

# lookserver for Open Access FAQ



# Contents

What is IBM Rational's Open Access for RPG? .....	3
<b>look</b> software products already support modern devices and multiple channels. So what is different about ROA?.....	3
Why is ROA creating so much interest throughout the IBM i community?.....	4
What is an Open Access handler? .....	6
Can customers and application developers create their own handlers?.....	6
If a third party handler is typically required, what is open about Open Access?.....	6
What programming effort is required for an existing RPG program to benefit from Open Access? .....	7
How does Open Access make programming the UX easier? .....	7
Does <b>look</b> software advocate RPG as the recommended path for IBM i customers?.....	11
How is <b>look</b> software supporting Open Access? .....	11
What is unique about <b>look</b> software's support for Open access ? .....	12
What impact does ROA and these new <b>look</b> software announcements have for <b>look</b> software customers and ISVs? .....	14
IBM i community and customer support for Open Access?.....	15
What does Open Access offer that RPG Special files does not provide? .....	17
Where can I learn more? .....	18

## What is IBM Rational's Open Access for RPG?

Many IBM i customers and application developers want to continue to use RPG in the long term. IBM Rational's Open Access for RPG (ROA) is a strategic investment in RPG, enabling native RPG code to deliver modern user experiences (UX) across multiple platforms, UIs and devices like web, RIA, iPhone, iPad and Android phones. This new native RPG capability is strong evidence of IBM's long term support for RPG, and will encourage future investment in RPG by customers and ISVs'.

## looksoftware products already support modern devices and multiple channels. So what is different about ROA?

Before ROA, looksoftware enabled i applications to support modern devices in the following ways:

- **Front-end refacing**

This involves refacing or screenscraping the 5250 datastream and enhancing it to support modern GUI controls like tabs, datagrids, combo boxes. The resulting UI can be delivered to multiple channels, for example, rich Windows clients, RIA clients and various mobile devices. This rapid, low risk solution, while successfully deployed across thousands of IBM i installations, does not provide a strong long term path forward due to the constraints of the 5250 datastream itself.

- **Back-end service-enablement**

Service-enabling existing 5250 applications involves exposing key chunks of back-end functions as reusable components or services. This approach has become popular because the UI is not restricted by the 5250 datastream. It is possible to design a completely new front-end using any web services compliant development environment, like looksoftware's development tools, .NET, Java IDEs, Web 2.0 tools, Websphere, Oracle Developer, our **smartclient** UI, even smartphones like the iPhone etc. Your service-enabled IBM i back-end can be connected to virtually any modern application, platform or device.

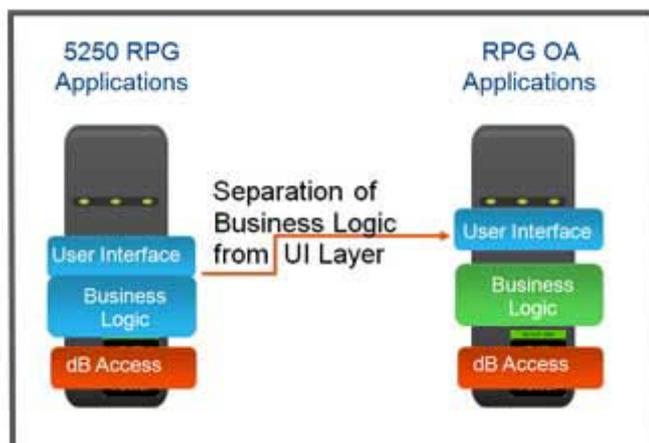
The services driven approach has become more popular than refacing over recent years, due to increasing wider interest in Service Oriented Architecture (SOA). By service enabling existing apps at the presentation layer (using a tool like **soarchitect's** transaction recorder) no changes are made to your existing RPG code. This approach introduces a new services layer providing logical separation of the presentation layer from the business logic, however the 5250 datastream still exists. Alternatively, you can invasively componentise your code. While time consuming, this approach can provide the benefit of physically separating the presentation layer from the business logic.

The great advantage ROA offers over both the above approaches is that it separates the presentation layer from the business logic layer by completely removing the 5250 datastream constraint, enabling much richer information to be available to external sources like browsers, mobile devices and other applications. The effort required to implement this separation simply involves the (automated) addition of a single keyword to existing programs.

Before ROA, there was no clear path forward for customers and ISVs wishing to deliver new 'native' RPG-based functionality with support for new devices and web services. ROA completely changes that.

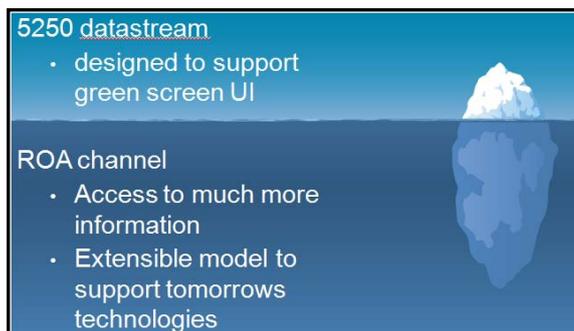
## Why is ROA creating so much interest throughout the IBM i community?

