

eCube Systems  
*Enabling Enterprise Evolution*

# Evolving Legacy Enterprise Systems: A White Paper

Summer 2004  
© eCube Systems L.L.C.

# Table of Contents

<b>1. Introduction</b> .....	3
<i>Reducing Cost and Extending Value</i> .....	3
<i>Improve business and technical flexibility</i> .....	3
<b>3. Evolutionary Process - Renewal</b> .....	4
<i>Renew</i> .....	4
<i>NXTera Benefits</i> .....	4
<i>NXTera Implementation</i> .....	5
<i>Renewal Topology: Entera with NXTera 5</i> .....	6
<b>4. Evolutionary Process - Evolution</b> .....	7
Evolution Phase: NXTware Evolution Server .....	7
NXTware Evolution Server Platform.....	7
Development Environment .....	7
Integration Capabilities with JCA.....	8
Integration Capabilities with XML .....	8
NXTware Cross-Architecture Integration.....	8
Multi-language Binding.....	9
Emulation features .....	9
NXTware EV Implementation .....	9
Overlaying NXTware EV .....	9
NXTware EV Topology .....	10
Overlaying NXTware Evolution Server .....	10
NXTware EV: New Evolution Features.....	11
NXTware EV Interoperability, Emulation and Federation .....	11
NXTware Servlet Engine .....	11
NXTware EV Data Agents.....	12
<b>5. Phased Implementation of NXTware EV</b> .....	13
NXTware Evolution Components.....	13
<b>6. Eliminating Legacy Dependencies with NXTware Evolution Server</b> .....	13
<b>7. Operational Efficiency and Minimized Risk</b> .....	13
<b>8. Summary</b> .....	14

*This white paper outlines how companies can extract further value from their Entera investments by using eCube's Enterprise Evolution products to evolve their distributed business logic to contemporary architectures and models, such as J2EE, .NET, XML and Web services.*

## 1. Introduction

Today, a variety of business requirements are forcing IT managers to evaluate the modernization of legacy systems. Primary drivers include the need to: increase functionality and performance, comply with industry, business and government standards, and reduce operational costs and risk.

At the same time managers want to protect the value of existing technology investments. Too much time and money has been invested in these systems to discard them with every paradigm shift. Practicality requires the regular evaluation of legacy systems to decide if they're candidates for the tactical adoption of new technologies or systematic modernization.

Given these business realities, strategic legacy systems built on distributed systems such as Entera and DCE<sup>1</sup> should be considered possible modernization targets.

eCube System's modernization process evolves existing business logic, allowing legacy Entera and DCE-based application logic to operate within contemporary computing platforms.

The evolution of strategic enterprise applications makes it possible for companies to extend the value of existing business logic by defending it from what the Aberdeen Group calls "software hardening – or the increased inflexibility of legacy systems over time.

## 2. Evolutionary Goals

eCube's evolutionary process is designed so that companies can avoid costly re-engineering, re-testing and re-deployment of replacement systems, while reducing their dependency on legacy Entera technology.

### ***Reducing Cost and Extending Value***

Many companies have realized hundreds of thousands of dollars in saving by modernizing legacy business logic. In one case, a large medical insurance firm was able to save more than \$600,000 a year by eliminating the cost of operating legacy Entera applications with out-of-date databases on non-supported operating systems. An eCube sales representative would be happy to help you develop a model that outlines the cost benefits of evolving your legacy Entera system.

### ***Improve Business and Technical Flexibility***

The nature of business is dynamic. Change is a constant consideration for both business and IT managers. In order for systems to truly meet today's rapidly changing

---

<sup>1</sup> Distributed Computing Environment

business needs it is imperative that they be able to support contemporary platforms, languages and architectures on demand. Changes in business processes, regulation and operation often require that IT systems be able to rapidly accommodate new technologies. Today, such flexibility is best achieved using contemporary services-based architectures and structures such as XML, J2EE, .NET and SOAP/Web Services

Service-Oriented Architectures (SOA) enable logical programmatic business functions to provide business process capabilities as services, which can then be consumed by other business functions as needed. For example, a tax lookup service in an SOA could be used by other applications to calculate the proper amount of sales tax in a given state.

**NXTera** and **NXTware EV** are two Enterprise Evolution products from eCube Systems that address the dynamic nature of business and enable legacy applications to support contemporary operating systems and databases, while enabling legacy business logic to be accessed via .NET, J2EE, XML and the Web.

We'll outline in this paper the steps Entera users can take to evolve their existing applications and thus extend the value of their IT investments.

