

# CL Programming Enhancements

Andy Youens  
IBM i Consultant/Instructor  
FormaServe Systems

## Andy Youens

- IBM i Consultant/Instructor at FormaServe, who has over 32 years IT experience working with IBM midrange servers, S/34, S/36, AS400, iSeries & IBM i
- Specialties are IBM i, RPG, PHP, CL, Zend Server, Zend Frameworks, SQL & DB2
- Accomplished instructor & an article writer specialising in programming on the IBM i
- Prior to IT, Andy is proud to have served 10 years in the Royal Navy as a communications expert

# CL Programming Enhancements

- When launched in 1988 through to V5R3 in 2004, IBM had made no enhancements to CL programming
- We thought we were burdened with the same CL forever!
- We were stuck with restrictions like, only being able to open one file in a program, cannot use printer or ICF files, use sub-files within display files, nor use program described display files
- Then IBM got very busy on the CL front & we now have many great programming features

# Version 5.3 Enhancements

- New variable types
- Using more than one file
- Structured programming

# Version 5.3 Enhancements

## New Variable Types

```
Dcl      &Msg      *Char    10      'Testing.....'  
Dcl      &Salary  *Dec     (9 2 )  
Dcl      &Ind     *Lgl     '1'  
Dcl      &Count  *Int  
Dcl      &Loopt  *UInt
```

No LEN parameter

# Version 5.3 Enhancements

## File Handling

```
/* 5 Files can be declared */
DclF      UtRst01p  OpnId( Restore )
DclF      UtJrn01p  OpnId( Journal )

/* Use the Open ID to retrieve the data */
RcvF      OpnID( Restore )
RcvF      OpnID( Journal )
```

External  
Name


Internal  
Name

Internal  
Name

# Version 5.3 Enhancements

## Structured Programming - DoFor



```
/* Do 10 times                                     */  
DoFor          Var(&Count) From(1) To(10)  
  ChgVar      &Msg %Sst(&Rcv &Count 2)  
  SndPgmMsg  &Msg ToPgmQ(*Ext)  
EndDo
```



# Version 5.3 Enhancements

## Structured Programming - DoWhile

```
/* Do While Count Is Less Than 100 */  
DoWhile    Cond( &Count *Lt 100 )  
    ChgVar    &Count    (&Count + 1 )  
    SndPgmMsg &Msg      ToPgmQ( *Ext )  
EndDo
```



Checks the  
value of  
&Count here!




# Version 5.3 Enhancements

## Structured Programming - DoUntil

```
/* Do Until 100 Times */
```

```
DoUntil      Cond( &Count *Gt 100 )  
  ChgVar      &Count  (&Count + 1 )  
  ChgVar      &Msg    %Bin(&Rcv &Count 2)  
  SndPgmMsg  &Msg    ToPgmQ( *Ext )  
EndDo
```



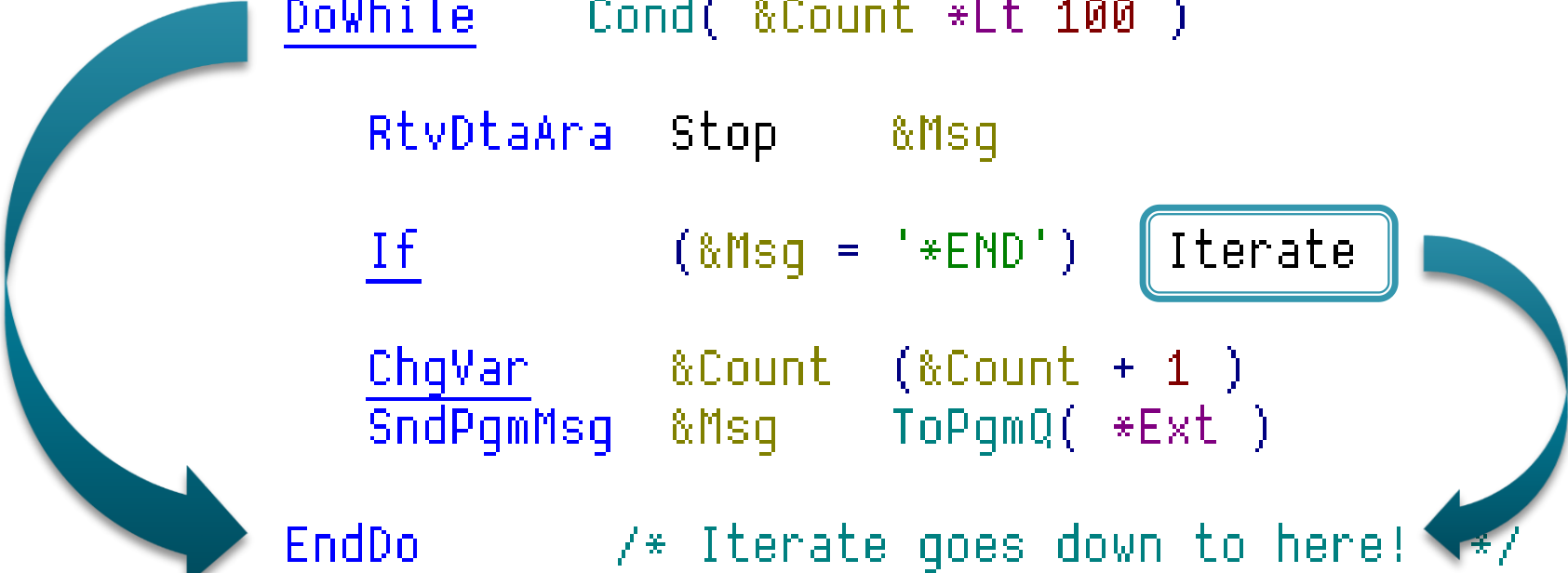
Checks the  
value of  
&Count here!

