BCI433 - IBM i Business Computing

Week 7: RPGLE Business Application

Agenda

- Passing Parameters to RPGLE program
- Using a CLLE driver Program
  - To run and manage RPGLE applications
  - File overrides for both PF and DSPF
- Dynamic and Static Calls
  - Calling a Program or a Procedure
- CL command CHKOBJ
- Lab 6

Lesson Objectives

Upon completion of this lecture and lab 2 you'll be able to:

- Pass a parameter to an RPGLE program
- Provide file overrides to produce different report results
- Create a CLLE module that passes a parameter to an RPGLE Module and combine them into a running program.
- Check object's access/existence in program using CHKOBJ
- Practice on using CL Driver program to run and manage business applications

Passing Parameters to an RPGLE Program

- When run an RPGLE program, we can pass a parameter to the program from command line, like Java or other programs.
- To implement this, we must define the data within the RPGLE program that receives the data:

```
Dcl-PI Main ExtPgm('PAYROLLPG2'); // must match the RPGLE program name
                                  // i.e. PAYROLLPG2
    ShiftType char(30);  // parameter name and type
End-PI;
```

- Run program from command line

```
Call PayRollPG2   Parm('Day Shift')
```

Note: don't forget the keywork Parm -> different form Java calls

- Practice this using Lab 5 program, and don't change Lab 5 code. How?

Using a CLLE driver Program

- One strategy regarding CLLE and RPGLE programming:
  - use CLLE program (as driver program) to run and manage (RPGLE program as) business application.
  - The driver program can be batch or interactive jobs

- In Lab 6, the CLLE driver program RUNPAYGM2 or 3:
  - as a interactive program
    - Provides a simple menu using 'SNDUSRMSG' (rather than a display file)
  - call an pass parameter to the RPGLE program
  - manage 2 kinds of file overrides
  - show reports of different results

File Overrides

- There are 2 kinds of file overrides used in the CLLE driver program:
  - Database file (PF) override with logical file
    - e.g. OVRDBF SHIFTWEEK NIGHTS
  - Spooled/printer file overrides
    - e.g. OVRPRTF FILE(PAYRPT2) SPLFNAME(NIGHTSHIFT)
    - This way, the reports with different names will be generated.

Creating Logical Files

- As a database object, similar to physical files, logical files can be generated in 2 ways:
  - Using SQL, i.e. CREATE View viewname AS Select…
  - Using DDS code
- In a source physical file, e.g. QDDSSRC, create a member of LF type for a logical file. Don’t forget to compile it!
- After compilation, a logical file will be created with the type of *file and the attribute of 'lf'

Logical File with DDS

- LF example: create a logical file that sorts Item file by Name and Stocking Size (for PF file):

```
R ITEMSR                PFILE(ITEMS)
K ITEMNAME
K STOCKSIZE
S STOCKQNTY             COMP(GT 0)
```

- Level of entries
  - R: record format
  - K: key field, or index field
  - S/O: select / omit
    - may have more select/omit entries
    - functions: e.g.
      - **COMP(GT 0), COMP(EQ 'JSMITH'), COMP(NE 'NY')**
      - **RANGE(10 59), VALUES (301542 306902 382101 486592 502356), ...**

Logical Files in Lab 6

- You need to create all three logical files:
  - DAYS, NIGHTS, AFTERNOON
- You may create a logical file ALLSHIFTS for all shifts
  - By removing the last (s or select) line from DAYS, or …
  - For easy implementation of your driver program
- Note: when you update a logical file, you must manually delete the existing logical file (object). Otherwise, new object will be produced.