- ROA separates the presentation layer from the application layer or business logic, allowing existing applications to better support a multi-tiered application architecture. Popular modern software design patterns like Model View Controller (MVC) and n-tier architecture increase flexibility and reuse, by isolating the application logic from presentation. Prior to OA, it has been difficult, expensive and risky to modernize existing RPG applications to conform to these models. Now, enabling Open Access for existing RPG source programs is a simple automated process. Thousands of existing IBM i applications, reflecting billions of dollars of investment, can better support a multi-tier architecture model. This makes the IBM i a more attractive, long term path forward for companies with significant RPG investments and skills.



**Figure 1:** ROA gives existing RPG applications the multi-tier architecture expected of modern applications.

- ROA enables RPG to support new devices and technologies by providing a much broader 'channel' than the narrow 5250 'datastream' it replaces. For example, by enabling ROA, the full functionality of modern datagrids can be utilized. Features like flexible, user controlled paging and column sorting and sizing can be implemented without the programmer having to make any program changes.



**Figure 2:** ROA channel is as rich as required, unlike the limited 5250 datastream.

- **ROA makes programming GUIs easier.**
  - ROA dynamically generates modern, functional, intuitive GUIs directly from unchanged native DDS and RPG, without any coding effort.
  - The ROA handler architecture can generate multiple GUIs for varying devices from a single definition. For example, a drop down list can be generated from existing native DDS by the handler for a Windows GUI, a browser and an iPhone.
- **ROA provides a long term path forward for RPG development and RPG developers.**
  - ROA provides a native RPG mechanism for supporting new devices and emerging technologies, making it a suitable language for new application development.
  - Existing applications can be reused and extended to support browsers, mobile devices, web services and virtually any application, device, database or appliance.
  - ROA allows one set of native RPG based business rules to support multiple devices, and databases.
  - ROA is simple for existing RPG programmers to implement. It reuses and extends the value of RPG programming skills, enabling any existing RPG programmer to create modern functionality that leverages new technologies. The complexity of dealing with various devices and technologies is contained within the handlers which are typically provided by third party experts.
- **ROA is a clean, simple, open, native architecture**
  - ROA is an IBM provided, extensible framework for providing program input and output. It is not tied to any specific front-end technology or platform and is implemented deep within the operating system, offering excellent performance and scalability.
  - RPG developers can focus on the application business logic using their existing skills, and leave the technology plumbing to the experts.
  - Handlers can be developed by any customer or vendor, and may include support for any technology or device.
  - ROA is supported by RPG and DDS. Customers and ISVs can easily provide more functionality from these native IBM i languages. DDS does not support all GUI constructs, however the handler can generate many modern GUI controls from native DDS including datagrids date pickers, radio buttons, combo boxes,

check boxes, drop down lists etc. By continuing to use RPG and DDS native code where possible and handler extensions only where necessary, customers and ISVs can have the long term investment protection offered by native source code.

## What is an Open Access handler?

An Open Access handler is a program that communicates with RPG programs using standard file operations. The handler can then perform input and output operations with the end user through any mechanism that it supports – handlers can connect with databases, devices, appliances, other applications etc. The handler architecture is not limited to display devices. The RPG program will write to common file operations (like WRITE, READ, EXFMT) and the handler then manages the specific functionality required to operate with the target(s).

## Can customers and application developers create their own handlers?

Yes, however a custom coded handler for a specific program will provide little benefit over techniques already available. Writing a comprehensive handler capable of working with new and existing applications, and capable of supporting multiple client platforms, will require substantial development effort and expertise in both RPG and multiple client UX technologies. If the handler needs to handle input from the device, including validation, the handler will be complex. For a detailed example, refer to:

<http://www.ibmssystemsmag.com/ibmi/january08/developer/19724p1.aspx>

Using a handler that can support multiple programs and clients will allow IBM i developers to focus their expertise on business software development while still providing the rich functionality of modern clients.

Barbara Morris from IBM's ROA Development team said "The intention for RPG Open Access is for RPG programmers to be able to use their existing expertise in using the RPG File I/O model, whilst using other's expertise in accessing new resources and devices. In most cases the handler will be provided by an outside provider." (Excerpted from Rational Open Access: RPG Edition :

<http://www-949.ibm.com/software/rational/cafe/docs/DOC-3414>)

## If a third party handler is typically required, what is open about Open Access?

Open Access is a fully extensible framework for providing program input and output. It is not tied to any specific front-end technology or platform. Flexible handlers will be able to leverage multiple channels and provide cross-platform support.

No specific handler is required by Open Access, but it is important to remember that creating a handler to fully utilize the capabilities of Open Access is a significant task. Customers will be able to create their own handler if they choose, but will also be able to utilize handlers written by application modernization specialists. Customers will be free to

select whichever handler suits their needs, and the author of each handler will be free to extend them using any technology that adds value.

## What programming effort is required for an existing RPG program to benefit from Open Access?

Adding the ROA handler keyword is an automated once-off task, requiring no manual programming effort. Refer to the example below, which shows the Handler keyword added after the file definition for the Work Station. After recompiling the program the handler can render a highly functional modern UX without any further programming effort.

```
0008.00 *
0009.00 Fyd1c02df cf e          workstn
0010.00 F                      HANDLER('LOOKROA/LOOKROA(HANDLER)')
```

## How does Open Access make programming the UX easier?

ROA makes programming the UX much easier because the handler has access to much more information than the 5250 datastream. The ROA handler has access to the full meta data defined in the DDS.

