Customising Windows™ Applications with AppServer™

An AppSwing™ White Paper

AppSwing
Business where you want it

AppSwing Limited
Atlantic House
Imperial Way
Reading
RG2 0TD UK

info@appswing.com

Copyright

Trademarks

Overview

The need to provide remote access to key enterprise business systems is becoming more and more of an necessity as the drive for greater efficiencies requires users to spend more time out of the office and with customers.

Read this document if you need to :

- provide real-time access to business systems remotely
- tailor the screens of your systems to meet the needs of individual groups of users
- provide a corporate branding to your systems externally.

Server based computing

Any typical enterprise will have a variety of IT Systems that have evolved over time. In the early ’80s, with the advent of the PC, businesses slowly replaced dumb thin terminals with PCs to improve performance and gain end user productivity with personal applications and office tools. In the mid-’80s, client/server systems began to rapidly emerge.

In the ‘90s, the trend toward commercial off-the-shelf software dominated, firstly with the adoption of Enterprise Resource Planning (ERP) applications. These packages were intended to provide the back office applications not adequately addressed by the legacy host systems. Later came the introduction of Customer Relationship Management (CRM) applications to provide front office management applications.

Now IT is coming back a full circle with the introduction of “thin” deployments such as those provided by Server Based Computing (SBC) systems (Citrix, Terminal Services) and web based systems.

With Server based computing the application is installed on a server housed within the enterprise, rather than on the user’s PC, and is accessed via a light-weight viewer, such as Citrix’s ica client.

The Problem: Deploying only what you want

Server based computing strategies take the enterprise one stage towards providing remote access to their existing business systems. Users can access any systems that the enterprise chooses to make available from anywhere there is a suitable viewer installed.

However, if the enterprise wants to take the deployment one stage further and in addition to simply providing access to the application also:

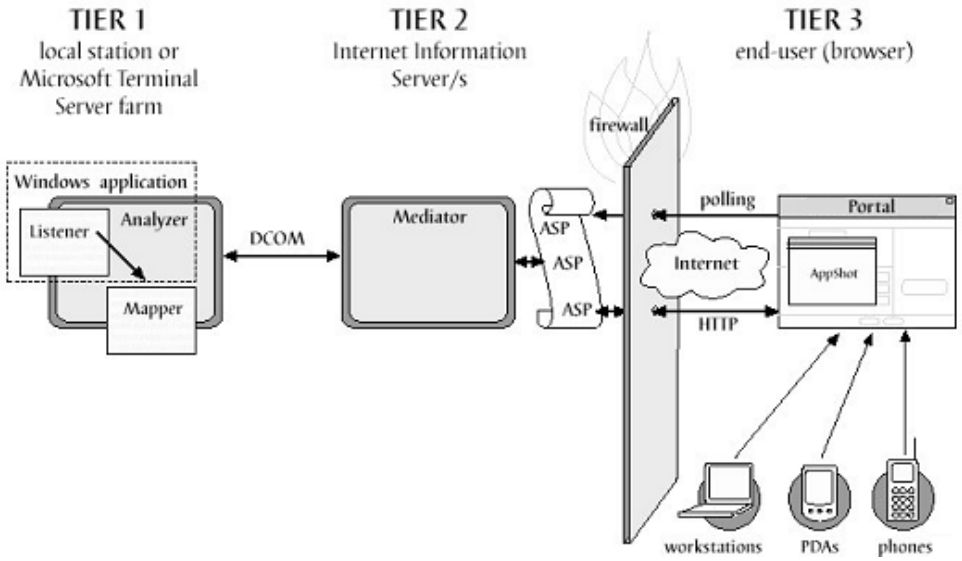
- Brand the application to reflect the corporate look and feel.
- Take a "slice" of the application and make only that available to certain user groups, rather than the whole application.
- Provide wide accessibility without the need of managing any client installations.

