

**Asset PT**

**An Overview**

Copyright Televisual Data Limited  
26 April 2003

Asset PT is designed to bridge the gap between off the shelf content management systems and custom developments. It is a development framework for content workflow and management applications, and is the foundation of the current Televisual Asset range of products for publishers.

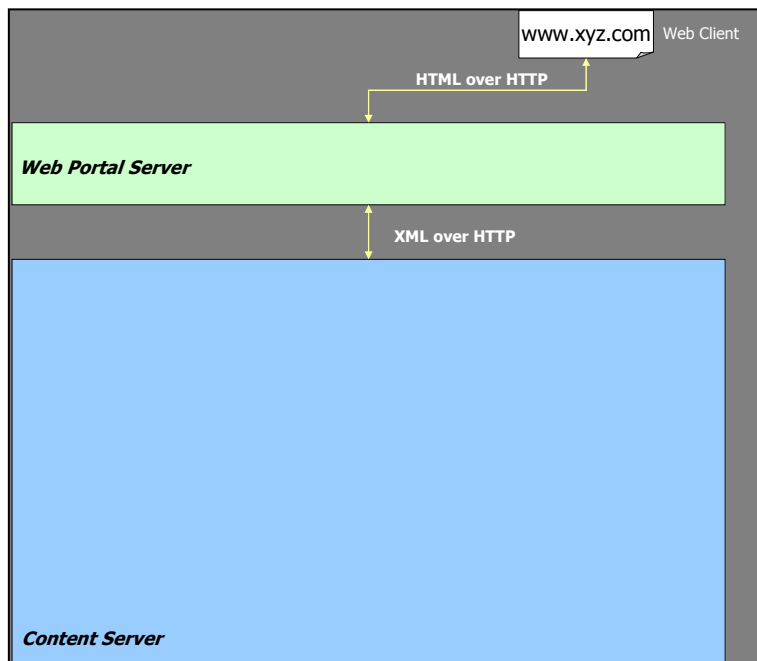
Asset PT is designed to meet three objectives:

- To minimise the cost and time to deployment of applications
- To minimise the risk of obsolescence
- To accommodate growth and evolution of applications

Key to achieving these aims are database and operating system independence. Asset PT has been built entirely in C++, without the use of any operating system specific environments such as MFC. Consequently the initial implementation on the Windows 2000 Server platform will port readily to other operating systems such as UNIX.

## The Concept

The concept of Asset PT is based on two servers: The Web Portal Server and the Content Server.



The Web Portal Server is an extension of a web service and communicates with web browser clients using HTML over HTTP.

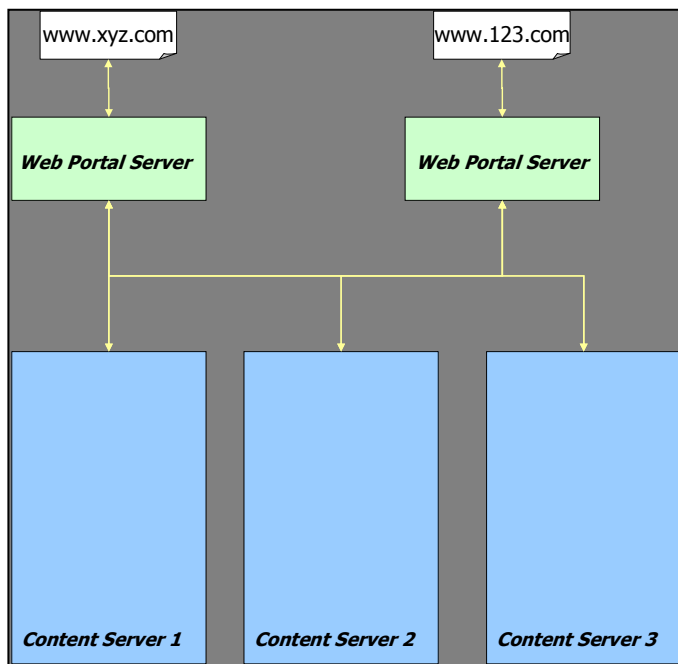
The Web Portal Server posts requests to the Content Server, which is also an extension of a web service, and gets responses as an XML stream.

Both the Web Portal Server and the Content Server can be run on the same physical server: in other words Asset PT can

be delivered as a single server solution. However, as the communication between the two servers is based on XML over HTTP, the servers can also

be in remote locations, as long as there is some form of connectivity between them.

This split server architecture also delivers one of the key strengths of Asset PT: its scalability.



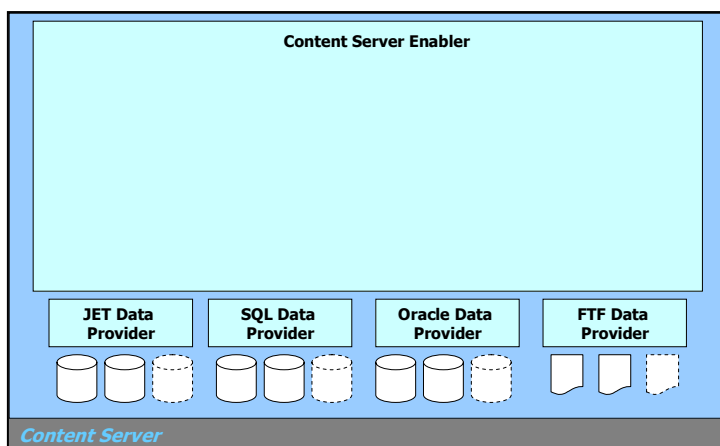
Any Web Portal Server can communicate with any number of Content Servers. This allows the construction of distributed content infrastructures, either at the instigation of a project, or by organic evolution as the corporate environment or objectives change.

