

A slide titled 'AGENDA' in large, bold, white, 3D-style letters. The background is dark blue with light streaks. In the top left corner, 'COMMAND JEOPARDY!' is written in white, slanted, 3D-style letters. In the top right corner, there is a small graphic of a blue circuit board with white lines. The main content is a bulleted list:

- What is a command?
- Why create our own commands?
- How to do it!
- Some practical examples...

COMMAND JEOPARDY!

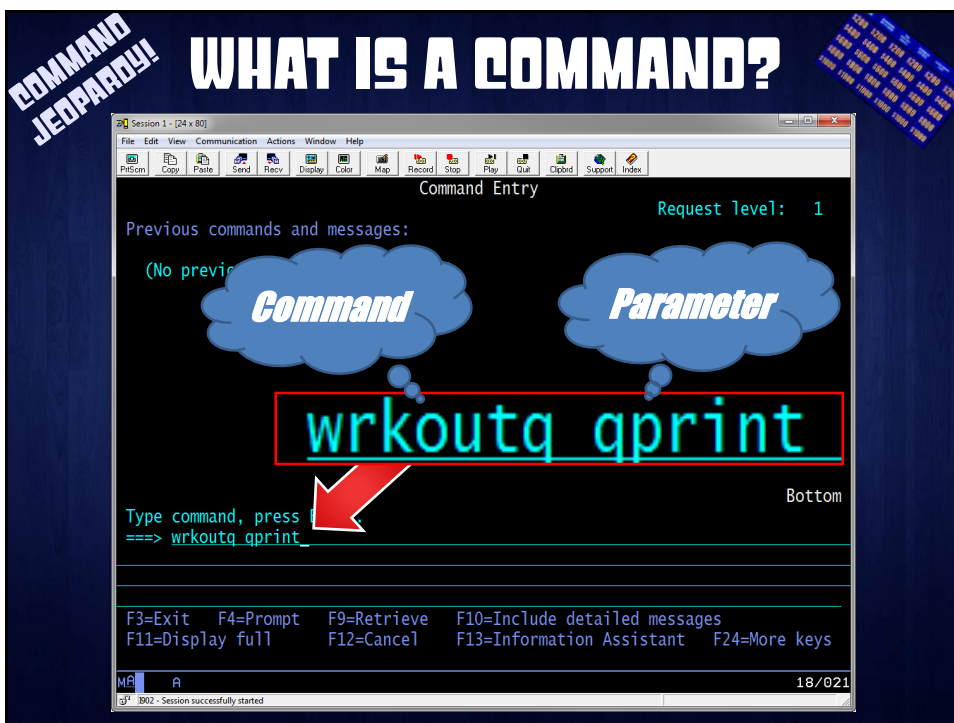
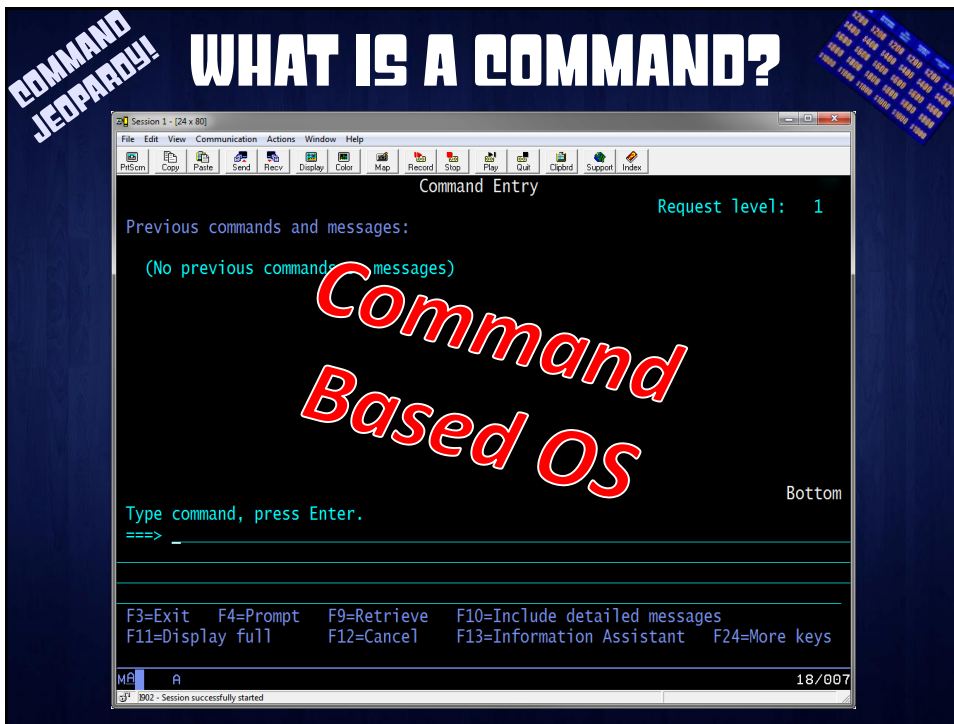
DISCLAIMER

ALL SOFTWARE IS PROVIDED "AS IS" WITHOUT ANY WARRANTY OF ANY NATURE WHATSOEVER. THE PROVIDER OF THIS SOFTWARE HEREBY DISCLAIMS ALL WARRANTIES, REPRESENTATIONS, AND CONDITIONS, STATUTORY OR OTHERWISE, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO WARRANTY OF TITLE AND THE IMPLIED WARRANTY OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE PROVIDER SHALL NOT BE LIABLE FOR ANY DAMAGES ARISING FROM OR AS A RESULT OF YOUR USE OF THIS SOFTWARE. USE IT AS YOUR OWN RISK.

COMMAND JEOPARDY!

DISCLAIMER

This product is meant for educational purposes only. Any resemblance to real persons, living or dead is purely coincidental. Void where prohibited. Some assembly required. **Batteries not included.** Contents may settle during shipment. Use only as directed. No other warranty expressed or implied. Objects in the mirror are closer than they appear. Baby tested, mother approved. Melts in your mouth, not in your hand. Past performance does not guarantee future results. Do not use while operating a motor vehicle or heavy equipment. Postage will be paid by addressee. Subject to approval. This is not an offer to sell securities. Apply only to affected area. May be too intense for some viewers. For recreational use only. All models over 18 years of age. If condition persists, consult your physician. No user-serviceable parts inside. Freshest if eaten before date on carton. Subject to change without notice. Times approximate. Simulated picture. No postage necessary if mailed in the United States. Breaking seal constitutes acceptance of agreement. For off-road use only. As seen on TV. One size fits all. Many suitcases look alike. Contains a substantial amount of non-tobacco ingredients. Colors may, in time, fade. **Slippery when wet.** For office use only. Not affiliated with the American Red Cross. Drop in any mailbox. Edited for television. Keep cool; process promptly. Post office will not deliver without postage. List was current at time of printing. Not responsible for direct, indirect, incidental or consequential damages resulting from any defect, error or failure to perform. At participating locations only. Penalty for private use. Substantial penalty for early withdrawal. Do not write below this line. Falling rocks. Lost ticket pays maximum rate. Your cancelled check is your receipt. Avoid contact with skin. Sanitized for your protection. Be sure each item is properly endorsed. Sign here without admitting guilt. Slightly higher west of the Mississippi. Employees and their families are not eligible. Beware of dog. Contestants have been briefed on some questions before the show. Limited time offer, call now to ensure prompt delivery. You must be present to win. No passes accepted for this engagement. No purchase necessary. Shading within a garment may occur. Use only in well-ventilated area. Keep away from fire or flame. Replace with same type. Check here if tax deductible. Some equipment shown is optional. Price does not include taxes. No Canadian coins. Not recommended for children. Prerecorded for this time zone. Reproduction strictly prohibited. No solicitors. No alcohol, dogs, or horses. Restaurant package, not for resale. List at least two alternate dates. First pull up, then pull down. Call toll free before digging. Driver does not carry cash. Some of the trademarks mentioned in this product appear for identification purposes only. Record additional transactions on back of previous stub. **Decision of judges is final.**



COMMAND JEOPARDY! **WHY CREATE COMMANDS?**

1. Self Documenting Interface

COMMAND JEOPARDY! **WHY CREATE COMMANDS?**

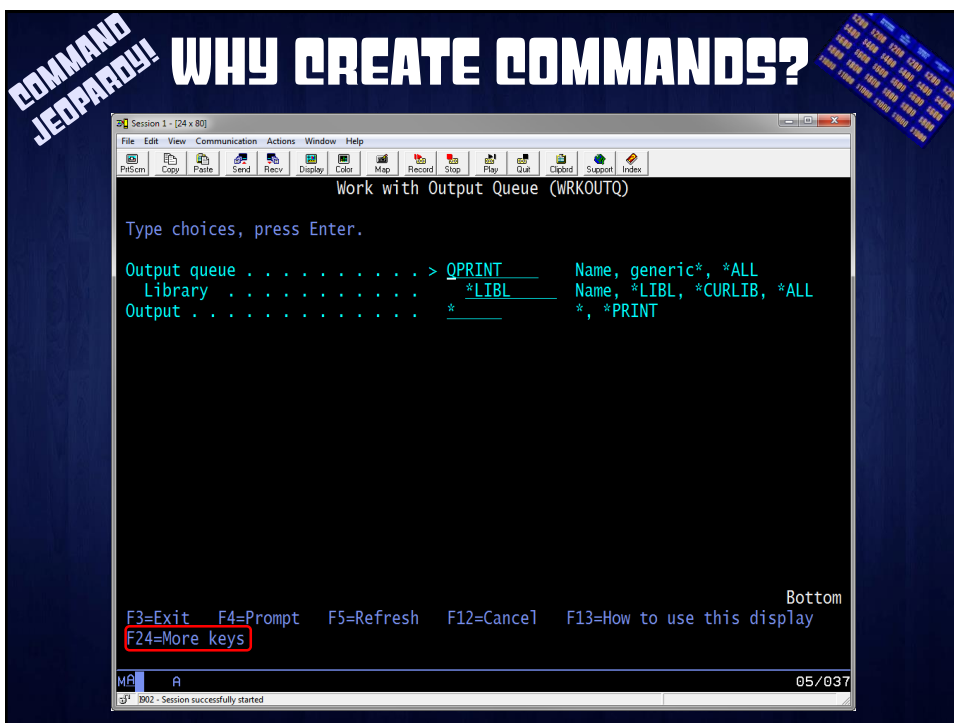
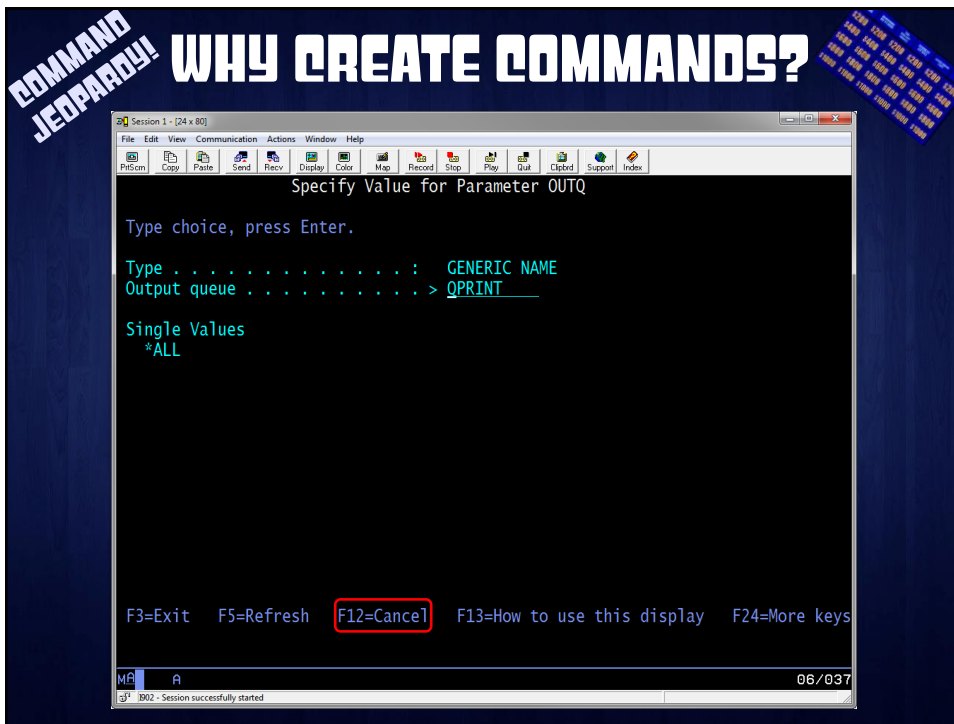
```
Session 1 - [24 x 80]
File Edit View Communication Actions Window Help
Work with Output Queue (WRKOUTQ)
Type choices, press Enter.
Output queue . . . . . > QPRINT_____ Name, generic*, *ALL
Library . . . . . *LIBL_____ Name, *LIBL, *CURLIB, *ALL
Output . . . . . *_____ *, *PRINT
```

F4 to Prompt any Command

Position the Cursor and F4 to Prompt any Parameter

F3=Exit **F4=Prompt** F5=Refresh F12=Cancel F13=How to use this display Bottom
F24=More keys

MA A 05/037
3 3002 - Session successfully started



COMMAND JEOPARDY! **WHY CREATE COMMANDS?**

Work with Output Queue (WRKOUTQ)

Type choices, press Enter.

```

Output queue . . . . . > QPRINT      Name, generic*, *ALL
Library . . . . . *LIBL          Name, *LIBL, *CURLIB, *ALL
Output . . . . . *                *, *PRINT
  
```

F11 to Toggle to Show Parameter Names

F9=All parameters F11=Keywords F14=Command string F24=More keys Bottom

05/037

COMMAND JEOPARDY! **WHY CREATE COMMANDS?**

Work with Output Queue (WRKOUTQ)

Type choices, press Enter.

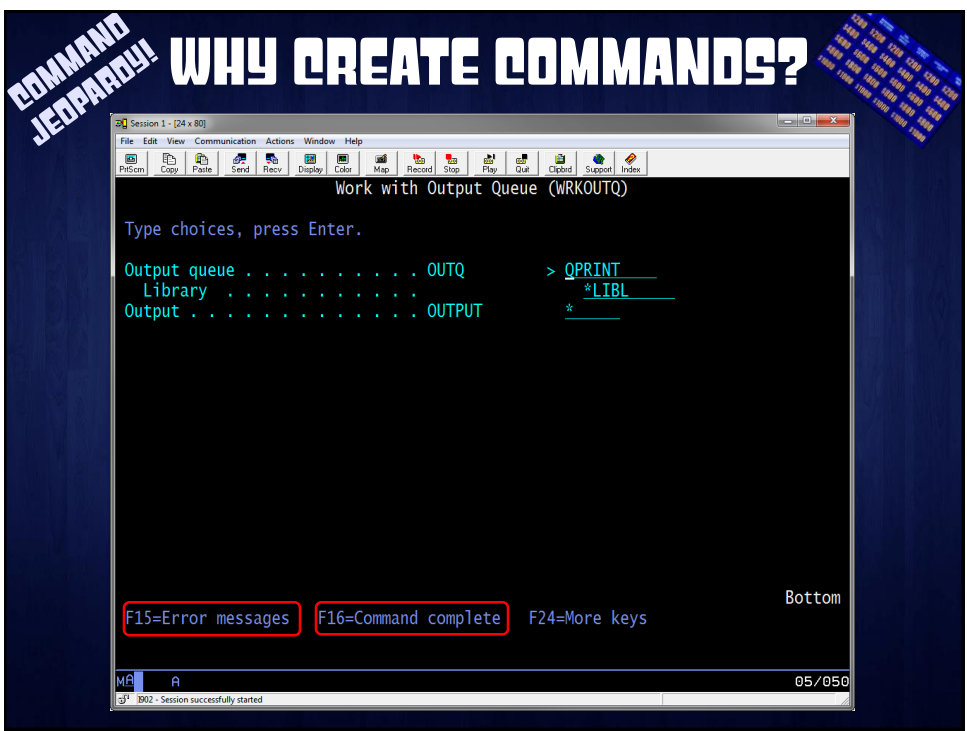
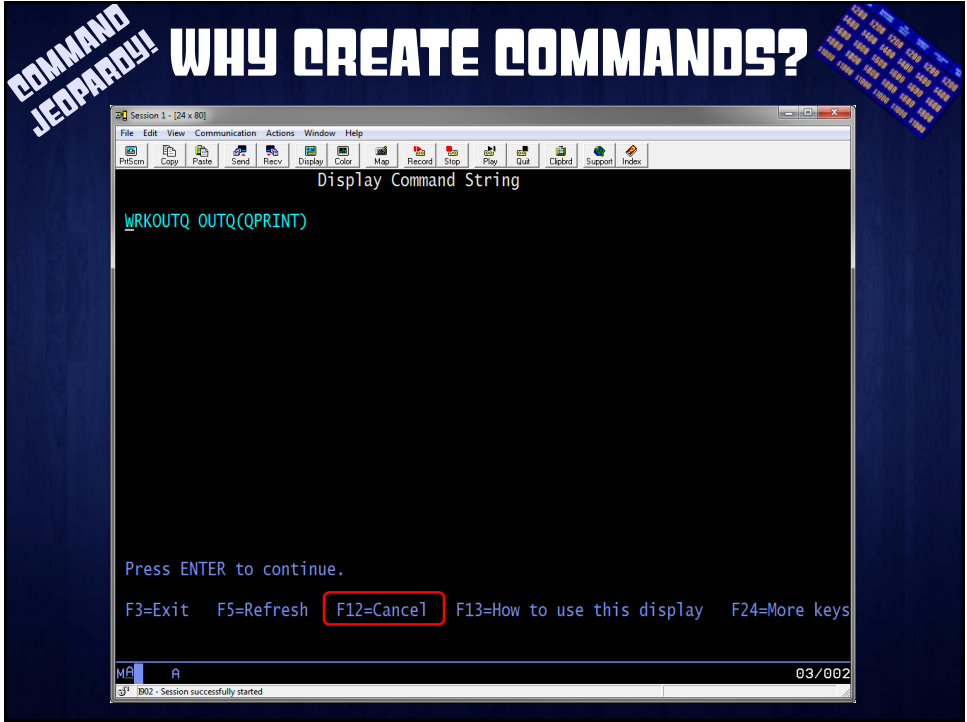
```

Output queue . . . . . OUTQ      > QPRINT
Library . . . . . *LIBL        *LIBL
Output . . . . . OUTPUT        *
  
```

**Like this view better?
CHGUSRPRF USROPT(*CLKWD) to make
it your Default!**

F9=All parameters F11=Choices F14=Command string F24=More keys Bottom

05/050



COMMAND JEOPARDY! WHY CREATE COMMANDS?

1. Self Documenting Interface
2. Help Text

COMMAND JEOPARDY! WHY CREATE COMMANDS?

Work with Output Queue (WRKOUTQ)

Type choices, press Enter.

Output queue OUTQ *ALL

Library *

Output OUTPUT

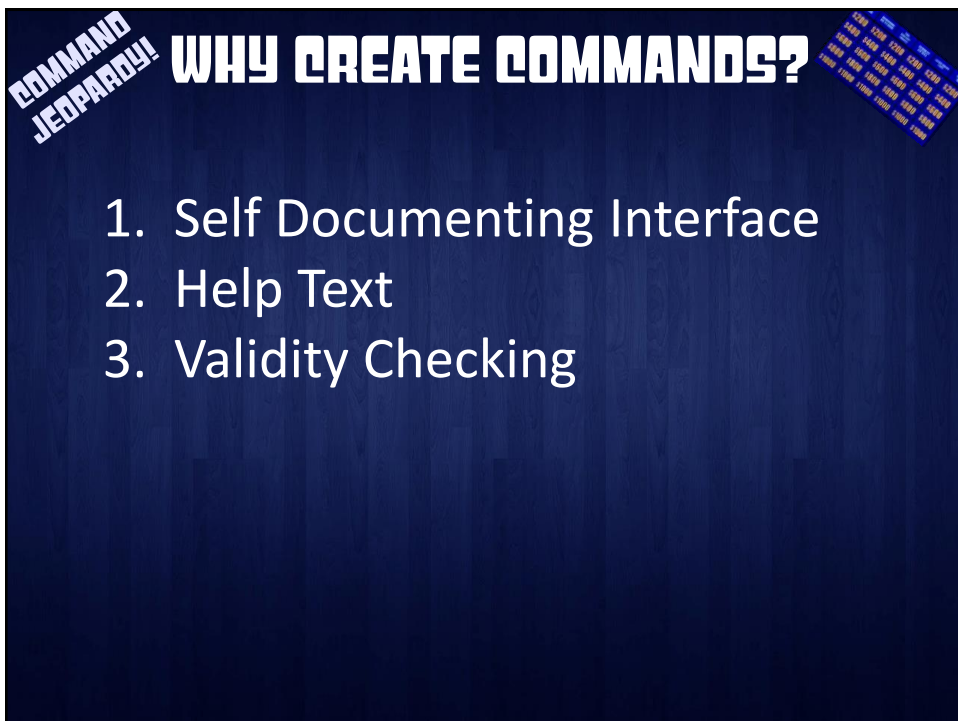
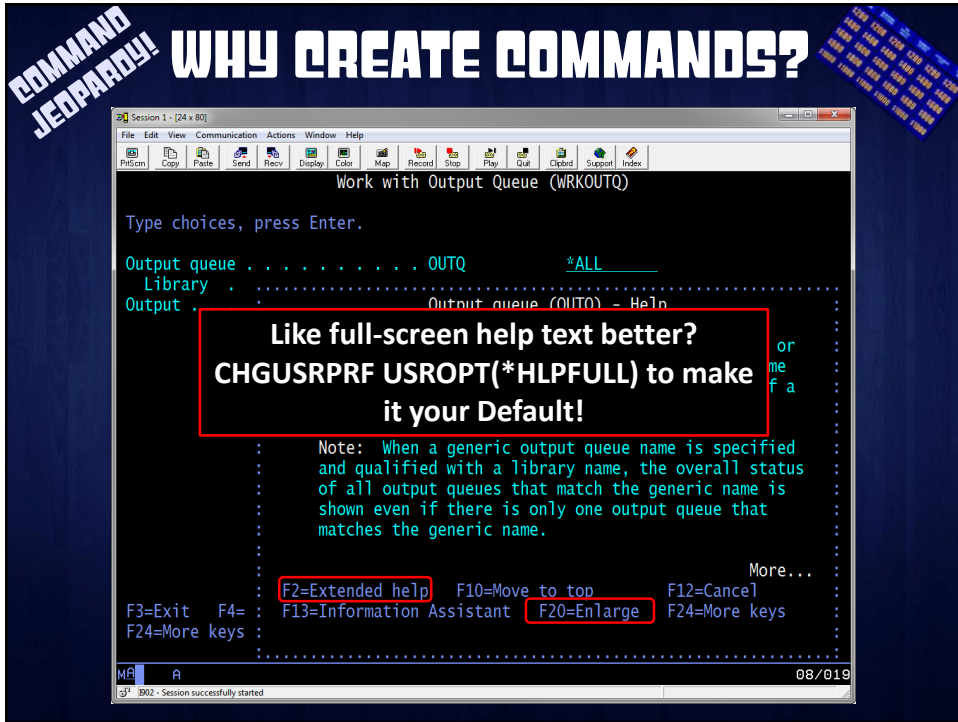
Position cursor and press HELP

Bottom

F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys

MA A 05/050

31 3002 - Session successfully started



COMMAND JEOPARDY! WHY CREATE COMMANDS?

1. Self Documenting Interface
2. Help Text
3. Validity Checking
4. Expandable Prompts

COMMAND JEOPARDY! WHY CREATE COMMANDS?

Type choices, press Enter.