Faced with these challenges the business has a number of choices:

Don't provide remote access	Disadvantage vis-à-vis competitors due to absence of business agility. Increased costs arising from inefficiencies. Demotivated staff due to lack of flexibility and increased trips to the office.
Use SBC technology	Solution is only available where the appropriate viewer has been installed. Does not provide any opportunities to only provide a slice of the application or any corporate branding.
Completely redevelop legacy systems for the web	May not be an option if a 3rd party developed the application. High cost option, often costlier than initially estimated. Risky. The success of large development projects depends on many factors: personnel availability and skills, planning, management commitment, etc. Time-to-market for a large-scale project is long, thus delaying crucial business benefits.

The Solution: AppServer

AppSwing's AppServer technology creates browser-based user-interfaces that are equivalent to the Windows application user-interface, on the fly. Without writing a single line of code, a Web version of the application can be created and accessed using only a browser. The generated application is Web-natural because it is based entirely on Web-native technologies and protocols: DHTML, XML and HTTP, and does not require any client installation on the browser.



For the enterprise, AppSwing provides the solution needed to Web-enable "off-the-shelf" and "in-house" applications for the different user groups and devices that make up the extended enterprise community. This includes the ability to deploy applications to mobile devices in a form that fits the available screen space, finally making mobile application deployment an acceptable user experience.

To Independent Software Vendors (ISVs), AppSwing offers a solution for web-enabling existing applications in a low-risk, timely and cost-effective manner. Further, AppSwing lets the vendor keep the installed customer base satisfied by providing a backwards-compatible Web version without compromising the existing Windows versions. The resulting application interface can be customised to accommodate both the image and functionality requirements of a specific client, thereby opening new sales opportunities with established products.

Branding with AppServer

The AppServer product permits the enterprise to configure and customise Windows based applications for the Web. Customisation in the AppSwing context is a unique process whereby the Web-enabled user interface is redesigned to meet with customer requirements.

This affords the enterprise with the opportunity not only to stamp their corporate identity on their external systems but also to give them a fresh, web-based, look and feel (figure 1).

For enterprises this allows them to take an existing product and, through branding, apply their corporate style before making the application available through the company intranet.

For ISVs individual branded versions of the same core Windows product can be created for different customers affording the ISV additional revenue streams through the customisation process.

All this is achieved using industry standard HTML and cascading style sheet (css) technologies. Implementation of a branded interface can be a simple process for any developer with currently available web skills.