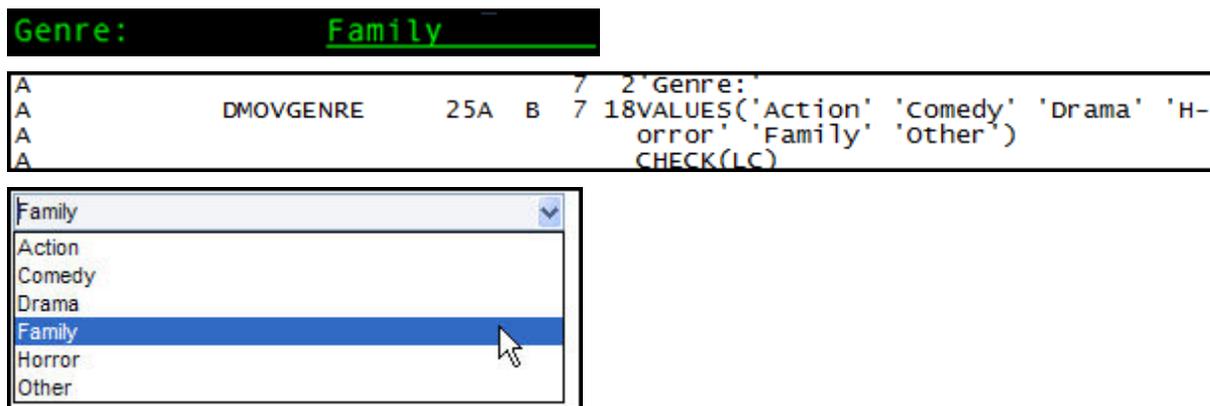
### Field level examples

Because the handler has access to more information such as datatypes, values or lists, more GUI controls can be automatically generated. These may include date pickers, radio buttons, combo boxes, check boxes, drop down lists etc. The 5250 sample green screen below uses DDS keywords like Values, Range and the Date Edit code which enable the ROA handler to automatically generate the appropriate visual control and load the associated values. The **look**software ROA handler supports native DDS, enabling the programmer to continue to use DDS and where desired add additional DDS keywords to generate more GUI controls.

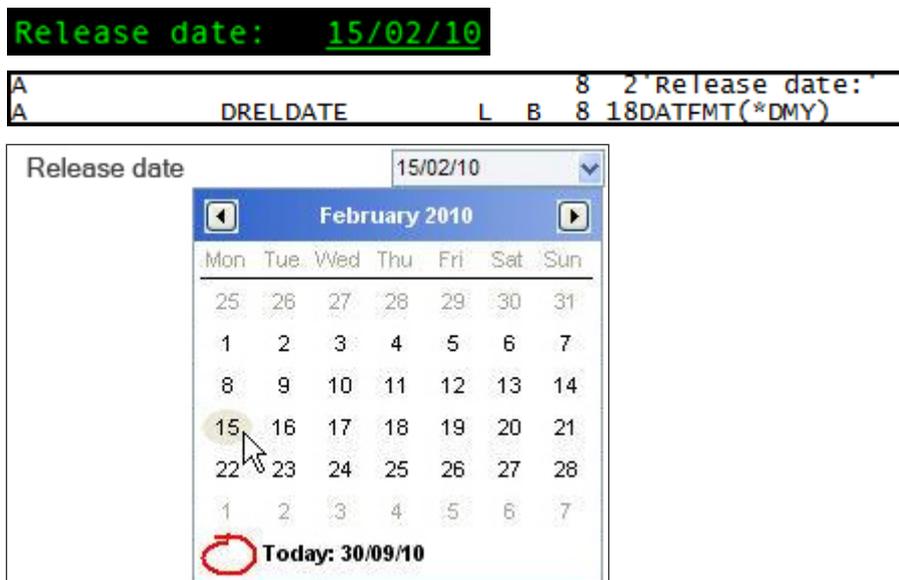
```
Genre:      Family
Release date: 15/02/10
Status:     Filming
Director:   John
Rating:     3
Cover Image:
In stock:   N
```

Genre	Family
Release date	15/02/10
Status	Filming
Director	John
Rating	3
Cover Image	
In stock	<input type="checkbox"/>

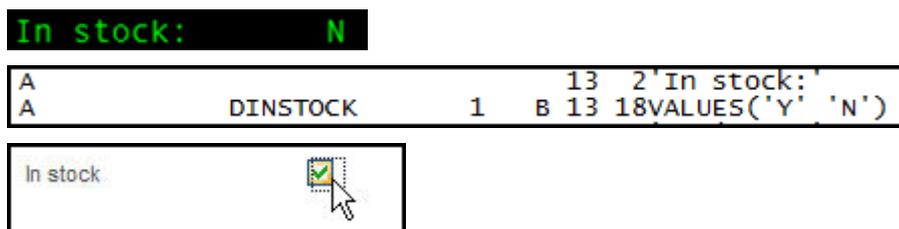
**Figure 3:** The ROA enabled green screen DDS generates the GUI above with no programmer intervention.



**Figure 4:** The VALUES keyword metadata for the Genre field allows the ROA handler to dynamically generate a drop down GUI control, eliminating the possibility of invalid data being entered.



**Figure 5:** The DATFMT keyword for the Release date field allows the ROA handler to dynamically generate a date picker GUI control, ensuring the business user can quickly select a valid date.