Object	OBJ	* * *
Object type	OBJTYPE	*ALL
Detail	DETAIL	*PRV
Display option	DSPOPT	*PRV

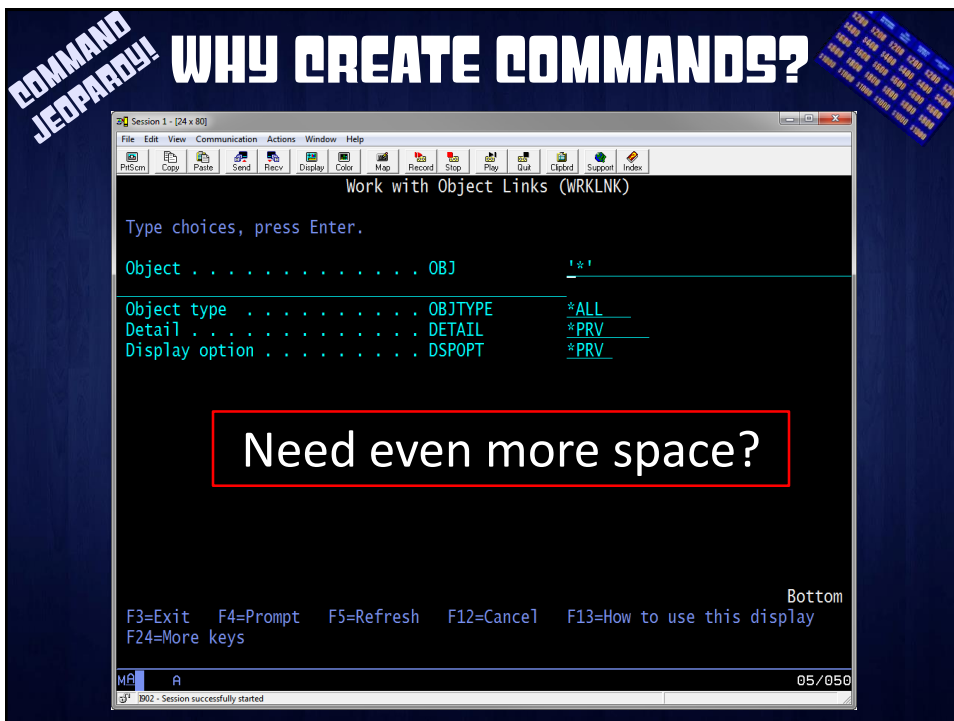
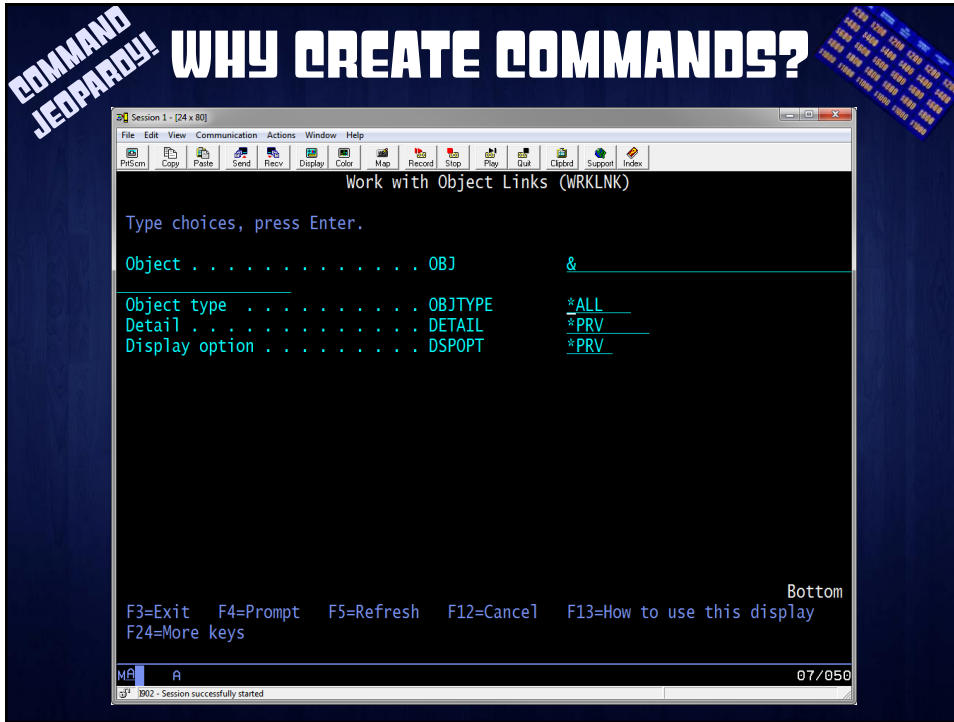
Bottom

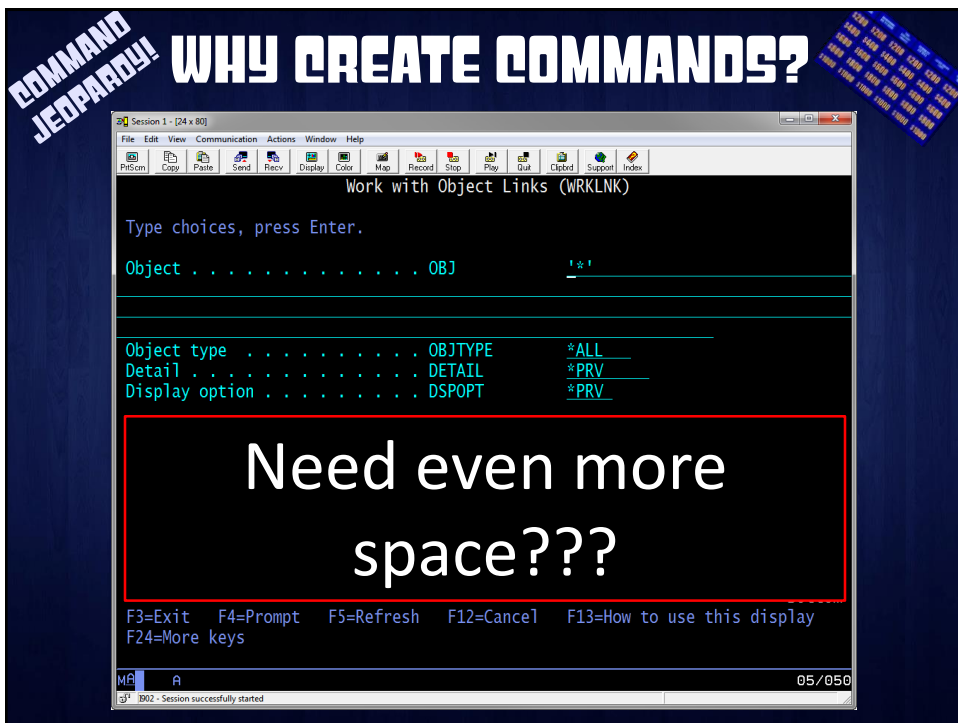
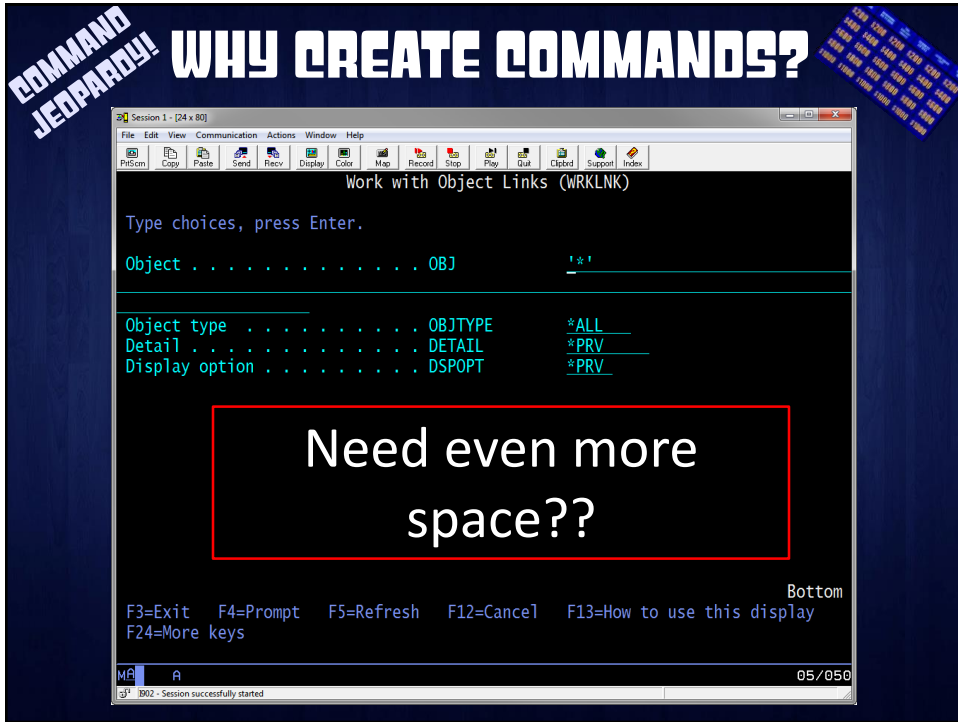
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys

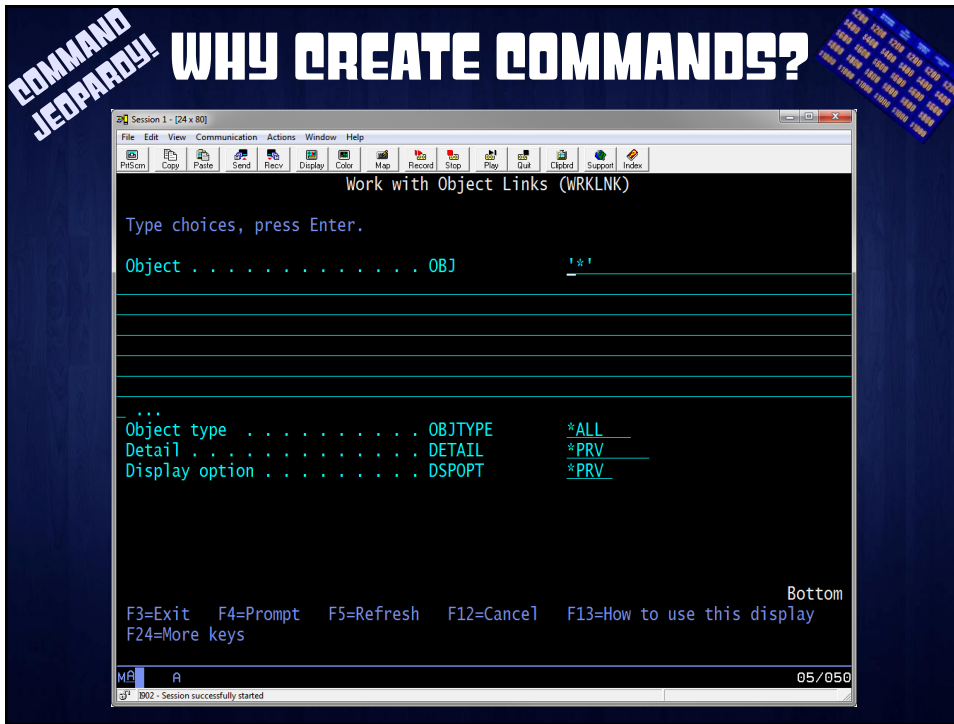
MA A 05/050

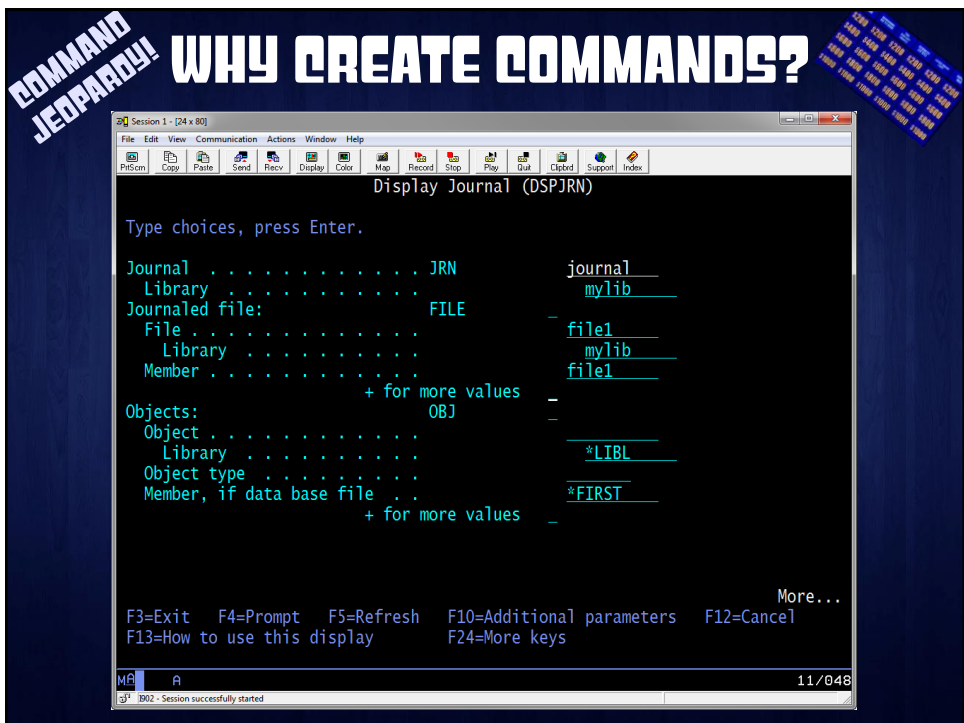
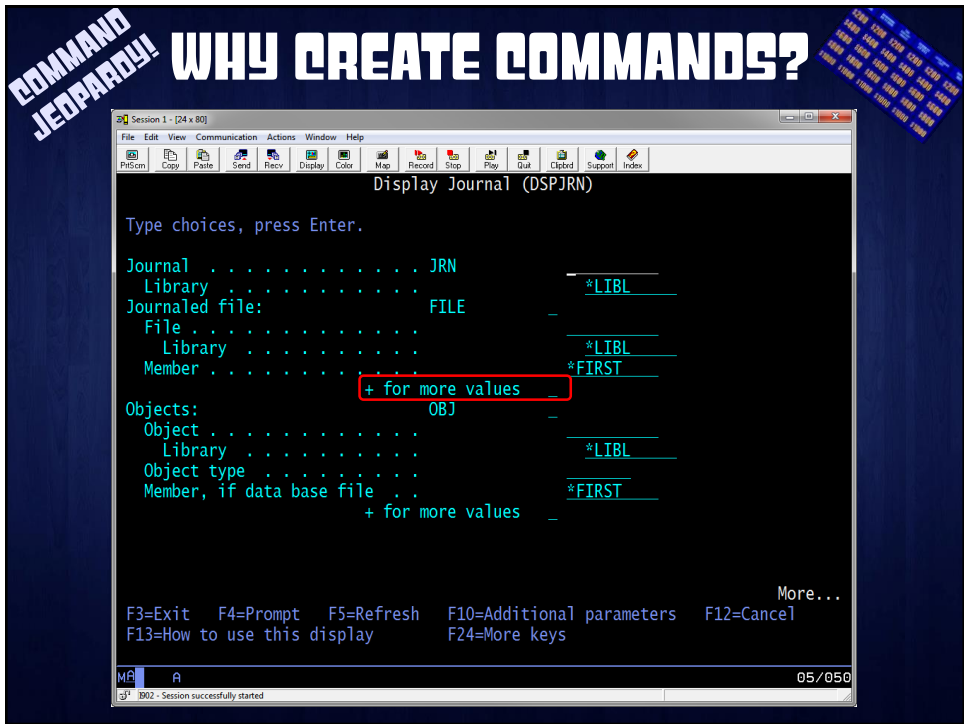
g3 3002 - Session successfully started

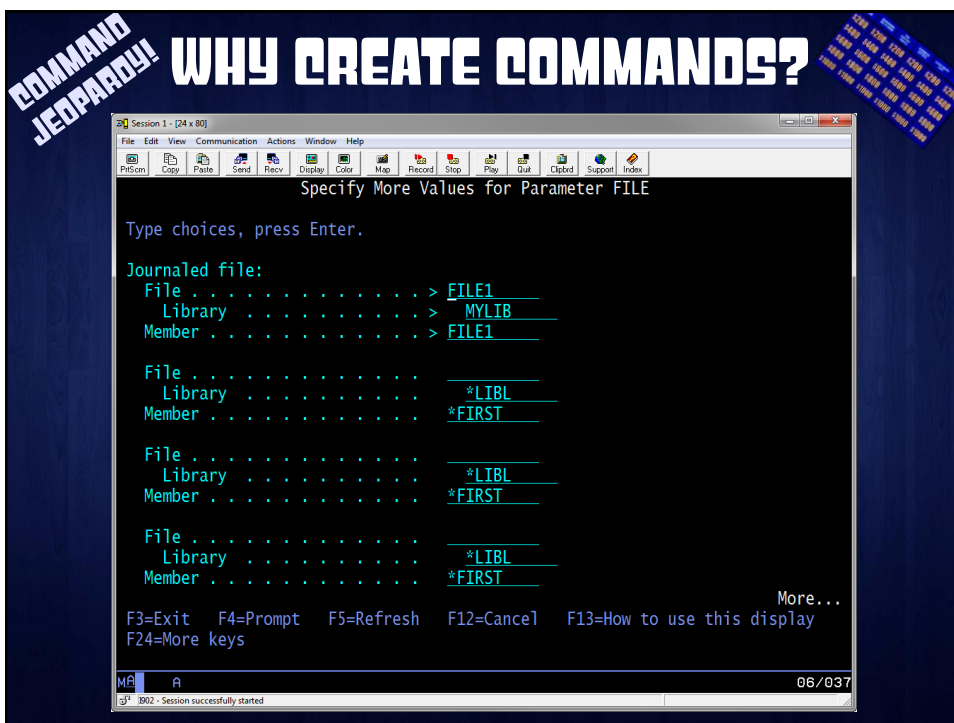
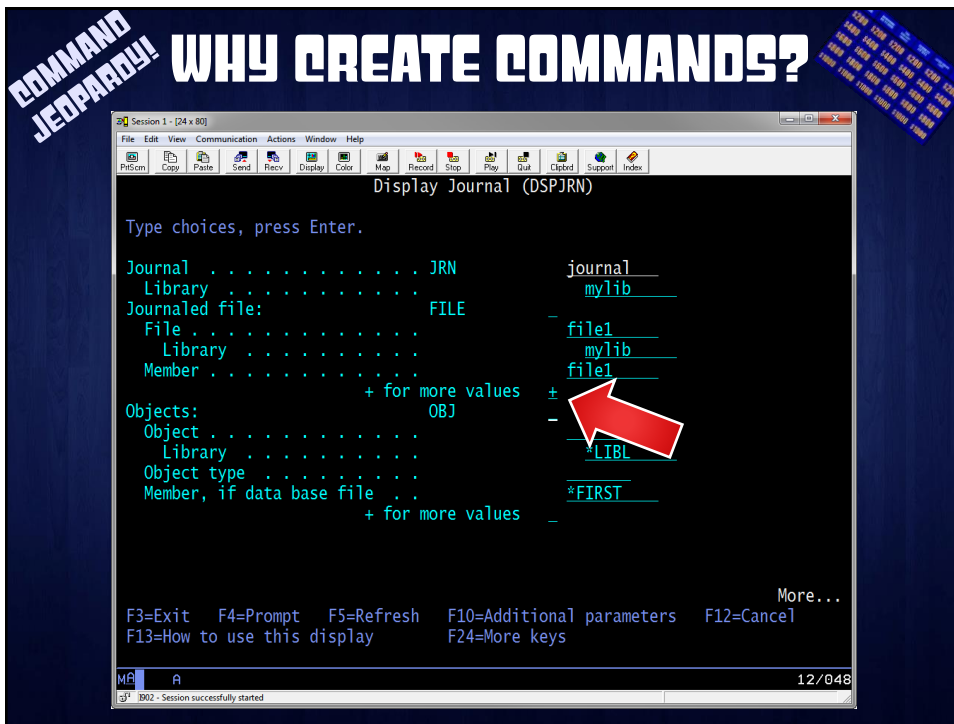
Let's say we need more space for OBJ











COMMAND JEOPARDY! **WHY CREATE COMMANDS?**

Session 1 - [24 x 80]

File Edit View Communication Actions Window Help

Specify More Values for Parameter FILE

Type choices, press Enter.

Journalled file:

File	>	FILE1
Library	>	MYLIB
Member	>	FILE1
File	>	file2
Library	>	mylib
Member	>	file2
File	>	file3
Library	>	mylib
Member	>	file3
File	>	file4
Library	>	mylib
Member	>	file4

More...

F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys

MA A 20/044

31 3902 - Session successfully started

COMMAND JEOPARDY! **WHY CREATE COMMANDS?**

Session 1 - [24 x 80]

File Edit View Communication Actions Window Help

Display Journal (DSPJRN)

Type choices, press Enter.

Journalled file: FILE

File	>	FILE1
Library	>	MYLIB
Member	>	FILE1
File	>	FILE2
Library	>	MYLIB
Member	>	FILE2
File	>	FILE3
Library	>	MYLIB
Member	>	FILE3

More...

F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys

MA A 09/048

31 3902 - Session successfully started

Edit the list:
 + = Add to End of List
 > = Insert within List
 < = Delete Following Entry

COMMAND JEOPARDY! **WHY CREATE COMMANDS?**

1. Self Documenting Interface
2. Help Text
3. Validity Checking
4. Expandable Prompts
5. Flexible Lists
6. Extending the CL Language

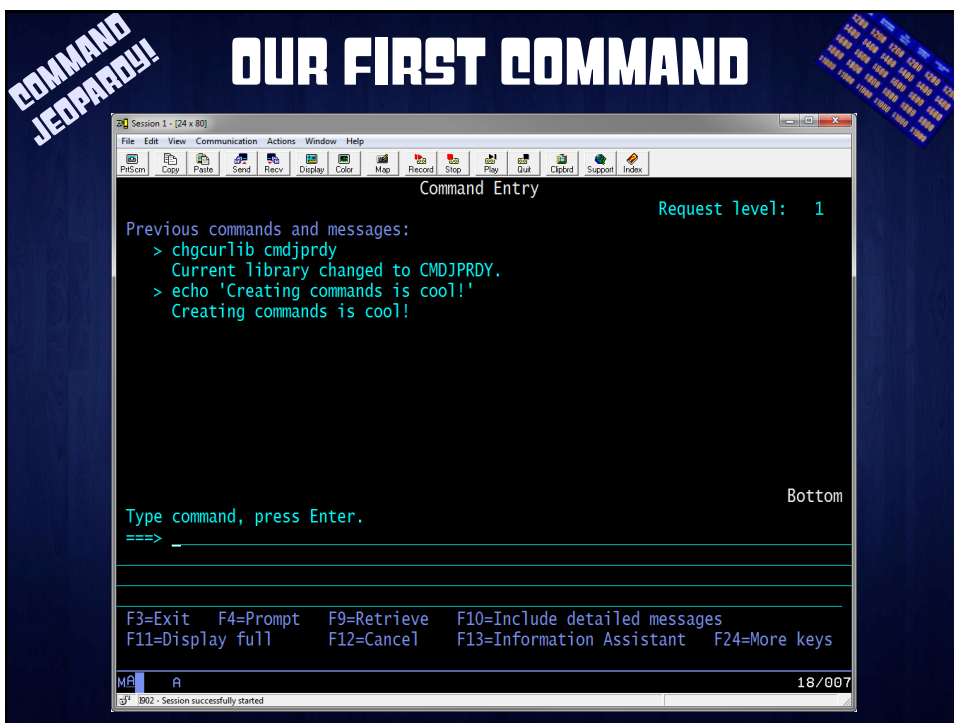
COMMAND JEOPARDY! **WHY CREATE COMMANDS?**

