APPWARES TECHNOLOGY QUESTIONS & ANSWERS

The answers to the following common questions are in most cases broken down into a short answer (A.) followed by a more complete technical answer (T.A.) with specifics.

Q. What is the IMP?

A.:

The IMP is our acronym for Integrated Media Portal – it is the engine that creates the client application running on the desktop.

T.A.:

The IMP is a compact Windows application that "plays" a datafile containing images, audio samples, webpages and script commands. It is usually delivered to a users desktop as a standard Windows installer with a custom datafile. Once installed, the IMP is associated with files having the extension .AWI (the datafile). These files can now be seperately delivered or distributed without the engine.

In effect the IMP is very mych like a browser plugin (i.e. Macromedia Flash) playing back plugin specific content (i.e. SWF files), except that it occurs outside of the browser.

Q. What can IMPs do?

A.:

Quite a bit! The IMPs are based on a collection of media assets and a scripted program all bundled up in encrypted .AWI files which are executed by the IMP engine. Whatever the programmer can imagine can be done (within the contraints of the engine).

T.A.

The IMP engine is a high-quality 2D, sprite based, graphics rendering engine that allows for the efficient creation of shaped visuals. It also incorporates an advanced, event based audio mixing engine to enhance the multimedia feel of resulting applications. The IMP applications are controlled by the internal script interpreter executing action commands and a state logic. Communication abilities are enabled through background html retrievals – in effect a scriptable but hidden web browser. The advanced media transfers between IMPs and the myApp2 backend are just one implementation using Appwares provided standard code modules. A key feature if the IMPs are the ability to treat an embedded Internet Explorer browser just like another sprite. This enables the IMPs to display and use any available web based content and plugin technology. Additionally a rich set of commands are available for: encryption, CD-ROM playback, file operations, image processing, and more.

Q.: What is actually executed on the users computer?

A.:

On the client computer the IMP executable is running for each active IMP. Additionally a service program is started when at least one IMP is running.

T.A.

On a Windows system, the IMP executable is usually stored along with required library files as IMP.exe. It is started via Desktop or Startmenu shortcut entries which are usually created by the installer.

To view an IMP application, the IMP engine is launched with a command line parameter specifying the application code and data. This is a filename of an AWI (for AppWares IMP) file which was previously created using various AppWares tools or the Producer application.

When the IMP engine starts, it checks for the service program AW_reflect – a message passing application – and launches that if it is not already running. AW_reflect terminates automatically after 2 minutes when the last IMP application has been closed.

Q. What is myApp?

A.:

myApp is web-based a backend application that facilitates network functionality such as page monitoring or media broadcasts from the server to many clients.

T.A.:

The current myApp backend implementation (Version 2) is used by many IMPs for basic, centralized functions required for many installations: unique ID generation, event polling for media delivery and software updates, page monitoring, statistics collection.

Functionality is enabled on the client side by the inclusion and activation of several standard code modules inside the IMP script. Following that, an IMP will need simply a unique name ("identity") and a server configuration for that particular identity to enable basic functions.

Media delivery to an IMP typically requires client side coding as well as backend configuration to function. Additionally media bundles have to be created using custom Appwares utilities.

Q. In what language is myApp written? Can it run on a Windows2000 server?

A.:

myApp2 is a CGI application which is based on several open source components that have proven reliable and efficient: apache, mod_perl, Mason and SQL. Windows2000 server support is technically possible but currently unsupported by Appwares.

T.A.:

Appwares has always leverages open source components for their applications to be cost effective. We started with the widely used apache webserver running on a Linux operating system using commodody PC based hardware. Then we added the very efficient mod_perl scripting environment and a proven content management system "Mason". The database interface uses basic SQL commands and we currently use the

freely available mySQL database. Other databases can be used with minimal changes to myApp provided that a perl driver for the database is available.

Currently the myApp software is designed for and runs only on the Linux (or similar Unix) operating system.

