

# WMCPA Working Knowledge Series

## Using Procedures in your programs

Tom Davidson

Cleindori Consulting, LLC

Certified ILE Specialist

[Tom.Davidson@CleindoriConsulting.com](mailto:Tom.Davidson@CleindoriConsulting.com)

# Using Procedures in your programs

- Ground Rules
  - Please set cell phones on 'Stun'
  - Feel free to ask questions
  - If you must take a call, please go outside the room

# Using Procedures in your programs

- Speaker Bio
  - 26 years on the box
  - Certified ILE Specialist
  - Developing in RPG all 25 years
  - Specialize in:
    - JDE
    - Global installations
    - Performance

# Using Procedures in your programs

- Cleindori Consulting, LLC
  - Specializing in Performance and JD Edwards
  - Contract Programming
  - CNC

# Using Procedures in your programs

## Agenda

- Why use procedures
- What is a procedure
- Anatomy of a procedure
- Using a procedure
- Re-useable (External) procedures
- Grouping procedures together
- Using procedure definitions to simplify API calls

# Using Procedures in your programs

- Why use procedures
  - Allows you to use more descriptive names:
    - CLR DREC vrs ClearDetailRecord
  - Isolates working data, reduces side-effects
  - Clearer code
  - Can be externalized, re-used easier
  - Separate definition from implementation

# Using Procedures in your programs

- What is a procedure
  - Like a subroutine, but less restrictive
    - Can call itself
    - Uses local variables
    - Parameters and return code are defined
    - I view them as 'programs' themselves

# Using Procedures in your programs

- What is a procedure (cont)
  - Restrictions
    - No F spec (gone in V6R1)
    - ILE Only



# Using Procedures in your programs

- Anatomy of a procedure
  - Procedure declaration (header)
  - Begin procedure declaration
  - Procedure interface
  - D-specs
  - Procedure body
  - End procedure declaration

# Using Procedures in your programs

- Anatomy of a procedure (cont)
  - Procedure declaration (header)
    - PR type
    - Using long names...
    - Define external program here
    - Return value (I use for status)
      - Not required
      - Can be any RPG data type
    - Parameter list
      - Optional
      - You should try to keep the number of parameters reasonable
      - OPTION(\*NOPASS), I don't use \*OMIT
    - include in a copybook

# Using Procedures in your programs

- Anatomy of a procedure (cont)
  - Procedure declaration (header)

```
0004.00  /if not defined($BDTRNGENPR)
0005.00  /define $BDTRNGENPR
0005.01  *
0006.00  * Convert a string to 6S0
0007.00  *
      FMT D  DName+++++++ETDsFrom+++To/L+++IDc.Keywords
0008.00  D Convert6S0          PR          1N
0009.00  D @TXT6              6A  CONST
0010.00  D @6S0                6S  0
0011.00  D @ISNULL            1N
0012.00  *
0013.00  * Get a PEID from BDPE using SOR and SORK
0014.00  *
0015.00  D GetPeopleIDFromBDSP...
0016.00  D                    PR          1N
0017.00  D @SOR                15P 0  CONST
0018.00  D @SORK                15P 0  CONST
0019.00  D @ID                  15P 0
0020.00  /endif
```

# Using Procedures in your programs

- Anatomy of a procedure (cont)
  - Begin procedure declaration
    - Defines name & return type
    - 'export' to make available outside the module
  - Procedure Interface (D spec)
    - Similar to PR spec, type is PI
    - Parameters MUST match PR & associated D specs

# Using Procedures in your programs

- Anatomy of a procedure
  - D-specs
    - 1<sup>st</sup> set are parameters
    - 2<sup>nd</sup> set are internal fields, starts with 1<sup>st</sup> non-blank 'Type'
      - I define the following here:
        - » Working fields for CONST Parameters here
        - » Return Code
      - Other work fields that are procedure specific

