

Infrastructure Planning
and Design

Internet Information Services 7.0

Internet Information Services 7.5

Version 2.2

Published: June 2008

Updated: November 2011

For the latest information, please see [www.microsoft.com/ipd](http://www.microsoft.com/ipd)

Copyright © 2011 Microsoft Corporation. All rights reserved. Complying with the applicable copyright laws is your responsibility.  By using or providing feedback on this documentation, you agree to the license agreement below.

If you are using this documentation solely for non-commercial purposes internally within YOUR company or organization, then this documentation is licensed to you under the Creative Commons Attribution-NonCommercial License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/2.5/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.

This documentation is provided to you for informational purposes only, and is provided to you entirely "AS IS".  Your use of the documentation cannot be understood as substituting for customized service and information that might be developed by Microsoft Corporation for a particular user based upon that user’s particular environment. To the extent permitted by law, MICROSOFT MAKES NO WARRANTY OF ANY KIND, DISCLAIMS ALL EXPRESS, IMPLIED AND STATUTORY WARRANTIES, AND ASSUMES NO LIABILITY TO YOU FOR ANY DAMAGES OF ANY TYPE IN CONNECTION WITH THESE MATERIALS OR ANY INTELLECTUAL PROPERTY IN THEM.

Microsoft may have patents, patent applications, trademarks, or other intellectual property rights covering subject matter within this documentation.  Except as provided in a separate agreement from Microsoft, your use of this document does not give you any license to these patents, trademarks or other intellectual property.

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places and events depicted herein are fictitious.

Microsoft, BranchCache, Windows, Windows PowerShell, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries and regions.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

You have no obligation to give Microsoft any suggestions, comments or other feedback ("Feedback") relating to the documentation. However, if you do provide any Feedback to Microsoft then you provide to Microsoft, without charge, the right to use, share and commercialize your Feedback in any way and for any purpose.  You also give to third parties, without charge, any patent rights needed for their products, technologies and services to use or interface with any specific parts of a Microsoft software or service that includes the Feedback.  You will not give Feedback that is subject to a license that requires Microsoft to license its software or documentation to third parties because we include your Feedback in them.

Contents

[The Planning and Design Series Approach 1](#_Toc303665952)

[Introduction to Internet Information Services 2](#_Toc303665953)

[Step 1: Determine the Project Scope 5](#_Toc303665954)

[Step 2: Characterize the Applications 6](#_Toc303665955)

[Step 3: Determine Windows Server 2008 Edition and Installation Mode 8](#_Toc303665956)

[Step 4: Determine the Location for Configuration and Content Storage 12](#_Toc303665957)

[Step 5: Determine the Fault Tolerance for the Infrastructure 16](#_Toc303665958)

[Step 6: Determine the Required Server Resources 20](#_Toc303665959)

[Conclusion 25](#_Toc303665960)

[Appendix A: Job Aids 26](#_Toc303665961)

[Appendix B: Collecting Performance Data 29](#_Toc303665962)

[Appendix C: IPD in Microsoft Operations Framework 4.0 32](#_Toc303665963)

[Version History 33](#_Toc303665964)

[Acknowledgments 34](#_Toc303665965)

#

# The Planning and Design Series Approach

This guide is one in a series of planning and design guides that clarify and streamline the planning and design process for Microsoft® infrastructure technologies.

Each guide in the series addresses a unique infrastructure technology or scenario. These guides include the following topics:

* Defining the technical decision flow (flow chart) through the planning process.
* Describing the decisions to be made and the commonly available options to consider in making the decisions.
* Relating the decisions and options to the business in terms of cost, complexity, and other characteristics.
* Framing the decision in terms of additional questions to the business to ensure a comprehensive understanding of the appropriate business landscape.

The guides in this series are intended to complement and augment the product documentation.

## Benefits of Using This Guide

Using this guide will help an organization to plan the best architecture for the business and to deliver the most cost-effective Internet Information Services (IIS) 7.0 or IIS 7.5 infrastructure technology.

Benefits for Business Stakeholders/Decision Makers:

* Most cost-effective design solution for an implementation. IPD eliminates over-architecting and over-spending by precisely matching the technology solution to the business needs.
* Alignment between the business and IT from the beginning of the design process to the end.

Benefits for Infrastructure Stakeholders/Decision Makers:

* Authoritative guidance. Microsoft is the best source for guidance about the design of Microsoft products.
* Business validation questions to ensure the solution meets the requirements of both business and infrastructure stakeholders.
* High-integrity design criteria that includes product limitations.
* Fault-tolerant infrastructure, where necessary.
* Proportionate system and network availability to meet business requirements.
* Infrastructure that is sized appropriately to meet business requirements.

**Benefits for** Consultants or Partners:

* Rapid readiness for consulting engagements.
* Planning and design template to standardize design and peer reviews.
* A “leave-behind” for pre- and post-sales visits to customer sites.
* General classroom instruction/preparation.

Benefits for the Entire Organization:

Using this guide should result in a design that will be sized, configured, and appropriately placed to deliver a solution for achieving stated business requirements, while considering the performance, capacity, manageability, and fault tolerance of the system.

# Introduction to Internet Information Services

This guide leads the reader through the process of planning an Internet Information Services (IIS) 7.0 or IIS 7.5 infrastructure. The first steps focus on determining requirements for the proposed infrastructure as well as introducing decisions that the reader might make. By addressing issues such as resource, application, and hosting requirements, the reader can determine the capacity required from the proposed hosting infrastructure.

This guide takes an application-centric approach to planning IIS: It focuses first and foremost on the applications to be hosted by IIS and their requirements and then applies these applications to specific IIS instances. Finally, it scales the hardware appropriately for each IIS instance.

The later steps in this guide shift the focus to the more organization-specific issues of planning points, workloads, architecture decisions, and choosing an edition of Windows Server® 2008 best suited to the business need.

## What’s New in Windows Server 2008 R2

This guide’s design process is valid for both Windows Server 2008 and Windows Server 2008 R2 environments.

Many features have been added or enhanced in Internet Information Services 7.5, which is the foundation of the Web Server role in Windows Server 2008 R2.

The following changes are available in the Web Server (IIS) role in Windows Server 2008 R2:

* Integrated extensions:
* WebDAV and FTP
* Request Filtering
* Administration Pack modules
* Management enhancements:
* Best Practices Analyzer
* Windows PowerShell™ provider and cmdlets
* Configuration logging and tracing
* Application hosting enhancements:
* Service hardening
* Managed service accounts
* Hostable Web Core
* Failed Request Tracing for FastCGI
* Enhancements to Microsoft .NET Framework support on Server Core
* BranchCache®

For more details, see “Changes in Functionality from Windows Server 2008 to Windows Server 2008 R2” at [http://technet.microsoft.com/en-us/library/dd391932(WS.10).aspx](http://technet.microsoft.com/en-us/library/dd391932%28WS.10%29.aspx).

For details on what changed from IIS 6.0 to IIS 7.0, see “Web Server (IIS) Role” at [http://technet.microsoft.com/en-us/library/cc730981(WS.10).aspx](http://technet.microsoft.com/en-us/library/cc730981%28WS.10%29.aspx).