Dynamic and Static Calls

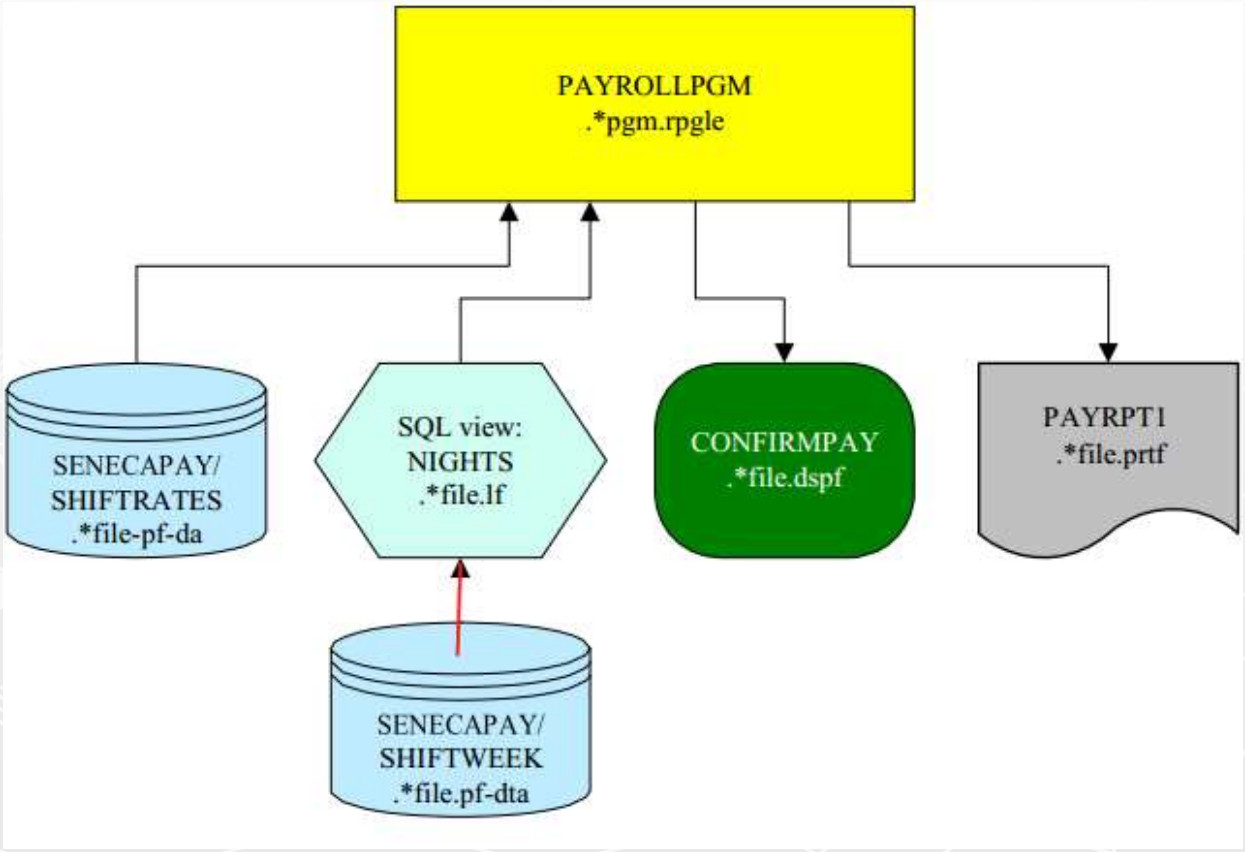
- In ILE (Integrated Language Environment), your programs can call either a program or a procedure. However, the caller uses separate call statements for programs and for procedures.
  - Dynamic Program Calls:
    - a program calls another program:
      - An OPM program can call another OPM program or an ILE program
      - An ILE program can call an OPM program or another ILE program
      - …
  - Static Procedure Calls:
    - used to call:
      - a procedure within the same module
      - a procedure in a separate module within the same program
      - …

