

WHITE PAPER



**SEVEN SUCCESS FACTORS
FOR EXTRAORDINARY EXTRANETS**



Extranets Deliver

The focus on managing customer relationships has translated into a focus on creating extranet applications to provide the needed customer-centric computing environment. Organizations may have several extranet applications, each defined by the intended audience. One might target end customers, another aimed at dealers, another for a different kind of agent or reseller.

As the leader in extranet application frameworks, Micro Focus has helped over 130 customers around the world build extranet applications to address three key areas of customer relationship management:

1. Dealer Systems that foster cooperation between companies and their dealers/distributors, providing the dealers and distributors with the information and processing they need to be successful.
2. Virtual Call Center applications that provide customers, agents, or distributors with the information they need, whenever they need it, wherever they are, at a much lower cost to the organization.
3. Customer Care sites that provide customers, dealers or resellers with self-serve and after-sales service information and another channel for managing the ongoing relationship with customers and partners.

While every extranet has some unique attributes and capabilities, there are some fundamental principles that are common to extraordinary extranets:

- They Speak User
- All Applications Share the Same Information
- Don't Just Protect Investments - Leverage Them
- Invest in Adding Value
- The Only Constant is Change
- Reliability, Availability, Scalability
- Security Matters

1. They Speak User

The goal of an extranet is to make your organization successful, which can only happen if the customers or business partners who use your extranet are also successful. Building an extraordinary win-win extranet requires an extraordinary user experience, which in turn requires a complete understanding of the user's role, the tasks they perform, and the mindset they bring to the extranet.

Historically, application user interfaces were designed to optimize the back-end processing or to make the programming easier. Today,

a good extranet liberates users from the limitations and tyranny of back-end systems designed for a different audience at a time when the computing world was ruled by very different economics. Extranet users don't care about the architecture and limitations of the back-end systems, nor should they. They simply want and need information and processing that helps create a successful relationship with your company, whether they are your customers, distributors, or employees.

There are two parts to an extranet user interface that must mesh together in order to create an extraordinary user experience. First, and most important, is that the function and flow of the extranet application must match the way the user thinks about the task, making the application easy and natural to understand and to use. This is what is meant by "speaking user" – dealer systems should speak "dealer," insurance agent extranets should speak "insurance agent," and so on. The second element is the way information is presented using text, graphics, buttons, lists, and other graphical elements – all of which can greatly enhance the usability of a well-designed extranet application.

An extranet application development environment that offers data abstraction capabilities is critical for building user-oriented extranets. Allowing extranet developers to work with an abstract representation of enterprise information means that all of the data becomes equally usable, and gives the developer the freedom to design the extranet application based on user needs rather than on the restrictions imposed by back-end applications and data sources. Removing the onerous requirement to spend large amounts of time and effort learning and wrestling with the intricacies of back-end enterprise systems also makes extranet developers much more productive.

Data abstraction essentially separates application business logic from the data logic. Similarly, an extranet application development environment can separate presentation logic from the business logic. Separating the functionality (semantics) of the application from its presentation (syntax) provides the flexibility to reuse data logic and business logic components in different applications. The same underlying extranet application can have specialized presentations for different audiences of users.

2. All Applications Share the Same Information

Extraordinary extranets are not stand-alone entities isolated from other business processes. Instead they are fully integrated with all other customer interaction systems. This means that if a customer uses the extranet application to place an order and five minutes later calls the toll-free number to ask a question about that order, it is imperative that the call center representative be able to see the just-placed order. It is disconcerting for customers when different parts of your organization appear to have contradictory information.

Making up-to-date information available immediately means no overnight database updates, and no “rip-and-read” implementations whereby extranet orders simply get printed out or faxed to a clerk who manually re-keys the information into the “real” order entry system. Extraordinary extranets are integrated with existing back-end systems to provide access and update of customer information in real time. Some companies are going a step further and making the integrated extranet application the primary internal application for dealing with customer information. Call center representatives, salespeople, and others who interact with customers also use the extranet application to retrieve and update information, thus ensuring that employees and extranet users see the same information.

