

February 2000

Prepared by Don Vickers
Applied Microsoft Technology
Group

Compaq Computer Corporation

Contents

Exchange 5.5 Outlook Web Access	3
Exchange 2000 – The Natural Internet Server	3
Exchange 2000 Outlook Web Access	3
Two Classes of Browsers – The Rich and The Reach	4
More Benefits	6
Fewer Pitfalls	6
Client Considerations	7
The Case for Rich OWA	7
The Case Against OWA	8
The Reach OWA	10
The Middle Ground	10
OWA and the Exchange 2000 Architecture	11
Web Store	12
Server Considerations	14
Installation	14
Integration with IIS	15
Administration	16
Number of Concurrent Clients	17
Front-End/Back-End Servers	18
Access Security	19
OWA as a Toolkit for Extensions	20
OWA as the Stateless Future Client	23
Appendix A: Related Documents	24
Compaq ActiveAnswers	24
Compaq White Paper Index	24
Compaq TechNote Index	24
Microsoft Exchange Server Web site	24

Exchange 2000: Outlook Web Access

Abstract: Selecting the client software for Exchange has always been a challenge. Outlook Web Access (OWA) is a most attractive client for access Exchange since it requires virtually no deployment effort. Previous versions of OWA have carried a number of penalties, including lack of scalability, features and flexibility. This paper investigates the next generation of OWA, which is a standard component of Exchange 2000 Server. This paper reviews the features available to different families of web browsers, the interaction between Exchange 2000 and Microsoft Internet Information Server (IIS), the internal OWA architecture, and how the Web Store and OWA can be used to build a custom web browser environment for clients.

The paper concludes with a review of the role of OWA as a stateless client for Exchange 2000 and the implications this client provides to the future evolution for messaging clients.

Notice

The information in this publication is confidential and proprietary to Compaq and is protected by the terms of an end-user license agreement. The information in this publication is subject to change without notice and is provided "AS IS" WITHOUT WARRANTY OF ANY KIND. THE ENTIRE RISK ARISING OUT OF THE USE OF THIS INFORMATION REMAINS WITH RECIPIENT. IN NO EVENT SHALL COMPAQ BE LIABLE FOR ANY DIRECT, CONSEQUENTIAL, INCIDENTAL, SPECIAL, PUNITIVE OR OTHER DAMAGES WHATSOEVER (INCLUDING WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION OR LOSS OF BUSINESS INFORMATION), EVEN IF COMPAQ HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The limited warranties for Compaq products are exclusively set forth in the documentation accompanying such products. Nothing herein should be construed as constituting a further or additional warranty.

This publication does not constitute an endorsement of the product or products that were tested. The configuration or configurations tested or described may or may not be the only available solution. This test is not a determination of product quality or correctness, nor does it enable compliance with any federal state or local requirements.

Compaq, Deskpro, Fastart, Compaq Insight Manager, Systempro, Systempro/LT, ProLiant, ROMPaq, QVision, SmartStart, NetFlex, QuickFind, PaqFax, and Prosignia are registered with the United States Patent and Trademark Office.

Netelligent, Systempro/XL, SoftPaq, QuickBlank, and QuickLock are trademarks and/or service marks of Compaq Computer Corporation.

Microsoft, MS-DOS, Windows, Windows NT and BackOffice are trademarks and/or registered trademarks of Microsoft Corporation.

Other product names mentioned herein may be trademarks and/or registered trademarks of their respective companies.

Copyright ©2000 Compaq Computer Corporation. All rights reserved. Printed in the U.S.A.

Exchange 5.5 Outlook Web Access

Outlook Web Access (OWA) is a useful component of Microsoft Exchange. It offers the ability for clients to easily access all the information in Exchange in a simple and easy manner. OWA has the great advantage that it can be deployed on any device that supports a modern web browser. This means that OWA can run on virtually any personal computer with absolutely no intervention by an information technology person. More importantly, OWA provides access to virtually all the information available through Outlook 2000, which is still the premier client for Exchange.

The current generation of OWA is limited by its architecture and implementation. Not only must OWA be installed and managed separately from Exchange, but also it does not have the ability to scale well. The architecture of being dependent on the Messaging API (MAPI) and Active Server Pages (ASP) requires that virtually all content be rendered on the OWA server, and this limits the number of concurrent OWA client connections. The interface between the OWA server and the Exchange server also limits the effective number of clients due to the use of MAPI RPCs to carry data between the IIS application and Exchange. Even with high-powered servers, the maximum number of concurrent connections can rarely exceed 700 to 800 OWA clients per server. In addition, the current generation of OWA only approximates the user interface of Outlook.

OWA in Exchange 5.5 is attractive for organizations that wish to have a client that provides access to the broad range of information stored in Exchange. The price of high server and network load is often higher than is desirable.

Exchange 2000 – The Natural Internet Server

Exchange 2000 is the next generation in the evolution of Exchange. Exchange 2000 is the first generation of Exchange to be designed from the beginning with the Internet in mind, and the architecture reflects this. IIS is now an integral part of Exchange 2000. IIS acts as the Internet protocol processor for Exchange 2000 and Exchange 2000, through its powerful and efficient Web Store acts as the best storage location for IIS. The Web Store is the most compelling new feature in Exchange 2000, and will be discussed in detail later. The Web Store provides an Exchange store that is accessible from more directions and more efficiently, with simple consistent URLs mapping every object. These significant changes will allow Exchange 2000 to better operate in the Internet generation. However, very few of the innumerable changes in Exchange 2000 are visible to the Outlook client. Someone using Outlook with Exchange 2000 will notice very little difference when they are moved from their old Exchange 5.5 server. This will certainly change as new versions of Office build on the foundation laid by Exchange 2000.

Exchange 2000 Outlook Web Access

Clients using OWA 2000 will most certainly see a huge difference when their Exchange server is upgraded to Exchange 2000. As we will see, OWA 2000 closes virtually all the gaps that existed in OWA 5.5. Almost every architectural change in Exchange 2000 directly or indirectly improves the effectiveness and efficiency of OWA. OWA 2000 uses the HTTP/DAV, XML, XSL and DHTML protocols where possible to provide improved service and performance, as discussed below.

Two Classes of Browsers – The Rich and The Reach

OWA 2000 introduces two levels of behavior. When OWA discovers a rich browser it utilizes the facilities of the rich client. These facilities provide a far better environment both for the client and for server and network load.

A rich browser is one that supports features such as XML, XSL, DHTML and HTTP/DAV. The only browser that is currently considered rich is Internet Explorer 5 (IE5), running on Windows systems. IE5 on the Macintosh does not support XML and HTTP/DAV, so it is a reach browser.

A rich browser provides a rich client experience while the reach browser allows Exchange to reach out to virtually any browser-equipped device.

Reach OWA occurs when OWA runs on a browser that is not rich. This includes every browser except IE5 on a Windows platform. The behavior of reach OWA is less smooth and appealing than rich OWA. Reach OWA uses standard HTML features such as frames and forms to provide the user interface, so it cannot approximate the user interface of Outlook as well as rich OWA. The reach experience is very similar to the OWA 5.5 experience.

When OWA discovers a reach client, through the user agent component of the HTML header, it renders the desired page in HTML 3.2 before sending it to the client for display purposes.

The Rich Browser

Figure 1 shows Exchange 2000 OWA in a rich browser. A static screen shot illustrates some of the major features of the rich browser, but does not show the behavior. The screen shot shows that the standard Outlook appearance is preserved almost perfectly:

- Standard buttons with options
- Standard panes
 - Outlook shortcuts or folder trees
 - Item lists
 - An optional preview pane

More importantly, the behavior of Outlook is also closely matched. For example:

- Buttons operate in the same way
- Keystrokes such as CTRL-K are supported
- Auto address completion is provided
- Single and double mouse clicks operate as in Outlook
- You can drag and drop within the browser
- Context sensitive drop-down menus
- Column can be sorted
- Rich views are provided, including expandable group type views
- Embedded objects are rendered properly
- Public folders containing contact and calendar items are supported
- Multimedia messages are supported

The most significant benefit of having the more powerful rich browser environment is the more complete involvement of the client in the client/server partnership. The use of the XML, DHTML and XSL protocols allow data for each object to be transmitted to the browser where it then renders the objects for the client, based on their selections. The client can select, manipulate and reorder the objects using simple, obvious interface controls. No communication with the server is required. This not only makes the client experience much closer to the Outlook experience, but also significantly reduces the network and server load.

