Target Audience

Microsoft Technology Associate

The Microsoft Technology Associate (MTA) is Microsoft's newest suite of technology certification exams that validate fundamental knowledge needed to begin building a career using Microsoft technologies.

Successful candidates earn MTA certificates as well as access to benefits on the Microsoft Certification member site.

This program:

- is targeted primarily at students who attend high schools and two-year colleges.
- provides an appropriate entry point to a future career in technology.
- assumes some hands-on experience or training but does not assume on-the-job experience.

Minimally Qualified Candidate

The minimally qualified candidate typically:

- is a student who attends high school or a two-year college.
- has worked with desktop and application virtualization.
Objective Domain

1. Understanding Operating System Configurations
   1.1. Configure Control Panel options.
        This objective may include but is not limited to: configuring administrative tools, configuring accessibility options
   1.2. Configure desktop settings.
        This objective may include but is not limited to: configuring gadgets, profiles, display settings, shortcuts, and Aero configurations and capabilities
   1.3. Understand native applications and tools.
        This objective may include but is not limited to: understanding Windows Internet Explorer, the snipping tool, Windows Media Player, Windows Media Center, and MSCONFIG
   1.4. Understand mobility.
        This objective may include but is not limited to: understanding Sync Center, Windows Mobility Center, and Remote Desktop
   1.5. Understand remote management and assistance.
        This objective may include but is not limited to: understanding MMC, and Windows PowerShell

2. Installing and Upgrading Client Systems
   2.1. Identify Windows operating system editions.
        This objective may include but is not limited to: identifying system requirements, using PC Upgrade Advisor
   2.2. Identify upgrade paths.
        This objective may include but is not limited to: identifying upgrade paths from Windows XP, Windows Vista, and other operating systems; identifying application compatibility
   2.3. Understand installation types.
        This objective may include but is not limited to: understanding removable media installations (DVD, ZTI, LTI, and USB), cloud and network installations, and product identification keys
   2.4. Understand virtualized clients.
        This objective may include but is not limited to: understanding Windows XP Mode, Remote Desktop, and Remote Desktop Services

3. Managing Applications
   3.1. Understand application installations.
        This objective may include but is not limited to: understanding local vs. network applications, Group Policy, and application removal
3.2. Understand user account control (UAC).
   This objective may include but is not limited to: understanding standard user vs. administrative user, understanding types of UAC prompts and levels
3.3. Remove malicious software.
   This objective may include but is not limited to: understanding Windows Defender, Action Center, the Malicious Software Removal tool, Windows Registry, and Microsoft Forefront Endpoint Protection
3.4. Understand services.
   This objective may include but is not limited to: understanding service start-up types, service accounts, and service dependencies
3.5. Understand application virtualization.
   This objective may include but is not limited to: understanding Med-V and VDI

4. Managing Files and Folders
   4.1. Understand file systems.
      This objective may include but is not limited to: understanding FAT, FAT32, NTFS, and 32 bit vs. 64 bit
   4.2. Understand file and print sharing.
      This objective may include but is not limited to: understanding NTFS and share permissions, HomeGroup, print drivers, and effective permissions; creating public, basic, and advanced shares; mapping drives
   4.3. Understand encryption.
      This objective may include but is not limited to: understanding BitLocker, encrypting file systems (EFS), and compression
   4.4. Understand libraries.
      This objective may include but is not limited to: understanding offline files, adding multiple local locations to a library, adding networked locations

5. Managing Devices
   5.1. Connect devices.
      This objective may include but is not limited to: connecting plug-and-play devices, connecting and disconnecting printers, installing third-party software for devices
   5.2. Understand storage.
      This objective may include but is not limited to: understanding disk types (NTFS, FAT, etc.), security (encryption), storage device types (eSATA, USB, USB 2.0, IEEE 1394, iSCSI), storage drive types (basic, primary, extended, logical, dynamic disk, VHDs), and cloud storage (Windows Live SkyDrive, OneNote to SkyDrive, Live mesh)
   5.3. Understand printing devices.
      This objective may include but is not limited to: understanding local printers, network printers, print queues, print-to-file, and Internet printing
5.4. Understand system devices.
   This objective may include but is not limited to: understanding video, audio, and infrared input devices, understanding Device Manager

