



# Powerful and Innovative



White Paper



INTERNATIONAL  
DECISION SYSTEMS®



## Increasing Information and Reducing Risk

### Executive Summary

In business, the contest isn't usually won by yards; it's won by inches. Competition can be fierce, and any advantage can be critical. Hence, the growing emphasis on streamlining, right-sizing, and process enhancement. *Rapport*, a web-enabled front-end, multiplies the effectiveness of *InfoLease*, the world's most powerful engine for managing leasing information, as well as the effectiveness of other back office systems by making information more accessible and usable inside or outside of your organization, via the web.

*Rapport* helps simplify operations by matching workflows and user interfaces to your actual business practices. By customizing information and its presentation to fit your organization and its processes, it reduces costs, increases flexibility, and helps eliminate errors. It makes these capabilities available both internally and to external partners, to get paperwork out faster and get sellers paid without unnecessary delay. In short, *Rapport* eliminates "friction," and that means increased competitiveness in today's fast-moving, increasingly global markets.

All business entails risk. You make an investment and nurture it in the expectation of future returns. But the return on your investment doesn't necessarily correspond to your effort; it depends on your ability to function in a competitive environment in which the winner gets gold, but there is no silver or bronze.

Business today means teamwork, not just within departments, but throughout the organization and with external partners. And every step along the way – each individual's work and every "handoff" – is an opportunity to either pull ahead or fall behind. So when a process is repeated hundreds or thousands of times, even a small increase in efficiency can make a big difference in bottom-line results.

### Windows of Opportunity

Business is all about timing. Openings appear, sometimes without warning, and disappear just as quickly, so there is simply no room for delay or wasted effort. The good news is that there is a powerful new tool that cuts through clutter, increases efficiency, and puts the information you need at your command. *Rapport* is a fully integrated, web-based front-end solution that provides complete access to the vast store of information needed for the critical process between application and booking to help users work more effectively and make better decisions.

Surveys have shown that, even with pricing, credit, service, and accounting systems in place, lessors still feel starved for information in the contract origination/booking phase of their operations. What they need is a way of connecting databases and putting information into a more usable format. That's what *Rapport* does with any Back Office System.

Figures 1 and 2 (on the following page) are examples of the custom screens that can be developed using *Rapport*. Each includes real-time information from your Back Office System database, but in a layout tailored to the user's standard workflow.





*Rappport* links seamlessly with any Back Office System, allowing information to be accessed in real time. It interfaces easily to most major credit bureaus, so users can choose the system they prefer for instant online credit information. It interfaces with a variety of credit scoring systems, and it integrates with web-based pricing applications such as the *InfoAnalysis* Quoting Engine, allowing users to provide fast, accurate online payment quotes.

But *Rappport* isn't just a one-way system; it streamlines data entry as well. Seamless integration makes *Rappport* the single point of entry for all kinds of data. Information that might otherwise have to be entered many times can be entered once and moved automatically throughout the company. "*Rappport rules*<sup>TM</sup>" can be defined such that data entered into one field triggers calculations of data in related fields. These capabilities not only speed up processes and eliminate redundant effort, but also reduce the likelihood of costly errors. Figure 3 shows how information from multiple users and systems can be collected and displayed in a customized format.

