

# Introduction to JES2 Table Pairs



SHARE 97, Session 2665

Tuesday, July 24, 2001

Permission is granted to SHARE Inc. to publish  
this presentation in the SHARE proceedings. IBM  
retains its right to distribute copies of this  
presentation to whomever it chooses.

**Chip Wood**  
**JES2 Design/Development/Service**  
**Poughkeepsie, NY**



**chipwood@us.ibm.com**

# Agenda

---



- What are table pairs?
- Extending JES2 commands and initialization statements with \$SCANTAB
  - Job commands
  - Output commands
- Extending the \$JQE using \$BERTTAB
- Other tables

## Disclaimers



- From [JES2 Installation Exits](#), page 1:

### Caution!

Defining exits and writing installation exit routines is intended to be accomplished by experienced system programmers; the reader is assumed to have knowledge of JES2.

If you want to customize JES2, IBM recommends that you use JES2 installation exits to accomplish the task. **IBM does not recommend or support alteration of JES2 source code.**

- From [JES2 Diagnosis](#):

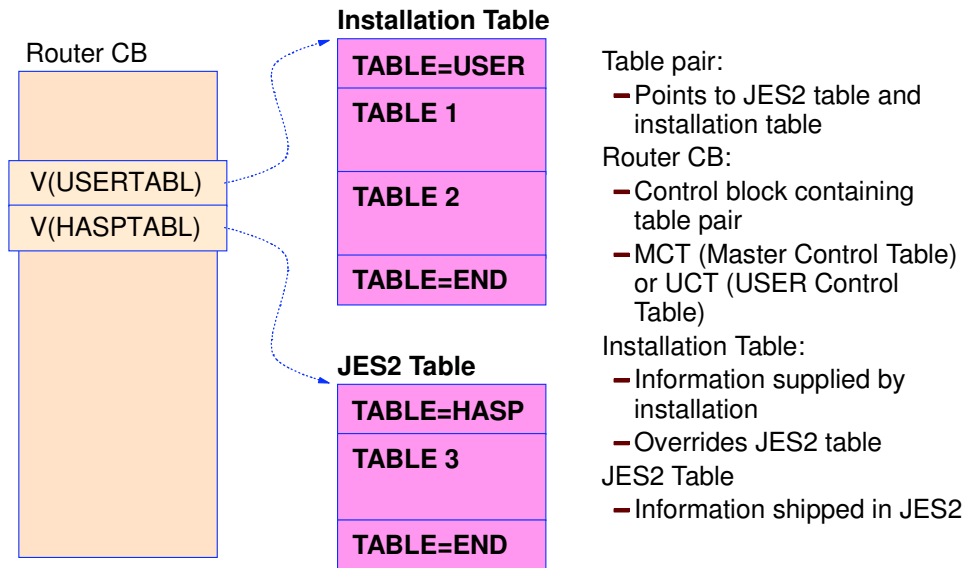
**CAUTION: IBM does not recommend or support modifications to JES2 source code. If you assume the risk of modifying JES2, then also assure that your modifications do not impact JES2 serviceability using IPCS. Otherwise, LEVEL2 may not be able to read JES2 dumps for problems unrelated to the modifications.**

