    0 = all done ... Get out
         MVC   DNLNAME(8),FUNLNAME(R5) Copy startup name
         L     R15,=X'0000B000'       tok Plist, COPY/FENCE flags
         SVC   204
*        (we ignore errors e.g.: function already cancelled)
         LA    R5,LENTRY(,R5)         -> next item in FUNLIST
         B     SV2042
*
SV2022   EQU   *
         L     R15,FUNOFFS(R5)        Any more to cancel?
         LTR   R15,R15
         BZR   R10                    0 = all done ... Get out
         MVC   DNLNAME(8),FUNLNAME(R5) Copy startup name
         SVC   202
         DC    AL4(1)                 Ignore errors
*        (we ignore errors e.g.: function already cancelled)
         LA    R5,LENTRY(,R5)         -> next item in FUNLIST
         B     SV2022
         POP   USING                  Restore USING status
```



Function Packages ...

```
                                                     EJECT
*
*   Equates
*
ARG1     EQU   0,8                    First argument
ARG2     EQU   16,8                   Second argument
         REGEQU
         CMSLEVEL
*
*   PLIST for invoking 'NUCEXT'  (setup as CANCEL PLIST)
*
NLIST    DS    0D                     NUCEXT Plist
         DC    CL8'NUCEXT'            Name
NLNAME   DC    CL8'&PACKAGE'          Function name
         DC    X'FF'                  System mask enabled
NLKEY    DC    X'04'                  System key
NLFLAG   DC    AL1(SYSTEM)            NUCEXT Flag
         DC    X'00'                  Spare flags
NLADDR   DC    A(0)                   Entry point address
         DC    AL4(*-*)               private
NLSTART  DC    A(0)                   Start address
NLLEN    DC    F'0'                   Length
LNLIST   EQU   *-NLIST                Length of list
*
*   NUCEXT PLIST Flags:
*
SYSTEM   EQU   X'80'
*
*   DSECT for NUCEXT plist
*
DNUCX    DSECT                        Based on register 13
DNLIST   DS    CL8 'NUCEXT'           Name
DNLNAME  DS    CL8 '&PACKAGE'          Function name
DNLMASK  DS    X 'FF'                 Mask
DNLKEY   DS    X '04'                 System Key
DNLFLAG  DS    AL1 (SYSTEM)           NUCEXT flag
         DS    X '00'                 Spare flags
DNLADDR  DS    A                      Entry point address (0=cancel)
         DS    AL4 (*-*)              private
DLSTART  DS    A                      Start address
DLNLLEN  DS    AL4 (FREELEN)          Length
         NUCON
         END
```