®

Microsoft

# Planning and Deploying Outlook Web Access

**Abstract**

This white paper discusses planning and deploying Microsoft® Outlook® Web Access for organizations running Microsoft Exchange Server version 5.0 or version 5.5.

The information that is included assists information technology professionals addressing Outlook Web Access architecture, security, capacity planning, and performance considerations.

For information about troubleshooting, see *Troubleshooting Guide for Outlook Web Access* at http://support.microsoft.com/support/Microsoft Exchange Server/content/whitepapers /whitepapers.asp.

*The information contained in this White Paper represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.*

*This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.*

*Microsoft Corporation • One Microsoft Way • Redmond, WA 98052-6399 • USA*

# CONTENTS

# INTRODUCTION

Microsoft® Outlook® Web Access for Microsoft Exchange Server provides secure access to e-mail, a personal calendar, group scheduling, and collaboration applications on Microsoft Exchange Server using a Web browser. Outlook Web Access offers the following:

- **Allow roaming or remote access**. Users who are away from their PCs—as well as users who share a PC—can use Outlook Web Access to gain secure access to their Microsoft Exchange Server information from any browser.
- **Support UNIX clients and PCs with hardware limitations**. Users running UNIX workstations, or users with PCs lacking the RAM or hard disk space for a traditional Outlook messaging and communication client, can use Outlook Web Access.
- **Provide a lowest common denominator cross-platform system**. Outlook Web Access may meet the needs of organizations that want identical clients on all platforms and require only simple e-mail, scheduling, and collaborative application functionality.
- **Deliver extranet applications**. Corporations that want to deploy mail, discussion, or collaborative applications to vendors, suppliers, or customers can use Outlook Web Access to distribute them easily, inexpensively, and efficiently over the Internet.

## Features

With Outlook Web Access, users can do the following:

- **Use basic e-mail**. Address mail using the Microsoft Exchange Server global address book, send and receive file attachments and hyperlinks, set message priority, request delivery and read receipts, set up hierarchical folders and the Outlook toolbar, and group and sort messages based on standard fields or conversation thread.
- **Access basic calendar and group scheduling features**. Create one-time and recurring appointments in a personal calendar, access day and week views, see free and busy times for multiple users and resources while scheduling a meeting, and automatically send and respond to meeting requests.
- **Access basic public folders**. Access custom views in table format or group and sort messages in a folder based on standard fields or conversation thread.
- **Create collaboration applications**. Develop custom forms for Outlook Web Access using the Microsoft Visual InterDev™ Web development system.

## Limitations

Outlook Web Access is not designed to satisfy advanced e-mail and collaboration requirements addressed by the other products in the Outlook client family. It is not intended to replace the full-featured Outlook messaging client for the 16-bit Windows® operating system or Macintosh. Outlook Web Access does not include the following advanced features in the following categories:

- **Offline use**. A user must connect to Microsoft Exchange Server to view information.
- **E-mail**. Personal address book, spell checking, auto-resolve addresses, Microsoft Exchange Server digital encryption or signature support, S/MIME support. It also does not include replied and forwarded flags in list view, message flags and inbox rules, three-pane view, drag and drop to folder, search for messages, and WordMail and Microsoft Office integration.
- **Calendar and group scheduling**. Monthly view and other customized views of calendar, display discontinuous days side by side, appointment list views, view details with free and busy, drag and drop to move appointments, track acceptance of meeting attendees, all-day or multiple-day events, task lists and task management, and export to DataLink watches or other devices.
- **Public folder access**. Outlook views not in table format; Outlook 97 forms.
- **Collaboration applications.** Outlook 97 forms; Microsoft Exchange Server digital encryption and signatures; synchronize local offline folders with server folders.

## Comparison of Active Server Components and Outlook Web Access

There are two different versions of the Web-based messaging client for Microsoft Exchange Server:

- Outlook Web Access is a component of the Web-based messaging client in Microsoft Exchange Server version 5.5.
- Active Server Components (ASC) is a component of the Web-based messaging client in Microsoft Exchange Server version 5.0.

The following table lists the features that are supported by each Web-based messaging client that is available with Microsoft Exchange Server.

| E-mail Features | Outlook Web Access 5.5 | Outlook Web Access 5.5 SP1 | ASC 5.0 | ASC 5.0 SP1 |
|---|---|---|---|---|
| **Message creation** | | | | |
| Receive rich-text messages | X | X | | |
| Send attachments | X | X | | X |
| Receive attachments | X | X | | X |
| Support hyperlinks | X | X | X | X |
| Set message importance | X | X | X | X |
| Request read or delivery receipt | X | X | X | X |
| **Message addressing** | | | | |
| Use global address list | X | X | X | X |
| Check names | | X | | |
| **Message folder viewing** | | | | |
| Sort by standard fields | X | X | | |
| View folder hierarchy in a pane | X | X | | |
| Use Outlook navigation bar | X | X | | |

| E-mail Features | Outlook Web Access 5.5 | Outlook Web Access 5.5 SP1 | ASC 5.0 | ASC 5.0 SP1 |
|---|---|---|---|---|
| **Message management** | | | | |
| Use Out of Office Assistant | X | X | | |
| Create folders on the server | X | X | X | X |
| **Calendaring** | | | | |
| Create single appointments | X | X | | |
| Create recurring appointments | X | X | | |
| View calendar by day and week | X | X | | |
| View free and busy information | X | X | | |
| **Collaboration Applications/Other** | | | | |
| Use custom HTML forms | | X | | |
| Change user account password | | X | | |
| Use personal contacts | | X | | |