All components of myApp are available on the Windows platform however, and can be ported to the Windows platform as special request by a client. Currently we do not want to commit the required resources for supporting more than one platform for the myApp system.

Q. What are those myApp CGI components again?

A.:

apache, mod_perl, Mason.

T.A.:

The main web server is implemented using a dual-instance apache configuration with one instance handling static requests and the second instance handling dynamic requests via mod_perl and php.

The two server approach improves speed and lowers memory pressure on the operating system. The efficiency of the server is furthermore enhanced by using the mod_gzip module to provide compressed HTTP streams to web-browsers or the IMP.

The processing and scripting engine for dynamic requests is build on top of the Mason delivery engine which in turn uses the mod_perl enabled apache server. The resulting web engine is very fast as most code and pages are cached in memory, yet highly configurable as everything can be modified on the fly without restarting the server.

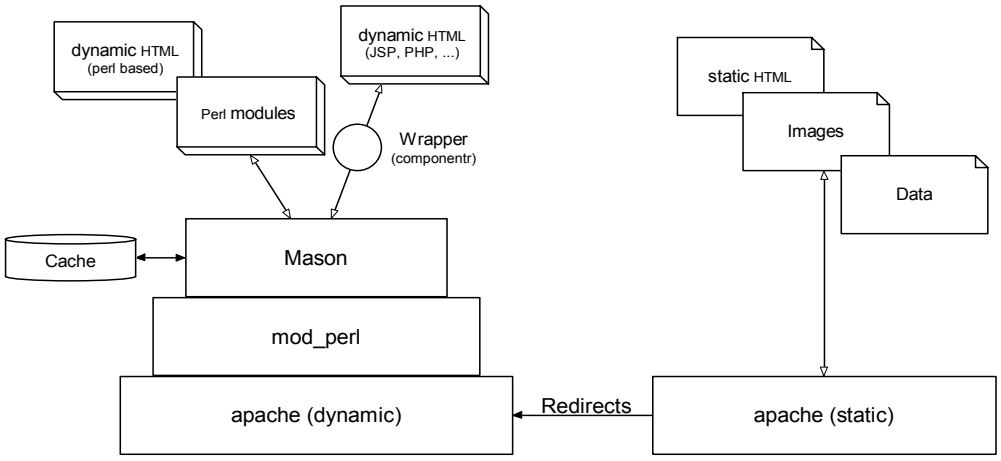
The web runtime system for myApp2 makes use of the following types of files:

- Static, unparsed HTML code from any generator or source (pages)

- Static media content such as images, audio files, etc. (data)

- Dynamic HTML code with embedded Mason/perl source (components)

- Perl-only Mason components



Q. What about using our MS SQL server as database?

A.:

This is technically possible but requires the installation and configuration of a commercial client- and server-side ODBC bridge software.

T.A.:

Event though perl provide a free ODBC bridge solution, this method is not recommended for performance reasons. Several vendors offer Linux/MS-Windows compatible ODBC bridge products. Once installed and configured, the ODBC database can be used with minimal changed to myApp.

Q.: Is myApp scalable and if so how?

A.:

myApp is scalable simply by adding more servers.

T.A.:

Since myApp is inherently simply a complex and dynamic invisible website (the IMP uses solely HTTP calls to the server) any method or technology that is applicable in creating scalable, large volume websites can be used to scale myApp2.

Typically we deploy several webservers and one database server which are virtualized for access at the server level as well as randomized at the client level.

Q.: Is the server system redundant?

A.:

Tes and No. Redundancy depends on custom implementations of the client and the server and typically a feature that has to be specially requested for an application.

