



## Case Study:

# Ingres Partnership with rPath Yields Industry-leading Icebreaker Software Appliance

### Introduction

From our early days as a University of California Berkeley research program, Ingres has focused on innovation and collaboration. Early Ingres code was developed by a devoted user community. The Ingres Project soon became Ingres Corporation, and after several years of stewardship by Computer Associates, the Ingres code was released to the community under an open source license.

Now independent of CA, it should come as no surprise that the new Ingres Corporation is once again focused on innovation and collaboration. Ingres' open source database (available under the GNU General Public License) enjoys wide use and support by a vibrant community. But the company was still looking for a way to make Ingres accessible to more organizations of all sizes.

In this paper, we will explore our decision to make the Ingres database available as a software appliance, and the process we undertook to get it ready for commercial use. We believe that delivering the database as a software appliance is key to our product strategy and gives us a distinct competitive advantage. We will talk about how we got to this point and what our experience has been.

### Situation

As an enterprise relational database company, Ingres is operating in a tough competitive landscape, dominated by a few large players. The strategic decision to offer software under the GPL structure made the database available to a large community of users and gave Ingres new revenue opportunities around support and maintenance.

To realize these revenue opportunities, we knew we had to make the software installation, maintenance and support easier and simpler. In particular, Ingres knew that a major factor in the success of the database was how well it interacted with the operating system. Dave Dargo, our CTO at the time said "The reason you install database software is to create database services. Unfortunately, most of the database software out there requires an operating system. Thus, you are stuck installing an operating system to support your database software."

But Ingres is not in the operating system business - it is in the database business. If we installed an operating system, we would also have to maintain it into the future. It would take quite a while to build an operating system team, and it is not among our core skills, nor where we wanted to devote resources.

The next best alternative was to engineer the database to one particular operating system or pre-approved software stack, but that would unreasonably narrow our options in the market. To ensure success, we want the broadest potential audience - we don't want to restrict access to only a certain portion of the market with the right infrastructure.

### The Software Appliance Lightbulb

As we worked with customers, we became more aware that they were less interested in another database or discussing various

feature sets that were available in the market place. Rather, customers were more interested in how they could quickly deliver their application to the market place or how fast they could take advantage of an application that they were rolling out. They were looking for ways to reduce cost and complexity in their environment without foregoing any of the functionality that their solutions demanded.

We quickly realized that a software appliance would meet our customers' objectives.

*A short definition: a software appliance is an application combined with a tailored operating system that readily installs on industry standard hardware or in a virtual machine. Software appliances simplify server applications by eliminating the hassles of installation, configuration and maintenance.*

In other words, one package contains everything you need to run an application - operating system, database, Web server, other third-party components, etc. Once we settled on this approach for packaging our database with other required components, we needed to actually build a working appliance.

### Enter rPath

rPath's rBuilder is a tool specifically designed to take the hard work out of building software appliances. Once we learned what it could do, and how much time it would save, we knew this was the best way to get our appliance to market quickly.

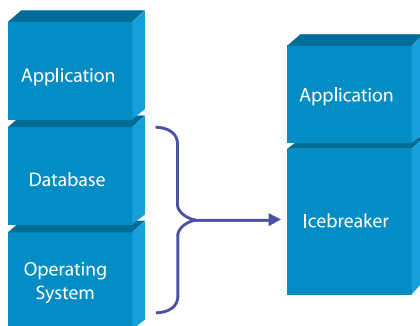
We also recognized that with the aid of the rPath solution, we could partner with other ISVs to create a complete solution that allows them to market their own products and solutions in a fraction of the time and significantly reduce total cost of ownership (TCO). Customers can now focus on the end game versus thinking of all the components required to build the solution. No longer does the customer have to piece the components together, test that all the pieces work

harmoniously and then repeat the process with each product update cycle. With rPath, Ingres can build complete solutions and deliver them to customers quickly and cost effectively.