Always performs loop!

# Version 5.3 Enhancements

## Structured Programming - Iterate

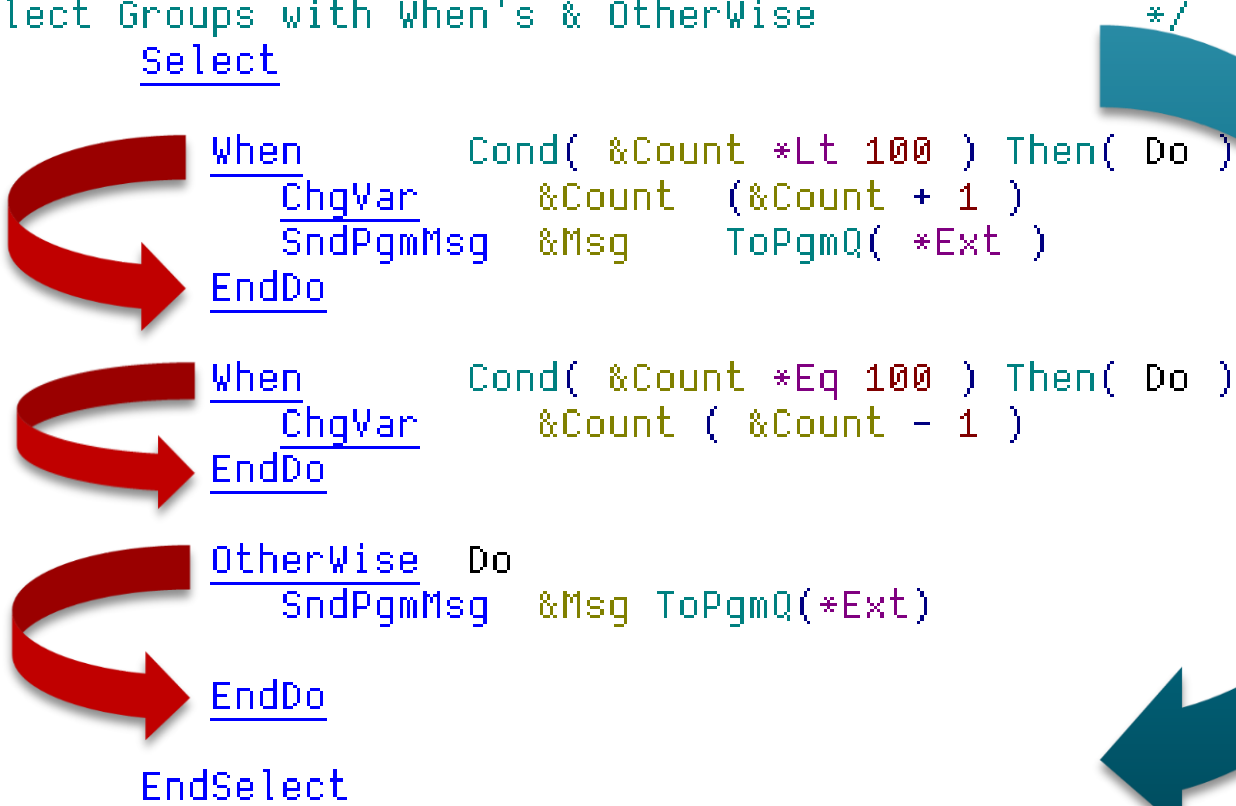
```
/* Iterate Example */  
DoWhile      Cond( &Count *Lt 100 )  
  
    RtvDtaAra  Stop      &Msg  
  
    If        (&Msg = '*END' ) Iterate  
  
    ChgVar    &Count  (&Count + 1 )  
    SndPgmMsg &Msg    ToPgmQ( *Ext )  
  
EndDo      /* Iterate goes down to here! */
```