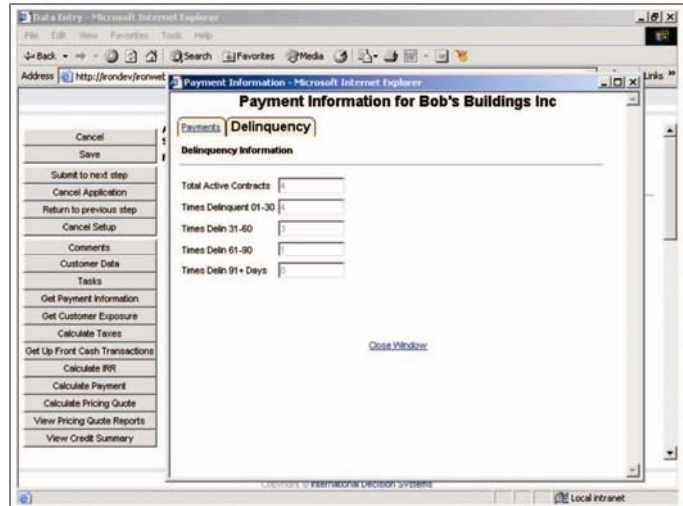


Fig 1

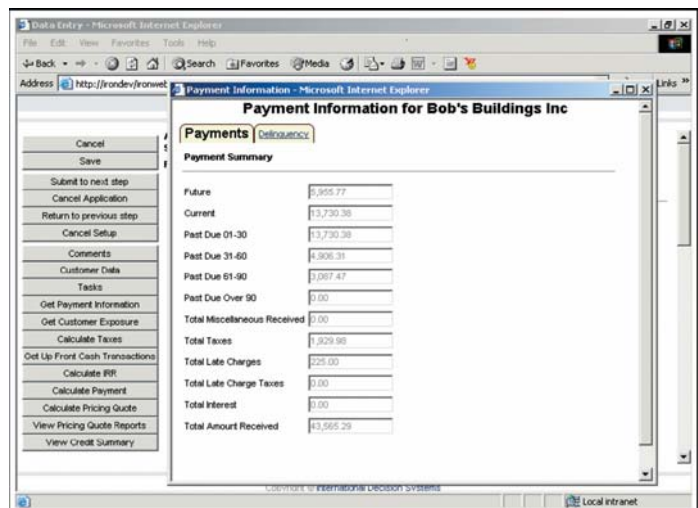


Fig 2



INTERNATIONAL  
DECISION SYSTEMS®

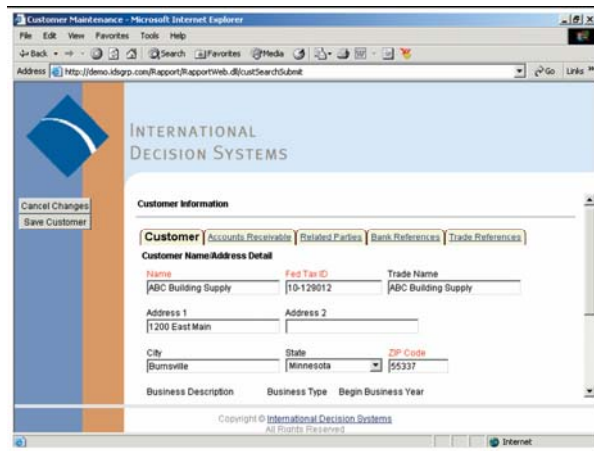


Fig 3

### Building a Business Community

*Rapport's* web-based access lets you reach beyond internal users to vendors and dealers. Online access lets them serve their own customers better with instant information directly from your Back Office System database. You can even let these partners, or your customers, update credit application or booking-related information directly. And by providing easy access, you can build your own competitiveness with service other systems can't match.

Best of all, you can tailor your system to serve multiple audiences, each with its own needs, without any kind of compromise. User-defined templates let you create as many different task-oriented screens as you want. Each screen contains all the information needed for a particular task, with no unnecessary data or irrelevant fields. That means that the user, whether internal employee or external partner, no longer needs to wade through pages of information to get what he or she needs. Access is quick and easy. The user sees only the information he or she routinely uses, which means more productivity at less cost.

*Rapport's* flexibility accommodates you instead of forcing you to adjust to the system. Every leasing company has its own processes for lease origination, approval, review and booking. And *Rapport* lets you customize workflow to your way of doing business using a simple administration tool. When your workflow changes, you can alter *Rapport* to match your new processes. This flexibility speeds up user acceptance of the system and can save thousands of dollars in custom development and training costs.