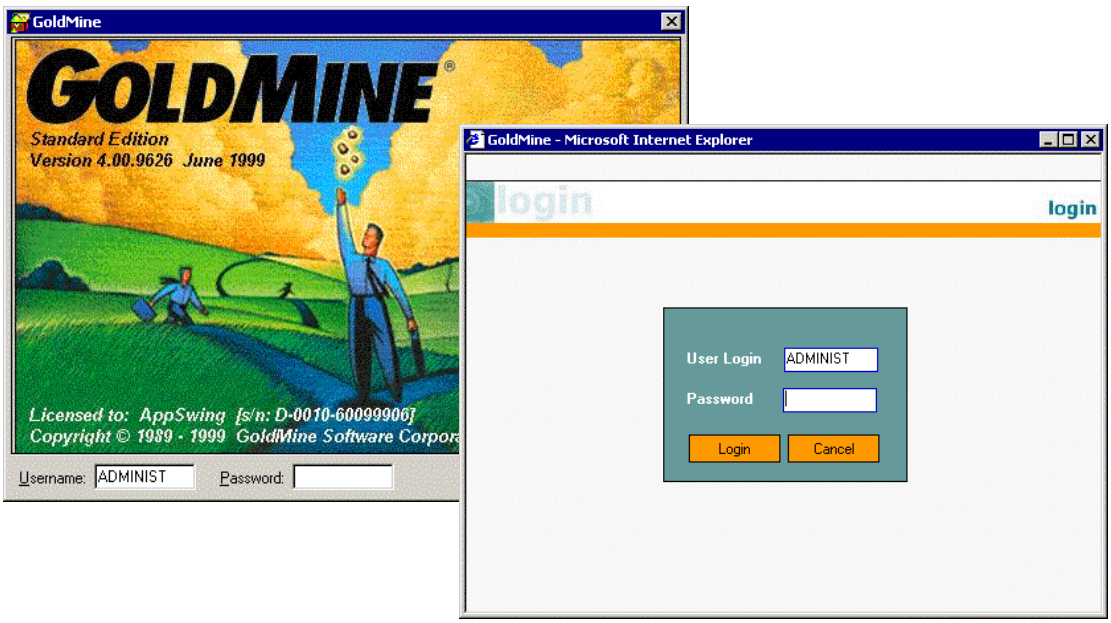


Figure 1: Application branding

Slicing with AppServer

Very few Windows applications allow the enterprise to take the existing user interface and change it so that it best fits their business. With AppServer's slicing technology the business can take their Windows applications and produce any number of user interfaces to suit their needs.

For example, a business may wish to make part of a finance system available to administration staff. Using current solutions this would mean deploying the whole application and allowing the users access to all areas of the system.

Using slicing with AppServer the enterprise can reduce the options that are available to a user by removing, buttons, tab or options on the menu, for example (figure 2).

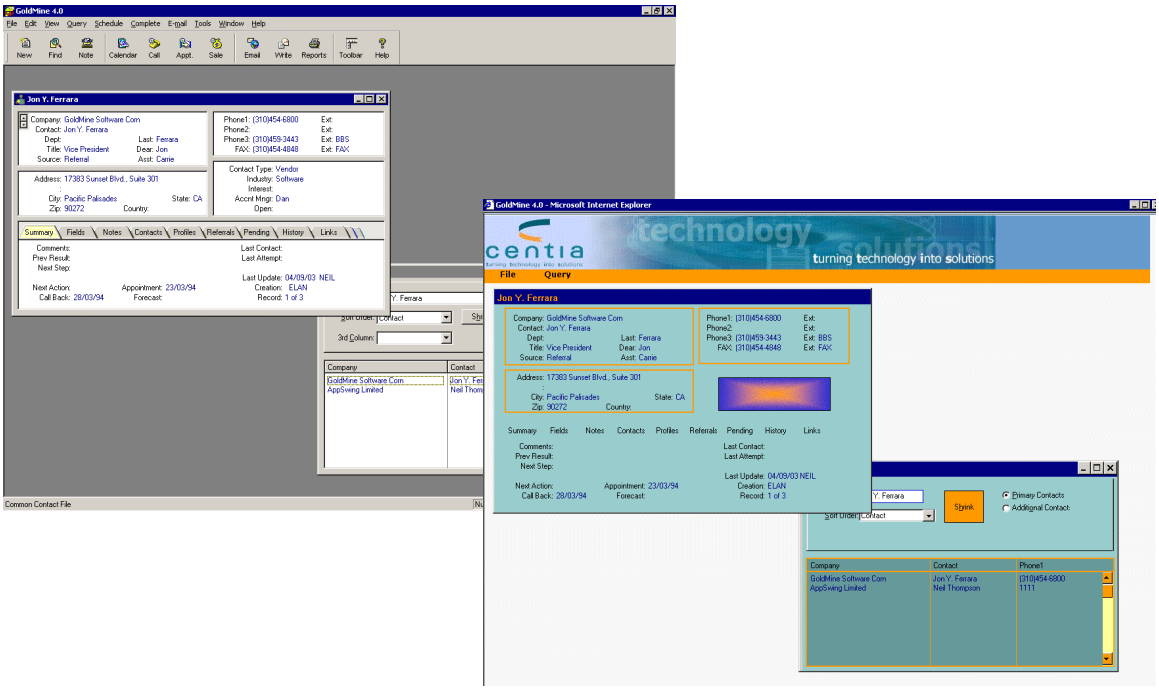


Figure 2: Application slicing

Product highlights

AppServer benefits include:

- Drive live business systems remotely
- User screens tailored to the needs of individual teams
- Only deliver the required parts of an application
- Improve service delivery and customer experience
- Reduce errors by avoiding duplication and delays
- Deliver multiple applications
- Do not interfere with existing system.
- Save time and money
- Minimise training overheads