6. Understanding Operating System Maintenance

6.1. Understand backup and recovery methods.
   This objective may include but is not limited to: understanding local, online, and automated backup methods; understanding backup options; understanding System Restore, recovery boot options such as Last Known Good, and various Safe Mode options

6.2. Understand maintenance tools.
   This objective may include but is not limited to: understanding Disk Defragmenter, Disk Cleanup, Task Scheduler, Action Center, and System Information

6.3. Understand updates.
   This objective may include but is not limited to: understanding Windows updates, Microsoft updates, and hotfixes
Exam Design

The Basics
This is a Microsoft Technology Associate (MTA) exam designed to assess candidates’ knowledge of fundamental software development concepts and basic programming skills. MTA is a new certification under the Microsoft Certification Program that validates the foundational knowledge needed to begin building a career in Microsoft technologies. It can also serve as a stepping stone to the Microsoft Certified Technology Specialist exams. Successful candidates for this exam will earn an MTA certification as well as access to benefits of the Microsoft Certification Program. The primary target audience for the MTA certification is students attending high schools and two-year colleges.

We are specifying an item pool of 75 unique multiple-choice items, which will be used on 1 form. Approximately 70 percent of the items should be written to the knowledge/comprehension level and about 30 percent to the application level. For more information about cognitive levels, refer to the Cognitive Domain in Bloom’s Taxonomy.

Categories in the cognitive domain of Bloom’s Taxonomy (Anderson & Krathwohl, 2001)

The following anatomy is required of each knowledge-level item in this exam:
- Question Statement (ex: What should you do?)
- Answer Choices (Preferably Multiple Choice, choose 1, with 3 distracters)
  - Ex: Which of the following is a valid ASP.NET variable name? A. _foo; B. &foo; C. foo#; D. foo 1

The following anatomy is required of each application-level item in this exam:
- Concise scenario, including any constraints/requirements necessary to make distracter answers 100% incorrect
- Goal Statement (You need to...)
- Question Statement (ex: What should you do?)
- Answer Choices (Preferably Multiple Choice, choose 1, with 3 distracters)

**Target Audience**

Candidates for this exam are seeking to prove core software development skills. Before taking this exam, candidates should have a solid foundational knowledge of the topics outlined in this preparation guide. It is recommended that candidates be familiar with the concepts of and have hands-on experience with the technologies described here either by taking relevant training courses or by working with tutorials and samples available on MSDN® and in Microsoft® Visual Studio®.

**Objective Domain**

1. **Understanding Core Programming**
   
   1.1. Understand computer storage and data types.  
       *This objective may include but is not limited to:*  
       how a computer stores programs and the instructions in computer memory; memory stacks and heaps; memory size requirements for the various data storage types; numeric data and textual data
   
   1.2. Understand computer decision structures.  
       *This objective may include but is not limited to:*  
       various decision structures used in all computer programming languages; If decision structures; multiple decision structures such as If...Else and switch/Select Case; reading flowcharts; decision tables; evaluating expressions
   
   1.3. Identify the appropriate method for handling repetition.  
       *This objective may include but is not limited to:*  
       For loops, While loops, Do..While loops, and recursion
   
   1.4. Understand error handling.  
       *This objective may include but is not limited to:*  
       structured exception handling

2. **Understanding Object-Oriented Programming**
   
   2.1. Understand the fundamentals of classes.  
       *This objective may include but is not limited to:*  
       properties, methods, events, and constructors; how to create a class; how to use classes in code
   
   2.2. Understand inheritance.  
       *This objective may include but is not limited to:*  
       inheriting the functionality of a base class into a derived class
   
   2.3. Understand polymorphism.
This objective may include but is not limited to: extending the functionality in a class after inheriting from a base class; overriding methods in the derived class

