



AltioLive Developer Training

Version 5.4

1. Introduction to the AltioLive Training

Summary

This is the introductory module for the Developer Training course.

Integra SP

88 Wood Street London
EC2V 7RS
United Kingdom

www.altio.com

tel: +44 (0) 20 8528 1045

Contents

Introduction to the Training	2
Organization of the Training	2
Course Prerequisites	2
Course Aims	2
Introduction to Altiolive	3
Product Overview	3
What is Altiolive?	3
Who needs Altiolive?	3
Make the data in your web site live.....	3
Drag and Drop	3
Personalized	3
Scalability	3
Multiple Users.....	4
Architecture Overview	4
The Evolution of Application Development and Delivery.....	4
The Altiolive Presentation Server and Developer Tools.....	5
How an Application Works in the Altiolive Environment.....	7
Altiolive Architecture Features	8
Altiolive Studio Edition Tools	8
Glossary.....	9
Demonstrations	11
Richmond Office Supplies Demonstration.....	11
Further examples	13
The File Structure for Altiolive	14
File structure below the 'webapps' directory.....	14
The Presentation Server directory (classes directory).....	15
The contents of the Applications directory (apps directory).....	15
Configuration (the conf directory)	16
Logging (the logs directory)	16

Introduction to the Training

The AltioLive Developer training course is designed for those intending to use the powerful, visual development environment of the AltioLive Studio to develop complex Web and XML-based applications.

The course looks at the configuration and control of application communication, integration and presentation. These training notes provide detailed instructions on how to create an application.

Organization of the Training

The training is organized in nine documents. We recommend going through them following a chronological order but you can jump from one document to another if you want more information on a specific tool or feature of Altio Live.

The training progress:

1. Introduction to the AltioLive Training
2. The First Steps to Build your Application
3. Referencing Data in AltioLive Applications using XPath
4. Using Events and Actions to create a dynamic application
5. Getting Live Data using Datapools
6. Creating a Report with Altio Report Writer
7. Using JDBC, SOAP and Java Service Functions
8. Using the Prototyping wizard
9. Debugging your application
10. Further Exercises

Course Prerequisites

There are no specific prerequisites for the course. A typical course delegate will have experience of web-site construction, and some experience of programming.

Course Aims

The course will enable you to:

- Understand the AltioLive architecture.
- Install the AltioLive platform and understand how it is integrated with an application server.
- Become familiar with the AltioLive tools that are used to maintain the software, integrate with a backend application, and create the graphical user interface of the AltioLive application.
- Create an application using a selection of the AltioLive functionality
- Gain an understanding of the markup language and referencing syntax, mainly XML and XPath, used within the AltioLive platform.

Introduction to AltioLive

This session introduces AltioLive. We begin by looking at a demonstration of a sample AltioLive application to discover the main features of the AltioLive presentation capabilities. We will explore application development methods and explain the benefits of the Presentation Layer. Finally we take a tour of the AltioLive Presentation Server environment and developer tools.

Product Overview

What is AltioLive?

AltioLive improves the user experience of the Internet by offering the interactivity and functionality of a desktop application. This familiar windowed interface means that users are productive with new applications quickly

Who needs AltioLive?

Altio technology solves the bottleneck of HTML forms-based applications, and so is ideal for any business that needs to increase the performance of internet-based applications. This applies to any web-based business: companies moving their business to the Internet or intranet environments, and software companies transitioning their applications to the web environment.

Altio allows businesses to reduce web application development time considerably. The Altio visual development environment provides an efficient way for businesses to configure applications quickly and get them to market faster. Altio customers reduce development time to market from months to weeks.

Make the data in your web site live

The efficient integration with backend applications allows data from existing servers to be managed by the Altio Presentation Server and delivered to the user in real time. User interfaces with the data are constructed easily using the AltioLive **Designer**.

Drag and Drop

The AltioLive windowed environment streamlines user interaction with an application. Data items may be dragged and dropped from one live window to the next, providing desktop functionality on a web site.

Personalized

Altio technology allows users to resize and arrange live windows on their web page. Multiple views of an application can be created to accommodate multiple role groups.

Scalability

Altio components can be clustered to allow for maximum flexibility of integration into existing back-end applications.