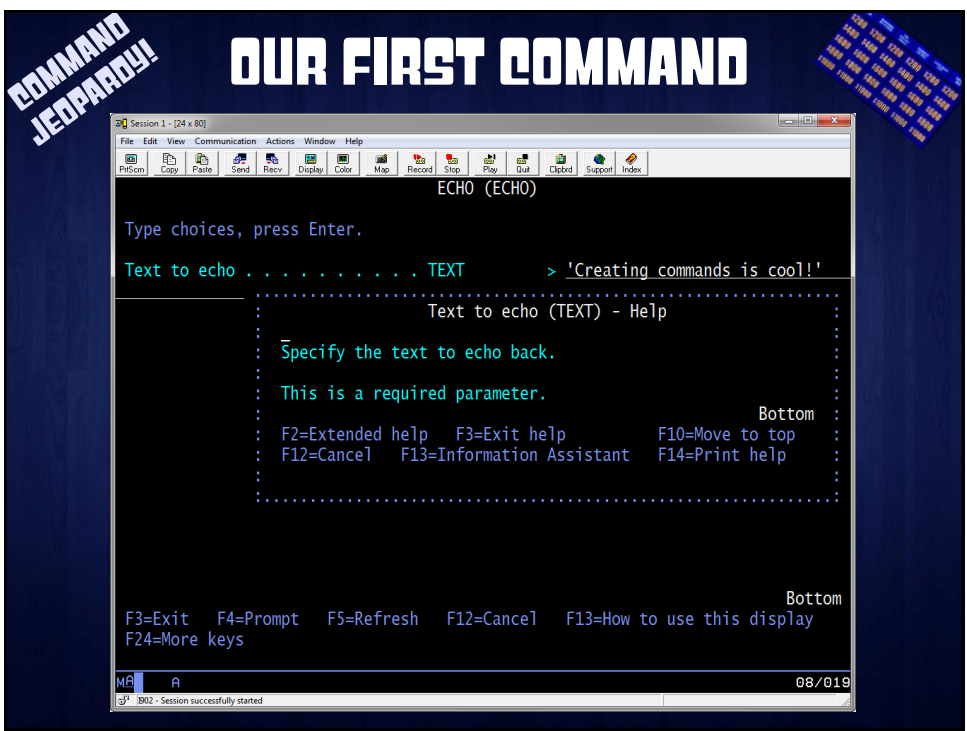
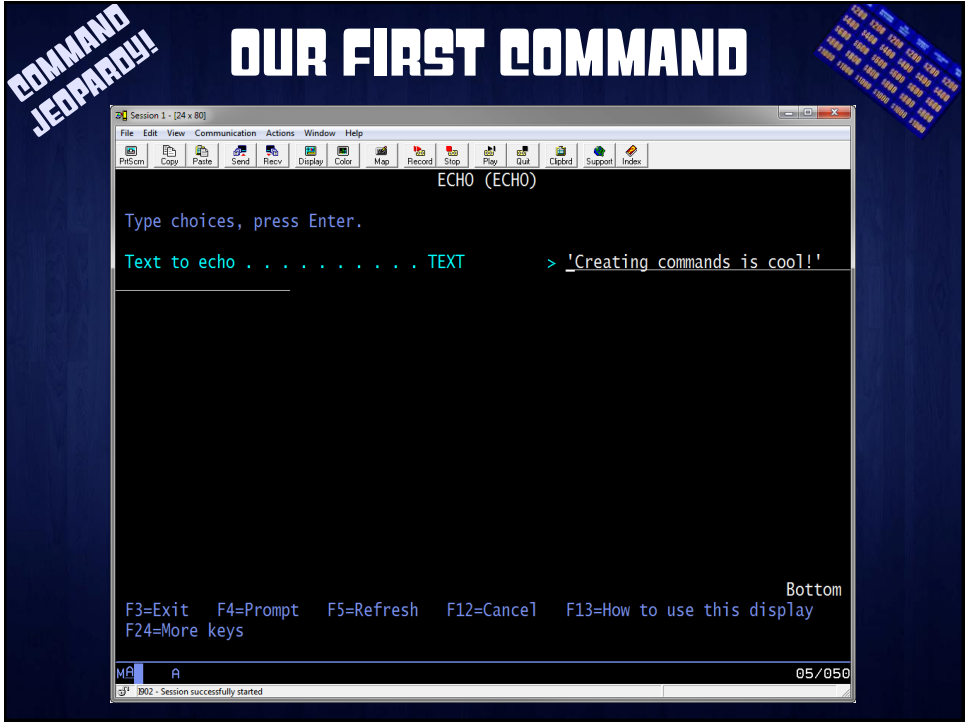
```

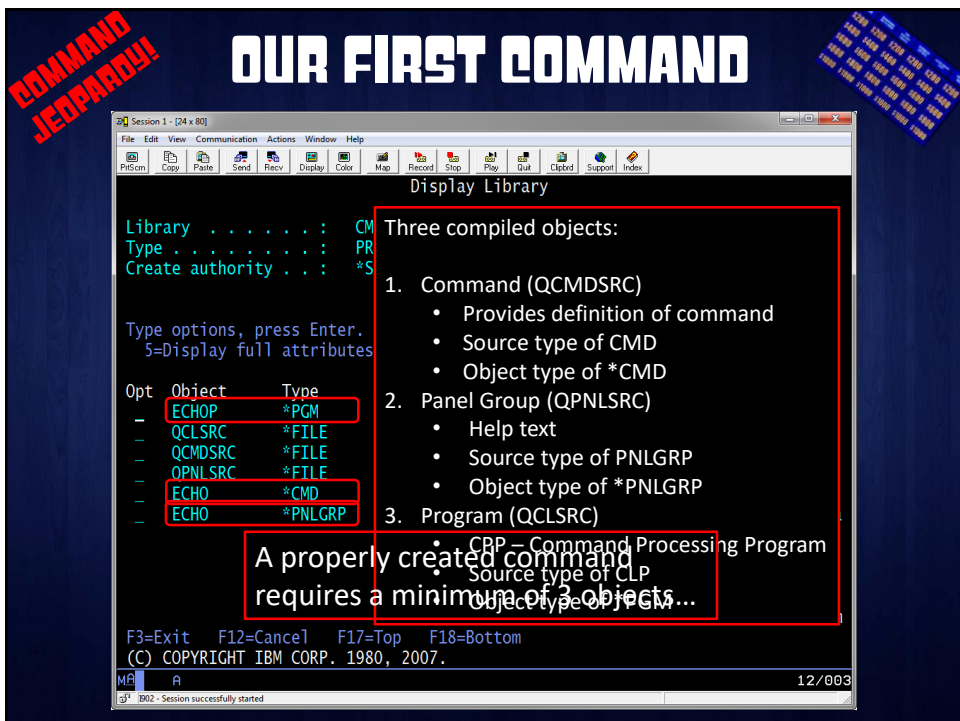
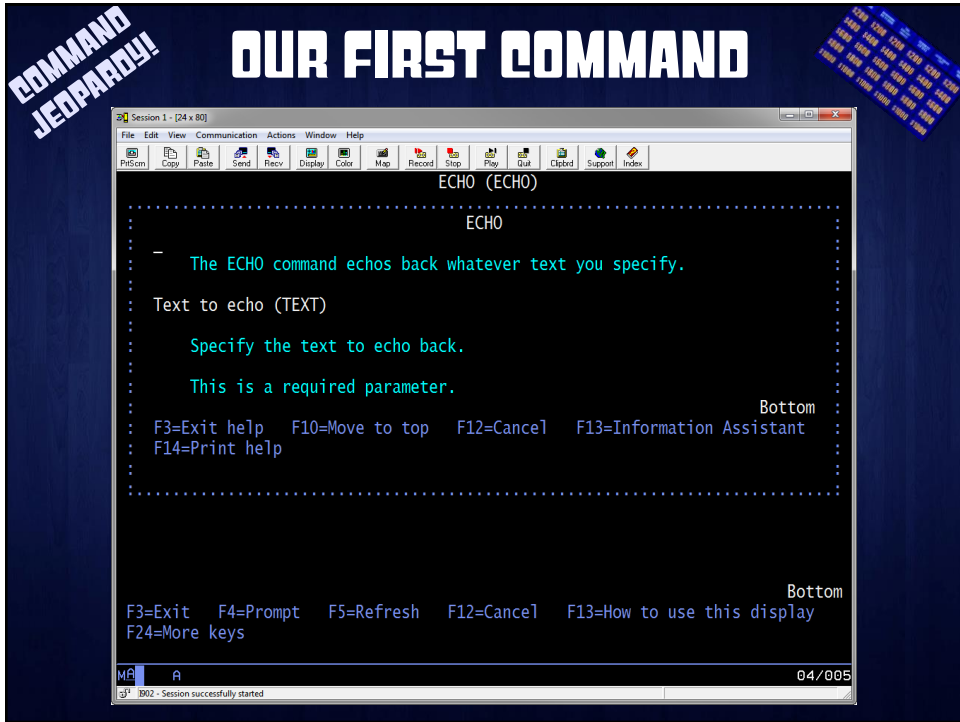
Session 1 - [24 x 80]
File Edit View Communication Actions Window Help
PGMSTEVE/QCLSRC
DEMO
Columns . . . : 1 71 Edit
SEU=>
***** Beginning of data *****
0001.00      PGM
0002.00      WRKOUTQ      OUTQ(QPRINT)
0003.00      ENDPGM
***** End of data *****
    
```

Each statement is a command, and can be an IBM supplied command, or one you've created!

06/001
31 3902 - Session successfully started







COMMAND JEOPARDY!

OUR FIRST COMMAND

```

Columns . . . : 1 71      Edit      CMDJPRDY/QCMDSRC
SEU=>
***** Beginning of data *****
0001.00      CMD      PROMPT(ECHO)
0002.00 /*-----*/
0003.00      PARM      KWD(TEXT) TYPE(*CHAR) LEN(50) MIN(1) +
0004.00                      EXPR(*YES) PROMPT('Text to echo')
0005.00 /*-----*/
***** End of data *****
    
```

Command Definition

MA A 02/009
31 002 - Session successfully started

COMMAND JEOPARDY!

OUR FIRST COMMAND

```

Columns . . . : 1 71      Edit      CMDJPRDY/QPNLSRC
SEU=>
***** Beginning of data *****
0001.00 :PNLGRP.
0002.00
0003.00 :HELP NAME='ECHO/ALL'.
0004.00 :IMHELP NAME='ECHO/TEXT'.
0005.00 :EHELP.
0006.00
0007.00 :HELP NAME='ECHO'.
0008.00 :P.The ECHO command echos back whatever text you specify.
0009.00 :EHELP.
0010.00
0011.00 :HELP NAME='ECHO/TEXT'.
0012.00 Text to echo (TEXT) - Help
0013.00 :XH3.Text to echo (TEXT)
0014.00 :P.Specify the text to echo back.
0015.00 :P.This is a required parameter.
0016.00 :EHELP.
0017.00
0018.00 :EPNLGRP.
***** End of data *****
    
```

Help Text – Panel Group...

Annotations:

- Begin the panel group (points to :PNLGRP.)
- Define extended help and include all parameters (points to :HELP NAME='ECHO/ALL' and :IMHELP NAME='ECHO/TEXT')
- Define command help (points to :HELP NAME='ECHO')
- Define help for each parameter (points to :XH3.Text to echo (TEXT))
- End the panel group (points to :EPNLGRP.)

MA A 02/009
31 002 - Session successfully started

COMMAND JEOPARDY!

OUR FIRST COMMAND



```

Columns . . . : 1 71      Edit      CMDJPRDY/QCLSRC
SEU=>
***** Beginning of data *****
0001.00      PGM      PARM(&ECHOTEXT)
0002.00      DCL      VAR(&ECHOTEXT) TYPE(*CHAR) LEN(50)
0003.00 /*-----*/
0004.00      SNDPGMMSG MSG(&ECHOTEXT)
0005.00 /*-----*/
0006.00      ENDPGM
***** End of data *****
    
```

CPP – Command Processing Program

02/009


002 - Session successfully started

COMMAND JEOPARDY!

OUR FIRST COMMAND

Compiling our command:


- **Panel group:**
 - CRTPNLGRP PNLGRP(CMDJPRDY/ECHO)
SRCFILE(CMDJPRDY/QPNLSRC)
- **Command processing program:**
 - CRTCLPGM PGM(CMDJPRDY/ECHOP)
SRCFILE(CMDJPRDY/QCLSRC)
- **Command:**
 - CRTCMD CMD(CMDJPRDY/ECHO)
PGM(CMDJPRDY/ECHOP)
SRCFILE(CMDJPRDY/QCMDSRC)
HLPNLGRP(CMDJPRDY/ECHO) HLPID(ECHO)



COMMAND JEOPARDY!

QHKIP

Our first really useful command...



COMMAND JEOPARDY!

QHKIP

Retrieve Device Description (QDCRDEVD) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Format name	Input	Char(8)
4	Device name	Input	Char(10)
5	Error Code	I/O	Char(*)

Default Public Authority: *USE

Threadsafe: Yes

The Retrieve Device Description (QDCRDEVD) API retrieves information about a device description.

COMMAND JEOPARDY!

CHKIP

DEVD0600 Format

This format returns detailed information about a device of category *DSP.

Offset		Type	Field
Dec	Hex		
0	0		Returns everything from format DEVD0100
104	68	BINARY(4)	Character identifier: graphic character set
108	6C	BINARY(4)	Character identifier: code page
112	70	BINARY(4)	Maximum length of request unit
116	74	BINARY(4)	Inactivity timer
120	78	BINARY(4)	DBCS feature: RAM size
124	7C	BINARY(4)	Activation timer
128	80	BINARY(4)	Switch setting
132	84	BINARY(4)	Device port

877	36D	CHAR(15)	Internet Protocol (IP) internet address in dotted decimal form
-----	-----	----------	--

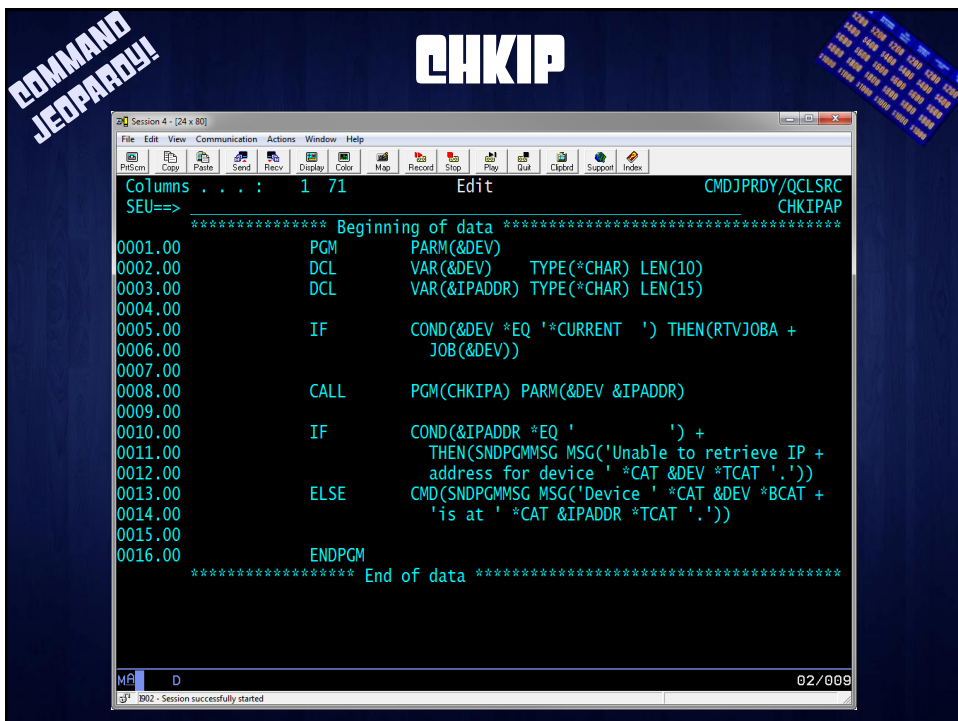
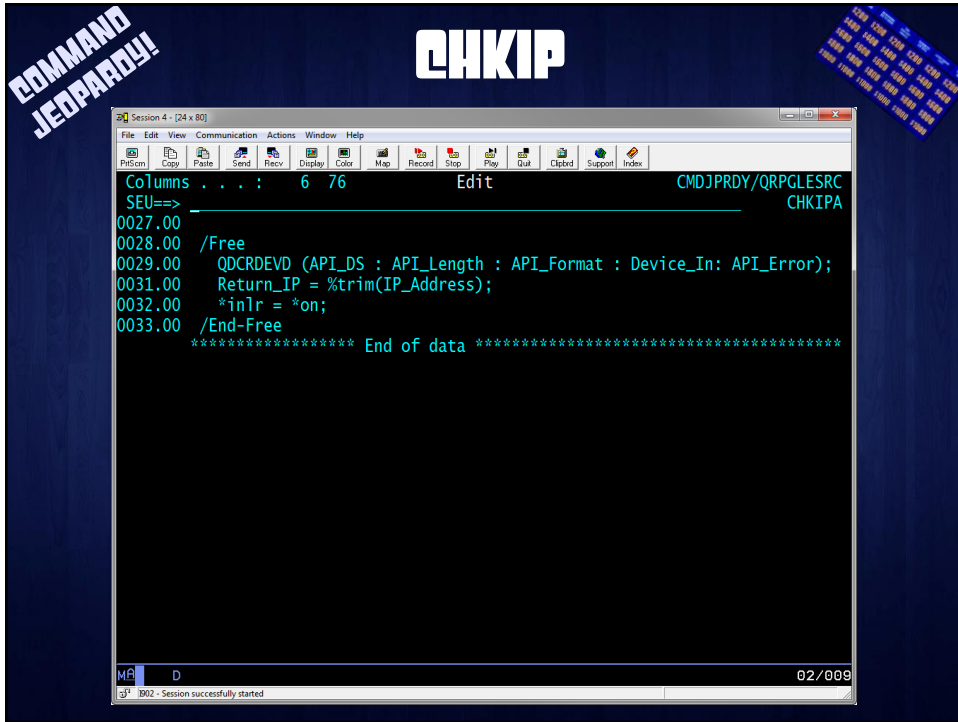
COMMAND JEOPARDY!

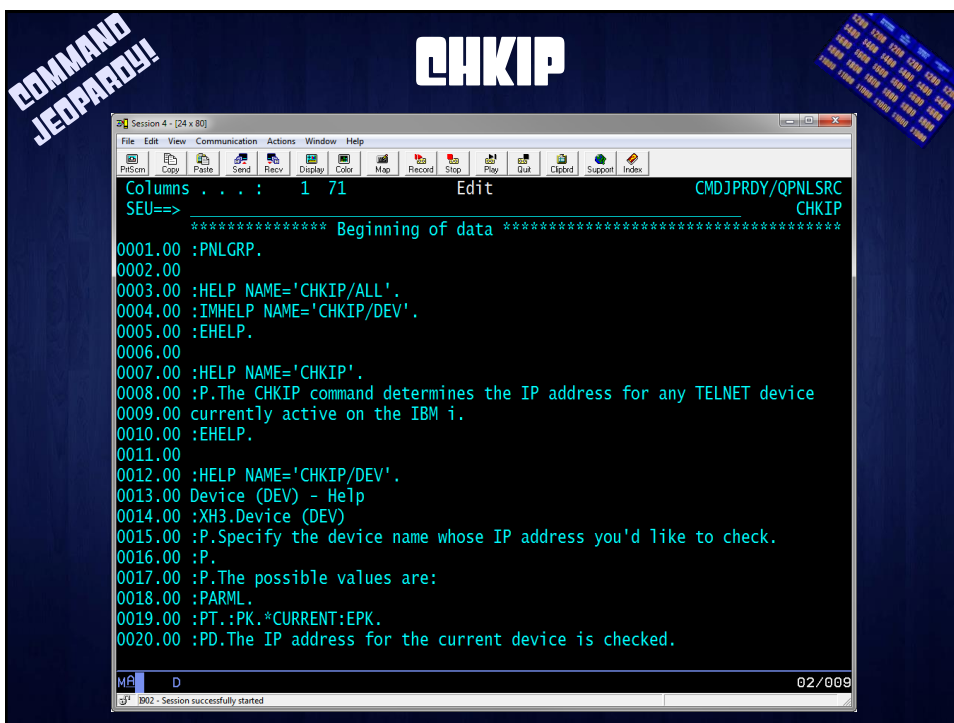
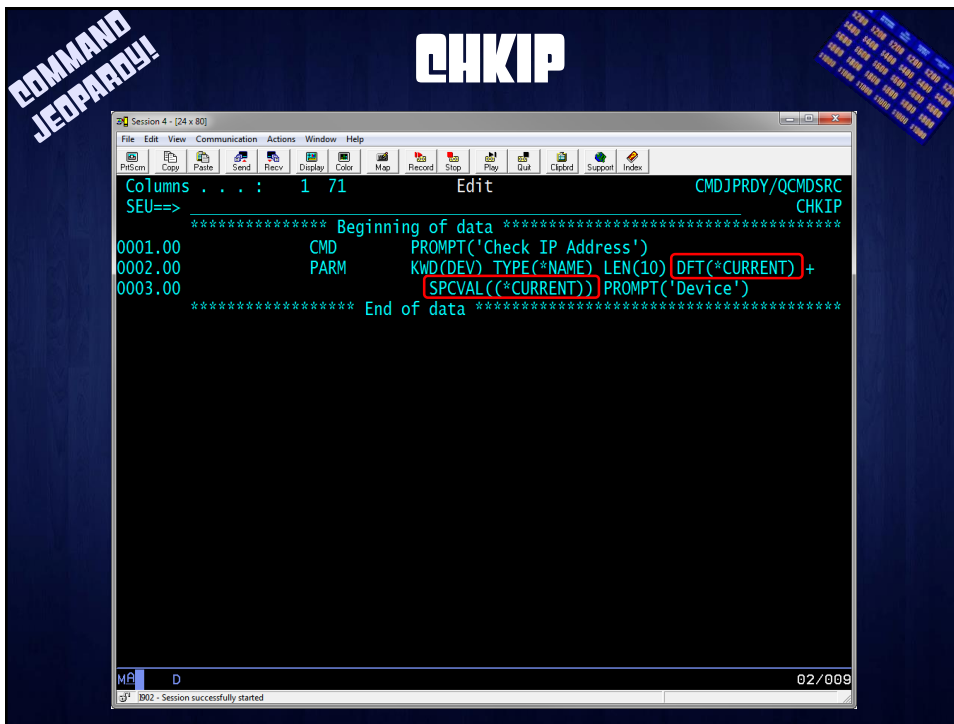
CHKIP

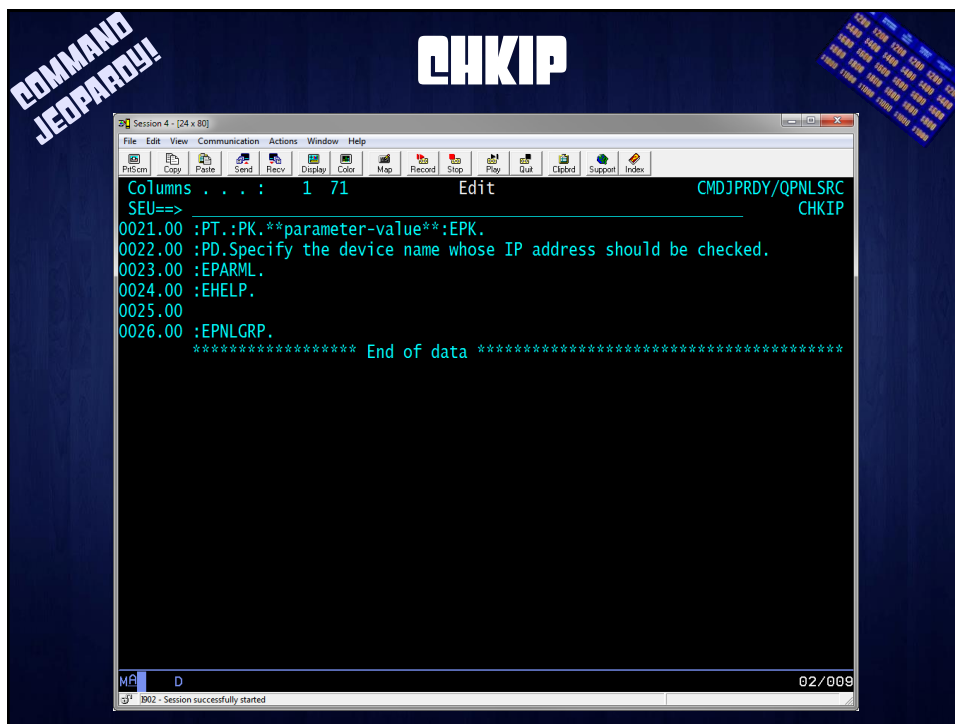
```

Session 4 - [24 x 80]
File Edit View Communication Actions Window Help
Columns . . . : 6 76 Edit CMDJPRDY/QRPGLSRC
SEU=> CHKIPA
***** Beginning of data *****
0001.00 d API_DS ds 1024
0002.00 d IP_Address 877 891
0004.00 d API_Length s 4
0005.00 d API_Format c 'DEVD0600'
0006.00 d API_Error s 256
0007.00
0008.00 d ThisPgm pr ExtPgm('CHKIPA')
0009.00 d 10
0010.00 d 15
0011.00
0012.00 d ThisPgm pi
0013.00 d Device_In 10
0014.00 d Return_IP 15
0020.00
0021.00 d QDCRDEVD pr ExtPgm('QDCRDEVD')
0022.00 d PI_API_DS 1024
0023.00 d PI_API_Length 4
0024.00 d PI_API_Format 8 Const
0025.00 d PI_API_DevNm 10
0026.00 d PI_API_Error 256
    
```