# Version 5.3 Enhancements

## Structured Programming – Select When & Otherwise


```
/* Select Groups with When's & Otherwise
   Select
   When      Cond( &Count *Lt 100 ) Then( Do )
     ChgVar    &Count ( &Count + 1 )
     SndPgmMsg &Msg   ToPgmQ( *Ext )
   EndDo
   When      Cond( &Count *Eq 100 ) Then( Do )
     ChgVar    &Count ( &Count - 1 )
   EndDo
   OtherWise Do
     SndPgmMsg &Msg ToPgmQ(*Ext)
   EndDo
   EndSelect
*/
```



# Version 5.3 Enhancements

## Structured Programming – Leave

```
/* Leave Do Group Example */  
  
    DoWhile    Cond( &Count *Lt 100 )  
        If      (&Msg = '*END') Then( Leave ) /* Get out of the Do Loop */  
    EndDo  
    SndPgmMsg 'All Done Time To Go Home!' ToPgmQ(*Ext)
```



# Version 5.3 Enhancements

## Structured Programming – Example

```
/* Loop Thro Employee File
   (c) - FormaServe Systems Ltd 1990 - 2015

Start:   Pgm

         Dcl      &Msg      *Char      250
         DclF      Employee

         MonMsg   Cpf0000 Exec(Goto Error)

         DoUntil  '0' /* Do Forever! */

         Rcvf

         MonMsg   MsgID(Cpf0864) Exec(Leave)

         ChgVar   &Msg ( &EmFir || &EmSur )
         SndPgmMsg &Msg

         EndDo

End:     Return
```



No GOTO's!

# Version 5.4 Enhancements

- Subroutines
- Pointer Variables
- Based Variables
- Defined Variables
- %Address & %Offset Built-in Functions

# Version 5.4 Enhancements

## Subroutines



# Version 5.4 Enhancements

## Pointers & Based

```
Dcl      &Auto      *Char      26      'abcdefghijklmnopqrstu
```

```
Dcl      &Ptr       *Ptr        Address(&Auto)
```

```
Dcl      &Based    *Char      10      Stg(*Based) BasPtr(&Ptr)
```

```
ChgVar   %Ofs(&Ptr) (%Ofs(&Ptr) + 10)
```

```
SndPgmMsg &Based
```

&Based =  
'klmnopqrst'

&Based =  
'abcdefghij'

```
Chgvar   &Auto 'FormaServe Systems Ltd'
```

```
SndPgmMsg &Based
```

&Based =  
'Systems L'



# Version 5.4 Enhancements

## Defined Variables

```
Dcl      &Full      *Char      20      'QGPL      FSS      '  
Dcl      &Library  *Char      10  
Dcl      &Object  *Char      10
```

```
DefVar( &Full )  
DefVar( &Full 11 )
```

**&Library = 'QGPL'  
&Object = 'FSS'**


# Version 6.1 Enhancements

- Close Command
- Include Command
- Declare Processing Options
- Dynamic prompt messages

# Version 6.1 Enhancements

## Close Command

```
/* 5 Files can be declared */  
  
    DclF      UtRst01p  OpnId( Restore )  
    DclF      UtJrn01p  OpnId( Journal )  
  
    RcvF      Restore  
    RcvF      Journal  
  
/* Explicitly Close an Open File                               */  
    Close     OpnID(Journal)
```



# Version 6.1 Enhancements

## Include Source

```
/* Include Coding From Another Source Member */
```

```
Include SrcMbr(Error01) SrcFile(Training/QCLLeSrc)
```

```
L3400-  
L3500- /* Include Coding From Another Source Member */  
L3600-  
L3700- /* START INCLUDE SRCMBR(ERROR01) SRCFILE(TRAINING/QCLLESRC) */  
L3800- /* +  
L3900 Error01 - Standard Error Handling Variables +  
L4000 +  
L4100 Andy Youens - FormaServe Systems +  
L4200 September 2011 +  
L4300 */  
L4400-  
L4500- Dcl &MsgId *Char 7  
L4600- Dcl &MsgDta *Char 256  
L4700- Dcl &MsgF *Char 10  
L4800- Dcl &MsgfLib *Char 10  
L4900- Dcl &ErrorSw *Lgl  
L5000-  
L5100- Copyright '@ 1990 - 2015 - FormaServe Systems Ltd.'  
~300- MonMsg Cpf0000 Exec(Goto Error)  
~0-  
/* END INCLUDE SRCMBR(ERROR01) SRCFILE(TRAINING/QCLLESRC) */
```