## Server Requirements

The following server hardware and software is required.

- Pentium 6/200 single or dual processor
- 256 MB RAM, minimum
- High-speed network connection to Microsoft Exchange Server
- Microsoft Windows NT Server 4.0 operating system with Service Pack 4 (SP4)
- Microsoft Internet Information Server (IIS)
  Microsoft Exchange Server 5.0 supports IIS 3.0 only. Microsoft Exchange Server 5.5 supports IIS 3.0 and later.
- Active Server Pages (ASP)
  Install ASP from the Windows NT Server operating system 4.0 SP3 compact disc, or download Windows NT 4.0 SP3 and the ASP components from http://www.microsoft.com
- Active Server Components (included with Microsoft Exchange Server 5.0) or Outlook Web Access components (included with Microsoft Exchange Server 5.5)
  Microsoft recommends that you install Microsoft Exchange Server 5.0 SP1 or Microsoft Exchange Server 5.5 SP2 because they include the enhanced Outlook Web Access components.

## Client Requirements

The following client software is required.

Each client requires a compatible browser to connect to the Active Server Pages on the Outlook Web Access server.

- Internet Explorer 3.02, 4.0 or later
- Third-party Internet browser software that supports frames (for example, Netscape Navigator)

# ARCHITECTURE

Outlook Web Access is a MAPI application that consists of binary, HTML, and Active Server Page script files. The scripts use CDO (Collaboration Data Objects) to access mailbox and public folder information stored on a Microsoft Exchange Server computer. Outlook Web Access also uses Microsoft Active Server Pages technology on the Web server, and JavaScript and a Java control downloaded on demand to the user's Web browser, to generate HTML pages.

Although the Web browser performs some processing (using the downloaded JavaScript) on the client computer, the Outlook Web Access server handles most of the processing normally done by the client. This server processing includes MAPI sessions, client logic, state information, address resolution, rendering, content conversion, and RPC communications with the Microsoft Exchange Server. The Microsoft Exchange Server receives and processes requests from Outlook Web Access that resemble requests from any MAPI client.

## How Outlook Web Access Works

The following process describes what happens when a user reads a message in a Microsoft Exchange Server mailbox using a Web browser with Outlook Web Access running on a back-end Web server.

1. A Web browser running the Outlook Web Access client sends a request to a computer running IIS and the Outlook Web Access server. This request includes a cookie that identifies the Web browser and user.
2. IIS hands the request to ASP for processing. ASP verifies that the cookie points to a valid ASP session, and the user is logged on.
3. The Internet Services API (ISAPI) filter determines which language to use when displaying messages in the browser.
4. ASP opens the script file named in the URL and executes any server-side Microsoft VisualBasic® script it contains.
5. These scripts use CDO to open the message in the user’s Microsoft Exchange Server information store. The message GUID is passed on within the query string of the URL.
6. The CDO rendering library (Cdohtml.dll) converts the requested message into HTML and IIS sends the HTML to the browser.
7. The Web browser renders the HTML including the embedded JScript.

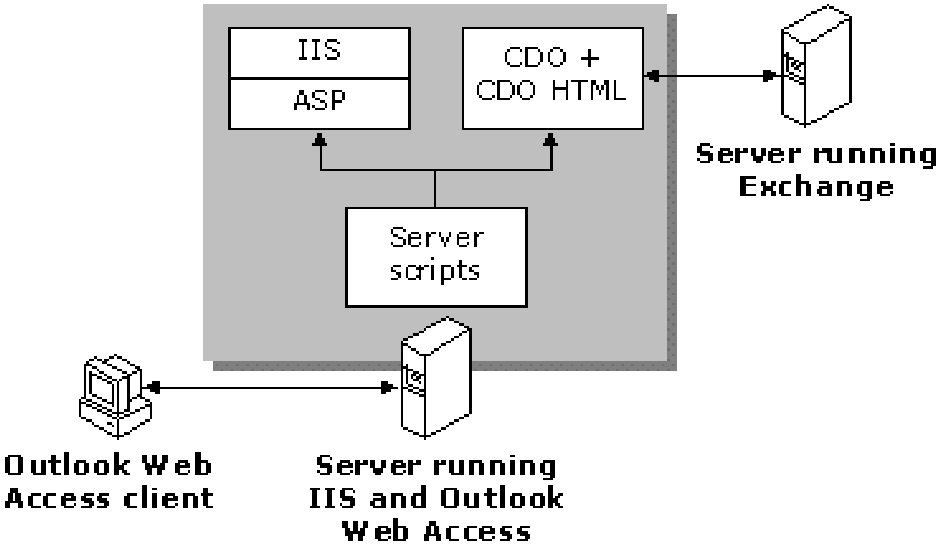
IIS
ASP
CDO +
CDO HTML
Server scripts
Server running Exchange
Outlook Web Access client
Server running IIS and Outlook Web Access

