

# Micro Focus

## *Legacy Transformation and Integration*

---

**WHITE PAPER**

---

## **Abstract**

This paper presents the concept of Legacy Transformation and Integration with a focus on Legacy Application Integration. Legacy Application Integration is a fast and low-risk method that not only reduces costs, but also increases usability and efficiency. This discussion details the features of Micro Focus EnterpriseLink Integration Server and EnterpriseLink Component Generator as vehicles to integrate legacy systems with other enterprise applications.

# CONTENTS

<b>Evolutionary process of moving Legacy systems to the Web .....</b>	<b>5</b>
<b>Micro Focus EnterpriseLink™ .....</b>	<b>6</b>
EnterpriseLink Integration Server Overview .....	6
Component Generator Overview.....	6
Comparison .....	7
<b>EnterpriseLink Integration Server .....</b>	<b>7</b>
EnterpriseLink 3270/5250 (LU2) to HTTP .....	7
Dynamic generation capabilities explained .....	8
Adding value to your existing user interface.....	9
Development and testing .....	9
Support for 3 <sup>rd</sup> party-editing tools .....	9
When host screens change .....	10
EnterpriseLink Cluster Server capabilities and purposes.....	10
EnterpriseLink as a Security Policy Manager (SPM) .....	11
SPM Components.....	11
Security Process Flow .....	11
External Security Interrogation.....	13
<b>EnterpriseLink Component Generator .....</b>	<b>14</b>
Turning CICS Applications into Components .....	14
Creating and Modifying Components .....	14
<b>AssetMiner and Componentization with Net Express .....</b>	<b>16</b>
Asset Mining and COBOL Reuse.....	16
Use of Net Express to create EJB's and or COMs for use from other languages .....	17
<b>Conclusion .....</b>	<b>17</b>

# Evolutionary process of moving Legacy systems to the Web

The ability to transform and integrate legacy systems for new technical and marketplace opportunities differentiates an enterprise from its competition. To achieve this goal, organizations must leverage their existing enterprise legacy assets and put them to work as re-usable components. Equally important, they must do it in an evolutionary manner to minimize any negative impacts to current business processes.

Legacy Transformation and Integration (LTI) is a Micro Focus solution for e-enablement of legacy systems. Legacy Transformation and Integration helps provide solution flexibility by making the most of your legacy assets. Legacy Transformation and Integration is a collection of methods to e-enable your company's legacy applications and advance them into component re-use architecture, ensuring a solid legacy foundation for Web Services support and future growth.

Legacy applications can be integrated with the Web using one of three predominant architectures. These architectures should not be considered a destination, but evolutionary steps toward the goal of re-use and solution flexibility. Figure 1. 1) Web-ification consists of rendering a text-based terminal application in a web form, leaving the presentation relatively unchanged. Web-ification presents the application in text mode in a browser or in a default web appearance format. 2) Legacy Integration consolidates multiple applications into a single interface, providing better control of host connections, security, availability and management with the use of a middleware integration server. Integration servers can also pave the way for component integration by providing component interfaces as a bridge to the next stepping-stone, Component Integration. 3) Component Integration is a complete transformation of an application and supports composite application construction from re-usable business logic components that can make up a self-contained application or an internal or external web service.