## Assumptions

This guide assumes that the reader is familiar with technologies such as Windows Server 2008, web servers, web applications, and virtualization technology and their applications.

## IIS Design Process

The goal of this guide is to address several scenarios common to a transition to IIS.
It describes decisions that the organization must make and the issues it should address to ensure a smooth transition.

This guide addresses planning decisions and activities that must occur for IIS deployments. The six steps below represent the most critical design elements in a well-planned IIS design:

* Step 1: Determine the Project Scope
* Step 2: Characterize the Applications
* Step 3: Determine the Windows Server 2008 Edition and Installation Mode
* Step 4: Determine the Location for Configuration and Content Storage
* Step 5: Determine the Fault Tolerance for the Infrastructure
* Step 6: Determine the Required Server Resources

Some of these tasks represent decisions that the organization must make. Where this is the case, this guide presents a corresponding list of common options. Other steps in this list represent tasks that the organization must perform in order to complete the infrastructure design.

Figure1 provides a graphical overview of the steps in designing an IIS infrastructure.



Figure 1. The Internet Information Services 7.0 and IIS 7.5 infrastructure decision flow

## Applicable Scenarios

This guide addresses considerations that are related to planning and designing a successful IIS infrastructure. Those considerations include:

* Server upgrades.
* New deployments.

## Out of Scope

Although the planning information in this guide applies to many applications of the IIS infrastructure, certain details are outside the scope of this guide. Such details include:

* Operating the IIS environment.
* Maintaining the IIS environment.
* Disaster recovery and business continuity planning.
* Versions of IIS prior to IIS 7.0 or Internet Information Server.

# Step 1: Determine the Project Scope

Most infrastructure services can be broken down into smaller units to simplify planning. Even a very large multisite application needs a site-by-site implementation plan. It is important, therefore, to thoughtfully plan the scope of the project and address each collection of work one unit at a time. A unit could be a single application, a web farm, a department, a site, or any other logical collection that has a clear boundary. This approach has two main benefits:

* It allows a large implementation to be broken down into smaller, repeatable instances, many of which can be virtually identical to the others as long as site infrastructure permits.
* Even for smaller applications, the consistent discipline of planning unit-by-unit will enable rapid evaluation and planning.

The steps in this guide are designed to be performed once per collection or site. If the organization is creating a plan for multiple sites, iterate through this guide once for each.

If a single large implementation will be done—for instance, to support a centralized approach to the infrastructure—consider the single implementation as one large site for the purposes of this guide.

This guide is applicable to IIS 7.0 and IIS 7.5.

## Task 1: Create a Site List

Use Table A-1 in Appendix A: “Job Aids” to create a site list to aid in planning the implementation of the infrastructure. Use this list to ensure that a plan has been compiled for each site in the organization.

## Validating with the Business

Asking questions of the business can provide further information about scope:

* Does the scope of work recorded in the site list represent a reasonable risk to the business units affected, or should the scope be reduced?
* Are there any timing issues or other business constraints that will restrict the ability of IT to implement the new architecture?

## Step Summary

In this step, the project scope was determined and a site list was created by using the job aid in Table A-1 in Appendix A. After completing this step, the organization and the project team will have a clear understanding of which business areas are in scope of this infrastructure design.

# Step 2: Characterize the Applications

The planning and design process begins with the creation of an application inventory. When the inventory has been completed, planners can determine the resources required to support those applications. This analysis involves collecting and analyzing the technical requirements of each application to determine the requirements of the final infrastructure.

Often, an organization will use this guide to plan the rollout of an infrastructure for a single application. If this is the case, simply record the application requirements and move on to Step 3.

## Task 1: Create an Application Inventory

Before designing the infrastructure for a site, create a list of applications that the site will host on the IIS infrastructure. This guide recommends using this application inventory to plan capacity, fault tolerance, and Windows Server 2008 and IIS installation.

For each existing or planned application in the site, collect the following information for use in later steps of the planning and design process:

* Software requirements. Analyze each application to determine whether any special software or configuration requirements must be met on the IIS server. Record each application’s software editions and configuration requirements. One such requirement is whether an application needs the Microsoft .NET Framework to operate.
* Isolation requirements. If an application cannot coexist with other applications or needs a separate server for hosting portions of the application data or logic, be sure to record this need in the inventory.
* Maximum concurrent users per application. To aid in capacity planning, estimate the maximum number of users who will access the application at this site.
* Locations of the users. To aid in capacity planning, determine whether the users are on the internal network locally, internally but across a wide area network (WAN), or external to the organization. Internal users across a WAN may benefit from the use of BranchCache, which is discussed in Step 6.
* Requirements for multiple languages. If multiple languages will be supported, an application may need to be installed with an instance for each language. Be sure to record any requirements for this in the application inventory, because it can affect the number of IIS instances or servers required.
* Bandwidth requirements. For each application, record the projected bandwidth requirements needed to support business operations at peak usage times. Appendix B: “Collecting Performance Data” offers guidance on methods of collecting performance data.
* Disk capacity requirements. Record how much disk space each application requires.
* Disk performance. Record I/O operations per second (IOPS) required for each application.
* Fault tolerance. Evaluate and record each application to determine whether fault tolerance is possible and if it is required. Step 5 describes this process further.
* Authentication. Analyze each application and record any special authentication requirements, such as the use of certificates, basic, Kerberos protocol, or anonymous authentication.
* CPU. Record the application’s CPU requirements. List the percentage of CPU use required to support the application based on the listed hardware requirements or the results of application testing.
* RAM. Record the memory configuration recommended for the application.
* Planned growth. For all the above items, ensure that both planned and projected growth is taken into account. For later steps in this guide, use the information in the application inventory to make decisions about the operating system installation method, virtualization potential, and capacity planning.

Record the application inventory information in Table A-2 in Appendix A.

## Validating with the Business

In the above task, several pieces of information were recorded for each application. However, clarifying several items with the business can provide further insight:

* Is the business aware of any events that might change the peak user size? For example, if this is an external website, are there promotions or events that are expected to draw significantly more traffic than normal?
* Review the list of in-scope applications with the affected business groups to confirm the list with them. Are any new applications in planning that should also be considered in the scope of this project?

## Step Summary

After completing this step, the organization will have recorded the business, technology, and total resource requirements in Table A-2 in Appendix A: “Job Aids” for all the applications that the IIS infrastructure will host. This information can then be used to plan server count and placement.

## Additional Reading

* *Windows Server 2008 Resource Kit*: [www.microsoft.com/MSPress/books/10345.aspx](http://www.microsoft.com/MSPress/books/10345.aspx)
* Performance Tuning Guidelines for Windows Server 2008: [www.microsoft.com/whdc/system/sysperf/Perf\_tun\_srv.mspx](http://www.microsoft.com/whdc/system/sysperf/Perf_tun_srv.mspx)
* *Viewing Web Site Dependencies*: [http://learn.iis.net/page.aspx/424/viewing‑the‑dependencies‑for‑a‑web‑site](http://learn.iis.net/page.aspx/424/viewingthedependenciesforawebsite)

# Step 3: Determine Windows Server 2008 Edition and Installation Mode

Microsoft offers 32-bit and 64-bit editions of the Windows Server 2008 operating system. In addition, Windows Server 2008 supports the Server Core installation option of Windows Server 2008, which minimizes the operating system footprint.