2.4. Understand encapsulation.
This objective may include but is not limited to: creating classes that hide their implementation details while still allowing access to the required functionality through the interface; access modifiers

3. **Understanding General Software Development**
   3.1. Understand application life cycle management.
       This objective may include but is not limited to: phases of application life cycle management; software testing

3.2. Interpret application specifications.
       This objective may include but is not limited to: reading and translating application specifications into prototypes, code, and components

3.3. Understand algorithms and data structures.
       This objective may include but is not limited to: arrays, stacks, queues, linked lists, and sorting algorithms; performance implications of various data structures; choosing the right data structure

       NOT: algorithm analysis

4. **Understanding Web Applications**
   4.1. Understand Web page development.
       This objective may include but is not limited to: HTML, Cascading Style Sheets (CSS), JavaScript

4.2. Understand Microsoft ASP.NET Web application development.
       This objective may include but is not limited to: page life cycle; event model; state management; client-side vs. server-side programming

4.3. Understand Web hosting.
       This objective may include but is not limited to: creating virtual directories and Web sites, deploying Web applications; understanding the role of Internet Information Services

4.4. Understand Web services.
       This objective may include but is not limited to: Web services that will be consumed by client applications; accessing Web services from a client application; SOAP and Web Service Definition Language (WSDL)

5. **Understanding Desktop Applications**
   5.1. Understand Windows® Forms applications.
This objective may include but is not limited to: Windows Forms event model; visual inheritance; UI design; use of Multiple Document Interface(MDI) and Single Document Interface (SDI) applications

5.2. Understand console-based applications.  
This objective may include but is not limited to: characteristics and capabilities of console-based applications

5.3. Understand Windows Services.  
This objective may include but is not limited to: characteristics and capabilities of Windows Service

6. Understanding Databases

6.1. Understand relational database management systems.  
This objective may include but is not limited to: characteristics and capabilities of database products; database design; Entity Relationship Diagrams (ERDs); normalization concepts

6.2. Understand database query methods.  
This objective may include but is not limited to: structured query language (SQL), creating and accessing stored procedures, updating data, selecting data

6.3. Understand database connection methods.  
This objective may include but is not limited to: connecting to various types of data stores such as flat file; XML file; in-memory object; resource optimization
Exam 98-362
TA: Windows Development Fundamentals

Exam Design

The Basics
This is a Microsoft Technology Associate (MTA) exam designed to assess candidates’ fundamental knowledge of and skills with Windows® development. MTA is a new certification under the Microsoft Certification Program that validates the foundational knowledge needed to begin building a career in Microsoft technologies. It can also serve as a stepping stone to the Microsoft Certified Technology Specialist exams. Successful candidates for this exam will earn an MTA certification as well as access to benefits of the Microsoft Certification Program. The primary target audience for the MTA certification is students attending high schools and two-year colleges.

We are specifying an item pool of 75 unique multiple-choice items, which will be used on 1 form. Approximately 70 percent of the items should be written to the knowledge/comprehension level and about 30 percent to the application level. For more information about cognitive levels, refer to the Cognitive Domain in Bloom’s Taxonomy.

Categories in the cognitive domain of Bloom’s Taxonomy (Anderson & Krathwohl, 2001)

The following anatomy is required of each knowledge-level item in this exam:
- Question Statement
- Answer Choices (Preferably Multiple Choice, choose 1, with 3 distracters)
  - Ex: Which of the following is a valid ASP.NET variable name? A. _foo; B. &foo; C. foo#; D. foo 1

The following anatomy is required of each application-level item in this exam:
- Concise scenario, including any constraints/requirements necessary to make distracter answers 100% incorrect
Target Audience

Candidates for this exam are seeking to prove Windows programming and application development knowledge and skills. Before taking this exam, candidates should have a solid foundational knowledge of the topics outlined in this preparation guide. It is recommended that candidates be familiar with the concepts of and have hands-on experience with the technologies described here by either by taking relevant training courses or by working with tutorials and samples available on MSDN® and in Microsoft® Visual Studio®. Candidates are expected to have some experience with a Microsoft .NET language such as C# or Microsoft Visual Basic®.NET.