### 3. Evolutionary Process - Renewal

#### **Renew**

Old applications running on unsupported platforms are expensive to maintain and increase the complexity of data center management.

Therefore, the first stage of the evolutionary process is renovation where existing Entera applications are updated with NXTera so they can run on lower-cost contemporary hardware, operating systems and database platforms and reduce the complexity of data center management.

#### **NXTera Benefits**

NXTera is eCube's evolution of the Entera platform. Through extensive use of Entera and based on valuable customer feedback, eCube has identified those functionality, performance and support issues which most limit Entera's usability and extensibility. With this valuable information, eCube has re-engineered and improved targeted portions of the base Entera platform. These changes, implemented in NXTera, range from obvious improvements (ports to new platforms like Linux and bug fixes for nagging issues) to inventive enhancements, such as adding new data handling and memory management features. In some cases eCube has completely re-engineered some modules. Following is a review of the most important changes made to the legacy Entera code base in NXTera.

#### Data Handling and Manipulation

eCube's engineers completely re-engineered NXTera data access, transport and parsing modules for Oracle, Sybase and DB2. The result is more pro-formant and flexible data access that is ready to meet today's high throughput requirements. Some users have reported a 42% increase in data-centric processing.

## Memory Management

Overtime, application usage has a way of exceeding original expectations. Mergers, regulatory changes and increased usage place stress on legacy systems and their managers. NXtera has addressed these types of performance issues by re-engineering memory management architecture of the NXtera/Entera runtime. Independent testing has shown a 15 to 35% increase in performance.

The combined benefits of the work described above can be seen in these results derived NXtera /Entera comparative tests performed by multiple customers.

- 35% performance increase in memory intensive tasks
- 25% performance *i*ncrease when handling large datasets
- 45% performance increase when handling many small datasets

## ***NXtera Implementation***

NXtera is source code compatible with Entera. The result -- individual modules of an existing Entera infrastructure may be updated independently of others. For instance, the data access module of a system can be updated without changing the client or server runtimes. The same is true when a user decides to update the server runtime and data access portion and not change the client.

However, even when the client needs to be updated, NXtera provides simple source code upgrades that lower the cost of updating these systems. The system usually only requires an installation, recompilation and distribution.

Here is a list of newly supported platforms for the client, server and data implementations, as well as the source files that have been modified:

### Client Support and File Modifications

- Windows XP, XP Professional, Windows 2000
- NXtera/TCP 5.1 for Windows NT/2000/XP
- NXtera/Broker 5.1 for Windows NT/2000/XP
- PowerBuilder 8/9 and VB.NET
  - odepb90.txt, ode50.pbl, odet50.dll, nxt50NET.dll
- Java
  - nxt50client.jar

### Server Support and File Modifications

Latest Versions of: HP, SUN, IBM, LINUX, Windows:

- NXtera/TCP 5.1 (32-bit) for Solaris 8
- NXtera/TCP 5.1 (32-bit) for HP-UX 11.11 (AKA 11i)
- NXtera/TCP 5.1 (32-bit) for AIX 5.1/5.2

- NXTera/TCP 5.1 (64-bit) for HP-UX 11.11 (AKA 11i)
- NXTera/TCP 5.1 (64-bit) for Solaris 8
- NXTera/TCP 5.1 (64-bit) for AIX 5.2
- NXTera/Broker 5.1 (32-bit) for HP-UX 11.11 (AKA 11i)
- NXTera/Broker 5.1 for AIX 5.2
- NXTera/Broker 5.1 for Solaris 8

#### New Files:

- NXTrun: Starts up a single-process NXTera/NXTware Java server.
- NXTware: Starts up a NXTware application server instance (multiple Java business logic and emulation servers)
- NXTmake: Generates Java, VB.NET and C# client stubs, translated WSDL definition files, and Java server templates.
- NXTware.jar: NXTware runtime
- NXTconfig.xml: Default NXTware configuration file
- NXTdebug: NXTera Test GUI (compare to Entera RPCDebug)

#### Server Support and File Modifications

- Oracle 9i, DB2 V7 and 8.x, Sybase 12.x, Informix, MS SQL, MY SQL
- Oracle 9i (32-bit and 64-bit)
  - ora\_start9i, libfxora9i
- Sybase 11.5 and up
  - syb\_start, libfxsyb
- IBM DB2 v7.x (AIX only)
  - db2\_start7, libfxdb2
- Informix 9.x
  - ifx\_start9, libfxifx9
- MS SQL Server 2000
- MySQL 4.x (RedHat only)

#### ***Renewal Topology: Entera with NXTera 5***

The figures below outline the architecture of a legacy Entera and the updated topology of a system upgrade to NXTera 5.