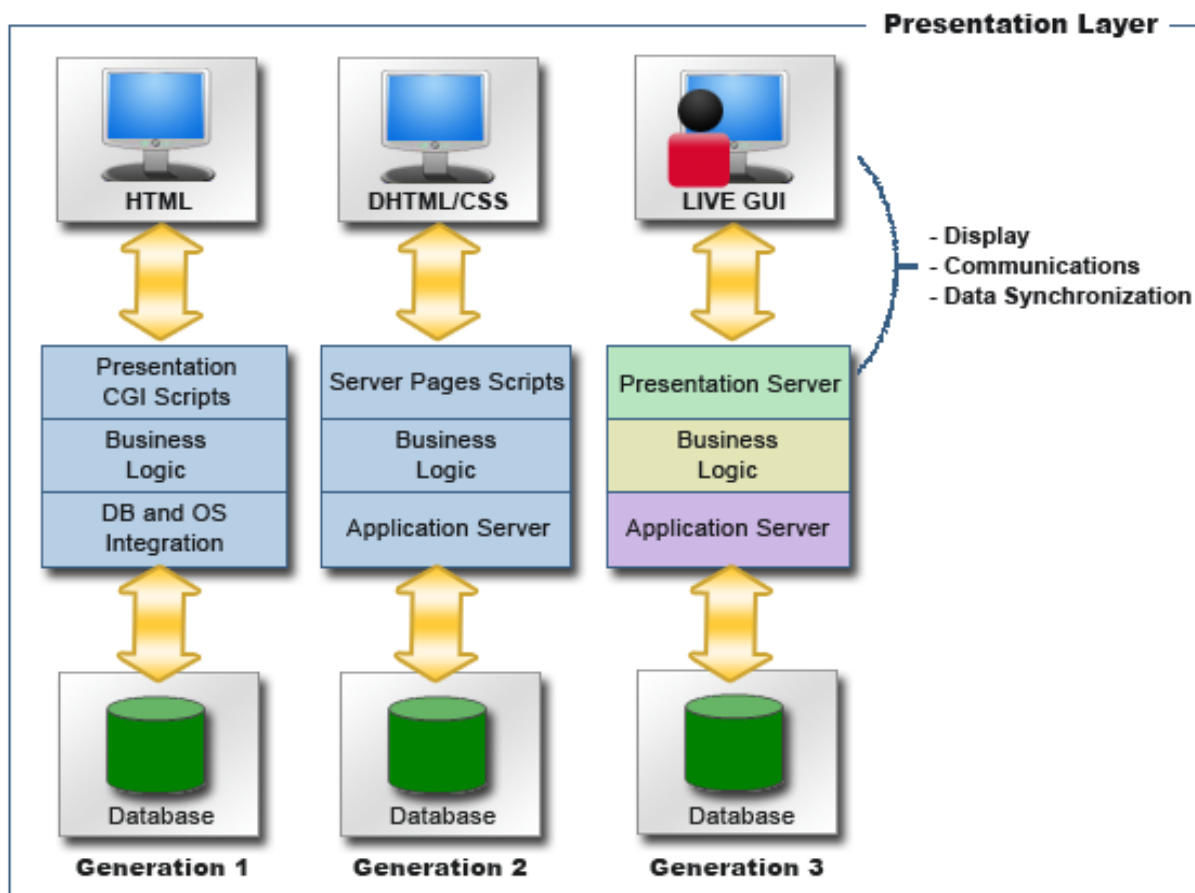
Multiple Users

Multiple clients can be connected to an application at the same time. Datapool technology ensures that each client sees an up-to-date view of the data as soon as a user submits it.

Architecture Overview

The Evolution of Application Development and Delivery

The development of web-based enterprise applications has changed considerably during the past few years. Here is a brief overview to that progression:



Generation 1: Complete Custom Coding

Development of enterprise web systems during the first half of the 1990s was highly specialized. A large amount of work was needed to integrate the database and operating system, and custom coding of business logic was required before the writing of scripts to provide transfer of data between client (browser) and database.

Generation 2: the Application Server

The Application Server addressed the challenges of database integration and utilities. This allowed the focus to be on the development of the business logic

Developers still had the laborious task of managing the connection between the application server and the client. Performance of applications at the client-side could be poor, and there were

significant limitations of presentation when compared to the feature-rich interfaces that exist on client/server-based desktop applications.

JavaScript, DHTML and plug-ins helped enrich the user interface, but the development and maintenance effort was still considerable. Plug-ins usually tied applications to specific computers and could introduce security issues.

Generation 3: the Presentation Layer

The AltioLive Presentation Server provides a level of benefit similar to that brought to the back-end system development by the application server. AltioLive creates a seamless process for integrating the front-end and back-end of systems using standard, non-proprietary technology.