*Figure 1:* **The interaction between the Outlook Web Access client, the IIS/Outlook Web Access server, and the Microsoft Exchange Server computer**

## Topology Recommendations

Consider the following when planning your Outlook Web Access topology.

- Dedicate one or more servers that are not Microsoft Exchange Server computers to IIS and Outlook Web Access components.
- Use load balancing software or hardware to best serve users and improve server availability.

### Use Dedicated IIS/Outlook Web Access Servers

The Outlook Web Access server performs most of the work for connected clients and handles all the load for active client sessions. Supporting one client connection is analogous to running an instance of Outlook on the Outlook Web Access server. Therefore, the Outlook Web Access server must run many active MAPI sessions to the Microsoft Exchange Server. Although the browser has a small footprint on the client computer, the sessions that clients establish with an Outlook Web Access server consume significant resources on the Outlook Web Access server. As a result, the Outlook Web Access server loads must be planned accordingly.

To ensure that Outlook Web Access remains scalable as your organization grows and changes, put IIS and Outlook Web Access components together on a dedicated server that is separate from other Microsoft Exchange Servers in your organization. As the number of clients increases, you can add more Outlook Web Access servers to support the load without affecting Microsoft Exchange Server mailboxes on other servers. Although adding more memory to servers improves performance, measurable performance improvements diminish after a certain point.

**Note** If Outlook Web Access and Microsoft Exchange Server are not installed on the same computer, Windows NT Challenge/Response (NTLM) authentication is not supported.

### Use Load Balancing Software or Hardware

Load balancing software and hardware allows multiple servers to handle requests addressed to a single IP address. Load balancing has several advantages:

- Users need only one URL to access their mail. The load balancing software or hardware determines which server handles user requests.
- Users are directed to another server if an Outlook Web Access server goes down.
- Load balancing software or hardware efficiently distributes users across multiple servers, so users do not overload a single server.

For software load balancing, Microsoft recommends Windows NT Server 4.0 Enterprise Edition Load Balancing Service (WLBS), which supports up to 32 servers. For hardware load balancing, Microsoft recommends Cisco LocalDirector, which supports up to 64,000 servers.

**Note** Because round robin DNS works only with stateless ASP applications, it will not work with Outlook Web Access. With round robin DNS, every user request is sent to a different server, interrupting the user’s ASP session. As a result, users are forced to log in again every time round robin DNS directs them to a different server during their session.

## NETWORK SECURITY

If Outlook Web Access clients access Microsoft Exchange Server over an Internet connection, Microsoft recommends protecting the Microsoft Exchange Server. Firewalls prevent unwanted access to data on servers behind the firewall. Firewalls implement many different layers of security to protect servers and data from attack. For more information about firewalls, see *Fight Fire with Firewalls* at http://www.microsoft.com/workshop/server/proxy/server072798.asp.

There are two ways to implement a firewall with the Outlook Web Access architecture. Review both options and choose the best one for your organization.

- Firewall between IIS/Outlook Web Access and Microsoft Exchange Server
- Firewall between the client and IIS/Outlook Web Access server

### Firewall Between IIS/Outlook Web Access and Microsoft Exchange Server

A firewall between the IIS/Outlook Web Access server and the Microsoft Exchange Server may seem practical if several web-enabled applications are already deployed on the IIS server and it is important to protect the Microsoft Exchange Server and domain controllers. In this configuration, all the client connections from IIS to the Microsoft Exchange Server are filtered by the firewall. Thus, users can access the IIS/Outlook Web Access server without going through the firewall, but they must go through the firewall to access data on the Microsoft Exchange Server. This configuration is not recommended because the IIS/Outlook Web Access server is not protected.

