



Q-Pulse Extensibility Overview

Contents

Introduction	3
Background	3
Objectives	3
Current Capabilities	4
Data Import and Export	4
Web Service APIs	5
Future Possibilities	10
Alternative forms of Integration.....	10
Extension Type APIs	10
Glossary of Terms	13

Introduction

This document is intended to provide key information about the current and future integration capabilities of Q-Pulse. Its intended audience includes but is not limited to Q-Pulse pre-sales and technical customer contacts.

Background

Database applications have generally been defined as “Islands” due to the complexities in sharing information and combining processes between them. As businesses strive to improve the efficiency and effectiveness of their operations, combining systems and processes becomes a key requirement in doing so.

With users needing to use more and more systems in their day to day tasks, they are required to learn a number of specific user interfaces, sometimes to be used only a few times over an extended period of time. Even though Q-Pulse follows many Microsoft application guidelines and is certified for use on Windows Vista, 7 and Windows Server 2008, the nature of its general purpose user interface may require a specific level of understanding or product knowledge to perform tasks.

Objectives

When reviewing the integration options available to Q-Pulse we considered a number of alternatives. Through discussions with a number of existing and potential customers it was clear that they needed to achieve two main goals. Firstly, they wanted to increase the reach and flexibility of data capture within Q-Pulse so that they could gain additional value from the data. Secondly, they wanted to make key Q-Pulse information available to users within other user interfaces.

As such it was felt that web services would provide the most flexible solution to meet these and many other customer needs.

Web services make available key Q-Pulse functionalities that allow programmers to integrate Q-Pulse data and operations into any compatible platform. These platforms include web portals, intranet systems and the Microsoft Office system. By doing this they can allow users to remain in the context or environment that they are comfortable using without the need to use Q-Pulse to complete a task.

This in turn reduces the deployment and training requirements for infrequent users while increasing the volume and value of information held in Q-Pulse.

Current Capabilities

Although Q-Pulse provides an integrated solution within its comprehensive module offerings, users may wish Q-Pulse to integrate with wider business processes or to be integrated with other systems. Integration APIs allow Q-Pulse to be “driven” by an external application or process whilst still maintaining applied business logic and security. This allows Q-Pulse to play a part in business processes that take place outside the scope of Q-Pulse.

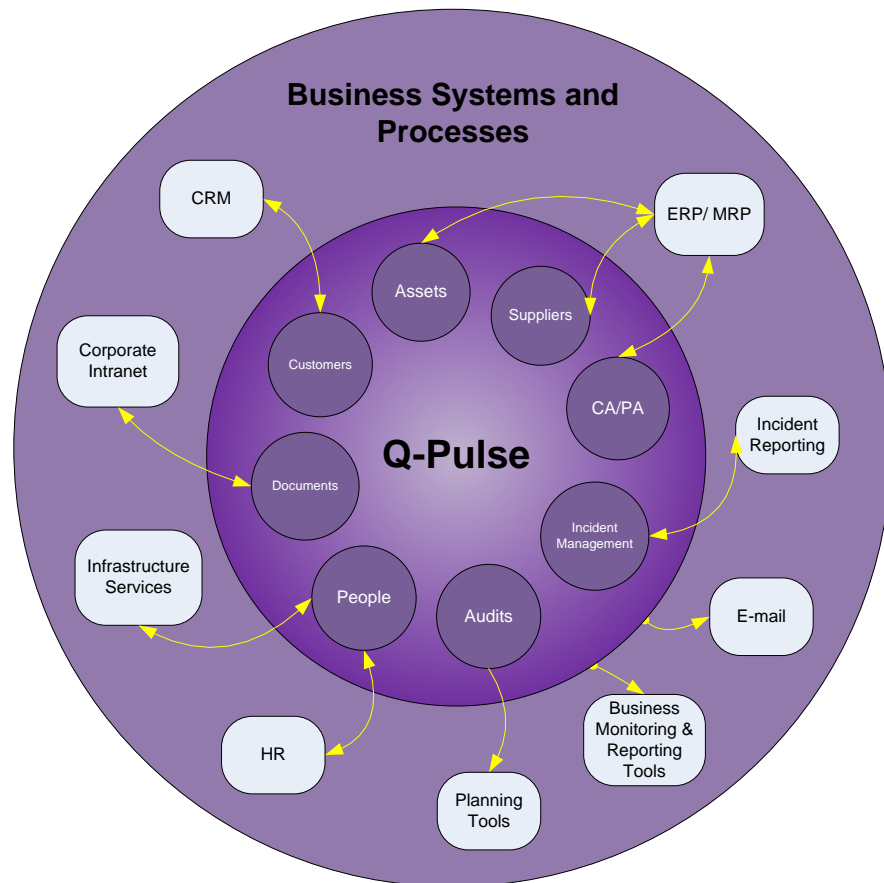
These capabilities are made available via two main features, ‘Data Import and Export’ and ‘Web Service APIs’.

