

BCI433 - IBM i Business Computing

**Week 3: RPGLE Programming with
Display files**

Agenda

- Display Files with DDS code
- Using RDi tool – Screen Designer
- RPGLE Programming
 - free Format syntax
 - using Display File
- QuickCheck (Questions)

Lesson Objectives

Upon completion of this lecture and lab 3 you'll be able to:

- Create a display file using Screen Designer with DDS code
- Code an interactive RPGLE program that uses the display file

DDS Concepts

- Data Description Specifications (DDS) – a traditional mean to describe data attributes (such as the names and lengths of records and fields) on the IBM® i operating system.
- The file types that use DDS describe:
 - **Physical** and **logical files** (database tables or views)
 - **Display files**
 - **Printer files**
 - ...

Display Files

- Display (or WORKSTN) files are
 - objects of type *FILE with attribute of DSPF on the IBM i.
 - ▶ e.g. INCOMETDSP.*file.dspf
 - used to communicate interactively with users at display terminals.
- In a record of display file, DDS describes:
 - named fields (as variables in RPG program)
 - constants (text, date, time, user)
- DDS is also used to specify:
 - positions (of fields), display attributes (e.g. highlight, reverse image), data validation, screen control (e.g. overlay), indicators (01~99), edit code (EDTCDE), edit word (EDTWRD) and subfiles.
 - 3 types of fields: input, output, and both (input/output)

Control screen record

➤ Example DDS code:

"000024 A

R YOURTAX

"000025 A

CF03(03 'EXIT')"
OVERLAY "

- Both CF03(03 'EXIT') and OVERLAY are record-level keywords or functions to specify the format/control for record YOURTAX
 - ▶ CF03(03 'EXIT'): indicator 03 was bound to function key F3
 - ▶ OVERLAY: the screen record YOURTAX will overlay the previous screen record (both screen content will show up)

Edit Code / Word

➤ Edit Code: EDTCDE

- Use predefined format to edit numeric fields in display (or printer) files
- e.g. "000111A TAXRATE 2Y 20 16 40**EDTCDE(1)**"
 - ▶ use predefined data format "1" to render the data of name field TAXRATE
- See Lab 1 demo:
CALL BCI433LIB/EDITCODES

➤ Edit Word: EDTWRD

- to make numbers more readable – custom data format
- Example

Your number is 01234

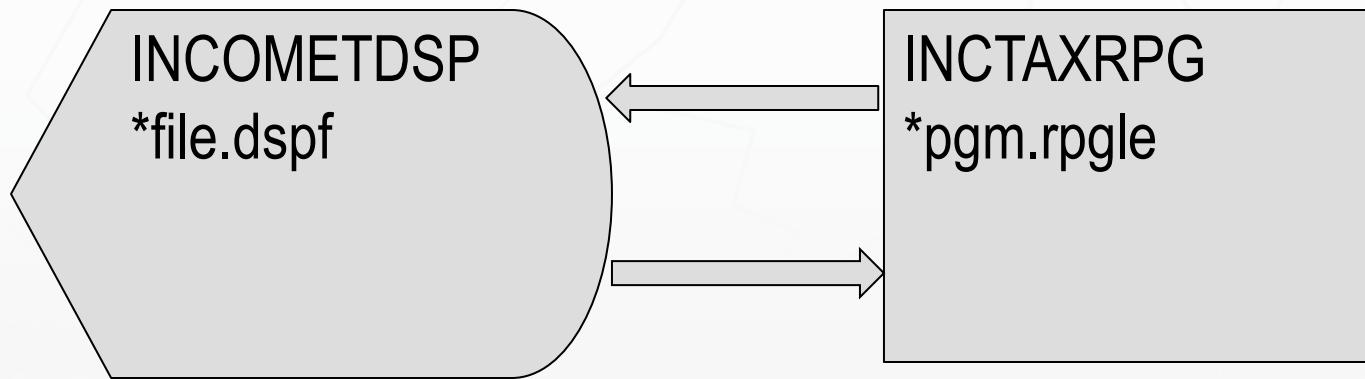
The default display is 1,234

You want it displayed as 012-34

Then it should be: EDTWRD ('0„„-„„')

(„ means 1 blank space)

Display Files and Programs



Create an RPGL E Program that uses the display file

- **RPG: Report Program Generator**
- To the RPGL E program, a display file is an externally-described file. So usually the first line of an RPG program is to specify the display file.
 - In Lab 2, we used fixed-form DDS code to define display file. That way are obsolete.

FSTUDACCTS IP E DISK

- In Lab 3 and other labs, we must use free-format DDS code, including the code lines to specify display files, e.g.
DCL-F INCOMETDSP workstn;

Operations on Screen Records

➤ EXFMT

- Write a screen record and waits for input (Write / Read operations)

e.g. LETGRADE = 'F' ;
EXFMT RECORD1;

➤ WRITE

- Writes a screen record to a file/display station (without a pause)

e.g. WRITE RECORD1;
EXFMT RECORD2;

Variables in RPG

- Named fields in a display file (such as TEST1, Test2,... in MARKSDSP.DSPF) can be used as variables in RPG language.
e.g. TESTOVERALL = (TEST1 + TEST2 + TEST3)/3;
- In addition, you can declare stand-alone variables in RPG, e.g.

```
DCL-S Premium          PACKED(4:2) ;
```

Indicators

- 'Boolean' variables – predefined
 - *IN00 - *IN24 are usually used to map 24 function keys in display file.
 - ***INLR** – Last Record Indicator
 - ▶ How we end RPG programs
- Have the values:
 - '0' or *OFF
 - '1' or *ON

RPG Syntax

- Each program statement ends with a “;” – semicolon.
- Comment
 - e.g. **//** This is a comment
- IF Statements
 - OPERATOR: =, <, <=, <>, NOT, AND, OR
 - e.g.

```
IF Test1 < Test2;  
    Minimum = Test1;  
ENDIF;
```
 - e.g.

```
IF Test1 < Test2;  
    Minimum = Test1;  
ELSE;  
    Minimum = Test2;  
ENDIF;
```

RPG Syntax

➤ Loops – DO While

- e.g.

```
DOW (a <= 5) AND (b + c = 0);  
    EXSR nextRecord;  
ENDDO;
```

➤ Loops – DO UNTIL

- e.g.

```
DOU X > 10;  
    Total = Total + Array(x);  
    X = X + 1;  
ENDDO;
```

RPG Syntax - Subroutines

- Blocks of logic
- Execute a subroutine
 - e.g. **EXSR** GETGRADE
- Define a subroutine
 - e.g.

```
BEGSR GETGRADE;  
TOTALMARK = 90;  
LETGRADE = 'A+';  
ENDSR;
```

RPG Syntax

- At the end of program

```
*INLR = *ON;  
RETURN;
```

- ***INLR = *ON;**
 - The standard way of ending an RPG program.
- **RETURN;**
 - To return control to the operation system.

Lab 3 Demo

Homework?

- Review lecture notes.
- Complete Lab 3
- Lab 2 due



The End