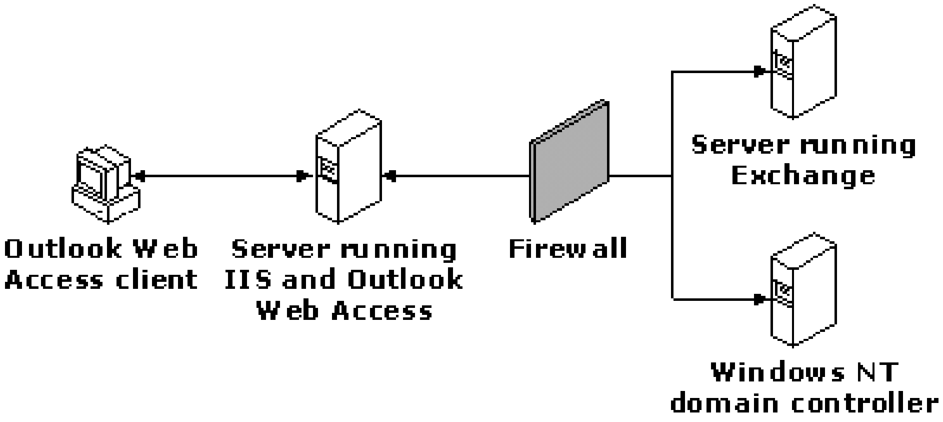
Outlook Web Access client
Server running IIS and Outlook Web Access
Firewall
Server running Exchange
Windows NT domain controller

*Figure 2:* **Firewall between the IIS/Outlook Web Access server and other servers in an organization**

Using this design, IIS (acting as a MAPI client to the Microsoft Exchange Server) requires similar access to the Microsoft Exchange Server and domain controller as a standard Outlook client.

On the firewall, it is important to enable several TCP ports to allow the Outlook Web Access server to connect successfully to the Microsoft Exchange Server directory and information store. Because Microsoft Exchange Server randomly assigns ports for the directory and information store by default, you must statically map the ports that clients will use.

The following steps describe the registry entries that must be added to allow static port mapping. In this example, port 1225 is mapped to the directory and port 1226 is mapped to the information store.

**Important** Do not assign ports immediately above the 1023 range to the directory and information store. This may cause other problems with Microsoft Exchange Server.

1. Using the Registry Editor on the Microsoft Exchange Server computer, add the following entry for the Microsoft Exchange Server directory in **HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services \MSMicrosoft Exchange ServerDS\Parameters**:
   Entry: **TCP/IP port REG_DWORD**
   Value: *port number to assign*
   For example, in the port number ”dword:000004C9(1225)" the decimal number 1225 (4C9 in hexadecimal format) is for the directory.
2. Add the following entry for the information store in **HKEY_LOCAL_MACHINES\System\CurrentControlSet\Services \MSMicrosoft Exchange ServerIS\ParametersSystem**:
   Entry: **TCP/IP port REG_DWORD**
   Value: *port number to assign*
   For example, in the port number "dword:000004CA(1226)" the decimal number 1226 (4CA in hexadecimal format) is for the information store.
3. Quit the Registry Editor.

After enabling these ports on the firewall, test your configuration and verify that a browser can connect successfully with the Microsoft Exchange Server over the Internet. For test purposes, use Outlook (using TCP transport) to connect to the Microsoft Exchange Server. If you are able to connect with Outlook, the Outlook Web Access client will also connect. If Outlook does not connect to Microsoft Exchange Server, check the firewall settings and verify the Microsoft Exchange Server configuration.

### Firewall Between the Client and IIS/Outlook Web Access

The most common and secure firewall configuration requires that users go through the firewall to access IIS/Outlook Web Access and Microsoft Exchange Server computers. This configuration secures all information that flows into the IIS/Outlook Web Access server (all incoming TCP packets must pass through the firewall or packet filter). This configuration is also useful for preventing Internet access to other applications and services running on your IIS/Outlook Web Access server. The firewall is located outside of the organization to prevent any attacks or unwanted access to the servers. For this design, configure the firewall to pass HTTP on port 80.

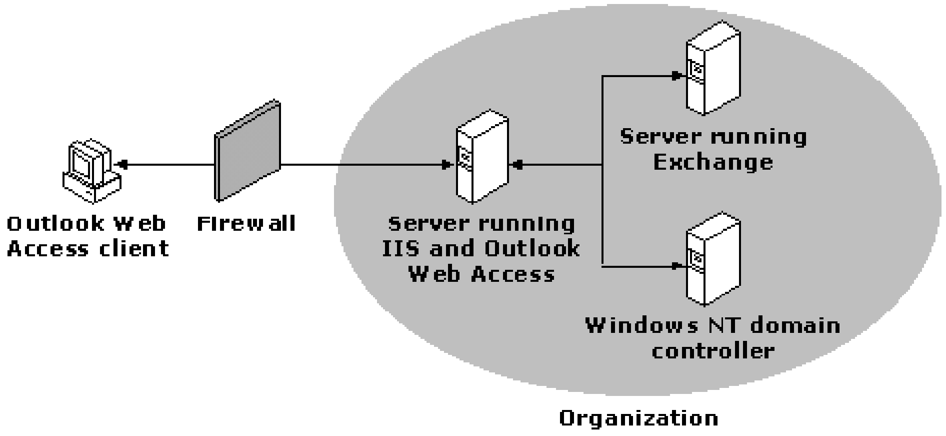
Outlook Web Access client
Firewall
Server running IIS and Outlook Web Access
Server running Exchange
Windows NT domain controller
Organization

***Figure 3:*** **Firewall between the Outlook Web Access client and the organization**

In this scenario, the browser connects over TCP port 80 through the firewall to IIS. IIS then communicates with Microsoft Exchange Server to access Microsoft Exchange Server data. The IIS/Outlook Web Access server then renders HTML for the client by sending the data through the firewall and over the Internet on TCP port 80.