## Building the Appliance

rPath's rBuilder is built on a foundation formed by Conary, an open source packaging tool. Conary uses repositories to store all the various bits and pieces of applications that will eventually go into an appliance. It also contains rPath Linux, a Linux distro that is optimized for appliance building. Our first step was loading the Conary repository with Ingres code.

We started by simply repackaging the binaries for our existing Linux release to build a proof of concept. Once the concept was proven, we moved to rebuilding our products in the rPath Linux environment entirely from source code. Building from source in the rPath Linux environment ensures that we have the tightest integration possible between our products and the underlying system components.



Since Conary build recipes are written in Python, we were also able to adapt our build recipe easily to leverage the packaging information in our existing source. This allows us to keep the naming conventions in our Conary packages consistent with the package names we use on our other platforms. It took 3 months for a single engineer to deliver a complete Conary implementation of our enterprise database product suite, built from a source base that includes nearly 25,000 files.

Conary comes pre-populated with many of the most popular open source applications. Conary also has a naming schema that assigns unique names to each item, or variation on that item, in the repository. Unique names also apply to builds, so you can clearly identify each component that went into a particular build. And because everything is in the repositories, you don't need to worry about pulling outdated versions of components into future builds.

Another element of the project was branding the rPath Appliance Platform console interface with Ingres' look and feel. The console also had to be modified to interact with the Ingres database setup routines.

The branding changes were accomplished through simple changes to the style sheets included in a dedicated branding package

## rPath

701 Corporate Center Drive  
Suite 450  
Raleigh NC 27607  
+1 919.851.3984  
+1 919.851.3985 Fax  
www.rpath.com

that comes with the rPath Appliance Platform. We were also able to extend the functionality of the Appliance Platform to include Ingres-specific administrative tasks, using the Appliance Platform plug-in architecture. Our plug-ins integrate seamlessly with standard system-level functionality, giving our users a single, cohesive interface for performing basic administrative tasks.

Building the actual appliance with rBuilder took practically no time. Once everything was loaded into Conary, and the Appliance Platform was customized, then the builds went quickly. rBuilder comes with business intelligence that analyzes which additional components from the repository are required in the appliance recipe. This dependency resolution process saves significant time and effort and gives us peace of mind that we are not missing anything the next time we build the appliance.

We are also able to output the appliance in a variety of formats to suit our customers' needs. Initially, we chose to do installable ISOs, but could easily use our appliance definition to output the appliance as a VMware virtual machine, if needed.

## Spreading the Word

The product went from concept to beta to launch in six months. We announced the beta release of "Project Icebreaker" at LinuxWorld in August 2006. Over a dozen customers tested the release with very promising results. In particular, customers experienced faster and simpler installations (database and operating system are installed in one step) and ease of management. In fact, Icebreaker reduced the installation time for both the operating system and database by as much as 4.5 minutes.

In late February 2007, we formally launched Icebreaker, the first fully integrated enterprise-class database and operating system. There are a number of customers taking advantage of this solution and Ingres is now expanding the portfolio of appliances to include an Icebreaker Business Intelligence (BI) Appliance. The Icebreaker BI Appliance is an appliance that consists of JasperSoft, Ingres and rPath Linux. Plans are underway to add a salesforce.com variation as well as a content management appliance to the portfolio later in 2007.

Although software appliances are not for everyone, we firmly believe that our new software appliance has opened up new opportunities for Ingres. We are able to reach smaller organizations that need a robust relational database to power their mission-critical applications, but do not have the capacity in their IT shop to implement such a complex solution.

rPath is leading the industry in tools for creating, configuring and maintaining software and virtual appliances. Our partnership with rPath ensures that we are providing our customers with high quality products that reflect our commitment to their success.

*The open source model you want...  
the enterprise strength you need*

**INGRES®**

Ingres Corporation  
500 Arguello Street, Suite 200  
Redwood City, CA 94063  
Tel: (650) 587-5500  
Email: info@ingres.com  
www.ingres.com