This step guides the organization through the decision between 32-bit and 64-bit as well as the installation mode of the servers. The goal of this step is to record the Windows edition and installation mode for each application.

## Task 1: Determine the Edition

Microsoft provides Windows Server 2008 in 32-bit and 64-bit editions; each edition offers benefits for application hosting and infrastructure growth and development. Record whether the organization will use a 32-bit or 64-bit edition of the operating system in the infrastructure. If the organization requires both editions, identify which applications will use the 32-bit edition and which will use the 64-bit edition.

Windows Server 2008 is available in 32-bit or 64-bit editions.

Windows Server 2008 R2 is available only in 64-bit edition.

Use Table A-2 in Appendix A to record the application compatibility requirements information.

### Option 1: 32-Bit Edition

The 32-bit edition of Windows Server 2008 takes advantage of and runs on 32-bit platforms. When considering the 32-bit edition, the organization should take the following into account:

* Hardware repurposing. The organization can repurpose existing 32-bit hardware for IIS. Doing so can reduce the capital expenditure that purchasing new hardware requires.
* Application compatibility. Applications designed for 32-bit environments will continue to run on 32-bit editions of Windows Server 2008. In addition, this edition of the operating system supports 16-bit applications.

Note   Windows Server 2008 R2 is not available in 32-bit edition.

### Option 2: 64-Bit Edition

Microsoft provides a 64-bit edition of the Windows Server 2008 operating system, which leverages the features of the x64 architecture. If an organization is considering migrating from 32-bit to 64-bit editions, several key points can guide the decision:

* Performance. A 64-bit edition of the operating system can access more memory and other resources, allowing for higher levels of performance than the 32-bit edition.
* Application support. The 64-bit edition of Windows Server 2008 offers full support for both 64-bit applications and, in most cases, 32-bit applications. The 32-bit applications can usually run on 64-bit editions of Windows Server 2008 with potentially better performance than on a 32-bit edition of Windows Server 2008. No 16-bit applications are supported on 64-bit editions of Windows Server 2008.

Note   Windows Server 2008 R2 is only available in 64-bit edition.

### Evaluating the Characteristics

The table below evaluates the impact to performance of the server operating system version selection.

Table 1. Evaluating Characteristics

|  |  |  |
| --- | --- | --- |
| Version | Justification | Rating |
| 32-bit | Diminished performance benefit due to bottlenecks caused by the inability to fully leverage higher performance hardware. | → |
| 64-bit | Greater performance benefit due to the ability to leverage higher-performance hardware more effectively and completely. | ↑ |

## Task 2: Determine the Installation Mode

At this point, the organization must decide how to install Windows Server 2008 to support IIS and the given application inventory. Windows Server 2008 supports two installation modes for each platform edition. The first is the traditional full installation of the Windows Server 2008 operating system, which offers the familiar look and feel of previous versions of Windows Server. The second is the Server Core installation option, a compact version that does not use a graphical user interface (GUI) but supports several server roles, including IIS.

Use Table A-2 in Appendix A to record whether the organization will use the full installation or Server Core installation option of Windows Server 2008 in the infrastructure.

### Option 1: Full Installation

The full, traditional installation of Windows Server 2008 offers all services and features available for installation and configuration. Points to consider when evaluating applications for full installation are:

* GUI desktop. The full installation of Windows Server 2008 includes the Windows® GUI. Personnel administering the server locally may find the GUI easier to use than non-GUI versions.
* Upgrade option for legacy Windows Server installations. Eligible legacy Windows Server installations can be upgraded only to the full installation of Windows Server 2008.
* Microsoft .NET support. Windows Server 2008 and Windows Server 2008 R2 support Microsoft .NET in the full installation. Refer to the application inventory to determine which applications require Microsoft .NET.

Table 2. .NET Supported Architecture

|  |  |  |
| --- | --- | --- |
| Server Version | Server Core | Full Core |
| Windows Server 2008 | .NET not available | .NET available only in 64-bit architecture |
| Windows Server 2008 R2 | .NET available only in 64-bit architecture | .NET available only in 64-bit architecture |

### Option 2: Server Core Installation

The other Windows Server 2008 installation option is the Server Core installation, which gives organizations a new choice for running compact and specialized Windows Server 2008 installations:

* Compact environment. The Server Core installation option installs only those features that a particular role requires (for example, a domain controller or web server). Installing only necessary features and files needed to support a specific role, and the necessary components for that role, creates a smaller attack surface, and fewer features to patch results in reduced maintenance.
* No GUI. This option does not provide a GUI on the local servers. All local management is performed in a command prompt window.
* No Microsoft .NET support on Windows Server 2008 (non R2). This option does not support Microsoft .NET technology, which directly affects any applications that require the Microsoft .NET Framework. Generally, the lack of Microsoft .NET support means that the Server Core installation option of Windows Server 2008 can only host static HTML and classic Microsoft Active Server Pages (ASP) content. Windows Server 2008 R2 Server Core installation option does support Microsoft .NET technology. See Table 2 above for an at-a-glance view of .NET support.
* Not an upgrade option for legacy Windows Server installations. Organizations cannot upgrade existing Windows Server 2003 installations to a Server Core installation option of Windows Server 2008—this option requires a new installation.

Before considering this installation option, the organization must carefully examine where it will place the installation and consider each application’s specific requirements. For example, applications that require ASP.NET, or another form of Microsoft .NET support, are supported on the core installation of Windows Server 2008 R2, and the full installation of either product. However, they are not supported on the server core installation of Windows Server 2008.

### Evaluating the Characteristics

The table below evaluates the impact to security of the server operating system installation choice.

Table 3. Evaluating Characteristics

|  |  |  |
| --- | --- | --- |
| Installation Option | Justification | Rating |
| Full installation option | Default installation includes a variety of services, applications, and other features that increase the server’s attack surface. | → |
| Server Core installation option | Default installation does not include non-essential services, applications, or features and therefore reduces the server’s attack surface. | ↑ |

## Validating with the Business

When choosing the platform edition of Windows Server 2008 and how the organization will install it, be sure to review the platform and installation decisions with the business units that will be affected by them.

Does the decision inappropriately limit the future application strategy plans?

## Step Summary

In this step, the organization decided whether to move to a 64-bit edition of Windows Server 2008 or use a 32-bit edition for each server. Also, it was decided whether to use the full installation or Server Core installation option of Windows Server 2008 for each server. These decisions were recorded in Table A-2 in Appendix A.

# Step 4: Determine the Location for Configuration and Content Storage

IIS enables the central storage of site configuration settings in a schematized XML file. Instead of a metabase located on each individual system as existed in IIS 6.0, configuration can be stored in a flat file located at a remote, central location, and applied to all servers in the infrastructure. This option offers much better consistency in configuration settings; however, it also adds the risk that one poor configuration might apply to all servers, which could take the entire farm offline.

In this step, location and fault tolerance of configuration and content is explored. Use these tasks to plan this aspect of the IIS infrastructure.