## OUTLOOK WEB ACCESS SECURITY

You can configure Outlook Web Access using different user authentication methods.

### User Authentication

The Outlook Web Access server can authenticate users with one or more of the following types of security.

- Anonymous
- Basic (Clear Text)
- Basic (Clear Text) over Secure Sockets Layer (SSL)
- Windows NT Challenge/Response (NTLM)

### Anonymous

If Outlook Web Access is configured for Anonymous authentication, users can use Outlook Web Access without specifying a Windows NT user account name and password. Each time a user establishes an anonymous connection, IIS logs the user on with an anonymous or guest account, which is a valid Windows NT user account. By default, the anonymous account is IUSR_ComputerName. Anonymous authentication provides access only to resources that are published anonymously, such as public folders and directory content.

The following are advantages and disadvantages of Anonymous authentication.

**Advantages**

- All browsers support Anonymous authentication.
- Users are not prompted for credentials.

**Disadvantages**

- Anonymous authentication is not secure.
- Users can only access the Global Address List and public folders that are configured for anonymous access.

### Basic (Clear Text)

If Outlook Web Access is configured for Basic authentication, users must specify a valid Windows NT user account name and password in order to use Outlook Web Access. Both the user name and password are transmitted as clear text over the network to the IIS/Outlook Web Access server. The advantage of Basic authentication is that users can access an unlimited number of resources, even if those resources are not on the user’s Outlook Web Access server. For example, a user can access public folders on one Microsoft Exchange Server and e-mail on another Microsoft Exchange Server.

**Caution** Because Basic authentication transmits passwords across the network as unencrypted information, Microsoft recommends that you use SSL with Basic authentication, which encrypts all information passing through IIS.

The following are advantages and disadvantages of Basic authentication.

**Advantages**

- All browsers support Basic authentication.
- Users can access all Microsoft Exchange Server resources.

**Disadvantages**

- Basic authentication is not secure.
- Users are prompted for a user name and password.
- Users must be granted the Log on Locally right on IIS.

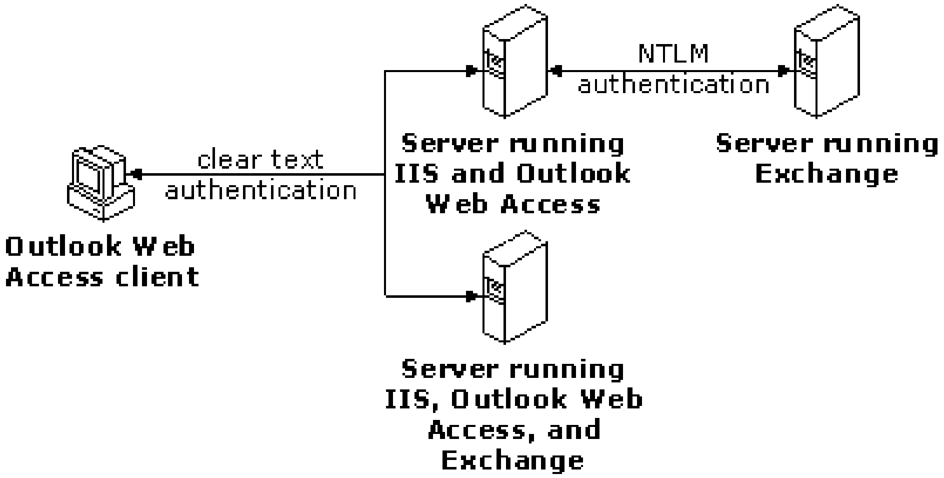
NTLM
authentication
clear text
authentication
Server running IIS and Outlook Web Access
Server running Exchange
Outlook Web Access client
Server running IIS, Outlook Web Access, and Exchange

*Figure 4:* **Accessing resources using Basic authentication**

## Basic (Clear Text) Over Secure Sockets Layer (SSL)

If Outlook Web Access is configured for Basic authentication over SSL, users must specify a valid Windows NT user account name and password to use Outlook Web Access. Both the user name and password are transmitted as encrypted information over the network to the IIS/Outlook Web Access server. As with Basic authentication, users can access an unlimited number of resources with Basic over SSL authentication, even if those resources are not on the user’s Outlook Web Access server.

The following are advantages and disadvantages of Basic over SSL authentication:

**Advantages**

- Most browsers support Basic over SSL authentication.
- Users can access all Microsoft Exchange Server resources.
- Basic over SSL authentication is very secure.