02/009

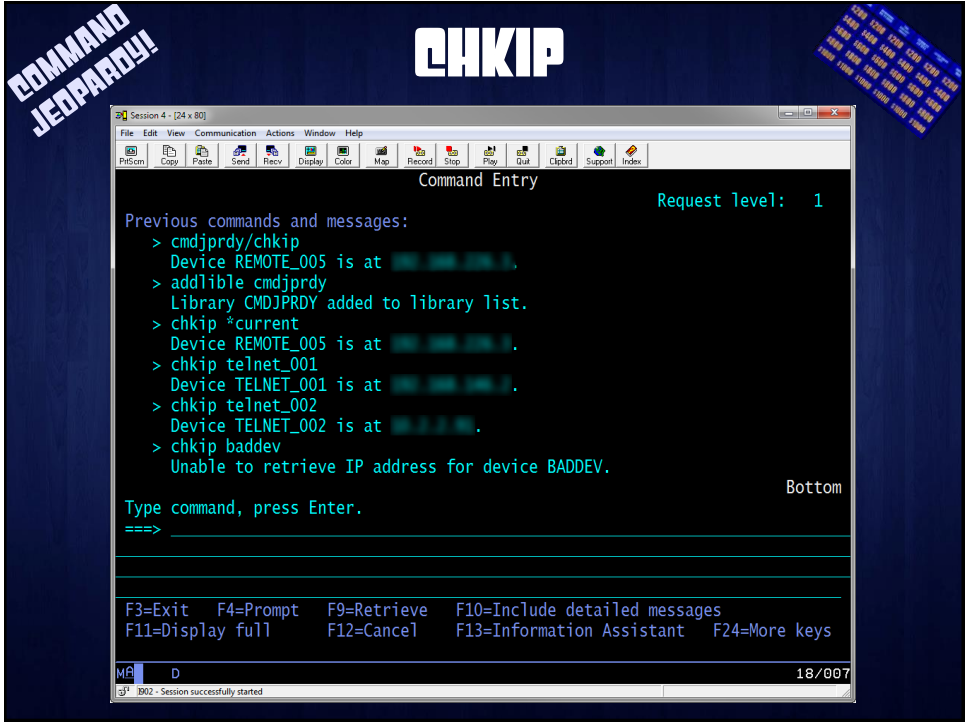


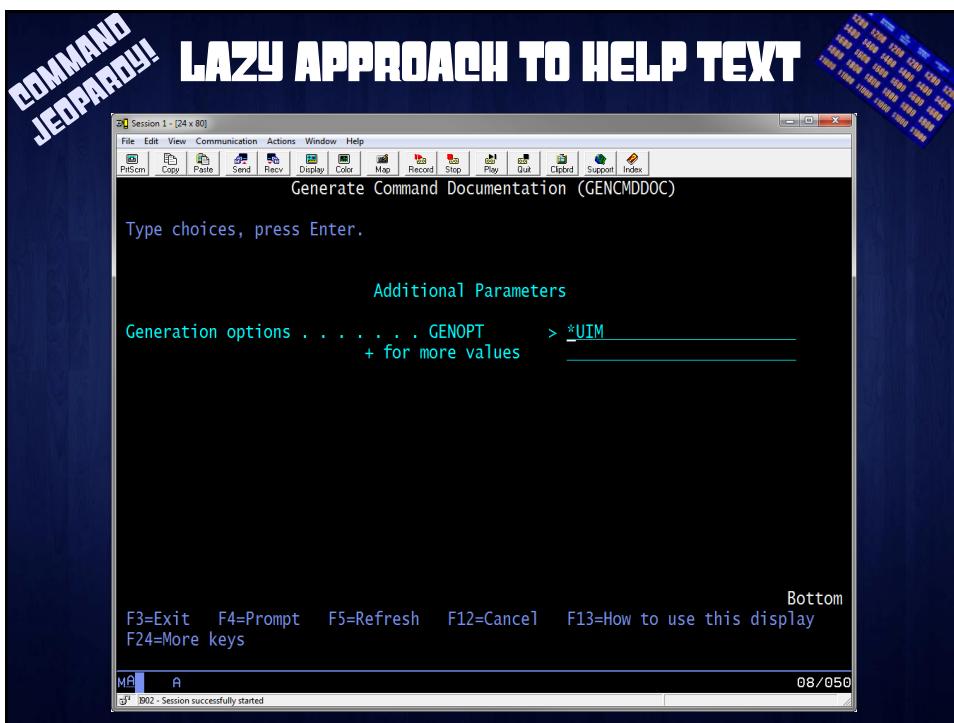
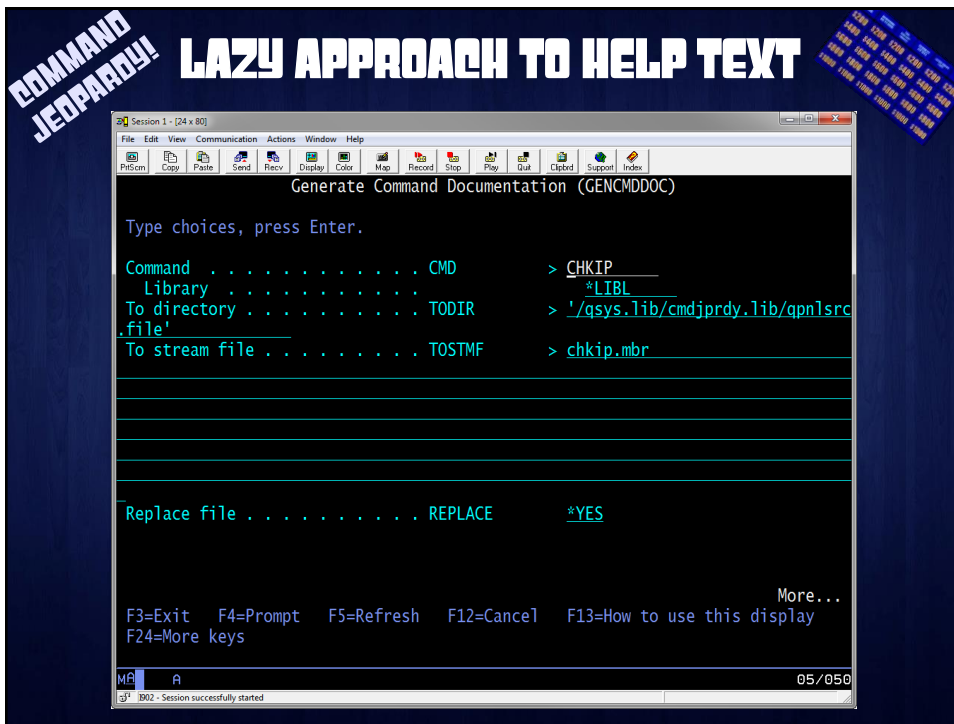




CHKIP

- **RPG Program:**
 - CRTBNDRPG PGM(CMDJPRDY/CHKIPA)
SRCFILE(CMDJPRDY/QRPGLESRC)
- **Command processing program:**
 - CRTCLPGM PGM(CMDJPRDY/CHKIPAP)
SRCFILE(CMDJPRDY/QCLSRC)
- **Panel group:**
 - CRTPNLGRP PNLGRP(CMDJPRDY/CHKIP)
SRCFILE(CMDJPRDY/QPNLSRC)
- **Command:**
 - CRTCMD CMD(CMDJPRDY/CHKIP)
PGM(CMDJPRDY/CHKIPAP)
SRCFILE(CMDJPRDY/QCMDSRC)
TEXT('Check IP Address for Telnet Device')
HLPPNLGRP(CMDJPRDY/CHKIP) HLPID(CHKIP)
PRDLIB(CMDJPRDY)





COMMAND JEOPARDY!

LAZY APPROACH TO HELP TEXT

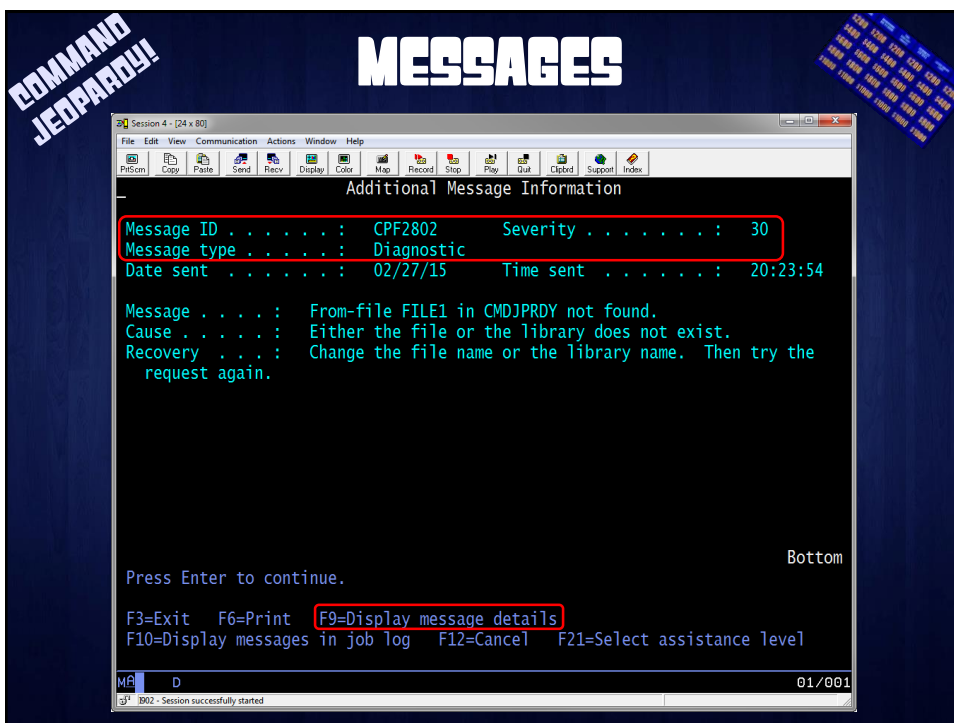
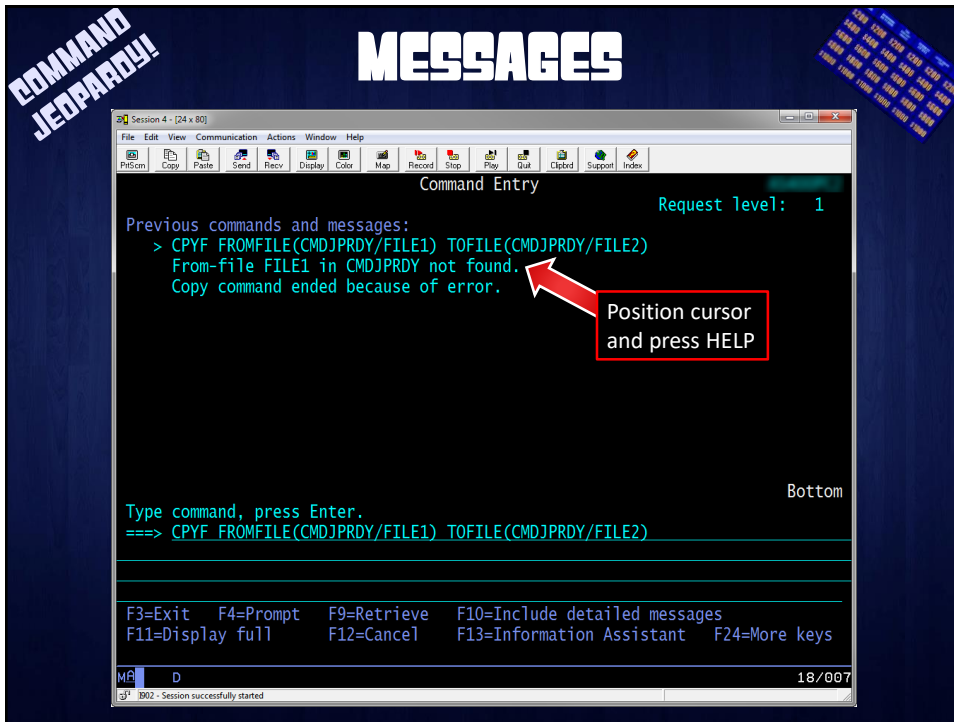
```

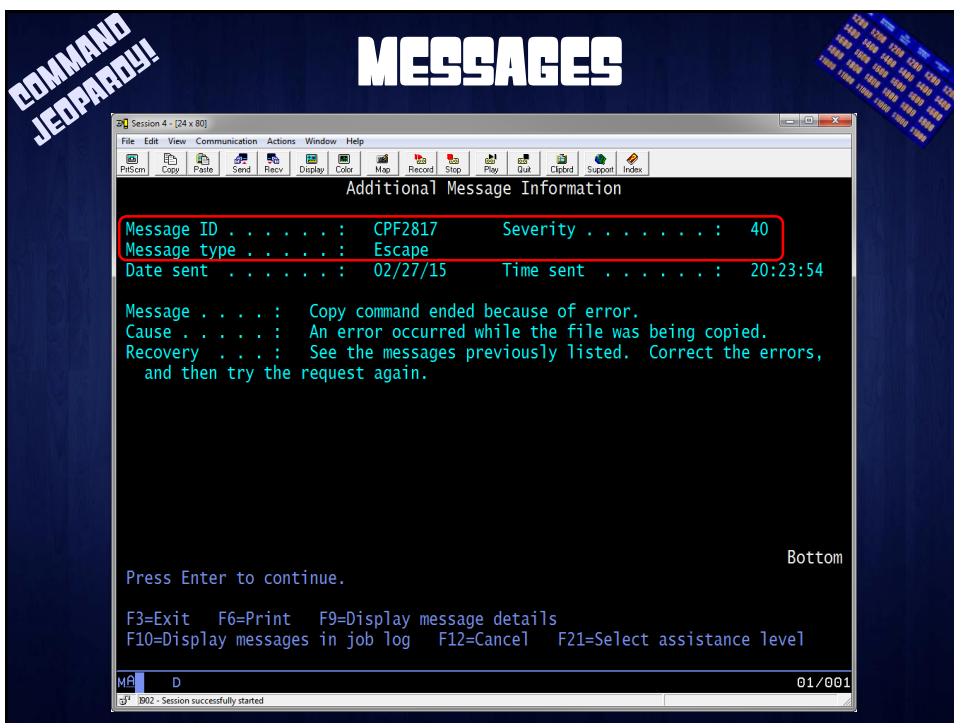
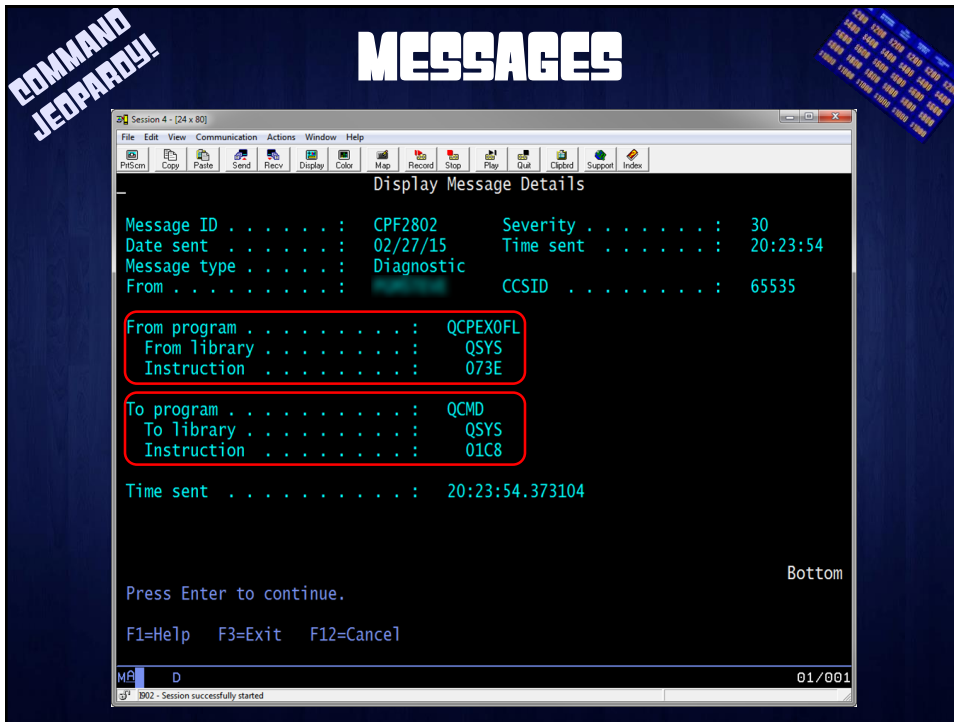
Columns . . . : 1 71      Browse      CMDJPRDY/QPNLSRC
SEU=>
***** Beginning of data *****
0001.00 :pnlgrp.
0002.00 :*****
0003.00 .* Help for command CHKIP
0004.00 :*****
0005.00 :help name='CHKIP'.
0006.00 Check IP Address - Help
0007.00 :p.The Check IP Address (CHKIP) command <...>
0008.00 .* Describe the function provided by the command.
0009.00 :p.:hp2.Restrictions:::ehp2.
0010.00 :u1.
0011.00 :li.
0012.00 You must have <...>
0013.00 .* List all special authorities required to run the command.
0014.00 :li.
0015.00 You must have <...>
0016.00 .* List object or data authorities required to run the command.
0017.00 :li.
0018.00 This command is conditionally threadsafe, <...>
0019.00 .* If conditionally threadsafe, list threadsafe conditions or restricti
0020.00 :li.
  
```

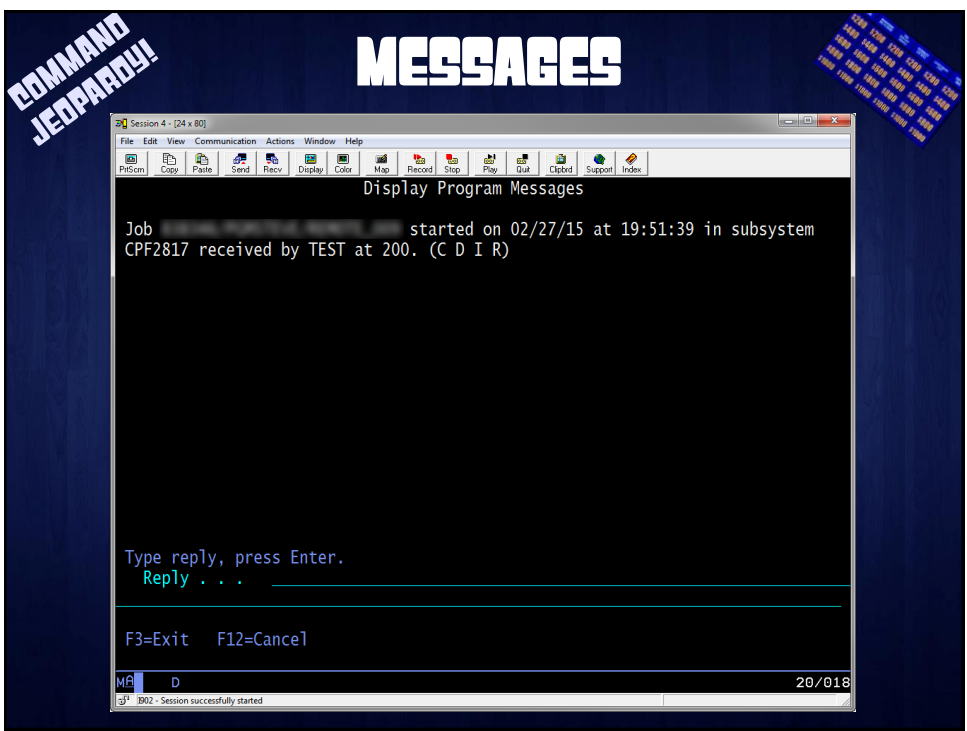
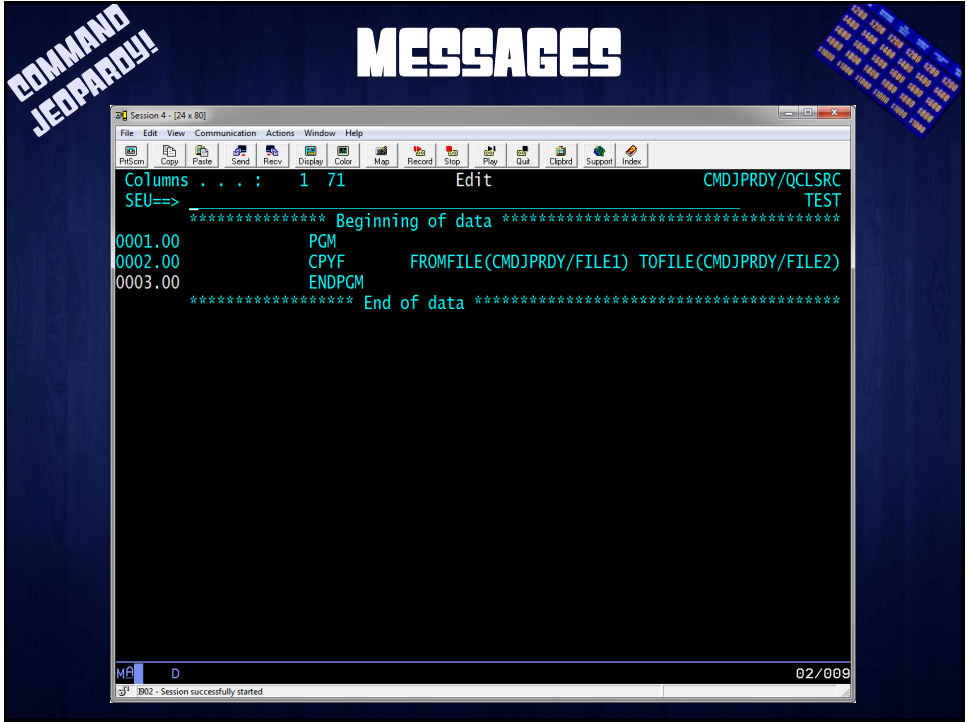
MA B 02/009
03 002 - Session successfully started

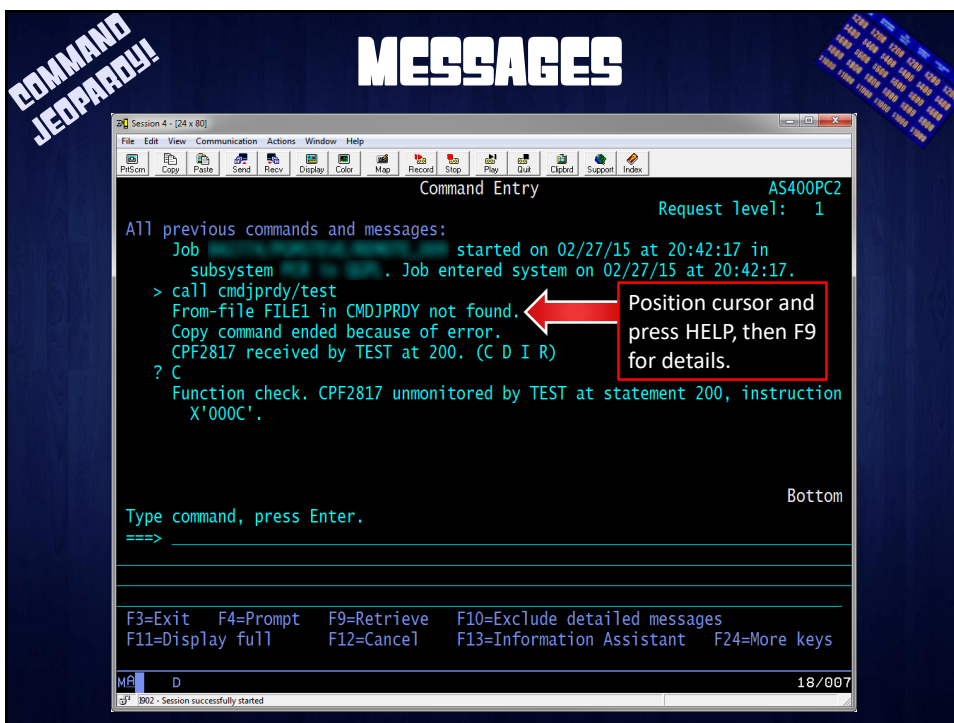
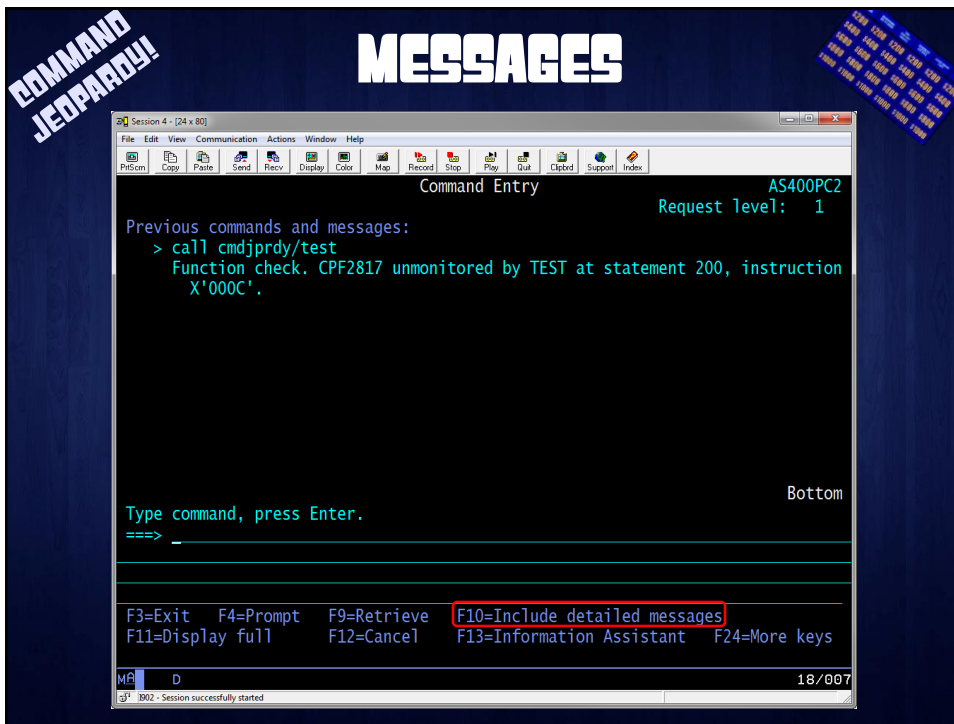
COMMAND JEOPARDY!