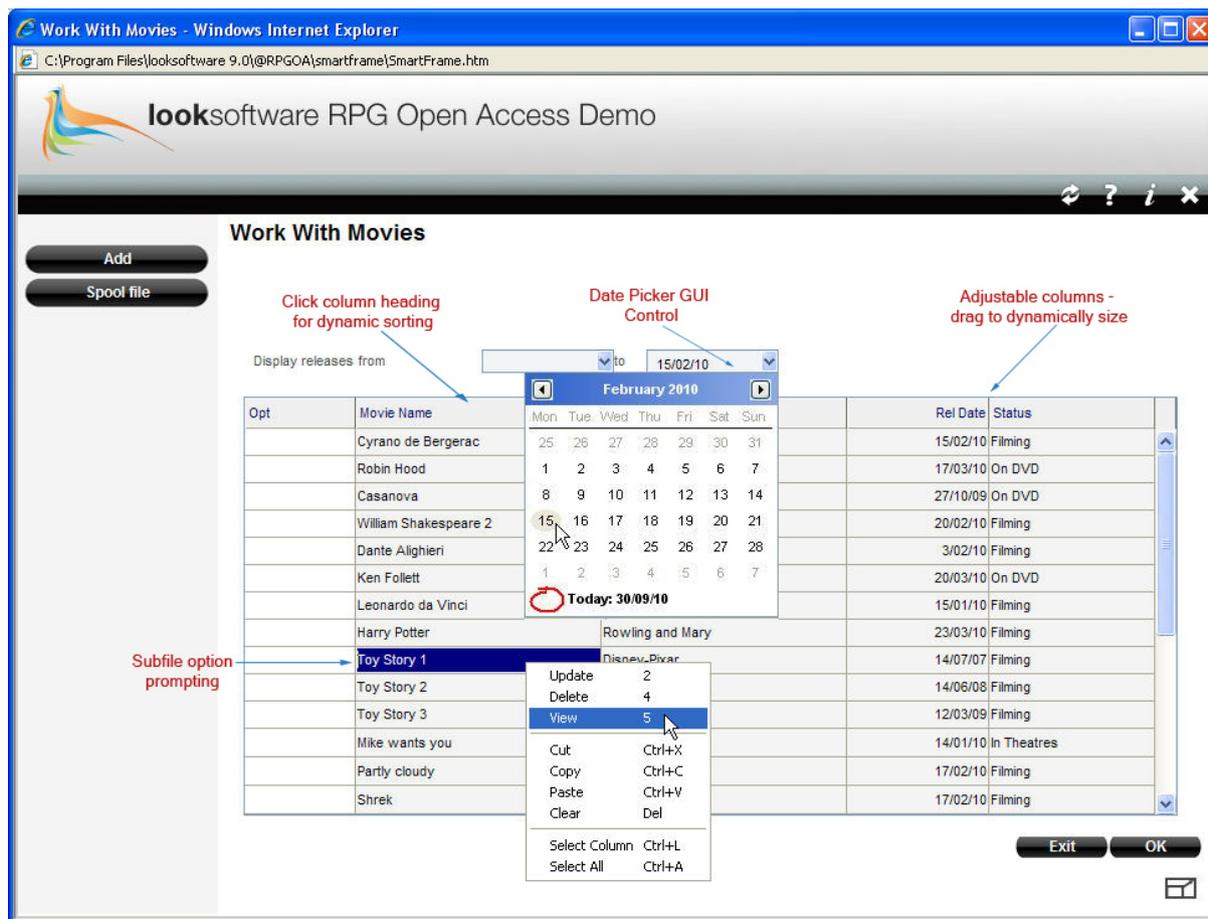


**Figure 6:** The VALUES keyword specifies Y and N and allows the ROA handler to dynamically generate a check box control.

### Subfile examples

ROA transforms green screen subfiles into functionality rich GUI datagrids and lists. Scrolling is no longer page based. The business user can scroll a single record at a time or drag the scroll bar to navigate quickly through the subfile. Datagrid generation, based on unchanged DDS, includes subfile option prompting, dynamic column sizing, dynamic column sorting and flexible subfile paging, supporting record at a time scrolling and the normal scroll bar.