This integrated approach delivers four key business benefits:

1. Users will not experience frustrations or errors from mismatched information. This helps to prevent your extranet from being the reason that customers look elsewhere.
2. Employees and customers work from the same page, giving your employees the information they need to provide superior customer service.
3. Information about past transactions and current requirements enables call center representatives and salespeople to cross-sell or up-sell additional products or services.
4. Eliminates the cost of creating, maintaining, and repairing separate versions of information for different applications.

3. Don't Just Protect Investments - Leverage Them

When organizations make a major investment to adopt a particular technology or architecture, their natural inclination is to ensure that all subsequent choices support the original decision – or at least don't work at cross purposes to it. In many cases, this approach makes good business sense. However, when a major technology discontinuity, such as the introduction of the PC or the Internet, is added into the equation, the benefits of the new technology can either be ignored or perverted by trying to retrofit new capabilities in the name of “protecting our investment.”

Organizations adopting extranet technology have three fundamental approaches for integrating the extranet with existing back-end systems and business processes:

1. Implement a stand-alone extranet with no realtime integration at all. Any integration is via periodic database refreshes or using a manual “rip-and-read” process whereby extranet transactions are re-keyed into the existing applications. This approach not only becomes very expensive as the volume of customer interactions increases, but the lack of integration and the resulting data synchronization problems can lead to missed opportunities and customer service disasters.

2. Protect the existing investment by investing even more time and effort into adapting the existing systems to provide an extranet interface. Typically, efforts to bridge an existing island of information out to an extranet merely extend the limitations of the existing employee-facing application out to the Web user. The extranet application ends up speaking “mainframe application” instead of speaking “user.” In addition, this approach doesn't offer any integration between one island of application information and another.

3. Use a three-tier approach which inserts an adaptive layer between the extranet users and the back-end enterprise systems. The server tier in the middle becomes a natural platform for harvesting information and functionality from multiple backend systems, weaving them into new customer-oriented applications, which are then presented to the extranet users.

In a competitive business environment, especially on the Internet, companies cannot afford the luxury of re-implementing business processes just to run in a new technology environment. The focus of the extranet application must be on the needs of the users, not the technology. If applications that implement a given business process already exist, then the organization must find a way to quickly leverage them, in order to take the extranet application to market as quickly as possible. Extranet application frameworks are very useful tools for building these kinds of extranets. The framework provides the necessary structure and “plumbing” for defining connections to back-end systems, creating data abstractions which factor out the useful data, applying new business logic, and delivering the results into the environment where the user experience is created. In a competitive world, decisions about customer-centric applications, i.e., extranets, cannot be at the mercy of past decisions about data-centric application implementations.

4. Invest in Adding Value

Extraordinary extranets know exactly where they add value for users and focus all of their development resources on maximizing that value. The dominant value of most extranets is the integration of multiple discrete business processes into a single, seamless, customer-oriented experience. This requires the exchange of information between the extranet application and existing business systems, and establishment of the relationships between the various elements of information. However, many so-called Web application tools do not work at all at the information level. They are designed for exchanging data formats at a programmer level. They lack any appropriate high level language for describing information relationships and therefore force everything to be described in low-level programming terms. This is like forcing a sportscaster to describe “Tiger Woods hit a high seven-iron at the pin” as “By exerting coordinated, well-rehearsed muscular patterns to swing the metal plate with a 35 degree angle to the horizontal around an angular momentum, Mr. Woods struck the dimpled spherical surface at rest with sufficient kinetic energy and at

the exact angle to transfer some of the kinetic energy to the spherical object such that its trajectory, given the prevailing atmospheric conditions and the effects of gravity, approached the negative cylinder in the surface of the earth.”

Programmers talk to computers – and skilled programming resources are scarce, especially those skilled in “hot” technologies such as Java programming. To optimize their time and efforts, we continually create higher level abstractions to help them cope with low-level complexity.

