



TECHNOLOGY AUDIT

# Progress Actional 8.2

## Progress Software

### SUMMARY

### CATALYST

Progress Actional combines transaction tracking, formulation and enforcement of security or service level agreement (SLA) policies, and web services testing tools aimed at distributed computing environments, both SOA and non-SOA based. Actional is relevant on account of the following industry challenges:

- Comprehensive transaction-oriented view of distributed environments is necessary for distributed applications that have a direct impact on productivity or revenue generation.
- The ability to automatically discover services and transaction paths is important because distributed systems are typically complex and lack central control by design.
- The SOA environment requires specialized web services testing tools.

### KEY FINDINGS

<b>Strengths:</b>	<ul style="list-style-type: none"><li>✓ Offers transaction tracking against SLAs</li><li>✓ Automatically generates a detailed topology of the application environment</li><li>✓ Offers web services modeling and web services testing tools</li><li>✓ Support for in-depth root cause analysis</li></ul>
<b>Weaknesses:</b>	<ul style="list-style-type: none"><li>✗ Actional's agent architecture does not easily plug into homegrown applications written in languages such as C++</li></ul>
<b>Key Facts:</b>	<ul style="list-style-type: none"><li>i Underpinned by Progress Software's understanding of many different systems and protocols</li></ul>



### OVUM VIEW

Highly distributed computing environments would typically include message-oriented middleware, an enterprise service bus (ESB), and integration technologies such as common object request broker architecture (CORBA). Such environments would frequently have web services and be based on service-oriented architecture (SOA). To facilitate management of such complex systems Progress provides Actional 8.0, a suite comprising three modules, each aimed at enabling streamlined management of highly distributed environments:

- Transaction tracking
- Web services testing
- Policy formulation and management.

Transaction tracking involves monitoring the flow of a transaction across nodes in a distributed environment, with the ability to zero in on the point of failure or the point of service degradation. Progress Software's self-professed differentiators in the transaction tracking market include low performance overhead, automated discovery of services, automated discovery of the distributed application infrastructure layers as well as the dependencies across the system components, and the ability to monitor service levels based on transaction parameters (such as 'high-value' transactions or transactions from a particular channel partner). We shall examine each of these in depth later. In summary, Ovum believes all the points of differentiation emphasized by the vendor are valid, as those capabilities would not be easy for Progress' competitors to replicate.

Testing capability allows developers and QA personnel to test web services behavior and performance. The important point about the testing suite is that the tool allows developers to model web services easily and without comprehensive knowledge of XML. This enables testing early in the development cycle. Testing web services response to input, load testing, and regression testing are all part of the suite. The testing tools also provide a good set of collaboration features. These enable pre-production and post-production teams to collaborate in the event of service outage or service degradation.

The third set of capabilities involves policy authoring, policy compliance monitoring, and policy enforcement. Policies can be about transactional SLAs or about security. Policies about SLAs are typically about setting thresholds and reporting in the case of violations. Policies about security are about enforcement and mediation. For example, if a service is not authorized to route messages to an end point, Progress Actional can stop a message if had not been routed through a policy enforcement point (PEP). It is important to mention that policy authoring and enforcement are both for design time and run time, and developers, architects, and business analysts can check and validate service contracts and messages against industry standards such as WS-Security and WS-Policy and organization-specific policies. Actional works with leading registries and repositories as well, deriving information for security policy enforcement (such as discovering 'rogue' services or those not registered), and adding log information to these repositories. Policy authoring capabilities are not limited to just policy authoring alone, but also enhancing the information content of service artifacts. For example, services contracts (typically Web Service Definition Language [WSDL] files) can be annotated through Actional's tools.

Overall, Progress Actional renders distributed computing environments more manageable. Such environments are often designed for high-performance environments where the cost of failure is high and the possible failure points numerous. Most enterprises have a number of point application management and system management products that report system-specific outage alerts. The view is not transaction-centric. Typically, after an outage (or service degradation) reported by a single management system the administrator would begin the process of root cause analysis through information from different management systems and by extracting information from application logs.

There are a number of problems with this approach. The first problem is that the system as a whole is seldom understood with a high degree of clarity by anybody in the organization. Therefore it is hard to understand how components relate to each other. Also, on account of the inherent difficulty in gathering and correlating information across a number of logs and management systems, the process of root cause analysis becomes tedious and the mean time-to-repair is often unacceptably high. Furthermore, the process typically begins with an outage report generated by one node in the distributed system and not triggered by degradation in transaction performance, as it ought to be. Progress Actional makes all of the above possible by reporting transaction statistics, generating alerts based on transaction-related SLAs, and by allowing the administration view of statistics and logs associated with every node and service involved in the transaction path.