### Multi-level Controls

Of course, with increased access, security is a critical issue. *Rapport* divides its functions among a database server, an application server, and a web server. These, in turn, can be separated from one another, and from end-users, by multiple firewalls to protect the servers from external attack. The data itself is protected with secure socket layer (SSL) encryption, ensuring that it cannot be misappropriated while in transit over the web.

Yet, despite all this flexibility and sophistication, *Rapport* is simple and economical enough for any organization with any number of users. Easy integration with your Back Office System eliminates the need for custom programming or interface development. And graphical, task-oriented screens eliminate much of the training needed with less user-friendly systems as shown in figure 4.

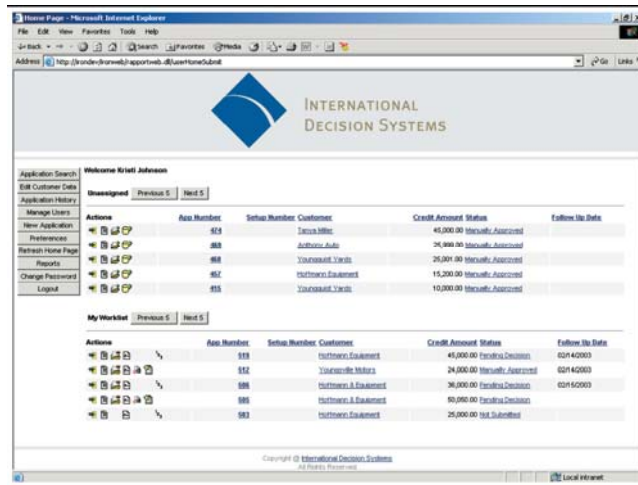


Fig 4

## Smart Systems for Smarter Business

*Rapport* simplifies the interface between users and the system, making information easier to enter and access, and far easier to use. This is increasingly critical in today's fast-moving market. As global competition cuts into profit margins, better decisions mean more income and fewer losses. And doing more with the systems you already have helps keep overhead low.

*Rapport* extends the reach of your Back Office System, both throughout your organization and into your community of vendors and dealers. It reduces the cost of entering and maintaining data. It speeds access at the critical point where leases are being booked, and lets you bring more information to bear in making leasing decisions. In short, it performs the most critical, yet most elusive of business functions: it lets you build your business while reducing risk.

## *Rapport* Technical Architecture

*Rapport* is a highly configurable lease origination system that combines lease pricing, bid tracking, credit decisioning, and the booking of a completed lease into your Back Office System. It uses an innovative technological approach to combine the no-configuration deployment of a web browser with the responsiveness of a Windows application. It also allows virtually unlimited flexibility in configuring both the user forms and the workflow of the system. In this way, it improves productivity by providing information on a "just-in-time" basis so the user sees only the information needed at the time.

Two features of *Rapport* that set it apart from other applications are its use of XML at the core of the system and its use of *Rapport Rules*<sup>™</sup>. *Rapport Rules*<sup>™</sup> are custom business rules defined specifically for each customer's operations. If an end user, for example, types a value into a field, *Rapport Rules*<sup>™</sup> can automatically calculate other related values based on that input. A typical customer may set up as many as 100 of these rules in their forms. Another key feature, more widely found in software today, is the use of web services for integration with other systems. These include *Windows InfoAnalysis* for stand-alone pricing, and credit bureau and credit scoring systems.



## Application Layers

*Rappport* is a traditional multi-tier application composed of three application layers and several web services interfaces. The three application layers are:

- ◆ Presentation Layer
  - Web browser for the end user
  - Windows-rich client for the administrator
- ◆ Business Logic Layer
  - COM+ components
- ◆ Persistence Layer
  - Microsoft SQL Server

The details of each of these layers are discussed below.