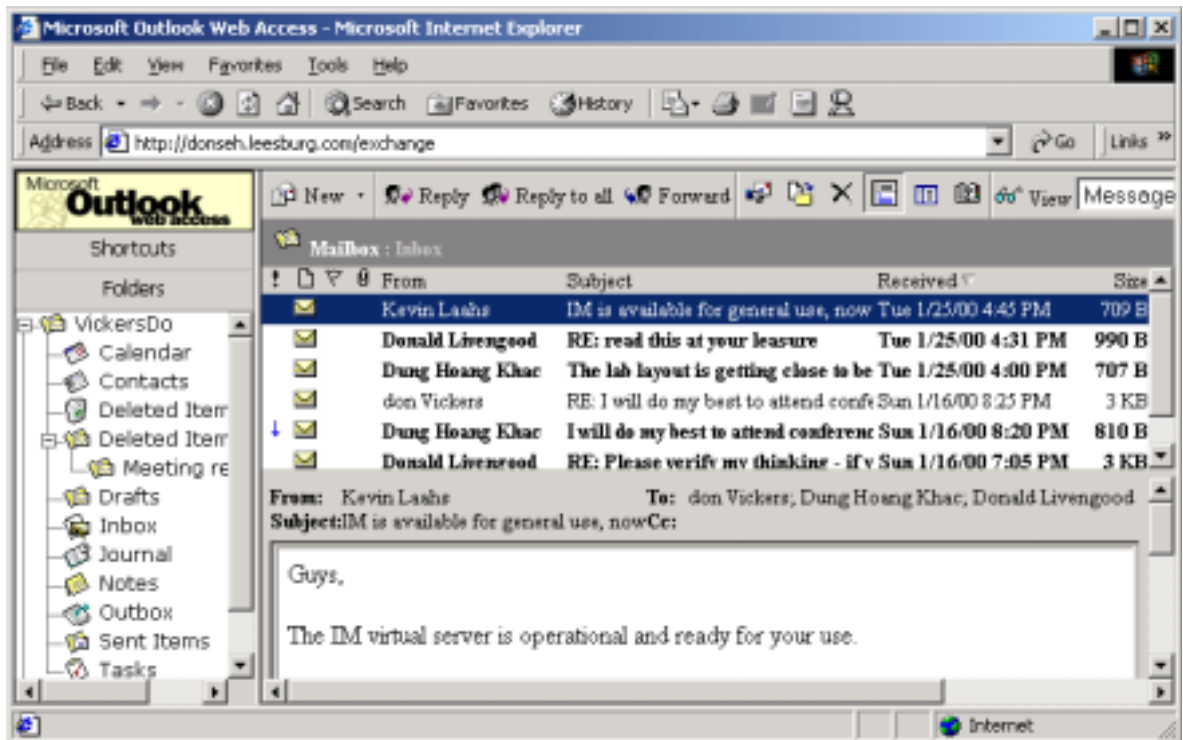


Figure 1 - OWA in a Rich Browser

The Reach Browser

Figure 2 shows the same information presented in a reach browser, Netscape Navigator 4.04. The display is far less complete and rich than with the rich browser. The appearance and behavior is almost identical to Exchange 5.5 OWA. The buttons are a subset of those available in Outlook and the rich browser. The reach browser presents a simple frame-based set of panes less featured than those available in the rich browser and Outlook. Selecting an item to process requires selecting its check box before clicking the appropriate button. Not only does this require extra work for the user, but significantly more network and server activity is used than with the rich browser. The reach browser does not support most of the other features, for example, direct mouse actions, drag and drop, drop-down menus and keystrokes.

While OWA 2000 operates with virtually any graphical browser, it only supports browsers that are fully compliant with the HTML 3.2 and European Computer Manufacturers Association (ECMA) script standards. As earlier browsers are not fully compliant with these standards, Microsoft recommends using Microsoft Internet Explorer 4.0 or later or Netscape Navigator 4.0 or later. These browsers have been tested with Outlook Web Access. As is discussed in this paper, Internet

Explorer 5 provides the best OWA environment. Both Compaq and Microsoft recommend the use of Internet Explorer 5.0 where possible to provide the best environment.

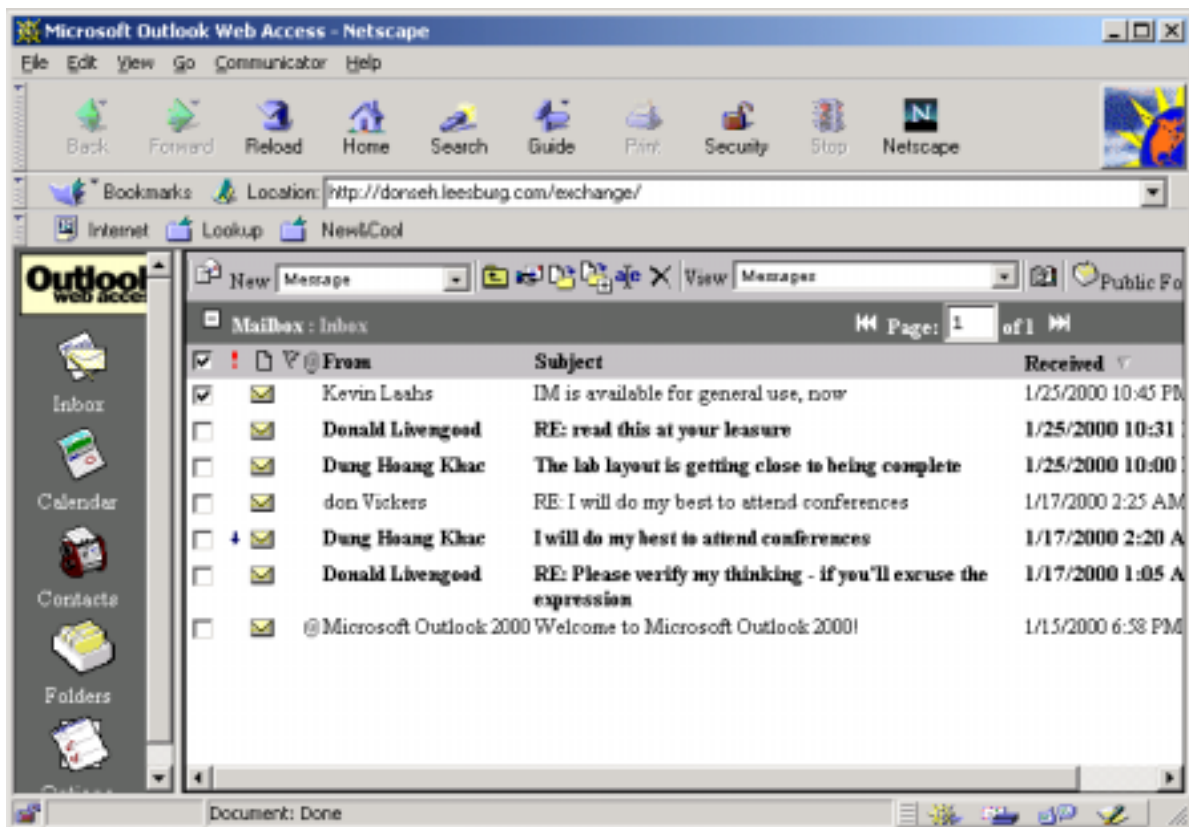


Figure 2 - Reach browser

More Benefits

Exchange 2000 OWA expands on the strengths of previous versions of OWA. Exchange 2000 OWA is significant enough to justify calling it a new generation. As we have discussed, the rich browser provides a very close approximation to Outlook 2000 on a personal computer. The look and behavior of OWA in a rich browser is close enough to Outlook that virtually no training is required. OWA is just as easy to deploy on the client device as previous versions. That is, there is no client installation for any version.