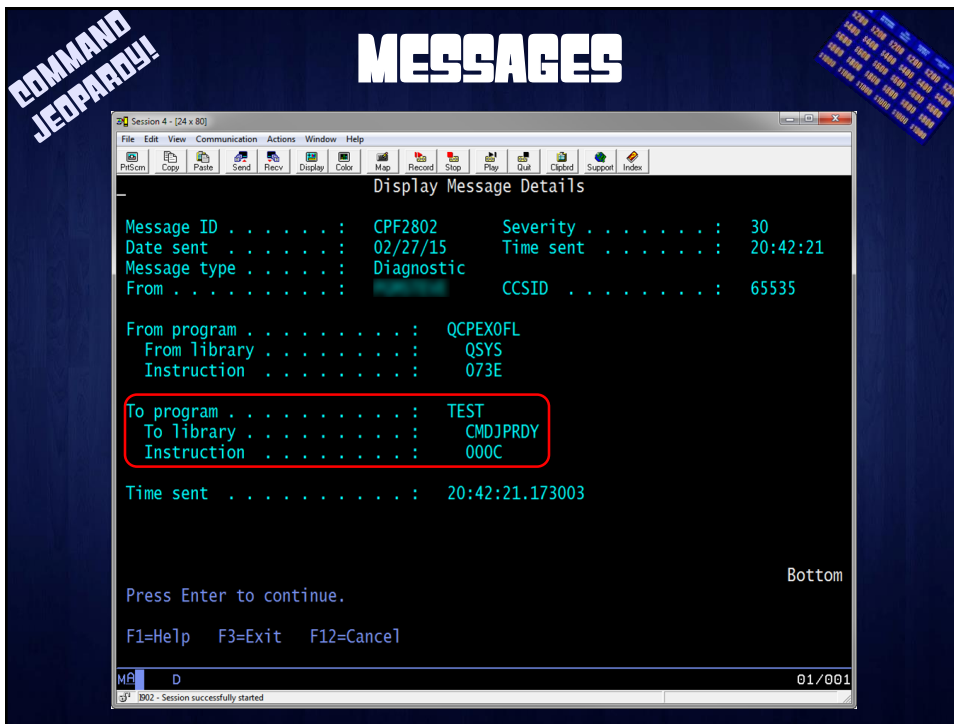
MESSAGES











COMMAND JEOPARDY!

MESSAGES

- SNDPGMMMSG** Send a message to the job's program message queue.
- RCVMSG** Receive a message from the job's program message queue into CL variables.
- RMVMSG** Removes messages from the job's program message queue.

COMMAND JEOPARDY!

MESSAGES

```
Session 4 - [24 x 80]
File Edit View Communication Actions Window Help
PrtScrn Copy Paste Send Recv Display Color Map Record Stop Play Quit Clipboard Support Index
Command Entry AS400PC2
Request Level: 1

All previous commands and messages:
Job ██████████ started on 02/27/15 at 20:42:17 in
  subsystem ██████████. Job entered system on 02/27/15 at 20:42:17.
> call cmdjprdy/test
From-file FILE1 in CMDJPRDY not found.
Copy command ended because of error.
CPFB2817 received by TEST at 200. (C D I R)
? C
Function check. CPFB2817 unmonitored by TEST at statement 200, instruction
  X'000C'.

Type command, press Enter.
===>

F3=Exit F4=Prompt F9=Retrieve F10=Exclude detailed messages
F11=Display full F12=Cancel F13=Information Assistant F24=More keys

MFA D 18/007
31 3902 - Session successfully started
```

COMMAND JEOPARDY!

MESSAGES

```
Session 4 - [24 x 80]
File Edit View Communication Actions Window Help
PrtScrn Copy Paste Send Recv Display Color Map Record Stop Play Quit Clipboard Support Index
Work with Message Descriptions (WRKMSGD)

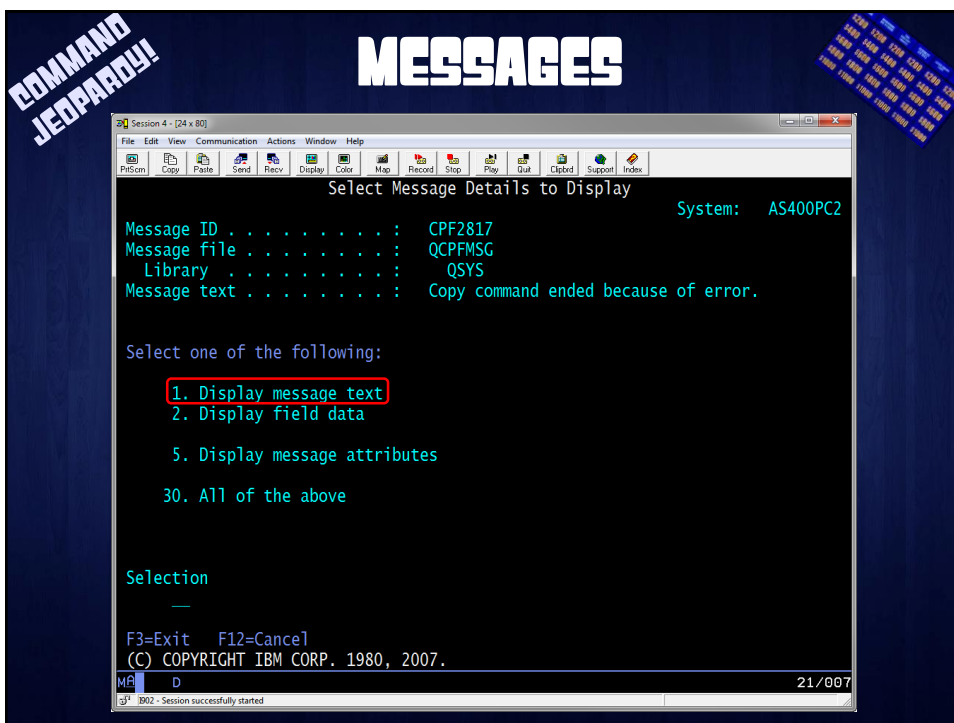
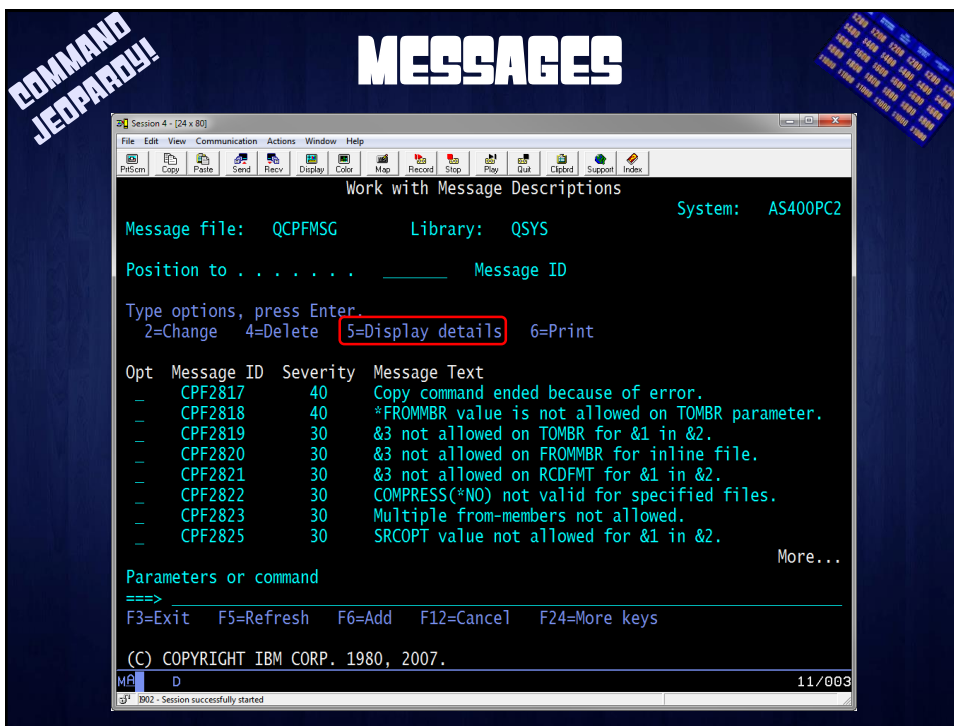
Type choices, press Enter.

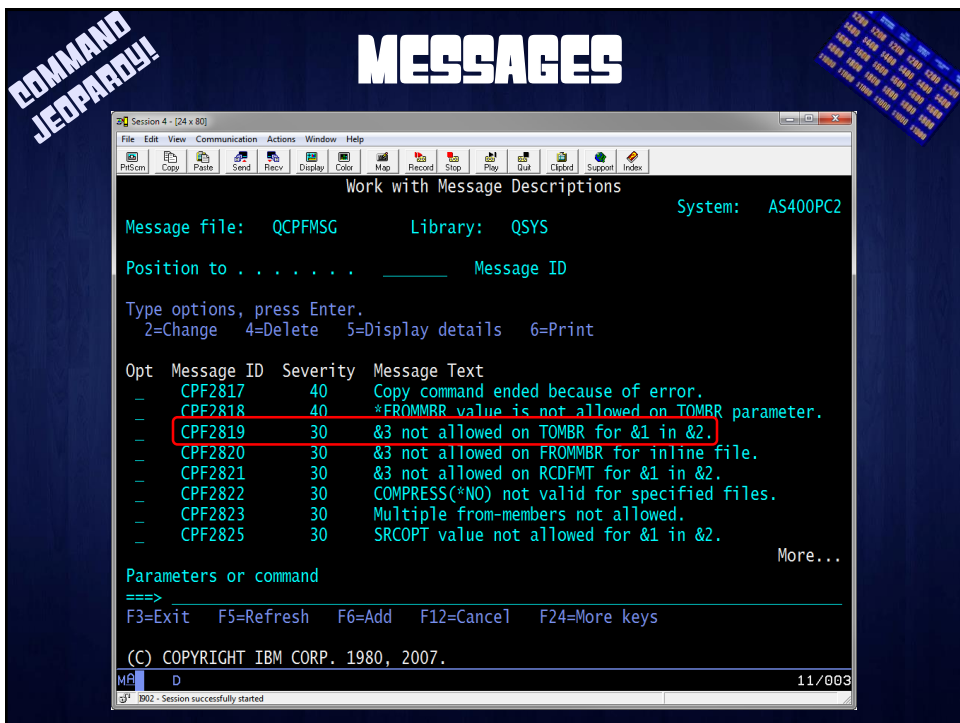
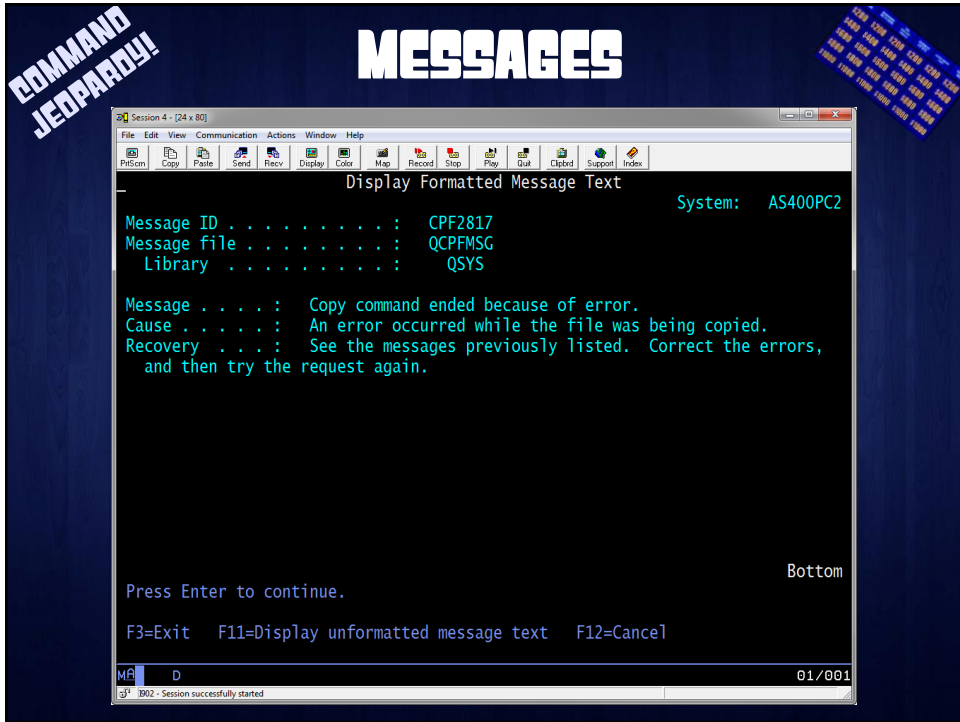
Message identifier . . . . . MSGID > CPFB2817
Message file . . . . . MSGF QCPFMMSG
Library . . . . . *LIBL

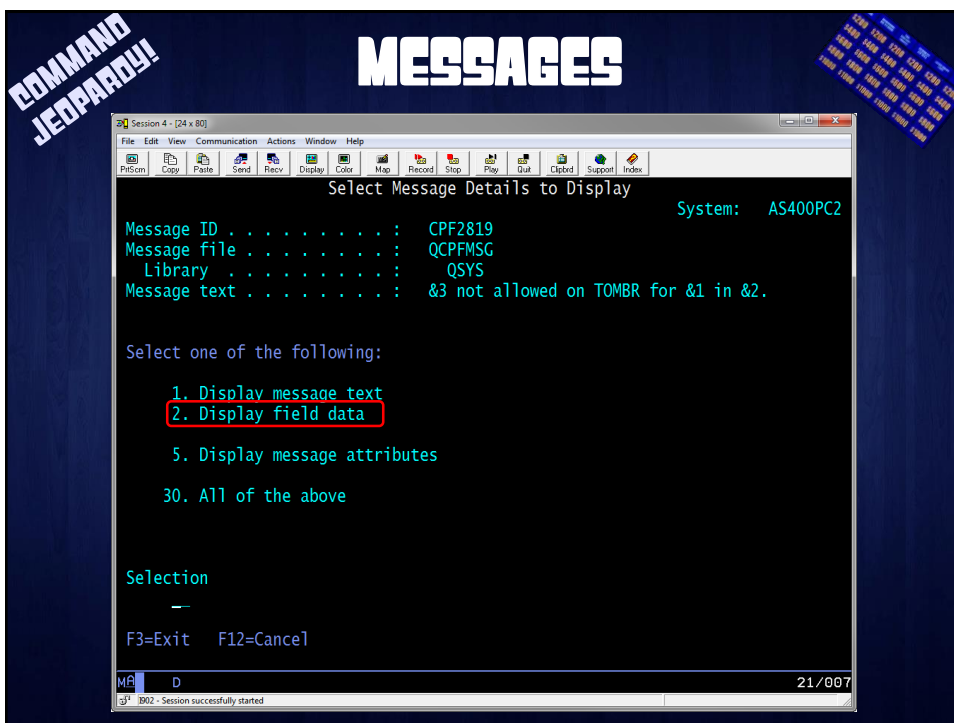
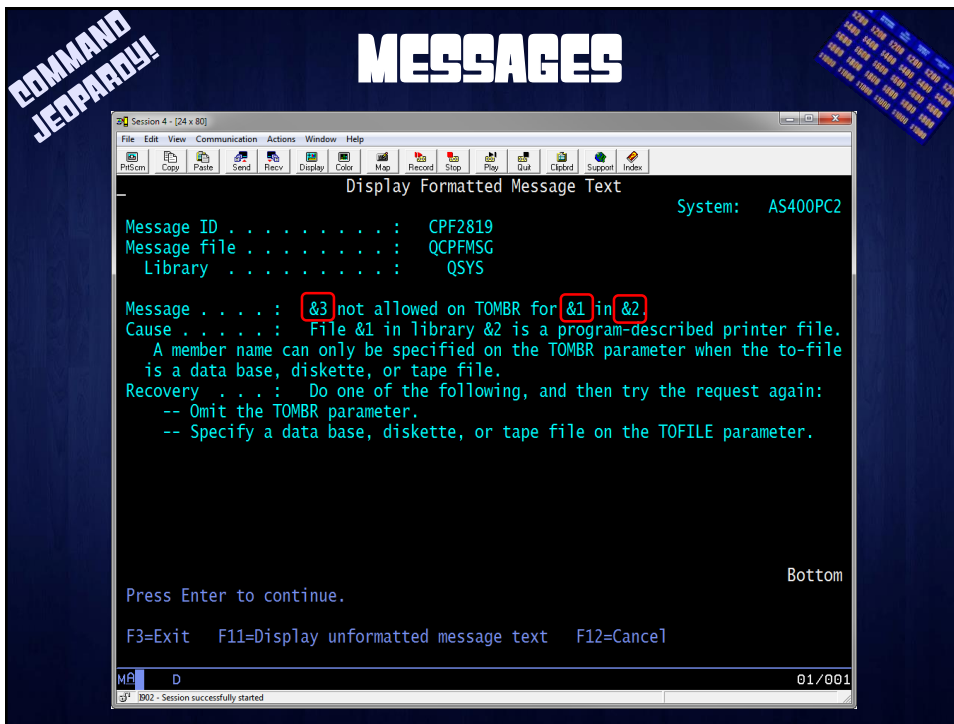
Bottom

F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys

MFA D 05/050
31 3902 - Session successfully started
```







COMMAND JEOPARDY!

MESSAGES

```

Session 4 - [24 x 80]
File Edit View Communication Actions Window Help
PrintScreen Copy Paste Send Recv Display Color Map Record Stop Play Duit Clipboard Support Index
Display Field Data

Message ID . . . . . : CPF2819
Message file . . . . . : QCPFMSG
Library . . . . . : QSYS

Field  Data Type  Length  Decimal  Vary  Dump
&1     *CHAR      10      Positions  Length  *NO
&2     *CHAR      10      *NO
&3     *CHAR      17      *NO

Press Enter to continue.

F3=Exit  F12=Cancel

MPA  D                                01/001
  
```

COMMAND JEOPARDY!

MESSAGES

```

Session 1 - [24 x 80]
File Edit View Communication Actions Window Help
PrintScreen Copy Paste Send Recv Display Color Map Record Stop Play Duit Clipboard Support Index
Columns . . . : 1 71 Edit CMDJPRDY/QCLSRC
SEU=> TEST2
***** Beginning of data *****
0001.00 PGM
0002.00 SNDPGMMSG MSGID(CPF2819) MSGF(QCPFMSG) +
0003.00 MSGDTA('AAAAAAAAABBBBBBBBBCCCCCCCCCCCCCCCC')
0005.00 ENDPGM
***** End of data *****

&1 &2 &3

Field  Data Type  Length  Decimal  Vary  Dump
&1     *CHAR      10      Positions  Length  *NO
&2     *CHAR      10      *NO
&3     *CHAR      17      *NO

MPA  A                                02/009
  
```

COMMAND JEOPARDY!

MESSAGES

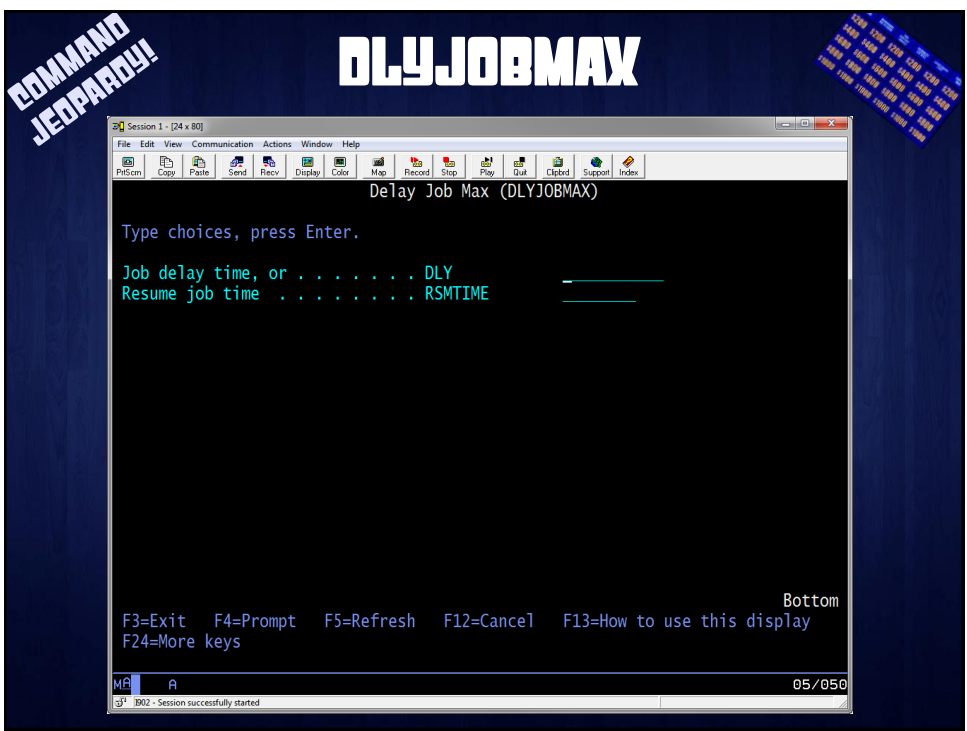
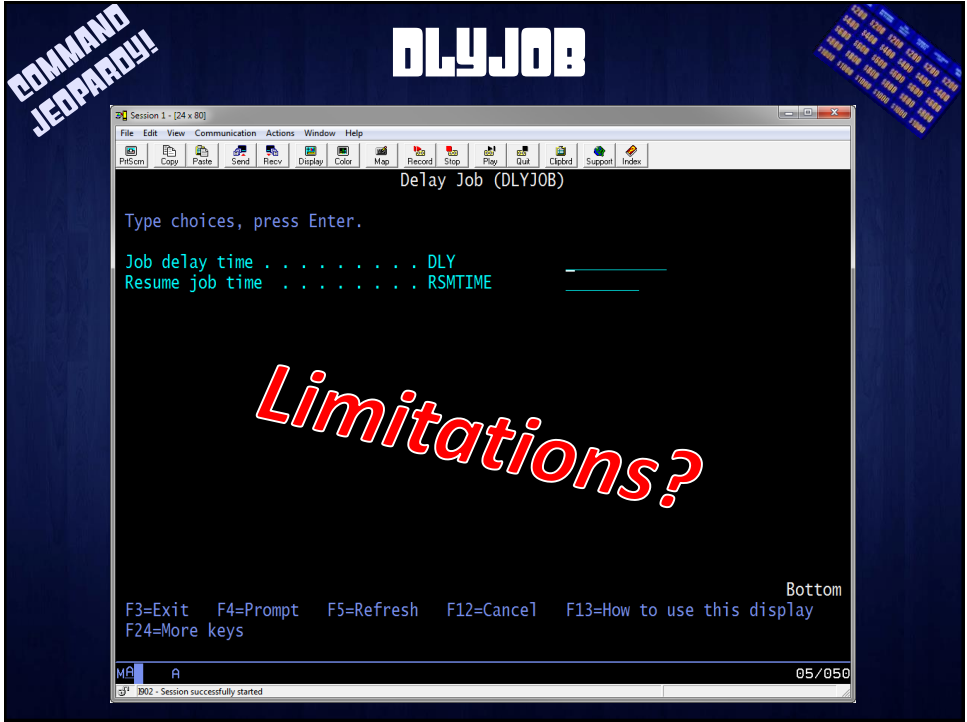
CRT/CHG/DLT/MRG/OVR/WRKMSGF
Create, change, delete, merge, override, or work with a message file.

