A White Paper

How to Migrate AS 400 applications to Windows

A Complete Solution to Achieve Platform Neutrality

This paper is a discussion of how to create platform independence by executing i OS (AS/400) applications on the Windows operating system and MS SQL as the database. The approach behind this paper is to explain how to execute AS/400 DB2 400 applications under Windows/MS SQL without having to rewrite the RPG or COBOL/400 application code.

OVERVIEW

Software companies and end users developed on the IBM AS 400 platform because it offered a number of advantages – database capabilities at the machine level, stability, wide acceptance and excellent support from IBM. Within the business community. As the industry matured, the marketplace has sought more open growth platforms such as Windows or Windows. Starting over every time a new business requirement arises is neither practical nor justifiable. An approach is needed which enables applications to adapt to rapid business and technology changes. Many companies wish to operate their AS 400 applications under Windows instead of i OS. They wish to run the software in a virtualized environment like VMWare, they wish to use industry standard blades instead of proprietary System i or IBM P series hardware running i OS and DB2/400. They would like to write to an industry standard database like MS SQL and they would like to provide a Cloud-based, graphical UI that is intuitive and not menu-driven. This paper will explain how these objectives can be achieved.

This white paper will discuss how to migrate RPG or COBOL code, CL and DDS to Windows and to execute on MS SQL. AS/400 applications were generally written in RPG/400 and were subsequently developed in RPG ILE. The usual compiled environment of AS/400 applications also include CL and DDS.

In order to accomplish the goals discussed earlier in this paper, the code must be recompiled. In this instance, we are using the compilers available from Infinite Corporation. Infinite has developed a family of products named Infinite i. As part of this product family, there are several compilers that will provide for the recompilation of RPG.400, RPG ILE, COBOL/400 and COBOL ILE, CL and DDS source code. Once recompiled, this code will be executed as native object code for Windows Enterprise Server 2008 r2. This is a 64-bit environment.

This approach uses what you already have; proven AS 400 code and resources, and allows you to execute functionally rich applications on a lower-cost 21st century platform.

Using infinite i, end-users can combine all the 'competitive edge' benefits of Windows with the power and stability of their RPG or COBOL applications. Since the underlying code is the original, the skills required to maintain the core logic remain the same. In fact, this process maintains source code integrity and simply allows it to be distributed on open platforms.

For AS/400 ISV's infinite i provides a way to access thousands of new sales prospects, without taking years to rewrite applications. ISV's can maintain the same source code and continue development on the IBM i OS (if they desire) and can deploy those programs on the i OS or Windows or even Windows via Infinite i.

To illustrate what infinite i can do we will look at how infinite i deals with re-hosting an AS/400 application to run on Windows.

The Challenge

Presented with the challenge of running an AS/400 application on Windows, the problem that first comes to mind is that of source language; how to find a way of compiling thousands of programs written in proprietary CL, RPG or COBOL on Windows? But, in fact, the language issues are only the small tip of a large iceberg.

An AS/400 comes with a standard, mature and sophisticated environment designed to meet the needs of commercial applications. i OS (OS/400) provides the fundamental services required by an application-service such as print spooling, message handling, batch job process and, of course, the database.

In contrast, the services provided by Windows are less complete and less rich and where an adequate service is provided, it is invariably *different* to the service on an AS/400 in some significant respect.

AS/400 applications depend on these services as much, if not more than, the CL, RPG and COBOL languages that are used to define the application algorithms. So, the most vital components of infinite i are not the language compilers but the Infinite Deployment Environment. This environment contains internal DLL's holding a legion of function that map OS/400 services on to underlying Windows services and so provide an OS/400-like 'shell', in which application programs can execute.

In this paper we examine the Big Four Services:

- Job management
- Database access
- User interface
- Printing

And other service functions, to see how infinite i enhances standard Windows architecture to implement OS/400 functionality.

We will then look at the process by which an iSeries application is built on Windows.

1. Job Management

When an application program is compiled, Infinite i first generates intermediate C code (analogous to the Machine Code generated by the OPM and ILE compilers) and then uses the Microsoft Visual C++ compiler and linker to build native Windows DLLs, LIBs and EXEs. The characteristics of these components are similar to those of a program object on the AS/400. The compiled components can be dynamically loaded into and unloaded from a running job. They can call other modules and return control without being unloaded from the calling job (so enabling the RPG RETURN functionality).

In addition to compiling application programs into Windows Standard DLLs, LIB and EXEs, Infinite i implements most of the standard OS/400 programs as DLLs with the same characteristics. An application running on Infinite i will find job and program models very similar to those on the AS/400. The job management service functions provide the glue to link the programs together at runtime and implements job attributes such as the library list, program stack and job message queue that provides the backbone of any application job.