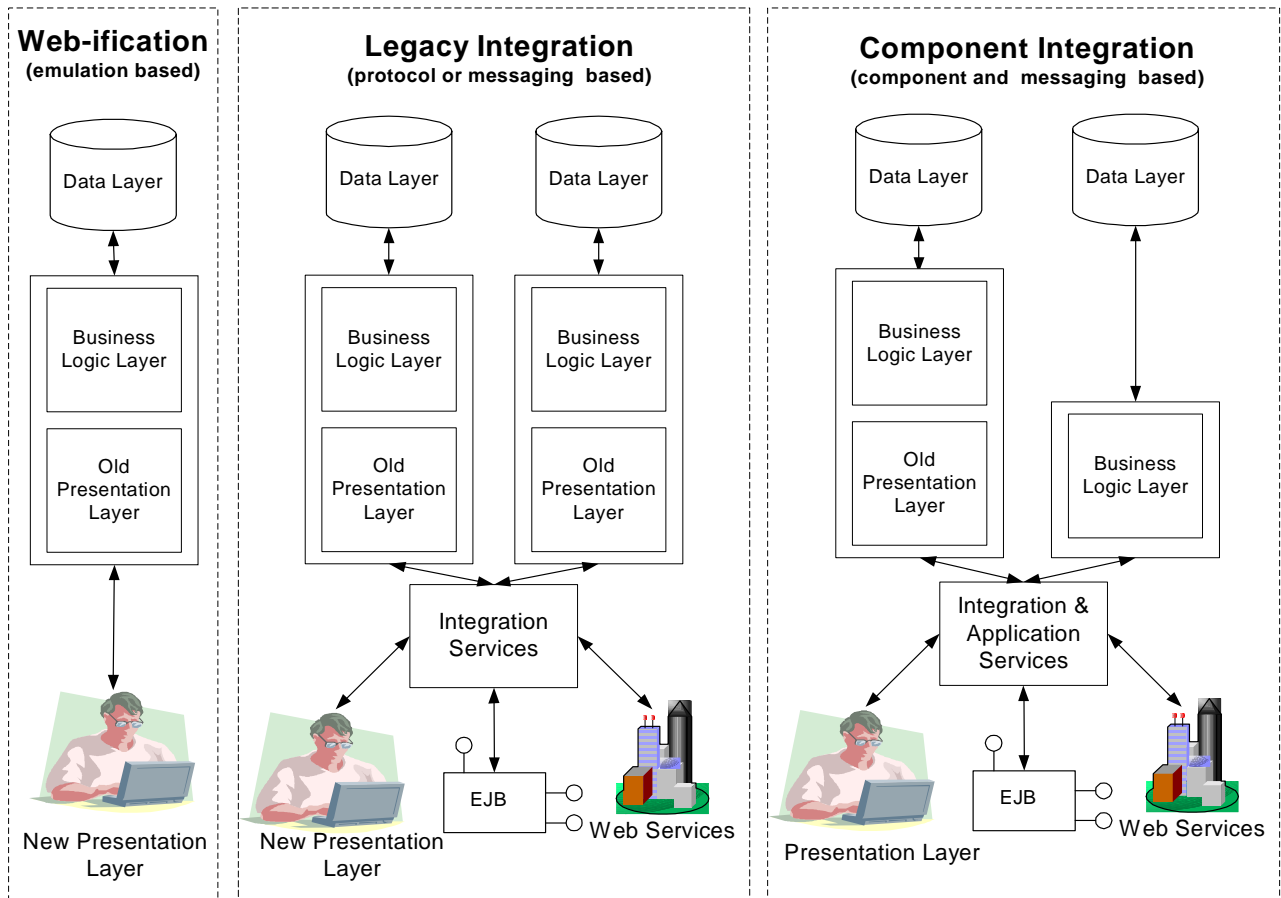


Figure 1. - Legacy Transformation and Integration

# Micro Focus EnterpriseLink™

Micro Focus EnterpriseLink provides a fast path to e-business development by providing the capabilities for application integration and transformation of Legacy Systems using HTML, XML, COM and Java Beans. Implementations are fast and the environment is easy to maintain, resulting in a low cost of ownership while delivering high performance, reliability and scalability.

As a part of the overall Legacy Transformation and Integration Solution, EnterpriseLink is sold as a single solution suite consisting of two products: 1) EnterpriseLink Integration Server and 2) Component Generator. They are separate products that are sold together, both providing the ability to integrate and extend mainframe applications to e-business and the Web without undertaking a major programming effort and without making changes to existing production systems. By taking advantage of direct connection to the legacy CICS systems via the IBM CICS Bridge, Component Generator does not use an integration server. However, as its name implies, EnterpriseLink Integration Server uses a middleware server and does not rely on the IBM CICS Bridge, allowing it to work with any application that has an IBM 3270 or 5250 screens. Following is a short overview of each technology.

## EnterpriseLink Integration Server Overview

EnterpriseLink Integration Server architecture is an execution unit that is deployed as an application server and can be viewed as a "data pump" that works in co-operation with existing popular web servers (such as WebSphere, iPlanet/Netscape & IIS). With EnterpriseLink Integration Server, the central point of integration is the Integration Server. It works in this way:

- Once an interaction with a mainframe (or AS/400) is started, the EnterpriseLink Integration Server responds and manages the connection to the mainframe
- Using customizable inference, it performs pattern matching to determine what the response from the server is (i.e., what screen) and what actions to take
- It invokes processing or collections of processes (based on event processing that is customized) to transform the results of the host screen into a format for presentation to a client (usually as HTML, but also as a COM interface or defined XML.)

EnterpriseLink accelerates Web enablement by using the EnterpriseLink Loader to capture the screen workflow of the existing application and creating an object repository that reflects that application – both screen format & sequencing. EnterpriseLink Builder is a GUI tool for helping the user create the integration logic that will run on EnterpriseLink Server at runtime. The GUI tool handles data mapping, function calling and events management. Additional business processes and adapters can be created using client or server side scripting. These routines can be saved in the repository as objects and can be re-used. A drag and drop metaphor is used to place these routines/objects on pages. The advantages of running a separate server are:

- No software to install on the mainframe
- Co-exists on the web server in the middle tier and provides enterprise-computing capabilities such as security and clustering
- Serves as the point of integration for a variety of backend systems (e.g. CICS, IMS, ISPF - anything that presents a 3270 or 5250 terminal)
- Provides component interfaces using XML

## Component Generator Overview

Component Generator creates business components deployed within industry-leading application servers (such as IBM WebSphere, BEA WebLogic or any app server that supports EJB and/or COM). With EnterpriseLink Component Generator, the focus is on:

- Creating industry-standard components of complete business processes from sequences of existing CICS transactions
- Exposing larger grained units of business function for integration with the Web and other applications via a variety of technologies (XML, EJB, JavaBeans and COM)
- component and middleware code generation for selected target platforms

The advantage of using Component Generator is the efficiency gained by creating the reusable units of business function on the mainframe (it only takes one trip to the mainframe to execute a sequence of transactions) and the creation of a business interface that abstracts above the original CICS/3270 technology. Once that packaging is done (called an e-Biz Transaction), it can be readily exposed for integration to meet new business needs via several types of technology.

There are two points of integration: primarily on the mainframe and secondarily on a web or application server. A tailored GUI tool (the e-Biz painter) assists the user in creating the integration logic for the mainframe. User interface development and component assembly is done using standard GUI development tools such as IBM's Visual Age for Java or Web Gain's Visual Café, and the generated components are deployed on standard web and application servers such as IBM's WebSphere or BEA's WebLogic.

## Comparison

Both EnterpriseLink Component Generator and EnterpriseLink Integration Server integrate legacy applications. The major difference is where the integration occurs: primarily on the mainframe for EnterpriseLink Component Generator and on a separate EnterpriseLink Server for EnterpriseLink Integration Server. Both have tools to create the integration logic for applications. Given that both technologies enable mainframe applications to be re-used, there are some key differences:

Description	EnterpriseLink	EnterpriseLink CG
Extend any 3270/5250 app.	Yes	CICS only
2-tier support	No, require middleware	Yes via MQ Series and WebSphere 390
Caching	Yes	Yes via MQ Series
Clustering support	Yes	Same as Mainframe
COM support	Yes	Yes
GUI Builder	Yes	No
EJB Support	Yes, via JAXP API	Yes
Open technology	Yes via COM, XML and API's	Yes via COM, XML, EJB and Java Beans
Requires software on mainframe	No	Yes via e-Biz transaction
Security	Yes	Yes
Transaction-based	Yes	Yes
Wireless Support	Yes – WML	Yes – WML and J2ME
XML Support	Yes	Yes

## EnterpriseLink Integration Server

EnterpriseLink Integration Server uses HTML, XML, or a COM or API interface to integrate legacy applications with Web or enterprise applications. The heart of this method is the legacy-to-Web integration server or what can sometimes be called a "presentation layer protocol router." The legacy-to-Web integration server converts mainframe screen block protocol data streams (LU2) to a web protocol (HTTP) with a tag-based mark up language (HTML or XML) that is then transmitted to a web browser for display via a direct connection to the web server. Since the integration server is middleware, it can control all LU2 and HTTP connections and provide enterprise-computing capabilities such as security and clustering to enhance security and scalability.