Candidates for this exam are in the process of expanding their knowledge and skills in the following areas:

- various types of applications that run on Windows
- how to create graphical user interface (GUI) applications that run on Windows by using Windows Forms or Windows Presentation Foundation (WPF)
- how Windows Services are programmed and hosted on a computer that runs Windows
- how to access data from various sources for use in a Windows-based application
- how to successfully deploy a Windows application to target computers

Objective Domain

1. Understanding Windows Programming Basics

1.1. Identify Windows application types.

   *This objective may include but is not limited to: Windows Forms, Windows Presentation Foundation (WPF), Windows Services, and Win32® applications*

1.2. Implement user interface design.

   *This objective may include but is not limited to: core user interface design principles for creating graphical-based applications*

1.3. Create Windows-based applications by using Visual Studio.

   *This objective may include but is not limited to: project types, importance of the various aspects of a Windows Application project*
2. Creating Windows Forms Applications

2.1. Create and handle events.
    *This objective may include but is not limited to:* methods for creating events in an application; handling events raised in an application

2.2. Understand Windows Forms inheritance.
    *This objective may include but is not limited to:* implementing forms inheritance in applications for visual inheritance

2.3. Understand how to create new controls and extend existing controls.
    *This objective may include but is not limited to:* creating a new GUI control or inheriting functionality from an existing control

2.4. Validate and implement user input.
    *This objective may include but is not limited to:* implementing the correct user input model based on application design and requirements; accepting keyboard and mouse input; validating user input through GUI controls such as text boxes and dialog controls

2.5. Debug a Windows-based application.
    *This objective may include but is not limited to:* using breakpoints and debugging techniques to identify issues in code; debugging a Windows Services application

3. Creating Windows Services Applications

3.1. Create a Windows Services application.
    *This objective may include but is not limited to:* inheriting the ServiceBase class; writing code in the Main method; overriding the OnStart and OnStop procedures

3.2. Install a Windows Services application.
    *This objective may include but is not limited to:* creating installers for Windows Services; installing services on a target computer

4. Accessing Data in a Windows Forms Application

4.1. Understand data access methods for a Windows Application.
    *This objective may include but is not limited to:* connecting to a database

4.2. Understand databound controls.
    *This objective may include but is not limited to:* how data is bound to controls; how to display the data in the appropriate manner; forms and WPF binding; validating databound items

5. Deploying a Windows Application
5.1. Understand Windows application deployment methods.

This objective may include but is not limited to: different methods of deploying Windows applications; choosing the appropriate method for deployment; deploying an application by using ClickOnce.

5.2. Create Windows setup and deployment projects.

This objective may include but is not limited to: creating setup projects for applications; specifying custom actions; creating special folders; security requirements; x64 deployment and program files location.
Exam Design

The Basics
This is a Microsoft Technology Associate (MTA) exam designed to assess candidates’ fundamental Web development knowledge and skills. MTA is a new certification under the Microsoft Certification Program that validates the foundational knowledge needed to begin building a career in Microsoft technologies. It can also serve as a stepping stone to the Microsoft Certified Technology Specialist exams. Successful candidates for this exam will earn an MTA certification as well as access to benefits of the Microsoft Certification Program. The primary target audience for the MTA certification is students attending high schools and two-year colleges.

We are specifying an item pool of 75 unique multiple-choice items, which will be used on 1 form. Approximately 70 percent of the items should be written to the knowledge/comprehension level and about 30 percent to the application level. For more information about cognitive levels, refer to the Cognitive Domain in Bloom’s Taxonomy.