## What are table pairs?

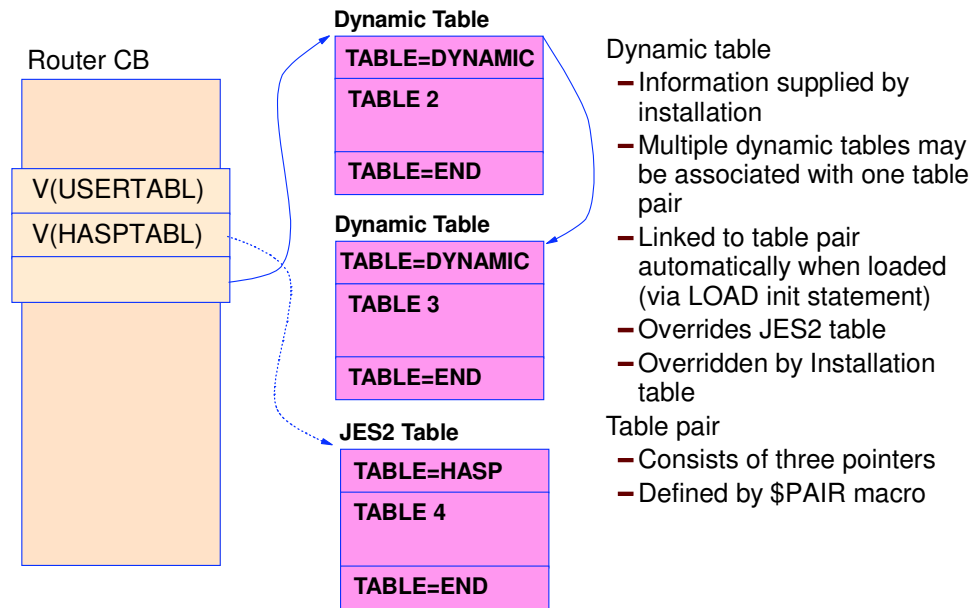


- Provide a mechanism to:
  - Add installation processing/function
  - Modify JES2 processing/function
- Involves:
  - Installation written tables/routines
  - Less detailed knowledge of JES2 code and control blocks
- **DOES NOT** replace need for exits
  - [Session 2664, Wed. 4:30](#)
- Main exploiter - \$SCANTAB
  - Also \$PCETAB, \$DTETAB, \$DCTTAB, \$WSTAB, \$TIDTAB, \$BERTTAB

# What are table pairs?



# Extended table pairs (OW32032)



## Extended table pairs



- \$PCETAB TABLE=DYNAMIC
- \$DCTTAB TABLE=DYNAMIC
- \$DTETAB TABLE=DYNAMIC
- \$TIDTAB TABLE=DYNAMIC
- \$BERTTAB TABLE=DYNAMIC
- \$SCANTAB TABLE=(DYNAMIC, pair-name, pair-name, ...)
- \$WSTAB TABLE=(DYNAMIC, pair-name, pair-name, ...)

Extending JES2 commands and  
initialization statements with

**\$SCANTABs**





## What are \$SCANTABs



- \$SCANTABs:
  - Define initialization statements and keywords
  - Define commands and keywords
    - Syntax rules
    - Fields to be displayed/modified/filtered
- Parsing command or statement handled automatically by \$SCAN facility
- Data conversion handled automatically by \$SCAN facility
  - EBCDIC to numeric, hexadecimal, or flag settings (SET, FILTER)
  - Numeric, hexadecimal, or flag settings to EBCDIC (DISPLAY)

## Example



- Add a new filter, MINUTES, to job and output commands, similar to the existing HOURS and DAYS filters
- Start with HOURS and DAYS \$SCANTABs and modify as necessary

# Existing tables



```
$SCANTAB NAME=HOURS, MINLEN=1,  
        CB=PARENT, DSECT=JQE,  
        FILTER=(YES, ALWAYS, GTLT) ,  
        PRESCAN=(PREFOAGE, DISPLAY, FILTER) , DISPALL=NO,  
        FIELD=JQXCRTME, CONV=(NUM, , 60),  
        CALLERS=($SCDCMDS, $SCSCMDS, $SCECMDS, $SCHCMDS,  
        $SCRLCMD, $SCCCMDS, $SCPCMDS, $SCDOCMD) ,  
        RANGE=(0, X' 7FFFFFFF' )  
  
$SCANTAB NAME=DAYS, MINLEN=2,  
        CB=PARENT, DSECT=JQE,  
        FILTER=(YES, ALWAYS, GTLT) ,  
        PRESCAN=(PREFOAGE, DISPLAY, FILTER) , DISPALL=NO,  
        FIELD=JQXCRTME, CONV=(NUM, , 60*24),  
        CALLERS=($SCDCMDS, $SCSCMDS, $SCECMDS, $SCHCMDS,  
        $SCRLCMD, $SCCCMDS, $SCPCMDS, $SCDOCMD) ,  
        RANGE=(0, X' 7FFFFFFF' )
```