Data Import and Export

Data can be imported to and exported from Q-Pulse in a number of ways. For example, any generated report can be exported into a number of file formats including Excel, CSV and PDF. Audit Checklists can be imported and exported to allowing sharing of checklist templates (via the Audit Checklist Exchange on Gael Community). Incident information can be exported in XML format to any nominated schema definition suitable for consumption by external sources such as industry regulators or third party systems. Analysis graphs can be exported to various graphic formats for inclusion in presentations and reports. Calendar entries in Audit and Training can be made available using the iCalendar standard to allow easy integration with PIMs such as Outlook.

Web Service APIs

Web services APIs expose key Q-Pulse functionality in the form of XML based web services, providing a mechanism for integrating Q-Pulse functionality into third-party applications.



Q-Pulse Web Services are:

- **Industry standards-based**

Web services provide a standard means of interoperating between different software applications, running on a variety of platforms and/or frameworks. Web services are characterized by their great interoperability and extensibility, as well as their machine-processable descriptions thanks to the use of XML.

- **Cross Platform**

Messages are sent using the standard formats and protocols used to build the World Wide Web, so provide great opportunities to build applications on any platform that supports these standards from Windows Desktop and enterprise servers to mobile phones.

- **Flexible**

Q-Pulse Web Services supports the popular SOAP, REST and JSON* programming styles. Your choice of which style to use is entirely up to you.

**Not available in the Occurrence Services*

Current Scope

Q-Pulse Web Services are currently limited to providing creation and reading of data within the following modules.

Occurrences

Retrieval of Occurrence form information in order to build custom input forms.

Creation of new Occurrences

Search and retrieval of existing Occurrences

Incidents

Search and retrieval of existing Incidents

CA/PA

Creation of new Non-conformances

Search and retrieval of existing Non-conformances

Audits

Creation of new Audits

Search and retrieval of existing Audits

Documents

Creation of new Document records with associated file attachments

Search and retrieval of existing Document records and files

Retrieval and Acknowledgement of outstanding document distribution acknowledgements.

Examples

Some examples of where Web Services could be used are:

1. To automatically place a batch of materials within an ERP on hold as soon as a non-conformance was raised in Q-Pulse.
2. When operational data such as trends within Occurrence report data exceed a specific acceptable operational range, a non-conformance could be raised to commence an investigation as to why this might have happened.
3. Submit a Document into revision control directly from Microsoft Word™.
4. Gather information from users using a specific input form on a business intranet.

A number of tutorial based examples are available as part of the API Extension Pack.

API Extension Pack

The Q-Pulse Web Service API extensions comprise the following items:

Web Services Installer

To install the Web Services to a nominated server with IIS.

Q-Pulse API Overview

This Document.

Q-Pulse API Developer Guide

Technical documentation for using the services.

“Live” Documentation

Installed alongside the Services enabling direct access to WSDL/Schema files, Operation and Data Contracts used by the Q-Pulse Services.

3 Example Integrations utilising the Occurrence Web Services

A Microsoft Outlook 2007 Plug-in

A Microsoft BizTalk Orchestration

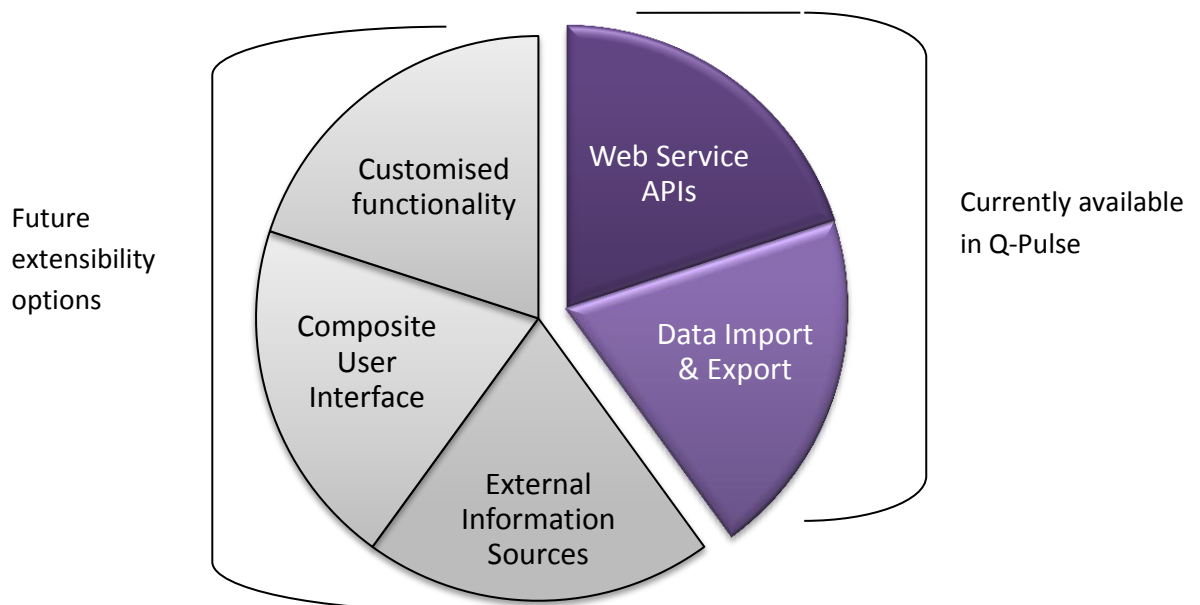
A Microsoft SharePoint “Web Part”

Future Possibilities

Alternative forms of Integration

Q-Pulse plays a key role in many business processes and as such can often benefit from integration with other key business systems such as ERP, CRM, etc.