Categories in the cognitive domain of Bloom’s Taxonomy (Anderson & Krathwohl, 2001)

The following anatomy is required of each knowledge-level item in this exam:
- Question Statement
- Answer Choices (Preferably Multiple Choice, choose 1, with 3 distracters)
  - Ex: Which of the following is a valid ASP.NET variable name? A. _foo; B. &foo; C. foo#; D. foo 1

The following anatomy is required of each application-level item in this exam:
• Concise scenario, including any constraints/requirements necessary to make distracter answers 100% incorrect
• Goal Statement (You need to ...)
• Question Statement (ex: What should you do?)
• Answer Choices (Preferably Multiple Choice, choose 1, with 3 distracters)

Target Audience

Candidates for this exam are seeking to prove knowledge of and skills in creating Web-based applications by using Microsoft® Visual Studio® 2008, the Microsoft .NET Framework 3.5, and managed code. Before taking this exam, candidates should have a solid foundational knowledge of the topics outlined in this preparation guide. It is recommended that candidates be familiar with the concepts of and have hands-on experience with the technologies described here by following a prescribed curriculum that maps to the exam or by working with tutorials and samples available on MSDN® and in Visual Studio. Candidates are expected to have some experience with a .NET language, such as C# or Microsoft Visual Basic®.NET. Candidates should also have a basic understanding of HTML syntax and usage.

Candidates for this exam are in the process of expanding their knowledge and job-related skills in the following areas:

• Web-based application development fundamentals
• Creating Microsoft ASP.NET applications by using server-side and client-side coding techniques and tools
• Web application event model
• Web services and communications with services
• Accessing and display data in a Web application
• Deploying and host Web applications by using Internet Information Services (IIS)
• Configuration options for ASP.NET applications

NOTE TO ITEM WRITERS: Items should be version agnostic if at all possible. In some cases, you may need to specify a version in the stem, but try to avoid this because more than one version of a product is available via DreamSpark.

Objective Domain

1. Programming Web Applications

1.1. Customize the layout and appearance of a Web page.

This objective may include but is not limited to: HTML, CSS, tables, embedding images, page layout for navigation
1.2. Understand ASP.NET intrinsic objects.

*This objective may include but is not limited to:* Request, Server, Application, Session, Response, HttpContext

1.3. Understand state information in Web applications.

*This objective may include but is not limited to:* how state is stored based on application design and hardware; different types such as session state, view state, control state, and application state

1.4. Understand events and control page flow.

*This objective may include but is not limited to:* application and page life cycle events; page events; control events; application events; session events; cross-page posting; Response.Redirect; Server.Transfer; IsPostBack; setting AutoEventWireup

1.5. Understand controls.

*This objective may include but is not limited to:* various types of controls, including user, server, Web, and validation; know which is the appropriate type of control for a scenario

1.6. Understand configuration files.

*This objective may include but is not limited to:* use of web.config and machine.config and the settings that can be made

2. **Working with Data and Services**

2.1. Read and write XML data.

*This objective may include but is not limited to:* XML, XML validation

*This objective does not include:* Web Services, XPath syntax, XmlDocument, XPathNavigator, XPathNodeIterator, XPathDocument, XmlReader, XmlWriter, XmlDocument, XmlNamespaceManager

2.2. Distinguish between DataSet objects and DataReader objects.

*This objective may include but is not limited to:* choose which data object to use based on application requirements/design

2.3. Call a service from a Web page.
This objective may include but is not limited to: creating a basic Windows Communication Foundation (WCF) Service or Web Service so that it can be consumed; App_WebReferences; <system.serviceModel> configuration

2.4. Understand DataSource controls.

This objective may include but is not limited to: LinqDataSource, ObjectDataSource, XmlDataSource, SqlDataSource

2.5. Bind controls to data by using data-binding syntax.

This objective may include but is not limited to: ensure that data is updated and displayed in data-aware controls

2.6. Manage data connections and databases.

This objective may include but is not limited to: database connections; connection objects; connection pools; transaction objects

3. Troubleshooting and Debugging Web Applications

3.1. Debug a Web application.

This objective may include but is not limited to: use in conjunction with custom error pages to display appropriate error information to the appropriate user; implementing tracing of a Web application, Trace.axd, Trace=True on @Page directive, <trace enabled="true"/>