## Task 1: Determine the Configuration and Content Location

IIS content and application configuration can be stored locally on each server or remotely in a central data store, with multiple servers accessing the configuration information from a single shared location. Likewise, application content can be stored either locally or remotely. The organization must determine where to store the configuration for each server and the content for each application. Centralized configuration and data storage offers benefits for fault tolerance and data consistency but can suffer total failure if the central data store is corrupted or lost. Take this into account as storage configuration is chosen. Use Table A-3 in Appendix A to record the location where configuration and content will be stored.

Note   Although it is possible to make this determination on an application-by-application basis, from a complexity and manageability standpoint, this guide recommends making a decision for all application data and all configuration data for each instance of IIS deployed.

### Option 1: Local Storage

With this option, configuration or content data is stored locally on the same machine as IIS. A local disk or a storage area network (SAN) can provide the disk storage. Using local storage provides the following advantages for both configuration and content:

* Availability. Locally stored data does not depend upon the availability of the remote file share. Thus, if one server’s data goes down, the entire IIS instance is not taken offline. However, the risk of locally stored content becoming corrupted may warrant a standby server or other backup configuration.
* Performance. Because the network is eliminated as a potential bottleneck, locally stored data generally sees improved performance over content stored remotely.
* Autonomy. By storing everything locally, the administrator can independently manage the data.

### Option 2: Remote File Shares

With this option, configuration or content data is stored on remote file shares, which can be hosted on remote file servers or on a network-attached storage system. Using remote file shares provides the following advantages for both configuration and content:

* Administration. Administration is simplified by having multiple servers pull information from a single location. However, there may be a risk if corrupt or inaccurate configuration data is created and used by multiple machines.
* Simplified backup. A central location allows for a simplified backup strategy, because only one storage location must be taken into consideration for the IIS instance.
* Scalability. Administrators can easily scale an IIS instance by adding another IIS server and pointing it to the remote store.
* Data synchronization. Because a single storage mechanism is used for each IIS instance, any changes made to the data are instantly available to other servers in the IIS instance.

If this option is chosen, there must be sufficient network bandwidth between the remote file share and the IIS servers handling clients.

Note   If configuration data is stored remotely but content is stored locally and the configuration share is unavailable, the servers will be unable to start correctly. Offline folder caching can be used in conjunction with remote file storage for configuration data to offset this issue.

### Evaluating the Characteristics

Table 4. Evaluating Characteristics

|  |  |  |
| --- | --- | --- |
| Complexity | Justification | Rating |
| Local storage | Each server’s local storage capacity must be sufficient to handle default IIS installations. | Low |
| Remote file share | Additional configuration is required for IIS to respond to client requests. | Medium |
| Cost | Justification | Rating |
| Local storage | Each server hosting its own content (configuration data tends to be small) will require additional disk capacity for the data. | Low |
| Remote file share | A remote file share that hosts content (configuration data tends to be small) will be able to address the needs of several servers using the capacity storage needed for just one server’s content. | Medium |

## Task 2: Determine the Configuration and Content Storage Requirements

Although IIS server and site configuration may not use a lot of space for storage, content storage needs can be significant. Organizations planning to deploy IIS should evaluate storage requirements of all applications in scope for the implementation and plan for growth in the storage needs based on anticipated business growth.

Storage performance is also important for achieving the best possible performance for web applications. Servers will cache static content in RAM, dramatically improving performance, but there are limits to the amount of content that can be cached. The performance of the storage infrastructure will be the gating factor in uncached performance.

In addition to the storage requirements for static content, data-driven applications will need sufficient read/write performance from their databases to provide optimal client experiences. This is also a function of storage performance, but it will be managed by those responsible for planning the database infrastructure and is out of scope for this guide.

In this task, storage capacity and performance are evaluated, and strategies for choosing optimal storage capacity and performance are presented.

### Storage Capacity

The storage requirements of the in-scope applications were recorded in Table A-2: “Application Inventory” in Appendix A. The next step involves planning adequately for each application’s growth. Using Table A-2, update the capacity value for each application to account for future growth requirements. In addition, space should be allocated for temporary files that may get created during the time when the application is running. The resulting numbers equal the total amount of existing and planned future capacity required for each application.

## Task 3: Determine the Configuration and Content Storage Fault Tolerance

If a content storage system becomes unavailable, the server will be unable to access data. In addition, configuration information stored with content cannot be updated. If content is stored in a shared location, ensuring the availability of the data is critical.

### Option 1: Built-in Storage

Network-attached storage systems typically offer multiple methods for ensuring data availability, reliability, and scalability. They can provide multiple hardware paths to the data, low-level redundant array of independent disks (RAID) capabilities, and fault-tolerant hardware. From a client system perspective, network-attached storage appears as a remote share.

### Option 2: Fault-Tolerant File Replication

If each server hosts content, use Distributed File System (DFS) Replication to keep the multiple storage systems synchronized. Likewise, content in a remote content storage system can use Distributed File System Namespace to achieve a measure of fault tolerance. If DFS Namespace is used, the availability of the data is increased, because DFS Namespace can redirect requests to another copy of the share if the targeted location is offline.

In Windows Server 2008 R2, DFS Replication supports clustering. For further details, see <http://blogs.technet.com/b/filecab/archive/2009/01/19/dfs-replication-what-s-new-in-windows-server-2008-r2.aspx> and [http://technet.microsoft.com/en-us/library/ee307957(WS.10).aspx](http://technet.microsoft.com/en-us/library/ee307957%28WS.10%29.aspx).

It is important to note with Windows Server 2008 without R2, however, that although DFS is supported on a server cluster (see Option 3 below), DFS Replication is not.

### Option 3: Server Clustering

Use server clustering to increase the fault tolerance of a single content storage system file share. With this option, the file share becomes a clustered resource running on a cluster with two or more computers. If the computer hosting the file share fails, the file share moves to a remaining active node.

Note   Although clustering the content storage system file share is supported, the product group does not recommend clustering for achieving high availability of IIS itself.

Note   Although a share hosted in a server cluster can become part of a DFS Namespace, the content of the share cannot be replicated using File Replication Service (FRS) or DFS Replication.

### Evaluating the Characteristics

Table 5. Evaluating Characteristics

|  |  |  |
| --- | --- | --- |
| Complexity | Justification | Rating |
| Built-in storage | Many devices, such as network-attached storage, are automatically configured to handle fault tolerance. | Low |
| Fault-tolerant file replication | Configuring and implementing file replication among the shares can be moderately difficult. Microsoft provides guidance for implementing this form of file protection using DFS Namespace and DFS Replication. | Medium |
| Server clustering | Server clustering tends to be extremely complex to set up because of the interaction among networks, shared storage, and specialized hardware and software configurations. | High |
| Cost | Justification | Rating |
| Built-in storage | The default space on a server is often sufficient for hosting the IIS data.  | Low |
| Fault-tolerant file replication | If the organization is using existing file servers, fault-tolerant file replication requires an additional server.  | Medium |
| Server clustering | The requirements of additional servers and shared storage make server clustering costly. | High |

## Validating with the Business

Review the decisions made for each application with the affected business units.

Is there an appropriate balance between the costs and the fault-tolerance capabilities?