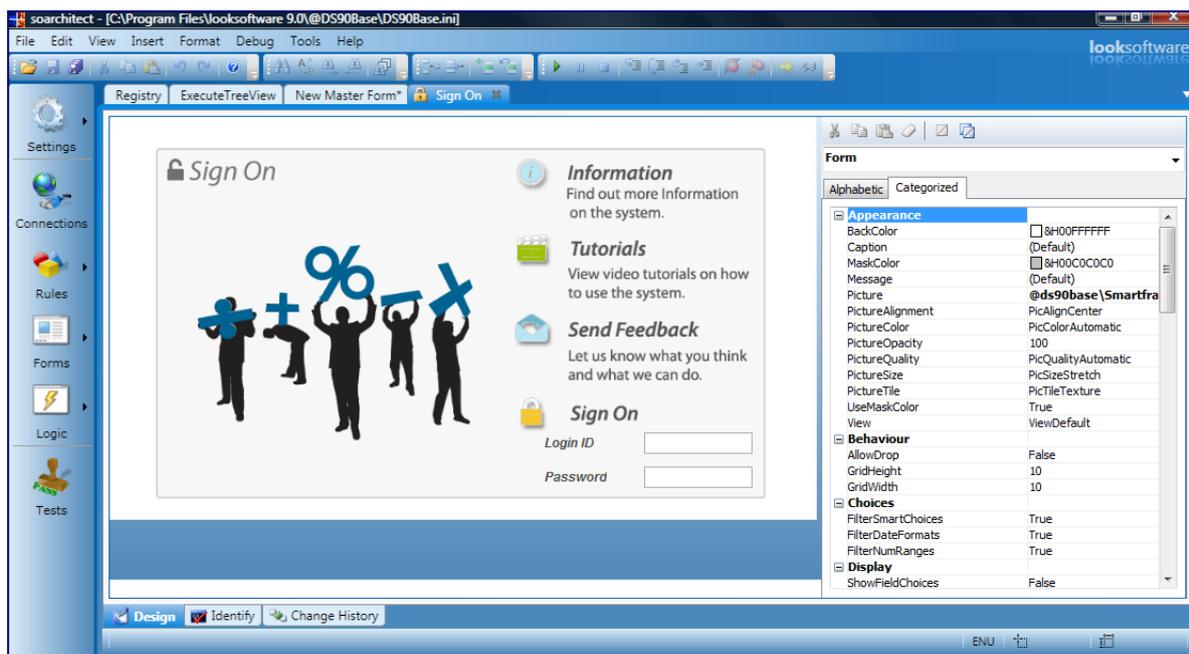


### Frameset look and feel

The previous example uses the handler's frameset to define the default look and feel which can be overridden at the application or screen level.

The open access architecture also enables the UX to be automatically optimized for multiple devices or channels. A sensible subfile paging default for smartphones may be different to a rich Windows user environment which may be different again to a zero deployment browser client. For example, the handler may send a few records to the iPhone user while the Windows rich client user (that has a powerful desktop processor) may receive a few hundred records. The handler can allow the developer (or if desired the business user) to set and override these channel specific defaults.

All the examples above require no changes to the underlying code. The looksoftware handler implementation preserves the native RPG and DDS. looksoftware's developer tools support Open Access and can be used to extend the UX to support additional visual controls like smart navigation, master forms, tree controls etc.



## Does looksoftware advocate RPG as the recommended path for IBM i customers?

There is no single 'right' modernization technology or product. As looksoftware's "*Secrets to Successful Modernization*" whitepaper advocates, customers need to conduct an assessment of their applications to determine the appropriate path forward. Please contact looksoftware for a copy of the whitepaper.

Over the years there have been many threats to RPG's popularity, however it has remained the dominant programming language for the IBM i, with Penton's recent surveys reporting RPG and ROA are the top programming language and development topics respectively.

Most IBM i customers have significant investments in RPG, and will appreciate the long term investment protection and reduced risk offered by ROA's native code approach. The ability to create your own handlers, use third party handlers and change handler suppliers while maintaining your business rules in native RPG provides flexibility to application developers.

## How is looksoftware supporting Open Access?

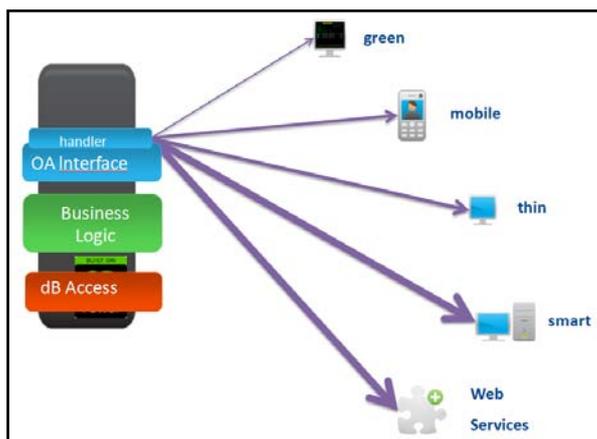
looksoftware is releasing a new IBM i based product, lookserver for Open Access and enhancing its existing product suite to leverage ROA.

looksoftware's support for IBM's Open Access for RPG allows RPG developers to deliver a 'native' web experience, supporting most popular platforms, browsers and mobile devices. By shielding the RPG developer from the technical complexities of the various front-end platforms, looksoftware enables the RPG developer to code business logic in RPG once, and deliver the application experience through multiple UX channels. looksoftware's support for any mix of Open Access-enabled RPG, existing traditional 5250 applications and commonly used operating system functions like Work with Spooled Files (WRKSPLF), means new Open Access code can integrate seamlessly with thousands of third party apps not enabled for Open Access. This provides the business user with a single

consistent User Experience (UX). The new RPG Open Access support is integrated with **looksoftware's** modernization suite to provide the RPG developer with a complete set of modernization tools and a long term path forward.

### Looksoftware's support for OA includes:

- Delivery of multi-channel UXs from a single RPG program
- Blending of ROA code with existing traditional 5250 code to produce a seamless, consistent UX
- **lookserver for Open Access**: a new IBM i based server product from **looksoftware**. Key capabilities include
  - Handler Support for ROA
  - Support for traditional 5250 based applications
  - Support for the IBM i HTTP web server
- Multi-channel support which delivers the optimum user experience for multiple audiences, GUIs and devices
  - Phased rollout for **smartclient**, **riaclient**, **thinclient**, **mobileclient**, web services and 5250 green screen support
  - Platforms supported include Windows, MAC and Linux
  - Browsers supported include Internet Explorer, Safari, Firefox and Chrome
  - Mobile device support includes iPhone, iPad, Blackberry, Playbook and Android.