## New table



- In HOURS/DAYS, CONV=NUM multiplier converts input value to minutes
  - PREFOAGE prescan routine converts input time to age in minutes and compares with input value (adjusted by multiplier)
    - ▶ Routine is called only for DISPLAYs and FILTERs
    - ▶ SET calls disallowed by other parameters
  - For MINUTES, use same table with no multiplier

```

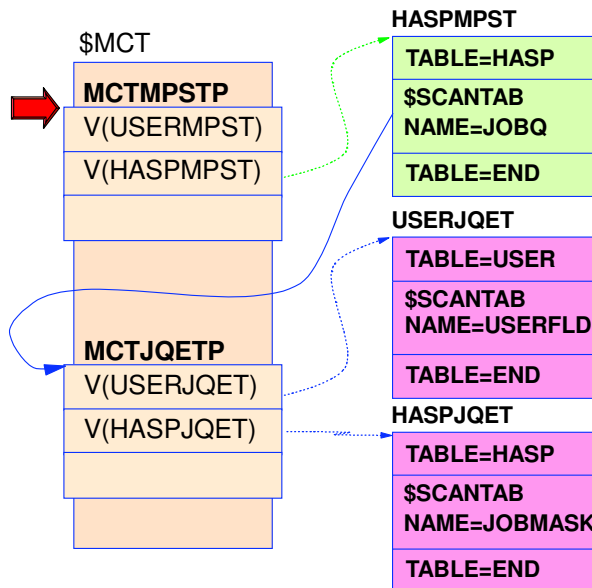
$SCANTAB NAME=MINUTES, MINLEN=3,
          CB=PARENT, DSECT=JQE,
          FILTER= (YES, ALWAYS, GTLT) ,
          PRESCAN= (PREFOAGE, DISPLAY, FILTER) , DISPALL=NO,
          FIELD=JQXCRTME, CONV=NUM,
          CALLERS= ($SCDCMDS, $SCSCMDS, $SCECMDS, $SCHCMDS,
                   $SCRLCMD, $SCCCMDS, $SCPCMDS, $SCDOCMD) ,
          RANGE= (0, X' 7FFFFFFF' )
  
```

## CONV=



- Specifies conversion/verification to be done on input values or fields:
  - For **SET** requests, validates input, converts and stores value
  - For **DISPLAY** requests, converts and displays value
  - For **FILTER** requests, converts and compares value
- Valid values
  - **CONV=CHAR** - character input
    - Optionally can specify valid character set
  - **CONV=NUM** - Numeric input (stored in binary)
  - **CONV=HEX** - Hexadecimal input (stored in binary)
  - **CONV=FLAG** - Maps EBCDIC names to bits
  - **CONV=SUBSCAN** - Scan for subparameters

# CONV=SUBSCAN



Commands and Init statements are all defined by tables pointed to by MCTMPSTP

- Tables for specific statements specify CONV=SUBSCAN to define subparameters
- \$SCANTAB NAME=JOBQ, CONV=SUBSCAN, SCANTAB=MCTJQETP

To define new commands or init statements, associate USER or DYNAMIC tables with MCTMPSTP

# CB, DSECT, FIELD



```
$SCANTAB NAME=MINUTES,MINLEN=3,  
          CB=PARENT,DSECT=JQE,  
          FILTER=(YES,ALWAYS,GTLT),  
          PRESCAN=(PREFOAGE,DISPLAY,FILTER),DISPALL=NO,  
          FIELD=JQXCRTME,CONV=NUM,  
          CALLERS=($SCDCMDS,$SCSCMDS,$SCECMDS,$SCHCMDS,  
                  $SCRLCMD,$SCCCMDS,$SCPCMDS,$SCDOCMD),  
          RANGE=(0,X'7FFFFFFF')
```

- **CB=PARENT** indicates to use control block located at parent level of \$SCAN (i.e. JOB)
  - Read-mode JQA for display job commands
  - Update-mode JQA for set job commands
- **DSECT=** and **FIELD=** define which field in control block should be used

# CALLERS

---

```

$SCANTAB NAME=M NUTES, M NLEN=3,
          CB=PARENT, DSECT=JQE,
          FILTER=( YES, ALWAYS, GTLT ) ,
          PRESCAN=( PREFOAGE, DI SPLAY, FI LTER) , DI SPALL=NO,
          FIELDS=JQXCRTME, CONV=NUM
          CALLERS=( $SCDCMDS, $SCSCMDS, $SCECMDS, $SCHCMDS,
          $SCRLCMD, $SCCCMDS, $SCPCMDS, $SCDOCMD),
          RANGE=( 0, X' 7FFFFFFF' )

```

- **CALLERS=** indicates which commands this table applies to:

- \$SCDCMDS - \$D	- \$SCRLCMD - \$A
- \$SCSCMDS - \$T	- \$SCCCMDS - \$C
- \$SCECMDS - \$E	- \$SCPCMDS - \$P
- \$SCHCMDS - \$H	- \$SCDOCMD - Display after
- \$SCSTCMD - \$S	set