# Version 6.1 Enhancements

## Define Compiler Options

```
Dcl      &Company *Char 50 'FormaServe'
```

```
DclF    UtRst@1p OpnId( Restore )
```

```
Copyright '© - FormaServe Systems Ltd 1990'
```

```
DclPrcOpt Log( *Yes ) UsrPrf( *Owner ) Aut( *Exclude ) ActGrp( Fss )
```

Precedes all other  
commands

In any order

# Version 6.1 Enhancements

## Dynamic prompt messages

```
/* Define Command Compiler Options */  
  
Cmd          Prompt(Usr0013) PmtFile(UtMsgF *Dynamic) +  
             HlpID(*Cmd) HlpPnlGrp(Ut521H)
```

\*LIBL is used for  
value at run-time

Very useful for multi  
language support

## Version 7.1 Enhancements

- Nested INCLUDE support
- Encrypted debug listing view support for ILE CL
- Compiler option to show DO loop & SELECT group nesting levels in compiler listing
- Longer integer variables
- Retrieve CL Source command support for ILE CL

# Version 7.1 Enhancements

## Nested Include Support

```
/* Include Coding From Another Source Member          */
   Include      SrcMbr(Error01) SrcFile(Training/QC1LeSrc)

/*                                                    +
   utErr01c1 - Standard Error Handling                +
                                                    +
   Andy Youens - FormaServe Systems                  +
   September 2011                                    +
                                                    */

   Dcl      &MsgId      *Char      7
   Dcl      &MsgDta     *Char      256
   Dcl      &MsgF       *Char      10
   Dcl      &MsgfLib    *Char      10
   Dcl      &ErrorSw    *Lgl

   Copyright '© 1990 - 2015 - FormaServe Systems Ltd.'

   MonMsg   Cpf0000 Exec(Goto Error)

   Include   SrcMbr(Error02) SrcFile(Training/QC1LeSrc)
```



# Version 7.1 Enhancements

## Encrypted debug listing view support for ILE CL

```
                Create Bound CL Program (CRTBNDCL)

Type choices, press Enter.

Allow RTVCLSRC . . . . . *YES          *YES, *NO
Replace program . . . . . *YES          *YES, *NO
Target release . . . . . *CURRENT      *CURRENT, *PRV, V5R4M0, ...
Authority . . . . . *LIBCRTAUT        Name, *LIBCRTAUT, *CHANGE...
Sort sequence . . . . . *HEX          Name, *HEX, *JOB, *JOB RUN...
  Library . . . . . _____        Name, *LIBL, *CURLIB
Language ID . . . . . *JOB RUN
Optimization . . . . . *NONE          *NONE *BASIC *FULL 10
Debugging view . . . . . *STMT        *STMT, *SOURCE, *LIST...
Debug encryption key . . . . . *NONE
Enable performance collection . . . . . *PEP      *PEP, *FULL, *NONE
INCLUDE file . . . . . *SRCFILE      Name, *SRCFILE
  Library . . . . . _____        Name, *LIBL, *CURLIB

Bottom

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

# Version 7.1 Enhancements

Show DO/SELECT group nesting levels in compiler listing

Create Bound CL Program (CRTBNDCL)

```
01          DoFor          Var (&Count) From (1) To (10)
           ChgVar          &Msg  %Bin (&Rcv &Count 2)
           SndPgmMsg       &Msg  ToPgmQ (*Ext)

02          DoUntil       Cond ( &Count *Gt 100 )
           ChgVar          &Count (&Count + 1 )
           ChgVar          &Msg  %Bin (&Rcv &Count 2)
           SndPgmMsg       &Msg  ToPgmQ ( *Ext )

02          EndDo

01          EndDo
```

# Version 7.1 Enhancements

Longer integer variables

Dcl

&LongPos

\*Int

8



Default value is 4

# Version 7.1 Enhancements

## Retrieve CL Source command support for ILE CL

```
Retrieve CL Source (RTVCLSRC)

Type choices, press Enter.