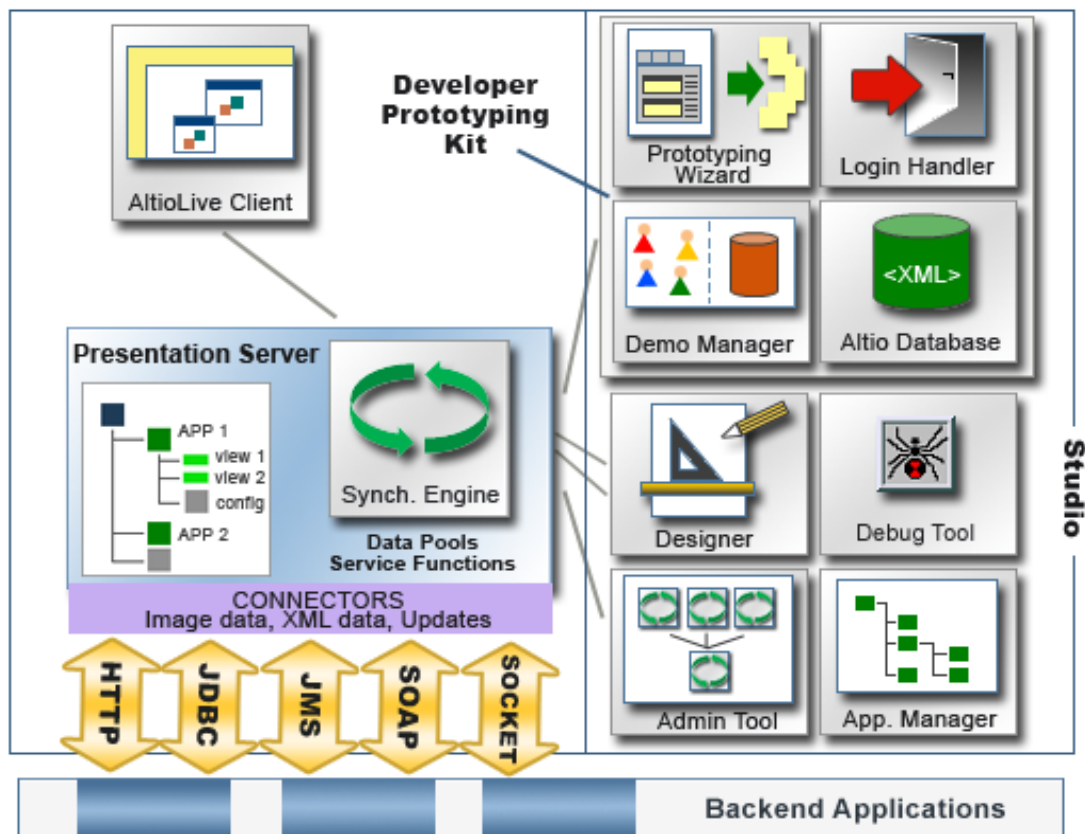
The Presentation Server manages the three elements of the presentation layer. It:

- Provides a rich user interface,
- Manages communications between server and client, and server with back-end,
- Synchronizes the distribution of data to all clients.

AltioLive developer tools provide a rich visual environment for the rapid prototyping, developing and testing of applications. The browser-based toolset allows collaborative development, instant switching from design mode to live test mode, and testing against live data or prototype data.

The AltioLive Presentation Server and Developer Tools

We can now take a tour of the AltioLive deployment environment and developer tools