The services approach to distributed systems is inherently complex, with limited centralized control of how services would be used. Also, the entire process is still new for a sizeable portion of the developer community. The ability to model services, test services early, document services artifacts well and collaborate across application management and development teams spanning the development lifecycle are all essential tasks in the contemporary distributed services-based environments. In addition, trouble shooting is aided by all the information captured at design time, and the log data captured at the time of failure, as well as the tooling that makes collaboratively analysis of this data possible.

A number of Progress Software products are potentially relevant to anybody evaluating Actional. For example, the complex event processing (CEP) product Apama: Actional and Apama work together, with Actional's FlowMapping (transaction tracking) technology generating events and Apama construing insights and action items out of the Actional-generated events. Just as importantly, Ovum believes that many organizations using SOA would benefit from Actional's integration and Progress Software's partnership with a wider range of SOA governance tools.

### **Recommendations**

- Scenario 1 – Distributed, revenue-generating applications: Progress' Actional merits closer evaluation by any enterprise with an application that spans many constituent applications in an n-tier or a SOA system. Ovum believes that Actional's ability to track transactions from end to end and its ability offer a transaction-centric view of performance would be relevant to any organization that has a distributed application with direct linkages with revenues.
- Scenario 2 – SOA scaling initiatives: Ovum believes that the services testing capability and the SOA security enforcement capabilities of Actional would be relevant to companies that are aiming to scale up their SOA projects.

- Scenario 3 – Transactional tracking in non-SOA distributed environments: It is important to mention that Actional's transaction tracking capabilities are not meant only for environments that have web services or those that have the stated goal of being aligned with SOA principles. The solution is relevant to tracking transactions in environments dominated by CORBA, for example. However, if an application does not span a sizeable number of underlying systems and databases, Actional might be overkill.

## FUNCTIONALITY

### SOLUTION OVERVIEW

Acquisitions have always been a part of the Progress strategy. The technologies that are now routed to market under the Actional brand were brought into the Progress fold through M&A activity. Progress acquired Actional in January 2006, which brought two key solutions into the Progress portfolio: Looking Glass (enabling business process monitoring in SOA environments) and SOAPstation (for policy formulation, monitoring, and enforcement). In 2008, Progress acquired SOA testing technology provider Mindreef. In 2009 Progress integrated Mindreef with Actional to release Actional Application Development, plus a free subset, Actional Diagnostics, which enables developers and QA personnel to test web services elements, such as SOAP and REST-based interactions. The rest of this section describes three major areas (Figure 1) within Actional: collaborative testing; transaction monitoring; and policy authoring, monitoring, and enforcement.

#### Collaborative testing

Actional provides workspaces that allow developers, architects, testers, and business analysts to work collaboratively on web services contracts, compliance rules, test cases, test results, and simulation runs. Testing tools allow functional, unit, regression, and load testing. It is important to mention in this context that testing capabilities include testing of web services security, which involves testing whether the authentication, authorization, credentials, and digital signature requirements governing messaging routing have been adhered to. Another set of tools allows developers to model services using WSDL files, interact with web services through an easy-to-use forms-based interface, provide an easy-to-read view of XML messages, and help navigate WSDL contracts and representational state transfer (REST) contracts, which increases the readability of these artifacts. Yet another set of tools enables design time validation of WSDL, SOAP, REST, and Plain Old XML (POX) against the WS-I standard to ensure that the services have been built in conformance with industry standards.

#### Transaction monitoring, management, and incident resolution

Actional works through a system of agents and a central server. Agents are deployed but they do not necessarily have to be situated on every node; they can detect activity remotely from one node (or hop) away. Every agent comprises an analyzer and one interceptor for every platform (such as an application server, e.g. BEA Weblogic) and protocol (such as JMS) combination on the node. After deploying the agents on nodes (as required), the agents are registered with the central servers. SLA policies are formulated at the central server and SLAs are set up to act on specific metadata, such as transaction value or attribute such as whether messages have a 'Social Security Number' field or a customer ID. The policy would then apply to every node that has a hosted service or application which has that metadata.