**Disadvantages**

- Performance can be slow as a result of the encryption.
- Users are prompted for a user name and password.

- Users must be granted the Log on Locally right on IIS.

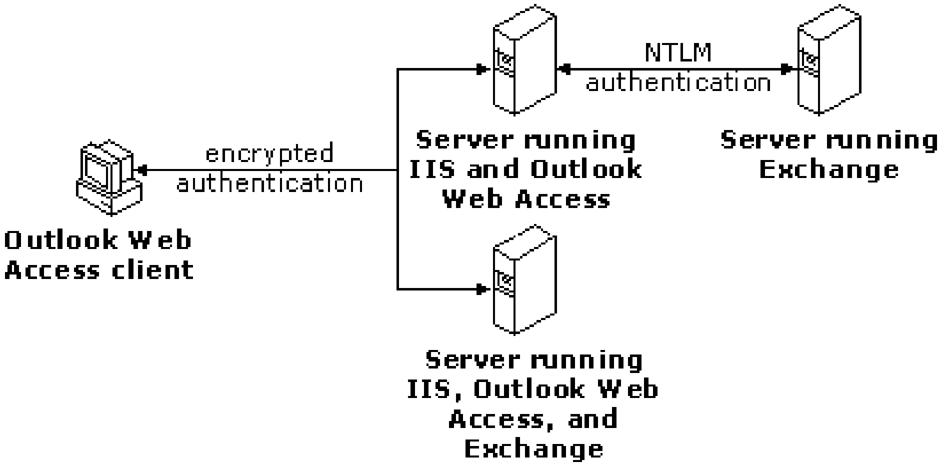
NTLM authentication
encrypted authentication
Server running IIS and Outlook Web Access
Server running Exchange
Outlook Web Access client
Server running IIS, Outlook Web Access, and Exchange

*Figure 5:* **Accessing resources using Basic over SSL authentication**

## Windows NT Challenge/Response (NTLM)

If Outlook Web Access is configured for Windows NT Challenge/Response, users must specify a valid Windows NT user account name and password in order to use Outlook Web Access. The user name and password are sent from the browser to the IIS server as encrypted information. A serious limitation of NTLM is that all resources the user wants to use must reside on the same server as IIS and Outlook Web Access. NTLM authentication is not supported if IIS/Outlook Web Access and Microsoft Exchange Server are located on different computers.

The following are advantages and disadvantages of NTLM authentication:

**Advantages**

- NTLM authentication is relatively secure.
- Users are not prompted for a user name or password.

**Disadvantages**

- Users can access only resources on the IIS/Outlook Web Access server.
- All browsers (for example, Netscape Navigator) do not support NTLM authentication.

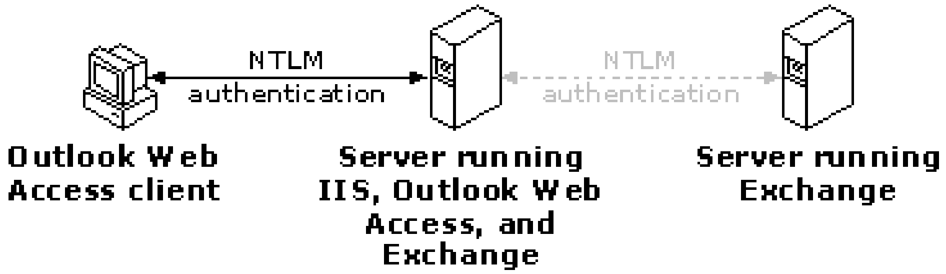
NTLM authentication
NTLM authentication
Outlook Web Access client
Server running IIS, Outlook Web Access, and Exchange
Server running Exchange

*Figure 6:* **Accessing resources using NTLM authentication**

### Roaming Users

If several users share a computer to access e-mail using Outlook Web Access, Microsoft recommends disabling local caching on the browser. If caching is not disabled, messages accessed during the previous Outlook Web Access session may still remain on the local disk, making it possible for someone to see another user’s messages.

For increased security, Microsoft recommends not using the Save Password feature in Internet Explorer. For other information about disabling local caching and the Save Password feature, see the user documentation for the Web browser.

### Additional Information

For additional information about user authentication, see the following technical notes at http://technet.microsoft.com/cdonline/default.asp:

- MS Internet Information Server Security Overview
- The Basics of Security
- Implementing a Secure Site with ASP

# CAPACITY PLANNING

This section describes how to plan the number of users per server and the critical bottlenecks that affect server performance.

## Planning Users Per Server

When planning how many servers are necessary to support the number of projected Outlook Web Access clients, consider the following:

- **Users Per Outlook Web Access Server.** This is the number of connected Outlook Web Access clients the IIS/Outlook Web Access server can comfortably support.
- **Users Per Microsoft Exchange Server.** This is the number of Outlook Web Access clients that the Microsoft Exchange Server can comfortably support.
- **Testing Your System.** Allow resources and time in your schedule to test your plans to make sure they work for your organization.