In contrast, business analysts talk to customers. To create extranet applications that speak to the needs of customers (users of all kinds), business analysts need application tools and abstractions that work at the information level – “rules” that describe how elements of information relate to each other, how information elements are transformed by business logic, and how applications are assembled from these information rules.

Extranet application frameworks enable people with these complementary skills to work together to build an extraordinary extranet. The framework provides high-level information tools that allow business analysts to work with information in a way that maps more easily to the underlying real-life business processes being implemented, but without requiring the analyst to become a programmer. Similarly, the application framework allows skilled programmers to focus on writing components to execute complex business logic by freeing them from the repetitive, mundane details of managing the interactions between multiple components. This allows both groups to focus on what they do best, and on what adds the most value to the extranet – the user application.

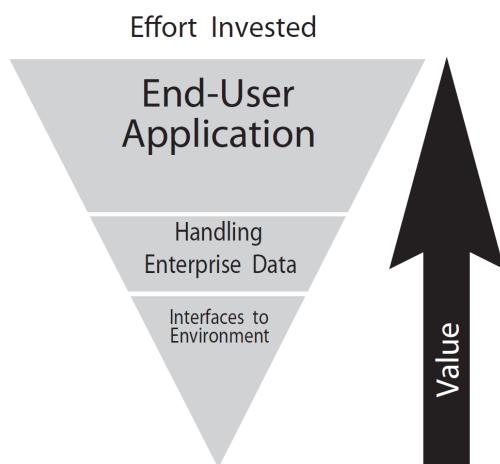


Figure 1: An extranet application framework frees developers from the low-level details and allows them to focus on the user application where they add the most value.

5. The Only Constant is Change

Because business changes, information needs change. Whether through cost reduction efforts, mergers, acquisitions, e-business initiatives, or the Next Big Thing, different kinds of information will be needed by different users in different places from what is available today.

Changing information needs shouldn’t come as a surprise, and therefore extraordinary extranets are built to readily adapt to changing needs. A crucial advantage of using a three-tier architecture with the adaptive layer in the middle is that it allows the extranet to adapt to changes on one side without changing the other. Whether it is the back-end data logic or the front-end presentation logic that needs to be modified, the extranet business logic in the middle can be adapted appropriately.

Making changes can be expensive and timeconsuming. Extraordinary extranets rely heavily on technologies such as component-based development that foster reusability. By encapsulating an element of information or business logic as a component, it can be readily reused by new components or new applications.

As its name suggests, an extranet application framework can provide a common framework for adding new capabilities:

- New connectors for providing access to new data sources, e.g., the back-end systems of a bank that was just acquired
- New functionality (business logic), that is instantly available to every other application and component within the framework, e.g., a component for processing credit card payments
- New delivery points for supplying all existing information to a new application environment, e.g., as XML output for use by business partner applications
- New information elements that build upon previous versions, e.g., a new tax calculation for e-commerce transactions

Change can be a double-edged sword. For extranets cast in stone, change becomes a millstone that holds back the organization. For extranets with adaptability built in, change creates new business opportunities.

6. Reliability, Availability, Scalability

Extranet applications are enterprise applications because they involve customers and business partners. They should demand the same, or even more, attention to operational issues.

Reliability

Extranet applications built using an extranet application framework can have significantly higher reliability because:

- They make extensive use of dynamic, reusable code. When a defect is fixed, it is fixed everywhere.
- The framework code and the “plumbing” between components is not visible to programmers and thus is not vulnerable to tinkering. This is in sharp contrast to application builders which rely on wizards and other forms of code generators, which invite clever programmers to “improve” the code.
- All of the multi-tasking, multi-user code is within the framework. It is done right once and is used by every extranet application.
- The more extranet development done at the information level, the less developer-written code, resulting in fewer lines of code where defects may be introduced.
- High-level information programming may have defects which produce incorrect results, but is less likely to have defects that bring down the application or the run-time environment.
- When a dynamically-loaded component fails, usually the failure is confined to the component and doesn’t bring down other users or other applications running in the same environment.