This is different from screen scraping because it occurs pre-3270 emulation and deals directly with the protocol data stream, allowing significantly better control and increased flexibility of the translated screen objects and events. This allows for features like: RACF and ACF II to LDAP security mapping, server clustering, LU2 to PDF translation, LU2 to XML translation, LU2 to WML/WAP translation, direct data host connectivity via MERANT SequeLink™ and Legacy Server.

In addition, if re-platforming is in your future, you can accommodate it without disruption to your user community once you have a Web-enabled legacy application using EnterpriseLink. You can easily, over time (evolution), transform the mainframe code to run on UNIX or NT/2000 as web components (COM or Enterprise Java Beans) using Micro Focus Net Express™ and Micro Focus Server Express™.

### EnterpriseLink 3270/5250 (LU2) to HTTP

So, how does this all work? It starts with EnterpriseLink Loader recording the mainframe screen interactions (workflow) and creating an application-transitions database – the EnterpriseLink repository. This database contains all items sent to a 3270/5370 terminal or terminal emulator that are within the LU2 block mode protocol. An LU2 record is the internal representation of your screen once it leaves the mainframe.

The repository contains all transitions, relations and derivations of LU2 records (screens) that make up your user legacy application experience, thus allowing flexibility to provide services such as caching (improved performance) and pattern matching to deal with host screen changes.

The EnterpriseLink server uses this repository to map LU2 protocol records to HTTP/XML messages. This is a significant differentiator for EnterpriseLink and a benefit for users, since it does not use terminal emulator technology and is not limited by that. It doesn't use the output of an emulator; it uses the output of the mainframe, effectively cutting out the middleman.

Also, since the protocol transformation takes place on the server, you can take advantage of server-based resources and capabilities. This allows you to better control the connection to the mainframe so you can supply better security and availability. This differs from server clustering of emulators that simply move the control of the emulators to the server. EnterpriseLink's server process manages a collection of LU2 connections, not a collection of emulators.

Additionally, as depicted in Figure 2, EnterpriseLink Integration Server allows integration of direct data calls, server- and client-side scripting and component connection. All this adds up to flexibility to evolve your legacy applications.

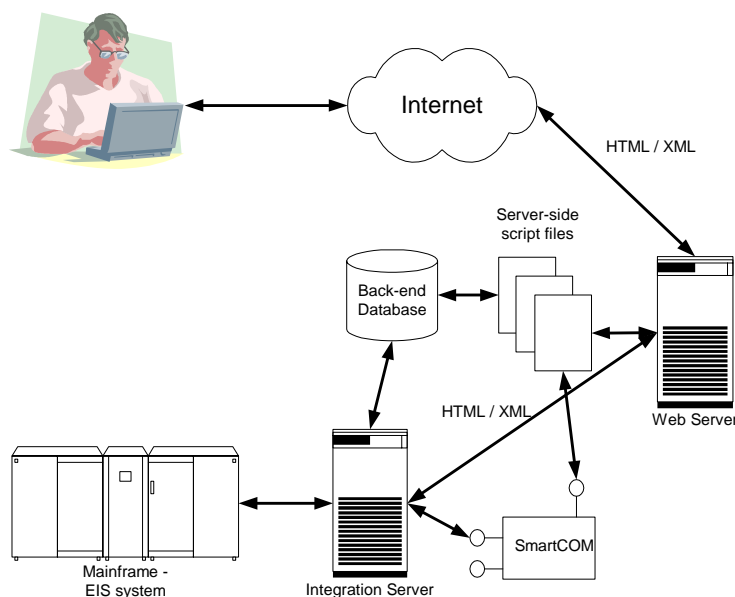


Figure 2. EnterpriseLink Integration Server

## Dynamic generation capabilities explained

EnterpriseLink creates XML, HTML and PDF representation of the mainframe screens that can be used immediately to access the mainframe system from a web browser. This can be in default translation or you can enhance it using our Authoring tool, EnterpriseLink Builder. With EnterpriseLink Builder, output is dynamically produced to supply the requesting browser with the correct information requested. **No Static HTML code** needs to be stored and maintained for the various browsers since all pages are generated on demand.

EnterpriseLink generates the following data output options:

### HTML

Generate HTML 1.0-3.2 based on user-specific browser capabilities.

### PDF

Macro pre-process Adobe PDF (portable document format) files for generation to Adobe Acrobat Reader browser plug-in. Updated support for Adobe 4.0, the first release capable of storing Japanese Shift-JIS data; server-side PDF file template is specified in the URL and indicates the file to be pre-processed with element names present on the page with their current values. This option is good for report generation.

**Note:** Adobe Acrobat, required to create new forms, is not included.

## **PDF**

Macro pre-process Adobe FDF (form definition format) files for generation to Adobe Acrobat Reader browser plug-in, which merges relatively small FDF data with client-installed PDF file formats resulting in reduced data transfer time to the browser minimal. This option is good for form generation.

## **XML**

Generate XML message representation of information on the Builder page in one of two formats: (1) all information presented in ELPAGE Document Type Definition (DTD) format or (2) processing a server-side XML file template (name specified in the URL) of arbitrary format; ELPAGE format is suitable for client-side processing by XSL style sheet, such as those supported by Internet Explorer 5.0.

