WHITE PAPER

JAVA CONTENT REPOSITORY

A universally-accessible repository for content management systems

SUMMARY

The Java Content Repository (JCR) API is a specification which provides a common and standard way to access content repositories. Content repositories are used in Content Management Systems (other typical applications include Document Management and Records Management Systems) and many of these applications ship with their own 'content repository'. A content repository is a generic data store. It handles and stores structured and unstructured content, binary objects and text formats (for example: images, Word documents, PDFs etc.) with associated metadata.

This paper explains in greater depth the reasons and advantages of improving content repository interoperability and the ability to meet the demands of business owners to have their business-critical content independent of their applications' infrastructure. Advantages include the cost considerations from, for example, only maintaining one repository which means lower support, administration and other on-going costs.

CONTENTS

1. INTRODUCTION

The Java Content Repository (JCR) Application Programming Interface (API) is a specification which provides a common and standard way to access content repositories. Web applications have to adapt to each proprietary API that comes with a Content Management System to interact with content repositories. This impacts businesses in a number of ways, not least by locking information assets in vendor defined formats which can negatively influence the ability for a system to evolve or migrate.

One key feature of a content repository is that it does not matter how the data is actually stored: data can be stored in a Relational Database Management System or a file system or as an XML document. In addition to storing and retrieving data it also provides support for other content services such as versioning of data, uniform access control, locking, transactions, observation of changes in data, import or export of data to XML and search.

Sometimes a content repository is embedded within the same server as an application; otherwise it may be based on separate servers due to availability and load balancing.

According to some estimates[1] there are now more than 800 proprietary content repositories in the marketplace. As a result, the need for a common, standardized API for content repositories has grown so that customers are not tied to any specific vendor with the extra costs that are required to access and administer them. In fact, a common API is a source of considerable savings with respect to maintenance, support and other on-going costs.

In the next chapter, the White Paper will discuss the concept of JCR and explain the features of a unified API.

[1] See http://ciurana.eu/site.php?page=musings&contentTag=TSS-video.html#JCR

2. THE CONCEPT OF JCR

The Content Repository for Java Technology specification, developed under the Java Community Process as JSR-170 (currently being enhanced by JSR-283) separates infrastructural services from application services. This unified API allows access to any compliant repository in a vendor- or implementation-neutral manner.

The JCR API is a type of Object Database that stores, searches and retrieves hierarchical data and allows true content repository infrastructure so that different applications can use the same interface for many purposes, making it universally accessible.

The API extends a storage layer with various content service facilities such as author-based versioning, full textual searching, fine grained access control, content categorization and content event monitoring. These content services characterize the 'content repository' and applications know how to interact with them since they are formally defined. The API proposes that content repositories have a dedicated and standard fashion in the way they interact with content applications.

But API standardization is not the only feature of the Java Content Repository (JCR). A major advantage of JSR-170 is that it is not tied to any specific architecture. The back-end data storage for a JSR-170 implementation could be a file system, a WebDAV repository, an XML-backed system, or even an SQL database. It also provides a straightforward interface that can be layered on top of a wide variety of existing content repositories, yet at the same time standardize complex functionality such as versioning, access control, and searching.

In the next chapter, the White Paper will look at the advantages of the JCR API and how it will benefit vendors, consumers and developers.

3. ADVANTAGES OF JCR API

Depending on your role, there are many advantages that the JCR API can afford you. Interface standards benefit (Java) application developers, software project managers, vendors, third party system integrators and not least, business and IT departments.

Application developers and custom solution integrators will be able to avoid the costs associated with learning the particular API of each repository vendor. Instead, programmers will be able to develop content-based application logic independently of the underlying repository architecture or physical storage. Using a common interface reduces both times and risk, in so much as a company will no longer need to rely on any one proprietary repository.

Businesses will benefit by being able to exchange their underlying repositories without touching any of the applications built on top of them. They will be able to make comparisons of applications without the complication of having to compare repositories as well. And they can focus any future Content Management strategy on just one standard.

There will be no vendor lock-in or associated maintenance, support and other on-going costs from having more than one content repository. Overall there will be a total cost of ownership (TCO) saving as the JCR is not limited to a one time use. Over time, content repositories can be consolidated to become true infrastructure. Reading and writing from and to the repository will only be limited by access control and ultimately possible from any environment. Business users will be able to buy the content management application most suited to their needs, and even be able to mix content applications on the same repository whilst the IT department will be happy to implement a standardized content repository.

Vendors too will benefit by being able to concentrate on their core skills and competencies of developing applications, GUIs etc. The common JCR means that applications can be created at a faster rate and more efficiently. Ultimately development costs behind a CMS will fall, allowing vendors to develop and mature their solutions.

