



# Integrating IBM i Applications with Web Services



## Contents

---

Integration and the IBM i - An Introduction .....	4
Service-Oriented Architecture (SOA) .....	4
Web Services .....	5
looksoftware support for Integration, SOA & Web Services.....	6
Summary.....	7

looksoftware is a registered trademark of looksoftware Pty Ltd and looksoftware Inc. Other company, product and service names mentioned herein may be the trademarks or service marks of their respective owners.

The information contained in this document represents the current view of looksoftware on the issues discussed as at the date of publication. Because looksoftware must respond to changing market conditions, it should not be interpreted to be a commitment on the part of looksoftware, and looksoftware cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for informational purposes only. looksoftware makes no warranties, express or implied in this document.

## Abstract

Integration with Web Services offers significant business benefits including lower costs that are aligned with usage, automation, real-time access, reduced time to value and reduced risk.

This whitepaper provides an overview of integration, why it is important and how it can be leveraged with the IBM i to automate data exchange with other applications and platforms, providing increased efficiencies and data integrity.

## Integration And The IBM i – An Introduction

We live in an interdependent world. Long gone are the days where an IBM i is the sole platform running in a business. Almost all users are accessing these systems from a PC, and often because of this there are Windows servers in place. Applications running on Windows servers, and the requirement to exchange data with customers, business partners, vendors and other 3<sup>rd</sup> parties makes integration an essential part of modern IT requirements. Also, with the increased adoption of mobile devices like smartphones & tablets, along with the Cloud ecosystem, integration is more important than ever.

Any organization that has a significant investment in existing applications and needs easier integration should be interested in service-enabling their applications. More often than not, these applications have evolved with years of refinement and provide robust support for the organization's business processes – in many cases reusing these applications makes the most business sense.

Composite Applications are about delivering new solutions by snapping together components, in particular components that are reused from existing systems. Reuse is the common theme of Composite Applications and SOA. If we are able to access our existing applications as useful chunks of function, termed 'services', then we have a means of accessing even monolithic applications in a more modular way. That makes the delivery of Composite Applications feasible – because now you can assemble/access the services, perhaps add some business rules and deliver the 'new' solution.

## Service-Oriented Architecture (SOA)

Service Oriented Architecture, or SOA, is a set of architectural and design principles that are probably new to most organizations with IBM i and z back-end applications. SOA considers an enterprise as consisting of many processes and services. SOA comes with the expected set of acronyms and jargon, such as "loose coupling" and "granularity" and is typically implemented with Web Services. By reusing chunks of application functionality across platforms, the vision is we can begin to refine, streamline, and perhaps even create new business processes - on the fly.

SOA is already delivering on its promise to deliver increased business agility by enabling easier integration. Examples of web services integration from our customer base include integration with front-ends like Office, other back-end applications like SAP, other systems like Interactive Voice Response and development environments like Visual Studio.

A simple way of understanding Web Services (and avoiding technical discussion of WSDL, SOAP, and UDDI) is to think of them as a standard way of calling a program that may be located on another platform. By using tools that:

1. Support the creation of component-level access for your existing applications, without requiring changes, and
2. Wrap the components as web services (generating the SOAP and WSDL etc.) then, any-to-any integration can be achieved much more painlessly than in the past.

The key benefit of SOA is simplifying application integration. Given that the underlying web service standards like SOAP and WSDL have been universally accepted, the continued growth of the service-based approach is guaranteed.

## Web Services

Let's take a look at a practical example of extending an RPG 5250 System i application to consume a Web Service. The customer maintenance function includes the customer's address details. Addresses are often very important and useful data!

For example, if you're in the logistics business, delivery failures impact productivity and therefore, profitability. If addresses could be validated as they enter the system against a postal database, then most delivery failures could be eliminated, improving customer service levels – and profits! Web Services provide a relatively simple mechanism of integrating the logistics application with the external Web Service module that checks the supplied address against the postal authority's address database. We'll assume the Web Service expects to receive an address as input and returns a valid/not valid flag. The simple, non-invasive way to extend the existing application is to do the following:

1. Intercept the address as it is entered on the customer address details screen.
2. Call (consume) the Web Service and pass the address entered to return a valid/not valid flag.
3. If not valid, display the error dialog and the steps required to correct the problem. If valid, allow the unchanged host program to continue updating the database with verified address information.

The result is that only correct addresses reach your DB2 database, reducing the cost of delivery failures and increasing profits.

## looksoftware support for Integration, SOA & Web Services

looksoftware provides an integrated suite of tools to allow for easy, powerful and secure integration. At the core of this support is soarchitect and it's Transaction Recorder.

soarchitect is designed specifically for IBM customers that need easy-to-learn tools to service-enable their current application environments. Typically, they don't want to pull apart their systems or redevelop them. soarchitect supports the most popular IBM technologies like 5250, RPG, COBOL and DB2.

One of the advantages of using soarchitect to expose services from existing code, rather than developing a new services-based application, is reduced cost and risk and faster time to value. By reusing existing application code, the risks associated with redevelopment are eliminated. In particular the challenge of "right-sizing" the services (the technical term 'granularity' refers to service scope), is not so difficult when working with existing code.

looksoftware's Transaction Recorder can define and redefine service granularity in minutes, allowing the project focus to be on exposing just the key services delivering a specific solution, rather than attempting to architect a complete services-based system. soarchitect reduces the risk, costs and timeframes associated with implementing service based projects, helping ensure the success of your first web services project.

## Summary

You can do this! The increased complexities of the environments we are required to support are offset by universal support of standards like XML, HTTP, HTML and JavaScript that simplify the process. The ubiquity of the Cloud and Internet infrastructure makes machine-to-machine and application-to-application integration far more accessible.



---

Suite 9 / 622 Ferntree Gully Rd  
Wheelers Hill  
VIC 3150  
Australia  
ACN : 071 351 776

Tel: +61 (0)3 9535 4444  
Fax: +61 (0)3 9535 4455  
[info@looksoftware.com](mailto:info@looksoftware.com)  
[www.looksoftware.com](http://www.looksoftware.com)