# FILTERs

---

```
$SCANTAB NAME=M NUTES, M NLEN=3,  
          CB=PARENT, DSECT=JQE,  
          FILTER=( YES, ALWAYS, GTLT ),  
          PRESCAN=( PREFOAGE, DI SPLAY, FILTER ) , DI SPALL=NO,  
          FI ELD=JQXCRTME, CONV=NUM,  
          CALLERS=( $SCDCMDS, $SCSCMDS, $SCECMDS, $SCHCMDS,  
                   $SCRLCMD, $SCCCMDS, $SCPCMDS, $SCDOCMD ) ,  
          RANGE=( 0, X' 7FFFFFFF' )
```

- **FILTER=** specifies filtering options:
  - **YES** - indicates this keyword may be used as a filter
  - **ALWAYS** - indicates the keyword is always a filter on set commands
  - **GTLT** - indicates that only > and < filtering is allowed

## More about filters

---

- DISPLAY commands
  - Filter by specifying **KEYWORD=value**, or by specifying **/KEYWORD=value**
  - Filter delimiter may be any of
    - ▶ = (equal)
    - ▶ **≠** or **<>** (not equal)
    - ▶ **>**, **>=**, **<**, **<=**, **≠**, **≠**, **≠>=**, or **≠<=** (greater than, less than)
  - **FILTER=** keyword on \$SCANTAB allows
    - ▶ **EQ** - indicates equal filters are allowed
    - ▶ **NEQ** - indicates not equal filters are allowed
    - ▶ **GTLT** - indicates greater than and less than filters are allowed

## More about filters

---

- Set commands
  - **KEYWORD=value** indicates set, **/KEYWORD=value** indicates filter
    - ▶ **ALWAYS** indicates keyword is always a filter , regardless of presence of /
  - **EQ, NEQ, or GTLT** filters may be specified as with displays
  - For same or related keywords, filter must precede set keyword
    - ▶ \$TJOBQ,/CLASS=A,CLASS=B is valid
    - ▶ \$TJOBQ,CLASS=B,/CLASS=A fails
    - ▶ \$TJOBQ,CLASS=A,HOURS>3 is valid

## DISPALL=

---

```
$SCANTAB NAME=M NUTES, M NLEN=3,  
          CB=PARENT, DSECT=JQE,  
          FI LTER=( YES, ALWAYS, GTLT ) ,  
          PRESCAN=( PREFOAGE, DI SPLAY, FI LTER ) , DI SPALL=NO,  
          FI ELD=JQXCRTME, CONV=NUM  
          CALLERS=( $SCDCMDS, $SCSCMDS, $SCECMDS, $SCHCMDS,  
                   $SCRLOMD, $SCOCMDS, $SCPCMDS, $SCDOCMD ) ,  
          RANGE=( 0, X' 7FFFFFFF' )
```

- **DISPALL=** specifies when keyword is to be displayed
  - **NO** - display only when requested explicitly (\$DJQ,MINUTES)
  - **LONGONLY** -display only when requested explicitly or on long displays (\$DJQ, LONG)
  - **YES** - display when requested explicitly, on long displays, and normal display-all situations (display after set, or when no keywords are explicitly requested) (\$DJQ)

# PRESCAN and PSTSCAN routines

---

```
$SCANTAB NAME=M NUTES, M NLEN=3,  
          CB=PARENT, DSECT=JQE,  
          FI LTER=( YES, ALWAYS, GTLT ) ,  
          PRESCAN=( PREFOAGE, DI SPLAY, FI LTER) , DI SPALL=NO,  
          FI ELD=JQXCRTME, CONV=NUM  
          CALLERS=( $SCDCMDS, $SCSCMDS, $SCECMDS, $SCHCMDS,  
                   $SCRLOMD, $SCOCMDS, $SCPCMDS, $SCDOCMD ) ,  
          RANGE=( 0, X' 7FFFFFFF' )
```

