



ObjectBuilders

WHITE PAPER

LiveApp Player Architecture

Player Technology
The Composite Applications
Deployment Style Architecture

WHITE PAPER

LiveApp Player Architecture

© ObjectBuilders
20134 Valley Forge Circle
King of Prussia PA 19406
Phone (610) 783-7748 • Fax (610) 783-0389

Restricted Rights

Printed in the United States of America

The information contained in this document is proprietary and confidential to ObjectBuilders, Inc. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose without the express written permission of ObjectBuilders, Inc.

This document is subject to change without notice, and ObjectBuilders does not warrant that the material contained in this document is error-free. If you find any problems with this document, please report them to ObjectBuilders in writing.

ObjectBuilders, Inc., the ObjectBuilders, Inc. logo, LiveIntegrator™, LiveModeler™, LiveXML™, and LiveSolutions are registered trademarks of ObjectBuilders, Inc. All other company and product names may be trademarks of their respective owners. The information contained herein is subject to change without notice. Copyright © 2006 ObjectBuilders, Inc. All rights reserved.

This document contains or may contain statements of future direction concerning possible functionality for ObjectBuilders' software products and technology. All functionality and software products will be available for license and shipment from ObjectBuilders only if and when generally commercially available. ObjectBuilders disclaims any express or implied commitment to deliver functionality or software unless or until actual shipment of the functionality or software occurs. The statements of possible future direction are for information purposes only and ObjectBuilders makes no express or implied commitments or representations concerning the timing and content of any future functionality or releases.

Document Purpose and Scope

This document provides an overview of ObjectBuilders LiveApp Player Architecture. It is written for a technical audience that is familiar with basic Internet, Web services and enterprise architecture concepts. It is intended to provide the reader with a general understanding of ObjectBuilders LiveApp Player Architecture.

Introduction	2
The Technology Behind the LiveApp Player.....	3
The Composite Application	5
Deployment Style Architecture	6

Introduction

Today solutions are made up of technology, services and products supplied by different vendors. The newer products allow for significant customizations without the need for traditional coding. This means that a large part of any business application can be completely configured with the vendor supplied products. Sometimes 100% configurable!

Configuring or assembling removes the need for custom coding – increasing quality and reducing the risk associated with creating large enterprise business solutions. Traditional APIs, frameworks and inflexible applications that require coding are becoming a thing of the past.

The next generation of application building products are flexible packages that are highly configurable. These products come with powerful configuration tools known as productivity tools for developers. Crystal Reports, ObjectBuilders' LiveApp Player Suite and FileNet's P8 Process Designer are examples these next generation products.

Crystal Reports, The LiveApp Player Suite and FileNet's P8 Process Designer all use a document strategy (rpt, LiveXML, pep respectively) to store configurations that are processed during runtime to produce the desired result. Documents created with productivity tools can produce even the most complex and unique business application.

Products like these will do for the development what Microsoft Office did for the desktop. The technology industry categorizes these next generation of application building products into the following categories: report writers, workflow tools, content management tools and presentation tools.

A Composite Application Builder is a next generation productivity tool focused on eliminating traditional coding in the presentation tier. It is designed to consume the business service tier directly and produce complex and unique business applications. The Composite Application Builder handles the complexity of bringing together multiple backend data sources, services, and legacy systems, handling multiple deployment styles, managing user interface security, tracking audit history and other presentation tier tasks.

ObjectBuilders LiveApp Player Suite is a powerful Composite Application Builder.

The Technology Behind the LiveApp Player

The flexibility of the LiveApp Player is the result of the entire application being represented as a set of XML documents. Similar to an HTML web site running in a browser, applications created with the LiveApp Player can run anywhere there is a Player. This Player Technology is familiar to anyone who browses the Internet. When surfing the Internet, you most likely have used Player Technology to view a document, listen to a sound byte or play a video clip.

Well-known examples of player technology include Adobe® Reader®, used primarily for electronic replication of high-quality print renditions, and Microsoft® Windows Media® Player, used primarily to play and organize digital media files on your computer. Adobe Acrobat uses a proprietary document type (PDF), while Microsoft Media Player uses both proprietary formats (WMV) and open formats (MP3, MPG).