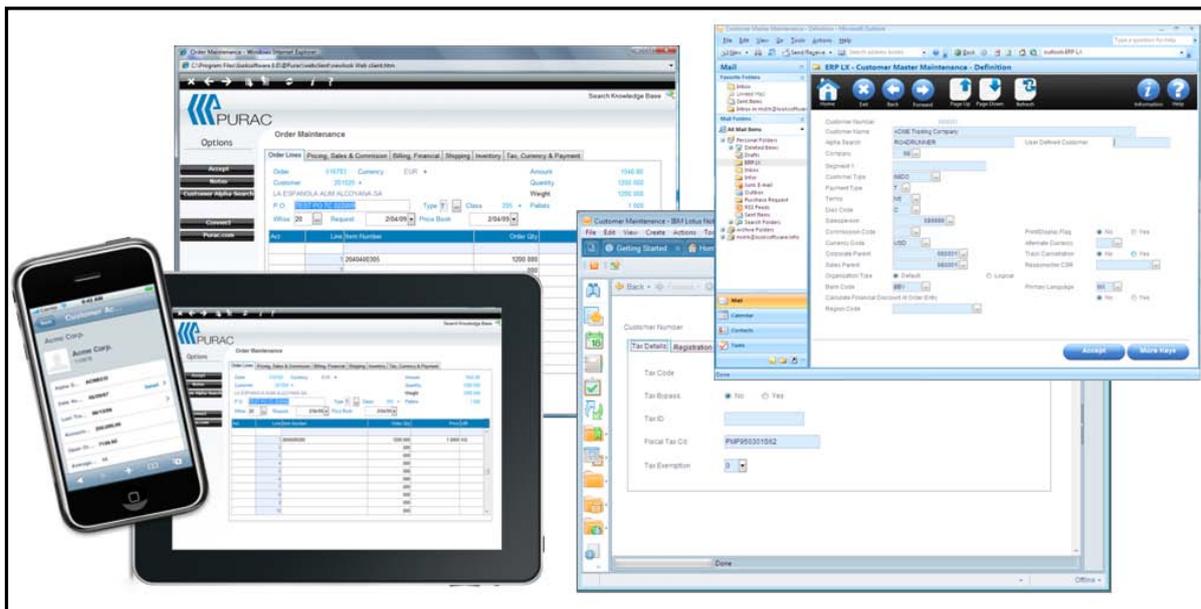


**Figure 7:** The IBM i OA architecture natively supports **looksoftware's** multi-channel handler.

## What is unique about **looksoftware's** support for Open access ?

- **Multi-channel support:**

Means your IBM i apps can easily reach more than 1 device or UX channel. Just as most IBM i customers found they needed to deliver GUI support, most have discovered (or will soon discover) they also need to deliver an acceptable user experience to mobile devices including iPhones, Android devices, iPads and next year's new devices that are in the labs today. Modern back-end applications need to reach multiple audiences and support multiple devices and UIs. **looksoftware** is delivering a single, generic handler to support multiple channels including green screen support, Windows rich clients, cross browser RIA, mobile device support and webservices.



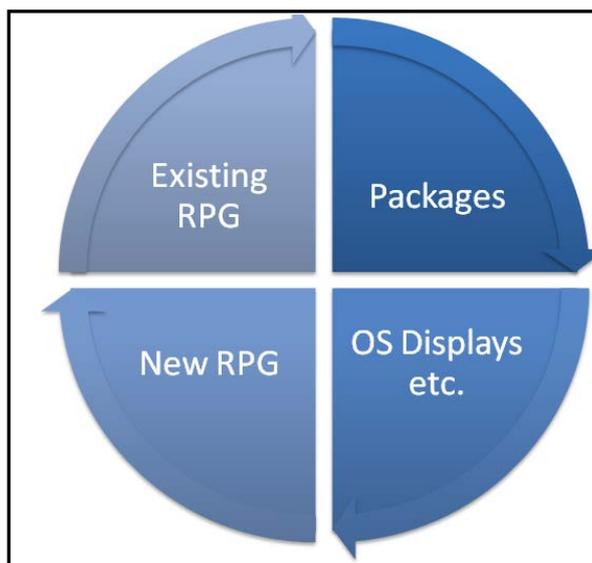
**Figure 8:** looksoftware’s generic handler enables a single set of RPG business rules to drive multiple channels including any browser, most mobile devices, IBM Lotus Notes, WebSphere, any Microsoft platform including Outlook, SharePoint, Silverlight, Apple’s iPad and iPhone and Google’s iGoogle Portal and Android based mobile devices.

- **Complete and holistic solution**

looksoftware believes a prerequisite for Open Access success is integrated support for existing 5250 based applications. We live in a world which is demanding integrated applications and streamlined business processes. If new ROA apps can’t seamlessly integrate with customers and third party ERP applications, deployments will fail. Given most existing IBM i applications are 5250 based, new ROA code needs to work with 5250 applications to provide the business user with a single, consistent, seamless User Experience in the same session. Some applications will not be suited to ROA modernization – like any modernization initiative, ROA modernization needs to be driven by a clear business need.

looksoftware’s support for OA includes:

- Automated modernization of non-ROA RPG to ROA RPG
- IBM i solutions often require access to non-ROA system functionality such as Operating System (OS) spool files and output queues. looksoftware’s handler supports these system functions, displays and OA functionality in the same UI and the same session.
- Delivery of a consistent UX across multiple channels, even if the source code is not available for all applications. Your customers (those that access your applications) expect a consistent user experience regardless of the device or channel they are using. If you want the business to accept the IBM i as the core back-end, you need to give them the seamless front-end experiences they expect.