## **COM**

Installation of the SmartCOM component on a client system allow for C++, Visual Basic, VBScript or other COM-compliant subsystems to access EnterpriseLink-defined web pages as COM objects; equivalents with the "web page world" are "component name" = "web page name," "element name" = "form object name," "method name" = "button name," etc.; installation of SmartCOM on a Microsoft Internet Information Server system allows for direct access of integrated EnterpriseLink data through Active Server Pages (ASP).

## **The advantage of dynamic page generation**

Development of multiple versions of web pages to accommodate all the browsers that might access your systems is not necessary; therefore you eliminate the effort and expense of maintenance and storage of multiple versions of the same page. Using the Integration Server and dynamic page generation approach makes it possible to keep your client as thin as possible. In addition, you use the integration server to centralize control of the host connections and security. This puts host connection and security activities where they should be (in the data center.) The Integration server also simplifies client development since you do not need an embedded emulator or fat (logic heavy) client. Finally, it provides a central location for XML generation for integration into web servers.

## **Adding value to your existing user interface**

EnterpriseLink Builder will help you create a great looking web site as a front end to your legacy system. Builder allows you to customize the web-enabled legacy screens beyond default translation. You can add radio buttons, pull down menus, selection lists, Java applets, CGI scripts, execution of client side and server side scripting, or additional HTML or XML to enhance your existing legacy system. By incorporating Java applets, Java scripts, Server Scripting, HTML and XML to your web-enabled legacy systems, you can add additional functionality and usability to your system without host changes. Also, a COM object representation of each screen is automatically created when you author a web page.

Builder does all this in a drag and drop development environment that is user friendly to facilitate rapid customization. However, if no customization is required, EnterpriseLink server offers out-of-the-box default translation and templates to customize default translation, allowing you to web-enable your whole application in a single day.

## **Java Connectivity**

Using the JAXP API from Sun (<http://www.sun.com/smi/Press/sunflash/2000-02/sunflash.20000229.1.html>), you can use the EnterpriseLink dynamically generated XML to drive JavaBeans or Servlets.

## **Development and testing**

You can develop and test your web-enabled legacy application without the mainframe. Using Micro Focus Mainframe Express (MFE) to provide a mainframe emulation environment, you can develop and test your web application without the overhead and logistical problems associated with testing on the mainframe. EnterpriseLink was designed to communicate with MFE as if it were a true mainframe.

## **Support for 3<sup>rd</sup> party-editing tools**

If you are using EnterpriseLink Integration Server, all HTML and XML are created on demand. Since no static markup language exists, you cannot use 3<sup>rd</sup>-party editing tools for dynamic creation of HTML or XML. However,

EnterpriseLink Integration Server has a COM component that can be used in Active Server Pages, and you can use JAXP for Servlets or Enterprise Java Beans programmatic connectivity. This allows for ultimate flexibility by using a component-based approach with server-side pages that drive the client page.

## When host screens change

Modification of presentation pages when a backend screen changes differ based on the web-enablement approach. Since Component Generator deals directly with the 3270 Bridge in CICS it does not care about any placement of elements on the CICS screen. Its only requirements are the names and offset in the copybook. However, if fields are added or removed, changes may be necessary based on the usage of the field and dependence on it. EnterpriseLink Integration Server handles this issue very differently. Since EnterpriseLink deals directly with the LU2 data stream it has great flexibility. When an LU2 record is sent for and received from the host by EnterpriseLink Integration Server it tries to match the record to a corresponding captured record in the EnterpriseLink database (repository) at a specific user-defined level of pattern matching. If it finds the record it then transforms it to a corresponding customized HTML page. If it can't find a match it will default to default translation. This prevents your system from becoming unavailable because of any unmanaged change. The definable pattern matching allows you to maintain customized HTML page generation for what you decide is the right level of screen change.

## EnterpriseLink Cluster Server capabilities and purposes

Cluster Server delivers 24x7 application availability with non-hierarchical clustering and no single point of failure. Additionally, it provides scaling to thousands of users and optimizes performance even with unpredictable spikes in user traffic. It provides load balancing transparently and fail-over for as many servers as needed, delivering 99.99% uptime for large-scale extranet and Internet applications. EnterpriseLink provides the following High Availability Services:

- Cluster Server – Provides high availability through fail-over
- License Server – Provides high availability while checking per session licensing restrictions
- Database Server – Utilizes native fail-over characteristics of database engines. Integration Server will reconnect as necessary if database restarts

As depicted in Figure 3, a single cluster server provides load balancing and fault tolerance for EnterpriseLink License and Security Services, and a second cluster server provides cluster fault tolerance for the primary cluster, providing 24x7 application availability and unlimited scalability by the inclusion of additional servers.

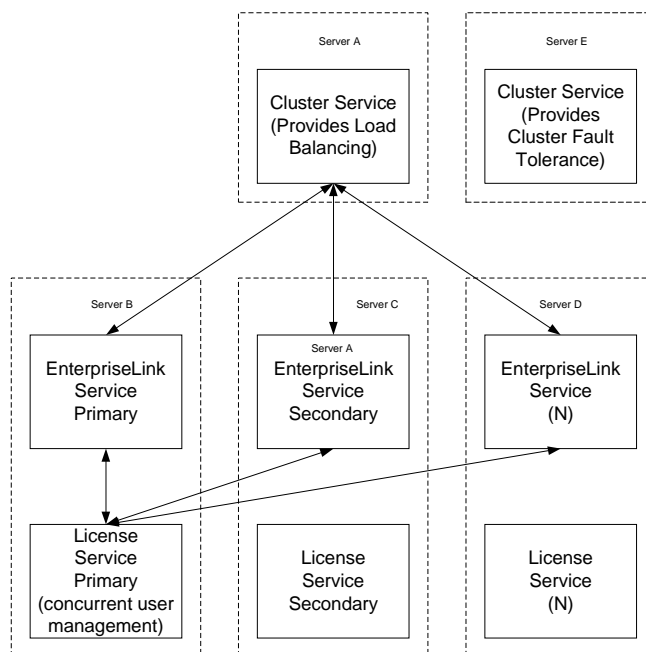


Figure 3. EnterpriseLink Cluster Server

## EnterpriseLink as a Security Policy Manager (SPM)

EnterpriseLink's Security Policy Manager controls the levels of security to which users connect interactive host applications to the Internet. EnterpriseLink's Security Policy Manager provides the following advantages:

- Leverage a host mainframe Security Subsystem, such as IBM's RACF or Computer Associates' ACF2 or TopSecret, to *authenticate* individuals before they access corporate resources. A single administration point (existing host process) is used to grant/revoke access privileges
- Provide a mechanism to discover the inherent security model that runs your company – that which is embedded into existing host applications. For instance, an insurance agent may view policies for his customers but not those of other agencies. Only complex business logic in host applications can supply this intricate knowledge – it cannot be represented as a series of user ID groupings
- Define one or more Security Objects that are made up from the security rules embedded in the applications. The Security Objects may *re-purpose* the security elements of the application so that, for instance, rather than a traditional Logon and Password, a Social Security Number and PIN combination are checked against the host application to initiate access
- Allow these discovered security rules to be interrogated by other programs, thus *authorizing* the person running that program to use it. Programs do not need to have an understanding of the security enforcement rules and can simply query the system to determine if the user can use it. Security information is delivered using Extensible Markup Language (XML) or through a simple API which interprets the XML data
- Applications that must be deployed to the Intranet, Extranet or Internet community can use existing EnterpriseLink features to bring those applications and data to the Web while automatically including the necessary security boundaries
- Optionally map to existing security mechanisms such as LDAP Servers, Windows NT Primary Domain Controller, Windows 2000 Active Directory Server or Public Key Infrastructure

## SPM Components

EnterpriseLink SPM consists of three phases: 1) EnterpriseLink Loader identifies the portions of the host application which define the security "hoops" through which users must jump; 2) EnterpriseLink Builder allows a security analyst to define the Security Objects, which will be queried by external programs to verify access; 3) EnterpriseLink Server delivers the security objects to the user/s requesting authentication or authorization. Additionally, EnterpriseLink Cluster Server can provide 100% uptime of the SPM as well as provide linear scaling across a wide range of computing devices.

## Security Process Flow

Typical access to enterprise data is through host applications embodied as interactive, dialog-based terminal sessions. A typical user may connect to a host screen and be asked to enter a Logon ID and Password. The host application starts after successfully being authenticated as shown in Figure 4.

The end user may wish to enter the account system, which is allowed only if the logon has been registered with the security subsystem. Otherwise, the user is denied global access to the accounting subsystem. After a series of question and answer sessions with the user (perhaps involving entering account numbers, policy numbers or other qualifying questions), the user will be granted access to the accounting data that he/she needs. This specific access is denied if any one of the questions is not answered properly. Most host systems will generate an audit trail of denied access. These embedded security checks are captured using EnterpriseLink Loader, a terminal emulator that learns the way an expert (the person using the accounting system) would normally access data.

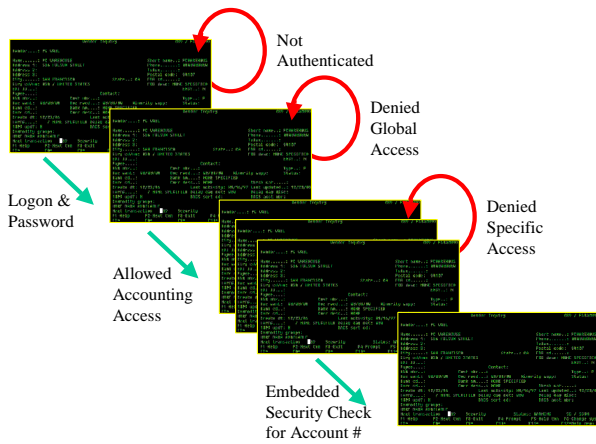


Figure 4. Authentication and Authorization embedded into host applications.



Figure 5. Three Security Objects, as displayed in their native (browser) form.

EnterpriseLink Builder is then used to develop one or more Security Objects, those that define certain levels of security that an end user may wish to attain. Figure 5 shows how security levels can manifest themselves in a web browser environment.

EnterpriseLink SPM security objects are mapped to one or more screens in EnterpriseLink Builder and can show one or more status levels as well as ask for additional information from which the next security level may be obtained. When using EnterpriseLink Builder, the author may include some additional presentation graphics or helpful hints on how to obtain proper access if such access has been denied.

EnterpriseLink Server acts as a proxy agent for the mainframe security subsystem and its applications by attempting to put the host application through its paces. When accessed through a web browser, as shown in Figure 6, EnterpriseLink Server drives the host application through the security checks providing information supplied by the browser user as it progresses. EnterpriseLink Server generates the next web page if, and only if, the security checks pass through the host applications.

Figure 6 shows that the client system accesses a web server (1). If security mapping is in force, steps (2) a-d are followed, as determined by the Internet security system that is active. Security mapping is a technique to automatically map from Digital IDs, or the public key infrastructure, to a host authentication system. Note that security mapping is only necessary if you have two competing security frameworks and wish to map them.