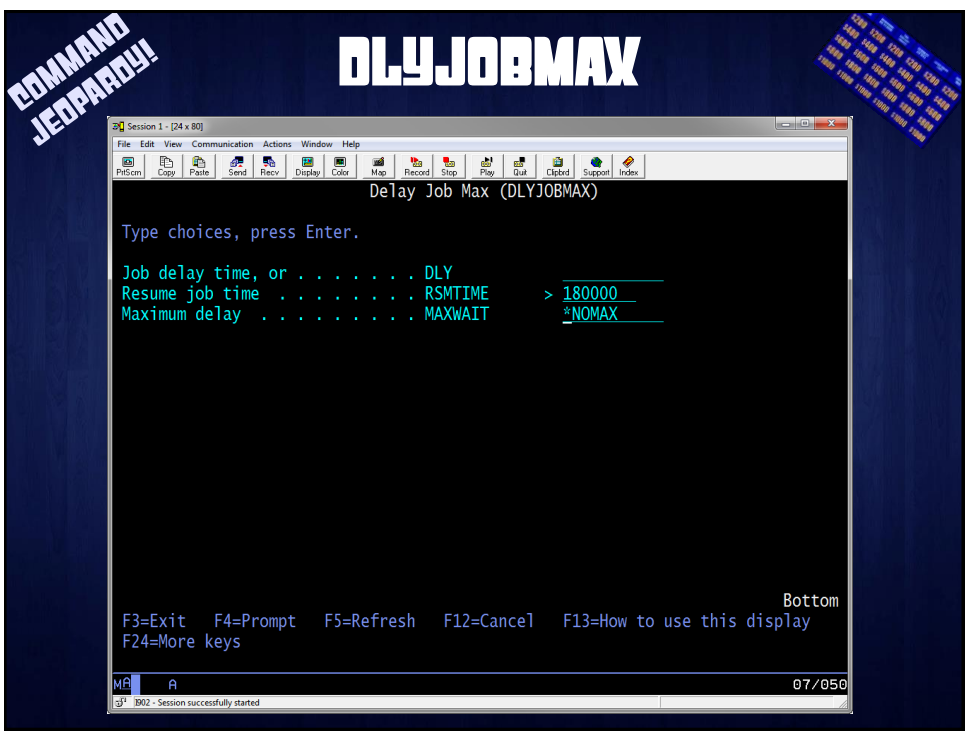
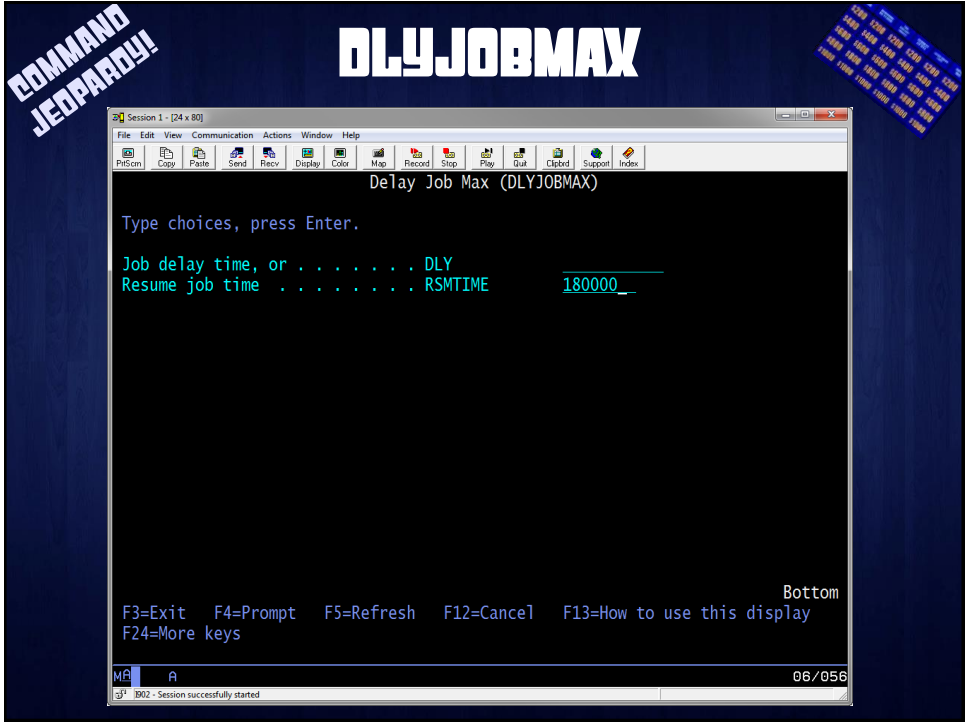
ADD/CHG/DSP/RMV/WRKMSGD
Add, change, display, remove, or work with message descriptions.

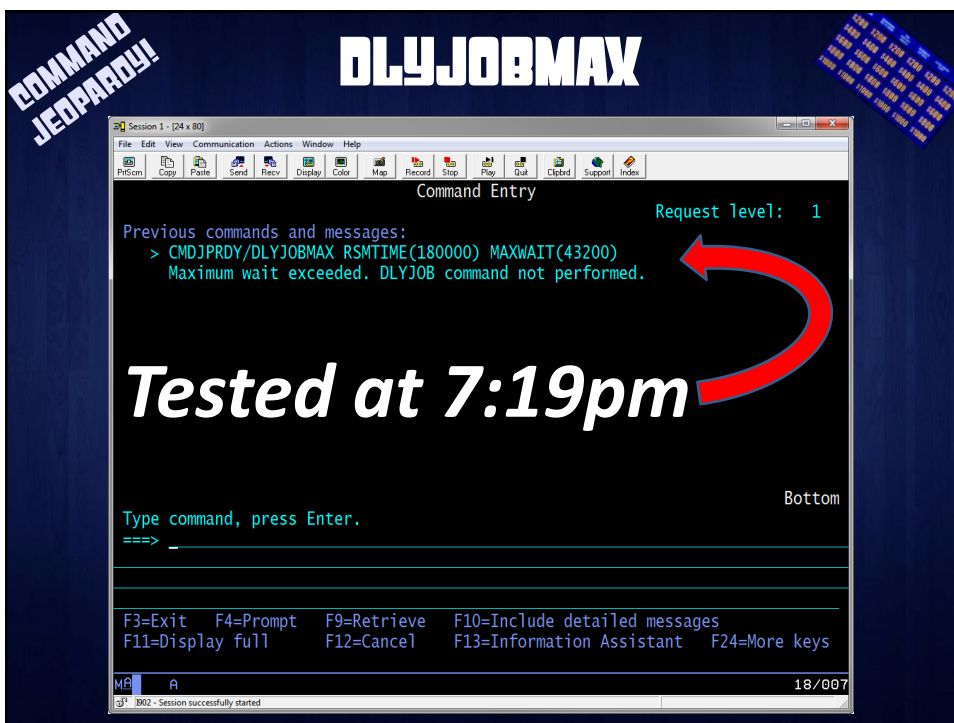
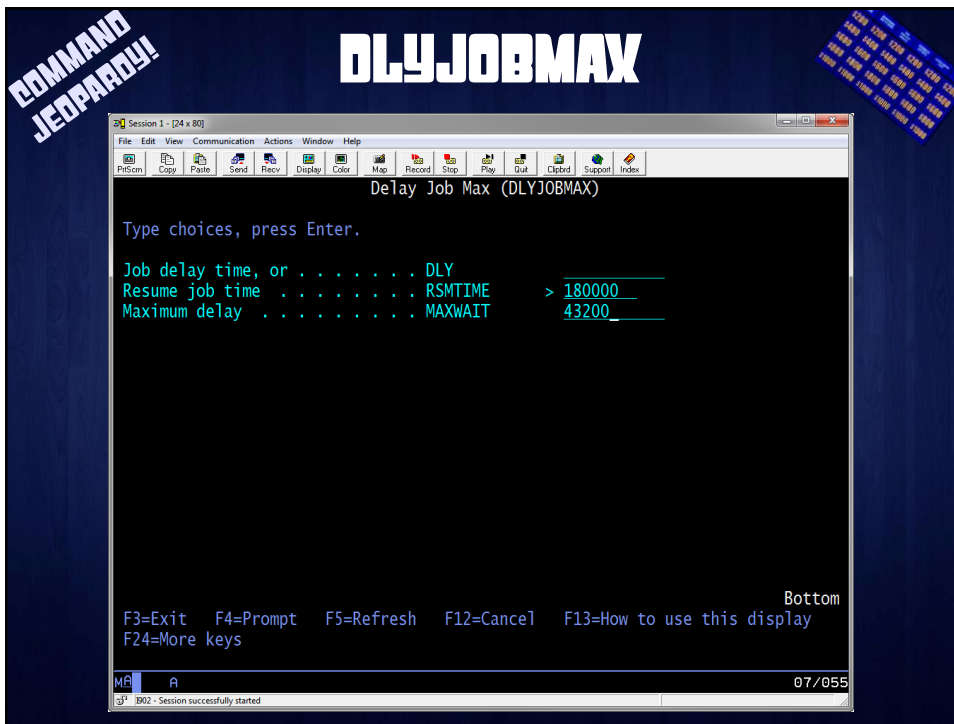
COMMAND JEOPARDY!

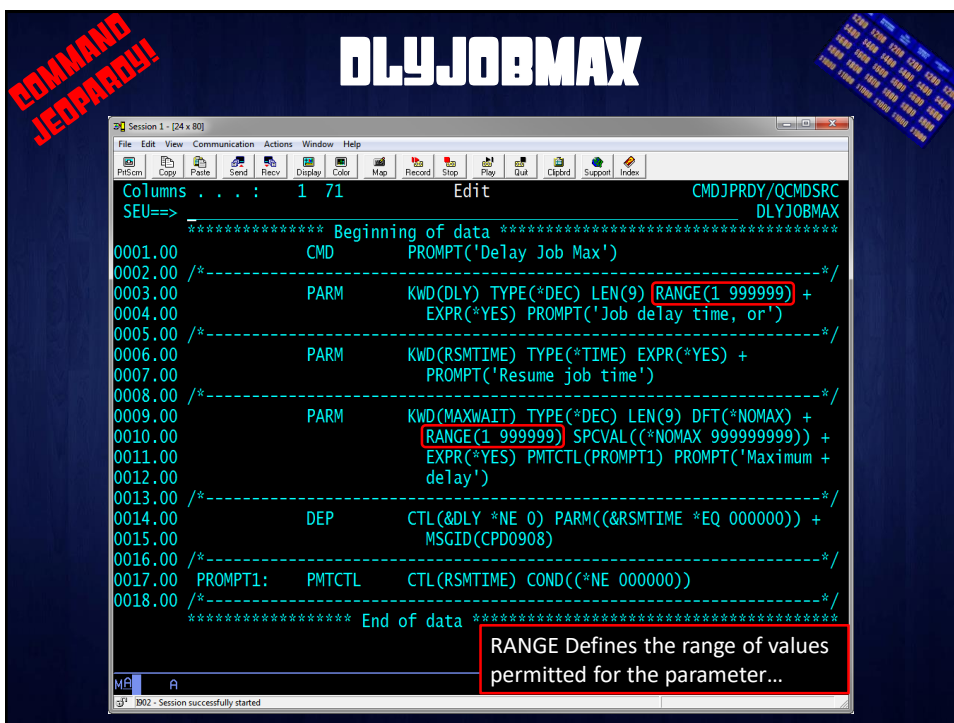
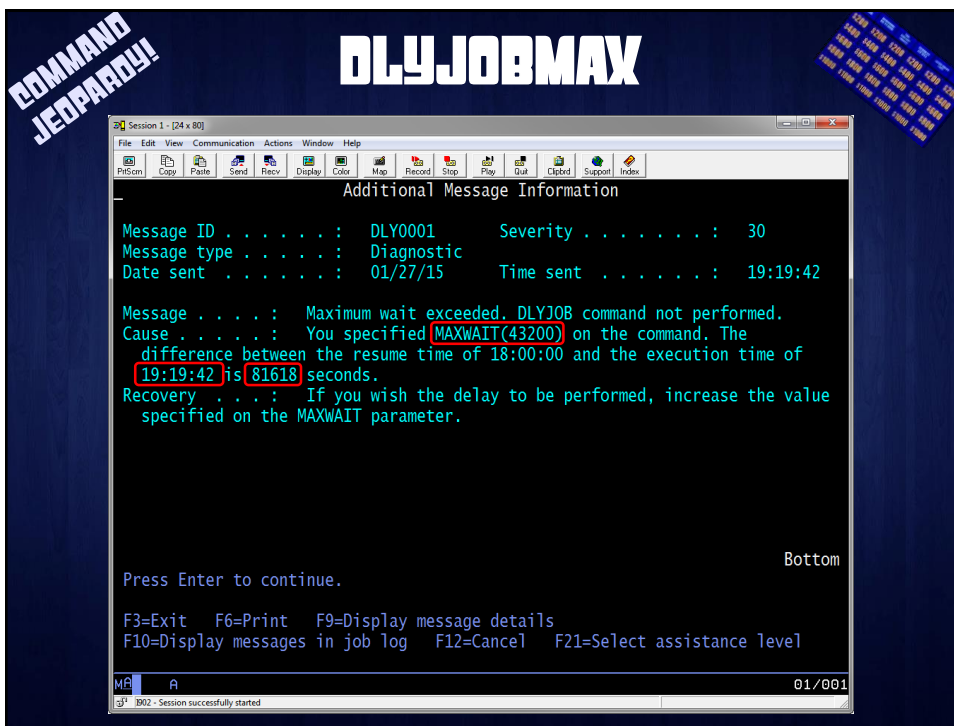
OLYJOB











COMMAND JEOPARDY!

DLYJOBMAX

```

Columns . . . : 1 71      Edit      CMDJPRDY/QCMDSRC
SEU=>
***** Beginning of data *****
0001.00      CMD      PROMPT('Delay Job Max')
0002.00 /*-----*/
0003.00      PARM      KWD(DLY) TYPE(*DEC) LEN(9) RANGE(1 999999) +
0004.00                      EXPR(*YES) PROMPT('Job delay time, or')
0005.00 /*-----*/
0006.00      PARM      KWD(RSMTIME) TYPE(*TIME) EXPR(*YES) +
0007.00                      PROMPT('Resume job time')
0008.00 /*-----*/
0009.00      PARM      KWD(MAXWAIT) TYPE(*DEC) LEN(9) DFT(*NOMAX) +
0010.00                      RANGE(1 999999) SPCVAL((*NOMAX 999999999)) +
0011.00                      EXPR(*YES) PMTCTL(PROMPT1) PROMPT('Maximum +
0012.00                      delay')
0013.00 /*-----*/
0014.00      DEP      CTL(&DLY *NE 0) PARM((&RSMTIME *EQ 000000)) +
0015.00                      MSGID(CPD0908)
0016.00 /*-----*/
0017.00 PROMPT1:  PMTCTL  CTL(RSMTIME) COND((*NE 000000))
0018.00 /*-----*/
***** End of data *****
    
```

SPCVAL defines special values to be specified for the parameter. These may or may not be the default...

COMMAND JEOPARDY!

DLYJOBMAX

```

Columns . . . : 1 71      Edit      CMDJPRDY/QCMDSRC
SEU=>
***** Beginning of data *****
0001.00      CMD      PROMPT('Delay Job Max')
0002.00 /*-----*/
0003.00      PARM      KWD(DLY) TYPE(*DEC) LEN(9) RANGE(1 999999) +
0004.00                      EXPR(*YES) PROMPT('Job delay time, or')
0005.00 /*-----*/
0006.00      PARM      KWD(RSMTIME) TYPE(*TIME) EXPR(*YES) +
0007.00                      PROMPT('Resume job time')
0008.00 /*-----*/
0009.00      PARM      KWD(MAXWAIT) TYPE(*DEC) LEN(9) DFT(*NOMAX) +
0010.00                      RANGE(1 999999) SPCVAL((*NOMAX 999999999)) +
0011.00                      EXPR(*YES) PMTCTL(PROMPT1) PROMPT('Maximum +
0012.00                      delay')
0013.00 /*-----*/
0014.00      DEP      CTL(&DLY *NE 0) PARM((&RSMTIME *EQ 000000)) +
0015.00                      MSGID(CPD0908)
0016.00 /*-----*/
0017.00 PROMPT1:  PMTCTL  CTL(RSMTIME) COND((*NE 000000))
0018.00 /*-----*/
***** End of data *****
    
```

PMTCTL allows defining under what conditions a parameter is displayed on the command prompt...

COMMAND JEOPARDY!

DLYJOBMAX

```

Columns . . . : 1 71      Edit      CMDJPRDY/QCMDSRC
SEU=>
***** Beginning of data *****
0001.00      CMD      PROMPT('Delay Job Max')
0002.00 /*-----*/
0003.00      PARM      KWD(DLY) TYPE(*DEC) LEN(9) RANGE(1 999999) +
0004.00                      EXPR(*YES) PROMPT('Job delay time, or')
0005.00 /*-----*/
0006.00      PARM      KWD(RSMTIME) TYPE(*TIME) EXPR(*YES) +
0007.00                      PROMPT('Resume job time')
0008.00 /*-----*/
0009.00      PARM      KWD(MAXWAIT) TYPE(*DEC) LEN(9) DFT(*NOMAX) +
0010.00                      RANGE(1 999999) SPCVAL((*NOMAX 999999999)) +
0011.00                      EXPR(*YES) PMTCTL(PROMPT1) PROMPT('Maximum +
0012.00                      delay')
0013.00 /*-----*/
0014.00      DEP      CTL(&DLY *NE 0) PARM((&RSMTIME *EQ 000000)) +
0015.00                      MSGID(CPD0908)
0016.00 /*-----*/
0017.00 PROMPT1:  PMTCTL  CTL(RSMTIME) COND((*NE 000000))
0018.00 /*-----*/
***** End of data *****
    
```

DEP, or Dependent Definitions, allows specifying complex validation rules.

COMMAND JEOPARDY!

DLYJOBMAX

```

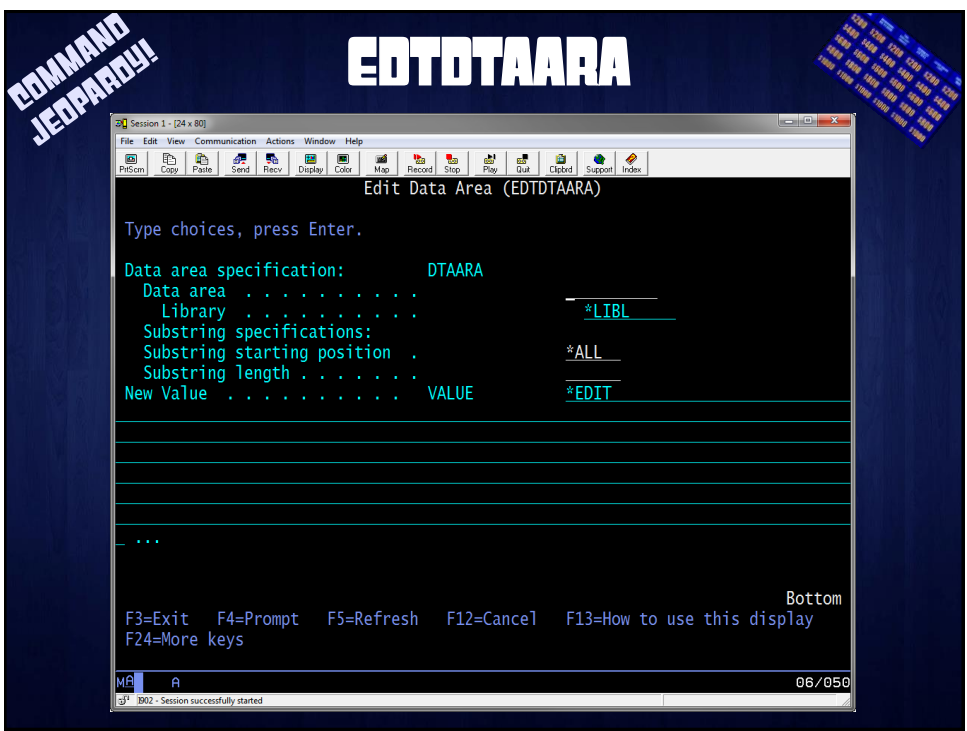
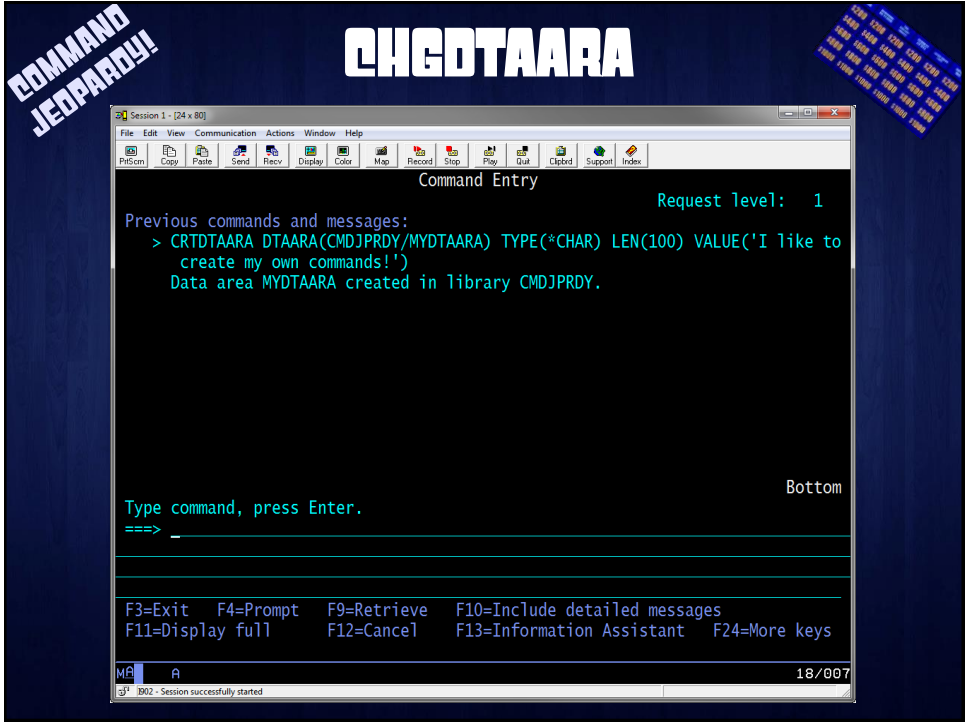
Delay Job Max (DLYJOBMAX)

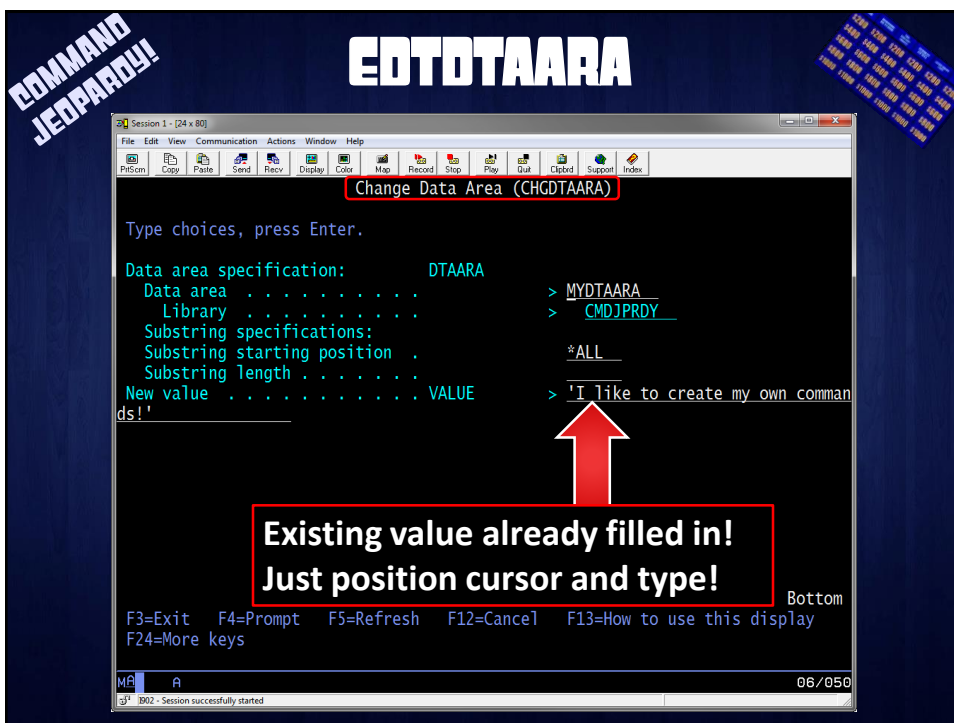
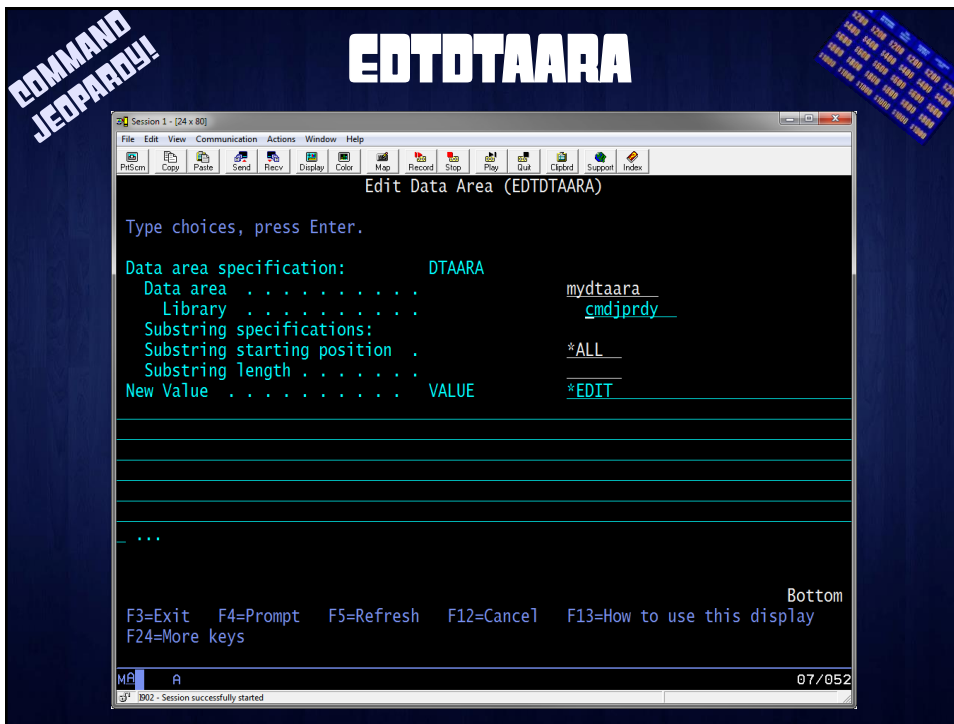
Type choices, press Enter.

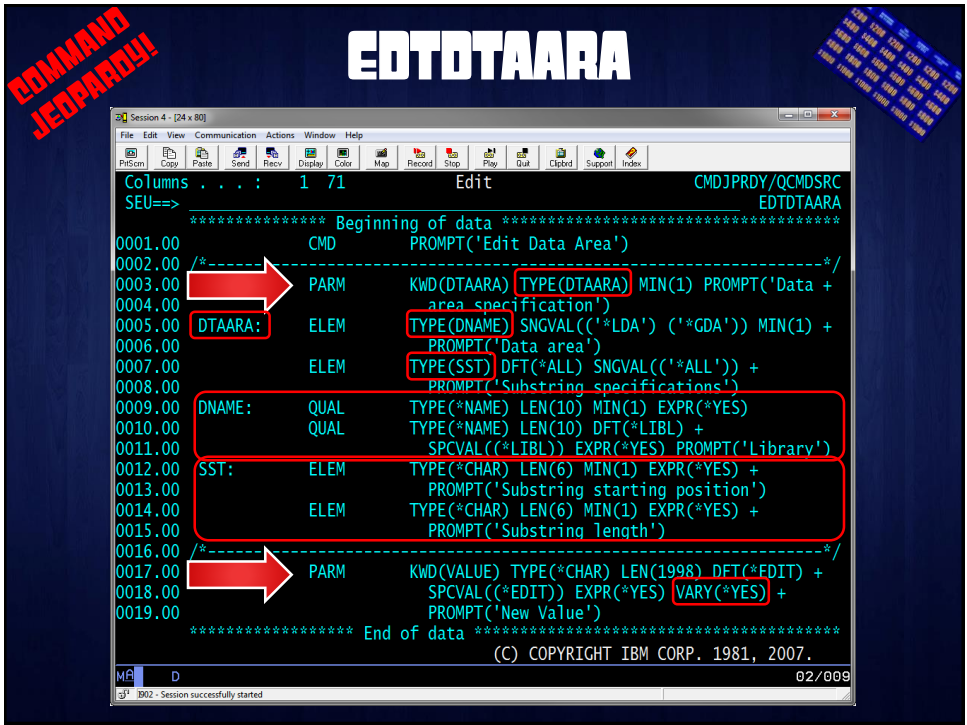
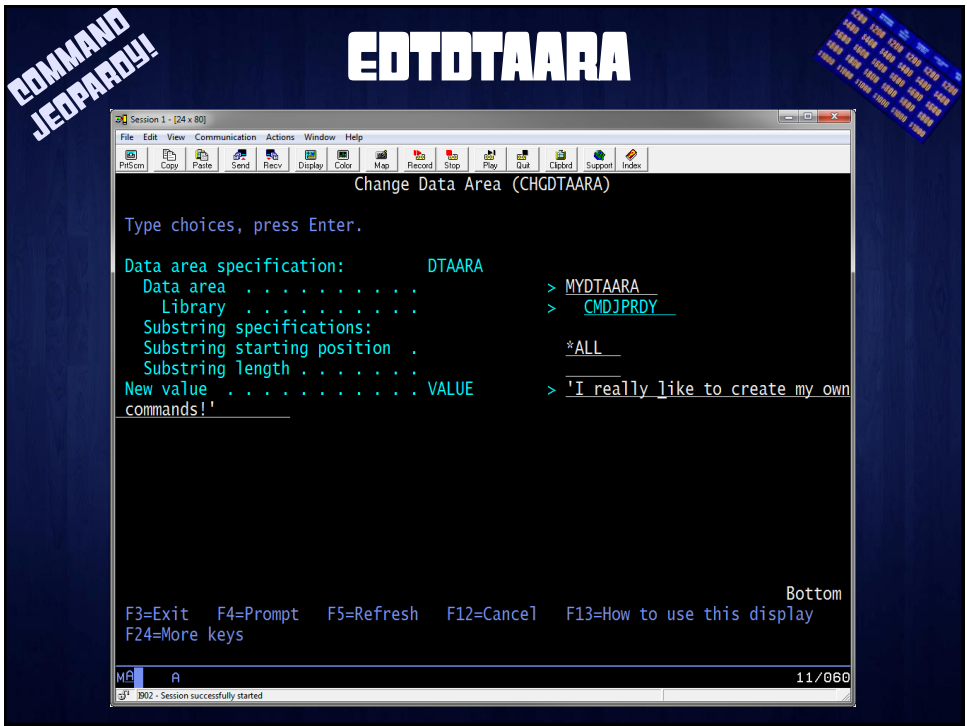
Job delay time, or . . . . . DLY      > 30
Resume job time . . . . . RSMTIME   > 180000
Maximum delay . . . . . MAXWAIT     *NOMAX
    
```