# Using Procedures in your programs

- Anatomy of a procedure(cont)

```
FMT PR PName+++++.....T.....Keywords
0157.00 P Convert6S0      B                export
0158.00 D                  PI                1N
0159.00 D @TXT6          6A  CONST
0160.00 D @6S0          6S 0
0161.00 D @ISNULL      1N
0162.00 *
0163.00 * Local fields
0164.00 *
0165.00 D ReturnCode    S                1N  INZ (*OFF)
0166.00 D                  DS
0167.00 D Txt6S0        1                6
0168.00 D Dec6S0        1                6S 0
```

# Using Procedures in your programs

- Anatomy of a procedure
  - Procedure body
    - 'C' specs, normal RPG
      - Can include subroutines, other procedures, even calls to the same procedure
      - /Free & /End-free OK
    - Exit with RETURN
      - RETURN not required if function does not return a value
  - End procedure declaration

# Using Procedures in your programs

```
0181.00 * Check valid
0182.00 C           if           %check(DIGITS:@TXT6) <> 0
0183.00 C           eval         ReturnCode = *OFF
0184.00 C           eval         @6S0 = 0
0185.00 C           return       ReturnCode
0186.00 C           endif
0187.00 * Is Valid
0188.00 C           eval         Txt6S0 = @TXT6
0189.00 C           eval         @6S0 = Dec6S0
0190.00 *
0191.00 * End function
0192.00 *
0193.00 C           return       ReturnCode
0194.00 P Convert6S0      E
```



# Using Procedures in your programs

- Using a procedure
  - CALLP (Call procedure)

```
0144.00 C          select
0145.00 C          when      RcdStatus = AddRecord
0146.00 c          eval      ENPROC = 'P'
0147.00 C          callp     BDEN_AddRec(BDENSRV_BDEN:ENID:SQLCode)
0148.00 C          when      RcdStatus = UpdRecord
0149.00 C          callp     BDEN_UpdateBYID(ENID:BDENSRV_BDEN:SQLCode)
0150.00 C          when      RcdStatus = RecordError
0151.00 C          callp     BDEN_WriteError('BDEN':BDENSRV_BDEN:SQLCode)
0152.00 C          endsl
```

# Using Procedures in your programs

- Use as a function (Like BIFF), can be in an IF or EVAL statement

```
0096.00 C          eval      ENNMBR = Fill6S0 (@TXT6:HasError:ISNULL)
0097.00 C          if        HasError
0098.00 C          eval      RcdStatus = RecordError
0099.00 C          eval      SQLCode = 'NMBR'
0100.00 C          endif

0109.00 C          if        NOT ConvertDateYYMD (@TXT8:ENRCDT:ISNULL)
0110.00 C          eval      RcdStatus = RecordError
0111.00 C          eval      SQLCode = 'RCDT'
0112.00 C          endif
```

- RPG Free example

```
0096.00          ENNMBR = Fill6S0 (@TXT6:HasError:ISNULL);
0097.00          if (HasError);
0098.00          RcdStatus = RecordError;
0099.00          SQLCode = 'NMBR';
0100.00          endif;
```

# Using Procedures in your programs

- Re-useable procedures
  - Modules (external procedures)
  - Must use EXPORT on interface
- Can't declare an activation group
- Must use NOMAIN
- Use CRTxxxMOD command (option 15)
- Can contain more than one procedure

# Using Procedures in your programs

- Modules

```
0001.00 H NOMAIN ALWNULL(*USRCTL)
- - - - ----- 32 data records excluded -----
0034.00 C/include QRPGLSRC,BDTRNGENPR
- - - - ----- 17 data records excluded -----
0052.00 * Get tournament System Of Record Information
0053.00 *
0054.00 P GetTournSORFromMoid...
0055.00 P          B          EXPORT
0056.00 D          PI          1N
0057.00 D @MOID          5P 0 CONST
0058.00 D @SOR          15P 0
0059.00 D @ASID          15P 0
- - - - ----- 27 data records excluded -----
0087.00 C          return      ReturnCode
0088.00 P GetTournSORFromMoid...
0089.00 P          E
- - - - ----- 6 data records excluded -----
0096.00 P ConvertDollars11P2...
0097.00 P          B          export
```