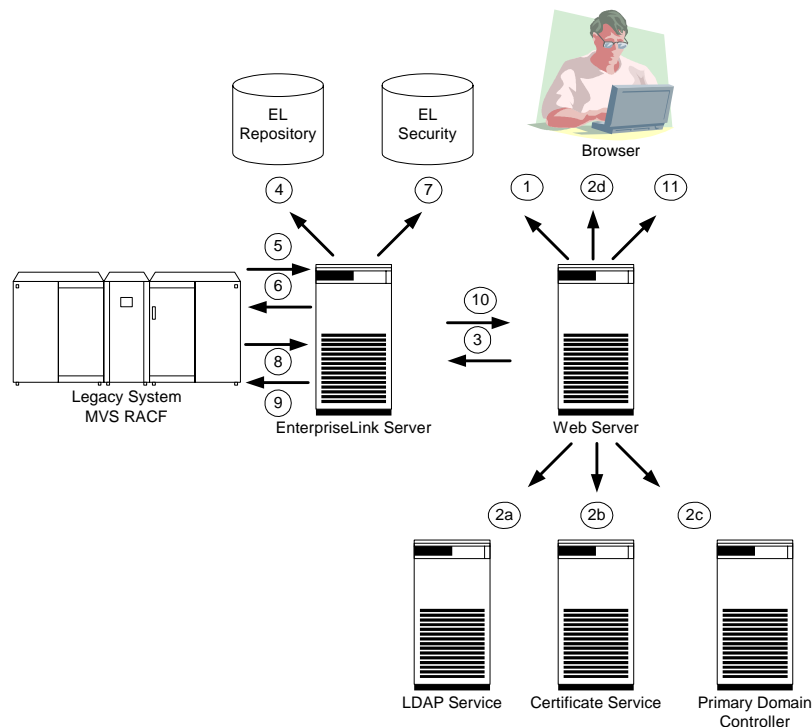


Figure 6. EnterpriseLink Server enforcing the SPM rules

EnterpriseLink Server will access the EnterpriseLink Repository (4) to determine which portions of the host application must be visited (under host security control) to attain the desired level. The Server communicates with the S/390 or AS/400 application (5,6,8,9), driving its state as necessary to reach the security object. For instance, if the browser user requests **SecurityLevel2** object (shown in Figure 5), which was built from the last screen in Figure 4, then all screens between the current state and the last will be visited. If the host application fails to authorize the current RACF user to see the information contained in that screen, the transition will fail and the application will fall back to a prior state – the logon state if the failure is severe enough. This would translate into the **SecurityLevel0** object being delivered when **SecurityLevel2** was requested.

### External Security Interrogation

External applications that need to use the authorization and authentication features of EnterpriseLink SPM query the EnterpriseLink SPM through one of two methods. Figure 7 indicates external applications (shown in red) communicating with EnterpriseLink SPM.

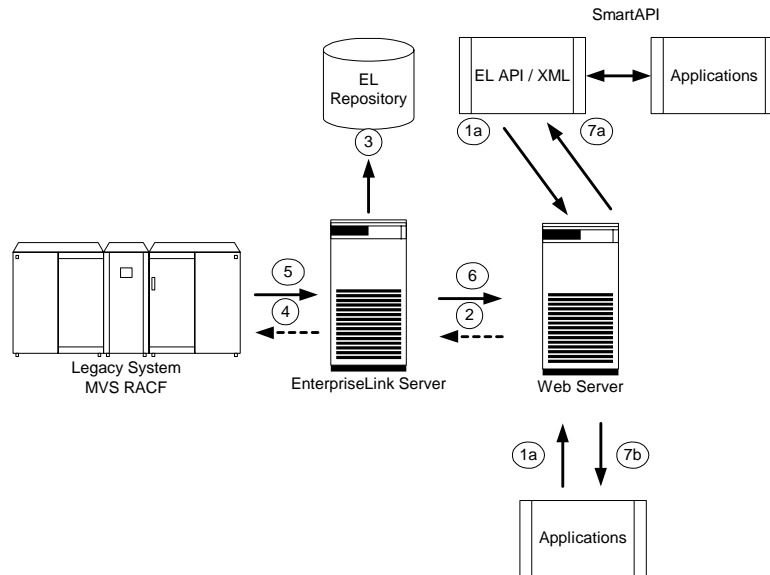


Figure 7. Programmatic access to EnterpriseLink SPM

In one case (1a, 7a), the host application communicates through SmartCOM. SmartCOM allows a Java client (e.g., applet, servlet, application) or VBScript client (e.g., Microsoft Active Server Page, Clarify, Seibol) to make a request for Security Objects and interrogate them when they are returned. SmartCOM relinquishes from the application its need to deal with the XML generated by EnterpriseLink SPM.

External applications, which must have low-level access to all information on a page, can process XML directly. This allows the same EnterpriseLink application to serve two masters: As a full-fledged web-to-host application and as a system that can be programmatically accessible to downstream Internet clients. F XML is generated from the **SecurityLevel1** Security Object.

**Note:** Some web-specific presentation information is contained in the XML. EnterpriseLink XML follows the ELPAGE Document Type Definition.

Requests for security objects are sent over HTTP protocol using the same request protocol utilized by a web browser. XML is returned over the same protocol and may be processed by the application through either a validating or non-validating XML parser, available from various sources.

These are some of the facilities of EnterpriseLink when acting in the Security Policy Manager role. When configured and used properly, EnterpriseLink SPM can turn existing host-based security subsystems and applications into a LAN-based security system. In this way, EnterpriseLink SPM acts as a *proxy* security system for a host-based security model, leveraging the processes and procedures that are already in place in most Global 2000 environments without additional investments and without incurring any additional administrative resources beyond what is already in place for host-based security systems.

## EnterpriseLink Component Generator

### Turning CICS Applications into Components

EnterpriseLink Component Generator allows you to capture complete business processes and automatically generate reusable, industry-standard components in the form of Java Beans, EJBs and COM for use throughout your enterprise. Using Component Generator, you can capture existing legacy application logic and user workflow, then automatically generate the necessary mainframe interface transaction code and component interface classes. The result is the creation of a Java or COM component interface that can drive mainframe transactions. Componentization allows deployment to open standards and accelerates development of new applications for the Web and e-business. This enables you to provide legacy-to-Web connections without the need for a web integration server or 3270 screen emulators. The component classes and the mainframe transactions communicate using the IBM MQ Series message queuing system. Figure 8 provides a graphic depiction of this method of legacy web-enablement. Component Generator also enables users to map existing mainframe data to vertical industry DTDs (Document Type Definition) and generate the required XML. This allows you to quickly and easily develop XML interfaces for exchanging information between disparate systems, improving efficiency and reducing operating expenses.