When policy violations occur, the violations are reported to the central server. SLAs are measured both in terms of metrics that represent a minimum acceptable value (such as message response time for any message should not be beyond a certain value) and in aggregate (average response time). When policies are violated Actional presents the path that the message traversed at the time of policy violation and also the node where the policy violation message was triggered. Root cause analysis is aided by a number of information points. The policy violation displays the other nodes where policies were violated. Also, the thickness of the lines connecting the nodes depicts the message response statistics. In addition the server collects logs for all the nodes involved to aid root cause analysis.

Actional facilitates troubleshooting, and maps the application environment topology by discovering services. Once agents are deployed and registered with the central server, every agent builds a map of requesting services (relative to the node on which the agent resides) and consuming services (again relative to the node on which the agent resides). This 'one hop' data captured by every agent, and the information from every agent aggregated at the central server, builds up the transaction path of every end-to-end process and the topology of the distributed environment. An important related point here is that statistics are collected for every service-consumer pair. Important metrics can therefore be collected for interactions with applications that the IT organization does not control, such as SaaS-delivered applications.

#### **Policy authoring, monitoring, and enforcement**

Policies authored, monitored, and enforced by Actional pertain to security and SLAs. The policies are enforced and monitored through a system of distributed agents and a management server, with much of the enforcement functions executed by the agents without interaction with the management server. Policies can apply to nodes, services, applications, business processes, and attributes such as transaction characteristics (for example, high-value transactions) and personally identifiable information.

The monitoring functions of agents include monitoring response time, status of service requests, requests, responses, faults, date, and timestamps. The enforcement functions of the agents include authorization and authentication, checking credentials, and digital signatures, encryption and decryption, message transformation and enrichment, and mediating protocols.

Policies are version controlled and an audit trail of policies is maintained. The design goal is to allow policies to evolve independently of services, which has an obvious impact on the agility with which services can evolve. This is achieved through the policies applying to metadata as opposed to policies working on specific services.

## **SOLUTION ANALYSIS**

### **Scalability**

There is a lot to suggest that Actional is used in some of the most demanding environments. According to Progress Software, Actional has been part of a deployment site with 3 million messages per hour. An Actional agent uses less than 2% (with every feature turned on) of the host system's CPU. Actional's transaction monitoring capability relies heavily on an understanding of how different systems log transactions and other relevant metadata.

Over the years Progress has developed a large number of interceptors covering the major vendors of application servers, messaging and transport protocols, databases, and others. By using this knowledge base, Actional reduces performance overhead by intercepting messages at predefined hook-points and avoiding unnecessary XML parsing requirements to decipher which service a specific message belongs to. Instead, the Actional agent can retrieve fields from the metadata in the XML-based requests (such as headers) or from the underlying application, rather than parsing the entire XML payload (it simply minimizes the parsing requirements by fetching XML elements until the necessary metadata is obtained). Actional's ability to display response time by attribute, such as customer region or product line or any other segmentation, is also based on this understanding. An Actional agent can also intercept, retrieve, and correlate runtime requests with logging information from the application logs themselves as they occur (as opposed to parsing the log messages by applications afterwards).