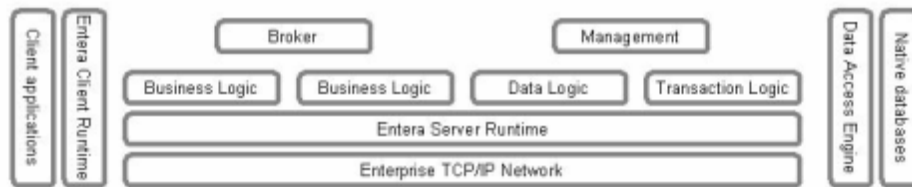


Figure 2: Entera Topology



Figure 3: NXTera Updated Architecture

## 4. Evolutionary Process - Evolution

Evolution enables the modernization of existing Entera (NXTera) business logic. During the first stage, legacy systems are enabled to support and interoperate with contemporary transports, protocols, language bindings and frameworks. During the second phase, the last vestiges of the legacy Entera system, legacy platform and/or legacy development language are dropped and converted to contemporary equivalents (for example COBOL/C replaced with JAVA).

The remainder of this paper focuses on the initial phase of the evolutionary process. The second phase is described in subsequent white papers.

### ***Evolution Phase: NXTware Evolution Server***

eCube's NXTware EV (Evolution Server) enables existing business logic developed in Entera and other legacy systems to interact, leverage and integrate with contemporary distributed service platforms, protocols and applications. Through a combination of emulation, new architectures and advanced provider/consumer support, NXTware EV allows companies to extend the value of their existing technology investments.

NXTware EV applies industry standards and a deep understanding of legacy systems to enable the evolution of existing business logic to contemporary platforms. Following is a list of benefits and industry standards available in NXTware EV, as well as a discussion of its architecture and product features.

#### NXTware Evolution Server Platform

NXTware is a lightweight evolution platform built entirely in Java. NXTware takes advantage of open industry standards to enable legacy systems to generate XML based on existing metadata and participate in contemporary architectures such as .NET, J2EE, and Web Services.

#### Development Environment

NXTware-based applications can be developed and configured using any Java

development environment. Development is done using industry standards for Java, XML and SQL programming.

#### Integration Capabilities with JCA

NXTware is compliant with J2EE Connector Architecture (JCA). This allows legacy systems to act as J2EE resources, with NXTware EV acting as a resource adapter.

The J2EE Connector Architecture is a standard architecture for connecting the J2EE platform to heterogeneous legacy systems.

Examples of supported legacy systems include ERP, mainframe transaction processing, database systems, and legacy applications written in non-Java programming languages, such as Entera. By defining a set of scalable, secure, and transactional mechanisms, NXTware (using the J2EE Connector Architecture) enables the integration of legacy applications with application servers and enterprise applications.

#### Integration Capabilities with XML

NXTware EV makes all legacy Entera/NXTera data, transactions, business logic server capable of responding to requests for responses in XML format. eCube makes it possible for NXTware to provide a way to generate XML-based business documents from legacy systems. Using NXTware uniform XML interface representations are generated, despite different implementation technologies such as Entera, RPC, DCE and others.

In addition, NXTware use of XML extends business scenarios across firewalls enabling secure flow of business documents without requiring changes to established security infrastructures.

#### NXTware Cross-Architecture Integration

NXTware EV supports a variety of integration models and standards to allow users the greatest amount of flexibility. They include:

- As a Consumer of
  - EJB functions
  - .NET
  - Of XML
- As a Provider to
  - Native client bindings
    - C, C++, C#, Java, J#, COBOL
  - EJB servers and clients
  - .NET servers and clients
  - Of XML
    - HTTP/BrowserJSP/Servlet
- Integration Architectures and Targets
  - Commercial ERP's (via JCA)

- CICS
- IMS
- Open VMS
- Multiple protocol support
  - RPC, DCE, JMS, RMI, SOAP

### Multi-language Binding and Development

NXTware EV supports a large number of client bindings and development platforms. Below follows a short list of contemporary development tools, environments and languages.

- J2EE, .NET and Servlet development environments
- Multiple language binding: Java, .C#, J#, C++, 4GLs, etc

### Emulation features

NXTware EV can fully emulate Entera, DCE and other RPC-based distributed systems. This allows NXTware to easily be added to existing legacy systems without disrupting the underlying applications.

## ***NXTware EV Implementation***

NXTware EV easily installs on top of NXTera. NXTware EV emulates Entera/NXTera servers and clients, enabling users to easily overlay NXTware on existing applications for complete integration with platforms such as .Net and J2EE.