Player Technology consists of three components: Documents, Productivity Tools and Players.

	ObjectBuilders LiveApp Player Suite	Adobe Reader	Microsoft PowerPoint
<i>Purpose</i>	Play high quality, portable business solutions	High quality print rendition	Creates Sales and Marketing Presentations
<i>Document</i>	LiveXML, XML	PDF	PPT
<i>Productivity Tools</i>	LiveIntegrator™, LiveModeler™, and any XML based editor or tool	Acrobat	PowerPoint
<i>Player (rendering engine)</i>	LiveApp Player	Adobe Reader	PowerPoint Viewer

Table 1 Player Technology Components and Examples

Microsoft PowerPoint is very similar to the LiveIntegrator™ (the Composite Application Builder of the LiveApp Player Suite). It allows a user to drag components off a palette and place them onto a canvas, configuring a presentation. The attributes of specific shape components can be manipulated through inspectors. While there are constraints and boundaries to Microsoft PowerPoint, there is no practical limitation to the types of presentations that can be created. Similarly, there is no practical limitation to the types of applications that can be created with the LiveIntegrator™.

- **Documents** control the behavior of the Player. *LiveXML* documents are used in combination with other XML schemas. *LiveXML* is an XML vocabulary that represents business applications. It allows the assembly of rich user interfaces and the integration of enterprise applications, databases and web services into a composite application.

- **Productivity Tools** are for building and maintaining the application. These tools are based on a drag and drop WYSIWYG configuration rather than coding or scripting. The LiveApp Player Suite comes with two productivity tools: the LiveModeler – used to create a virtual model of enterprise components – and the LiveIntegrator – the Composite Application Builder.
- **Players** read the documents and render the application. The LiveApp Player’s runtime component is a player that is not specific to any solution or business domain, rather it is a horizontal engine that processes the LiveXML documents and renders the composite application from the documents.

The documents (LiveXML) that are used by the LiveApp Player can easily be maintained and modified using ObjectBuilders *Assembly Tools* as well as any third party XML tools.

Because Player Technology uses *Documents* to represent the application, rather than code, it can be easily managed and delivered simultaneously to multiple deployment styles including full client, rich client, thin client, disconnected client and even to other devices like PDAs. The documents contain device independent information that the players (which are device dependent) need to render and run the application.

The Composite Application

Many of today's enterprise business solutions are Composite Applications. A Composite Application consists of functionality drawn from several different sources within a Service Oriented Architecture (SOA). A Composite Application exists in the presentation tier and assimilates the SOA into a powerful and unique business application.

Most presentation tiers contain significant code to produce even the most simple business solutions. The LiveIntegrator™, a Composite Application Builder, brings together the application in the presentation tier, producing even the most complex business applications without the code.

