BCI433 - IBM i Business Computing

Week 2: Intro to Dev Tool – RDi, Creating physical file using DDS

Agenda

- Introduction to RDi
- Library Lists
- Simple RPGLE code in fixed-form(at)
- Creating a physical File
- Lab2
- QuickCheck (Questions)

Lesson Objectives

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

- Apply and set up IBM development tool RDi and maintain library list in RDi.
- Create and successfully compile an RPGLE program in RDi
- Create a physical file using DDS code
- Enter data into the physical file and use different way to display or query data in the physical file
- Print out a compiler listing

Introduction to RDi

RDi - IBM Rational Developer for i

- Formerly RDp - Rational Developer for Power Systems
- Based on Eclipse Technology
  - RDi = Eclipse + "Remote System Explorer" package
- Part of IBM's suite of products designed to enhance programmer's productivity
  - the PC version of PDM
- Connection (to Seneca ZEUS server)
  - zeus.senecac.on.ca

Using RDi

- Tutorial or resources: under 'Help' menu item
  - Help>Help Content
  - Help>IBM I RSE Getting Started
  - Help>Web Resources
- Other useful operations:
  - Window>Show View>…
  - Window>Perspective>Reset Perspective…
  - File>Switch Workspace>…
- Workspace
  - starts from an empty folder.
  - used to store configuration (connection, initial LIBL, initial program, filters,…) only. Source code is on remote server, not in workspace!

Using RDi

- Local (PC) Files vs IFS (IBM i) files
  - BCI433 course materials:

    IFS Files/Root file system/ BCI433/BCI433…
  - Your PC folders: Local/Local File
  - Copy and paste within RDi's Remote Systems view (only)
- Using RDi as PDM
  - Expanding Work with objects…, … under Objects
- Library List
  - Adding Library List Entry…
- Initial Library List & Initial Command
  - Objects -> property -> …
- WRKACTJOB, WRKSPLF in RDi

Library Lists

- A Library List usually consists of 3 parts:
  - System libraries (from system value)
  - Current library (from user profile)
  - User libraries (from system value)
- How the IBM i finds Objects
  - CL commands prompt you for the object name and *library* name.
  - *LIBL is the default for *library*, means library list
- In ACS, to view your library list: DSPLIBL
  - How to do this in RDi?
- In ACS, to add a library to library list:

  ADDLIBLE BCI433LIB

  - How to do this in RDi?

Library Lists

- Library lists are built when you sign on System i and destroyed when signoff.
- ADDLIBLE command can only add a library to the list for the current session; the library you added will no longer on the list after your sign on next.
- How keep a library, e.g. BCI433LIB, always on the list? Solution:
  - In RDi: use *Initial Library List*, and Initial Program
  - In ACS 'green screen'? See Lab 4 in week 5

Languages we Learn in BCI433

- CL (Control Language)/CLLE
  - How we communicate with operating system, manage (RPG) applications
  - Similar idea to Unix Schell

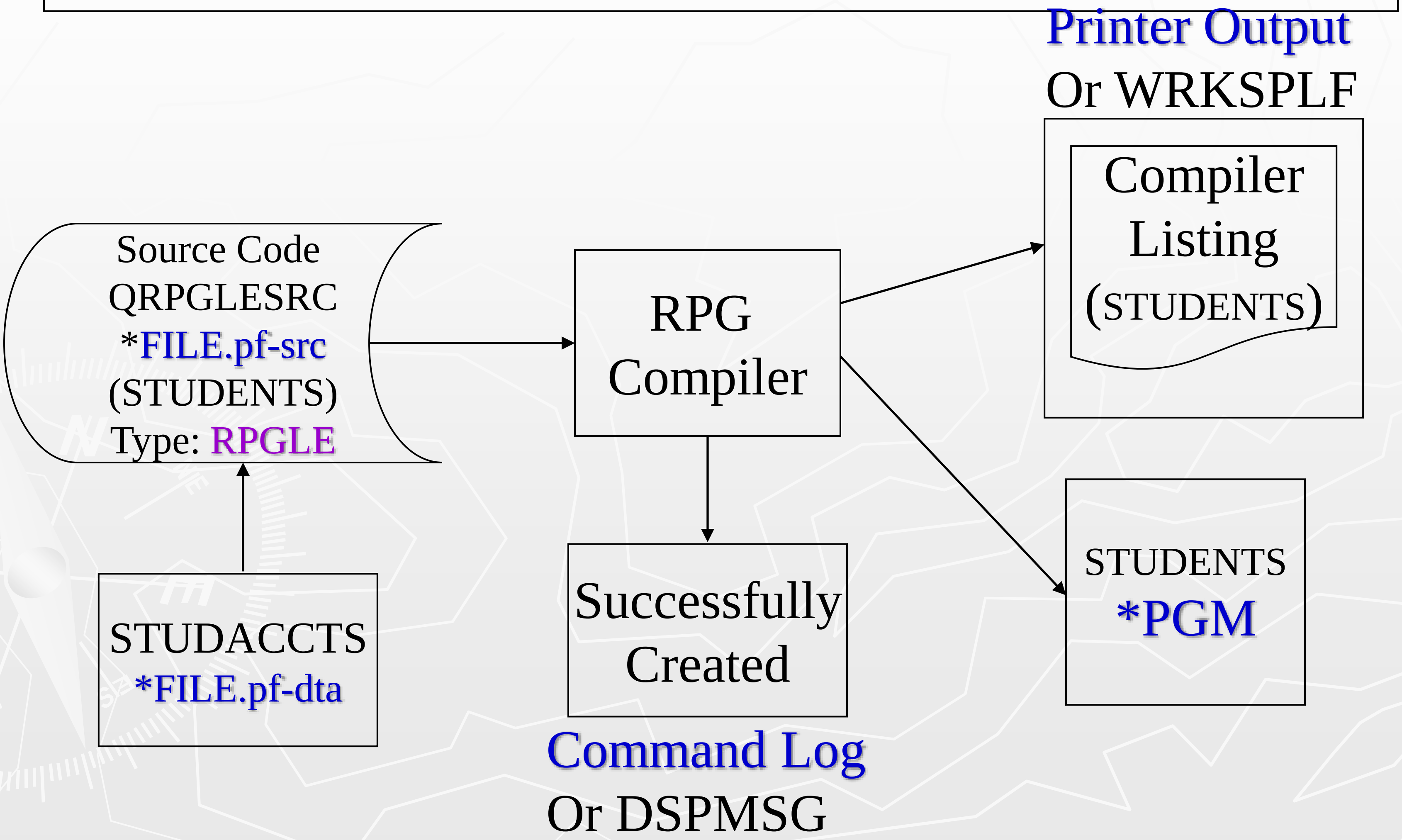
- RPG (Report Program Generator)/RPGLE
  - How we **access** database and generate reports
  - C does the same thing