## Step Summary

In this step, storage options and requirements were evaluated and recorded in Table A-2 and Table A-3 in Appendix A. This information will help define the plan for configuration of each IIS instance and ensure that proper availability is achieved for the IIS infrastructure.

# Step 5: Determine the Fault Tolerance for the Infrastructure

This section addresses the fault-tolerance options available to the organization. Planners can reference the information recorded in Step 2 to determine the most suitable availability option for each application. As each application is evaluated, record the fault-tolerance configuration in Table A-2 in Appendix A.

In addition to ensuring fault tolerance of servers and storage, it is critical to ensure that infrastructure services such as Domain Name System (DNS), backbone connectivity, and routing are designed for fault tolerance as well. This will help ensure that IIS applications remain available to clients regardless of single failures in the network infrastructure.

## Task 1: Determine the Fault-Tolerance Approach

If an application is hosted on IIS and considered a critical component of business operations, one technique for providing fault tolerance is load balancing. Certain load-balancing solutions provide fault tolerance at the TCP/IP level—that is, if the entire machine fails, the load-balancing cluster no longer sends requests to that system. However, application failures are not recognized, so client requests will still be sent to the affected server. Other load-balancing solutions can provide higher levels of fault tolerance by recognizing when the application has stopped responding. Additional protection can be provided through the use of multiple IP addresses assigned to a single DNS name, ensuring that in the event of loss of a single server or network segment, the site is still operable and the remaining servers continue to handle functionality.

Note   Load balancing does not prevent some users from experiencing the results of server failures. In both examples above, some users will experience the server outage when their requests are rerouted from the unavailable server to an available server.

### Option 1: Software-Based NLB

Network load balancing (NLB) is a cost-effective method for providing load balancing as well as a very basic level of fault tolerance and scalability. Up to 32 systems can be placed in a single software-based NLB cluster using Microsoft NLB. It has been observed in production that the effective performance of the system drops for cluster groups containing more than six members, so the organization might need to conduct independent verification testing.

Software-based NLB from Microsoft is limited to the TCP/IP stack. If the web server’s network is responding, the NLB cluster assumes the server is fully functioning and routes requests to the web server, even though the web service or application may be unresponsive.

### Option 2: Hardware Load Balancer

Providing access to the IIS group of servers and recognizing when a server has stopped responding to requests automatically requires a hardware load-balancing solution that supports HTTP. This level of configuration adds complexity to the overall deployment of the IIS servers.

Hardware load balancers also add costs to the overall hosting solution. Two or more hardware load balancers are necessary; if only one is implemented, the hardware load balancer becomes the single point of failure. Because client connections are handled by specialized hardware, hardware load balancers tend to scale to handle more concurrent client sessions than software-based load balancers. Many hardware load balancers also employ virtual IP technology, which presents the client with a single IP address and routing incoming requests to one of many servers in the background.

### Option 3: Application Request Routing

Microsoft Application Request Routing (ARR) for IIS is an extension that is available under Windows Supplemental License. ARR is a proxy-based routing module that forwards HTTP requests to application servers based on HTTP headers, server variables, and load-balancing algorithms. ARR further monitors the health of the application servers to provide high availability and scalability.

While ARR provides high availability and scalability for the pages being served, the overall deployment is not highly available or scalable because:

* ARR is a single point of failure.
* The ARR is the entry point to the back-end farm, so if the ARR is overloaded, the back-end farm will appear unresponsive to the user.

In order to reduce this impact, administrators should consider using multiple ARR servers with load balancers. If using NLB as the load balancer, NLB creates a single point of failure in that it is not aware of failures above the TCP/IP layer.

For more details on using ARR in conjunction with hardware load balancers, see <http://learn.iis.net/page.aspx/510/achieving-high-availability-and-scalability---arr-and-hardware-load-balancer/>.

For more details on using ARR in conjunction with NLB, see <http://learn.iis.net/page.aspx/511/achieving-high-availability-and-scalability---arr-and-nlb/>.

### Evaluating the Characteristics

Table 6. Evaluating Characteristics

|  |  |  |
| --- | --- | --- |
| Complexity | Justification | Rating |
| Software NLB | NLB is simple to implement. | Low |
| Hardware load balancer | Hardware load balancers tend to increase the complexity of the environment, because two are needed and additional knowledge is required. | Medium |
| Application Request Routing | ARR is simple to implement, and is built on top of IIS. | Low |

Table 6. Evaluating Characteristics (Continued)

|  |  |  |
| --- | --- | --- |
| Cost | Justification | Rating |
| Software NLB | NLB is available in all editions of Windows Server 2008. As a best practice, include an additional network adapter to all nodes in the cluster to create a private network for the cluster heartbeat. | Low |
| Hardware load balancer | Two or more hardware load balancers are needed to ensure fault tolerance. If only one hardware load balancer is used, it becomes the single point of failure. | High |
| Application Request Routing | ARR is available in all editions of Windows Server 2008. | Low |
| Fault tolerance | Justification | Rating |
| Software NLB | NLB provides fault tolerance only at the machine level. If the application layer fails, NLB will not detect it. | → |
| Hardware load balancer | Hardware load balancers are able to detect application-layer failures. | ↑ |
| Application Request Routing | ARR can detect application layer failures and when used in conjunction with hardware load balancers, provides a high level of fault tolerance and scalability. | ↑ |
| Security | Justification | Rating |
| Software NLB | NLB does not affect security if properly implemented. | → |
| Hardware load balancer | Hardware load balancers can increase the security of the infrastructure because of rudimentary packet-screening features. | ↑ |
| Application Request Routing | ARR can increase the security of the infrastructure when used with the IIS extension request filtering. | ↑ |

Use Table A-2 in Appendix A to record the fault-tolerance solution for each application.

## Task 2: Determine the Need for Additional Services

Any additional services that the applications rely on for operation, such as DNS, require fault tolerance. Identifying how to make these additional services fault tolerant is beyond the scope of this guide. However, for each required service, consider how to increase the fault tolerance of the service, and record the means of adding fault tolerance.

## Validating with the Business

Review the decisions made for each application with the affected business units.

Is there an appropriate balance between the costs and the fault tolerance capabilities?

## Step Summary

In this step, fault-tolerance options for the web servers were evaluated and recorded in Table A-2 in Appendix A. This information will help define the plan for configuration of each IIS instance and ensure proper availability is achieved for the IIS infrastructure.

## Additional Reading

* An Overview of Windows Clustering Technologies: Server Clusters and Network Load Balancing: [http://technet.microsoft.com/en-us/library/cc739634(WS.10).aspx](http://technet.microsoft.com/en-us/library/cc739634%28WS.10%29.aspx)
* Microsoft Knowledge Base article 240997 “Configuring Network Load Balancing”: <http://support.microsoft.com/kb/240997>

# Step 6: Determine the Required Server Resources

This step determines the size and number of servers required to serve the applications in scope as well as their scalability and fault-tolerance requirements. Identify requirements for each instance, and then scale the servers to meet those requirements. The output of this task will be a list of all server resources required for proper operation of the IIS infrastructure. Record this list in Table A-4 in Appendix A: “Job Aids.”