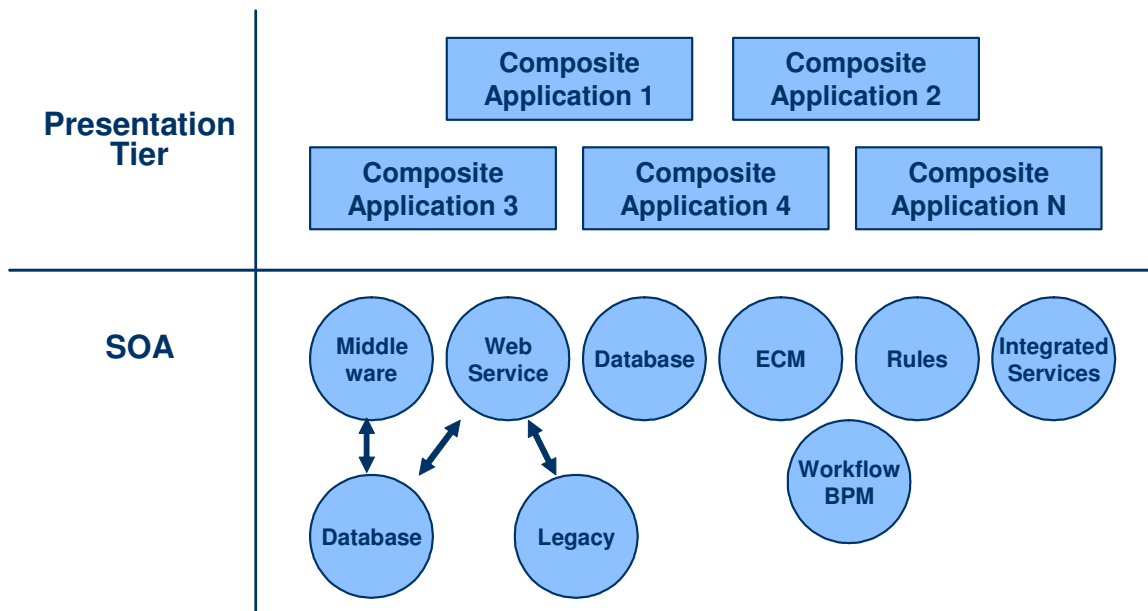


Table 2 Composite Application Assimilates the SOA

To bring the services together, the ObjectBuilders LiveApp Player includes a set of connectors. These connectors are the product or technology specific integrations that are abstracted by the LiveApp Player. The XML documents that represent the front end of the application reference only the abstracted concepts and the technology implementation details are hidden from them. This allows the user interfaces of the application to remain loosely coupled to the data sources.

Deployment Style Architecture

The LiveApp Player Suite is focused on the presentation tier, allowing it to live in virtually any architecture that uses a presentation tier. Since the LiveApp Player Suite architecture uses Player Technology, and the application is represented as a set of LiveXML documents, it can be centrally maintained while being simultaneously deployed to different deployment architectures without sacrificing functionality.

ObjectBuilders products are designed to be development platform and technology agnostic. They work in both J2EE and .NET environments. Everything a customer creates is represented in XML and rendered depending on the deployment style. The LiveApp Player is device dependent and requires the use of base technology. While the LiveApp Player itself currently uses the technology described in this document, it does not require developers to use these technologies for anything other than deploying the LiveApp Player Suite.

The style or combination of styles that are chosen will depend on the particular solution needs. The following application deployment styles are currently available:

- **Thin Client.** This Server centric style offers the thinnest client footprint capable of running a pure HTML or HTML / AJAX interface. The architecture is best described as a traditional 3-tier architecture.
- **Full Client.** This Client centric approach offers a rich, intuitive interface. This style requires a client install and has a larger client footprint.
- **Rich Client.** This hybrid Server and Client centric approach uses a small plug in that is automatically downloaded to the client one time, after which only the Documents are downloaded. Documents usually load as fast as a typical HTML web page.
- **Disconnected Client.** This variation of the Full Client version allows application to run when offline or disconnected from a network or the Internet.
- **PDA Client.** This variation of the Full Client version allows application to run on a PDA device.

Each deployment style uses a device dependent version of the LiveApp Player. The LiveApp Player is running either on the client or the server. Differences in deployment styles, as well as a consistent look and feel with existing enterprise applications, can be controlled through environment settings and style sheets.