## Presentation Layer

For users of the system, the presentation layer consists of a web browser interface, specifically Microsoft Internet Explorer (IE) 6.0 or above. The system takes advantage of the native XML capabilities of IE to create a user experience similar to that of a Windows application in terms of responsiveness. All manipulation and formatting of the data, such as sorting, moving from form to form, and entry, is done at the client. This avoids the problem, typical in browser applications, of even the smallest change to a web form requiring a round-trip to the server. (In traditional web browser applications, this can cause numerous small delays, which become irritating when using an application all day.)

*Rappport's* client architecture allows the work of displaying and editing data to be done by the client, which has plenty of power for this type of work. This, in turn, allows the web and application servers to do much less work than in a traditional web application. Another benefit of this architecture is that it minimizes network traffic. This makes the system highly scaleable by off-loading all display work to the clients, minimizing network traffic.

*Rappport Rules*<sup>™</sup> are also implemented at the client level using ECMAScript (a.k.a. JavaScript). This allows maximum responsiveness for actions commonly implemented in *Rappport Rules*<sup>™</sup> such as default values, calculated values, and conditionally required values.

For system administrators, the presentation layer consists of a rich Windows client. This was chosen because *Rappport* is highly configurable and the administration tool user interface can be quite complex. Based on experience with other web applications, it was determined that the administration tool had outgrown the capabilities of a web browser.

We also chose a Windows client for the enhanced security it provides; all connections to the servers are via the corporate network, which limits unwarranted access to the program. This is additionally justified by the enhanced experience of the administrator. Since the administration tool is used by just a few people, the requirement of a client installation is a small issue.

The administration tool is where the configuration of forms, workflow, pricing and documents is done.



## Business Logic Layer

The business logic layer of *Rapport* consists of “engines” written as COM objects and running on the Windows Server operating system. Some of the business logic is contained in the *Rapport Rules*, which actually execute in the client web browser. However, the definitions of the *Rapport Rules* are contained in the database and managed by the application server. Diagram 1 show the six engines that comprise the *Rapport* business logic layer.

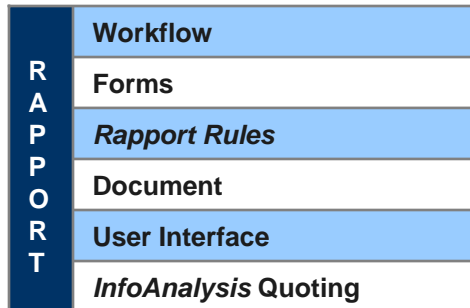


Diagram 1

The engines allow the system to be configured for a customer’s way of doing business. There are six primary engines in *Rapport*.

- ◆ Workflow engine
- ◆ Forms engine
- ◆ *Rapport Rules* engine
- ◆ Document engine
- ◆ User interface engine
- ◆ *InfoAnalysis* quoting engine

### Workflow Engine

The workflow engine is configured for the specific workflows of a leasing company. It allows creation of user roles for tasks such as pricing, quoting, initial entry of data, credit decisioning, documentation, setup, and booking. The engine allows flexibility based on size of transactions, geographic location of customers, type of transaction and many other parameters. Almost any field that being tracked for a credit application can be used in controlling the flow of the application through your company.

### Forms Engine

The forms engine is where *Rapport* goes far beyond other systems. It puts complete control of form creation in the hands of the administrator without requiring programming changes in *Rapport* itself. Forms are simply configured using the administration tool.

Variables include:

- ◆ Fields included on a form
- ◆ The order of fields
- ◆ Tabs under which the fields are grouped
- ◆ Names of the fields
- ◆ Creation of user-defined fields
- ◆ Properties such as required, disabled, hidden, etc.

The customer controls all setup and modification of forms, allowing easy updates as the business evolves. This customer-controlled flexibility permits creation of forms for specific products and programs. At the same time, it minimizes data entry and the number of fields any individual needs to deal with, preventing avoidable errors.