## Task 1: Determine Whether to Use Virtualization

The first task in determining server resource requirements is to decide whether to implement each application in a physical or virtual environment.

### Option 1: Physical Hardware

Physical hardware is the traditional choice in platform selection and the one that IT organizations typically have the most experience using. Physical hardware can provide flexibility in the type of hardware used in the IIS infrastructure, and can potentially handle more clients than a virtualized environment due to the lack of overhead that virtual services introduce.

### Option 2: Virtual Machine

Virtualization introduces flexibility into an environment by allowing virtual machines to be moved easily between hosts. Services that may not be compatible with each other can run on the same host because of the isolation that the virtual machine provides. However, virtualization overhead may affect the number of clients that a given virtual machine can support.

This guide describes several reasons to choose virtualization. However, additional planning is required for the physical computers that will host the virtual machines. The Infrastructure Planning and Design Guide for Windows Server Virtualization at <http://go.microsoft.com/fwlink/?LinkID=147617> provides the planning process for server virtualization.

### Evaluating the Characteristics

Table 7. Evaluating Characteristics

| Complexity | Justification | Rating |
| --- | --- | --- |
| Physical hardware | The unique characteristics of the physical hardware must be addressed. This, in addition to the setup of the hardware, makes this option more complex than using a virtual machine. | Medium |
| Virtual machine | The creation of a single virtual machine is less complex, because of standardized virtualized hardware. In addition, isolating applications in their own virtualized environments reduces the complexity of each application. | Low |

Table 7. Evaluating Characteristics (Continued)

|  |  |  |
| --- | --- | --- |
| Cost | Justification | Rating |
| Physical hardware | This option involves additional capital expenses as well as a time investment in setting up and maintaining the physical hardware. | High |
| Virtual machine | Use of virtual machines can reduce capital costs because large servers can safely share loads. | Low |

### Validating with the Business

Review the list of applications to be virtualized.

* Can the applications be licensed for use in a virtual environment?
* For each in-scope application, will the IIS infrastructure be physical or virtual?

## Task 2: Define the Number of IIS Instances

Until this point, this guide has dealt with the applications and their requirements for running on IIS. In this task, the applications are grouped into one or more IIS instances.

The first step in defining the number of IIS instances required is to set aside all applications that were identified as requiring isolation. Each application of this type will require a separate IIS instance.

The remaining list of applications will then be analyzed to see what applications can be combined together in the same IIS instances. The goal of this effort is to map applications to a specific server configuration in order to achieve the best resource utilization without overloading the server. Application combinations should take into account the CPU, disk, memory, fault tolerance, and network usage of each application to ensure that the combined list of applications stays below the capabilities of the platform that will host it. For this reason, several iterations of applications, hardware, or virtual configurations may need to be analyzed to find the optimal configuration. When an application is mapped to a server instance, it can be taken out of the list of applications. This process is repeated until all applications are assigned to an instance.

The process of going through this task and the next task, “Server Scaling Considerations,” may need to occur several times to find an optimal balance between the scaling of servers and the number of IIS instances. This can be necessary when considering the impact of multiple different hardware platform configurations on the capacity and performance of IIS instances.

The number of IIS instances is now equal to the number of instances requiring isolation plus the number of physical machines or virtual instances (virtual machines) required to accommodate all other applications. Use Table A-4 in Appendix A to record each IIS instance and its server configuration.

One thing to consider is that the final list of instances and implementations may need to be reviewed and rationalized against business criteria, and may be subject to tradeoffs. For example, if an organization is hosting 100 applications in IIS and all but one can be appropriately handled by scaling out and using software NLB, but the remaining one application’s requirements indicate the need for hardware NLB, the business may wish to review the conclusion in light of the cost and complexity of adding a cluster for a single application and accept the capabilities available for the application in a load-balanced instance.

## Task 3: Server Scaling Considerations

For each instance, the IIS servers and remote file servers (if used) must be scaled to handle the expected load. IIS can be bound by network, memory, CPU, and the speed with which it can read the content data from the disk. Planning for adequate resources is critical to ensuring satisfactory performance of the IIS instance.

The server scaling process can involve two methods of scaling:

* Scale up. This scaling method increases server resources to accommodate expected application loads.
* Scale out. This method adds additional servers to a load-balanced instance as client load increases.

Infrastructures can use a mix of both methods to define a platform that can accommodate application demands.

Use the sections below to calculate and record resource requirements for the infrastructure. Record the hardware configuration for each IIS instance in Table A-4 in Appendix A. For sites with access to installed applications, see Appendix B: “Collecting Performance Data*,*” which offers guidance on collecting relevant data to assist with resource planning.

### Use of BranchCache for Intranet Sites

BranchCache is a new feature available with Windows Server 2008 R2 and Windows 7 clients. When BranchCache is enabled on the IIS server, a copy of the content retrieved from the web server is cached within the branch office. If another client in the branch requests the same content, the client can download it directly from the local branch network instead of retrieving it from the web server across the WAN.

This can potentially reduce the load on the intranet web server. See “BranchCache in Windows 7 and Windows Server 2008 R2 Overview” at [http://technet.microsoft.com/en-us/library/dd755969(WS.10).aspx](http://technet.microsoft.com/en-us/library/dd755969%28WS.10%29.aspx) for more details. If BranchCache will be used, adjust the resource requirements accordingly.

### CPU

IIS controls CPU utilization primarily by monitoring worker processes and application pools. Use the application characterization information recorded in Step 2 as a baseline for determining the architecture and number of CPUs.

Additional CPUs can be added to the base form factor for the following reasons:

* High CPU use. If under moderate load testing CPU use is trending abnormally high, add processors to the base form factor and retest to help increase the CPU capabilities of the system.
* Need for additional CPU-intensive applications. If additional CPU workloads must be placed on the server, add processors to the server to address the increased workload requirements.

If additional services are placed on the same server as IIS, adjust the type, number, and speed of processors to handle the additional load. Additional performance monitoring and testing may be required to determine whether the processor is creating a bottleneck.

Note   Refer to Table A-2 in Appendix A when evaluating CPU requirements.

### Memory

IIS attempts to cache the application content in memory after the initial client request. This behavior decreases the response time for additional requests, because the server does not read the information from disk again. Increasing the memory capacity of the server to allow caching of more data can improve server performance.

Use the application characterization information recorded in Step 2 as a baseline for determining the amount of RAM for servers supporting the IIS infrastructure.

Note   Because of limitations on the amount of physical RAM that a 32-bit system can reference, it may be beneficial to run IIS on a 64-bit hardware and operating system.

### Network

For each server, determine the size and number of network adapters. The available bandwidth and the latency of the network between the IIS instance and the location of the content data have the greatest impact on the performance of the infrastructure.