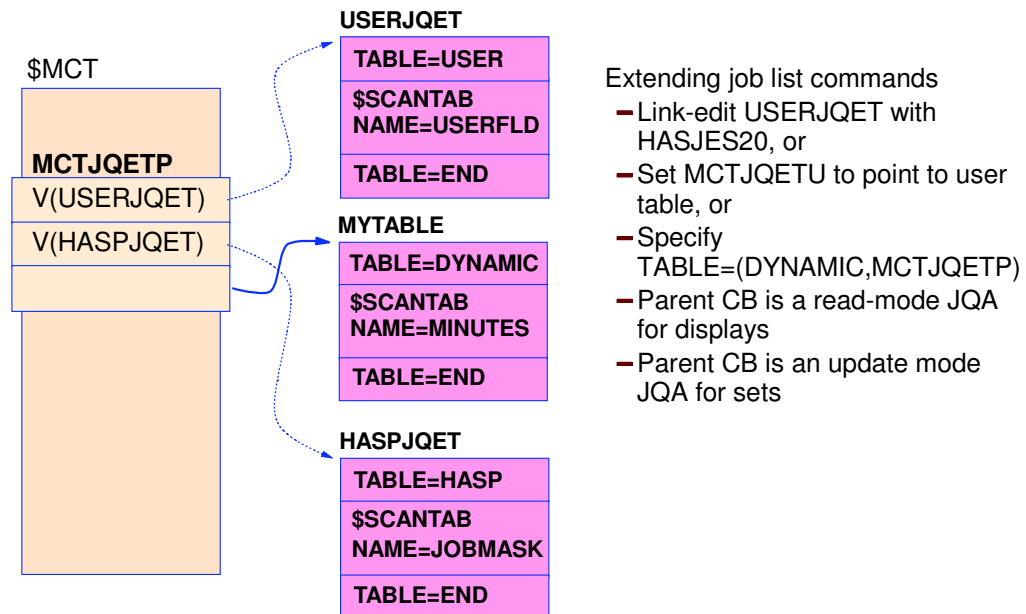
- **PRESCAN=** - allows routines to supplement or override normal processing (before processing keyword)
- **PSTSCAN=** - allows routines to supplement normal processing (after processing keyword)
  - **DISPLAY, FILTER, SET, DELETE** - Indicate specific types of calls a particular routine is to be called for
    - ▶ **PSTSCAN=(routine1,SET,routine2,DISPLAY)**

## PRESCAN and PSTSCAN services

---

- **\$SCANB**
  - Saves contents of field
  - Contents are automatically restored if command fails
- **\$SCAND**
  - Adds text to message area
- **SCWA** control block
  - Basic information about current keyword
    - ▶ Control block address (SCWACBAD)
    - ▶ \$SCANTAB address (SCWASTAB)
    - ▶ Current input address and length
    - ▶ Address of "higher level" SCWAs

# Job command extended table pair



# How to include the new table (1)



## USERJQET \$SCANTAB TABLE=USER

```

$SCANTAB NAME=MINUTES,MINLEN=3,
          CB=PARENT,DSECT=JQE,
          FILTER=(YES,ALWAYS,GTLT),
          PRESCAN=(PREFOAGE,DISPLAY,FILTER),DISPALL=NO,
          FIELD=JQXCRTME,CONV=NUM,
          CALLERS=($SCDCMDS,$SCSCMDS,$SCECMDS,$SCHCMDS,
                  $SCRLCMD,$SCCCMDS,$SCPCMDS,$SCDOCMD),
          RANGE=(0,X'7FFFFFFF')
  
```

## \$SCANTAB TABLE=END

- Link-edit with HASJES20 load module
  - V-con of **USERJQET** is resolved by linkage editor
- Link-edit in separate load module
  - Store address of table in **MCTJQETU**
  - \$SXADDR allows PRESCAN/PSTSCAN routines in HASJES20 load module to be used as of R4



# How to include the new table (2)



MYTABLE \$SCANTAB TABLE=(DYNAMIC,MCTJQETP)

```

$SCANTAB NAME=MINUTES,MINLEN=3,
          CB=PARENT,DSECT=JQE,
          FILTER=(YES,ALWAYS,GTLT),
          PRESCAN=(PREFOAGE,DISPLAY,FILTER),DISPALL=NO,
          FIELD=JQXCRTME,CONV=NUM,
          CALLERS=($SCDCMDS,$SCSCMDS,$SCECMDS,$SCHCMDS,
                  $SCRLCMD,$SCCCMDS,$SCPCMDS,$SCDOCMD),
          RANGE=(0,X'7FFFFFFF')
  
```

\$SCANTAB TABLE=END

- Link-edit in separate load module
  - LOAD of module automatically links table with table pair **MCTJQETP**
  - Multiple tables may be added to table pair this way

## Existing output tables



```
$SCANTAB NAME=HOURS,MINLEN=1,  
          CB=PCE,DSECT=JOE,CBIND=(COJWORK,PCE,LA),  
          FILTER=(YES,ALWAYS,GTLT),  
          PRESCAN=(PREFOAGE,DISPLAY,FILTER),DISPALL=NO,  
          FIELD=JOECRTME,CONV=(NUM,,60),  
          RANGE=(0,X'7FFFFFFF')
```

```
$SCANTAB NAME=DAYS,MINLEN=2,  
          CB=PCE,DSECT=JOE,CBIND=(COJWORK,PCE,LA),  
          FILTER=(YES,ALWAYS,GTLT),  
          PRESCAN=(PREFOAGE,DISPLAY,FILTER),DISPALL=NO,  
          FIELD=JOECRTME,CONV=(NUM,,60*24),  
          RANGE=(0,X'7FFFFFFF')
```