IBM supplied message description CPD0908. Sometimes we need to create our own though.

Cannot use DLY and RSMTIME parameter together.







```

CMD          PROMPT('Edit Data Area')
/*-----*/
PARM        KWD(DTAARA) TYPE(DTAARA) MIN(1) PROMPT('Data +
area specification')
DTAARA:    ELEM      TYPE(DNAME) SNGVAL(( '*LDA' ) ( '*GDA' )) MIN(1) +
            PROMPT('Data area . . . . .')
            ELEM      TYPE(SST) DFT(*ALL) SNGVAL(( '*ALL' )) +
            PROMPT('Substring specifications')
DNAME:     QUAL      TYPE(*NAME) LEN(10) MIN(1) EXPR(*YES)
            QUAL      TYPE(*NAME) LEN(10) DFT(*LIBL) +
            SPCVAL(( '*LIBL' )) EXPR(*YES) PROMPT('Library +
            ')
SST:       ELEM      TYPE(*CHAR) LEN(6) MIN(1) EXPR(*YES) +
            PROMPT('Substring starting position . . .')
            ELEM      TYPE(*CHAR) LEN(6) MIN(1) EXPR(*YES) +
            PROMPT('Substring length . . . . .')
/*-----*/
PARM        KWD(VALUE) TYPE(*CHAR) LEN(1998) DFT(*EDIT) +
            SPCVAL(( '*EDIT' )) EXPR(*YES) VARY(*YES) +
            PROMPT('New Value . . . . .')

PGM         PARM(&NAMELIB &STRING)
DCL         VAR(&NAMELIB) TYPE(*CHAR) LEN(38)
DCL         VAR(&STRING) TYPE(*CHAR) LEN(2000)
DCL         VAR(&VALUE) TYPE(*CHAR) LEN(2000)
DCL         VAR(&NAME) TYPE(*CHAR) LEN(10)
DCL         VAR(&LIB) TYPE(*CHAR) LEN(10)
DCL         VAR(&STRLEN) TYPE(*CHAR) LEN(2)
DCL         VAR(&SLEN) TYPE(*DEC) LEN(5 0)
DCL         VAR(&START) TYPE(*CHAR) LEN(6)
DCL         VAR(&LEN) TYPE(*CHAR) LEN(6)

CHGVAR     VAR(&STRLEN) VALUE(%SST(&STRING 1 2))
CHGVAR     VAR(&SLEN) VALUE(%BIN(&STRLEN))
IF         COND(&SLEN *GT 0) THEN(CHGVAR VAR(&VALUE) +
            VALUE(%SST(&STRING 3 &SLEN)))

CHGVAR     VAR(&NAME) VALUE(%SST(&NAMELIB 3 10))
CHGVAR     VAR(&LIB) VALUE(%SST(&NAMELIB 13 10))
CHGVAR     VAR(&START) VALUE(%SST(&NAMELIB 27 6))
CHGVAR     VAR(&LEN) VALUE(%SST(&NAMELIB 33 6))
    
```

```

CMD          PROMPT('Edit Data Area')
/*-----*/
PARM        KWD(DTAARA) TYPE(DTAARA) MIN(1) PROMPT('Data +
area specification')
DTAARA:    ELEM      TYPE(DNAME) SNGVAL(( '*LDA' ) ( '*GDA' )) MIN(1) +
            PROMPT('Data area . . . . .')
            ELEM      TYPE(SST) DFT(*ALL) SNGVAL(( '*ALL' )) +
            PROMPT('Substring specifications')
DNAME:     QUAL      TYPE(*NAME) LEN(10) MIN(1) EXPR(*YES)
            QUAL      TYPE(*NAME) LEN(10) DFT(*LIBL) +
            SPCVAL(( '*LIBL' )) EXPR(*YES) PROMPT('Library +
            ')
SST:       ELEM      TYPE(*CHAR) LEN(6) MIN(1) EXPR(*YES) +
            PROMPT('Substring starting position . . .')
            ELEM      TYPE(*CHAR) LEN(6) MIN(1) EXPR(*YES) +
            PROMPT('Substring length . . . . .')
/*-----*/
PARM        KWD(VALUE) TYPE(*CHAR) LEN(1998) DFT(*EDIT) +
            SPCVAL(( '*EDIT' )) EXPR(*YES) VARY(*YES) +
            PROMPT('New Value . . . . .')

PGM         PARM(&NAMELIB &STRING)
DCL         VAR(&NAMELIB) TYPE(*CHAR) LEN(38)
DCL         VAR(&STRING) TYPE(*CHAR) LEN(2000)
DCL         VAR(&VALUE) TYPE(*CHAR) LEN(2000)
DCL         VAR(&NAME) TYPE(*CHAR) LEN(10)
DCL         VAR(&LIB) TYPE(*CHAR) LEN(10)
DCL         VAR(&STRLEN) TYPE(*CHAR) LEN(2)
DCL         VAR(&SLEN) TYPE(*DEC) LEN(5 0)
DCL         VAR(&START) TYPE(*CHAR) LEN(6)
DCL         VAR(&LEN) TYPE(*CHAR) LEN(6)

CHGVAR     VAR(&STRLEN) VALUE(%SST(&STRING 1 2))
CHGVAR     VAR(&SLEN) VALUE(%BIN(&STRLEN))
IF         COND(&SLEN *GT 0) THEN(CHGVAR VAR(&VALUE) +
            VALUE(%SST(&STRING 3 &SLEN)))

CHGVAR     VAR(&NAME) VALUE(%SST(&NAMELIB 3 10))
CHGVAR     VAR(&LIB) VALUE(%SST(&NAMELIB 13 10))
CHGVAR     VAR(&START) VALUE(%SST(&NAMELIB 27 6))
CHGVAR     VAR(&LEN) VALUE(%SST(&NAMELIB 33 6))
    
```

COMMAND JEOPARDY!

EDTOTAARA

CPP Excerpt...

```

?      CHGDTAARA ??DTAARA(&NAME) ??VALUE(&DTA)
    
```

Selective prompting:
 ?? = Allow editing this parameter
 Blank = Do not allow editing this parameter
 ?- = Do not display this parameter

Forces Prompting of the Command

COMMAND JEOPARDY!

EDTOTAARA

```

PGM
?      CHGDTAARA DTAARA(CMDJPRDY/MYDTAARA *ALL) +
      VALUE('Selective Prompting is Fun')
ENDPGM
    
```

Nothing is editable!

COMMAND JEOPARDY!

EDTDTAARA

```

PGM
?      CHGDTAARA ??DTAARA(CMDJPRDY/MYDTAARA *ALL) +
      ??VALUE('Selective Prompting is Fun')
ENDPGM

```

Everything is editable!

COMMAND JEOPARDY!

EDTDTAARA

```

PGM
?      CHGDTAARA ?-DTAARA(CMDJPRDY/MYDTAARA *ALL) +
      ??VALUE('Selective Prompting is Fun')
ENDPGM

```

We've hidden the data area name parameter, but allowed editing the new value of the data area!



```

0004.00      DCL      VAR(&ERRMSG)      TYPE(*CHAR)  LEN(512)
0005.00      DCL      VAR(&ERRMSGID)   TYPE(*CHAR)  LEN(7)
0006.00      DCL      VAR(&ERRMSGDTA)  TYPE(*CHAR)  LEN(512)
0007.00      DCL      VAR(&ERRMSGF)    TYPE(*CHAR)  LEN(10)
0008.00      DCL      VAR(&ERRMSGFLIB) TYPE(*CHAR)  LEN(10)
0009.00      DCL      VAR(&ERRQUIT)    TYPE(*LGL)
0010.00      MONMSG  MSGID(CPF0000)  EXEC(GOTO  CMDLBL(ERROR))
0011.00      /*****
0012.00      ERROR:   IF      COND(&ERRQUIT) THEN(SNDPGMMSG MSGID(CPF9999) +
0013.00                MSGF(QCPFMSG) MSGTYPE(*ESCAPE)) /* Func chk*/
0014.00                CHGVAR  VAR(&ERRQUIT) VALUE('1')
0015.00      DIAG:   RCVMSG  MSGTYPE(*DIAG) MSG(&ERRMSG) +
0016.00                MSGDTA(&ERRMSGDTA) MSGID(&ERRMSGID) +
0017.00                MSGF(&ERRMSGF)  MSGFLIB(&ERRMSGFLIB)
0018.00                IF      COND(&ERRMSG = ' ') THEN(GOTO  CMDLBL(ESCAPE))
0019.00                IF      COND(&ERRMSGID *NE ' ') THEN(SNDPGMMSG +
0020.00                MSGID(&ERRMSGID) MSGF(&ERRMSGFLIB/&ERRMSGF) +
0021.00                MSGDTA(&ERRMSGDTA) MSGTYPE(*DIAG))
0022.00                ELSE    CMD(SNDPGMMSG MSG(&ERRMSG) MSGTYPE(*DIAG))
0023.00                GOTO    CMDLBL(DIAG)
0024.00      ESCAPE: RCVMSG  MSGTYPE(*EXCP) MSG(&ERRMSG) +
0025.00                MSGDTA(&ERRMSGDTA) MSGID(&ERRMSGID) +
0026.00                MSGF(&ERRMSGF)  MSGFLIB(&ERRMSGFLIB)
0027.00                IF      COND(&ERRMSG *NE ' ') *AND (&ERRMSGID *NE +
0028.00                'CPF0001')) THEN(SNDPGMMSG +
0029.00                MSGID(&ERRMSGID) +
0030.00                MSGF(&ERRMSGFLIB/&ERRMSGF) +
0031.00                MSGDTA(&ERRMSGDTA) MSGTYPE(*DIAG))
0032.00                RMVMSG  CLEAR(*ALL)
0033.00                SNDPGMMSG MSGID(CPF0001) MSGF(QSYS/QCPFMSG) +
0034.00                MSGDTA(STDERR) MSGTYPE(*ESCAPE)
    
```

OLD STANDBY

```

0004.00      DCL      VAR(&ERRMSG)      TYPE(*CHAR)  LEN(512)
0005.00      DCL      VAR(&ERRMSGID)     TYPE(*CHAR)  LEN(7)
0006.00      DCL      VAR(&ERRMSGDTA)    TYPE(*CHAR)  LEN(512)
0007.00      DCL      VAR(&ERRMSGF)      TYPE(*CHAR)  LEN(10)
0008.00      DCL      VAR(&ERRMSGFLIB)   TYPE(*CHAR)  LEN(10)
0009.00      DCL      VAR(&ERRQUIT)      TYPE(*LGL)
0010.00      MONMSG  MSGID(CPF0000)     EXEC(GOTO  CMDLBL(ERROR))
0011.00      /*****
0012.00      ERROR:   IF      COND(&ERRQUIT) THEN(SNDPGMMSG MSGID(CPF9999) +
0013.00                MSGF(QCPFMSG) MSGTYPE(*ESCAPE)) /* Func chk*/
0014.00      CHGVAR  VAR(&ERRQUIT) VALUE('1')
0015.00      DIAG:   RCVMSG  MSGTYPE(*DIAG) MSG(&ERRMSG) +
0016.00                MSGDTA(&ERRMSGDTA) MSGID(&ERRMSGID) +
0017.00                MSGF(&ERRMSGF) MSGFLIB(&ERRMSGFLIB)
0018.00                IF      COND(&ERRMSG = ' ') THEN(GOTO  CMDLBL(ESCAPE))
0019.00                IF      COND(&ERRMSGID *NE ' ') THEN(SNDPGMMSG +
0020.00                MSGID(&ERRMSGID) MSGF(&ERRMSGFLIB/&ERRMSGF) +
0021.00                MSGDTA(&ERRMSGDTA) MSGTYPE(*DIAG))
0022.00      ELSE    CMD(SNDPGMMSG MSG(&ERRMSG) MSGTYPE(*DIAG))
0023.00      GOTO    CMDLBL(DIAG)
0024.00      ESCAPE:  RCVMSG  MSGTYPE(*EXCP) MSG(&ERRMSG) +
0025.00                MSGDTA(&ERRMSGDTA) MSGID(&ERRMSGID) +
0026.00                MSGF(&ERRMSGF) MSGFLIB(&ERRMSGFLIB)
0027.00                IF      COND((&ERRMSG *NE ' ') *AND (&ERRMSGID *NE +
0028.00                'CPF0001')) THEN(SNDPGMMSG +
0029.00                MSGID(&ERRMSGID) +
0030.00                MSGF(&ERRMSGFLIB/&ERRMSGF) +
0031.00                MSGDTA(&ERRMSGDTA) MSGTYPE(*DIAG))
0032.00      RMVMSG  CLEAR(*ALL)
0033.00      SNDPGMMSG MSGID(CPF0001) MSGF(QSYS/QCPFMSG) +
0034.00                MSGDTA(STDERR) MSGTYPE(*ESCAPE)
    
```

OLD STANDBY

Replace STDERR with name of your command

ERROR HANDLING

```

Columns . . . : 1 71      Edit      CMDJPRDY/QCLSRC
SEU=>          STDERR2
***** Beginning of data *****
0001.00 /*****
0002.00 /***** SIMPLE ERROR ROUTINE - RESEND ALL DIAG AND ESCAPE MSGS *****/
0003.00 /*****
0004.00      DCL      VAR(&ERRCODE)      TYPE(*CHAR)  LEN(16) +
0005.00                VALUE(X'0000000000000010')
0006.00      DCL      VAR(&ERRQUIT)    TYPE(*LGL)
0007.00      MONMSG  MSGID(CPF0000) EXEC(GOTO  CMDLBL(ERROR))
0008.00 /*****
0009.00      ERROR:   IF      COND(&ERRQUIT) THEN(SNDPGMMSG MSGID(CPF9999) +
0010.00                MSGF(QCPFMSG) MSGTYPE(*ESCAPE)) /* Func chk*/
0011.00      CHGVAR  VAR(&ERRQUIT) VALUE('1')
0012.00      CALL   PGM(QMHMOVPM) PARM(' ' '*DIAG ' ' +
0013.00                X'00000001' '* ' ' X'00000001' +
0014.00                &ERRCODE)
0015.00      MONMSG  MSGID(CPF24A3) EXEC(RCVMSG MSGTYPE(*LAST))
0016.00      CALL   PGM(QMHRSNEM) PARM(' ' &ERRCODE)
0017.00      ENDPGM
***** End of data *****
    
```

COMMAND JEOPARDY!

SIMPLE API

```

0004.00      DCL      VAR(&ERRMSG)      TYPE(*CHAR)  LEN(512)
0005.00      DCL      VAR(&ERRMSGID)   TYPE(*CHAR)  LEN(7)
0006.00      DCL      VAR(&ERRMSGDTA)  TYPE(*CHAR)  LEN(512)
0007.00      DCL      VAR(&ERRMSGF)    TYPE(*CHAR)  LEN(10)
0008.00      DCL      VAR(&ERRMSGFLIB) TYPE(*CHAR)  LEN(10)
0009.00      DCL      VAR(&ERRQUIT)    TYPE(*LGL)
0010.00      DCL      VAR(&ERRKEY)     TYPE(*CHAR)  LEN(4)
0011.00      DCL      VAR(&ERRCODE)    TYPE(*CHAR)  LEN(16) +
0012.00      VALUE(X'0000000000000010')
0013.00      MONMSG  MSGID(CPF0000)  EXEC(GOTO CMDLBL(ERROR))
0014.00      /*****
0015.00      ERROR:   IF      COND(&ERRQUIT) THEN(SNDPGMSG MSGID(CPF9999) +
0016.00      MSGF(QCPFMSG) MSGTYPE(*ESCAPE)) /* Func chk*/
0017.00      CHGVAR  VAR(&ERRQUIT) VALUE('1')
0018.00      DIAG:   RCVMSG  MSGTYPE(*DIAG) RMV(*NO) KEYVAR(&ERRKEY) +
0019.00      MSG(&ERRMSG) MSGDTA(&ERRMSGDTA) +
0020.00      MSGID(&ERRMSGID) MSGF(&ERRMSGF) +
0021.00      MSGFLIB(&ERRMSGFLIB)
0022.00      IF      COND(&ERRMSG = ' ') THEN(GOTO CMDLBL(ESCAPE))
0023.00      IF      COND(&SEND_IT) THEN(DO)
0024.00      CALL    PGM(QMHMOVPM) PARM(&ERRKEY ' ' *DIAG ' ' +
0025.00      X'00000001' '* ' X'00000001' +
0026.00      &ERRCODE)
0027.00      MONMSG  MSGID(CPF24A3) EXEC(RCVMSG MSGTYPE(*LAST))
0028.00      ENDDO
0029.00      ELSE    CMD(RMVMSG MSGKEY(&ERRKEY))
0030.00      GOTO    CMDLBL(DIAG)
0031.00      ESCAPE:  RCVMSG  MSGTYPE(*EXCP) RMV(*NO) KEYVAR(&ERRKEY) +
0032.00      MSG(&ERRMSG) MSGDTA(&ERRMSGDTA) +
0033.00      MSGID(&ERRMSGID) MSGF(&ERRMSGF) +
0034.00      MSGFLIB(&ERRMSGFLIB)
0035.00      IF      COND((&ERRMSG *NE ' ') *AND (&ERRMSGID *NE +
0036.00      'CPF0001')) THEN(CALL PGM(QMHRSNEM) +
0037.00      PARM(&ERRKEY &ERRCODE))
0038.00      RMVMSG  MSGKEY(&ERRKEY)
0039.00      SNDPGMSG MSGID(CPF0001) MSGF(QSYS/QCPFMSG) +
0040.00      MSGDTA(STDERR) MSGTYPE(*ESCAPE)
    
```

ERROR HANDLING

COMMAND JEOPARDY!

API FOR ALL BUT CPF0001

COMMAND JEOPARDY!

OUR FINAL COMMAND TOGETHER...

RTV OBJ



COMMAND JEOPARDY!

RTV OBJ

Retrieve Object (RTV OBJ)

Type choices, press Enter.