The user-defined fields are “first class citizens” in *Rapport*. They can be:

- ◆ Used in *Rapport Rules*
- ◆ Criteria in the workflow
- ◆ Fields in documents
- ◆ Passed to credit bureau and credit scoring systems

Creation of multiple forms can be simplified using a Form Group Template (FGT). This defines a set of fields that are logically grouped together, e.g., asset-related fields. These can be used across multiple forms. When the FGT is updated in one place, all forms that are using it are updated automatically.

### **Rapport Rules Engine**

*Rapport Rules* are a sort of hybrid, stored on the database server, managed by the application server, set up by the administrator, but finally executed on the client. All the details of a rule, such as the field that triggers it, the fields involved, and the field updated by it, are stored in XML. In the web browser, the ECMA Script executes rules based on trigger parameters. This architecture results in extremely flexible forms that can do a lot of work “behind the scenes”. It also enhances flexibility, in that processing is done at the client and does not require a round-trip to the server for execution of each rule.

Events that can trigger *Rapport Rules* include the addition, deletion, or replication of groups, e.g., asset groups. In such a case, it would allow the form to automatically keep a running total of total cost or number of assets.

*Rapport Rules* can be reused across multiple forms. And because they can be tied to specific forms, you can create *Rapport Rules* that are specific to a program, a product, or a step in the workflow. Hidden fields can be referenced by *Rapport Rules* as well. This means you can do as much defaulting or as many calculations “behind the scenes” as you wish, all without cluttering up the screen for the end user.

### **Document Engine**

*Rapport* supports two different approaches to the creation of documents. Customers who don’t require complex documents can create Adobe PDF templates. These contain placeholders for the dynamic data that is pulled from the actual application data. The resulting PDF document contains data fields that can be e-mailed to the client or displayed in a web browser for printing on-site.

For those customers who need more complex documents, *Rapport* incorporates XSLFO technology for the ultimate in flexibility. XSLFO stands for Extensible *Stylesheet* Language Formatting Objects. It is a W3C standard specifying how XML data can be formatted for output to various devices, including PDF files. Since *Rapport* stores data in XML, this is a natural fit. XSLFO contains a simple scripting language that allows almost unlimited flexibility in document creation. The *Rapport* document engine inserts credit application data into either the PDF template or the XSLFO definition. The end result is a PDF document that is available for the client to print.

### **User Interface Engine**

The user interface engine controls the look and feel of web browser forms and the correct display of header and footer information. This is accomplished using a User Interface Template (UIT). The UIT contains form header and footer information that can be customized for different dealers and vendors. There is also a Cascading Style Sheet (CSS) associated with each UIT, which determines the font, font size, colors, backgrounds, etc. If there is no CSS associated with a UIT, the default CSS is used. This presents a consistent look and feel for all users, which can be particularly helpful for internal users.



Header and footer information also allows the creation of private label programs, so vendors and dealers can present custom-tailored forms to their customers. A unique UIT can be associated with each user of the system.

### *InfoAnalysis Quoting Engine*

The *InfoAnalysis* Quoting Engine (IAQE) has been tightly integrated into *Rappport* to allow flexible, but controlled pricing that can be safely rolled out to vendors and dealers. This lets you control yield and residual parameters. It offers greater flexibility than a rate sheet for vendors and dealers, while assuring that any pricing they come up with, while creative, still meets your profitability standards.

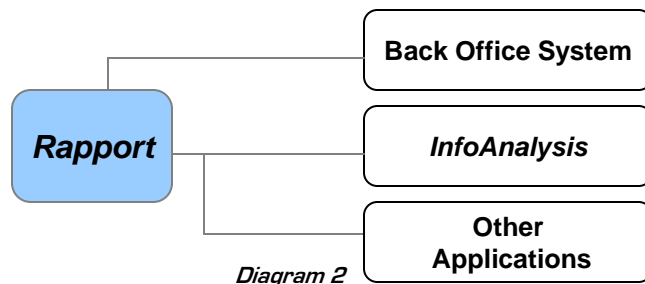
While the IAQE is a stand-alone product, it functions within *Rappport*, through the use of web services, as an integral part of the system. Fields used for pricing are embedded in the web browser forms. By judicious use of *Rappport Rules* you can ensure that all pricing is done through the engine to help maintain your profitability. Configuration of the IAQE, including the entry of rate, equipment, residual, and balloon values, is done through the *Rappport* administration program.