# Using Procedures in your programs

- Grouping procedures together
  - Binding directories
    - Definition: i5 object that defines a group of modules and/or service programs that will be searched to compile a program. This object contains NO actual code
  - Service programs
    - Definition – i5 object that actually contains a group of modules that should be related to each other. They contain actual code





# Using Procedures in your programs

- Binding directories

## – WRKBNDDIRE

```
*****
*                               Work with Binding Directory Entries                               *
*                                                                                               *
* Binding Directory:  BDNEW           Library:  BDNEW                                         *
*                                                                                               *
* Type options, press Enter.                                                                    *
*   1=Add   4=Remove                                                                    *
*                                                                                               *
*                                                                                               *
* Opt      Object      Type      Library      Date      Time      *
*  _      _          _          _          _          _          *
*  _      AWDIDC      *MODULE  *LIBL          05/03/02    10:25:35  *
*  _      DUPCHKAI    *MODULE  *LIBL          02/20/04    11:25:29  *
*  _      DUPCHKAN    *MODULE  *LIBL          11/23/04    17:34:00  *
*  _      DUPCHKAO    *MODULE  *LIBL          08/03/01    07:45:53  *
*  _      DUPCHKBA    *MODULE  *LIBL          08/18/06    11:57:40  *
*  _      DUPCHKBI    *MODULE  *LIBL          01/30/08    10:34:06  *
*  _      DUPCHKBM    *MODULE  *LIBL          01/20/04    10:08:31  *
*  _      DUPCHKCC    *MODULE  *LIBL          *          *          *
*                                                                                               *
*                                                                                               *
*                                                                                               *
*                                                                                               *
* Parameters or command                                                                    *
* ===> _____ *
* F3=Exit   F4=Prompt   F9=Retrieve   F5=Refresh   F12=Cancel   F17=Top   *
* F18=Bottom *
* (C) COPYRIGHT IBM CORP. 1980, 2005. *
*****
```



# Using Procedures in your programs

- Grouping procedures together
  - Binding directories
    - Using
      - CRTPGM

```
CRTPGM PGM(MYLIB/MYPGM)
      MODULE(MYLIB/MYMAINMOD)
      BNDDIR(MYLIB/MYBNDDIR)
```

- 'H' spec

```
0001.00 H alwnull(*usrctl)  bnmdir('BDNEW/BDNEW': '*LIBL/BDTOURNSRV')
0002.00 H ACTGRP('QILE')  DFTACTGRP(*NO)
- - - - ----- 35 data records excluded -----
0038.00 *****
0039.00 * Prototypes
0040.00 /include BDENSRVPR
0045.00 * Parameters
0047.00 D @BDBATCHds      E DS              EXTNAME(BDBATCH)  PREFIX('@')
0048.00 * Numeric info
0049.00 D  @SIGN              1A  OVERLAY(@BATCHPCDT)
0050.00 D  @11P2             11A  OVERLAY(@BATCHPCDT:*NEXT)
```

# Using Procedures in your programs

- Grouping procedures together
  - Service programs
    - Binder source
      - Defines the order that procedures (functions) are searched
      - Defines the signature, you can use \*PRV for older lists.
      - I use the YYYYMMDDHHMMSSxx as my signature.
      - Only 3 commands:
        - » STRPGMEXP
        - » EXPORT
        - » ENDPGMEXP

# Using Procedures in your programs

- Grouping procedures together
  - Service programs
    - Binder source

```
0011.00          STRPGMEXP  PGMLVL (*CURRENT) SIGNATURE ('20080221110845MS')
0014.00          EXPORT    SYMBOL (GetTournSORfromMoid)
0015.00          EXPORT    SYMBOL (ConvertDollars11P2)
0016.00          EXPORT    SYMBOL (Convert6S0)
0017.00          EXPORT    SYMBOL (Convert6S2)
0018.00          EXPORT    SYMBOL (Fill6S0)
0019.00          EXPORT    SYMBOL (Fill6S2)
0141.00          ENDPGMEXP
0142.00
0143.00          STRPGMEXP  PGMLVL (*PRV) SIGNATURE ('20080201132408MS')
0144.00          EXPORT    SYMBOL (GetTournSORfromMoid)
0145.00          EXPORT    SYMBOL (ConvertDollars11P2)
0146.00          EXPORT    SYMBOL (Convert6S0)
0147.00          EXPORT    SYMBOL (Convert6S2)
0150.00          ENDPGMEXP
```