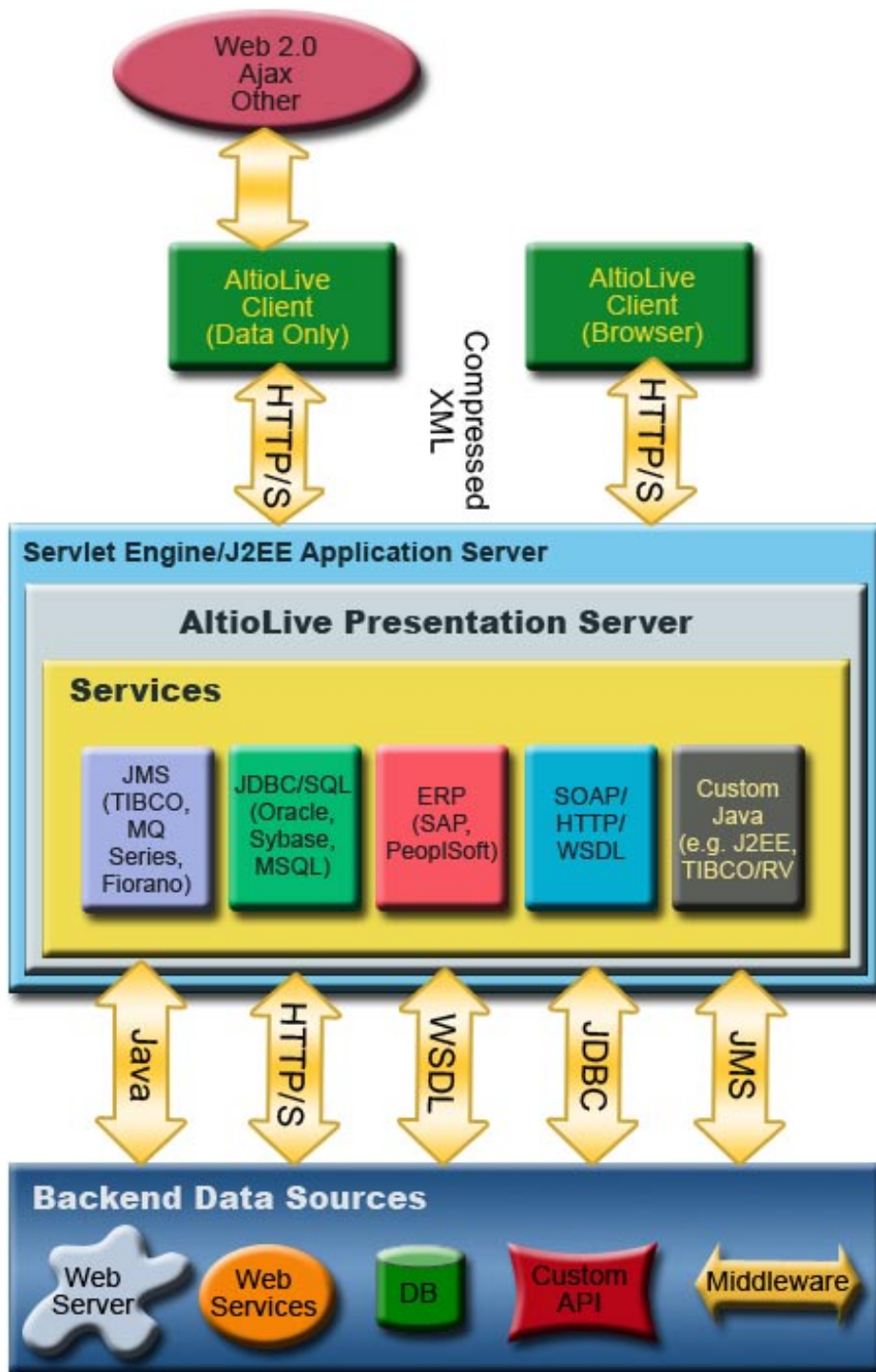


Here is a representation of the Altiolive architecture showing the Presentation Server, Clients, and connectivity options with Back-end applications.

The Altiolive Studio tools enable:

- The development and maintenance of applications,
- The configuration and control of the Presentation Server.

The following diagram details the components on the Altiolive architecture.



How an Application Works in the AltioLive Environment

The usual Client interface for an AltioLive application is a Smart Client running in version 4 and above browsers (i.e. Internet Explorer 4 and above, Netscape 4 and above).

The application is invoked from an HTML **Host Page**, which requests the AltioLive Client applet. The client is a small applet targeted for maximum browser compatibility.

The **Client** accesses the AltioLive **Presentation Server**. This is implemented as Java Servlets that support JSDK 2.0 for use on Servlet engines. This allows installation on all major Application Servers.

Once the initial configuration has been set, the application is then loaded.

The Client requests the Presentation Server to provide the **Application View Configuration XML** (known as the view file) and the initial data. These are interpreted by the Smart Client, which builds the application view from them according to the **Altio XML API**.

Any **Application Images** (such as backgrounds, button images, icons and skins) are then requested and served from the Presentation Server. The image components may be packed into composite files to minimize download times.

Moving onto the Initial Data to be presented in the application, the Application Server processes the **view file** and continues to process the **Application Config XML** to determine the initial **Service Requests** to be used. The initial data obtained from the web application is served to the Client where it is stored in the **XML Desktop** and presented to the user.

The application data may have an association with image files. When these are required from a Back-end application, image specific Service Requests are used. With the Initial Data served, the Presentation Server continues to process the **view file** to set up any **Data Pools** to be managed by the **Synchronization Engine**. Data updates can be fed via a polling mechanism, HTTP Streaming, JDBC, Java, JMS, SOAP or sockets.

The data in Data Pools is managed by a timestamp/sequencing mechanism. This ensures that updates to all the Clients can be maintained asynchronously; new data is either polled-from or streamed-to the Client to maintain the XML Desktop.

When data is updated from the Client (i.e. data that has been entered by the user) it is passed to the Presentation Server and then posted to the Back-end application through a Service Request. The Data Pool ensures that the change is reflected to all Clients.

Changes to the state of a Client desktop, for example window positions, are automatically posted to the Presentation Server. The desktop state may be saved locally (on the server), saved in an external database through a JDBC connection or posted to the Back-end application. This depends on the Server configuration.

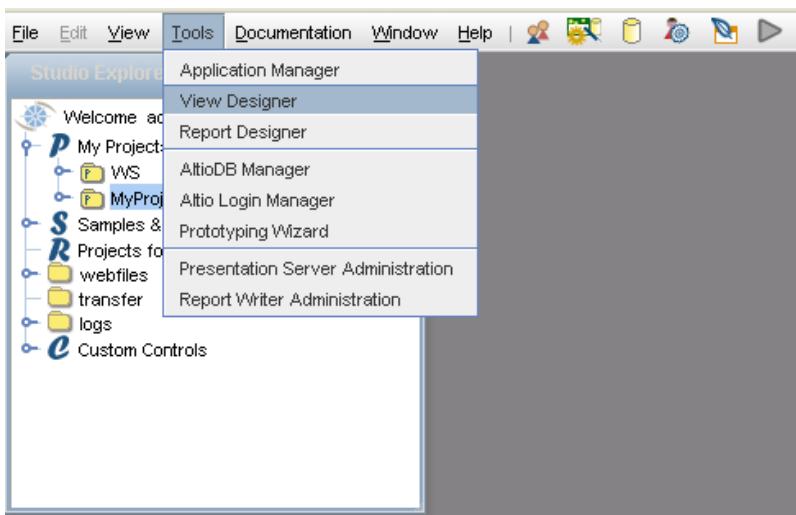
Presentation servers can be clustered/distributed to serve large communities of users using a master/slave configuration.