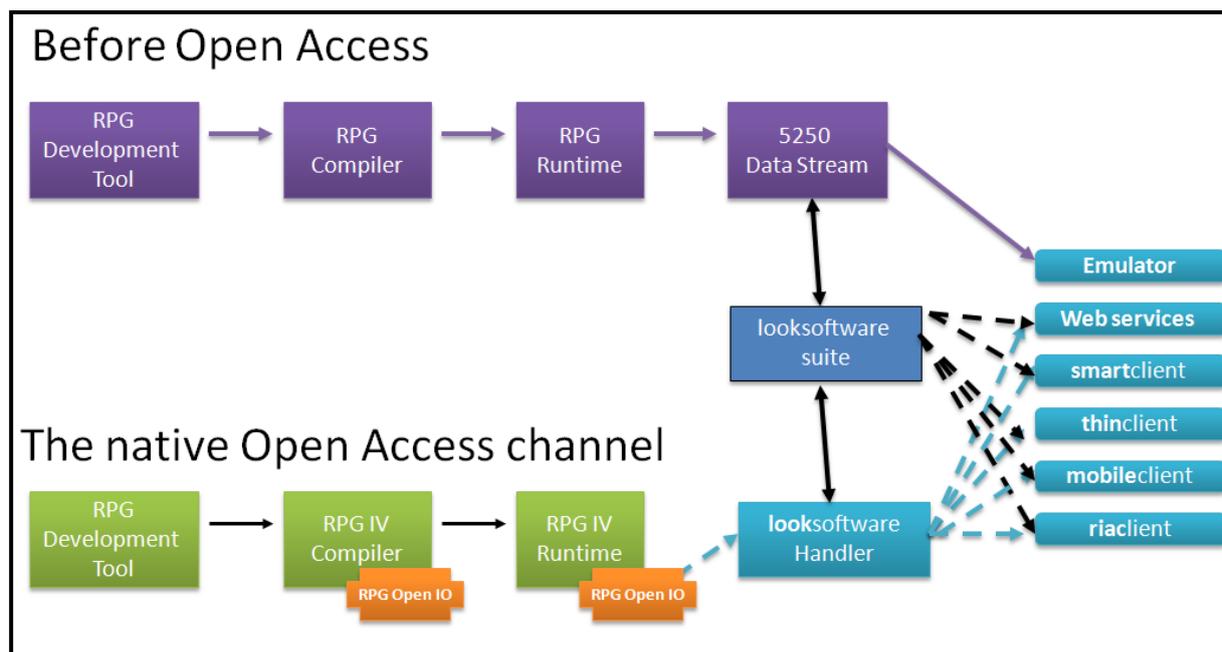
**Figure 9:** looksoftware's generic handler seamlessly integrates new ROA functionality with your IBM i application portfolio including your existing applications, 3<sup>rd</sup> party packages and IBM i operating system displays and functions.

- **Encourages native code.**

IBM's ROA has been delivered in response to requests for a native GUI. The ROA architecture supports native RPG and DDS and an extensible model to support specific devices and technologies. looksoftware's handler supports native code where possible, keeping your long-term options open and your investment protected.

## What impact does ROA and these new looksoftware announcements have for looksoftware customers and ISVs?

- ROA extends looksoftware's support beyond reuse to new development. Customers and ISVs can code new native RPG functionality that directly leverages all looksoftware functionality, without the constraints of the 5250 datastream.
- looksoftware's investment in ROA includes ROA enabling all our products. For example, we are making significant enhancements to newlook for new OA functionality, however newlook will continue to work with 5250 based applications.
- Protection and enhancement of your looksoftware investment. ROA is an IBM recommended path forward for companies with RPG investments.
- Reduced modernization effort. By ROA-enabling your existing applications, we can generate a richer UX out of the box, and it will require fewer looksoftware repository rules.
- smartclient is used by hundreds of thousands of IBM i users to provide a rich user experience supporting desktop integration, web services support etc. for existing 5250-based applications. By using looksoftware's new IBM i-based lookserver ROA Edition, smartclient users will be able to access their IBM i back-ends via a web services connection, rather than a telnet connection, without needing to change their looksoftware repositories.
- Easier new RPG-based development, because looksoftware's existing development tools are being extended to provide a seamless developer experience for the RPG/looksoftware developer.
- Customers may blend ROA and non OA based functionality.



**Figure 10:** looksoftware ROA support integrates seamlessly with non-ROA applications in the same UI in the same session.