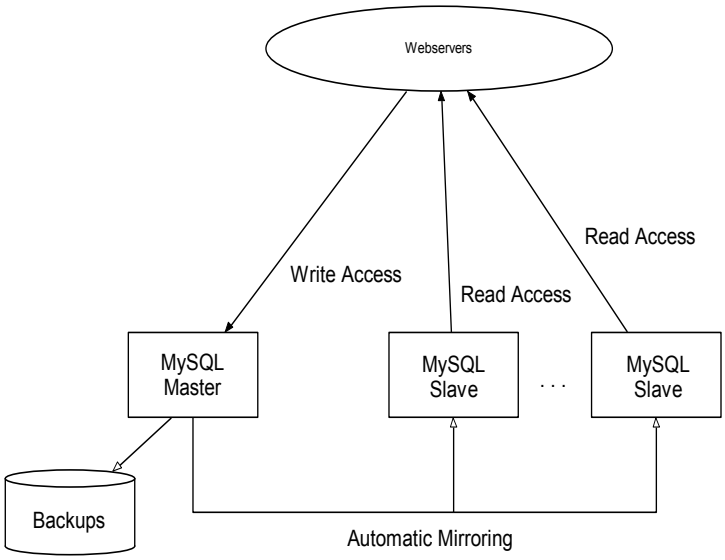
It is important to note, that the client application is normally designed in such a way, that connection outages are invisible to the user – i.e. the tool provide cached content. This is especially important for mobile workers which are not always online.

T.A.:

To provide myApp connection redundancy we employ two strategies:

- The IMP client engine can be configured to use several server URLs which are used on a random, round-robin fashion.
- If more than one server is deployed, either round-robin DNS distribution is used to spread requests on the servers, or an active virtual server using IP-tunnelling is installed (this method is typically only required for very large installations).

Database redundancy can be achieved using a mySQL master-slave configuration. Since the SQL server is not particularly taxed by the myApp application this is an effective means of providing redundancy. myApp2 code provides configuration hooks for this configuration keeping myApp services running even if the master server stops operating.



For content serving over a certain bandwidth amount (>100G) we recommend to use third party provides such as SpeedEra or Akamai which provide inhernt redundancy in their system by mirroring content over hundreds of “edge-servers” as part of their core service to the customer.

Q.: How are Appwares servers hosted?

A.:

myApp servers are hosted outside of our regular webpage and demo facilities at a leading colocation provider in Toronto, ON, Canada.

T.A.:

AppWares Development Group utilizes state of the art co-location server facilitation through Peer 1 Network. A state of the art Network Operations Centre (NOC) is the focal point of the hosting facilities activities. This exceptional facility, which is manned 24 hours per day, seven days per week by both technical staff and security, monitors the entire network and co-location space. The NOC also oversees all company systems and equipment to provide enhanced downtime protection and maximum network performance

Each facility is fully outfitted with emergency generator power that is available to all co-location customers. In addition to anti-static flooring and pre-action systems for fire suppression, each of our facilities is climate controlled using multiple air conditioning units safeguarded by emergency generator power. Our facilities also utilize distributed Uninterruptible Power Supply (UPS) so there is never a possibility of downtime due to power outages.

Implemented procedures ensure that security, as well as the network, is absolutely redundant. The Network Operations Centre (NOC) and co-location facilities are patrolled round the clock by a private security firm. Closed Circuit Television (CCTV) has been installed in the Data Centre, and all access points are monitored continuously by Network Operations staff.

AppWares Staff have unescorted, round the clock access to their equipment using personalized security access cards. Access to all doors is monitored, recorded and time stamped on a card-by-card basis. Admittance to the co-location facilities and adjacent co-location facility is through a highly secure man trap, featuring both key card and keypad access, monitored continuously by personnel and security cameras.

A high performance and redundant backbone network connects all Internet Data Centres using multiple high-speed OCn lines. With the ever-growing number of latency-sensitive applications such as Voice Over IP, Video Streaming and Conferencing, and a multitude of other client server applications, route diversity is a key component in ensuring Internet connectivity is fast and always available.

Appwares hosted servers are equipped with remote access and control facilities that allow out customer or Appwares staff to monitor status and bandwidth at all times. SNMP monitoring is used to provide accurate bandwidth aggregates for each server.

Q.: What is this server-in-a-box?