AltioLive Architecture Features

- **Small:** Since the Altio Client applet is small in size, downloading to the client machine is efficient and fast: about the same as a page loading. Once the user has finished working with the AltioLive software the applet is destroyed when the browser is closed and hence nothing remains.
- **Fast:** The data is held in the XML desktop on the Client. It is then rendered to the user according to the user request. Thus the number of direct data requests to the Application Server is reduced, increasing the speed of the performance of the Back end application.
- **Runs Anywhere:** The Full Altio Client can run within Java compatible browsers (Internet Explorer 4.0 and Netscape 4.0 and above). The Desktop client allows AltioLive applications to run as a desktop application and in both disconnected and connected mode. Similarly the AltioLive Presentation Server can be installed on any J2EE compliant Application Servers.
- **Client/Server:** AltioLive is implemented as Client/Server configuration. This is the efficient configuration for large volume and heavy traffic oriented Systems.
- **The Application XML:** Altio is configured with XML, which is fast becoming the standard mark-up language for data transfer over the Internet.

AltioLive Studio Edition Tools

The AltioLive Presentation Server and AltioLive Applications can be configured and managed using the **Studio** tools described below. They can be launched from the Tools menu in AltioLive Studio:



Application Manager

The Application Manager configures the integration between the Presentation Server and one or more back-end web applications.

Designer

This tool creates AltioLive application views for use with the Smart Client.

Report Designer

Just as the Designer (above) creates views, the Reports Designer creates report designs. Please note that although the Reports Designer is available and will allow users to create report designs, the Altio Report Writer upgrade must be purchased and installed before any reports can be run.

AltioDB Manager

This is a simple in-memory XML database used to simulate a Back-end application (it is a straightforward process to reconfigure the service interface to another Back-end application if this is required at a later date)

Altio Login Handler

This is a simple login Servlet that creates a session object and serves an HTML wrapper to launch the AltioLive Smart Client. It is for demo use and may not be sophisticated enough for production use.

Prototyping Wizard

This is a generator that takes an XML data structure definition and automatically generates the AltioDB data and config, sets-up the necessary Service Functions, Datapools and Datakeys and creates the basic View definition.

The Presentation Server Administration Tool

This tool is used to administer one or more Presentation Servers (within a cluster). Administration roles include the registration of applications and registration of administration users.

Report Writer Administration

This tool shows the user the current status of the report writer, any schedules running and lists the reports already built and stored.

Glossary

Name	Directory/File	Notes
Presentation Server	tomcat\webapps\altio54\WEB-INF\lib\ syncEngineXXX.jar	Coordinates the transfer of data between the client applet and the back end web application(s). A Presentation Server may comprise one or more Synchronization Engines in a cluster.
Client	tomcat\webapps\altio54\client\ clientRelease.jar ; clientDebug.jar	The AltioLive Client running within a browser. This is a java applet that is downloaded and exists on a HTML web page
Server configuration	tomcat\webapps\altio54\WEB-INF\classes\conf\ altioserver.xml	XML definition for the Presentation Server (IP address, logging levels etc)

Application configuration	tomcat\webapps\altio54\WEB-INF\classes\apps\NAME_PROJECT\ altioapp.xml	XML definition for each application on the Presentation Server (logging levels, service requests, datapools, datakeys)
Application View		Particular interface of the application. Different users may access the web application through different views, depending on their access level
Application View configuration	tomcat\webapps\altio54\WEB-INF\classes\apps\NAME_PROJECT\ views\ appViewName.xml	XML definition of the application view
Application Images	tomcat\webapps\altio54\WEB-INF\classes\apps\NAME_PROJECT\ Images directory	Images that are used to define the look of the applications e.g. Skins, Buttons, Icons etc.
Back end application		The Back end application/database resides outside of the AltioLive environment and supplies data to the Presentation Server and/or accepts transactions from it.
Service Request	Within altioapp.xml	An access method to request data from, or post data to the Back end application.
Data Pool	Configured in altioapp.xml, logs in data directory	A table of time-stamped data records managed by the Presentation Server to track changes to data from the client or Back-end application.
Host Page		The HTML page that invokes the Client
User preferences		Details held within AltioLive relating to the appearance of the application view specific to a particular user
XML desktop		Buffer area on client machine where XML is stored and from where the application view can be rendered.