## New output table



```
$SCANTAB NAME=MINUTES,MINLEN=3,  
          CB=PCE, DSECT=JOE, CBIND=(COJWORK,PCE,LA),  
          FILTER=(YES,ALWAYS,GTLT),  
          PRESCAN=(PREFOAGE,DISPLAY,FILTER),DISPALL=NO,  
          FIELD=JOECRTME, CONV=NUM,  
          RANGE=(0,X'7FFFFFFF')
```

- **CB=PARENT** is a read-mode JQA
- Copies of work and characteristics JOEs are in PCE
  - **CB=PCE,CBIND=(COJWORK,PCE,LA)** for work JOE
  - **CB=PCE,CBIND=(COCHAR,PCE,LA)** for char JOE
- **DSECT=** and **FIELD=** define which field in control block should be used

# CALLERS



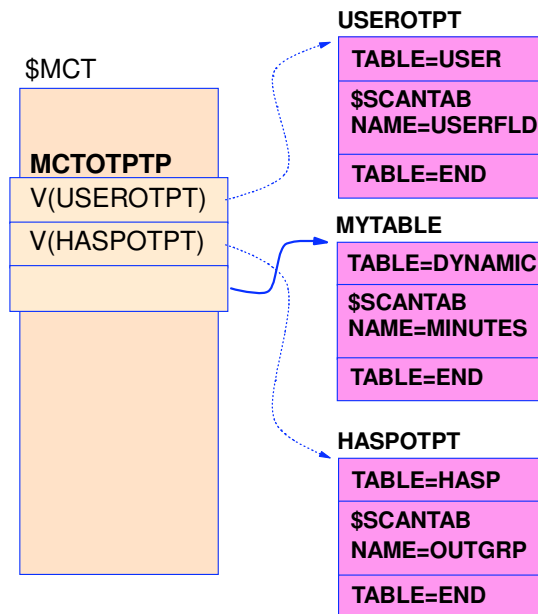
```

$SCANTAB NAME=MINUTES,MINLEN=3,
          CB=PCE,DSECT=JOE,CBIND=(COJWORK,PCE,LA),
          FILTER=(YES,ALWAYS,GTLT),
          PRESCAN=(PREFOAGE,DISPLAY,FILTER),DISPALL=NO,
          FIELD=JOECRTME,CONV=NUM,
          RANGE=(0,X'7FFFFFFF')

```

- **CALLERS=** is unspecified, indicating all callers from parent \$SCANTAB should be used:
  - **\$SCLTCMD** - \$DO
  - **\$SCTOCMD** - \$TO
  - **\$SCPOCMD** - \$PO
  - **\$SCCOCMD** - \$CO
  - **\$SCOCMDS** - \$O
  - **\$SCLOCMD** - Display after set

# Output command extended table pair



Extending job list commands

- Link-edit USEROTPT with HASJES20, or
- Set MCTOTPTU to point to user table, or
- Specify TABLE=(DYNAMIC,MCTOTPTP)
- Parent CB is a read-mode JQA
- Locate work JOE via CB=PCE, CBIND=(COJWORK,PCE,LA)
- Locate char JOE via CB=PCE, CBIND=(COCHAR,PCE,LA)

# How to include the new table



## USEROTPT \$SCANTAB TABLE=USER

```
$SCANTAB NAME=MINUTES,MINLEN=3,  
          CB=PCE,DSECT=JOE,CBIND=(COJWORK,PCE,LA),  
          FILTER=(YES,ALWAYS,GTLT),  
          PRESCAN=(PREFOAGE,DISPLAY,FILTER),DISPALL=NO,  
          FIELD=JOECRTME,CONV=NUM,  
          RANGE=(0,X'7FFFFFFF')
```

## \$SCANTAB TABLE=END

- Link-edit with HASJES20 load module
  - V-con of **USEROTPT** is resolved by linkage editor
- Link-edit in separate load module
  - Store address of table in **MCTOTPTU**