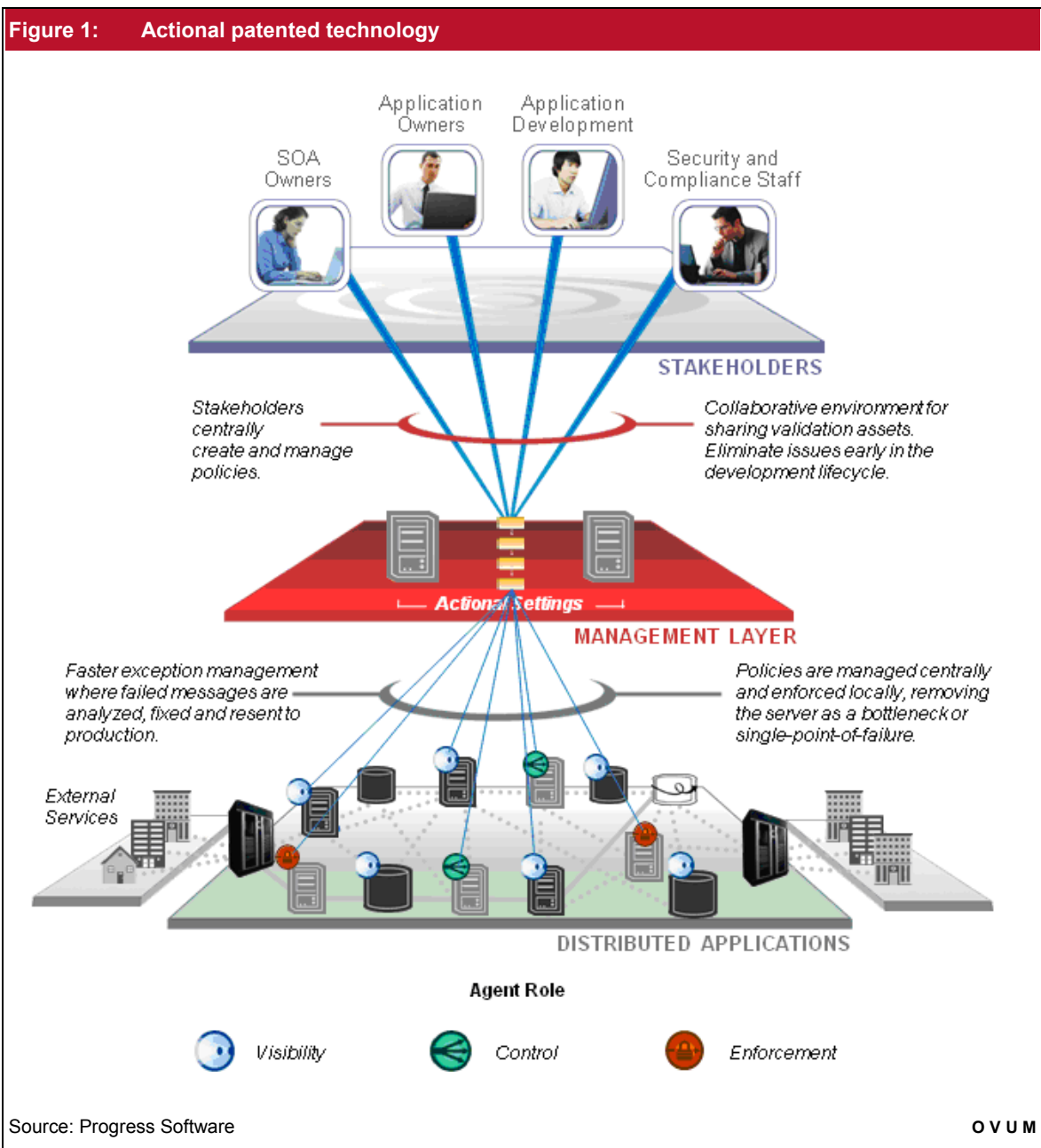
Another construct that enables Actional's scalability is the distributed nature of information gathering and correlation. To a large extent, the agents interact with each other to build the information required (as dictated by policy) without the need to report every time to the central server. Different information items are found with different applications hosted on different nodes and the information on what to retrieve can be added to message tag. For example, consider a scenario involving an order management application, a logistics application, and an inventory application, all connected to each other. To correlate a transaction across these three systems and collect the customer ID and product ID data items (which would enable performance tracking by customer type and product type), the customer ID (which is likely to be found in the order management system) can be added to the message tag (in Actional parlance, this is known as adding a 'Flow field'), and the product ID tags (linked to the customer ID in the tag) can then be retrieved from the inventory application. So the correlation and attribute collection can take place through interaction between agents without interacting with the central server.

Yet another construct that enables scalability is selective and policy-based logging. Storing logs for every transaction path has too high an overhead. Instead every agent has a local buffer, and only in the event of unexpected transaction behavior do the concerned points in the path report to the central server. The report server then polls the agents with the transaction tag to gather data. Therefore only interesting transactions, from the point of view of trouble shooting or preventive maintenance, are logged. Also, the system can be configured to gather log data only from major check points. In addition, policies can be formulated to persist application logs selectively, such as those linked to a transaction tag. This feature, introduced in the 8.0 version, naturally facilitates the trouble shooting process, in addition to enhancing scalability.

### **Comprehensive automated monitoring and SLA management**

One of the biggest challenges with highly distributed environments is the complexity which eclipses individual or collective ability to understand or document the system. Furthermore, different parts of the system are under the control of different organizations. A number of transaction tracking tools use the event correlation method, where the correlation conditions need to be manually configured. An example of such an approach would be an administrator specifying that the fields with similar names and same values could belong to the same transaction. Naturally, the manual configuration approach reaches its limits quickly.