Fewer Pitfalls

The worst feature of previous versions of OWA was its inability to scale well. Exchange 2000 OWA solves the scaling problem by having a more complete integration between Exchange and IIS. This is achieved by using the new Internet protocols to offload work to the client. This dramatically reduces the use of Active Server Pages, which perform virtually all server side processing via DLLs. It is too soon to know how many clients Exchange 2000 OWA can support, but it is safe to say the number will be several times the current 700 to 800 clients. The number of clients supported is discussed later in this paper. Being an integral part of the Exchange 2000 Server reduces the administrative effort and complexity. Customizing OWA in previous versions required fairly simple text editing of ASP files. However, the lack of any automated version control or merge of

customizations with standard code meant that customizations were subject to being overwritten with every new release or service pack of Exchange. OWA 2000 provides a far more rational environment for extensions and programming as discussed below.

Client Considerations

Exchange offers service to a wide range of client types. OWA is ideal for clients who require or want to have the smallest possible support effort in order to gain access to Exchange. OWA is ideal for people who operate in environments like education, ISPs, ASPs. These types of environments have a high ratio of clients to support staff, which can be due to economics, low service expectations by the clients, or a separation of the clients' environment from the Exchange organization's environment. OWA has virtually no impact on the client device so customer support is significantly reduced compared to MAPI based clients.

OWA is meant to be a secondary client for accessing Exchange information in an enterprise environment. OWA allows someone whose primary access is from a Windows desktop using Outlook to easily access their information from other places and devices. This could include access from a public place where only a web browser is available.

OWA does offer some very exciting and compelling features and benefits, and some organizations may be tempted to use it in place of Outlook. These benefits include having zero impact on the client device used, and almost total access to the information in Exchange. In general, replacing Outlook with OWA in an enterprise environment is not the best approach since Outlook has features that make it much richer than OWA, as discussed below

In the next section, we will examine the case for and against OWA 2000 to better understand the features and benefits of the two client approaches.

The Case for Rich OWA

As discussed above, the rich OWA provides an environment that very closely emulates Outlook. Rich OWA is very attractive, especially as a secondary client to Outlook.

Low Cost and Effort Desktop Deployment

OWA runs completely inside the client web browser. There is absolutely no setup or configuration required for the client. Once a browser, preferably IE5, is installed on the client hardware the only setup requirements are that the client know the URL and access information to reach their mailbox. One of the most painful parts for most Outlook clients on a new PC is to setup their MAPI profile. OWA eliminates this painful process.

Access to all Exchange information

OWA provides access to all information in Exchange including electronic mail, public folders, contacts and calendars. No other client other than Outlook provides the breadth of information access provided by OWA.

Looks, Behaves and Acts like Outlook

To a casual client, OWA 2000 is virtually the same as Outlook. OWA 2000 supports virtually every button, keystroke and operation in Outlook. There are only a few key features missing, such as spell checking and off-line access.

Goes Through Firewalls in a Single Port or Two

OWA is built with standard industry Internet protocols such as HTML/DAV and XML. All of these protocols use HTTP, normally through port 80. If additional security is required then the SSL port, normally port 443, can also be opened.

Moderate Network and Server Load

OWA 2000 reduces the server and network load due to the significantly improved Exchange 2000 architecture and the improved client processing offered via XML and DHTML. Clients operating in a rich browser environment will have virtually no client/server traffic caused by message and hierarchy navigation as the information is stored on the client via XML. The actual rendering is also performed on the client as much as possible to further reduce server load.

Some organizations considered using OWA in low bandwidth situations only to discover that OWA actually had a higher network load than MAPI. OWA 2000 in a rich environment has approximately the same overall network traffic as MAPI, but since it uses HTTP OWA is able to survive the high latency encountered in low bandwidth environments with better service to the clients.

The Case Against OWA

As powerful and as fast as OWA 2000 is, it actually has some deficiencies. The impact of these deficiencies depends very much on the needs of the clients in organizations that deploy Exchange 2000.

New Features Require IE5

Virtually all the exciting features discussed require the use of a rich browser to receive either full or partial benefits. The reach browsers derive some benefits from the many improvements, but results for the clients are far less compelling and exciting. The appearance and operation of the reach browser is not nearly as smooth and responsive as the rich browser. The reach browser is also much less like Outlook in appearance.

Deploying IE5 may appear to be easy from the Microsoft perspective. However, even when desktops are running Windows, the efforts required to deploy applications to them can be daunting. In reality, many desktops do not run Windows so they will be forced to remain as reach clients.

Some Fundamental Outlook Features are Missing

Outlook 2000 is still the top client family for Exchange 2000. Of any client, Outlook provides the widest range of features and functions for interacting with Exchange. Outlook provides an excellent desktop portal into enterprise information systems. The Digital Dashboard is an example of what can be implemented using the very rich environment offered by Outlook.

Offline Operation

Most large enterprises have a significant number of clients who travel and require access to their Exchange information while not directly connected to the server. This travel may be business travel to various corners of the planet or just between the office and home. Outlook's ability to support offline working is invaluable in this environment. OWA currently has no way to work when not directly connected to the server.

1Local Storage

There is a constant battle between server administrators and the clients who utilize the servers. Administrators strive to keep the server working smoothly and reduce the space required by the store. The clients try to have access to every message and document that might someday be important. Virtually every Exchange site implements mailbox size quotas. Personal storage files (PSTs) provide a way to satisfy the needs of both groups. They provide additional storage for items that don't fit into a quota-restricted mailbox. PST files are a useful feature of Outlook, but are not available in OWA.

2Secure Messages

Outlook supports Public Key Infrastructure (PKI) security so that messages can be encrypted for privacy and non-repudiation. OWA does not support this feature. An interesting but useless observation is that security seems to be a pressing issue only when a system is being selected and when security breaches are in the news.

3Spell Checking

Many of us are horrible spellers. Outlook provides spell checking facilities, including the ability to automatically spell check a message before it is sent. OWA has no spell checking facility.

4Access to Rules

Rules are powerful ways to process incoming messages. Outlook has the ability to maintain rules for both the server and client. OWA has no access to either set of rules, even though any server side rules are executed regardless of client. There are no client side rules available to OWA and no way for a person who uses only OWA to manipulate server rules.

OWA 2000 does have an Out of Office Assistant option, which sets a rule in this situation.

5Directory Browsing is Incomplete

OWA uses LDAP to access the Global Address List. LDAP access is fast and efficient, but it is based on the concept of searching rather than browsing. This rules out the ability to find a close match and then browse through a list of names to locate the desired individual. MAPI-based clients like Outlook support browse functionality.

6No Tasks or Journal

Currently, OWA 2000 has no access to the Tasks or Journal stored in Exchange.

7No Reminders

OWA 2000 does not provide reminders.

8No Copy between Public Folders and Mailboxes

OWA 2000 does not support copy operations between public folders and folders in mailboxes.

9Reduced Features in Contacts

OWA 2000 does not support the telephony options and user-defined fields when accessing contacts.

10No Time-Based Features in Mail

OWA 2000 does not support "Do not deliver before," and expiration options.

11No Direct Editing in Calendar view

OWA 2000 does not support direct editing in the Calendar view.

The Reach OWA

The concept of reach OWA is that it allows OWA to reach client devices that are not enriched with a rich browser. This reach allows Exchange to extend to a virtually unlimited variety of systems from UNIX to the Macintosh.

The reach OWA may be the more important aspect, even if it is not as exciting and compelling from a usage perspective. Web browsers are appearing on more and more devices. The current generation of OWA can be viewed from cellular telephones and other very mobile devices. The basis for reaching so many devices under which there is no control, is to reach for the lowest common denominator.

