An Infinite White Paper written by Bruce Acacio

Migrating RPG or COBOL Applications to C and Java

How to migrate RPG or COBOL applications to C and Java then build/integrate new Java modules and plug-ins
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>INTRODUCTION</strong></td>
<td>3</td>
</tr>
<tr>
<td><strong>OVERVIEW</strong></td>
<td>3</td>
</tr>
<tr>
<td>Background</td>
<td>3</td>
</tr>
<tr>
<td>Solution</td>
<td>3</td>
</tr>
<tr>
<td><strong>THE CHALLENGE</strong></td>
<td>4</td>
</tr>
<tr>
<td>Job Management</td>
<td>4</td>
</tr>
<tr>
<td>Database</td>
<td>5</td>
</tr>
<tr>
<td>Printing</td>
<td>5</td>
</tr>
<tr>
<td>Other Services Functions</td>
<td>5</td>
</tr>
<tr>
<td><strong>THE REHOSTING SUITE</strong></td>
<td>5</td>
</tr>
<tr>
<td>Save Application On AS/400</td>
<td>5</td>
</tr>
<tr>
<td>Load Application On <em>Infinite iSeries</em></td>
<td>6</td>
</tr>
<tr>
<td>Build Application</td>
<td>6</td>
</tr>
<tr>
<td>Application Changes</td>
<td>6</td>
</tr>
<tr>
<td><strong>APPENDIX: SUMMARY OF AS/400 FEATURES</strong></td>
<td>8</td>
</tr>
</tbody>
</table>
INTRODUCTION

This paper is a discussion of how to create platform independence by migrating RPG or COBOL applications to C and Java. This approach uses both compilers and a deployment environment to execute recompiled RPG or COBOL written for IBM iOS (AS/400) and deployed under Linux without having to rewrite the RPG or COBOL 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 iOS. 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 iOS and DB2/400. They would like to write to an industry standard database like Oracle 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.

Solution

Infinite i is an advanced software toolset which extends IBM AS/400-based applications to open architecture platforms running on Linux and Windows and using Oracle and SQL databases. This approach uses what you already have, proven AS/400 code and resources. By migrating core logic to C and Java, it allows developers and end-users to add functionality using Java or C instead of continuing development in RPG or COBOL.

This approach allows you to build functionally rich applications, accelerating the development and delivery while reducing cost and overall risk.

Using Infinite i, AS/400 users can combine all the “competitive edge” benefits of growth platforms with the power and stability of proven RPG or COBOL applications systems and effectively harness the same applications and skills across platforms.

For AS/400 ISV’s Infinite i provides a way to access thousands of new sales prospects, without taking years to rewrite applications.

To illustrate what Infinite i can do we will look at how Infinite i migrates RPG or COBOL to C and then deploys the AS/400 application on Linux. Once migrated, the screens are deployed in Java and new development can be done in Java which is integrated with the migrated business logic.
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. iOS (OS/400) provides the fundamental services required by an application-service such as print spooling, message handling, batch job processes 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 i enhances standard Linux architecture to implement OS/400 functionality. We will then look at the process by which an iSeries (AS/400) 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 i generates C code (analogous to the Machine Code generated by the OPM and ILE compilers) and then uses the Linux C compiler and linker to build native Linux 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 Linux 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.

**Database**

*Infinite i* replicates the AS/400 DB2/400 database within the suite of programs, providing a complete replacement of the AS/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 SQL Server can be designated as the repository for all or just a subset of an 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.

**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.

**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.

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*. 
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 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, CRTRPGPGM, 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.

BROWSER DEPLOYMENT

The display file management service functions provide full support for display file DDS (including subfiles, window formats and user-defined data streams). *Infinite i* completes the process with tools to convert AS/400 green screens to browser deployed graphical screens with true browser deployment and functionality. Built on the foundation of the Java/Eclipse, *Infinite i* combines a number of features including a visual editor to design graphical rendering of text screens, configuration and creation of standard recognition’s distributed agents, browser deployment of legacy applications, remote client printing, an environment to develop new functionality and integrate applications and support for smart clients, wireless and handheld clients. Infinite Cloud re-engineers workflow to consolidate screens or accommodate changing business processes.

NEW DEVELOPMENT

The Java/Eclipse IDE included in *Infinite i* is already integrated into the newly migrated environment. Access to Eclipse and Java plug-ins and libraries are available on an open source basis. It also manages
Application Modules so applications from the IBM AS/400, IBM Mainframe, Linux, UNIX, and Windows environments will share the same interface with each of them able to integrate transparently with the others by a shared common format—Because programs developed in Java are designed to integrate tightly with enterprise applications written in native C, the programs developed outside of the newly-migrated RPG programs now in C will easily integrate with new modules built in Java or with other Java or C based applications in your enterprise. These will be tightly integrated—both visually and functionally—using Infinite tools.

**Conclusion**

Infinite offers a path to migrate AS/400 applications that is fast, efficient and cost effective. In one project, the Infinite tools can translate your RPG or COBOL source code to C, execute your application to Linux without any further code changes, migrate data from AS/400 DB2/400 to Oracle, graphically render screens and update how the user interacts with the applications.

Once in this environment, you can take advantage of industry standard systems and storage, consolidate and virtualize servers, employ an HA/DR schema at no additional cost, integrate applications in your environment and take advantage of the features of middleware offerings like JBoss, WebSphere or WebLogic.

# # #
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