### Persistence Layer

The persistence layer of *Rappport* consists of storing all data in Microsoft SQL Server, using both XML and standard database fields.

### Web Services Interfaces

*Rappport* supports two different web services interfaces to allow integration with other applications. Diagram 2 show these interfaces.



### Real-Time Back Office System Interface

*Rappport* uses a web services interface directly to your Back Office System for real-time access to data and real-time synchronization of customer data. This is in contrast with some other systems, which rely on daily – or less-frequent – batch transfers of data from your Back Office System. Such batch transfers can allow data used for credit decisions to be out of date. The customer update process between *Rappport* and your Back Office System, on the other hand, provides true two-way synchronization. Data changed in either system is updated to the other system in real time. Information that can be retrieved for making credit decisions includes payment history, delinquency, and customer exposure.

Finally, this real-time interface is used to book new contracts directly into your Back Office System. The booking process in *Rappport* calls the Universal Contract Interface (UCI) on an interactive basis. If any errors occur in this process, an error message is returned directly to the user for immediate action. If the contract books successfully, a contract number is returned as confirmation.



### Standard Web Services Interface

*Rappport* provides a standard SOAP-based web services interface that can be used to initiate new credit applications. Windows *InfoAnalysis*, the IDS stand-alone lease pricing system, supports this interface, allowing credit applications in *Rappport* to be initiated from within Windows *InfoAnalysis*.

Windows *InfoAnalysis* packages the current quote into an XML message. This is passed, via SOAP, to *Rappport*, where authentication and customer matching are done, allowing salespeople to send *InfoAnalysis* quotes directly into *Rappport*. This turns the quote into a credit application and then processes the credit decision, saving time and avoiding duplicate entry.

This web services interface is also available to other applications, using a standard SOAP interface. Such applications might include the web site of a vendor or dealer, who is collecting application data on their site, or your ERP system, which is initiating credit applications as part of a sales process.

### Physical Layers

Diagram 3 represents the different physical layers of *Rappport*, along with protocols used to communicate between layers.

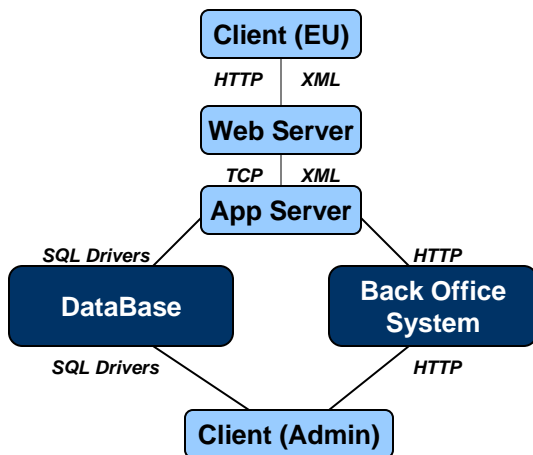


Diagram 3

*Rappport* is composed of six different physical layers. This discussion assumes that they reside on different machines; however, some layers can be combined onto a single machine if appropriate.

The physical layers are:

- ◆ End-user client – web browser
- ◆ Administrator client – Windows
- ◆ Web server
- ◆ Application server
- ◆ Database server
- ◆ Back Office System server



### **End User Client**

The end-user client consists of Microsoft Internet Explorer 6.0 or higher, running on Windows 98 or later. This client communicates with the web server via HTTP. The main data type being passed between the web browser and the web server is XML.

### **Administrator Client**

This is a Win32 application running on Windows 98 or later. It requires the installation of Microsoft SQL Server drivers on the client in order to access the *Rapport* database. This application communicates directly with the database without using the application server or web server. It also communicates with your Back Office System using WebConnect technology.