OWA uses two frames and standard HTML V3.2 to interface with reach browsers. As noted, this does not give the smooth and rich interface of the rich browser and requires significantly more network round trips due to the lack of XML and DHTML. The good news is that the use of the server-side DLLs and a tighter coupling of IIS and Exchange 2000 Server reduces the server load.

The Middle Ground

As with most things in life, the best course is often to take a moderate direction and stay in the middle ground. OWA offers a powerful middle ground for client access to Exchange 2000. There are two ways of doing this. The first is that OWA can operate in the middle ground between the rich and heavy weight MAPI protocol at the top, and the more lightweight protocols of POP and IMAP at the bottom. The second meaning is that there is a middle ground in regard to being open to deploying both a primary and secondary client.

OWA offers much of the power of the richest client, Outlook 2000, yet has deployment costs very close to the POP and IMAP protocols. Selecting client software requires decisions based on many factors, starting with the business needs of the clients, including server and network load, and the actual client devices to be used and supported. In most enterprise oriented Exchange based organizations there will be a need for multiple clients, depending on the factors in play.

Due to the many improvements in OWA 2000, it may be tempting for some organizations to move to OWA 2000 in place of Outlook. This can be an improvement over POP and IMAP but may be a step back compared to Outlook. The POP and IMAP protocols offer limited access to Exchange; being limited to just electronic mail messages. OWA offers enrichment compared to these protocols. The rich browser implementation of OWA offers a network and server load very close to the POP and IMAP protocols so there can be a net gain for the organization. Many organizations that are using Outlook are using Outlook as a key component in their overall information management toolset. Outlook provides access and features not found in OWA 2000, as discussed above. ASPs may find that OWA 2000 gives them a very good way to deliver a significant amount of information to their customers at a moderately low cost.

One approach for enterprise organizations with clients that require multiple devices, is to have Outlook on each device. This carries the cost of deploying and maintaining multiple instances of Outlook. While this can be cost effective for power clients, but may be inefficient and painful for clients who access their Exchange information infrequently from 'remote' locations. These clients can be better served by using OWA as their secondary client of choice, with Outlook remaining the primary client. OWA 2000 works very well as a secondary client.

The table below provides a summary comparison of the features supported by the three richest Exchange 2000 clients and the OWA 5.5 client:

Feature	Outlook 2000	OWA 2000 Rich	OWA 2000 Reach	OWA 5.5
Electronic mail	√	√	√	√
Calendaring	√	√	√	√
Contacts	√	√	√	√
Rich text	√	√	√	√
HTML text	√	√	√	
Drag-and-drop editing	√	√		
Shortcut menus	√	√		
Shortcut keys	√	√		
Preview pane	√	√		
Folder tree	√	√		
Multimedia	√	√		
Embedded objects	√	√		
Offline use	√			
Spell checker	√			
Outlook rules	√			
Journal	√			
Printing templates	√			
Timed delivery	√			
Expiration	√			
Tasks	√			
Reminders	√			
Front-end/Back-end servers		√	√	

Table 1 Client feature comparisons

OWA and the Exchange 2000 Architecture

As we have discussed, OWA benefits from virtually every change made in the Exchange 2000 Server architecture. By far, the greatest benefit to OWA 2000 is the tight integration between IIS and the Exchange 2000 Server. This allows for Internet protocol processing directly into IIS via the powerful and extensible Exchange store.

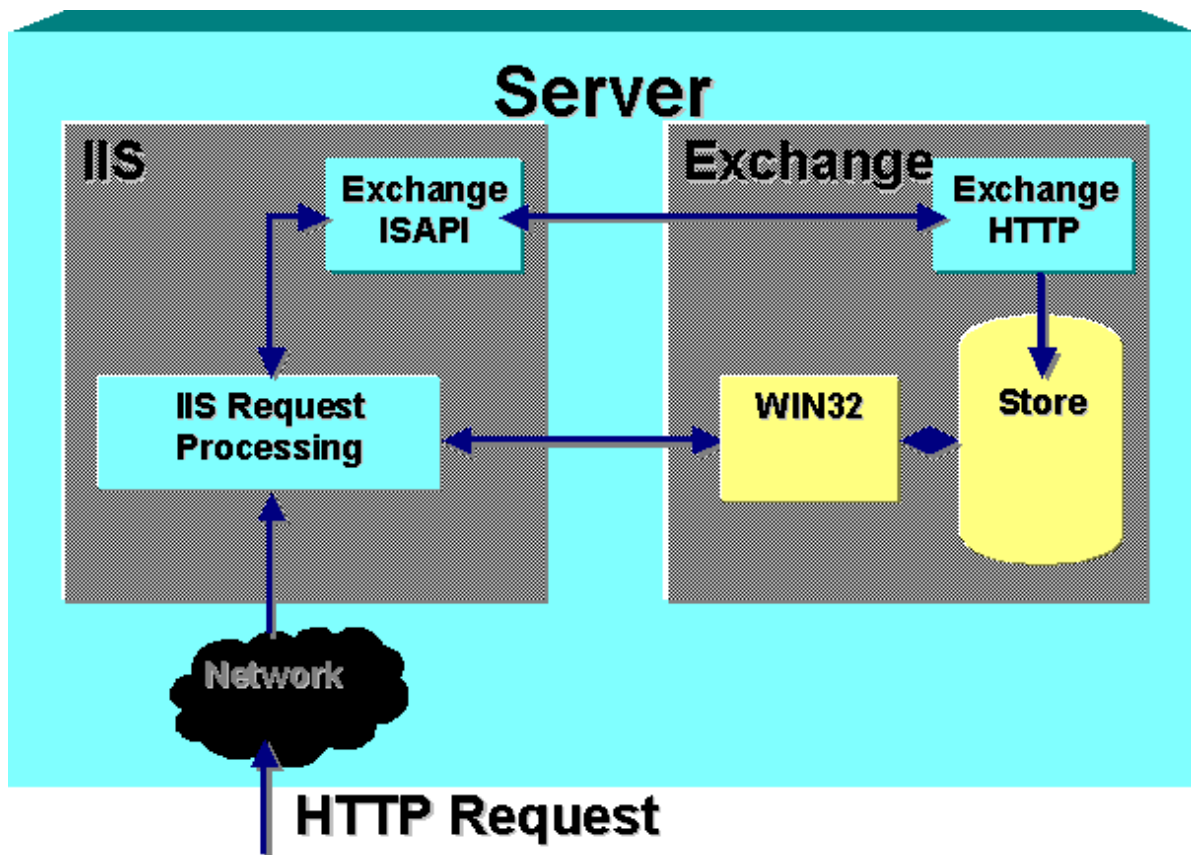


Figure 3 - OWA 2000 Architecture

Figure 3 illustrates how the Exchange 2000 architecture relies on IIS to support access to its information from any client that uses Internet protocols. All rendering of information is performed by the ISAP extension, and the Store side driver passes relevant data back to this extension. A high-speed link called Epoxy is used between the two processes, and the Exchange Installable File System is also used in order to push and pull streaming information from the Web Store.

IIS is responsible for authentication, and IIS logs can be used to monitor client usage of the HTTP protocol.

Web Store

It is easy to dismiss the Web Store as Microsoft marketing, but the Web Store is the visible result of the widespread integration of Internet standards and protocols throughout the Exchange 2000 Server. The Web Store provides access to every component in the Exchange store via a consistent and logical URL namespace.

Reasonable and Ubiquitous

13URLs

Figure 4 illustrates how a fairly simple URL accesses an item in the store. The item is a posting titled **Welcome** in a Public Folder named **Dung's World**, and is served by the Exchange server, **DonSEH**. The URL of the item is **http://DonSEH/Public/Dung's World/Welcome.EML**. The EML extension indicates that the item is an E-Mail element.