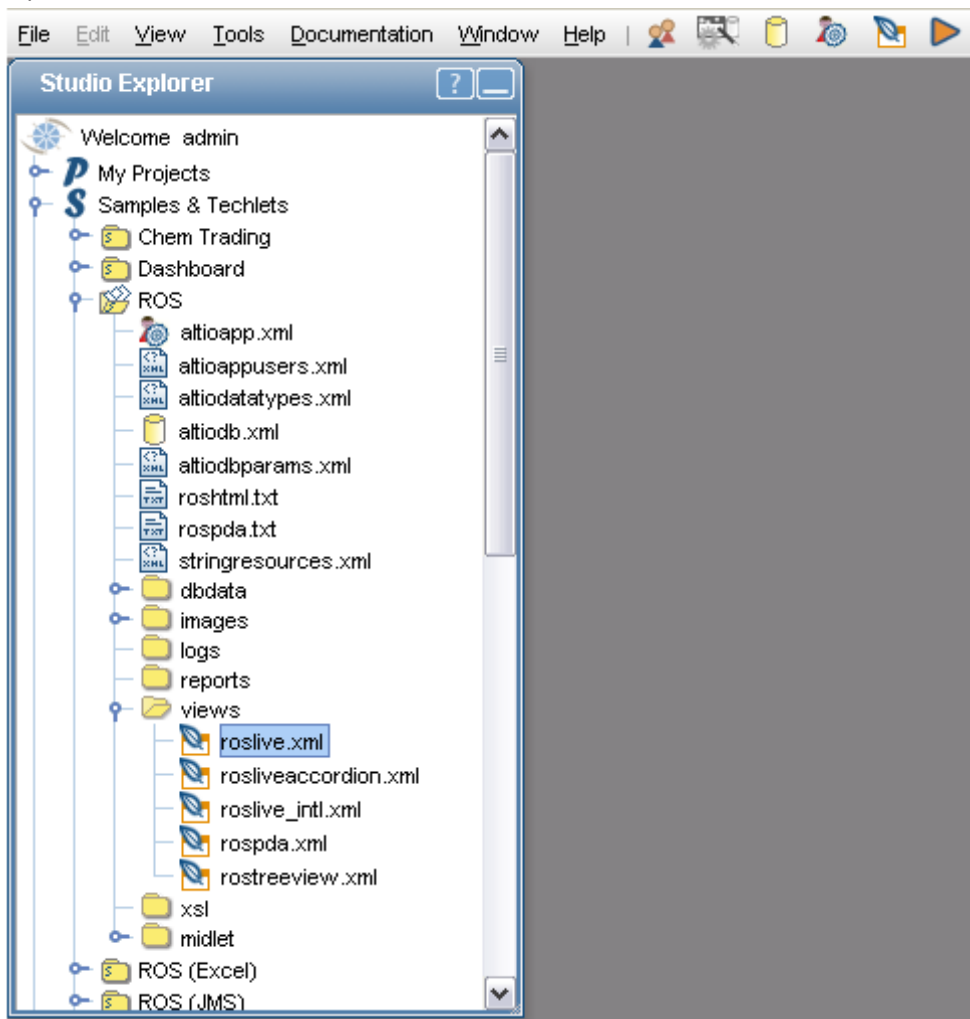
Demonstrations


Richmond Office Supplies Demonstration

Altiolive can be applied to any web-enabled application. This demonstration shows the example application of **ROS** (Richmond Office Supplies application) created with Altiolive.