# How to include the new table



```
MYTABLE $SCANTAB TABLE=(DYNAMIC,MCTOTPTP)
```

```
    $SCANTAB NAME=MINUTES,MINLEN=3,  
            CB=PCE,DSECT=JQE,CBIND=(COJWORK,PCE,LA),  
            FILTER=(YES,ALWAYS,GTLT),  
            PRESCAN=(PREFOAGE,DISPLAY,FILTER),DISPALL=NO,  
            FIELD=JOECRTME,CONV=NUM,  
            RANGE=(0,X'7FFFFFFF')
```

```
$SCANTAB TABLE=END
```

- Link-edit in separate load module
  - LOAD of module automatically links table with table pair **MCTOTPTP**
  - Multiple tables may be added to table pair this way

## \$PAIR Macro



- Use the **\$PAIR** macro to define your own table pairs (**CONV=SUBSCAN**)

- IBM-supplied table pairs in the \$MCT

- ▶ **CONV=SUBSCAN, SCANTAB=MCTxxxTP**

<b>MCTxxxTP</b>	<b>\$PAIR</b>	<b>TABLE=SCAN</b>	
+MCTxxxTP	DS	0H	
+MCTxxxTU	DC	V(USERxxxT)	User table
+MCTxxxTH	DC	V(HASPxxxT)	JES2 table
+MCTxxxTD	DC	A(*-*)	Dynamic tables

- Installation-defined table pairs in the \$UCT

- ▶ **CONV=SUBSCAN, SCANTAB=(UCTxxxTP, UCT)**

<b>UCTxxxTP</b>	<b>\$PAIR</b>	<b>TABLE=SCAN, HASPENT=NONE</b>	
+UCTxxxTP	DS	0H	
+UCTxxxTU	DC	V(USERxxxT)	User table
+	DC	A(*-*)	No JES2 table
+UCTxxxTD	DC	A(*-*)	Dynamic tables



## \$PAIR Macro



- In inline code
  - ▶ When extendability is not important (e.g. HASPMSG or installation code)
  - ▶ **CONV=SUBSCAN,SCANTAB=(MYTABLE,ADDR)**

```
MYTABLE $PAIR ,
+MYTABLE DS 0H
+ DC A(MYTABLE_U) User table
+ DC A(0) No JES2 table
+ DC A(0) No Dynamic tables
+MYTABLE_U $SCANTAB TABLE=USER

. . . <insert tables here>

$SCANTAB TABLE=END
```

Extending the \$JQE control block  
with

**\$BERTTABS**



## Extending the JQE



- **Method 1** - add fields to the base JQE
  - Use when the JQE was extended prior to R4 and migration via warm start is required.
  - Include fields in section moved by \$DOGJQE:

```
JQEBB4    EQU    *           Begin bracket
JQENOTIF  DS     CL8         Notify userid
JQEEB4    EQU    *           End bracket
JQESB4    EQU    JQEBB4,JQEEB4-JQEBB4,C'C'  Bracket
```

- Conditional assembly in \$DOGJQE routine does moves and compares accordingly
- **JQE~~BB~~4**, **JQE~~BB~~5**, and **JQE~~BB~~6** are currently unused by IBM code

## Extending the JQE



### ■ Method 2 - Using BERTs to extend the JQA

- Define your field (or fields) in the JQA

```
JQANOTIF DS      CL8
```

- Define a \$BERTTAB to represent the field(s)

```
$BERTTAB CBTYPE=JQE,  
          NAME=NOTIFY,  
          CBOFF=JQANOTIF-JQE,  
          LEN=L' JQANOTIF,  
          PAD=C' '
```

- Number of \$BERTTABs for each CBTYPE is limited (253), so group fields if possible
- Use \$DOGJQE service to access JQA fields

## Extending the JQE



```
$BERTTAB CBTYPE=JQE,  
          NAME=NOTIFY,  
          CBOFF=JQANOTIF-JQE,  
          LEN=L' JQANOTIF,  
          PAD=C' '
```

- **CBTYPE=** - defines the control block type
- **NAME=** - defines a symbolic name for the field in the BERTs
- **CBOFF=** - defines the offset of the field in the control block
- **LEN=** - defines the length of the field
- **PAD=** - defines the pad character

## Extending the JQE



- Include the \$BERTTAB in the table pair

```
USERBRTT $BERTTAB TABLE=USER
```

```
$BERTTAB CBTYPE=JQE,  
        NAME=NOTIFY,  
        CBOFF=JQANOTIF-JQE,  
        LEN=L' JQANOTIF,  
        PAD=C' '
```

```
$BERTTAB TABLE=END
```