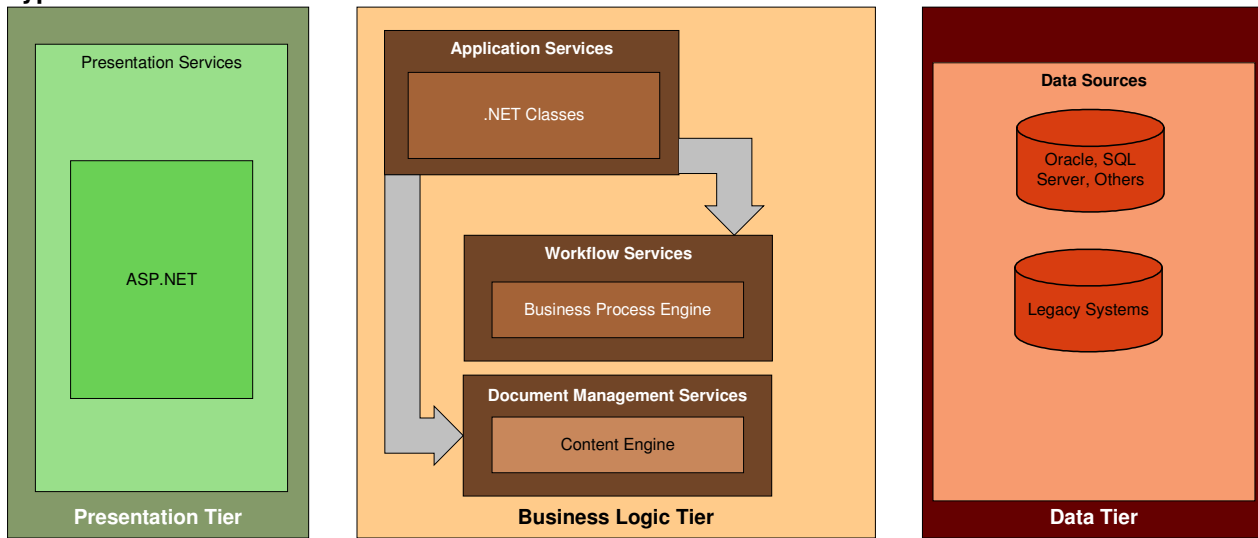
Thin Client

The thin client deployment style is deployed in a standard .NET n-tier architecture. In this deployment style, the LiveApp Player is a .NET assembly in the presentation tier, and it is accessed by ASP.NET to render the designated XML documents into HTML. The application server technology used is the standard Microsoft

Windows / IIS application server technology. The player can go directly against the data sources or it can go through a business logic tier. The business logic tier may be a set of custom web services, or it may also be implemented using ObjectBuilders Connector Service.

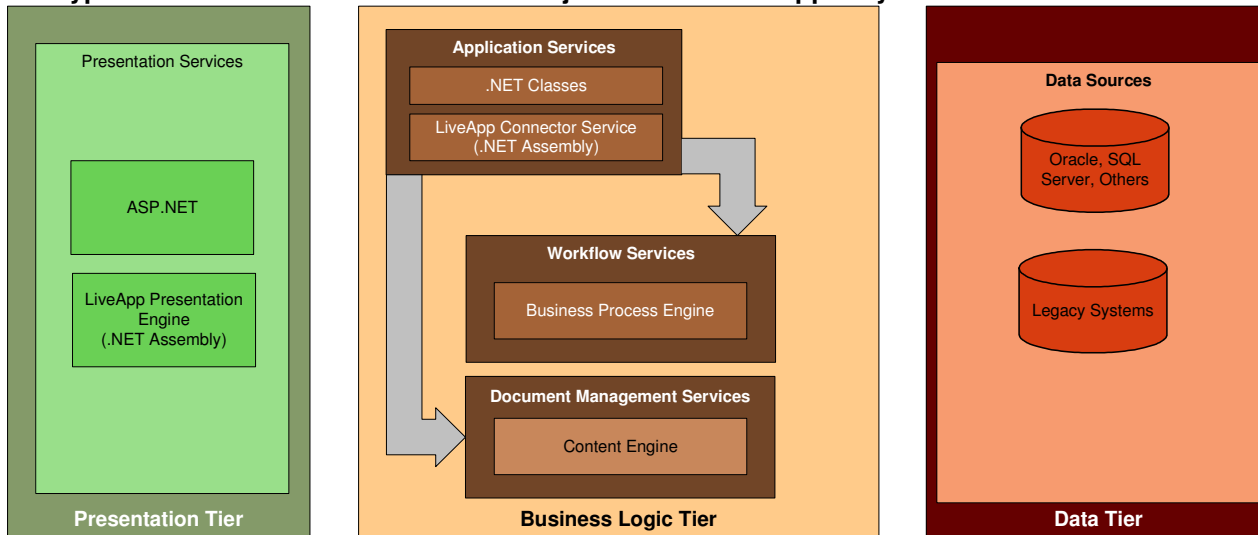
Note: In the diagrams Workflow Services and Document Management Services are shown. These are not required and are provided for illustrative purposes only.

Typical 3-Tier .NET Architecture



ObjectBuilders thin client deployment style plugs into the .NET 3-tier architecture as a set of .NET Assemblies.