### Users Per Outlook Web Access Server

In a Microsoft Exchange Server environment, the load placed on IIS by Outlook Web Access clients is determined by the number of ASP requests per second that are processed on behalf of each user. Clients perform a number of tasks, such as reading, deleting, and sending mail messages, along with scheduling activities—all of which require the server to process ASP requests. The challenge of planning the number of users per server is to characterize user workload and then monitor test users on a Microsoft Exchange Server.

The number of users per Outlook Web Access server is different for every organization because each user population has a different profile, or usage characteristics. It is best to perform a test deployment and monitor test users with Performance Monitor, then add additional Outlook Web Access servers when the number of ASP requests/sec reaches its limit on the server.

When planning for a deployment, gather data about how often users will access the Microsoft Exchange Server and what actions they will perform. For example, determine the number of connections per day, the number of messages sent, read and deleted, and the number of calendar actions per day. Users in some organizations simply need a means for checking messages, whereas others are very active users who send large messages, use their calendars heavily, and read and delete a large number of messages. Other companies may need more Outlook Web Access servers for the same number of Microsoft Exchange Server mailboxes than an organization that uses the client to simply check e-mail occasionally over the Web.

After deciding what features users use, take a sampling of some average users and ask them to start using the client from their workstations while you monitor the performance of the server with Performance Monitor. Use the recorded log of ASP requests users are generating to estimate user usage profiles. Before deploying Outlook Web Access throughout an organization, use Performance Monitor to measure the overall number of ASP's processed per second and fine-tune the installation. If the Performance Monitor counters are consistently too high and users frequently get the “server too busy” error, consider adding additional Outlook Web Access servers.

### Users Per Microsoft Exchange Server

After determining how many users will be on each Outlook Web Access server, determine how many Microsoft Exchange Server servers are required to handle the traffic from all users. A single Outlook Web Access user places more load on a Microsoft Exchange Server than a single MAPI user (using Microsoft Exchange client or Outlook 97). As a result, a Microsoft Exchange Server can support fewer Outlook Web Access users than MAPI users.

### Testing Your System

After determining a user profile for the organization, test the environment to make sure it can support the total number of users. In a test system, start small and deploy Outlook Web Access to only 50 to 200 users. Monitor performance and add servers as needed to scale the deployment.

## Considerations

The following are the most critical bottlenecks that impact Outlook Web Access server performance. Consider these issues when planning a deployment:

- ASP requests per second
- Number of MAPI sessions
- Number of ASP sessions

### ASP Requests Per Second

Monitor the active ASP requests by using the ASP request/sec counter in Windows NT Performance Monitor. As a rule of thumb, keep the number of ASP requests per second below 15. When this counter exceeds 15 ASP requests per second, the server will respond more slowly to user requests, it will start to queue incoming user requests, and CPU usage will reach 100 percent. When Outlook Web Access queues requests, overall server performance degrades, which is another reason to deploy Outlook Web Access on dedicated servers separate from Microsoft Exchange Server.

### Number of MAPI Sessions

A MAPI session describes the set of resources on the Outlook Web Access server that enables the server to communicate with a Microsoft Exchange Server. Because the Outlook Web Access server must maintain state information for each MAPI session, the number of MAPI sessions affects server load and overall performance. As a result, server scaling is limited by the number of MAPI sessions.

### Number of ASP Sessions

An ASP session stores the client state information for each user that is logged on to the Outlook Web Access server. ASP sessions are expensive to create, update, and destroy. They expire after they exceed an idle time-out period (the default is one hour), or when a user explicitly ends the session by logging off.

If ASP sessions are not properly shut down when the client is finished connecting to the server, ASP sessions accumulate and degrade server performance. Abandoned sessions will continue to consume server resources until the sessions time out. To properly shut down ASP sessions, instruct users to click **Log Off** in Outlook Web Access, instead of just closing their browser to end the session.

Even if users log off from their Outlook Web Access sessions, the server can still perform poorly. Because ASP memory cleanup happens as a background process, a busy server will consume memory throughout the day. This is another reason to use dedicated servers.

## PERFORMANCE CONSIDERATIONS

Use the following tips to fine-tune the system's performance.

- Monitor server performance regularly.
- Fine-tune ASP session timeout, as necessary.
- Optimize IIS and ASP settings to improve overall performance.