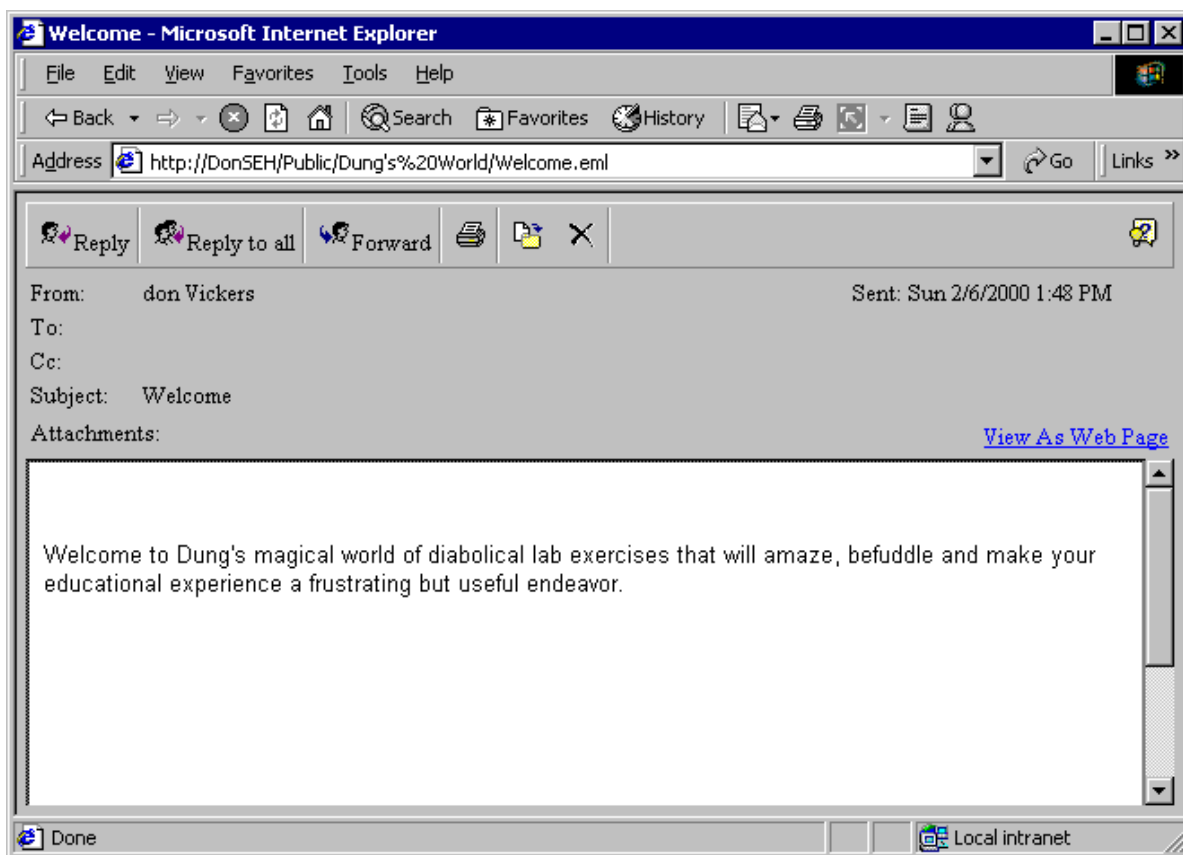


Figure 4 - Sample use of direct access via a URL

Below are several examples of other URL constructs available into the Web Store via OWA 2000:

http://server/exchange/alias

Access the mailbox for the account named **alias** on the server named **server**.

http://server/exchange/alias/calendar

Access the **calendar** folder for the account named **alias** on the server named **server**.

[http://server/exchange/alias/inbox/See Attached.eml](http://server/exchange/alias/inbox/See%20Attached.eml)

Open the message titled, **See attached** in the **inbox** folder of the mailbox for **alias** on **server**.

[http://server/exchange/alias/inbox/See Attached.eml/Budget.xls](http://server/exchange/alias/inbox/See%20Attached.eml/Budget.xls)

Open the attachment named **Budget.xls** in message titled **See attached** in the **inbox** folder of the mailbox for **alias** on **server**.

<http://server/public/policies>

Access the public folder called **policies** on the server named **server**.

Rich Manipulation of Data with HTTP/DAV and Other Protocols

HTTP/DAV, (Document Authoring and Versioning) provides the ability to maintain the various document attributes of documents within HTTP. This protocol allows you to add and modify document attributes and to move, copy and search for items within the Web Store. HTTP/DAV is an IETF accepted development of HTTP 1.1 (RFC 2518). HTTP/DAV is also called WebDAV, and more information it can be found at <http://www.webdav.org/>. HTTP/DAV is a very powerful tool for manipulating documents via the web. The most important features offered by HTTP/DAV are:

- Document locking to make sure that a single client is in use, regardless of the software
- Namespace extension allows clients to manage distributed documents throughout the network
- Rendering by clients to reduce load on servers
- Legacy support of older HTTP methods such as GET, PUT and POST
- Access to document attributes

Stylesheets provide a way of augmenting the presentation and layout associated with an HTML element. DHTML behaviors are reusable components that can be used to extend the properties and methods associated with an HTML element. In other words they can extend the behavior of an element.

Server Considerations

The improved architecture makes OWA 2000 more standard and easier to use than OWA 5.5. OWA and IIS are now standard components of the Exchange 2000 Server.

Installation

As OWA and IIS are standard components of the Exchange 2000 Server, there is virtually no work required to install and implement OWA when installing the Exchange 2000 Server. Unlike previous versions, there is no separate installation for OWA; every Exchange 2000 Server is capable of hosting OWA clients. Should this be an issue, HTTP access to the Exchange server can be disabled, either globally or on a per mailbox basis.

Integration with IIS

The Exchange 2000 Server configures a number of applications (also known as ‘virtual directories’, ‘virtual roots’ and ‘vroots’) into the IIS default web site on the server. Figure 5 shows the view from the IIS Manager:

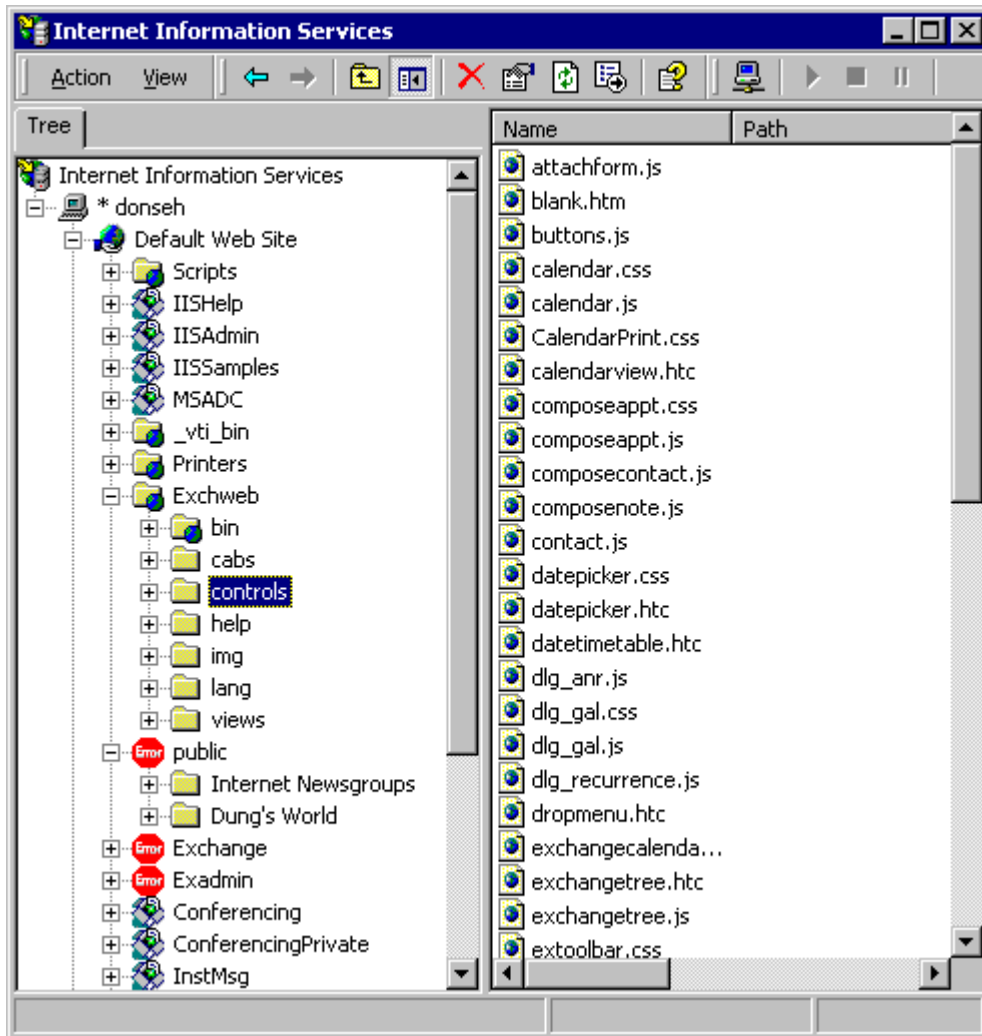


Figure 5 Exchange Applications in IIS

Exchweb

The directories mapped by Exchweb hold various components used by OWA, for example, XSL stylesheets and DHTML behaviors. These objects should not be customized, as any upgrade installation will overwrite these files. See later for a discussion on how you can extend/customize the functionality of OWA.

Public

This is the root for browser access to public folders. The root is mapped to the default Public Folder hierarchy within the organization. In EXIFS terms, this equates to **M:\organization name\Public Folders**.

Exchange

This is the root used to enable browser access to user mailboxes. The mailbox root is also mapped to drive M: by ExIFS. The IIS console can expand this root, but it will only reveal a simple list of mailboxes. Access cannot be gained to mailbox content because the IIS console cannot provide the necessary credentials to authenticate itself to Exchange.

Exadmin

This is used by the Exchange Administration tool to administrate public folders.

Administration

Details of the Exchange ISAPI application are held in the IIS metabase. Details of the application are also accessible through the Exchange System Manager console. However, the Exchange System Manager reads its data from the configuration held in the Active Directory, so clearly a potential gap exists between the two sets of data. The general rule is to always perform management on Exchange components via the Exchange System Manager, including those objects that are also accessed by IIS. The Exchange System Attendant process performs updates to the IIS metabase behind the scenes by replicating changes made to the Active Directory into the IIS metabase¹. No function exists to take changes made to the IIS metabase and replicate them back to the Active Directory. The danger always exists that a change made to the metabase will be overwritten by a subsequent change made by the Exchange System Manager. To avoid this problem, we recommend that you always manage OWA from the Exchange System Manager.

The Exchange System Manager does not provide access to all of the possible settings that can be set on an IIS virtual root. The IIS snap-in is useful to set such properties.

¹ The updates are performed by a service called MExchangeMU, where "MU" means metabase update.

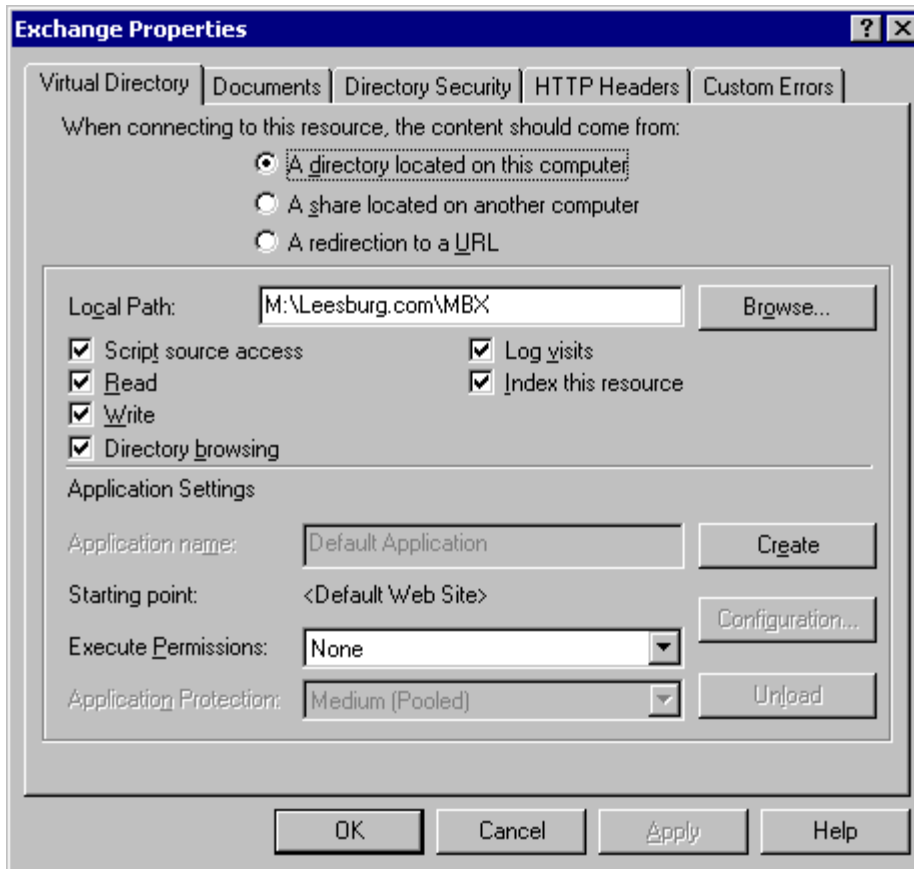


Figure 6 IIS Exchange Properties

Number of Concurrent Clients

As mentioned previously, the many improvements made to OWA 2000 provide very significant improvements in performance. Improved performance was one of the key goals for OWA 2000 as Microsoft hopes to be able to support approximately the same number of OWA clients as MAPI clients on the same server. Microsoft also have the goal of reducing the network load for OWA to approach the load for MAPI.

The actual loading, as always, depends very much on the usage patterns of the people using the system. OWA 2000 holds the folder hierarchy and folder contents locally to reduce the number of roundtrips. This significantly improves response and reduces traffic, assuming the person using the client stays within the information stored locally. If the user jumps from one part of their Exchange information to another in a more random fashion they can, in effect, defeat the feature.

It is too soon to know how well these performance goals will be met, but it is clear that the number of concurrent clients supported by OWA 2000 will be significantly greater than OWA 5.5. Given that the number of clients per server is gated by the time taken to restore the Information Store, it is possible that the concurrent client limit will be effectively lifted by OWA 2000. Exchange 5.5 supports a maximum number of mailboxes per server of no more than 2,000 to 3,000. The improved multiple stores in the Exchange 2000 Server may increase this number by 3 to 4 times to approximately 12,000 clients. It is not unreasonable to expect that a well-designed server will be

able to support this number of OWA clients, possibly through the use of the front-end/back-end architecture discussed below.

The most important factor in determining the number of clients supported, is to determine the level of service the clients themselves experience. This 'experience' can be difficult to express in quantitative measurements such as bytes required per transaction. The experience relates to how comfortable users are with the response and the appearance of the information presented. Low bandwidth has always been an area of irritation with Exchange clients. Some customers have tried to use OWA in place of Outlook in low bandwidth environments with generally unfortunate results due to the heavy server and network loads. OWA 2000 may offer a better answer even if it does not lower the network loading below Outlook. Since Outlook uses MAPI and RPC there is a need for lower network latency. OWA uses HTTP, which accepts much higher latency. It is probable that OWA 2000 will be more favorable in the low bandwidth environments.

Front-End/Back-End Servers

Exchange 2000 introduces the ability to separate servers into front-end and back-end servers in order to provide improved scalability. The front-end servers process the incoming protocol requests and map the requests to the required back-end server via directory examination. This architecture allows us to add additional servers at the required points as traffic and load increase. Front-end scenarios offload the authentication and directory lookups from back-end servers. This architecture currently supports only the Internet protocols since IIS processes these protocols, and not MAPI.