A.:

Server-in-a-Box is a server appliance, that comes complete with software and preconfigured MyApp2 Integrated Media Portal backend.

T.A.:

To enable a quick turn around for IMP/myApp based projects we have a standard server that is included in each license and can be used by the customer or hosted by Appwares.

Hardware Features

- Shuttle High Performance Small Form Factor PC
- Chassis Size 8" x 5" x 11.5"
- CPU: Pentium4 2.4GHz 533 FSB
- RAM: 1024MB DDR 333
- Harddisk: 120GB ATA100 with 8MB cache
- Ethernet: 100/10Mbps (Gigabit upgradeable)
- Storage: 32x CD-R/W drive

Preconfigured Software

- myApp2 Integrated Media Portal backend
- phpMyAdmin database administration
- RT/2 request tracker
- awstats website statistics

myApp2 Software Bundle includes

- Mandrake Linux 9.0
- Apache webserver
- MySQL database
- apache/mod_perl/Mason scripting engine

Q. We are concerned about "big brother" issues and open ports. What are your comments about this?

A.:

The IMP engine is designed to provide a secure application using several build in, standard features such as “sandboxing” file access and data encryption.

If, on the other hand, big brother means the hosting provider of the backend broadcast server than the issue really is one of trust between the sender and it's audience. As with any application that resides on the desktop and you are communicating with there are

potential issues that are addressed by corporate privacy and trust between the end user and the sender.

T.A.:

The IMP can be considered as secure as its internal source code of the Application, the InternetExplorer browser as well as its internal encryption system for source and datafiles. Below are some details on each of these areas:

- The default IMP engine uses a consumer grade encryption system that provides a industry standard 64bit code to secure most sensitive input. Encrypted are typically the IMP source code bundle, media data bundles and any executables launched from within the IMP. Control over the content production facilities for the IMP and the password used for data bundles ensures security for these types of input sources to the running IMP application. We optionally provide a custom IMP engine with higher-grade 512bit AES encryption on request.

- All web communications typically use standard HTTP transfers over port 80 (or the a proxy connection on the configured port). This transfer mode is inherently unsecure as data moves in plain text between the server or data source and the IMP. Should this be an issue, the IMP supports SSL encrypted HTTPS transfers on port 443 on request.

- The IMP is based on scripted code which allows for file operations and application launching only in the IMPs startup path ("sandboxing"). Windows registry access is limited to reading values. If security is an issue for a particular application, sensitive scripting code can be omitted from the IMP or code can be secured using additional programming logic on request.

- Since the IMP leverages the IE browser for most content, any security issues present in the browser or browser plugins can affect the IMPs security indirectly. But the IMPs usage of the browser, is no different than regular Internet surfing bu the user. We always recommend the latest version of IE to be used in conjuction with the IMP to avoid security issues.

- The IMP provides a local utility server which is locked to service only connections to IMPs and browsers running on the same machine. The ports used for this tiny utility is 51322 and 51323. Running local firewall software (i.e. ZoneAlarm) in conjuction with the IMP will alert the user of such port usage.

Q 2.: What are the issues we should be aware of when going through firewalls?

A.

The Appwares IMP uses http:// type connections on port 80. It also uses a InternetExplorer configured proxy server by default. Therefore, if a user can surf the Internet using Internet Explorer the IMP can send and receive information from the myApp backend servers.

T.A.:

The IMP will try to read the InternetExplorers proxy setup as configure in the "Internet Connection" advanced setup options. If none are set the IMP attempts to connect to the

Internet directly. If they are set, the IMP uses the first Proxy that is configured. Username and password options for the proxy configuration are supported.

As an alternative form of proxy configuration, the IMP provides command line parameters for the Proxy setup which can be enabled as required for a coporate installation. Custom installers implementing this feature can be made available on request.

Some custom configuration and firewalls may use additional automation to enable proxy access using unique Javascript configurations in a .PAC file. This is not compatible with the IMP since the IMP does not provide the necessary Javascript runtime environment to parse these files. In those cases the user must set the proxy server address and the proxy port manually.

