AO IBM i Advanced Modernization Workshop Curriculum

This workshop is intended to provide the IBM i professional, specifically the RPG programmer, with an overview of the newest capabilities which have been provided in the RPGILE language and elsewhere in IBM i to allow for vastly improved productivity, agility and performance for the development of systems.

In addition this workshop will provide methods and techniques to allow for the gradual low-risk modernization of existing applications to take advantage of all the developments and advancements of the IBM i platform, the DB2 database and the RPGILE language.

The following sections briefly outline the topics to be covered during the approximately 48 hours that the workshop will last. From experience of this type of workshop, it is not really possible to say which topic/s will be covered in which session. This depends largely on the feedback and involvement of the students attending, as we tailor every workshop based on the actual coding and design practices in use at the particular installation.

As the workshop is designed to build progressively from one session to another it is essential that the successful applicant be committed to the full workshop. As there will also be requirements for additional exercises and additional reading outside the workshop sessions it would be advisable to provide for about 3 hours of time between sessions to accommodate this “homework”.

IBM Rational Developer for Power Systems (RDi)

RDi is a highly functional and flexible systems management and powerful integrated development interface for, but not limited to IBM i.

As a result of this capability and RDi’s ability to allow the user to personalize and customize the interface to suite individual preferences, it is not possible to provide a step-by-step “how to” for the product.

This “Walk-About” will introduce the student to the multiple capabilities of RDi and provide guidelines for the use and customization of the features and functions.

RDi will be used exclusively throughout the workshop and will therefore provide many opportunities to consolidate understanding and skill with the product.

DB2 Relational Database

The DB2 relational database definition, contents and structure are the basis of all systems development.

Normalizing Data

DB2 on IBM i is a relational database and to gain the best results the data in the database should be properly normalized and correctly related.

Normalization Theory

Discuss and review the generally accepted theories of normalization in a relational database.

Practical Normalization

Examine the differences between theory and practice in the normalization of data and discuss the reasons for the differences in practical applications.

Database Components

Taking a look at all the different components which go to making up a DB2 Normalized, Relational Database.

Data Dictionary

What is a data dictionary? Why is it necessary and how does one go about implementing one.

Additional advantages of a properly constituted and managed Data Dictionary.

The Data Store

The “Data Store” is the heart of any database as it is the definition of the structures which contain the company’s data. This is the single most valuable asset of any company no matter how big or how small.

DDS Physical Files

DDS is the original definition tool for “Physical Files” in the DB2 database. Access from application code is by ISAM (Indexed Sequential Access Method).

DDL Tables

DDL is the modern way to define “Tables” for the DB2 database. This uses the SQL engine for table access.

DDS vs. DDL

Discuss the pro’s and con’s of DDS as opposed to DDL.

Viewing the “Data Store”

The normal method of accessing the data and information in the “Data Store” is through various different logical constructs. These constructs, despite being given different names, contain no data, only routes to the data and are therefore logical or virtual rather than physical.

This section discusses the various different logical constructs which are now available on the DB2 database, looking at comparative advantages and disadvantages of each type with very specific focus on the differences between DDS vs. SQL types.

DDS Logicals – Views, Joins, Multi-Formats

DDL Indexes - BR Indexes, EV Indexes

DDL Views – Views

Relational Components

In a DB2 relational database, the “relational” components are those elements which provide the relationship management and other powerful functionality to the IBM i DB2 database.

This section discusses the what, why and how of these relational components, illustrating the benefits obtained from the use of these components for a very small investment in time and effort.

Primary Keys, Unique Keys, Check Constraints and Referential Constraints

Journals & Receivers

Journals and their associated receivers are an absolutely mandatory requirement when constructing a DB2 Relational database. Many other benefits are available from the use of these journals.

This section looks at the setup and usage of journals in the context of a modernized DDL/SQL database.

Commitment Control

As with journals, commitment control is required for a correctly functioning DB2 relational database.

We look at the setup and usage of commitment control and the “commit” cycle within a modernized database and also within its associated applications.

Trigger Programs

Although not mandatory, trigger programs are certainly considered “best practice” when it comes to DB2 databases.