Figure 7 illustrates an environment of three front-end servers for three back-end servers. The front-end servers all share a common DNS address that is served in a round robin fashion. In the scenario illustrated, the following actions occur after the user has authenticated themselves and begun to look for information:

1. A browser sends a request to the virtual server to retrieve information from the public folder with the URL, **\\Public\Discussions\Platinum**.
2. A front-end server processes the request and queries the nearest Active Directory (Global Catalog) to discover on which back-end server the folder is available. The folder may exist in multiple replicas.
3. The Active Directory reads its public folder configuration data and discovers that a replica of the folder is available on back-end **server A**. If multiple replicas exist (as on **server A** and **server B**), the Active Directory directs the connection to the default public folder location for the mailbox, normally the same server as the mailbox.
4. The front-end server passes the request to **server A** and the information is retrieved from the public folder and returned to the browser.

The interaction between a browser and a server when fetching mailbox information, is similar. The following actions might occur:

1. A browser logs on and requests to connect to the mailbox with URL, **\\Mail\Exchange\TonyR**.
2. A front-end server accepts the request and queries the Active Directory to discover which server hosts the mailbox.
3. The Active Directory responds with **server B**.
4. The front-end server passes the request to **server B** and the browser is connected to the mailbox.

Further requests, for example, to the `\Public\Discussions\Exchange 5.5` public folder or other mailboxes are processed in exactly the same manner. Note the close connection between the Active Directory, DNS, and Exchange 2000, which makes this process work.

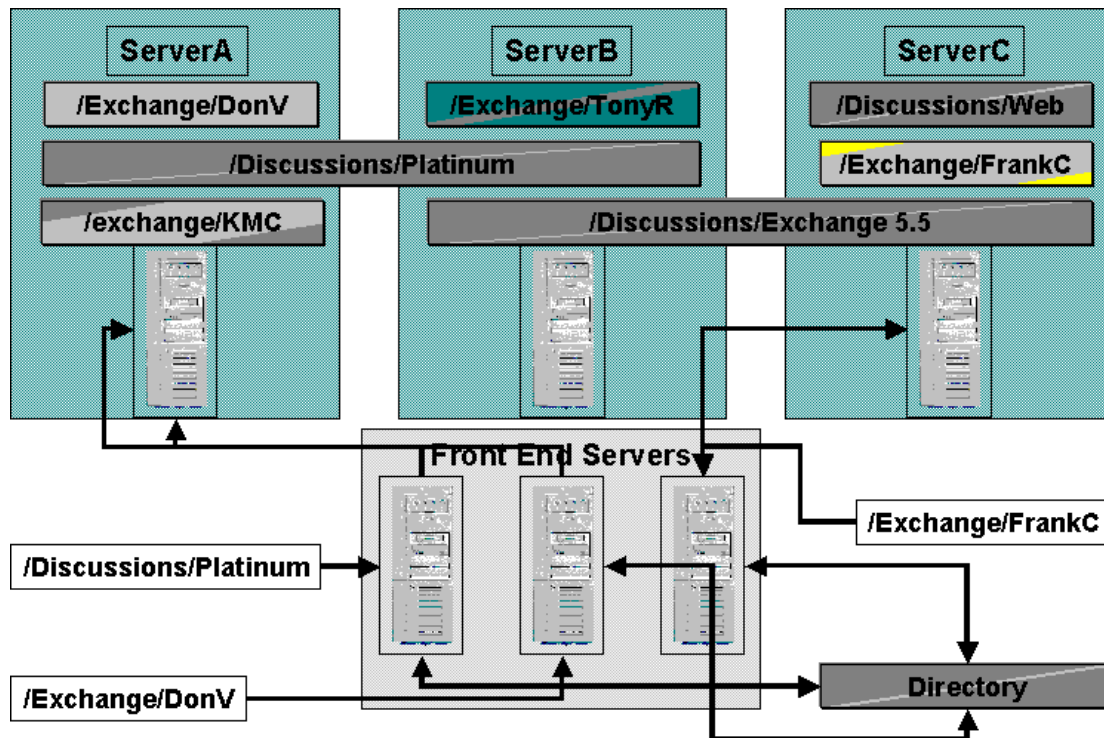


Figure 7 - Accessing information via front-end/back-end servers

Access Security

The two fundamental components of security relating to messaging system security are network security and message security. Network security is aimed at protecting the organization's internal network from being penetrated from the outside, typically from the Internet. Message security is aimed at making sure the information in messages is not compromised, either intentionally or inadvertently by internal or external sources.

OWA can work very well in an environment designed to offer a high degree of network security, especially if the front-end/back-end server architecture is used. The normal tools for providing network security usually involve placing firewalls at key points of the network. OWA is able to operate in an environment that contains firewalls and can operate in conjunction with SSL so that the messages travel within a security shell over the network.

OWA does not offer many benefits in regard to message security. Message security usually involves having messages encrypted and signed using PKI technologies. OWA 2000 does not support these technologies and therefore does not offer extended message level security.

Client Authentication

A key part of providing security is to be certain that clients are authorized to access the system. This is done through authentication. There are a variety of authentication options available for OWA:

- 0 **Basic.** This option is available in virtually every browser and uses simple challenge/response that uses clear text. Since the name and password are sent in clear text this option is not desirable over the Internet unless SSL has been implemented.
- 1 **Integrated Windows.** This option is preferred when the clients are running a Windows client. It provides a relatively high level of security, but requires IE4 or IE5 on the client.
- 2 **Digest.** This option provides an effective and efficient means of providing secure authentication. It uses Message Digest 5 (MD5), which is very fast and efficient.
- 3 **Anonymous.** This option is reasonable to provide access to public folders that are intended for general access.

Security and the Internet

Virtually every messaging system is connected to the Internet today. However, connecting to the Internet requires careful consideration of security.

The placement and design of the firewalls are key design issues for providing network security. The details of this technology is beyond the scope of this document, but a brief overview of the major options is presented. A firewall acts as a boundary between the Internet and a corporate network. The firewall processes packets coming in from the Internet and allows only those deemed appropriate to pass through. The normal approach is to specify which ports are allowed to pass through the firewall. The HTTP port is normally 80 and SSL is normally 443 so these are usually allowed through.

The simplest approach is to place a firewall in front of the Exchange server that allows ports 80 or 443 to pass through. Once the packets pass through the firewall they progress to the Exchange server in the normal fashion for processing. This scheme works whether there is a front-end/back-end design or not. This is the only approach available if no front-end server is used.

If there is a front-end/back-end design in place for the Exchange server there are additional placement options for the firewalls. The firewall can sit between the front-end and back-end as well as being in front of the front-end. Having firewalls on both sides of the front-end servers provides the highest level of protection, but costs are also higher. This approach creates a 'DMZ' in which the front-end server operates. The firewall between the front-end and back-end must allow ports 80 and 3268. Port 80 is for HTTP, while Port 3268 is used to access the Windows 2000 Active Directory. Note that OWA itself uses LDAP for searches, which would be port 389 but the front-end must be able to search more than the local domain, resulting in the use of the 3268 port and not 389. The firewall in front of the front-end operates in the same manner as the firewall in the most simple case described in the paragraph above

OWA as a Toolkit for Extensions

The current OWA generation is implemented using ASPs (Active Server Pages). Changing the behavior of OWA required modification of the ASP pages. While this was moderately easy from a programming perspective, it resulted in a complex ASP page code that was difficult to maintain.

As we have seen, OWA does not use ASPs, relying instead on server DLLs to render items as they are accessed via their URLs. For example when entering a folder URL, the items within the folder are returned in normal view. Entering the URL of a message item or calendar item displays the item in the same way as it would be displayed if accessed through the Inbox or Calendar folders.

Furthermore, parameters can be assigned to the URL to control how items are processed. For example, if the URL is for a folder, a parameter can be provided to indicate the view used for displayed items, or to specify the date required for a calendar type view.