As for restrictions on downloads of .exe files, this is strictly a site policy. We provide authenticoded installer executables for easy deployment via a webbrowser. We recommend to have users download the installer via a webpage which provides installation help and information, rather than sending executables by email which might be blocked by security software.

Q. 3: What about protection of files that are sent out – that is if we don't want them being used or distributed by the recipient?

A.

The IMP provides a secure transport mechanism but uses third part functionality to implement viewing restrictions and digital rights management.

T. A:

Incoming data bundles – although publically available on a webserver – are chunked, compressed and encrypted with proprietary Appwares algorithms and the customers password. This protects the transport part of file delivery as they have a wrapper around them which can only be opened by the specific customized IMP. Once received however, the IMP engine does extract and cache the bundle on the users harddisk in unencrypted form.

The IMP does not provide digital rights management inherently but uses the browsers mechanisms for this task. As a low security option, data bundles can be set up so that they allow for X numbers of viewings or an expiry date before they are deleted. A knowlegable user could locate a particular file if they found the appropriate directory on their system and copy it in its native format say flash or powerpoint and then make a copy to somwhere outside of the IMPs reach.

The IMP is however compatible to many DRM solutions via browser or utility plugins, which include:

- password protected office documents via Microsoft Office
- password protected PDF documents via Acrobat Reader
- DRM enabled media files via Microsoft MediaPlayer
- DRM enabled media files via RealOne RealPlayer
- DRM enabled media files protected with future technologies

Q. 4.: What technology are you using in the tool for video playback? MPEG2? Windows MediaPlayer?

A:

We provide HTML templates for audio and video playback based on an embedded Windows MediaPlayer plugin for the Internet Explorer browser. The MediaPlayer is a default installation option for any IE browser of version 5.x and up and provides automatic codec installation and full screen playback.

T.A.:

The IMP is not a media player by itself. It rather leverages the multimedia facilities of Internet Explorer which are typically installed on the users system anyhow. Media is presented as a webpage and all available web technologies for media playback can be used.

We can easily incorporate any plug in into the system such as the RealPlayer, DivX MPEG4 codec or Quicktime for high quality video. A key component is for us to easily incorporate the best video tool of the day to any specific customized application.

Additionally there we also recommend the use of Macromedia Flash for content – including audio and video presentations via the Sorenson codec – since over 95% of computer users have already the appropriate plugin installed on their system and plugin upgrades are quick and seamless.

The software requirements for the users browser can be met in several ways:

- automatic query and installation of the required plugins on the IMP download page (i.e. Flash)
- inclusion of all required plugins in the installer (i.e. DivX codec)
- inclusion of third party executables for media presentation in the installer (i.e. PowerPoint viewer)

Q 5. What part of the product does the patent or copyright apply to?

A.

The key intellectual property is our IMP engine source code protected through the copyright act of 1980. This includes the client engine, which provides all the flexibility for: application launching inside and outside of the app, content display, time event triggers, background download facilitation, encryption code, messaging services, etc.. Additionally, we own the copyright to backend server protocols and their implementation in CGI source code. We also own the source code for content prodction and packaging facilities.

T.A.:

Appware does license many part of our system to our clients: complete turnkey Producer content production system, complete API engine API and protocol, myApp2 backend including complete source code, bundling templates and encryption facilities.

Apart from Appwares owned, prorietary content, the IMP engine also uses extensively OpenSource components in the engine (several libraries) and server (operating system, webserver, database). This allows Appware to create a smaller, faster and more secure system with significanly lower development efforts, resulting in a better product at a lower cost for our customers.

Q 6. How much RAM does the IMP require?

A.:

This depends on the IMPs design and capabilities and can vary between 3-20 Mbytes.

T.A.:

The IMP core engine is a very compacy application system that uses less that 2 MBytes when running. Additionally the media assets of the application (images, audio, script) have to be held in memory resulting in a typical application size of 3MB for an IMP.