Actional, by contrast, uses the concept of message tagging. Tracking the tags, Actional tracks transactions across the entire execution path. Actional's transaction tracking capability can be used in conjunction with application and systems performance management tools in a way that enables transaction-oriented troubleshooting. Any alerts generated by an Actional SLA violation would lead to the administrator browsing through the path the transaction traverses through, aided by statistics for each service provider-consumer pair and log information captured at every node.





Progress is able to make the tagging approach work on account of the company's comprehensive knowledge of how many different platform and applications store log data (described in the Scalability section). It is important to mention in this context that Actional uses the manual correlation approach in conjunction with the tagging approach. The correlation approach becomes particularly important for non-standard custom applications. Overall, Ovum believes that the Actional approach is among the most reliable, comprehensive, and automated way to ensure that all transactions would be monitored according to policy, and incident resolution would be based on the most relevant information possible.

**Related Progress Software products**

As with most applications addressing distributed environment requirements, Actional can only be one part of a broader architecture comprising multiple commercial and in-house developed solutions. Solutions from adjacent areas that work well in conjunction are therefore a natural part of the decision framework. Progress Software's Apama, a CEP product, is likely to add value to an Actional deployment. With Actional generating events, Apama can construe actionable inferences from the events. Apama and Actional are both a part of Progress Software's Responsive Process Management (RPM) portfolio. The portfolio also includes a business process management (BPM) system, via the Savvion acquisition. The RPM initiative launched in March 2010 is an important part of the Actional roadmap and the company expects that about 25% of Actional deals would be part of a bundle that includes other RPM solutions.

**PRODUCT STRATEGY**

Licensing is perpetual and priced per server and per managed node. Development and run-time environments are priced separately. For the largest accounts, Progress offers custom licensing models. The average entry-level deal is priced at \$100,000, with licensing accounting for 93% of deal value; the average-sized deal is \$350,000, with licensing accounting for 90% of deal size; the largest deals are in the region of \$1 million, with licensing accounting for 88% of deal value.

**MARKET OPPORTUNITY**

The Actional solution suite is aimed at Fortune 1000 enterprises in 12 countries: the US, Canada, the UK, France, Germany, Netherlands, Brazil, Mexico, Australia, Hong Kong, India, and Japan. Actional is a sector-agnostic solution but, as would be expected of a solution of this nature, the suite has been particularly relevant to companies in the financial services, telecoms, and travel, transportation, and logistics sectors.

**GO TO MARKET STRATEGY**

Direct sales and reseller partnerships are the two routes to market for Actional. The list of reseller partners includes Fujitsu, Layer7, and HP. For Actional, Progress Software also has three OEM partners, including Software AG. Software AG OEMs Actional and rebrands it as Software AG Insight. Actional partners with a subset of Progress's SI partners, including TCS, Accenture, HCL, VASS, and Momentum SI.

The two-year roadmap focuses on expanding the set of platforms that Actional can provide transaction visibility for, enhanced policy support for managing applications across geographically distributed business units, and enhanced support for monitoring dependencies across business units. The long-term roadmap includes adding remedial features. For example, a more remedy-capable Actional would detect a transaction incident, change the message payload, and resubmit the payload or change the prioritization settings.

Another focus area is integration with the other parts of the RPM portfolio. Integration is both in terms of product integration wherever applicable and a joint go-to-market with shared development and sales team. Progress Software states, and Ovum agrees, that there are a number of possible scenarios wherein solutions under the RPM umbrella can work well. Apart from the Apama and Actional scenario already described in this report, Savvion can provide the remediation workflows and Actional can add data about the transactions path that are not part of the model Savvion executes. Of course, Sonic ESB's considerable installed base would also benefit from Actional.

## IMPLEMENTATION

As would be expected of a solution of this nature, the implementation project can be fairly resource-intensive if the enterprise wishes to build a complete picture of the environment and when a set comprehensive policies needs to be built. The following is a representative list of typical projects and the corresponding resource requirements:

- Pilot project: Requires one FTE with systems administration skills and three to five implementation days.
- Departmental rollout: Typically has 30 users, requires one to one and a half FTEs and 20 implementation days. The skills required: administrative skills, Java, knowledge of the application, and knowledge of the application architecture.
- Enterprise-wide rollout: Typically serves up to 500 users, requires two to three FTEs and 40–60 implementation days. As expected from an enterprise-wide rollout, the project would require knowledge of the architecture and business processes.