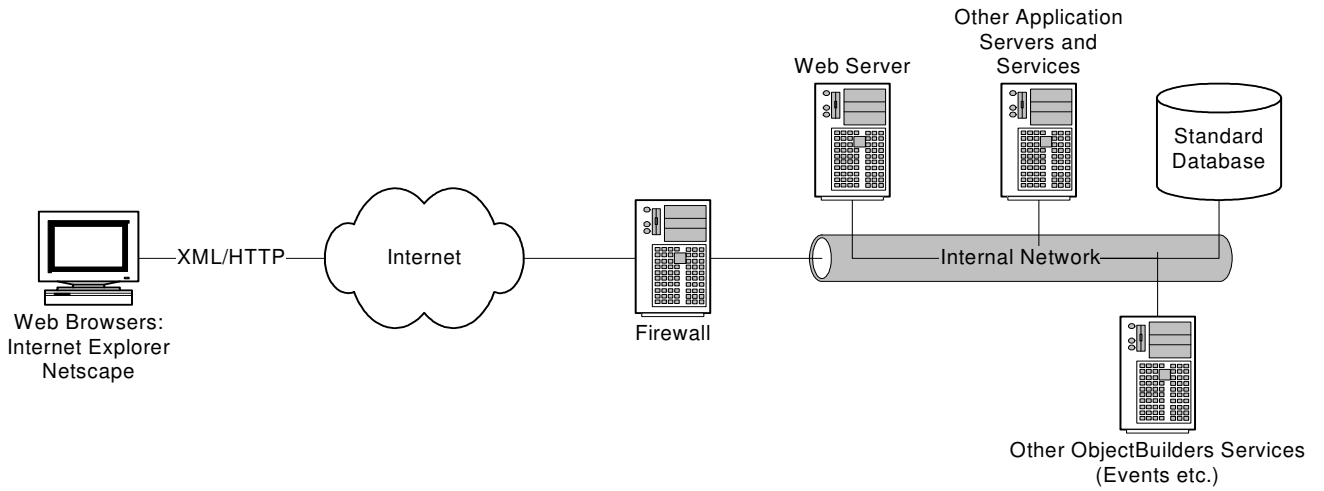
Typical 3-Tier .NET Architecture with ObjectBuilders LiveApp Player



J2EE can be accessed through web services. This mechanism is widely used as an integration paradigm for cross-platform integration between J2EE and .NET architectures.

The thin client deployment style leverages AJAX technology by default and uses the .NET ATLAS capability provided with .NET 2.0. Optionally, the thin client deployment style may also be deployed without AJAX in which case the ATLAS capability is not needed.

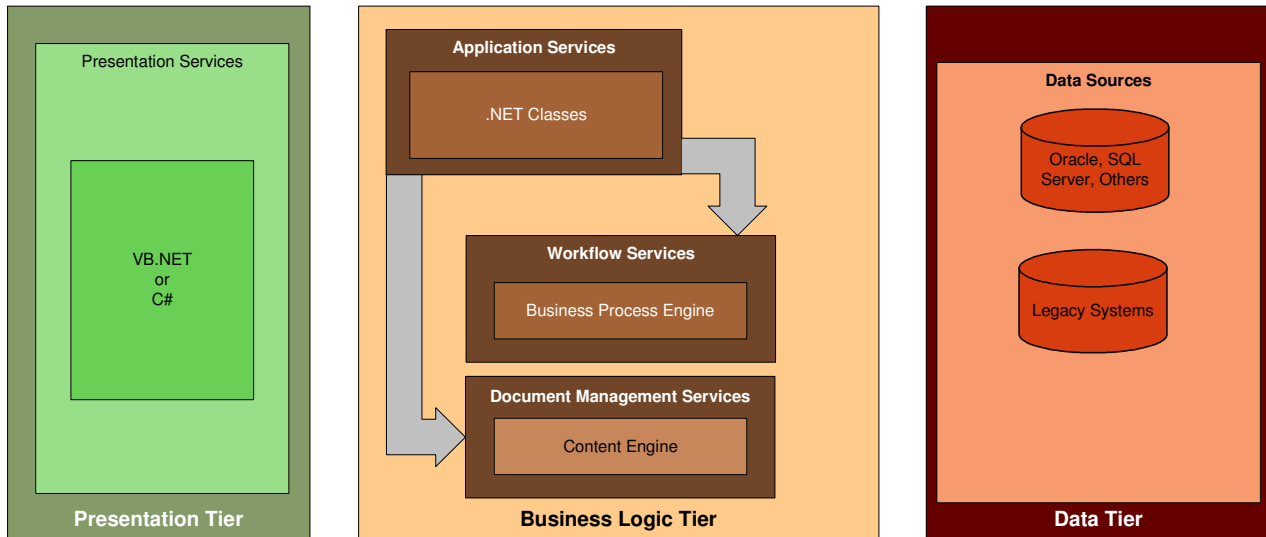
As is the case with a standard .NET architecture, it is possible to run the .NET components on the web server or to host them on a separate physical tier. .NET Remoting or web services can be used to achieve this. A typical deployment architecture is shown in the diagram, assuming the component is running on the web server.