2. Database

Infinite i replicates the DB2/400 database within the suite of programs, providing a complete replacement of the IAS/400. The internal database provides all the routine functionality (physicals, simple and join logicals, select/omit, triggers, etc) as well as functionality that is specific to DB2/400 such as multi-member files, multi-format logicals, etc.

Alternatively, either Oracle or MS SQL Server can be designated as the repository for all the application's data files. In this case, Infinite i creates the database schemas (for example, mapping physical files to tables, logical files to views) and, at runtime, translates the application database calls (READ, READE, SETLL etc.) to the appropriate SQL statements. This translation takes place within the database

service function so the application programs remain independent of the implementation.

3. Printing

Infinite i replicates the traditional AS/400 print service functions that provide a high level of support for printer file DDS, together with an implementation of the OS/400 print spooling system (including out queues, device descriptions and writer jobs) that integrates naturally with the target operating system's Print Manager. This allows for the same level of application and operator control of print activity, using the standard commands (e.g. RLSSPLF) and user interfaces (e.g. WRKSPLF). Alternatively a user can choose to have output for designated printers released immediately to the Print Manager so that print jobs can be managed using standard interfaces.

4. Other Services Functions

The OS/400 object management system provides a variety of object types, most of which are listed in Figure 1. The Infinite i service functions implement all the object types that are in common use by applications. In each case, the object is implemented using one or more of the target operating system files and wrapper functions to provide controlled access to the object and to implement the standard OS/400 commands and APIs.

*JOBQ	Job Queue
*DATAQ	Data Queue
*PNLGRP	Panel Group
*DEVD	Device Description
*OUTQ	Output Queue
*JRN	Journal
*LIBRARY	Library
*MENU	Menu
*CMD	Command
*FILE	File
*USRPRF	User Profile
*USRIDX	User Index
*MODULE	Module

*SBSD	Subsystem Description
*TBL	Table
*PGM	Program
*MSGF	Message File
*MSGQ	Message Queue
*DTAARA	Data Area
*EDTD	Edit Description
*JRNRCV	Journal Receiver
*QRYDFN	Query Definition
*JOBD	Job Description
*USRSPC	User Space
*JOBSCD	Job Schedule

Fig 1 –AS/400 Object types

THE REHOSTING SUITE

1. Save Application On AS/400

Migrating an application to Infinite i is a straightforward procedure. The application needs to be saved on the AS/400 using SAVLIB to save all the application components (source and object) to save files. The save files are then transferred to Infinite i.

2. Load Application On Infinite i

Infinite i provides the command RSTAPP - Restore Application, to load the application components from the save files. This command:

- Restores all the source files and members
- Uses catalogue information in the save files to generate instructions describing how the source members need to be compiled
- Automatically restores other object types that are not created from source (e.g. job descriptions, message files)

3. Build Application

The Infinite i command RUNBIF - Run Build Instructions, executes the build instructions that were generated by the load phase, by making calls to the appropriate 'create' commands (CRTPF, CRTRPGPGM, CRTBNDRPG etc). When this command has completed, all the source members should have been compiled and the application will be ready for testing.

4. Application Changes

Infinite i provides development tools to allow changes to be made, built and tested. The SEU, PDM and the language compilers implement full source validation – they don't assume that they will be compiling valid source programs. Alternatively, if the application source is still hosted on an AS/400, changes can be made there and selectively transferred to and rebuilt on Infinite i using the procedure described above.

Appendix: Summary of ISERIES Features

- Supported Languages
  - ILE RPG
  - RPG/400
  - SQL RPG/400
  - COBOL/400
  - SQL COBOL/400
  - CL/400
  - CMD/400
  - DDS/400 (pf, lf, dspf, prtf)
- Development Support
  - Programming Development Manager (PDM)
  - Source Entry Utility (SEU)
  - Debugging
  - Interface w/ C language from within HLLs.
- Database Support
  - Database Engine
    - Internal DB2/400
    - Connectivity w/ MS SQL Server
    - Connectivity w/ MS SQL
  - Support for Multiple Members
  - Support for Multiple Formats
  - Joined Files
  - OPNQRYF
  - Triggers
  - Commitment Control
  - Query/400
  - DFU (Data File Utility)
  - LAPIS
- Job Management
  - Batch job processing
  - Job Scheduling
  - Subsystems
  - Joblog
  - Job Description
  - Data Queues (DTAQ)
  - Data Areas (DTAARA)
  - Message Handling
- User Interface
  - “Work With” panels
  - Menus
  - Command prompts
  - Panel Groups (PNLGRP)
  - Telnet 5250 Data Stream
  - User profiles
- Object Management
  - Save / Restore
- Printing
  - Spooling
  - Output queues (OUTQ)
  - Spool Writers (SPLWTR)
  - Local and remote printing
- Migration Support
- OS/400 API support
  - Most common APIs
  - USRSPC handling