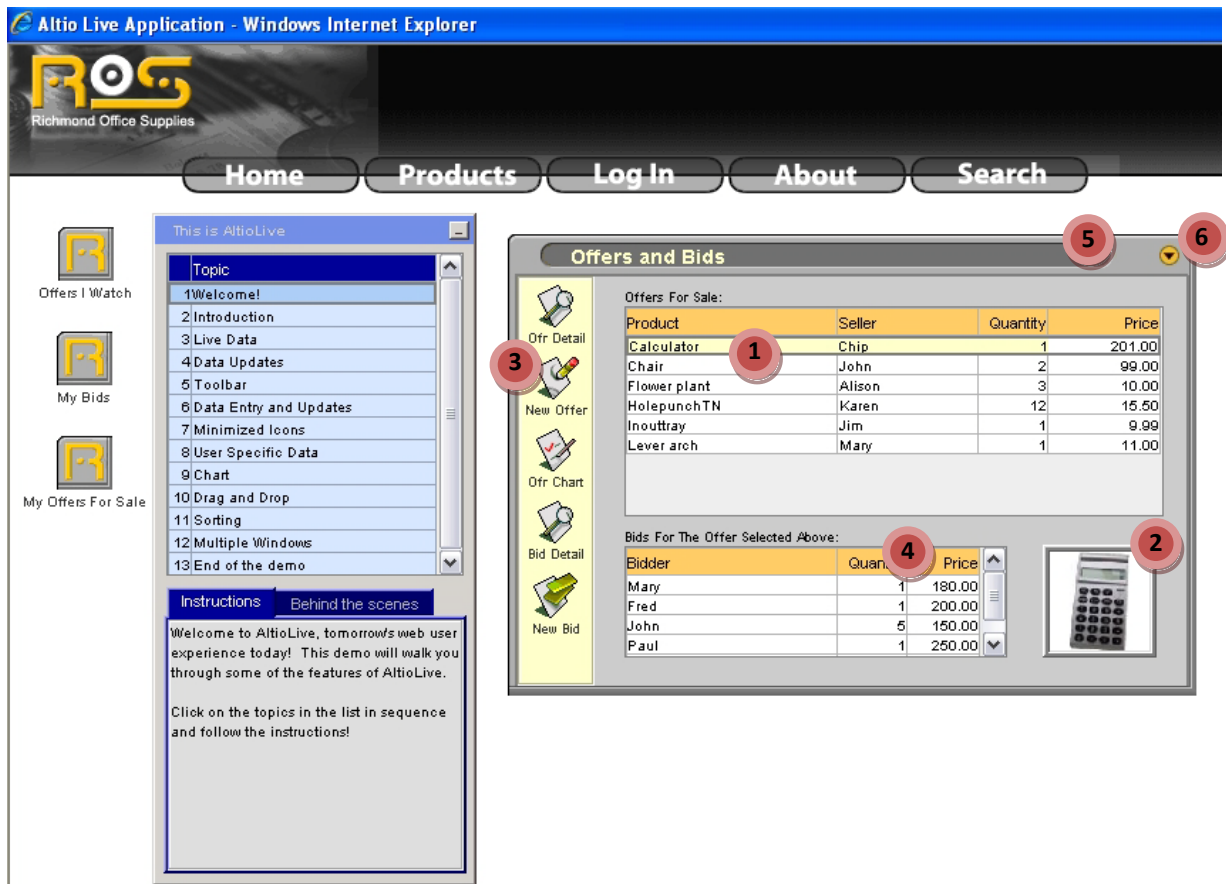
Accessing the ROS Demonstration

1. From the Altiolive Studio, expand the **Samples and Techlets** node in the **Studio Explorer** window.
2. Expand the **ROS** node.
3. Expand the **views** folder and select the **roslive.xml** file.



4. Click the run icon .

The ROS Demonstration application is run through the Smart Client:



Controls used in this application include

- Lists, **1**
- Drop-down (select) menus,
- Free text fields,
- Date pickers,
- Images, **2**
- Charts,
- Buttons, **3**
- Labels,
- Treeview, ...

Features used in this application include:

- Data filtering within a list on the client, **4**
- Moveable windows, **5**
- Minimized windows with an icon, **6**
- Scrolling through lists,
- Resizable Columns in a list by dragging column divider,
- Sorting Data in lists by clicking on column heading,
- Multiple instances of windows (e.g. Details window),
- Drag and Drop data between windows,
- Launching additional windows, **3**

- Data entry forms,
- Preservation of a user's preferred layout of the desktop when the user next logs in,
- And many more...

Important points to remember:

- Data is displayed mainly through list and form views.
- Windows can be moved, resized and minimized as though they were presented directly on a user desktop.
- Dragging new data into windows can refresh views.
- User preferences are saved between sessions.
- Since Altio maintains its own XML desktop, the performance is the same when connected 'live' to a server; no more 'click and wait.'

Further examples

Your instructor may show another demonstration at this point to highlight additional features of AltioLive applications.

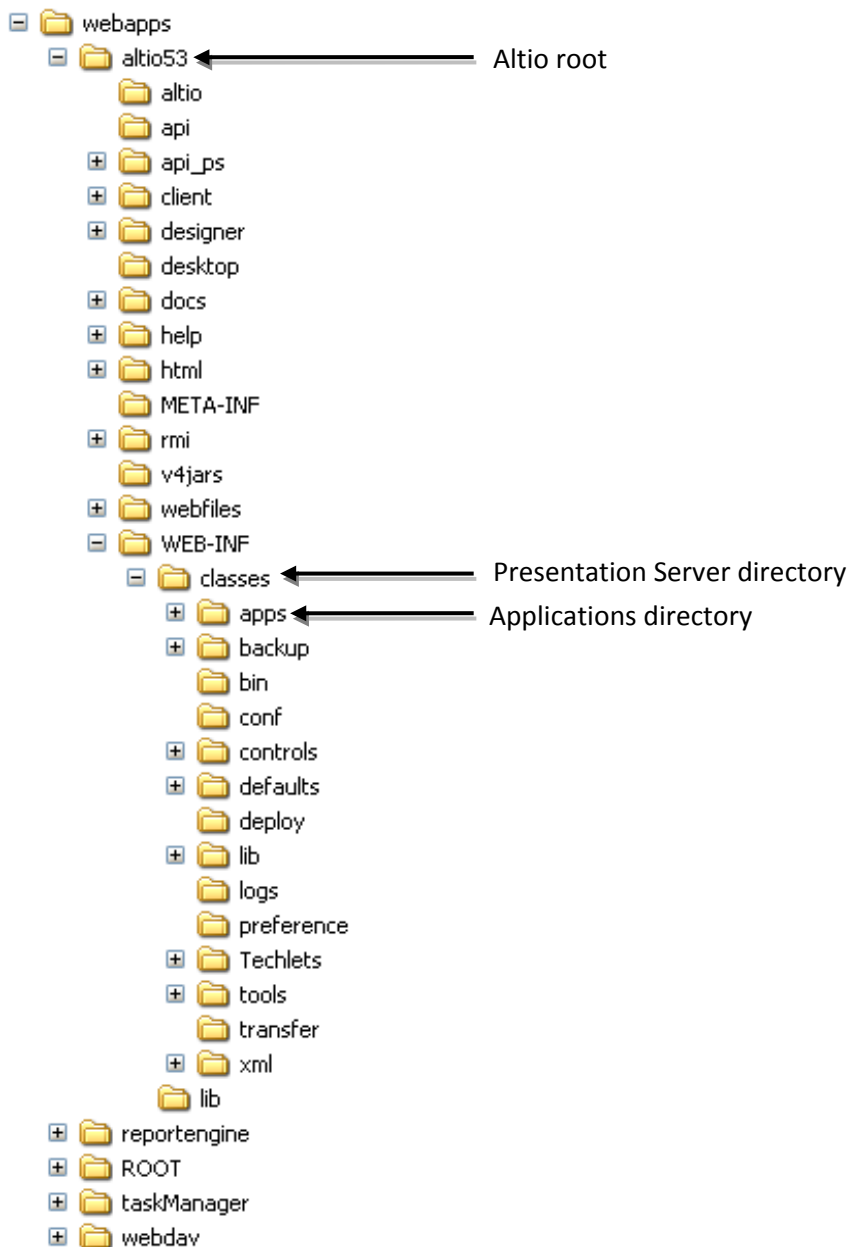
The File Structure for AltioLive

The AltioLive files are stored inside the web server public directory. The exact location of the web server files depends on the particular choice of web server installation.