Technical support comes in three options. The baseline option offers remote diagnosis and support through phone, email, and the Web, and access to updates and service packs. In this option, services are available between 8x5 and priced at 18% of licensing costs. The standard option offers support services 24x7, access to senior support engineers (on a maximum call-back time of one hour), time-based escalation of open issues, and a priority queue. The standard option is priced at 22% of licensing cost. The highest tier option is aimed at supporting mission-critical environments. This option includes all the features of the standard option with more stringent SLAs and a Technical Account Manager who develops and maintains expertise of the deployment site. Option 3 is priced at 25% of licensing cost plus \$25,000. The \$25,000 fee is waived if the annual support fee exceeds \$250,000.

Progress provides three types of professional service packages, JumpStart, Product Enablement, and Product Implementation. JumpStart is an entry-level program aimed at enterprises with some relevant experience and comprises training and mentoring on architecture, design, and solution pattern best practices. Product Enablement comprises everything the JumpStart scheme offers plus reviews, recommendations, and assistance throughout the development cycle. Project Implementation is a direct implementation service which involves the Progress professional services team executing the first integration project. Employees from the client organization are embedded in the implementation team for knowledge transfer. Progress provides training sessions for architects. The course lasts five days and can be both classroom and web-based.

Actional is available both on on-premises customer-managed and on-premises service-provider managed models. Also, a number of Progress' OEM partners provide SaaS offerings that are enabled by Actional. In terms of platform support, Actional supports Microsoft Windows (Windows 2000 to Windows Server 2008), Linux (Red Hat and SuSE), Solaris (Solaris 8 through 10), IBM AIX, and HP-UX.



The Actional installed base has 235 customers, not counting the Software AG Insight customers (about 25).

DEPLOYMENT EXAMPLES

An online trading brokerage lacked end-to-end transaction visibility and relied primarily on application performance management tools. A mature SOA adopter, the company had been operating a SOA system for nine years when Actional was deployed across the pre-production and management environments. Progress states that the Actional implementation led to a 92% reduction in the cost of resolving major production incidents and a 25% reduction in the number of major production incidents on account of transaction SLA-based management and the alerts generated before outage.

At a financial services firm, Actional was deployed as part of a SOA initiative, which was aimed at reducing siloed development and bringing products to market faster. The company also had an application infrastructure spanning many different systems. The first phase of the project involved implementing Actional primarily to gain visibility into the infrastructure. The next phase involved active monitoring of transaction execution and setting up policies across different business units, both at the business unit level and at the global level. Part of the implementation project's scope was to set up security policies with Actional, and SAML was implemented for security across domains.

Table 1: Contact Details	
<p><b>Progress Software Corporation</b>            14 Oak Park Drive            Bedford, MA 01730            USA            Tel: +1 781 280 4000  <a href="http://web.progress.com/en/actional/index.html">http://web.progress.com/en/actional/index.html</a></p>	<p><b>Progress Software UK</b>            210 Bath Road, Slough            Berkshire, SL1 3XE            UK            Tel: +44 01753 216300</p>
Source: Progress Software	OVUM

Ovum's Knowledge Centers are new premium services offering the entire suite of Ovum information in fully interactive formats. To find out more about Knowledge Centers and our research, contact us:

**Ovum Europe**  
119 Farringdon Road  
London, EC1R 3DA  
United Kingdom  
t: +44 (0)20 7551 9000  
f: +44 (0)20 7551 9090/1  
e: info@ovum.com

**Ovum Australia**  
Level 5, 459 Little Collins Street  
Melbourne 3000  
Australia  
t: +61 (0)3 9601 6700  
f: +61 (0)3 9670 8300  
e: info@ovum.com

**Ovum New York**  
245 Fifth Avenue, 4th Floor  
New York, NY 10016  
United States  
t: +1 212 652 5302  
f: +1 212 202 4684  
e: info@ovum.com

All Rights Reserved

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the publisher, Ovum Europe Limited. Whilst every care is taken to ensure the accuracy of the information contained in this material, the facts, estimates and opinions stated are based on information and sources which, while we believe them to be reliable, are not guaranteed. In particular, it should not be relied upon as the sole source of reference in relation to the subject matter. No liability can be accepted by Ovum Europe Limited, its directors or employees for any loss occasioned to any person or entity acting or failing to act as a result of anything contained in or omitted from the content of this material, or our conclusions as stated. The findings are Ovum's current opinions; they are subject to change without notice. Ovum has no obligation to update or amend the research or to let anyone know if our opinions change materially.

© Ovum. Unauthorised reproduction prohibited

This report is a licensed product and is not to be reproduced without prior permission.