AppServer Architecture

AppServer is installed on the same Microsoft Terminal Server as the Windows application client. AppServer includes:

- Analyser, composed of the Listener and the Mapper
- Mediator

See figure 3 below:

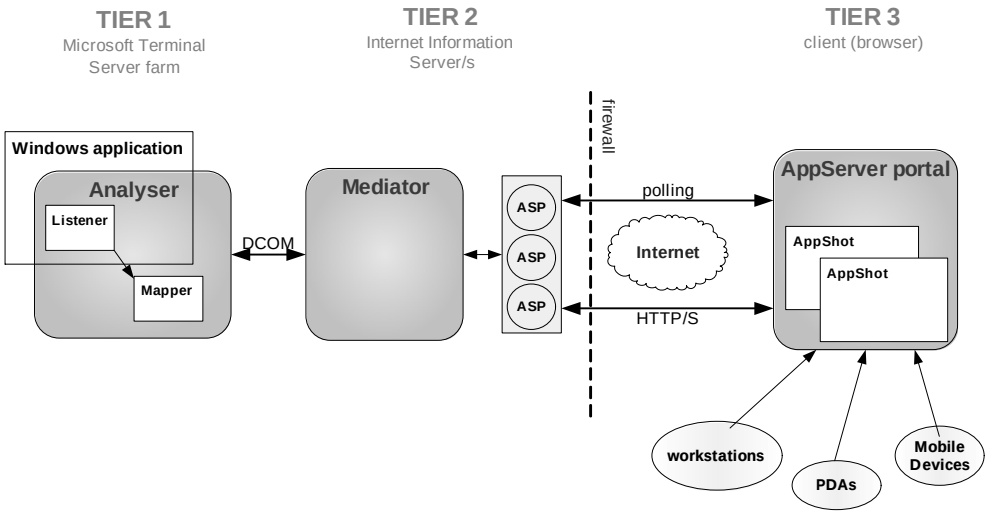


Figure 3: AppServer architecture

To start a Web-enabled application, the user opens a browser and accesses the AppServer sign-on screen. At login, the system recognises the user's Windows username password and opens the user's profiled portal page with links to the applications Web-enabled by AppServer (AppShot).

When the user selects the icon or link of the desired AppShot, a request to run the application is sent to the Mediator on the Web server. The Mediator, in turn, forwards the request to the AppManager database. Using a proprietary load-balancing algorithm, the Windows application is launched on the most appropriate Terminal Server. At this point, an Analyser starts to monitor the Windows application GUI, sending information to the Mapper where the Windows controls are mapped to DHTML elements.

The first time a Windows screen is accessed, the Mapper builds the equivalent HTML layout and saves it in the Mapper database for future use.

The Analyser sends the layouts and other information about the Windows GUI activities to the Mediator using the DCOM protocol. Communicating with the Mediator via the HTTP protocol, the browser polls the Mediator for updates. In this way, the browser transmits changes that have taken place on the AppShot due to user activities and these changes are then sent to the Analyser. At the same time, the Mediator transmits to the browser changes that have taken place in the Windows application GUI, and the AppShot is appropriately updated. Data updates are transmitted in XML format.

Competitive Advantages

AppSwing offers a unique and unmatched solution to the problem of the mobile enterprise. AppServer provides a level of performance, customisation, security and user experience not available with any alternative product or method. This is achieved in a time-scale, and for a cost, that cannot be matched by integrators or developers, ensuring that the return on investment is secured far sooner.

The following table compares AppServer to alternative techniques for deploying Windows applications remotely:

Solution	**Why AppServer?**
Redevelopment	AppServer requires no core product development, client software, business interruption or development resources. AppServer is therefore more cost effective, more time efficient, less interruptive and more flexible than redevelopment.
Remote Deployment	AppServer is able to present the user with a customised interface, designed to fit the real estate of the chosen remote device. This means no panning and scaling and the user can spend time driving the application, not searching for the area required. There is also no client software required. AppServer is therefore more flexible, easier to manage and is able to provide an unparalleled user experience when compared to the traditional “screen scraping” approach of remote deployment applications such as Citrix and GraphOn.