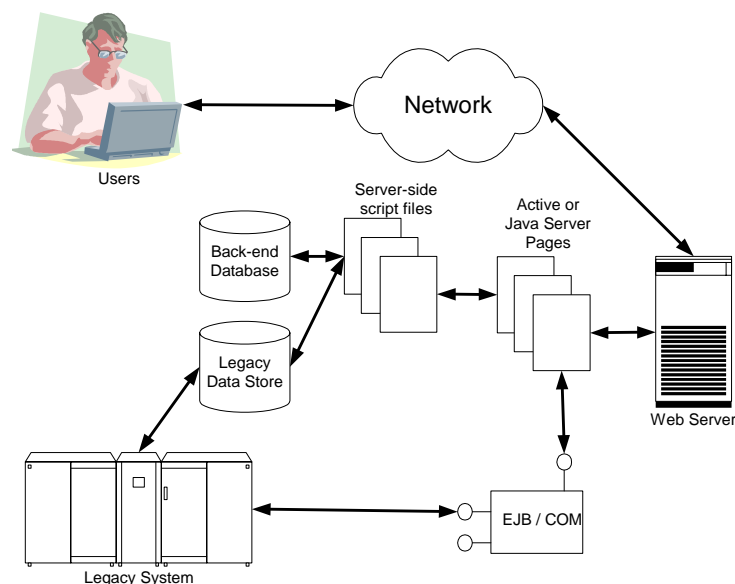


Figure 8. EnterpriseLink Component Generator

Because the component class, the mainframe transaction, and all MQ Series calls are generated for the developer, it eliminates months of manual development and allows the developer to focus on the customer's business problem and not the development of communication interfaces between the mainframe, MQ Series and Java. The result is reduced development cost.

Componentization is attractive because it minimizes the effort required to create business components that can be shared throughout the organization from existing mainframe applications. Even more importantly, it offers this ability without making any changes to mission-critical mainframe applications and is considered non-intrusive because all interface code is generated.

### Creating and Modifying Components

The captured workflows are used as a basis for creating components. However, these workflows represent the "success path" of the individual functions. Possible error conditions as well as restart/recovery operations must also be taken into account. The Component Generator IDE is a visual programming tool that allows developers to specify a captured workflow, create a business component based on the selected workflow and make any

necessary modifications before generating a component. Once a component is “primed” with the workflow, the IDE is used to:

### **Create Input-Output Fields**

In order to make use of a component once it is generated, you must define the inputs and outputs that will be used within the new user interface to present data to and receive data from the user. This is a simple drag-and-drop process through which you can specify the mainframe application’s I/O fields required to successfully complete the component’s task. Because this component layer is between the user and the mainframe application, you can select only the fields you wish the user to see or manipulate, in effect, hiding sensitive data from the users who may use this application, but who are not authorized to view or manipulate all of the data available through the legacy application. You can also create a generic “ReturnMessage” field into which you would move all error and informational messages rather than defining an output field for each user message on each legacy screen.

### **Add Error Processing**

As mentioned earlier, a component created using a captured workflow represents the success path of the workflow. Using the Component Generator IDE, you can add processing for error or unexpected situations. Because the Revolve knowledgebase provides you with a complete understanding of the relationships between the legacy application’s program and data objects, you are able to account for all possible outcomes of each legacy transaction executed within the workflow., ensuring all error situations are covered. This is also a simple drag-and-drop process in which the error message is returned from the mainframe application, through the component, to the new user interface. You can also specify a new, customized error message that more fully describes the error condition and is appropriate for the target user.

### **Create Supplemental Logic**

The Component Generator IDE also provides you with the ability to augment the legacy application’s existing program logic with additional calculations or supplemental conditional logic. For example, if a legacy screen returns marriage status information in the form of a numerical code that the traditional user knows and understands, 1 = single, 2 = married, etc., then, using the Component Generator IDE, you can easily add logic to evaluate this code and return any desired text to the user interface in its place so the target user will see “married” rather than the numerical code.

### **Specify Restart/Recovery Options**

One of the unique features of Component Generator is its Restart and Recovery support. Using the Revolve knowledgebase, Component Generator is aware of any possible database or file update operations available within the legacy transactions and warns developers of their presence. You then have the option of enabling Component Generator’s Restart/Recovery feature and may select Component Generator’s default Restart/Recovery processing, or specify an external program to call in the event of a system failure.

### **Generate Components**

When all modifications to the components are completed, it is time to generate components for deployment. This is a simple process of selecting one, several, or all components and pressing a button. Component Generator is capable of generating JavaBeans, Enterprise JavaBeans and COM components, as well as XML and WML.

### **CICS Program (e-Biz transactions)**

For each generated mid-tier component, there is a corresponding new CICS mainframe program generated. The generated component and CICS program work together to perform the function of the component. The CICS program does NOT duplicate the logic of the existing application programs. Instead, it collects the data it receives from the component then executes the existing legacy transactions in the proper sequence, supplying the necessary data as needed. When the process is complete, it sends the output data back to the component and the data are presented to the end user through the new user interface.

## IBM's MQ Series

The generated Java or COM component and CICS component communicate using IBM's MQ Series middleware. All the middleware code is also automatically generated, so again, there is no hand coding required.

## User Interface and Component Deployment

Before the new application can be deployed, a new user interface must be created using a standard Java IDE such as Visual Age for Java or Visual C++. After the user interface objects have been created, import the Component Generator generated component(s) into the new interface and tie the entry fields, list boxes, etc. to the corresponding interface fields of the generated component. Once completed, the user interface is moved to the mid-tier and registered with the web server and accessed via a URL within a browser.

## CICS Program Deployment

As mentioned earlier, for each generated component a corresponding generated CICS program works with the generated mid-tier component and the existing legacy application to complete the component's business function. These generated CICS programs are moved to and registered within the same CICS region in which the legacy application resides. Once this process is complete, the CICS program will be able to communicate directly with the existing legacy transactions within that region and will use MQ Series Message Queues to communicate with the generated component on the mid-tier.