# Using Procedures in your programs

- Grouping procedures together
  - Service programs

```
CRTPGM PGM(MYLIB/MYPGM)
```

```
MODULE(MYLIB/MYMAINMOD)
```

```
BNDSRVPGM(MYLIB/MYSRVPGM)
```

- Or both

```
CRTPGM PGM(MYLIB/MYPGM)
```

```
MODULE(MYLIB/MYMAINMOD)
```

```
BNDDIR(MYLIB/MYBNDDIR)
```

```
BNDSRVPGM(MYLIB/MYSRVPGM)
```

# Using Procedures in your programs

- Example of program with signature

```
*****
*                                     Display Program Information                               *
*                                     Display 4 of 7 *
* Program . . . . . : FILBDDV      Library . . . . . : TDAVID *
* Owner . . . . . : QPGMR *
* Program attribute . . : RPGLE *
* Detail . . . . . : *SRVPGM *
* *
* *
* Type options, press Enter. *
* 5=Display *
* *
* Service *
* Opt Program Library Signature *
* _ BDTOURNSRV *LIBL F2F0F0F8F0F2F2F1F1F1F0F8F4F5D4E2 *
* _ QRNXIE QSYS D8D9D5E7C9C5404040404040404040 *
* _ QLEAWI QSYS 44F70FABA08585397BDF0CF195F82EC1 *
* *
* *
* Bottom *
* F3=Exit F4=Prompt F11=Display character signature F12=Cancel F17=Top *
* F18=Bottom *
* *
*****
```

# Using Procedures in your programs

- Using the procedure definition to call API's
  - Easy way to easily extend RPG
  - A little tricky deciding what variable types to use
  - 'Normal' API restrictions (EXTPGM('ABC123'))
    - Can't have a return code
    - All variables must be passed by reference

# Using Procedures in your programs

The screenshot shows the IBM Systems Information Center interface in a Windows Internet Explorer browser. The page title is "Retrieve System Status (QWCRSSTS) API". It lists required and optional parameters for the API. Below the parameter list, there is a description: "The Retrieve System Status (QWCRSSTS) API allows you to retrieve a group of statistics that represents the current status of the system." and a section for "Authorities and Locks".

Required Parameter Group:			
1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Format name	Input	Char(8)
4	Reset status statistics	Input	Char(10)
5	Error Code	I/O	Char(*)

  

Optional Parameter Group:			
6	Pool selection information	Input	Char(*)
7	Size of pool selection information	Input	Binary(4)

Default Public Authority: \*USE  
Threadsafe: No

```
FMT D DName+++++++ETDsFrom+++To/L+++IDc.Keywords+++++++
0008.00 DQWCRSSTS PR EXTPGM('QWCRSSTS')
0009.00 D Rcvr 1A
0010.00 D RcvrLen 10I 0
0011.00 D Format 8A
0012.00 D Reset 10A
0013.00 D ErrorCodeDS 1A
0014.00 D PoolSelectDS 1A OPTIONS(*NOPASS)
0015.00 D PoolSelDSSize 10I 0 OPTIONS(*NOPASS)
```

Define Char(\*) as alpha 1

Define Binary(4) as 10I0

Define Binary(2) as 5I0

# Using Procedures in your programs

- Suggested use of directives

```
/if not defined(PROCNAME_{$})
```

```
/define PROCNAME_{$}
```

```
...Prototype here
```

```
/else
```

```
/if defined(($_CSPEC$)
```

```
/if not defined(PROCNAME_BODY$)
```

```
/define PROCNAME_BODY$
```

```
...Procedure here
```

```
/endif
```

```
/endif
```

```
/endif
```



# Using Procedures in your programs

- Questions?