Conclusions

Implementing remote access solutions with AppServer gives the enterprise a single, cost-effective, solution for deploying Windows systems with the added advantages of slicing and branding.

Using AppServer the business has:

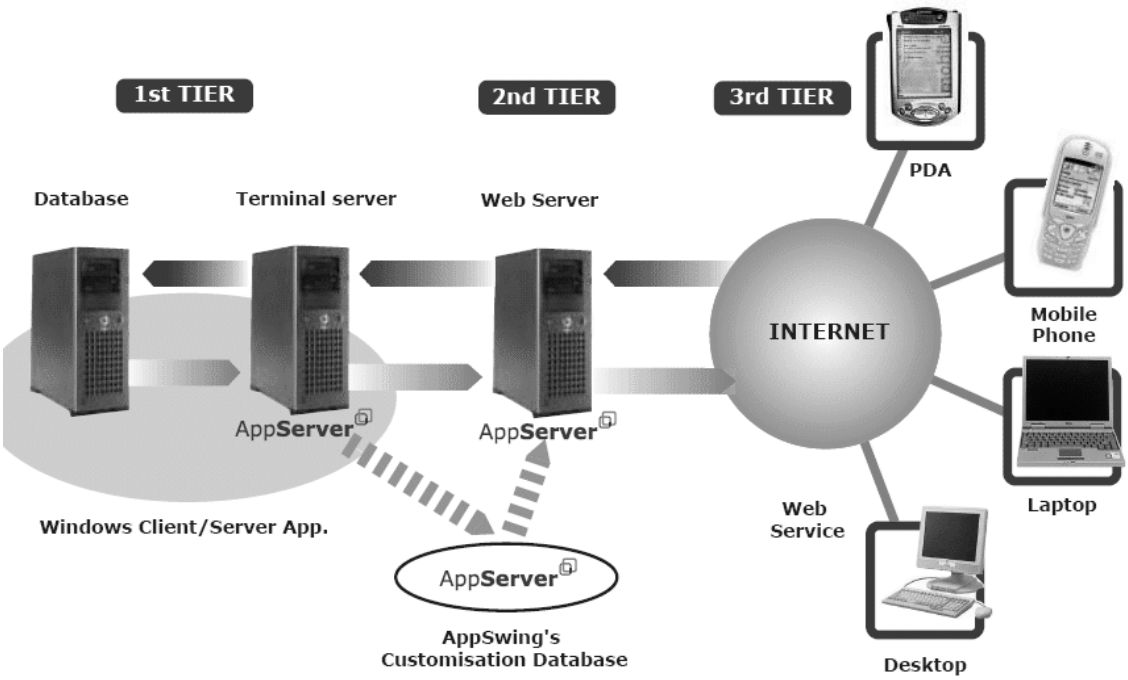
- Only one solution to implement and work with
- Chance of greater return-on-investment with the rollout of other applications.
- Opportunity to deliver only elements of original system it requires.

Furthering AppServer's ROI

Introducing AppServer to the enterprise not only allows Windows applications to be seamlessly extended to corporate portals but also extends the reach and value of existing business applications to the web and mobile devices.

With AppServer there is:

- No development overhead
- No client software to install, works with any web browser
- No interruption to the business
- No additional retraining costs for support and service teams



AppServer solutions

Web-Enablement

Deliver an HTML translation of existing Windows applications to a desktop browser without changing a single line of code.

- Provide remote workers with a truly web-enabled experience through the desktop browser on their home PC.
- Deliver sliced versions of complete applications to broaden market opportunities or control access.

Application Mobility

Rearrange the Windows application interface to fit the screen of any mobile phone or PDA and drive backend applications securely without installing any client software or any synchronisation.

- Connect mobile teams to existing office-based applications
- Choose the tablet, PDA or phone to suit your requirements.

Application Integration

Pass data from Windows applications as XML or as a web service for integration into a portal or a third party application.

- Enable staff, customers and partners to read and update core business data on-line through the company portal or web site.
- Integrate data from multiple applications for delivery in a single user interface.

Version 1.1, November 2003