

Migrate AS 400 Applications to Linux

Infinite Corporation **White Paper**

date

March 2011

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

INFINITE
c o r p o r a t i o n

TABLE OF CONTENTS

INTRODUCTION..... 3
 OVERVIEW 3
 Background 3
 Solution..... 3
 THE CHALLENGE 4
 Job Management..... 4
 Database 5
 Printing 5
 Other Services Functions 5
 THE REHOSTING SUITE 5
 Save Application On AS/400 5
 Load Application On Infinite iSeries 6
 Build Application 6
 Application Changes..... 6
 APPENDIX: SUMMARY OF AS/400 FEATURES **Error! Bookmark not defined.**

INTRODUCTION

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

OVERVIEW

Background

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. As the industry matured, the marketplace has sought more open growth platforms such as Linux 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 Linux instead of i OS. They wish to run the software in a virtualized environment like VMWare or Red Hat Enterprise Linux, they wish to use industry standard blades instead of proprietary System i or IBM P series running i OS and DB2/400. They would prefer to write to an industry standard database like Oracle.

Solution

This white paper will discuss how to migrate RPG or COBOL code, CL and DDS to Linux and to execute on Oracle. 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 includes CL and DDS.

In order to accomplish these objectives, 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 Red Hat Enterprise Linux 6.0. 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 Linux 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 deploy those programs on the i OS or Linux 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 Linux.

THE CHALLENGE

Presented with the challenge of running an AS/400 application on Linux, 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 Linux? 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 Linux 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 Linux services and so provide an OS/400-like 'shell', in which application programs can execute.

In this paper we examine the Big Five Services to see how infinite iSeries enhances standard Linux architecture to implement OS/400 functionality. We will then look at the process by which an iSeries application is built on Linux.

1. Job management
2. Database access
3. User interface
4. Printing
5. Other service functions

Job Management

When an application program is compiled, Infinite iSeries 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 ISERIES. 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).

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

In addition to compiling application programs into Windows Standard DLLs, LIB and EXEs, Infinite iSeries implements most of the standard OS/400 programs as DLLs with the same characteristics. An application running on Infinite iSeries will find job and program models very similar to those on the I SERIES. 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.

Database

Infinite iSeries replicates the I SERIES DB2/400 database within the suite of programs, providing a complete replacement of the I SERIES. 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 SQL Server can be designated as the repository for all or just a subset of an application's data files. In this case, Infinite iSeries 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.

Printing

Infinite i replicates the traditional I Series 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.

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 iSeries 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.

THE REHOSTING SUITE

Save Application On AS/400

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

Fig 1 –AS/400 Object types

Load Application On Infinite iSeries

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 catalog 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)

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, CRTRPGGM, CRTBNDRPG etc). When this command has completed, all the source members should have been compiled and the application will be ready for testing.

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.

The benefits of an AS 400 Migration

Most people would agree that operating RPG or COBOL applications written for the AS 400 environment can significantly reduce costs by migrating them to less expensive hardware. Of course, by migrating you are eliminating the costs for i OS, DB2 400, 5250 interactive costs, high maintenance fees hardware refresh costs etc.

What is less obvious is that executing RPG and COBOL applications under Linux means that they can be virtualized using industry standards like VMWare. Virtualization technology, which isn't available for the AS/ 400 incorporates High Availability as well as dynamic resource allocation. The costs and resources required for HA on the AS/400 platform are enormous. Since the operating environment is open, it can be allocated amongst non-IBM or IBM servers. The AS/400 isn't designed to address virtualized storage. Linux is multi threaded in this respect and provide high speed write capabilities to virtualized storage. Once the original RPG or COBOL applications are recompiled using Infinite i they run natively as 64-bit Linux objects and can take complete advantage of today's most advanced storage virtualization technology.

Perhaps the greatest benefits, however, are those enjoyed by the ability to write to Oracle instead of having to maintain data in DB2 400. Once the data is migrated and begins writing to Oracle it is now in a data structure that is consistent with most current enterprise applications. Of course, reporting is easier, browser-based and more intuitive, but the information can be analyzed and integrated with information from other enterprise applications like SAP using Oracle BI tools that already exist in most medium-size and large businesses.

APPENDIX: SUMMARY OF AS/400 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/ Oracle
- 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

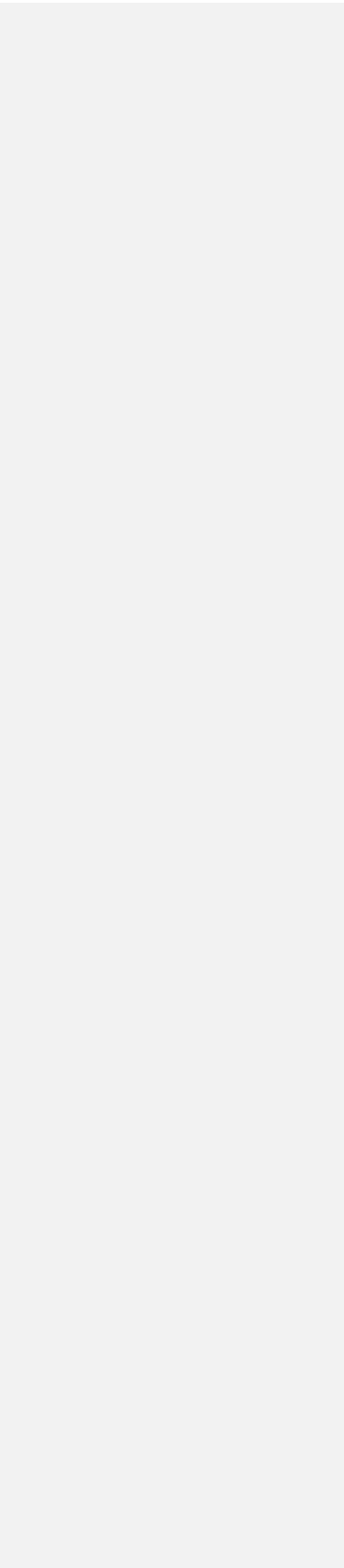
Printing

- Spooling
- Output queues (OUTQ)
- Spool Writers (SPLWTR)
- Local and remote printing

Migration Support

OS/400 API support

- Most common APIs
- USRSPC handling



INFINITE
c o r p o r a t i o n