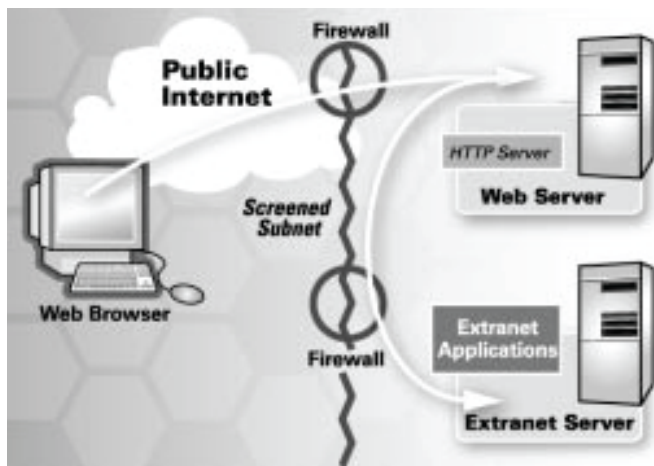


Figure 2: Fire walls and other secure networking technology can make Extranets secure.

Availability

There are many factors that determine overall availability of an application, many of which are related to the operating system environment outside the control of the application environment. However, extranet application frameworks do offer some architectural features that improve availability, specifically:

- With applications built from components that are dynamically executed only upon demand, if one application component fails,

other parts of the application may continue to run. For example, a bad connector to one data source may not disrupt other parts of the application.

- Standard Web server techniques such as load balancing and hot failover, are used with extranet applications to improve their availability.

Scalability

Extranet application frameworks can provide superior scalability for extranet applications:

- Component-based applications load components only when they are executed, and therefore consume resources only when actually needed. The corollary is that as soon as a component finishes executing, its resources are freed and become available again.
- The framework automatically provides multi-user and multi-threading support, which enables optimal scalability at the operating system level.

7. Security Matters

Concerns about security have consistently emerged as the primary obstacle preventing both consumers and businesses from using e-business technology. Like any other risk, extranet security can be measured and managed to allow informed decisions about the cost of potential security breaches compared to the cost to protect against those threats.

Extranet security is non-trivial, but achievable. Firewalls and other secure networking technology can make extranets secure.

Extraordinary by Design

Extraordinary extranets don’t just happen, they come about when organizations pay attention to all 360 degrees of issues: content, integration, development, and operation.

As a key interface to customers, an extranet can become the centerpiece of an entire customer-centric computing environment. It requires organizational will to continually focus on user needs and allow customer information requirements to dictate application design instead of the applications being limited by back-end restrictions.

The rewards of extranets can be very high. Darby Pharmaceuticals implemented an online catalog application allowing doctors to order medical supplies over the Internet, and saw sales volume climb almost immediately. An insurance company is in the process of creating an extranet application to complement their existing call center, which currently handles over 66 million interactions per year at an average cost of \$6 per call. Given that 17 percent of their six million policy

holders are regular Web surfers, if a significant portion of their calls shift to extranet interactions, at an average cost of \$0.40 per interaction, then tremendous cost savings are possible.

With that kind of high customer visibility, that level of revenue growth or cost savings, it's worth building the right kind of extranet, the right way, using the right tools. OnWeb® from Micro Focus is the leading extranet application framework and is the right tool to help your organization seize the extranet advantage.

About Micro Focus

Micro Focus, a member of the FTSE 250, provides innovative software that allows companies to dramatically improve the business value of their enterprise applications. Micro Focus Enterprise Application Modernization and Management software enables customers' business applications to respond rapidly to market changes and embrace modern architectures with reduced cost and risk.

For additional information please visit: www.microfocus.com

Copyright © Micro Focus (IP) Limited 2009. All rights reserved. The software and information contained here in are proprietary to, and comprise valuable trade secrets of, Micro Focus (IP) Limited, which intends to preserve as trade secrets such software and information. This software is an unpublished copyright of Micro Focus and may not be used, copied, transmitted, or stored in any manner. This software and information or any other copies thereof may not be provided or otherwise made available to any other person. sWPSSFA1209