In the next chapter we look in depth at how GX embraces the concept of a separate JCR.

4. VISION OF GX

GX embraces the concept of a separate and generic JCR as its vision is to create technology that is based on open standards. GX is implementing it across all the software it develops.

GX has always been focusing on open source and open standards as a primary means to enrich and enhance its web content management platform. Openness and preventing vendor lock-in has been the goal of GX from the beginning. These two factors give customers greater choice and greater flexibility, and leaves GX to do what it does best, applying modern technology and business insights to the job of developing and delivering products and solutions which provide real benefits for the customers that GX and its partners serve.

The next chapter explains how and why GX has implemented Apache JackRabbit as its default JCR Implementation.

5. IMPLEMENTATION IN WEBMANAGER

GX WebManager 9 uses the open source Apache JackRabbit implementation of the JSR 170/283 standard. Apache JackRabbit is a fully conforming implementation of the Content Repository for Java Technology API (JCR).

The choice for Jackrabbit as a default JCR implementation was simple: it is the reference implementation, and has ended being the de facto standard. This has now been successfully proven by the wide number of large customers using GX WebManager 9.

To further support and enhance Jackrabbit, GX also invests in the software, both financially and in code, by contributing patches to Jackrabbit and contributing on the various mailing lists. As GX and GX partners implement Jackrabbit in various large-scale Web sites, our knowledge of how Jackrabbit behaves in high traffic professional Web sites is now proven which is worth a lot to the community and how it subsequently matures.

GX also is actively involved in the standardization effort itself. Martijn van Berkum CTO of GX participates as an expert group member in the JSR-283 standardization group.

6. CONCLUSION

As we have illustrated in this paper, by implementing a common content repository, customers are vendor independent and see considerable savings with respect to maintenance, support and other on-going costs. There are many benefits for vendors as well and today JSR-170 and JSR-283 are finding usage in many applications. The software industry is supportive in establishing their use and importance across the enterprise. Already JCR API has begun to significantly change the way content-centric applications are developed.

7. ABOUT GX

GX is a successful web technology specialist and the largest independent supplier of web content management solutions in the Netherlands.
GX's perceptive view of the market and clear strategy strongly differentiate the company from its competitors.

Vision

The Internet is a mature communications medium that now also offers attractive commercial opportunities. The second Internet revolution, or more accurately evolution, has already started. Less conspicuous than the first, but with far-reaching consequences. The behavior and expectations of consumers who currently use the on-line channel have changed fundamentally. Moreover, today's possibilities encourage and reward the new consumer's changed behavior and expectations. The generation that will flow onto the labor market in a number of years sees the Internet as a channel without limitations and the difference between the traditional consumer and the on-line consumer will soon disappear.

Existing software systems that support business processes and handle transactions are based on internal processes. The gap between consumer expectations and existing business software is becoming increasingly larger and will be more and more difficult to bridge in the future. Software suppliers will need to approach business processes from the opposite direction in order to satisfactorily service the new digital consumer, in other words from the outside in.

Mission

GX's objective is to support organisations and businesses that focus on the modern digital consumer and improve their competitive position by developing products and solutions that support the business processes involved in an appropriate way. Regardless of what our customer's customers wants to do on-line - buy goods, provide information, lodge a complaint, play games, access entertainment, submit an enquiry, participate in on-line activities, etc. - GX, as the leading supplier in the Netherlands, aims to provide the best systems and solutions.

Strategy

GX's strategy is based on the principle of 'Outside in'. An increasing share of business processes is initiated from outside the organisation via the on-line channel.
GX WebManager was specially developed to support processes of this nature. A robust and flexible platform that acts as a stable foundation for the many solution frameworks offered by GX and GX's implementation partners for specific market segments or applications. This approach ensures maximum flexibility and effectiveness in adapting to the continuous stream of new developments in this field and making them available to customers. A natural consequence is that the functionality of GX WebManager will infiltrate ever further into the organisation, but always starting from the outside. Outside in.

8. FURTHER READING

Jackrabbit Web site
– http://jackrabbit.apache.org/

JSR-170
– http://jcp.org/aboutJava/communityprocess/final/jsr170/index.html
– http://jcp.org/en/jsr/detail?id=170

JSR-283
– http://jcp.org/aboutJava/communityprocess/pr/jsr283/
– http://jcp.org/en/jsr/detail?id=283

Tips for modeling your content model
– http://wiki.apache.org/jackrabbit/DavidsModel

Contact

For more information regarding JCR
or our other products and services you can
contact us:

GX
Wijchenseweg 111
6538 SW Nijmegen
The Netherlands
T +31 (0) 24 - 388 82 61
E info@gxwebmanager.com
www.gxwebmanager.com