Indeed, given that the Content Servers are isolated from the client browsers by the Web Portal Server, Content Servers can be located behind firewalls if desired (and the firewall is set up to accept this).

This simple architectural concept delivers real flexibility and scalability, and the infrastructure delivered by Asset PT ensures that this can be exploited in applications without the development overhead normally inherent in such distributed environments.

## Content Server

Content Server is the repository for content in Asset PT. It consists of two key components: Content Server Enabler and Data Providers.

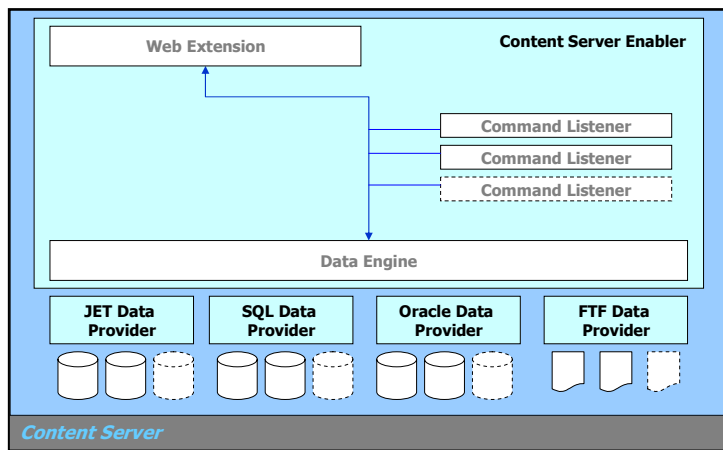


Content Server Enabler accepts HTTP posts from and dispenses XML to Web Portal Servers. It is the mechanism by which Web Portal Enablers and Content Servers can operate remotely from one another.

Content Server Enabler does not interrogate the databases directly. This is done through a Data

Provider. The Data Provider is specific to the type of database used by Asset PT. It is in essence a "driver". Isolating the database from the Content Server Enabler delivers the database independence of Asset PT. At present there are Data Providers for Microsoft JET, Microsoft SQL Server 2000, Oracle, and Televisual FTF (the format used by Televisual's free text index engine).

One Content Server can support multiple databases, and these can use the same or different Data Providers.



The Content Server Enabler also has several components. The Web Enabler is an extension to the web service on the Content server which allows clients, such as Asset Web Portals, to access the content over the Internet.

This then accesses the appropriate Data Providers through a Data Engine. The Data Engine has an API which can be used to develop server

based applications. The API is database independent, and is typically used to build applications such as bulk content loaders and database configuration tools.

The functionality of the Content Server is provided by a series of Command Listeners. These "listen" for known commands in the incoming data stream and implement them through the Data Engine. Asset PT includes a set of standard Command Listeners for functions such as QUERY, INSERT and ROLLBACK, but the major strength of this design philosophy is that custom command listeners are easy to develop and integrate. As these are implemented above the Data Engine layer, they are database independent, and if the Asset PT design guidelines are followed, will also be operating system independent.

Asset PT gives you control over which Command Listeners are active on any Content Server, and the order in which they are called. It also includes the ability to specify that a Command Listener should act "before all others" or "after all others". This is particularly useful for altering the contents of the XML stream before or after a query or insert. So, for example, it would be possible to write a Command Listener to automatically add a dialling code after every city name in a XML stream returned from a query.

## Web Portal

The function of the Web Portal is to merge simple HTML template files with XML data obtained from an Asset Content Server, or any other HTTP server that can return XML of any schema.

The strength of the web portal lies in three areas:

- It can display up-to-the-second data, but the look and feel can be easily changed by editing the HTML templates.
- One page can consolidate and display data from multiple sources
- pages can merge data from existing corporate servers and Asset PT Content Servers

It does this using a Template Merge Engine which is a web extension to the web service on the Web Portal Server. It works in a similar way to ASP, looking for the extension .TVT on pages (rather than .HTM, or in the case of ASP, .ASP). Any pages with a .TVT extension are directed by the web service to the Resolver. A TVT may request XML data from one or more servers, and all returned XML may be referenced by the dynamic set of "resolvers", which resolve simple tokens and server tags into live data.

This approach means that TVT pages can be readily built on conventional web page design programmes such as FrontPage and Dreamweaver. Equally, XSL can be used to resolve XML to HTML if desired.

The Web Portal Enabler also supports Command Listeners to allow the easy addition of customised functionality. It also provides generic XML parsing, allowing the integration of XML responses from other databases.

## Security

Asset PT has a strong integral security system. This uses a separate Asset PT database to control access to Content Servers. Each Content Server requires a security database. Multiple Content Servers can share a common security database, or can have individual ones. All posts to a Content Server require a log-on. The log-on process can be explicit or hidden. If explicit, the log-on process generates a user ticket with a defined expiry time to maintain the session. These tickets are all server side maintained, and do not require client side cookies.

Asset PT includes an encryption system that can be used to encrypt user names and passwords between Web Portal and Content Servers which do not use SSL technology.

The security system is a permissive denial system, where any permission can be explicitly granted or denied. Denials always override permissions, in a similar way to the file access security system in Microsoft Windows 2000.

Asset PT is entirely object oriented, and this of course extends to the security system. Security can be scoped at any level from server wide to a set of object.

As well as users, the Asset PT security system supports groups.

As with all Asset PT functionality, the integral security model can be extended to cover custom application specific permissions.

## The Integrated System

Drawing these elements together provides an integrated system for delivering highly scalable content management and workflow solutions.

As the full Televisual Asset product range is now built on Asset PT, it also provides rapid and easy route to add custom functionality to any Asset application.