Using Network Driver Interface Specification (NDIS) 5 (and later) network cards with miniport drivers will enhance IIS servers’ network performance. This option will provide offload capabilities and improve system performance by up to 30 percent. IIS leverages all of the following offload capabilities:

* Checksum offload. The networking stack can offload the calculation and validation of both TCP and User Datagram Protocol (UDP) checksums on send and receive operations for both IP version 4 and IP version 6.
* Segmentation offload. Giant Send Offload (GSO) offloads the segmentation of large TCP packets to the network adapter.
* TCP offload engine. The TCP Offload Engine (TOE) enables a network adapter that has the appropriate capabilities to offload the entire network stack.
* Receive-side scaling. Receive-side scaling enables the network driver, in conjunction with the network adapter, to distribute incoming packets among processors so that packets belonging to the same TCP connections are handled by the same processor.
* Start with one network adapter. Add network adapters for the following reasons:
* Network bottlenecks. If the network adapter becomes the bottleneck for the server, install additional network adapters on the server to provide additional bandwidth for the system.
* Traffic segmentation and management. Some configurations, particularly Internet-facing servers, are configured with multiple network adapters. One or more adapters handle the client traffic, while another set handles back-end communication (for example, with database servers or management servers). This allows for additional security policies to be put in place to control the type of traffic the adapters handle.

Determine the type of network adapter required for the load to be placed against the system. If multiple network adapters are used in the server to increase the bandwidth, ensure that the network switching hardware is compatible with the configuration. For each server, record the type and number of network adapters being used in Table A-2 in Appendix A.

## Step Summary

For each IIS instance, the organization has identified the requirements and decided whether to use physical hardware or virtual machines for the IIS infrastructure. The configuration and content storage location have been identified. Finally, the size and number of servers has been determined for each IIS instance. The decisions in this step were recorded in Table A-2 and Table A-4 in Appendix A.

## Additional Reading

* Infrastructure Planning and Design Guide for Windows Server Virtualization: <http://go.microsoft.com/fwlink/?LinkID=147617>
* Performance Tuning Guidelines for Windows Server 2008: [www.microsoft.com/whdc/system/sysperf/Perf\_tun\_srv.mspx](http://www.microsoft.com/whdc/system/sysperf/Perf_tun_srv.mspx)
* *TechNet Webcast: Windows Server 2008: Server Virtualization Solution Scenario (Level 300)*: [http://msevents.microsoft.com/CUI/WebCastEventDetails.aspx?culture=en‑US&EventID=1032336484&CountryCode=US](http://msevents.microsoft.com/CUI/WebCastEventDetails.aspx?culture=en-US&EventID=1032336484&CountryCode=US)
* IIS Download Center: [www.iis.net/downloads/default.aspx?tabid=34&i=1467&g=6](http://www.iis.net/downloads/default.aspx?tabid=34&i=1467&g=6)

# Conclusion

This guide has outlined the step-by-step process for planning an IIS infrastructure. By proceeding from step to step, major decisions regarding the IIS infrastructure were revealed and explained. The guide has explained how to record choices of location, implementation method, server platform, server resources, and fault tolerance and make them available to infrastructure planners.

Using the information recorded from the steps completed in this guide, the organization can effectively deploy an IIS infrastructure with assurance that it is properly planned and prepared for a successful deployment.

# Appendix A: Job Aids

Use the job aids that follow to record the information required for the organization’s IIS rollout.

Table A-1. Site List

|  |  |  |
| --- | --- | --- |
| Site name | Notes | Plan completed |
| Engineering | CAD group intranet |  |
| Internet Promotions | Campaign application on Internet |  |
| Denver | Denver office intranet |  |
|  |  |  |
|  |  |  |
|  |  |  |

Table A-2. Application Inventory

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Application name | Application 1 |  |  |  |
| CPU % | 20 |  |  |  |
| CPU count | 1 |  |  |  |
| CPU architecture (32\64) | 32 |  |  |  |
| Memory required | 2 GB |  |  |  |
| Disk space: |  |  |  |  |
|  Capacity | 100M |  |  |  |
|  IOPS | 325 |  |  |  |
| Network throughput | 2.4 Mbps |  |  |  |
| Special/unique software requirement | Microsoft .NET |  |  |  |
| Special/unique configuration requirement | CDOSYS |  |  |  |
| Peak users | 100 |  |  |  |

Table A-2. Application Inventory (Continued)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Growth requirements: |  |  |  |  |
|  User growth: |  |  |  |  |
|  Planned | 20%/year |  |  |  |
|  Projected | 10%/year |  |  |  |
|  Data needs | 20%/year |  |  |  |
| Isolation requirements | Database on separate server (PCI-DSS) |  |  |  |
| Location of the users | External |  |  |  |
| Language requirements | English only |  |  |  |
| Fault tolerance | None required |  |  |  |
| Authentication | Application |  |  |  |
| Windows core or full | Core |  |  |  |
| Application content location | N |  |  |  |
| Application content capacity | 30 GB |  |  |  |
| Application content IOPS | 325 |  |  |  |
| Application content fault tolerance | Built in |  |  |  |
| Application fault tolerance | Hardware NLB |  |  |  |
| Additional service requirements | None |  |  |  |
| Physical or virtual | Virtual machine |  |  |  |
| Instance\server name hosting app | Server2 |  |  |  |

Table A-3. IIS Configuration and Content Storage

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IIS instance | Configuration location | Configuration size | Content location | Content size |
| Instance 1 | \\Server\share | 100 MB | \\Server\share | 12 GB |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Use the following table to record the number and configuration of servers to be used to support IIS instances for the site. If multiple sites are to be planned, this information can be rolled up to a master summary.

Table A-4. Site Infrastructure Summary

|  |  |  |  |
| --- | --- | --- | --- |
| IIS instance | Instance 1 | Instance 2 |  |
| Virtual machine(Y/N) | N | Y |  |
| Applications | Application 1Application 3 | Application 2 |  |
| Number of servers | 4 | 2 |  |
| CPU | 2 | 1 |  |
| RAM | 4 GB | 6 GB |  |
| Storage size | 4 GB | 12 GB |  |
| Network adaptersInt/Ext | 1/1 | 1/1 |  |
| Storage type(conf/ cont) | Central/ Central | Local/ Local |  |
| Fault tolerance | Hardware LB | None |  |

# Appendix B: Collecting Performance Data

When characterizing applications, keep in mind their unique natures. Each application will have its own specific resource utilization profile and other factors that make up its footprint. Each application can be monitored in a reference production environment or in a preproduction environment, or it can be tested in a lab. The goal of this kind of testing and monitoring is to gather meaningful data that can then be used to properly plan and allocate resources.

For each application, record the following requirements in Table A-2 in Appendix A:

* CPU resources
* Memory
* Network bandwidth
* Disk space

Include information from sources such as:

* Similar applications or technologies.
* Key stakeholders with knowledge needed to estimate requirements or make an educated guess.
* Technical data that developers provide about specific requirements for the application to function.
* Analysis of features in the application and what is needed in IIS to support these features.

In the event the organization is evaluating a new application, IT pros should perform lab-based testing, then compare the data gathered from testing to intended performance goals and refine the configuration until the desired level of performance is reached.

The following sections describe in more detail how to measure and record each resource requirement.

## CPU

Over-committing CPU resources can adversely affect all workloads on the same host server, causing significant performance issues for a large number of users. Because CPU resource usage patterns can vary significantly, no single metric or unit can quantify total resource requirements. One approach, however, is to collect CPU requirement specifications for particular applications and workloads. The table below provides the Performance Monitor counter to collect over time.