Label		
Object	OBJ	
Library		*LIBL
Object type	TYPE	
Object attribute	ATTR	
CL var for RTNOBJ	(10) ..	RTNOBJ	
CL var for RTNLIB	(10) ..	RTNLIB	
CL var for RTNTYP	(7) ..	RTNTYP	
Comment		

} Input Params
} Returned Data

- Returns data, so can only be executed inside a program.
- How do we get the rest of the objects returned? Hint: F11

Bottom

F3=Exit F4=Prompt F5=Refresh F10=Additional parameters F12=Cancel
 F13=How to use this display F24=More keys

MFA B 05/050

31 3902 - Session successfully started

COMMAND JEOPARDY!

RTVOBJ

```

Session 2 - [24 x 80]
File Edit View Communication Actions Window Help
Retrieve Object (RTVOBJ)

Type choices, press Enter.

Label . . . . .
Object . . . . . Name, generic*, *CONTINUE...
Library . . . . . *LIBL Name, *LIBL, *USRLIBL...
Object type . . . . . *ALL, *ALRTBL, *AUTL...
Object attribute . . . . . *ALL Name, generic*, *ALL
CL var for RTNOBJ (10) . . . . . Character value
CL var for RTNLIB (10) . . . . . Character value
CL var for RTNTYP (7) . . . . . Character value
Comment . . . . .

Bottom
F3=Exit F4=Prompt F5=Refresh F10=Additional parameters F12=Cancel
F13=How to use this display F24=More keys

MFA B 05/037
31 3902 - Session successfully started
    
```

- Blanks are returned when we are out of objects.
- Where is the list of objects stored? Hint: F10

COMMAND JEOPARDY!

RTVOBJ

```

Session 2 - [24 x 80]
File Edit View Communication Actions Window Help
Retrieve Object (RTVOBJ)

Type choices, press Enter.

Label . . . . .
Object . . . . . Name, generic*, *CONTINUE...
Library . . . . . *LIBL Name, *LIBL, *USRLIBL...
Object type . . . . . *ALL, *ALRTBL, *AUTL...
Object attribute . . . . . *ALL Name, generic*, *ALL
CL var for RTNOBJ (10) . . . . . Character value
CL var for RTNLIB (10) . . . . . Character value
CL var for RTNTYP (7) . . . . . Character value

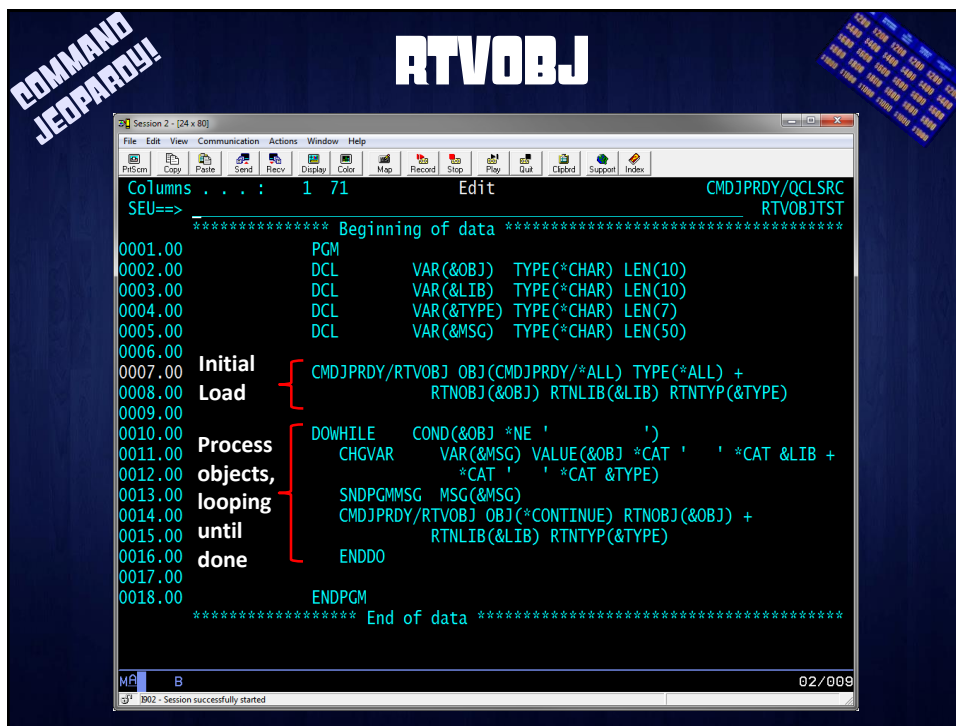
Additional Parameters

User space . . . . . RTVOBJ Name
Library . . . . . QTEMP Name, *CURLIB

Controlled by Prompt Request

More...
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys

MFA B 05/037
31 3902 - Session successfully started
    
```

COMMAND JEOPARDY!

RTVOBJ

```

Columns . . . : 1 71      Edit      CMDJPRDY/QCLSRC
SEU=>
***** Beginning of data *****
0001.00      PGM
0002.00      DCL          VAR(&OBJ)  TYPE(*CHAR) LEN(10)
0003.00      DCL          VAR(&LIB)  TYPE(*CHAR) LEN(10)
0004.00      DCL          VAR(&TYPE) TYPE(*CHAR) LEN(7)
0005.00      DCL          VAR(&MSG)  TYPE(*CHAR) LEN(50)
0006.00
0007.00      CMDJPRDY/RTVOBJ OBJ(CMDJPRDY/*ALL) TYPE(*PGM) +
0008.00          ATTR(RPGLE) RTNOBJ(&OBJ) RTNLIB(&LIB) +
0009.00          RINTYP(&TYPE)
0010.00
0011.00      DOWHILE      COND(&OBJ *NE ' ')
0012.00      CHGVAR      VAR(&MSG) VALUE(&OBJ *CAT ' ' *CAT &LIB +
0013.00          *CAT ' ' *CAT &TYPE)
0014.00
0015.00      SNDPGMMSG  MSG(&MSG)
0016.00      CMDJPRDY/RTVOBJ OBJ(*CONTINUE) ATTR(RPGLE) +
0017.00          RTNOBJ(&OBJ) RTNLIB(&LIB) RINTYP(&TYPE)
0018.00      ENDDO
0019.00      ENDPGM
***** End of data *****
(C) COPYRIGHT IBM CORP. 1981, 2007.
MF C
02/009
002 - Session successfully started
    
```

If we just wanted RPG ILE Program objects...

COMMAND JEOPARDY!

```

Columns . . . : 1 71      Edit      CMDJPRDY/QCDSRC
SEU=>
***** Beginning of data *****
0001.00      CMD          PROMPT('Retrieve Object')
0002.00      /*-----*/
0003.00      PARM          KWD(OBJ) TYPE(OBJTYPE) SNGVAL((*CONTINUE)) +
0004.00          PROMPT('Object')
0005.00      OBJTYPE:    QUAL          TYPE(*GENERIC) LEN(10) SPCVAL((*ALL)) MIN(1) +
0006.00          EXPR(*YES)
0007.00      QUAL          TYPE(*NAME) LEN(10) DFT(*LIBL) +
0008.00          SPCVAL((*LIBL) (*USRLIBL) (*CURLIB) +
0009.00          (*ALLUSR) (*ALL)) EXPR(*YES) +
0010.00          PROMPT('Library')
0011.00      /*-----*/
0012.00      PARM          KWD(TYPE) TYPE(*CHAR) LEN(7) RSTD(*YES) +
0013.00          VALUES(*ALRTBL *AUTL *CFGL *CHTFMT *CLD +
0014.00          *CLS *CMD *COSD *CSPMAP *CSPTBL *CTLD +
0015.00          *DEVD *DOC *DOCL *DTAARA *DTADCT *DTAQ +
0016.00          *EDTD *FCT *FILE *FLR *FNTRSC *FORMFD +
0017.00          *GSS *IGCDCT *IGCSRT *IGCTBL *JOBQ *JOBQ +
0018.00          *JRN *JRNRCV *LIB *LIND *MENU *MODD +
0019.00          *MODULE *MSGF *MSGQ *OUTQ *OVL *PAGSEG +
0020.00          *PDG *PCM *PNLGRP *PRAVL *PRDDFN *PRDL0D +
0021.00          *QMORY *QMFORM *QRYDFN *RCT *SBSD *SCHIDX +
0022.00          *SPADCT *SRVPCM *SSND *S36 *TBL *USRIDX +
0023.00          *USRPRF *USRQ *USRSPC SPCVAL((*ALL)) +
0024.00          MIN(1) EXPR(*YES) PROMPT('Object type')
0025.00      /*-----*/
0026.00      PARM          KWD(ATTR) TYPE(*GENERIC) LEN(10) DFT(*ALL) +
0027.00          SPCVAL((*ALL)) PROMPT('Object attribute')
    
```

If we just wanted RPG ILE Program objects...

RTVOBJ

COMMAND JEOPARDY!

RTVOBJ

```

Columns . . . . : 1 71      Edit      CMDJPRDY/QCMDSRC
SEU=>
0031.00      PARM      KWD(RTNOBJ) TYPE(*CHAR) LEN(10) RTNVAL(*YES) +
0032.00      PROMPT('CL var for RTNOBJ (10)')
0033.00 /*-----*/
0034.00      PARM      KWD(RTNLIB) TYPE(*CHAR) LEN(10) RTNVAL(*YES) +
0035.00      PROMPT('CL var for RTNLIB (10)')
0036.00 /*-----*/
0037.00      PARM      KWD(RTNTYP) TYPE(*CHAR) LEN(7) RTNVAL(*YES) +
0038.00      PROMPT('CL var for RTNTYP (7)')
0039.00 /*-----*/
0040.00      PARM      KWD(USRSPC) TYPE(SPCTYPE) PMTCTL(*PMTRQS) +
0041.00      PROMPT('User space')
0042.00 SPCTYPE:  QUAL      TYPE(*NAME) LEN(10) DFT(RTVOBJ)  EXPR(*YES)
0043.00      QUAL      TYPE(*NAME) LEN(10) DFT(QTEMP)  +
0044.00      SPCVAL((*CURLIB)) EXPR(*YES) +
0045.00      PROMPT('Library')
0046.00 /*-----*/
0047.00      DEP      CTL(&OBJ *NE '*CONTINUE') PARM((TYPE)) +
0048.00      MSGID(OBJ0001)
0049.00      DEP      CTL(&OBJ *EQ '*CONTINUE') PARM((TYPE)) +
0050.00      NBRTRUE(*EQ 0) MSGID(OBJ0002)
0051.00 /*-----*/
MFA      D      03/001
IBM 3902 - Session successfully started
    
```

Return Params

Optional Hidden Params

Validate Input

RTVOBJ

COMMAND JEOPARDY!

RTVOBJ

```

Message ID . . . . . : OBJ0001
Message file . . . . . : CMDJPRDY
Library . . . . . : CMDJPRDY

Message . . . . . : You must specify TYPE unless you also specify
OBJECT(*CONTINUE).
Cause . . . . . : You attempted to execute the RTVOBJ command without
specifying an object type.
Recovery . . . . . : If you wish to continue retrieving the results of a
previous RTVOBJ command, specify OBJECT(*CONTINUE). Otherwise, specify an
object type. If you do not know the object type, you may specify special
value *ALL.

Message ID . . . . . : OBJ0002
Message file . . . . . : CMDJPRDY
Library . . . . . : CMDJPRDY

Message . . . . . : You may not specify TYPE with OBJECT(*CONTINUE).
Cause . . . . . : You attempted to execute the RTVOBJ command with
OBJECT(*CONTINUE) and an object type specified. RTVOBJ OBJECT(*CONTINUE)
continues retrieving the results of a previous RTVOBJ command in which an
object and type were specified.
Recovery . . . . . : Do not specify a value for TYPE, or specify an object and
library rather than *CONTINUE.
    
```

RTVOBJ

COMMAND JEOPARDY!

- **Creating our RTVOBJ Command:**
 - CRTCMD CMD(CMDJPRDY/RTVOBJ)
 PGM(CMDJPRDY/RTVOBJ)
 SRCFILE(CMDJPRDY/QCMDSRC)
 TEXT('Retrieve Object')
 ALLOW(*IPGM *BPGM)
 MSGF(CMDJPRDY/CMDJPRDY)
 HLPPNLGRP(CMDJPRDY/RTVOBJ)
 HLPID(RTVOBJ) PRDLIB(CMDJPRDY)

Ensures Command can only be used in an interactive or batch program

←

Links Command with Message Descriptions in Message File

RTVOBJ

COMMAND JEOPARDY!

```

Columns . . . : 1 71      Edit      CMDJPRDY/QCLSRC
SEU=>
***** Beginning of data *****
0001.00      PGM          PARM(&OBJLIB &TYPE7 &ATTR &RTNOBJ &RTNLIB +
0002.00                                     &RTNTYPE &USR20)
0003.00      DCL          VAR(&OBJLIB)          TYPE(*CHAR)  LEN(20)
0004.00      DCL          VAR(&TYPE7)           TYPE(*CHAR)  LEN(7)
0005.00      DCL          VAR(&ATTR)            TYPE(*CHAR)  LEN(10)
0006.00      DCL          VAR(&RTNOBJ)          TYPE(*CHAR)  LEN(10)
0007.00      DCL          VAR(&RTNLIB)          TYPE(*CHAR)  LEN(10)
0008.00      DCL          VAR(&RTNTYPE)          TYPE(*CHAR)  LEN(7)
0009.00      DCL          VAR(&USR20)           TYPE(*CHAR)  LEN(20)
0010.00      DCL          VAR(&TYPE)            TYPE(*CHAR)  LEN(10)
0011.00      DCL          VAR(&PASOBJ)          TYPE(*CHAR)  LEN(10)
0012.00      DCL          VAR(&PASLIB)          TYPE(*CHAR)  LEN(10)
0013.00      DCL          VAR(&PASTYPE)          TYPE(*CHAR)  LEN(10)
0014.00      DCL          VAR(&USRSPC)          TYPE(*CHAR)  LEN(10)
0015.00      DCL          VAR(&USRLIB)          TYPE(*CHAR)  LEN(10)
0016.00      DCL          VAR(&ERRMSGID)         TYPE(*CHAR)  LEN(7)
0017.00      DCL          VAR(&ERRQUIT)         TYPE(*LGL)
0018.00      DCL          VAR(&ERRKEY)          TYPE(*CHAR)  LEN(4)
0019.00      DCL          VAR(&ERRCODE)         TYPE(*CHAR)  LEN(16) +
0020.00                                     VALUE(X'000000000000010')
    
```

MF B 02/009
31 3902 - Session successfully started

COMMAND JEOPARDY!

RTVOBJ

```

Columns . . . : 1 71      Edit      CMDJPRDY/QCLSRC
SEU=>
0021.00      MONMSG      MSGID(CPF0000)  EXEC(GOTO CMDLBL(ERROR))
0022.00      MONMSG      MSGID(MCH3601)
0023.00      /*-----*/
0024.00      CHGVAR      VAR(&USRSPC)  VALUE(%SST(&USR20 1 10))
0025.00      CHGVAR      VAR(&USRLIB)  VALUE(%SST(&USR20 11 10))
0026.00      /*-----*/
0027.00      IF          COND(%SST(&OBJLIB 1 10) *EQ '*CONTINUE ') +
0028.00          THEN(DD)
0029.00          CHKOBJ    OBJ(&USRLIB/&USRSPC) OBJTYPE(*USRSPC)
0030.00          GOTO      CMDLBL(CONTINUE)
0031.00          ENDDO
0032.00      /*-----*/
0033.00      SNDPGMMSG  MSGID(CPF9898)  MSGF(QSYS/QCPFMSG) +
0034.00          MSGDTA('Searching for objects ' *CAT +
0035.00          %SST(&OBJLIB 11 10) *TCAT '/' *CAT +
0036.00          %SST(&OBJLIB 1 10) *BCAT 'type ' *CAT +
0037.00          &TYPE7) TOPGMQ(*EXT) MSGTYPE(*STATUS)
0038.00      /*-----*/
0039.00      CHKOBJ    OBJ(&USRLIB/&USRSPC) OBJTYPE(*USRSPC)
0040.00      MONMSG      MSGID(CPF9801)  EXEC(DD)
0041.00      CALL      PGM(QUSCRTUS)  PARM(&USR20 'TEMPORARY ' +
0042.00          X'00001400' ' ' *CHANGE ' 'Temporary +
0043.00          User Space for RTVOBJ Object List ')
0044.00      RMVMSG    CLEAR(*ALL)
0045.00      ENDDO
    
```

Global Error

Set up User Space

Display Status Message

Create User Space

COMMAND JEOPARDY!

RTVOBJ

```

Columns . . . : 1 71      Edit      CMDJPRDY/QCLSRC
SEU=>
0046.00      /*-----*/
0047.00      CHGVAR      VAR(&TYPE)  VALUE(&TYPE7 *CAT ' ')
0048.00      CALL      PGM(QUSLOBJ)  PARM(&USR20 'OBJL0100' &OBJLIB +
0049.00          &TYPE)
0050.00      /*-----*/
0051.00      CONTINUE:  CALL      PGM(CMDJPRDY/RTVOBJ)  PARM(&USR20 &ATTR +
0052.00          &PASOBJ &PASLIB &PASTYPE)
0053.00      CHGVAR      VAR(&RTNOBJ)  VALUE(&PASOBJ)
0054.00      CHGVAR      VAR(&RTNLIB)  VALUE(&PASLIB)
0055.00      CHGVAR      VAR(&RTNTYPE)  VALUE(%SST(&PASTYPE 1 7))
0056.00      IF          COND(%SST(&OBJLIB 1 10) *NE '*CONTINUE ') +
0057.00          THEN(SNDPGMMSG  MSGID(CPI9801) +
0058.00          MSGF(QSYS/QCPFMSG)  MSGDTA(' ') +
0059.00          TOPGMQ(*EXT)  MSGTYPE(*STATUS))
0060.00      RMVMSG    CLEAR(*ALL)
0061.00      RETURN
    
```

List Objects to User Space using API

Erase Status Message

We pass a different set of variables to the RPG. Why?
Hint: MCH3601

COMMAND JEOPARDY!

RTVOBJ

```

Columns . . . : 1 71      Edit      CMDJPRDY/QCLSRC
SEU=>
0063.00 ERROR:      IF      COND(&ERRQUIT) THEN(SNDPGMSG MSGID(CPF9999) +
0064.00              MSGF(QCPFMSG) MSGTYPE(*ESCAPE)) /* Func chk*/
0065.00              CHGVAR  VAR(&ERRQUIT) VALUE('1')
0066.00              CALL   PCM(QMHMOVPM) PARM(' '*DIAG ' ' +
0067.00              'X'0000001' '*      'X'0000001' +
0068.00              &ERRCODE)
0069.00              MONMSG  MSGID(CPF24A3) EXEC(RCVMSG MSGTYPE(*LAST))
0070.00              RCVMSG  MSGTYPE(*EXCP) RMV(*NO) KEYVAR(&ERRKEY) +
0071.00              MSGID(&ERRMSGID)
0072.00              IF      COND(%SST(&OBJLIB 1 10) *NE '*CONTINUE ') +
0073.00              THEN(SNDPGMSG MSGID(CPI9801) +
0074.00              MSGF(QSYS/QCPFMSG) MSGDTA(' ') +
0075.00              TOPGMQ(*EXT) MSGTYPE(*STATUS))
0076.00              IF      COND(&ERRMSGID *NE 'CPF001') THEN(CALL +
0077.00              PGM(QMHRSNEM) PARM(&ERRKEY &ERRCODE))
0078.00              RMVMSG  MSGKEY(&ERRKEY)
0079.00              SNDPGMSG MSGID(CPF001) MSGF(QSYS/QCPFMSG) +
0080.00              MSGDTA(RTVOBJ) MSGTYPE(*ESCAPE)
0081.00              ENDPGM
***** End of data *****

```

Erase
Status
Message

COMMAND JEOPARDY!

RTVOBJ

```

Columns . . . : 6 76      Edit      CMDJPRDY/QRPGLESRC
SEU=>
***** Beginning of data *****
0001.00 h dftname(rtobj)
0002.00 d rcvvar          ds
0003.00 d offset          1      4b 0
0004.00 d objcnt         9      12b 0
0005.00 d lendta        13     16b 0
0006.00
0007.00 d rcvar1          ds
0008.00 d objlib          20
0009.00 d objnam        10      overlay(objlib)
0010.00 d libnam         10      overlay(objlib:11)
0011.00 d objtyp        10
0012.00
0013.00 d attrds       ds      100
0014.00 d extatr       91     100
0015.00
0016.00 d attrbt       s       10
0017.00 d match        s       10
0018.00 d pos          s        2  0
0019.00 d star         s        1
0020.00

```

Declare
Variables
and Data
Structures

COMMAND JEOPARDY!

```

Columns . . . : 6 76      Edit      CMDJPRDY/QRPGLESRC
SEU=>
0021.00 d ThisPgm      pr          ExtPgm('RTVOBJ')
0022.00 d              20
0023.00 d              10
0024.00 d              10
0025.00 d              10
0026.00 d              10
0027.00
0028.00 d ThisPgm      pi
0029.00 d usrspc        20
0030.00 d attr          10
0031.00 d rtnobj        10
0032.00 d rtnlib        10
0033.00 d rntyp         10
0034.00
0035.00 d QUSRTVUS      pr          ExtPgm('QUSRTVUS')
0036.00 d PI_API_usrspc 20      Const
0037.00 d PI_API_hdrofs 8b 0    Const
0038.00 d PI_API_hdrlen 8b 0    Const
0039.00 d PI_API_rcvvar  8
0040.00
0041.00 d QUSCHGUS        pr          ExtPgm('QUSCHGUS')
0042.00 d PI_AP2_usrspc  20      Const
0043.00 d PI_AP2_hdrofs  8b 0    Const
0044.00 d PI_AP2_hdrlen    8b 0    Const
0045.00 d PI_AP2_rcvvar   8
0046.00 d PI_AP2_force    1      Const
0047.00
0048.00 d QUSROBJD        pr          ExtPgm('QUSROBJD')
0049.00 d PI_AP3_attrds    100
0050.00 d PI_AP3_atrlen  8b 0    Const
0051.00 d PI_AP3_format     8      Const
0052.00 d PI_AP3_objlib  20
0053.00 d PI_AP3_objtyp  10
    
```

Retrieve Data from User Space

Change Data in User Space

Retrieve Object Description

COMMAND JEOPARDY!

RTVOBJ

```

Session 2 - [24 x 80]
File Edit View Communication Actions Window Help
Printon Copy Paste Send Recv Display Color Map Record Stop Play Duit Clipboard Support Index
Columns . . . : 6 76      Edit      CMDJPRDY/QRPGLESRC
SEU=>
0056.00 match = *blanks;
0057.00 pos = %checkr(' :attr);
0058.00 star = %subst(attr:pos:1);
0059.00 if star = '*';
0060.00     pos = pos - 1;
0061.00     %subst(match:1;pos) = %subst(attr:1;pos);
0062.00 else;
0063.00     match = attr;
0064.00 endif;
0065.00
0066.00 QUSRTVUS (usrspc : 125 : 16 : rcvvar);
0067.00 offset = offset + 1;
0068.00
0069.00 Dou 1=2;
0070.00     if objcnt = 0;
0071.00         rtnobj = *blanks;
0072.00         rtnlib = *blanks;
0073.00         rntyp = *blanks;
0074.00         *inlr = *on;
0075.00         return;
0076.00     endif;
    
```

Set up for attribute matching, including generics

Retrieve header from user space

Set up loop. Exit with blanks when we're out of objects

02/009

31 3902 - Session successfully started


```

Columns . . . : 6 76      Edit      CMDJPRDY/QRPGLESRC
SEU=>
0078.00  QUSRTVUS (usrspc : offset : lendta : rcvar1);
0079.00  offset = offset + lendta;
0080.00  objcnt = objcnt - 1;
0081.00
0082.00  if attr = '*ALL  ';
0083.00  leave;
0084.00  endif;
0085.00
0086.00  *in77 = *off;
0087.00  monitor;
0088.00  QUSROBJD (attrds : 100 : 'OBJD0200' : objlib : objtyp);
0089.00  on-error;
0090.00  iter;
0091.00  endmon;
0092.00  %subst(attrbt:1:pos) = %subst(extatr:1:pos);
0093.00  if match = attrbt;
0094.00  leave;
0095.00  endif;
0096.00
0097.00  enddo;
0098.00
0099.00  offset = offset - 1;
0100.00  QUSCHGUS (usrspc : 125 : 16 : rcvvar : '0');
0101.00  rtnobj = objnam;
0102.00  rtnlib = libnam;
0103.00  rntntyp = objtyp;
0104.00  *inlr = *on;
0105.00  return;
    
```

Retrieve object from user space

All attributes selected

Match attributes

Update header of user space and exit

COMMAND JEOPARDY!

Thank you!

- Download the software from <http://bit.ly/1FnaiyG>
- Any questions?
- Please send any feedback to:

Steve Wolk
SWolk@PCRichard.com