



Date: May 18, 2011

# Migrate System/36 to Linux

Migrating from AS/400 in System/36 applications to open systems

Copyright © 2011 INFINITE Corporation. IBM Midrange, AS/400, OS/400 and System/36 are registered trademarks of the IBM Corporation. Windows, Internet Explorer, Visual Basic, VB Scripting, HyperV are registered trademarks of Microsoft Corporation. Dell is the registered trademark of Dell Corporation. Oracle, Sun and Solaris are registered trademarks of Oracle Corporation. HP, HP-UX and Integrity are registered trademarks of Hewlett Packard Corporation. Red Hat Enterprise Linux (RHEL), Red Hat Enterprise Virtualization (RHEV) and JBoss are registered trademarks of Red Hat Corporation. VMWare is a registered trademark of VMWare, Inc. Java and Sun are registered trademarks of Oracle Corporation. INFINITE Cloud, INFINITE i and Infinite 36 are registered trademarks of Infinite Corporation. All other product and brand names are registered trademarks of their respective companies.

No part of this document may be reproduced without written permission of INFINITE Corporation.

**CONTENTS**

**OVERVIEW..... 3**

**THE CHALLENGE ..... 3**

**REHOSTING SYSTEM/36 APPLICATIONS..... 3**

    Job Management..... 4

    Data Handling..... 4

    Printing ..... 5

**SUPPORTED FEATURES ..... 5**

    XML Support..... 5

    POP LIBR Browse Termination Options..... 5

    SUBRWD RPG Subroutine ..... 5

    FORMS OCL Statement Now Supports ROTATE and DRAWER Keywords ..... 5

    Sending Printer Reports via Email ..... 5

    WRKUSER Shows Information on RPG Sub-Programs ..... 5

    Improvements to Performance of POP..... 6

    Printer Output Merging With Predefined Templates via OVERLAY Keyword ..... 6

    Ability to Suspend New User Sessions from Starting..... 6

    Virtual VTOC File Groups..... 6

    RPG Debugger Indicates Execution Coverage ..... 6

    Barcode Printing Support..... 7

This paper is a discussion of how to create platform independence by executing SSP (System/36) applications on the Linux operating system and Oracle as the database. This paper details in non-technical terms how to execute System/36 flat file applications under Linux/Oracle without having to rewrite the application code.

## **OVERVIEW**

The System/36 was arguably IBM's most successful system introduction. Software companies and end users developed on the IBM System/36 platform because it offered a number of advantages – stability, wide acceptance within the business community 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.

Many companies wish to continue to use their System/36 applications because the core business logic still supports the functional requirements. However, there are inherent limitations to the System/36 platform – hardware and operating system compatibility, data access, communications, etc.

An approach is needed which enables applications to adapt to rapid business and technology changes. Application need to be able to run in a data center on 64Bit servers under Linux instead of SSP. Companies wish to run the software in a virtualized environment like VMWare or Red Hat Enterprise Linux. They wish to use write to an industry standard database like Oracle. They would like to provide a cloud-based, graphical UI that is intuitive and not menu-driven.

This paper will explain how Infinite allows the developer to achieve these goals.

## **THE CHALLENGE**

Presented with the challenge of running a System/36 application on Linux, the problem that first comes to mind is that of source language. How can you find a way of compiling thousands of programs written in proprietary OCL, RPG or COBOL on Linux? But, in fact, the language issues are only the small tip of a large iceberg.

The Infinite approach uses the proven System/36 code as well as the resources you already have and allows you to execute functionally rich applications on a lower cost 21<sup>st</sup> century platform.

Using infinite 36, 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 System/36 ISV's, Infinite 36 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 in a S/36-compliant development suite and on Linux.

To illustrate what Infinite 36 can do we will look at how the system deals with rehosting an RPGII or COBOL application to run on Linux.

## **REHOSTING SYSTEM/36 APPLICATIONS**

Moving from the propriety Midrange environment to open systems requires the source code be recompiled. In this instance, we are using the compilers available from Infinite Corporation. Infinite has developed a family of products named Infinite 36. As part of this product family, there are several compilers that will provide for the recompilation of RPG and COBOL. Once recompiled, this code will be executed as native object code for Red Hat Enterprise Linux 6.0.

A System/36 comes with a standard, mature environment designed to meet the needs of commercial applications. The SSP operating system provides the fundamental services required by an application-service such as print spooling, message handling, batch job processing and data handling. 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 Midrange service in some significant respect.

These SSP services are as critical to the execution of Midrange applications as the OCL, RPG and COBOL application algorithms. So the most vital components of infinite 36 are not the language compilers but the Infinite Deployment Environment. This environment contains internal DLL's holding a legion of functions that map SSP services to underlying Linux services and thus provide a System/36-like 'shell' in which application programs can execute.

In this paper we examine those services:

- Job management
- Data management
- Printing

### Job Management

When an application program is compiled, Infinite 36 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 Midrange program object. 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 native objects, libraries and executables, Infinite 36 implements most of the standard SSP programs as DLLs with the same characteristics. An application running on Infinite 36 will find job and program models very similar to those on the Midrange system. 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.

### Data Handling

Applications developed on the System/36 were written in a flat file data structure. The data exists in a single file in the form of rows and columns, with no relationships or links between records and fields except the table structure. This means that the data is only usable when it is called and displayed using the application's business logic.