The directory of most interest is **altio54/WEB-INF/classes**, referred to as the Presentation Server directory. This contains all the files for use with the AltioLive Synchronization Engine.

Please be aware that some of the files listed are part of Altio DB. The Altio DB is used to simulate behavior that would usually be part of a Back end application.

File structure below the ‘webapps’ directory



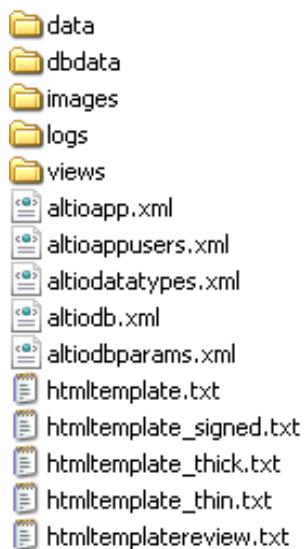
The Presentation Server directory (classes directory)

The contents of the Applications directory (apps directory)

When an application is configured using the Presentation Server Administration tool, the application ID and path are specified.

The preferred path for each application is within the **apps** directory of the Presentation Server.

Taking the stocks application as an example, the directory structure for an application is as follows:



The root of an application contains the main configuration files:

- **data directory:** holds the datapool files that ensure that all clients remain up to date.
- **dbdata directory (AltioDB only):** holds XML data files for the application accessible from the prototyping kit.
- **images directory:** contains gif and/or jpeg files relating to an application. The images can also include window skins that, once partitioned, will have an accompanying XML file.
- **logs directory:** The application logging is specified using the Application Manager. The logs directory is the default location for log files. The files may be viewed with a text editor or with the Debug Tool.
- **views directory:** several views can be configured for an application. This allows different sections of data to be made available to different users. Each **view.xml** file contains the configuration of a particular view of the stocks application in terms of windows, window styles, font styles, actions, controls, data etc... The file may include 'static-data' XML which can be useful for data driven controls. The view configuration is edited using the Designer.
- **altioapp.xml:** holds the configuration of service requests, datapools, datakeys and application logging levels for an application. The Application Manager is used to edit this file
- **altioappusers.xml:** contains a list of users associated to an application including their username, rights, associated view and html wrapper, and encoded password. This is part of the AltioLive Prototyping Kit

- **altiodatatypes.xml**: contains all the datatypes created for the application.
- **altiodb.xml (AltioDB only)**: specifies the paths to the XML data files and update files. This file is managed via the **AltioDB** button at the console (part of the AltioLive Prototyping kit)
- **altiodbparams.xml (AltioDB only)**: where an AltioDB XML file uses auto-incrementing values, this file maintains the current increment value (for each auto-incrementing attribute)
- **roshtml.txt and rospda.txt**: these are templates for an html wrapper for an application run from the Demos area of the AltioLive Studio. Other sample html template files are also generated here. These can be ignored for now.

Knowing the layout of each application directory is useful when adding existing applications to a Presentation Server and performing maintenance.

Configuration (the conf directory)

altiolicense.xml: the license file contains information required by Altio including the product's version and name, the license code and the duration of the license.

altiologin.xml: This file contains the login details of the application. This file is edited using the Login Manager tool (from the AltioLive Studio **Tools** menu).

altioserver.xml: This file contains Presentation Server configuration details including IP address of server, any slave configurations, list of applications on the Presentation Server with their App Ids, Presentation Server logging levels, configuration of user preference store. The file is edited using the Presentation Server Administration tool (from the Studio tools menu).

altiostatus.xml: gives the status of the Altio server, whether it is started or stopped and when it happened.

ias-web.xml: This file is used to connect to an IAS server.

portlet.xml: This file is used for portal servers.

Logging (the logs directory)

Logs_nnnnnn.txt: log files relating to the activity of the Presentation Server and at the levels specified in the altioserver.xml file. The files may be viewed with a text editor or viewed within Studio.

Next Step: The First Steps to build your application

AltioLive Developer Training

Document Information

KEYWORDS: INTRODUCTORY, BUILDING APPLICATION, STARTING.

Integra SP – Altio

Telephone: +44 (0) 20 8528 1045

Internet: www.altio.com

Copyright © 2010 Integra SP

Copyright in this document is vested in Integra SP. The contents of the document (wholly or in part) must not be reproduced, distributed, used or disclosed without the prior written permission of Integra SP.

Integra recognizes the trademarks or registered trademarks of any third party product or company name referenced in this document at the time of its publication.