The architecture of Q-Pulse lends itself to being extended or interacted with in a number of ways and as such allows a flexible way of contributing to wider business processes. Given its distributed nature, these extensions can be implemented in various tiers of the application to achieve different goals.



Extension Type APIs

Extension type APIs allow users to programmatically extend the functionality of the application to perform specific functionality based on their needs. These have been broken down into 3 areas.

Composite User Interface

Definition: Composite UI APIs allow for the creation of a common façade to a number of applications providing a single consistent access interface to these applications and shielding users from having to learn to interact with different software packages.

In the context of Q-Pulse this would enable developer to create specific UI to be incorporated into the standard Q-Pulse user interface to provide custom functionality or contextual information from 3rd party systems.

Examples:

1. For an aviation customer, displaying associated flight data information relating to an Incident would provide the user with the knowledge that further information existed and could provide them with quick access to either view this information or perform an action.
2. When viewing a Non-conformance relating to a Part in an ERP system, it would be beneficial to see live extended information about that Part without needing to navigate to the ERP and locate it.

External Information Sources / Data Integration

Ensuring that information in multiple systems is kept consistent. This is also commonly known as EII (Enterprise Information Integration).

Providing access to external information sources can be useful in reducing the amount of data duplication required in a business and as a result reduces the likelihood of incorrect or inaccurate information being used.

Examples:

1. Use Microsoft Active Directory as both the master repository for people as well as the place where Q-pulse permissions could be controlled.
2. When raising a non-conformance allow users to select a Part item from live data in an ERP system.

Extended functionality

It is sometimes useful to extend the core functionality of the existing Q-Pulse client applications in order to meet a specific business requirement. This could involve implementing custom business rule processing or reacting to specific events that take place either in the client UI or on the business objects.

Examples:

1. Validate the state of a business object data field against an external system at time of save and prevent the data from being saved if it were not suitable. E.g. does the person defined really belong to the department who is responsible for actions?
2. Validate two dependant fields based on a value entered, e.g. if a value is entered into one field then another associated field becomes mandatory and requires a value to be specified.
3. Depending on the nature of the event, the processing could take place either directly in the client application or for all users and client types (including web services) at the server.

Examples of server based events would be to either update an external database whenever a record was created in Q-Pulse or to provide additional notification options such as SMS when notifications were being created.

Glossary of Terms¹

API

Application programming interface (API) is an interface in computer science that defines the ways by which an application program may request services from libraries and/or operating systems. An API determines the vocabulary and calling conventions the programmer should employ to use the services. It may include specifications for routines, data structures, object classes and protocols used to communicate between the requesting software and the library.

Integration

System integration is the bringing together of the component subsystems into one system and ensuring that the subsystems function together as a system. In information technology, systems integration is the process of linking together different computing systems and software applications physically or functionally.

Extensibility

In software engineering, extensibility (sometimes confused with forward compatibility) is a system design principle where the implementation takes into consideration future growth. It is a systemic measure of the ability to extend a system and the level of effort required to implement the extension. Extensions can be through the addition of new functionality or through modification of existing functionality.

ERP

Enterprise resource planning (ERP) is a company-wide computer software system used to manage and coordinate all the resources, information, and functions of a business from shared data stores.

CRM

Customer relationship management (CRM) consists of the processes a company uses to track and organize its contacts with its current and prospective customers. CRM software is used to support these processes; information about customers and customer interactions can be entered, stored and accessed by employees in different company departments.

Web Service

¹ Glossary terms taken from Wikipedia

A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards. (<http://www.w3.org/TR/ws-gloss/>)

REST

Representational state transfer (REST) is a style of software architecture for distributed hypermedia systems such as the World Wide Web. The term Representational State Transfer (REST) was introduced and defined in 2000 by the doctoral dissertation of Roy Fielding, one of the principal authors of the Hypertext Transfer Protocol (HTTP) specification versions 1.0 and 1.1.

SOAP

SOAP, originally defined as Simple Object Access Protocol, is a protocol specification for exchanging structured information in the implementation of Web Services in computer networks. It relies on Extensible Markup Language (XML) as its message format.

XML

XML (Extensible Markup Language) is a set of rules for encoding documents electronically. It is defined in the XML 1.0 Specification produced by the W3C and several other related specifications; all are fee-free open standards.

JSON

JSON (JavaScript Object Notation), is a lightweight text-based open standard designed for human-readable data interchange



Gael Ltd.

Orion House,
S. E. Technology Park,
East Kilbride,
Scotland. G75 0RD

t: +44 1355 593400

f: +44 1355 579191

e: info@gaelquality.com

w: www.gaelquality.com

Q-Pulse is a registered trademark of Gael Products Ltd. All rights reserved worldwide.
Copyright © 2010 Gael Products Ltd. Gael Quality is a trading division of Gael Ltd.