### Overlaying NXTware EV

NXTware EV supports standard Entera communication protocols, allowing it to easily communicate with existing clients and servers. Installing NXTware in an existing environment provides legacy applications with all of the features and benefits of NXTware EV.

Additionally, by using applications upgrade to NXTera from Entera, users experience better performance with NXTera optimized for NXTware integration

Here is a snippet of a NXTware configuration file that defines a Java server class. As you can see, it uses standard Entera ENV and DEF files for compatibility. However, the settings in these files can be overridden by later entries in the XML configuration file.

```
<nxtconfig name="local">
  <defaultServerPort>7777</defaultServerPort>
  <classPath></classPath>

  <server name="basicsrv" dedicated="true">
    <class>ecube.examples.basicsrv</class>
    <defFilePath>basicsrv.def</defFilePath>
    <envFilePath>server.env</envFilePath>
    <transport>tcp</transport>
    <serverPort>0</serverPort>
    <maxPoolSize>1000</maxPoolSize>
  </server>
</nxtconfig>
```

```

    <varServerName></varServerName>
    <brokerlist>
      <broker host="localhost" port="7080"/>
    </brokerlist>
  </server>
</nxtconfig>

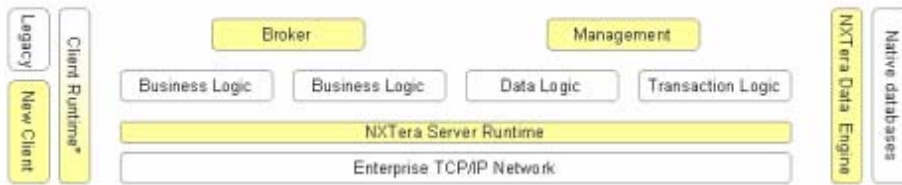
```

Unlike Entera, multiple servers can be defined to listen at one port, simplifying firewall configuration for NXTware servers. Providing additional ease of use, a whole set of NXTware application servers can be started by just one command:

**NXTware -start nxtconfig.xml**

NXTware is also compatible with old Entera Netminder instances for simplifying legacy management.

***NXTware EV Topology***



**Update NXTera Topology**

Overlaying NXTware Evolution Server

By installing NXTware EV on top of NXTera 5.0, your applications can take full advantage of NXTware’s lightweight NXTera emulation capabilities. Existing applications will seamlessly communicate with the Evolution Server without any changes to the code or required recompilations. Existing NXTera/Entera servers will be able to receive requests from web browsers, .NET clients and J2EE clients and servers, instantly addressing modernization and integration requirements.

At the same time, legacy client applications can easily access contemporary service providers. By making simple changes to the NXTera broker, legacy clients will be able to access new resources that take the place of old ones.



**Overlaid NXTware Topology**

## ***NXTware EV: New Evolution Features***

### NXTware EV Interoperability, Emulation and Federation

NXTware EV accomplishes these capabilities by enabling contemporary clients and servers to emulate your applications' expected APIs. For example, when using .NET, a .NET stub is generated that emulates legacy communication protocols. It can then be compiled into the client to enable interoperability with legacy systems.

In the case of contemporary services, NXTware acts as a service intermediary that switches protocols as required for interoperable communications processing. In combination, these capabilities enable the federation of legacy servers and contemporary servers. Now a client can make a request that requires results from multiple clients and servers, with federation logic that conditionally orders and calculates the federated results.

Here is a snippet of a NXTware configuration file that now defines a server emulation class. This will now act as a "bridge" from a SOAP client to a legacy Entera server.

```
<nxtconfig name="local">
  <defaultServerPort>7777</defaultServerPort>
  <classPath></classPath>

  <server name="basicsrv_bridge" client_emulation="true">
    <transport>soap</transport>
    <defFilePath>basicsrv.def</defFilePath>
    <envFilePath>client.env</envFilePath>
    <serverPort>0</serverPort>
    <maxPoolSize>1000</maxPoolSize>
    <varServerName></varServerName>
    <brokerlist>
      <broker host="localhost" port="7080"/>
    </brokerlist>
  </server>
</nxtconfig>
```

NXTware provides the translation from legacy Entera DEF files to SOAP WSDL files using the `nxtmake` command:

```
nxtmake -soap basicsrv.def
```

This converts the WSDL file to "basicsrv.wsdl" which can then be used to make SOAP calls to NXTware.

### NXTware Servlet Engine

NXTware EV contains a 100% Java HTTP Server and Servlet Container. This means that you do not need to configure and run a separate web server (like Apache) in order to use Java, servlets and JSPs to generate dynamic content. Based on Jetty, it is a full featured web server for static and dynamic content. Unlike separate server/container solutions, now your web server and web application run in the same process, without

interconnection overheads and complications.