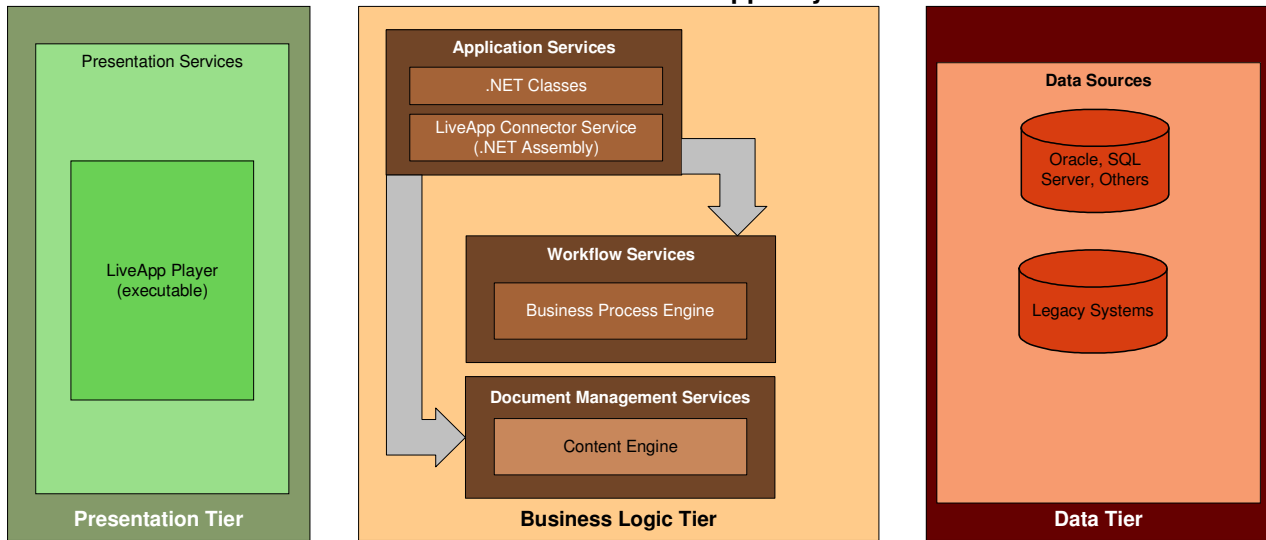
Full Client

The Full Client deployment style requires an installation on the desktop and may exist in an n-tiered or client server application architecture. The LiveApp Player full client executable runs in the presentation tier and takes the place of a full client application coded in a language like VB.NET or C#.

Traditional 3-Tier Full Client .NET Architecture



Traditional 3-Tier Full Client .NET Architecture with LiveApp Player



Rich Client

The Rich Client deployment style requires a small plug-in on the desktop and may exist in an n-tiered or client server application architecture. The Rich Client is a plug-in to the browser, and is similar to the full client.

The plug-in to the browser is an Active-X component, similar to Adobe Reader. As a user accesses a web page that references the LiveApp Player, they are presented with a screen asking them if they want to install the LiveApp Player. After the LiveApp Player is installed, the LiveXML documents are sent to the LiveApp Player by a server. The Rich Client will access the data sources the same way as the Full Client shown above. A deployment diagram is provided that shows how the Rich Client Player may be deployed.

