

**Course Name:** Crossover to Modern RPGLE with ILE

**Course Code:** ISR4X

**Duration:** 3 Days

**Price:** £ 1,750 plus VAT

**Course Outline:** This course brings RPG/400 programmers' skills up to date using modern RPGLE programming techniques. Introducing features of the programming language such as FREE FORMAT coding, built in functions, true date fields and pointers, sub-procedure coding, debug tools and an introduction to using the integrated programming environment (ILE) with reusable procedures, modules and service programs.

**Prerequisites:** Delegates attending this course must be proficient in RPG/400, either through attending the RPG/400 programming courses and/or gaining the equivalent practical experience using RPG/400. In practice, this means delegates are expected to have at least 6 months experience using RPG/400.

#### **Course Content:**

- **Conversion & Specification Changes**
  - Converting RPG to RPGLE
  - Benefits of RPGLE
  - Two stage program creation
  - RPGLE H-Spec layout and associated keywords.
  - Free-form CTL-OPT statement
  - RPGLE F-Spec and associated keywords.
  - Free-form DCL-F statement
  - RPGLE layout of the C-Spec and need for Operation Extenders
  - Coding 'Extended Factor 2' and expressions
  - Logic control operation codes with extended factor 2

#### **Variable Definition & Data Areas**

- D-spec definition of standalone fields and constants.

- Variable length strings
- D-spec definition of data structures:
  - Qualified, External, Like, Indicator, Program status
- Data Area Definitions (hard and soft coded)
- Using Data Structures & Data Areas
- Free Format DCL- statements
- External ALIAS support

## Free Format Code & Built In Functions

- Free format calculation specifications
- EVAL and EVALR operation codes
- Arithmetic precision control in free format code
- Free format alternatives to traditional RPG operation codes
- Free format keyed database access %KDS
- Selective updating %FIELDS
- File I/O without indicator using BIF's instead of result indicator
- Intercept errors with %ERROR or MONITOR
- String handling BIFs
- Date & time related BIFs
- Data type conversion BIFs

## Arrays and Pointers

- D-Spec Array definition
- Overlaying arrays
- Compile time data
- Using array BIF's: %LOOKUP, %XFOOT, %SUBARR
- Multiple occurrence data structures
- Multi-dimensional arrays
- Complex Data Structure definition and interrogation
- Sort & Search data structure array
- Pointer variables and %ADDR
- BASED variables and Templates
- Dynamic storage and associated BIFs
- Null terminated strings and %STR

## Prototyped Calls & Sub-procedures

- Sub-procedure Definition
  - Prototypes & Procedure Interface
  - Procedural Call – CALLP
  - Local and Global variables
  - Local Files
  - Reusing a sub-procedure
  - Activation group implications
  - Methods of passing parameters: By Reference, Read-Only Reference, Value
- Optional parameters: \*OMIT, \*NOPASS, %ADDR, %PARMS
  - Faster return values with RTNPARM (and %PARMNUM)
  - Free format procedure statement DCL-PROC

## ILE Concepts and introduction

- ILE v OPM
- Relationship between program and module objects
- Dynamic call v Static call
- ILE program and module object information – DSPPGM, DSPMOD
- Convert CLP to CLLE & procedure Call – CALLPRC
- Binding CL and RPG together into a single program
- Update ILE program objects
- Include copyright into module objects
- Application isolation with Activation groups
- Controlling which activation group is used
- Reclaim resources for an activation group

## RPG Modules

- Bind RPG modules into programs
- Make source code available for interactive debug
- Defining PEP
- Understand export / import concepts
- Using export on sub-procedure definitions
- Define and use NOMAIN modules
- Code Cycle-Main and Linear-Main procedures

- Teraspace storage model v Single-level storage model
- Bindable APIs

## Service Programs & Binding Directories

- Bind by copy v Bind by reference
- Service Program creation and Maintenance
- Service Program exports and Public Interface
- Control of Service Program Signatures
- Signature Versioning
- Binder Language and QSRVSR
- Public and Private procedures
- Binding Directory objects
- Use of Binding Directory to simplify the program creation process

## Finding Runtime Errors – Debug

- Enabling a program for debug
- Start debug (STRDBG)
- Set break points and check variables
- Track program logic
- Debug Encryption
- Tips n' techniques

**Follow on Courses:** Add internal power to application programs with the Power RPG with SQL course (ISSQE).

**Schedule:**

**2026**

Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun		

Dates on request – Please call or email for details