Table B-1. Performance Monitor Statistics

|  |  |  |
| --- | --- | --- |
| Object | Counter | Instance |
| Process | % Processor Time | Process Name |

Document the CPU that the application is running on, and express the CPU demand as a percentage.

## Memory

Applications that are allotted too little memory experience frequent disk page faults, resulting in decreased performance and additional disk resource use. In contrast, allocating too much physical memory leaves physical hardware resources unused, leading to lower overall host server use.

Collect memory use information when the system is running at peak load to ensure that the appropriate amount of memory is allotted; record the results in Table A-2 in Appendix A. The table below provides the Performance Monitor counter data to collect.

Table B-2. Performance Monitor Statistics for Required Memory

|  |  |  |
| --- | --- | --- |
| Object | Counter | Instance |
| Process | Working Set ‑ Private | Process Name |

## Disk

Proper planning for a disk subsystem relies both on disk capacity requirements as well as performance requirements.

### Disk Capacity

Every application requires disk space to store program, data, and other supporting file types. For the application inventory, only consider the space that the application uses. Common storage requirements include:

* Application-related storage space
* User data storage
* Databases and other required files

For each application, record the amount of disk space that the application uses.

### Disk Performance

To determine the actual disk performance, measure and record the physical I/O that occurs over a period of a time, then calculate the IOPS—that is, the total number of I/O operations that occur per second. Plot this value over time to determine requirements at peak usage. By understanding the IOPS that the application requires, IT can design the disk subsystem to provide the necessary performance for the running application.

Use Performance Monitor to measure the current application IOPS, as shown in Table B-3. However, note that that measurement does not indicate whether the disk subsystem has a bottleneck. To see whether the system is disk-bound, look at the queue length of the physical disk. The queue length should be zero on a well-performing system.

Note   Because physical disk counters do not exist per application, obtain these statistics by monitoring the server with the application started and with the application stopped to find the difference in the measurements. Conduct enough samples to ensure that the IOPS for the application itself are being detected.

Table B-3. Performance Monitor Statistics for Disk Performance

|  |  |  |
| --- | --- | --- |
| Object | Counter | Instance |
| Physical disk | Disk Reads/sec | \_Total |
| Physical disk | Disk Writes/sec | \_Total |
| Physical disk | Current Disk Queue Length | \_Total |

Total the Physical Disk counters from Table B-3 to calculate the IOPS usage for each application, and then record IOPS in Table A-2 in Appendix A.

## Network

Most workloads require access to production networks to ensure communication with required applications and services and to communicate with users. Network requirements usually include throughput—that is, the total amount of traffic that passes a given point on a network connection per unit of time. The presence of multiple network connections is sometimes another requirement, because workloads might require access to multiple networks that must remain secure. Examples include connections for:

* Public network access
* Networks for performing backups and other maintenance tasks
* Dedicated remote-management connections
* Network adapter teaming for performance and failover
* Connections to the physical host server
* Connections to network-based storage arrays

Table B-4 describes the Performance Monitor counter to collect and graph over time to record peak usage. Record each application’s required throughput for each network connection in Table A-2 in Appendix A.

Table B-4. Performance Monitor Statistics for Network Performance

|  |  |  |
| --- | --- | --- |
| Object | Counter | Instance |
| Network interface | Bytes Total/sec | (Specific network adapters) |

Note   Because network interface counters do not exist per application, obtain these statistics by monitoring the server with the application started and with the application stopped to find the difference in the measurements. Conduct enough samples to ensure that the throughput requirements for the application itself are being detected.

# Appendix C: IPD in Microsoft Operations Framework 4.0

Microsoft Operations Framework (MOF) offers integrated best practices, principles, and activities to assist an organization in achieving reliable solutions and services. MOF provides guidance to help individuals and organizations create, operate, and support technology services, while helping to ensure the investment in technology delivers expected business value at an acceptable level of risk. MOF’s question-based guidance helps to determine what is needed for an organization now, as well as providing activities that will keep the organization running efficiently and effectively in the future.

Use MOF with IPD guides to ensure that people and process considerations are addressed when changes to an organization’s technology services are being planned.

* Use the Plan Phase to maintain focus on meeting business needs, consider business requirements and constraints, and align business strategy with the technology strategy. IPD helps to define an architecture that delivers the right solution as determined in the Plan Phase.
* Use the Deliver Phase to build solutions and deploy updated technology. In this phase, IPD helps IT pros design their technology infrastructures.
* Use the Operate Phase to plan for operations, service monitoring and control, as well as troubleshooting. The appropriate infrastructure, built with the help of IPD guides, can increase the efficiency and effectiveness of operating activities.
* Use the Manage Layer to work effectively and efficiently to make decisions that are in compliance with management objectives. The full value of sound architectural practices embodied in IPD will help deliver value to the top levels of a business.



Figure C-1. The architecture of Microsoft Operations Framework (MOF) 4.0

# Version History

|  |  |  |
| --- | --- | --- |
| Version | Description | Date |
| 2.2 | Minor updates to document for current template. | November 2011 |
| 2.1 | Added in Step 4, Task 3, Option 2, “In Windows Server 2008 R2, DFS Replication supports clustering.”Changed references from DFS to DFS Namespace.Minor updates to this guide to reflect current IPD template. | July 2010 |
| 2.0 | Added “What’s New in Windows Server 2008 R2” section. Minor updates to this guide to reflect current IPD template.Added Step 1, Task 1: Create a Site List.Added additional item to collect in Step 2, Task 1: “Location of the Users.”Added multiple references throughout for Windows Server 2008 R2 only available in 64-bit edition.Added several references throughout for Windows Server 2008 R2 Core installation supporting Microsoft .NET.Added Step 5, Task 1, Option 3: Application Request Routing.Added new item in Step 6, Task 3 for “Use of BranchCache for Intranet Sites.” | July 2009 |
| 1.0 | First release. | June 2008 |

# Acknowledgments

The Solution Accelerators–Management and Infrastructure team acknowledges and thanks the people who produced the Infrastructure Planning and Design Guide for Internet Information Services 7.0 and IIS 7.5. The following people were either directly responsible for or made a substantial contribution to the writing, development, and testing of this guide.

**Contributors:**

* Jude Chosnyk – GrandMasters
* Charles Denny – *Microsoft*
* Dave Field – *Studio B Productions*
* Michael Kaczmarek – *Microsoft*
* Robin Maher – *Microsoft*
* Sean-Philip Oriyano – *Studio B Productions*
* Fergus Stewart – *Microsoft*
* Melissa Stowe – *Microsoft*

**Reviewers:**

* Jerry Honeycutt – Studio B Productions
* Timothy Landers – Studio B Productions
* Srinivas Somu – *Microsoft*
* Won Yoo – *Microsoft*

**Editors:**

* Michelle Anderson – Studio B Productions
* Vernon Barr *– Volt Technical Services*
* Laurie Dunham – *Xtreme Consulting Group*
* Ruth Preston – *Volt Technical Services*
* Pat Rytkonen – *Volt Technical Services*

## Feedback

Please direct questions and comments about this guide to ipdfdbk@microsoft.com.