- HTTP/1.1 server, it can be configured in a jar file under 300KB
- Consistently benchmarks as one of the fastest servlet servers
- HTTP server scales to thousands of simultaneous connections

Here is a snippet of a NXTware configuration file that now defines an automatic HTTP/XML interface to an existing Entera server:

```
<server name="basicsrv">
  <class>ecube.examples.basicsrv</class>
  <defFilePath>basicsrv.def</defFilePath>
  <envFilePath>server.env</envFilePath>
  <transport>tcp</transport>
  <serverPort>0</serverPort>
  <maxPoolSize>1000</maxPoolSize>
  <varServerName></varServerName>
  <brokerlist>
    <broker host="localhost" port="7080"/>
  </brokerlist>

  <servlet expose="true">
    <urlPath>/basicsrv</urlPath>

    <excludedMethods></excludedMethods>
    <transformers>
      <transformDirectory></transformDirectory>
      <defaultTransformer>defaultxform.xml</defaultTransformer>
      <method name="method1" transformer="xform1.xml"/>
      <method name="method2" transformer="xform2.xml"/>
    </transformers>
  </servlet>
</server>
```

### NXTware EV Data Agents

NXTware EV Data Agents replace the old Entera SQL Start, with a new Java-based JDBC data access agents. These flexible agents are designed to natively communicate using SOAP to Binary or XML data. (Optionally high speeds NXTware DataStarts are also available for the fastest possible data processing. DataStarts are compatible with legacy Entera SQL Starts)

Here is a snippet of a NXTware configuration file that defines an automatic HTTP/XML interface to an existing Entera server. As you can see, it uses old Entera SQL-style files for compatibility.

```
<dataserver name="orastart_1" type="ora9i">
  <sqlFilePath>dbserver1.sql</sqlFilePath>
  <envFilePath>server.env</envFilePath>
  <transport>tcp</transport>
  <serverPort>0</serverPort>
  <maxPoolSize>1000</maxPoolSize>
```

```
<varServerName></varServerName>
<brokerlist>
  <broker host="localhost" port="7080"/>
</brokerlist>

</dataserver>
```

## 5. Phased Implementation of NXTware EV

A phased approach is available for companies wanting to implement segments of the capabilities provided by NXTware.

### ***NXTware Evolution Adapters***

NXTware Evolution Adapters are evolution agents that provide given subsets of the functionality available in NXTware. The NXTware Evolution Adapters is available in the following configurations:

- XML Provider with Servlet Support
- JCA
- CICS
- IMS
- Open VMS
- ERP

Contact your eCube salesperson for more information.

## 6. Eliminating Legacy Dependencies with NXTware Evolution Server

“Entera Evolution: A White Paper, Part Two” addresses the next phase of the evolution process when all communication is updated to contemporary protocols and transports. In this phase legacy business logic is either:

- Recompiled with substitute call implementations
- Converted to UML/XML and re-provisioned
- Integrated with an eCube NXTware RMI Evolution wrapper

This process is outlined in the forthcoming second part of this white paper. Please visit our web site and register for NXTware update information.

## 7. Operational Efficiency and Minimized Risk

Modernization has a cost, but that cost shouldn't be your system performance. In many cases, IT organizations are building new systems to replace legacy applications, only to find the legacy systems outperform the new. A true evolution strategy embraces a

commitment to steady improvement in performance as well as the fulfillment of service level goals.

Likewise, risk is something every business executive has to deal with. Whether a company decides to “stay put”, redevelop, or evolve, there is risk involved. With proven technology and methods that insure the value of IT efforts are moving into the future, eCube is committed to helping you manage your risk. Old applications can be maintained, renewed, evolved, transformed or harvested to speed new development in a way that assures the ability of IT to meet its commitments to the business and exceed expectation to reliability.

## **8. Summary**

IT managers are facing a variety of business pressures that are forcing them to evaluate the modernization of legacy systems. These pressures include the ongoing business requirements for lower operational costs, the need for better performance and reduced risk, and compliance with industry and corporate standards.

With NXTera 5.0 and NXTware EV, eCube Systems provides a systematic approach to legacy evolution the extension of technology equity. This process is based on the evolution of existing business logic and the integration/Implementation of contemporary platforms, such as .NET, J2EE, Web Services, HTTP/Servlets and XML.

eCube Enterprise Evolution enables companies to extend the value of existing applications and business logic by defending it from “software hardening” -- the growing inflexibility of legacy systems -- and enabling it to participate as an enterprise service provider.