## IBM i community and customer support for Open Access?

*"IBM's announcement to 'free' RPG from the bond of the 5250 datastream makes RPG the best solution for the development of business applications supporting the Model View Controller architectural pattern. RPG's suitability for enterprise application development is enhanced by looksoftware's ability to support multiple devices with a single handler."*

**Toscano Alberto, CTO, Sidim Spa**

*"Open Access architecture allows the IBM i sites to reuse their RPG investments, to support today and tomorrow's new devices and technologies including web services and the Cloud, rather than starting again."*

**Rod Riley, Stafford Group**

*"For me, the largest significance of Open Access is that it removes the last possible reason that anyone may have had for not starting the modernization process for their applications."*

**Paul Tuohy, ComCon**

*"The combination of RPG Open Access and looksoftware finally gives us the application development future we have been waiting for. Now we can modernize our existing programs efficiently and at the same time take full benefit of the new RPG Open Access options. It also covers the need of our users for flexible graphical clients - including Rich Internet Clients and mobile devices."*

**Norbert Bruckschlögl, Manager IT DEHN + SÖHNE**

"IBM is making an investment in the future of IBM i with RPG Open Access. **look**software adds value to their investment, enabling customers to take advantage of IBM's native GUI. With the capability to deliver multiple UIs from a single RPG program, the **look**software tools manage the complexity of the myriad of modern user interface clients. RPG developers will focus less on the differences of browsers and mobile clients, and become more productive with the business logic of their applications."

**Trevor Perry, Angus Thinks**

"Traditionally, all output from RPG programs were forced into the limited 5250 data stream. In 7.1, the output could be sent to, for example, a web page or hand-held mobile device. I think this is what developers have always wanted."

**Nick Cummins, Parker Hannifin, IBM I customer**

"RPG Open Access is very much the kind of RPG enhancement we had been lobbying to get for years - even before we left IBM more than 12 years ago."

**Jon Paris, Partner400**

"This Open Access function is a dramatic new option for RPG applications—if you want to bypass the 5250 requirement, this lets you write applications for the first time without outputting them to a 5250 data stream, and this will really open up a wide range of new application possibilities".

**Ian Jarman, Manager of Power Systems Software, IBM**

"ROA satisfies the #1 request from customers and ISVs in an open architecture and is developed and supported by IBM, the platform provider. ROA provides the long term, native code, path forward for any company with RPG investments"

**Marcus Dee, CEO, looksoftware.**

"We are also excited about the release of Rational Open Access: RPG Edition, a new product that has the potential to transform the landscape of RPG development."

**Hayden Lindsay, VP Rational, IBM**

"For years, we were almost ashamed to admit that we are "old fashioned" RPG programmers. Now, with ROA and **look**software's Handler, you can come out of the closet - we're back on the leading edge .. by simply adding ONE line of code to existing applications."

**Alex Reiss, CTO, Eclipse Corporation, IBM i Application Vendor**

"This technology potentially permits RPG programs to take full, direct advantage of an enterprise service architecture. ROA is really attractive because it lets the RPG programmer focus on logic and content."

**Mandy Shaw, iPerimeter, IBM i consultant**

"For me, ROA is simply the proof that IBM is staying in the RPG game. I have applications written in a variety of languages but my fastest method of getting information from the DB to the UI is still RPG."

**Brian Bradley, Vilden, IBM I Application Vendor**

*"With its versatility and ease of use, it's good to see RPG can still be a contender in our current age of technology. From someone who has worked with it for the past 25 years I see endless possibilities."*

**Scott Miller, Atlas Energy, IBM I customer**

*"RPG Open Access announcement really excited the IBM i application development community. This architecture is the promise to re-launch IBM i as the platform where developing modern business applications is easier and more productive. This can be achieved only with the support of products, like **lookserver** for Open Access, that complete the base technology with the tooling to develop the new GUI for RPG applications and to integrate it via RPG Open Handlers."*

**Massimo Marasco - IBM i Technical Support - Italy**

## What does Open Access offer that RPG Special files does not provide?

An ROA enabled Display file is similar in nature to a SPECIAL file. A SPECIAL file also uses a user-written program to handle the operations for the file, and it allows additional parameters to be passed to the handler from the RPG program.

The two main differences between an ROA enabled Display file and SPECIAL files are

1. A SPECIAL file only allows the operations available for a sequential (SEQ) file. An Open-Access file can be used for any type of RPG device.
2. A SPECIAL file handler only receives a minimal amount of information about the file operation. An Open-Access file handler receives much more information such as the name of the file, record format, the names and types of the fields.

ROA alternatives like SPECIAL files and third party proprietary alternatives were developed over the years. Their limited use is an indication of the value they provided. ROA offers more functionality, an open architecture and is developed and supported by IBM, the platform provider. ROA provides the long term, native-code path forward for those companies with RPG investments.

## Where can I learn more?

- **IBM** and **looksoftware** jointly deliver regular webinars about ROA. You may register [here](#) for the next webinar or email [marketing@looksoftware.com](mailto:marketing@looksoftware.com) to learn more about ROA.

The webinars include live demonstrations and show how new functionality coded with RPG Open Access can deliver a web GUI natively, using the Open Access handler feature, without the requirement for any 5250 coding or data stream. The demonstration will also show how new Open Access functionality can seamlessly integrate with traditional 5250-based green screen programs including system functions like WRKSPLF and ERP packages likes JDE.

- IBM i customers and ISVs interested in knowing more can register for an Early Adopters program to receive more information, by emailing [marketing@looksoftware.com](mailto:marketing@looksoftware.com)
- Rational Open Access: RPG Edition by Barbara Morris : <http://www-949.ibm.com/software/rational/cafe/docs/DOC-3414>
- Visit <http://www.looksoftware.com/products/deployment/lookserver-rpg-oa.aspx>



looksoftware USA 678 354 1094

email: [info@looksoftware.com](mailto:info@looksoftware.com)

web: [www.looksoftware.com](http://www.looksoftware.com)