This default processing of items enables a very simple and powerful way of providing vast chunks of OWA functionality inside applications. An example would be to develop a Digital Dashboard and host a discussion in part of a page. Framing is a typical technique used when building Web Pages in which many virtual pages are displayed in a single page. By including a frame that has as its source a URL pointing to a public folder with a parameter specifying a 'Conversation Topic' view, the public folder can be displayed with no coding at all! Not only is the conversation topic view included, but all the associated functions of being able to post, reply and delete items in the discussion forum are provided also. Again – with no code required!

Changing the appearance of a particular folder or the default processing of particular types of items can be achieved easily by associating a 'default document' with the particular folder or content class of the item (each item is associated with a particular content class). When a default document is present, the standard OWA processing described above does not take place. Instead the default document (an HTML page or ASP page) is invoked. Think of default documents as being very similar to Folder Home Pages. They can be set on folders via Outlook in order to display web pages when folders are accessed, rather than the standard item view. Note, however, that default documents and folder home pages are two totally separate features. If you have developed a web page and you want it to appear when a particular folder is accessed regardless of client (ie Browser or Outlook) then you would have to set both a default document and a folder home page to point to your new web page.

Figure 8 illustrates an example of creating an appointment in a user's calendar via a URL. The **calendar** is in the mailbox **VickersDo**. The **?Cmd** specifies a new object and the **&Type** indicates that the object type is an appointment.

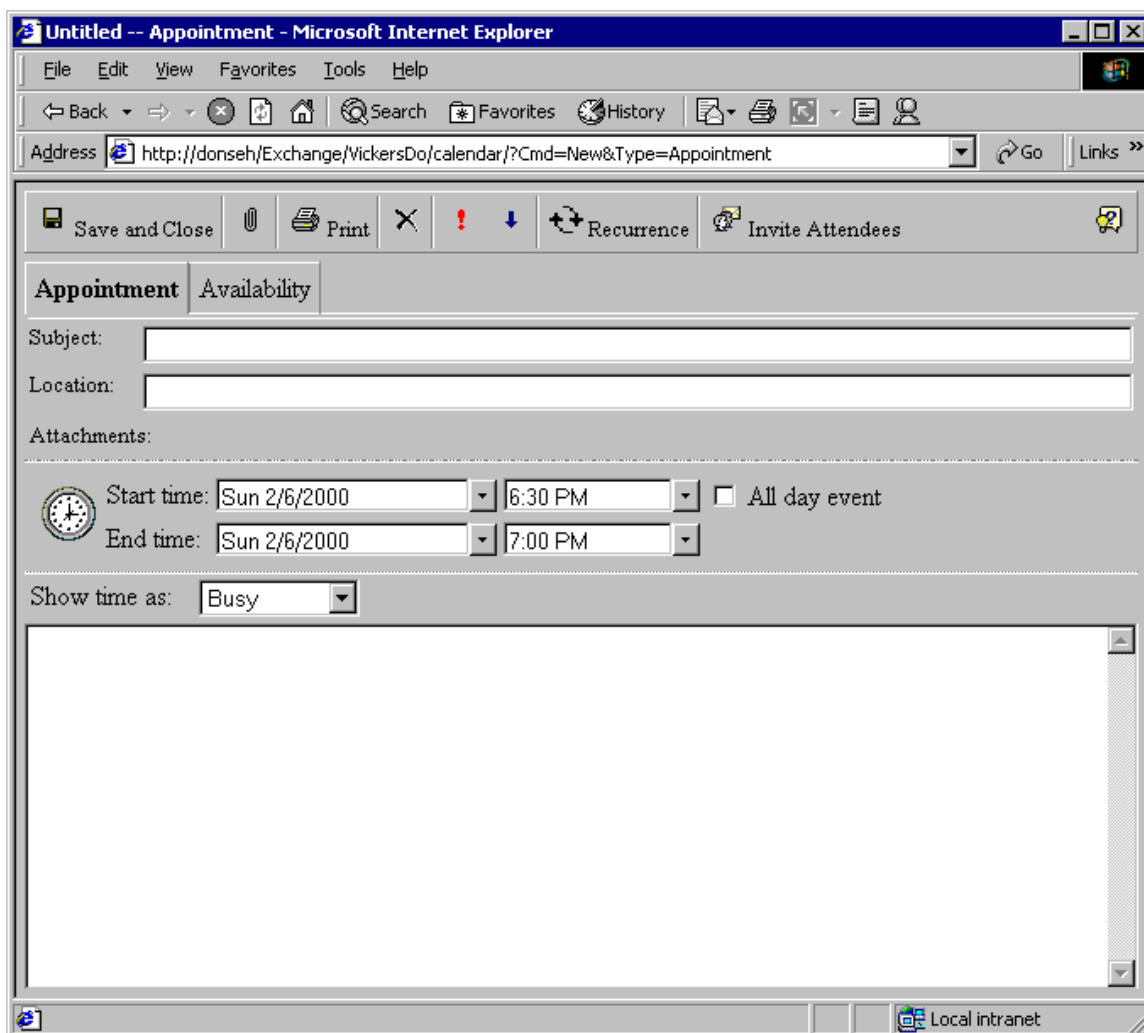


Figure 8 - Creating an appointment via URL commands

Additional values for URL suffixes are shown in Table 2:

<i>URL Suffix</i>	<i>Function</i>
Cmd=Navbar	Display the navigation bar in a frame
Cmd=Contents	Display the contents panel in a frame
Cmd=New	Create a new item (of a specified type)
Cmd=Options	Display the options panel (set out of office notification, etc.)
Cmd=Open	Open an item for reading
Cmd=Edit	Open and edit an item
Cmd=Reply	Reply to the sender of the current message
Cmd=ReplyAll	Reply to everyone in the header of the current message
Cmd=Forward	Forward the current message
Cmd=Delete	Delete the current item
Type=Message	Set the item type to be a message
Type=Post	Set the item type to be a post form
Type=Appointment	Set the item type to be an appointment
Page=x	Display page x (of a set of pages). For example, page 3 of the inbox folder.
View=x	Use a specified Outlook view. For example, View=Daily will

	display the Daily view in the calendar
Sort=x	Sort by the specified column. For example, Sort=Subject
Date=x	Display the specified date in the calendar. Dates are always passed in the format, YYYYMMDD. For example, 20000317 is March 17, 2000.
Table 2: URL suffixes	

OWA as the Stateless Future Client

There is always a huge risk in predicting the future, especially when dealing with people and high technology. It is safe to say that the trend toward smaller and more mobile computing devices will continue and, probably, accelerate. XML and related protocols will almost certainly play a significant role in the interaction of these devices. The work done in OWA 2000 will almost certainly pave the way for even those clients that rely on more standard stateless protocols. The deployment of clients like PDAs, cellular telephones and even smaller and thinner client devices will be assisted significantly by the work done in OWA 2000. OWA is not only the most visible client component in the Exchange 2000 Server, but is probably the component that will change the future the most.

Outlook and MAPI have not received notable attention in Exchange 2000. However, they offer significantly more features and capabilities than OWA 2000, as we have discussed. Only time will tell whether the features in Outlook and MAPI that make it more suitable for enterprises will be provided in future versions of OWA. It is conceivable that OWA could make Outlook obsolete, but that is certainly not the case today.

Appendix A: Related Documents

The following key documents and locations provide a wealth of information regarding successful deployments of Microsoft Exchange Servers on Compaq platforms.

Compaq ActiveAnswers

www.compaq.com/activeanswers

Managing and Monitoring Microsoft Exchange Server

Microsoft Exchange Server Backup and Restore Performance using Compaq 35/70 DLT Arrays

Microsoft Exchange Server Performance and Tuning Guide

Microsoft Exchange Server Deployment and Configuration Guide

Implementing High Availability for Microsoft Exchange Server

Compaq White Paper Index

www.compaq.com/support/techpubs/whitepapers

Compaq TechNote Index

www.compaq.com/support/techpubs

RAID Technology for Database Servers

Microsoft Exchange Server Web site

www.microsoft.com/exchange