Since the IMP embeds InternetExplorer, the reported size grows by about 12 Mbytes when a mini-browser window is used inside the IMP. The thing to remember is though, that this is a virtual memory usage which is shared by other running applications. IE and its components are typically used on the system by other applications or open browser windows. Applications that use IE components are for example: Outlook, OutlookExpress and WindowsExplorer.

Extensive use of media assets such as long, high-quality audio clips and detailed animation sequences can also increase the runtime memory usage for an IMP.

Q 7.: How polite is the system to the network? What takes priority if I am downloading from several different sources?

A.:

The IMP is designed to be a “polite” system and can be extensively configure to have a low impact on the users network. The IMP does not however implement throttling or bandwidth monitoring on a byte or packet level.

T.A.:

The IMP tries to be polite using several strategies:

- The default TCP/IP transmission policy for the IMPs download engine is to use the lowest priority as provided by the network driver. This is only implemented on newer operating systems (Windows 2000, Windows XP).

- The runtime engine makes a a standard HTTP connection to the internet via a proxy server by default rather than a direct connection to the internet. This ensures proper use of cached content where facilities exist and are configured.

- Individual connections are paced in at least one second intervals allowing other applications to get full bandwidth during these pauses. It is up to the content provide to “chunk” content in appropriately sized pieces to facilitate this mode. For example a 6 MByte media bundle download can be split into 600 chunks of 10 KByte in size. The IMP would spread the download into 600 individual requests over a minimum of 10 minutes in time.

- The download engine in the IMP makes uses of chunk level and byte level restarts of downloads to avoid downloading files or parts of files twice.

- The download engine supports LZW encrypted HTTP transmissions which compress textual content by a factor of 4:1. This required a webserver with support for compressed request.

Additionally, all throttling techniques applicable to webservers can be applied to IMP transmissions. Such a configuration is available as an option for myApp2 server-in-a-box installations on request.

8. Why only 100,000 users/box?

A.:

The number 100,000 results from performance measurements on our server-in-a-box system and some assumptions on event frequency and server use for a typical IMP.

We use this empirical number as a basis for our basic licensing “block” of 100,000 active users to create as an easy to purchase, all-inclusive package for IMP deployment.

T. A.:

A single CPU webserver – such as the Appwares server-in-a-box turnkey system - can only handle a certain amount of concurrent requests as well as a certain number of requests per second. This ultimately limits the number of users a single server can support.

The size and number of media files downloaded from the server (this might include the initial download of the tool) is mostly limited by the concurrent connections the server can support, since each connection is usually longer in duration. Therefore, large volume media deliveries and high-frequency download pages are typically stored and served from content hosting providers server such as SpeedEra or Akamai.

The frequency of event polling and number of users determines the requests-per-second load on the server. A 100,000 user server might actually be able to service more users if the event polling rate is reduced to “once a day” or “on startup only”.

The server needs for each individual project has to be analyzed and assessed on a case by case bases. The Appwares server system is easily scalable by adding more servers and by using third party content providers. Appware provides consulting and documents to aid in server requirement assessments.

Q 9.: Can updates/enhancements be pushed to the tool w/o user having to delete old program and reinstall new?

A.:

Yes. Both form and functional updates can be sent to the tool via the myApp backend. The IMP needs to be server enabled and connected to the Internet for a sufficent time to receive updates.

T.A.:

Updates to the as form of an IMP typically require only the modification of the source script and media bundle. These are packaged in an encrypted source package (AWI file). These files are auto-installing and trigger an automatic application relaunch when installed. Typical file sizes for these updates are 50 – 700 KBytes.

If required for functionality or security updates, the complete IMP engine can be replaced via an encrypted executable (AWX file). A custom hidden installer is build for this purpose that reinstalles the complete IMP engine and data and relaunches the application. Typical file sizes for these updates are 1 – 2 MBytes (i.e. the same as the interactive installers).

Last updated: 14 Jan 2003