### **Web Server**

The web server is Microsoft Internet Information Server (IIS). It serves mainly as a conduit between the web browser and the application server and performs some XML transformations.

Communication with the web browser is via HTTP, and communication with the application server typically uses a sockets interface. SSL security is supported with both the web browser and the application server. Session state is stored in the database, allowing use of multiple web servers (a “web farm”) if warranted by the load.

### **Application Server**

The application server runs on Windows Server 2000 or higher and consists of a set of COM+ components. Communication with the web server is typically via a sockets interface. SSL security is supported. Communication with the database server uses Microsoft SQL Server drivers. Communication with your Back Office System is via WebConnect, which employs web services. Because session state is stored in the database, multiple application servers can be employed if the load demands it.

### **Database Server**

The database server is Microsoft SQL Server 2000 or higher. It communicates with the application server using Microsoft SQL Server drivers. The system is slated for future porting to Oracle.

### **Back Office System Server**

For communication with *Rapport* and *InfoLease*, your Back Office System server uses WebConnect, which employs HTTP as a communication protocol. WebConnect manages a pool of connections to your Back Office System to minimize the need for UniData and *InfoLease* licenses. For communication with non *InfoLease* Back Office Systems, *Rapport* employs HTTP as a communication protocol.

## **Credit Bureau / Credit Scoring Interfaces**

Web services are becoming the standard way of integrating applications, especially applications from different vendors. This is the approach *Rapport* uses with credit bureau/scoring companies. The process typically involves writing an Extensible Stylesheet Language Transformation (XSLT), which transforms the XML document into an XML document that can be understood by the credit bureau/credit scoring company. The XML document is passed using SOAP and HTTP.

Note that the transfer of data from *Rapport* to the credit bureau/scoring company need not be done over the Internet. It can be done over a leased line or other private facility, which allows enhanced security and reduced latency. Typically, *Rapport* sends the appropriate customer data to the credit bureau/scoring company, which returns a “hit list” of potential matches on the customer name and address. The appropriate choice is selected in *Rapport* and returned to the credit bureau/scoring system. *Rapport* then receives back the credit data, which is stored in *Rapport* and available for review. The resulting credit score can also be used as a criterion in the workflow system to decide what to do with the application.



INTERNATIONAL  
DECISION SYSTEMS

### **UNITED STATES**

International Decision Systems  
1500 IDS Center  
80 South 8th Street  
Minneapolis, MN 55402 USA  
Tel: +1.612.851.3200  
Fax: +1.612.851.3207  
Inquiries: [info@idsgrp.com](mailto:info@idsgrp.com)

### **EUROPE**

IDS Limited  
Norton House  
1 Stewart Road  
Basingstoke  
Hampshire RG24 8NF  
United Kingdom  
Tel: +44.1256.302000  
Fax: +44.1256.302005  
Inquiries: [info@idsgrp.com](mailto:info@idsgrp.com)

### **AUSTRALIA**

IDS Pty Ltd  
Level 5  
234 George Street  
Sydney, NSW 2000  
Australia  
Tel: +61.02.9247.9166  
Fax: +61.02.9247.9212  
Inquiries: [info@idsgrp.com](mailto:info@idsgrp.com)

### **ASIA/PACIFIC**

International Decision Systems  
9 Temasek Boulevard  
#15-02A Suntec Tower Two  
Singapore 038989  
Tel: +65.6333.9866  
Fax: +65.6333.9877  
Inquiries: [info@idsgrp.com](mailto:info@idsgrp.com)

IDS Software Solutions (India) Private Ltd.  
6th Floor, Tower D, Corporate Block  
Diamond District, Airport Road  
Bangalore, 560 008 India  
Tel: +91.80.55150910  
Inquiries: [idsindia@idsgrp.com](mailto:idsindia@idsgrp.com)