Dynamic and Static Calls

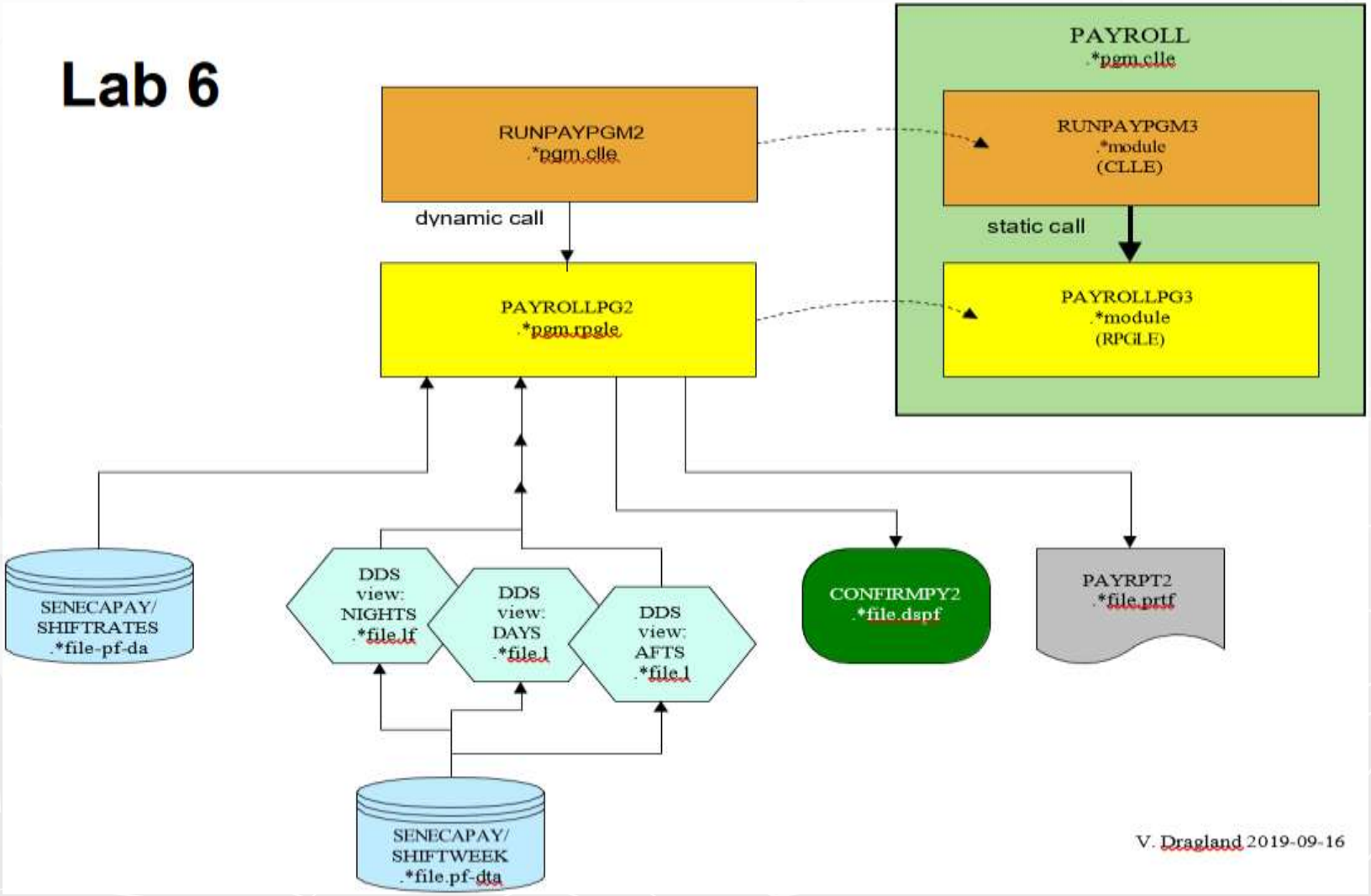
Program	Procedure	Comments
*PGM	*MODULE	Compiled to
CRTBNDRPG	CRTRPGMOD	Compile command for RPGLE code
CRTBNDCL	CRTCLMOD	Compile command for CL code
	CRTPGM	Command to create *PGM with modules combined
DYNAMIC CALL	STATIC CALL	
CALL	CALLB (CALL BOUND)	The statement called by RPGLE code
CALL	CALLRPC (CALL PROCDURE)	The statement called by CLLE code
	DSPPGM	Display program with module(s)

RPGLE Business Application
(with Printer/Database Files)

Lab 5 files:



RPGLE Business Application
(CLLE Driver/DP&SP Calls/File Overrides)



Homework

- Review lecture notes.
- Complete Lab 6

Lab 6 Demo

The End