Moreover, these applications were written at a time when disk space was architecturally limited and costly. Once a period was closed on the system, it purged all the transactional data and retained only summary data. Unless this data was copied on to a tape drive prior to closing, a tremendous amount of useful data was lost. This is the single greatest limitation of the System/36 environment. Infinite 36 has overcome this hurdle without requiring changes in the application itself. It takes the data stored in the ISAM structure and writes directly to a relational database. This means the data is available to other applications in real time without having to do awkward extracts,

During the migration process, the Infinite team will use file and input specifications to create externally defined files. The project team will migrate the existing data and, once validated, the new environment will write to an external database. This means that they System/36 applications that are now executing from Linux will write directly to Oracle and all data created from the system can be reported, analyzed and shared in Oracle's industry standard format

## Printing

Infinite 36 supports the traditional System/36 functions that provide support for printer files together with an implementation of the SSP print spooling system (including out queues, device descriptions and writer jobs). 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).

Infinite has also updated the basic EBCDIC dot matrix System36 printing capabilities so that it integrates naturally with the Linux operating system's Print Manager and can print to any network printer. With implementation of Infinite Cloud, all printers in the cloud are available for use regardless of type or physical location.

## **SUPPORTED FEATURES**

### XML Support

XML is perhaps the most influential industry standard to date, enabling instant access to structured data. XML is often used to represent business data exchanged between enterprises or within an enterprise's own business units. XML adds value to these relationships as it provides a common and consistent means for data definition through a combination of content and markup.

Infinite 36 support for XML allows data from flat files to be used to generate XML for consumption by other applications.

### POP LIBR Browse Termination Options

Support for termination options for FILE and LIBR browsers that allow you, for example, to exit a file browse and specify that you want to edit the same file using the Infinite 36 file editor.

### SUBRWD RPG Subroutine

For RPG programmers the simple task of calculating the day of week for a specific date required many lines of RPG code. This frequent programming requirement has now been made trivial through the provision of a new RPG subroutine, SUBRWD. SUBRWD takes a date in YYMMDD format and returns the corresponding day of the week.

### FORMS OCL Statement Now Supports ROTATE and DRAWER Keywords

The FORMS OCL statement now allows default session values to be set for spool job rotation and drawer selection values. The STATUS SESSION printer value screen has been extended to show the current session defaults for ROTATE and DRAWER values.

### Sending Printer Reports via Email

Printer reports generated by Infinite 36 can now be automatically emailed to a named recipient via a new MAILTO keyword of the PRINTER OCL statement.

### WRKUSER Shows Information on RPG Sub-Programs

Information on currently running jobs is available via the WRKUSER procedure. WRKUSER includes information on running sub-programs. A '+' character against an active job indicates that sub-programs exist and an option is available to allow the operator to drill down into the sub-program information. Alternately the DSPACTJ command may be used..

### Improvements to Performance of POP

Infinite 36 does not use disk catalogues. This means that libraries, source members and files can be created using native operating system commands. Setting the configuration variable S36POPCATALOG to YES can enable a POP-specific catalog to be utilized by POP LIBR and FILE to store library and file names. On start up POP need only read this catalogue and hence load times are significantly faster.

Creation or deletion of libraries and files from within Infinite 36 automatically update this catalogue. Any updates from the native operating system level will not be reflected and command key 20 is provided to resynchronize the catalog from within POP.

### Printer Output Merging With Predefined Templates via OVERLAY Keyword

A significant cost for companies is that of producing pre-printed stationary for reports. Simple changes such as telephone area codes can lead to further cost and wastage, as all existing stationary must be replaced.

Pre-printed stationary also requires careful alignment to ensure that the printer output is positioned correctly.

Printer overlay support in Infinite 36 now makes the creation of custom spool reports significantly cheaper as the need for pre-printed stationary is removed. Stationary templates can be designed using popular Windows applications such as Microsoft Word and automatically merged with printer output as the report is printed.

### Ability to Suspend New User Sessions from Starting

It is a frequent requirement for Infinite 36 system administrators to prevent new user sessions from starting but allow an administrator session to be active to perform necessary application or system maintenance.

The STOP LOGINS control command provides such a feature. Once executed only users with a security level of Security Officer or higher can start new interactive sessions.

START LOGINS re-enables all user interactive sessions.

### Virtual VTOC File Groups

The virtual VTOC facility supports the ability to specify a file group name against a named directory. This allows an entire file group to be located in a specific directory using just one entry in the VTOC file. The file group is identified by placing an asterisk after the file group name - for example AA.\* denotes the file group AA. So a valid VTOC source file could be:

```
DEFAULT=/usr/s36
DIR=/s36/test
AA.*
DIR=/s36/live
BB.*
```

### RPG Debugger Indicates Execution Coverage

If an RPG program is being executed under the control of the RPG debugger then it is possible to request that all lines executed are marked with a period in column one of the code display window. At any breakpoint or at the end of the program the extent to which the program has been executed can therefore be determined. Potential uses include code debugging and evaluating test coverage.

Barcode Printing Support

RPG programs can now easily print barcodes using the SUBR50 subroutine.

The printer database has been extended to include escape sequences that cause specific printer models to enter and exit barcode print mode. This ensures that RPG programs to generate reports with barcodes are independent of the type of printer to which the report will be printed. Further the programmer is freed from the onerous task of generating the appropriate escape sequences manually within the RPG code.

###