Trigger programs are conventional programs written in any ILE language, including SQL, and attached to an event occurring on a “physical” database construct. A new trigger event has just been made available for SQL logical files.

This section discusses how to create trigger programs and the benefits of triggers in modernizing applications.

External (RPG) vs. SQL

We will also investigate and discuss the differences, advantages and disadvantages of SQL triggers as opposed to RPGILE (External) triggers.

I/O Servers

I/O Servers, although not an official IBM i DB2 construct, are considered “best practice” in the process of application modernization around an already modernized DB2 database.

We discuss the principles and creation of the I/O Server programs and/or procedures and how these entities form the “bridge” between the application and the database, reducing application requirements in the process.

RPG IV

RPG IV is the latest version of the RPG language available for IBM i and is the primary development language for this platform.

These sections of the workshop will examine the extensive capabilities of the RPG IV language and the various ways in which the language fits into the ILE (Integrated Language Environment) supported by the IBM i platform.

Free Form in RPG IV

The free-form variation in RPG IV provides a huge benefit to the RPG programmer and provides access to capabilities of the language not available in the fixed-form style.

We will discuss the benefits and dangers of free-form coding and consider a set of “best practices” for an organization. Please be aware that we consider that coding in the free-form style is in itself a “best practice” and therefore all provided examples and exercises will be in free-form.

Built-In Functions

There are a large number of built-in functions (BIF’s) available in free-form and the workshop will examine these and discuss their actions and the various techniques and ways these BIF’s can be used.

Programs

The workshop will examine the many different ways that programs can be constructed using the various constructs available in ILE RPG IV. The emphasis here is on the packaging of re-useable code blocks to reduce code redundancy and improve maintainability and agility.

Procedures
Discussion on the various methods of coding procedures and the use of parameter interfaces.

Functions
Creating application functions similar to the RPG BIF’s and the benefits of these in the application modernization process.

Modules
Creating program modules and how these are used in the building of programs and service programs.

Packaged Code

Here we will examine various different types of modernized packaged code and understand the advantages of modular application development as well as the way to code and compile these packages.

These will include Trigger Programs, I/O Servers and Service Programs.

Pointers

Pointers are an incredibly powerful, high performance data manipulation tool available in RPG IV. The application of pointers in the modernization of applications is very diverse and can be beneficial to the programmer in many ways to reduce the number of code lines required and to improve performance.

The workshop will examine many ways in which pointers can be used by the programmer.

Data Exports/Imports

Due to the trend in modernized code for a more modular construction of application programs and the increased use of service programs to contain re-usable code, the transfer of information and parameters between these modules becomes more important.

This workshop explains the usage of data exports and imports between modules to simplify the connection between modules and improve performance.

Activation Groups

Activation groups are a part of the ILE environment designed to compartmentalize the resources for any specific job running on the IBM i.

This workshop will examine the various types of activation groups, and the ways in which the programmer can utilize their capability to enhance the control of application resources and once again improve application performance and agility.

Error Handling

Due to the modularization and fragmentation of code in the modernized application it is necessary to have a consistent and easily managed way of handling both program and database errors, to ensure that the application does not “crash”.

This workshop looks at the “best practice” method of handling error at all call levels of an application. This method closely reflects the methods the O/S itself uses to handle/manage errors.

User Spaces/Indexes

User space (*USRSPC) and user index (*USRIDX) objects are powerful, high performance objects which can be used for an unlimited number of purposes. User spaces are used extensively by system API’s (next topic).

We examine these two objects and the various ways they can be used in detail not only to better understand there usage in the API’s but to make their capability and flexibility available to the programmer in the creation of application code.

System API’s

System API’s (Application Programming Interfaces) are callable system supplied programs giving the application programmer access to a wide variety of system functions and information which is not available in the normal way. There are an increasingly large number of API’s available for many purposes.

In the workshop we will first examine the generic principles of working with API’s and then will look at the list of available API’s and examine some of those, more relevant to the application programmer, in greater detail.

User Defined Commands

User defined commands have been available for a considerable amount of time and are the best way to collect parameters required to initiate a job, both batch or interactive.

We look briefly at the source language required to create a user defined command and the capability that is available to the developer with very few lines of code.