Program . . . . . █ _____ Name
Library . . . . . *LIBL _____ Name, *LIBL, *CURLIB
Service program . . . . . _____ Name
Library . . . . . *LIBL _____ Name, *LIBL, *CURLIB
Module . . . . . *PGM _____ Name, *PGM
Library . . . . . _____ Name, *LIBL, *CURLIB
Source file . . . . . QCLSRC _____ Name
Library . . . . . *LIBL _____ Name, *LIBL, *CURLIB
Source member . . . . . *PGM _____ Name, *PGM
Retrieve included source . . . . *NO _____ *NO, *YES

Bottom

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

## Version 7.2 Enhancements

- Six new built-in functions (BIFs) for working with character strings
- Six new built-in functions for conversion operation
- Two new built-in functions for size operation

## Version 7.2 – Check BIFs

- %Check – let's you know the first position of a string that does not exist in the compare string
- %CheckR – will inform you of the last position of the string that does not appear

# Version 7.2 Enhancements

## %Check

```
/* Check if £ in A_Number                                     +  
   A_Number = '£123.44'                                     */  
ChgVar      &Len Value(%Check('£' &A_Number))
```

Check the first position  
that does not contain the  
string

&Len  
= 2

Returns zero if  
not found

# Version 7.2 Enhancements

## %CheckR – Check Reverse

```
/* Where is the first non CR character from right?      +  
   Text = ' £10,000 CR'  Sign = 'CR'                  */  
ChgVar      &Len Value(%CheckR(&Sign &Text))
```

Check from the right, the first position that does not contain the string

&Len  
= 9

Returns zero if not found



## Version 7.2 – Trim BIFs

- %Trim – Trims a string from both the left & the right
- %TrimL – Trims a string from the left
- %TrimR – Trims a string from the right

# Version 7.2 Enhancements

## %Trim, %TrimL & %TrimR

```
Dcl          &Amount  *Char  256  '   *** 10.55 ***  '  
ChgVar      Var(&Result) Value(%Trim(&Amount ' *'))
```

&Result = '10.55'  
Leading blanks & asterisks removed

# Version 7.2 Enhancements

## %Scan

```
/* &Text is 'CL now supports %CHECK, %CHECKR and %SCAN' */  
ChgVar    Var(&Pos) Value(%Scan('%' &Text))
```

Search for a string  
within another string

&Pos = 17

Returns zero if  
not found

# Version 7.2 Enhancements

## %Char

```
/* &Count is 50 */  
ChgVar    &String    Value( %Char( &Count ) )
```

Convert to a  
character string

&String = '50'

# Version 7.2 Enhancements

%Dec

```
Dcl          &Total          *Char      10      '123.45'  
Dcl          &Decimal        *Dec       (10  5)  
  
ChgVar      &Decimal      Value(%Dec(&Total 5 2))
```

Convert to a  
numeric

&Decimal =  
00123.45000

# Version 7.2 Enhancements

%Int

```
Dcl          &Total      *Char    10      '123.45'  
Dcl          &Result    *Int  
  
ChgVar      &Result    Value( %Int(&Total) )
```

Convert to an  
integer

&Result = 123

# Version 7.2 Enhancements

%UInt

```
Dcl      &Total      *Char      10      '-123.45'  
Dcl      &Result      *Int  
  
ChgVar   &Result      Value( %UInt(&Total) )
```

Convert to an  
unsigned integer

&Result = 123

# Version 7.2 Enhancements

%Lower

```
/* &CoUpper is 'FORMASERVE'                                     */  
ChqVar    &String    Value( %Lower( &CoUpper ) )
```

Convert to  
lowercase

&String =  
'formaserve'



# Version 7.2 Enhancements

%Upper

```
/* &CoLower is 'formaserve' */  
ChgVar    &String  Value( %Upper( &CoLower ) )
```

Convert to  
uppercase

&String =  
'FORMASERVE'

# Version 7.2 Enhancements

%Size

```
/* &CoLower is defined as *Char 50                                     */  
ChgVar      &Size Value( %Size( &CoLower ) )
```

Return no of  
bytes

&Size = 50

# Version 7.2 Enhancements

%Len

```
Dcl      &Text      *Char      50      'FormaServe  '  
Dcl      &Len      *Int  
ChgVar  Var(&Len)  Value(%Len(%Trim(&Text)))
```



&Len = 10

# PowerWire Articles

- August 2015 - RPG Debugging Alternative
- June 2015 - Using wizards with RPG & RD i
- April 2015 - How Windows 10 Works With Power
- March 2015 - Why you should embrace CL's new features
- January 2015 - Free form RPG, will it take off?



# Anything Else – Thank you!



01908 609500



Andy@formaserve.co.uk



[www.formaserve.co.uk](http://www.formaserve.co.uk)



@FormaServe @AndyYouens