### Monitoring Performance

To ensure that the system maintains satisfactory performance, monitor the following Performance Monitor counters.

| Counter | Component | Description |
|---|---|---|
| Requests Per Second | Active Server Pages | The number of ASP requests per second, which is important in determining server load capabilities. This counter should be between 10 and 15. |
| Requests Executing | Active Server Pages | The number of ASP requests currently executing. Restart IIS if the server is idle and requests are executing. |
| Requests Queued | Active Server Pages | The number of ASP requests that are backlogged. This queue will vary between 1 and 20. If it grows unbounded, restart IIS and consider rebalancing the server's load. |
| Requests Total | Active Server Pages | The total number of ASP requests since the IIS service started. |
| Active Sessions | Active Server Pages | The number of ASP sessions that are open on the IIS/Outlook Web Access server. |

| Counter | Component | Description |
|---|---|---|
| Sessions Timed Out | Active Server Pages | The number of ASP sessions on the IIS/Outlook Web Access server that have timed out since the IIS service started. |
| Messages Rendered | MSExchangeWEB | The number of messages opened by clients. This counter helps to classify the user profile. |
| Attachments Rendered | MSExchangeWEB | The number of attachments opened by clients. This counter helps to classify the user profile. |

### Fine-Tune ASP Session Time-out

Reducing the session time-out can reduce the average number of active sessions, and therefore decrease the resources required by Active Server Pages. The drawback is that user sessions time out sooner and users are forced to log on again after a shorter idle time.

Outlook Web Access overrides global ASP session time-out for its sessions.

Complete the following the steps to modify the Outlook Web Access session time-out. Note that this only affects the ASP sessions for Outlook Web Access. Other Active Server applications are not affected.

1. Back up the Exchsrvr/Webdata/Lang/Lib/Logon.inc file.
2. Open Logon.inc in Notepad.
3. On the **Search** menu, click **Find**.
4. Type the string **itimeout**, and then click **Find Next**.
   **Note** The variable name **itimeout** appears eight times in Logon.inc. Modify the first four references, which relate to authenticated access**.** The next four references relate to anonymous access.
5. Locate the following code fragment and modify only the number of minutes (as shown in bold below):

```
iTimeout =
    objRenderApp.ConfigParameter("AuthenticatedSessionTimeout")
If iTimeout = 0 Then
    iTimeout = 60 minutes
End If
Session.Timeout = iTimeout
```

6. Save Logon.inc. Any new sessions will reflect the changes made. Restarting the web service is not necessary.

## Optimize IIS and ASP Settings

In many cases, you can optimize the system by editing registry entries using the Registry Editor.

**Caution** Be careful using the Registry Editor because using it incorrectly can corrupt your system's configuration and can require reinstalling Windows or Active Server Pages.

Edit the following entries in **HKEY_LOCAL_MACHINE\SYSTEM \CurrentControlSet\Services\W3SVC\ASP\Parameters**:

**ProcessorThreadMax**
Entry: **REG_DWORD**
Range: 1 - 0x000000C8
Default: 10

Specifies the maximum number of worker threads to create per processor. Do not create more than 20 threads per processor. If this value is changed, the Web server must be stopped and restarted for the change to take effect.

**RequestQueueMax**
Entry: **REG_DWORD**
Range: 1 - 0xFFFFFFFF
Default: 500

Specifies the maximum number of .Asp file requests to maintain in the request queue available for each thread. When the limit is reached, clients are sent the value from the registry value ServerTooBusy. If this value is changed, the Web server must be stopped and restarted for the change to take effect. To prevent server thrashing, set this value to a lower number (such as 15 to 20). This prevents the server from queuing up too many ASP requests, which will overload the server. After decreasing this value, users can receive a "server too busy" error when using the Outlook Web Access server if there is heavy load on the server. This reduces the queue size and prevents requests from waiting in the queue for a long time.

Edit the following entries in **KEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet \Services\InetInfo\Parameters**:

**UserTokenTTL**
Entry: **REG_DWORD**
Range: 0 -0 x7FFFFFFF
Default: 600 (10 Minutes)

When a request is made to the server, the security credentials for the request (or the configured anonymous user) are used to create a user token on the server that the server uses when accessing files or other system resources. The token is cached so the Windows NT log on only occurs the first time that the user accesses the system or after the user's token is purged from the cache. NTLM authentication tokens are not cached.

Do not set this value (in seconds) too low. Monitor how users use the client, but do not force them to log on to again because logging on creates many ASP requests. On the other hand, do not set this number too high because the token will be cached much longer than necessary.

**MaxPoolThreads**
Entry: **REG_DWORD**
Range: 0 - 0xFFFFFFFF
Default: 10

Specifies the number of pool threads to create per processor. Each pool thread watches for and processes the network request. Do not exceed 20 threads per processor.

**MaxConnections**
Type: **REG_DWORD**
Range: 0 - 0xFFFFFFFF
Default: 1000

Specifies the maximum number of simultaneous connections that the server allows at any given time. When the number of current connections exceeds this value, the server rejects the request.

## Additional Information

For more information on tuning IIS/ASP performance, see the following white papers:

- *Server Performance Optimization on Microsoft's Web Site*
  http://www.microsoft.com/workshop/server/feature/serveroptms.asp
- *Tuning Internet Information Server Performance*
  http://www.microsoft.com/ISN/whitepapers/tuningiis.asp.