- With OW32032, dynamic table may be used

```
MYTABLE $BERTTAB TABLE=DYNAMIC
```

```
$BERTTAB CBTYPE=JQE,  
        NAME=NOTIFY,  
        CBOFF=JQANOTIF-JQE,  
        LEN=L' JQANOTIF,  
        PAD=C' '
```

```
$BERTTAB TABLE=END
```

## Extending the JQE



- **Method 3** - This can be done without a \$JQE modification:

```
$BERTTAB CBTYPE=JQE,  
          NAME=NOTIFY,  
          CBOFF=*,  
          LENGTH=8,  
          PAD=C' '
```

- **CBOFF=\*** - indicates offset to be assigned dynamically

## Extending the JQE



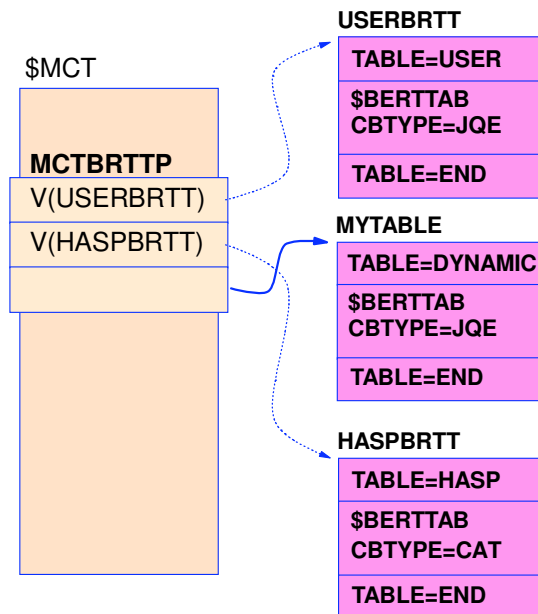
- Obtain offset, when needed, using \$DOGBERT:

```
$DOGBERT ACTION=GETOFFSET,  
          CBTYPE=JQE,  
          NAME=NOTIFY
```

- ▶ **CBTYPE=** matches **CBTYPE=** on \$BERTTAB
  - ▶ **NAME=** matches **NAME=** on \$BERTTAB
  - ▶ Offset is returned in register 1
  - ▶ Length is returned in register 0
- This technique is the only one that is **fully supported!!**



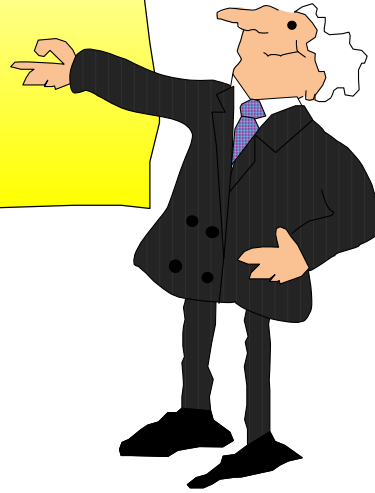
# \$BERTTAB extended table pair



Extending job list commands

- Link-edit USERBRTT with HASJES20, or
- Set MCTBRTP to point to user table, or
- Specify TABLE=DYNAMIC
- Use CBOFF=\* in \$BERTTAB to add fields independently of control block mapping
- Use \$DOGBERT ACTION=GETOFFSET to obtain the offset of fields defined with CBOFF=\*

## Other JES2 table types



# \$WSTAB



## ■ \$WSTAB

- Allows definition of additional work selection criteria for a device (printer, punch, etc)
- Automatically compares data in \$JQE or \$JOE with data in the \$DCT
  - ▶ **RTN=** specifies type of comparison
  - ▶ Standard routines: FLAG, RANGE, COMPARE
  - ▶ Other routines may be used for non-standard comparisons

## Other table types



- **\$PCETAB**
  - Allows installation-defined JES2 processors
- **\$DTETAB**
  - Allows installation-defined JES2 subtasks
- **\$DCTTAB**
  - Allows installation-defined JES2 devices
- **\$TIDTAB**
  - Allows installation-defined JES2 \$TRACE ids
- All require considerably more knowledge to implement

## HASXDYNT



- Shipped as sample part in **SHASSAMP**
- Contains simple examples of various table types
  - **\$SCANTABs** for MINUTES=
  - **\$SCANTABs** for NOTIFY= (notify userid)
  - **\$WSTABs** for WS=(NOTIFY)
  - **\$BERTTABs** for notify userid in \$JQE
    - Also \$DOGBERT calls to locate section
  - **\$EXIT 20** to copy notify userid from \$JCT to \$JQE