- DDS (Data Description Specifications)
  - How we **define** database (physical file), display/printer files

Creating a simple RPGLE program

- Source physical file name: QRPGLESRC
- Member type: RPGLE
- Compile
  - severity level <= 10
  - compiler listing:
    - a printer (spooled) file, with the name same as program's
    - Access using ACS Pinter Output (the 8th icon) or WRKSPLF
  - make sure the library that contains physical file(s) used by the RPGLE program must on the library lists in both
    - RDi (the development environment), and
    - ACS (the program execution environment)

Creating & Compiling an RPGLE program



DB2 – the Database

- The database – DB2 is built into IBM i.
- Database objects, e.g. physical files, are created using either DDS coded programs or SQL.
- DB2 consists of
  - Physical files (also called tables)
  - Logical files (also called indexes or views)

Data Description Specifications

- DDS is used to define data.
- DDS is a sensitive language. Everything bust be typed in UPPERCASE.
- DDS program structure:
  - start out with file level keywords, or attributes that apply to a file itself.
  - Record format means a layout or a screen.
  - The fields are then listed
  - At end, the access path (key) information.

DDS program example

- This DDS code will create a physical file to store Account Information.

```
T.Name++++++RLen++TDp.......Functions++++++++++++++++++++
_ __________ ________       UNIQUE___________________
R ACCTPFR___ ________       TEXT(‘ACCOUNT INFORMATION’)__
_ ACCT   ___ ____3S_0       COLHDG(‘ACCOUNT’ ‘NUMBER’)___
_ ACCTNAME _ ___30A__       COLHDG(‘ACCOUNT’ ‘NAME’)____
_ ACCTDATE__ _____L__       COLHDG(‘ACCOUNT’ ‘DATE’)____
_ AMTOUT  __ ____7P_2       COLHDG(‘AMOUNT’ ‘OWING’)____
K ACCT____ ________       ____________________________
```

  - **R: ACCTPFR** is the **r**ecord format name
  - **K: ACCT** is the **k**ey, i.e. index field
  - **UNIQUE:** makes **ACCT** a PK

Field Data Types

Entry	Meaning	
P	Packed Decimal	Takes up less space in memory than a Zoned Decimal (A Packed Decimal stores two digits in one byte)
S	Zoned Decimal	If you leave the number of decimal places blank, then none are assume. (A Zoned Decimal stores 1 digit in byte)
B	Binary	
F	Floating-Point	
A	Character	If you leave the data type blank, then character is assumed.
H	Hexadecimal	
L	Date	
T	Time	
Z	TimeStamp	

* We don’t often see Binary, Floating-Point and Hexadecimal fields in business applications

Creating a Physical File in RDi

- Source Physical File: QDDSSRC.*file.pf-src
- Member name: e.g. ACCOUNTS
- DDS code member type: PF
- Compile:
  - CRTPF vs CHGPF?
  - The Physical File created: ACCOUNTS.*file.pf-dta
    (the created object after compilation)

- In ACS (white screen)
  - Entering data using DFU: UPDDTA ACCOUNTS
  - Query data using Query/400: RUNQRY *N ACCOUNTS
  - Query data using SQL: Select * from ACCOUNTS
  - Show data in RDi?

Lab 2 demo

Homework?

- Review lecture notes.
- Install RDi 9.6 (before Lab 2 starts)
- Complete Lab 2
- Lab 1 due

The End