---

## AssetMiner and Componentization with Net Express

Application Mining is a relatively new approach to leveraging the value in existing systems and reusing their functionality in new component-based architectures. If legacy systems continue to meet current business needs, they should be retained first, redesigned second, and rewritten only when there is no other alternative. The current software development crisis caused by the convergence of escalating software development costs, resources shortages, competitive pressure and new technology, necessitates an approach that extracts value from existing systems. Analysis of these facts leads us to conclude that to maximize productivity, software reuse is a preferred choice, especially if the fundamental business processes have not changed.

Consequently, at some point you may decide to re-platform some or all of the mainframe code. If you have web-enabled your legacy systems using EnterpriseLink Integration Server or Component Generator, this can be done piecemeal to minimize impact on your development resources and minimize the impact on your customers.

Using Micro Focus AssetMiner™, you can move your legacy business systems forward into new technology by extracting business rules in COBOL and deploying them as Java or COM components on an application server. You can do this in an evolutionary process replacing the screen-based web-enablement of EnterpriseLink Integration Server or Component Generator piece-by-piece, screen-by-screen.

Micro Focus Net Express® can be combined with AssetMiner to build components as Enterprise Java Beans or COM components using Net Express wizards. The results are COBOL-based components that have language and platform independent interfaces. This allows you to reuse your legacy system by transforming it into components so they can be integrated with new e-business or distributed environments.

## Asset Mining and COBOL Reuse

In order to reuse software, you must have both a structure for reuse and a source of reusable components. To do this you need to mine your existing application for business rules. Asset Mining is a process of extracting business rules from your legacy system. Four main deliverables or processes exist in an application mining approach to Legacy Transformation and Integration:

### Documentation

Application mining is the process of identifying and documenting business rules, creating a comprehensive business process repository.

## **Selective Extraction**

In a simple e-business application, the objective may be to componentize and expose key order entry functions to start selling over the Internet. Selective extraction of pre-determined functions is that aim.

## **Modularize Application**

Sometimes the existing application continues to serve a business need, but there may be significant maintenance costs if the application is frequently updated or changed. Replacement cannot be justified, but modularizing the application to limit the impact of changes and lower maintenance costs can offer tremendous benefit. This approach isolates and re-packages COBOL functionality one piece at a time as routine updates are scheduled.

## **COBOL Replacement**

To enable greater flexibility and responsiveness to changing business requirements, some mission-critical applications may be targeted for replacement using a rules-based development approach, or, for strategic architectural reasons, be targeted for rewriting in an object-oriented language such as Java. When using this approach, it is necessary to ensure that critical business requirements are retained and accommodated. In this case, XML output will be desired for tool-to-tool communication such as exporting to a rules-based application development tool, or a business process-modeling tool.

## **Strategic Re-architecting**

Strategic re-architecting componentizes functionality for reuse in a different architectural model such as COM or EJB. Long-term usage and preservation of transaction throughput, scalability, security, adaptability and integration with other processes must be considered. This approach represents a thorough top-to-bottom componentizing project, which will include development of new code and infrastructure and could include browser-based clients, message-oriented middleware or application server software.

If properly executed, application mining is an innovative method to cost-effectively extract useful components from existing systems and ensure that the process yields complete, accurate and timely results.

## **Use of Net Express to create EJB's and or COMs for use from other languages**

### **Support for Enterprise JavaBeans**

With Net Express Enterprise Java Bean (EJB) wrapping COBOL, assets can be extended to applications running under industry-standard application server environments. Class and Method Wizard enhancements support the wrapping of COBOL Objects as EJB's for deployment under application server environments such as IBM WebSphere and BEA WebLogic.

### **Support for COM+ Objects**

Net Express provides complete support for accessing OLE Automation Server from COBOL, allowing you to use desktop applications (Microsoft Word and Excel) to automate business functions such as report generation. You can also turn existing COBOL business logic into a COM object that can be accessed by other applications. A wizard automatically generates the interface needed for a COM object and all you need to do is drop in your existing code. By combining the COM and Java support within Net Express you can create Object COBOL classes that can be used either by Java as Java classes or through OLE automation as ActiveX classes.

---

## **Conclusion**

Economic rules are evolving. Markets are expanding and geographical reach has been extended because of the Web. But the changes are going well beyond that. Supply chains are integrated and value chains are evolving as the cost and risk of business transactions drop. Collaboration with outsiders to share benefits and risks has become a requirement of survival. The Internet has become one of catalysts and vehicles for this. However, the opportunities of the Internet also present significant problems for the legacy enterprise system.

In a perfect world, you would throw out the old, replacing them with all new, state-of-the-art, component-based systems. But the world is not perfect and we have seen the risk of whole scale re-writes or package replacements. You have a huge investment in legacy business systems. Even though they are using antiquated interfaces and are unable to provide simple ubiquitous access, they are critical to your business and contain immense value. Because of this, you must move these systems forward and make them accessible with a responsible approach.

This white paper has attempted to outline options for such an approach. We have discussed Micro Focus Solutions for Legacy Transformation and Integration and provided ideas for evolutionary technology steps to help you take advantage of technological advances. Component re-use architecture is an important way to help your organization have a solid legacy foundation for web services and future growth.

# Micro Focus

Choosing the right partner is as critical as choosing the right technology. As you move forward to meet these demands and the demands of your customers, Micro Focus continues to move forward with you as your strategic ally for legacy change. Unlike other e-business vendors, our approach starts with your enterprise legacy system and is designed to leverage, integrate and build upon your legacy assets. We have no computers or applications to sell. Our focus is to build the best tools to make your legacy system better. For more information on this approach or any of the supporting Micro Focus technologies, please contact your Micro Focus representative, or use the contact information listed.

Micro Focus  
United Kingdom  
Newbury . . . . . (+44) 1635 32 646

---

Copyright © 2001 Micro Focus. All Rights Reserved. Revolve is a registered trademark, and EnterpriseLink and Mainframe Express are trademarks of Micro Focus. Other trademarks are the property of their respective owners.

**1-877-772-4450**

**Fax 650-404-7217**

**[www.microfocus.com](http://www.microfocus.com)**