3.2. Handle Web application errors.

This objective may include but is not limited to: HTTP error codes

4. Working with Client-Side Scripting

4.1. Understand client-side scripting.

This objective may include but is not limited to: purpose of client-side scripting, various client-side scripting languages

4.2. Understand AJAX concepts.

This objective may include but is not limited to: ASP.NET AJAX implementation, working with client-side libraries, EnablePartialRendering, Triggers, ChildrenAsTriggers, Scripts, Services, UpdateProgress, Timer, ScriptManagerProxy, extender controls

5. Configuring and Deploying Web Applications

5.1. Configure authentication and authorization.
This objective may include but is not limited to: Forms Authentication, Windows Authentication; authorization; file authorization; impersonation

This objective does not include: Windows CardSpace™ authentication, Passport (Windows Live™ ID) authentication, Custom authentication

5.2. Configure projects and solutions and reference assemblies.

This objective may include but is not limited to: local assemblies, shared assemblies (GAC), Web application projects and solutions; configuration files; AppSettings

5.3. Publish Web applications.

This objective may include but is not limited to: choosing the method to deploy an application based on the existing or intended environment; updatable vs. not updatable; MSI deployment; IIS installation and configuration

5.4. Understand application pools.

This objective may include but is not limited to: purpose of application pools; effect of application pools on Web applications

This objective does not include: configuring or assigning application pools
Exam Design

The Basics
This is a Microsoft Technology Associate (MTA) exam designed to assess candidates’ fundamental knowledge of database administration. MTA is a new certification under the Microsoft Certification Program that validates the foundational knowledge needed to begin building a career in Microsoft technologies. It can also serve as a stepping stone to the Microsoft Certified Technology Specialist exams. Successful candidates for this exam will earn an MTA certification as well as access to benefits of the Microsoft Certification Program. The primary target audience for the MTA certification is students attending high schools and two-year colleges.

We are specifying an item pool of 75 unique multiple-choice items, which will be used on 1 form. Approximately 70 percent of the items should be written to the knowledge/comprehension level and about 30 percent to the application level. For more information about cognitive levels, refer to the Cognitive Domain in Bloom’s Taxonomy.

Categories in the cognitive domain of Bloom’s Taxonomy (Anderson & Krathwohl, 2001)

The following anatomy is required of each knowledge-level item in this exam:

- Question Statement (ex: What should you do?)
- Answer Choices (Preferably Multiple Choice, choose 1, with 3 distracters)
  - Ex: Which of the following is a valid ASP.NET variable name? A. _foo; B. &foo; C. foo#; D. foo 1

The following anatomy is required of each application-level item in this exam:

- Concise scenario, including any constraints/requirements necessary to make distracter answers 100% incorrect
• Goal Statement (You need to ...)
• Question Statement (ex: What should you do?)
• Answer Choices (Preferably Multiple Choice, choose 1, with 3 distracters)

Target Audience

Candidates for this exam are seeking to prove introductory knowledge of and skills with databases, including relational databases such as Microsoft® SQL Server®. It is recommended that candidates be familiar with the concepts of and have hands-on experience with the technologies described here either by taking relevant training courses or by working with tutorials and samples available on MSDN® and in Microsoft Visual Studio®. Although minimal hands-on experience with the technologies is recommended, job experience is not assumed for these exams.

Candidates for this exam are in the process of expanding their knowledge and skills in the following areas:

- core database concepts
- relational database concepts
- security requirements for databases and the data stored in them
- database objects, such as tables and views
- graphical tools and T-SQL scripts
- database queries
- stored procedures

Objective Domain

1. Understanding Core Database Concepts

1.1. Understand how data is stored in tables.

   *This objective may include but is not limited to:* understanding what a table is and how it relates to the data that will be stored in the database; columns/fields, rows/records

1.2. Understand relational database concepts.

   *This objective may include but is not limited to:* understanding what a relational database is, the need for relational database management systems (RDBMS), and how relations are established

1.3. Understand data manipulation language (DML).

   *This objective may include but is not limited to:* understanding what DML is and its role in databases

1.4. Understand data definition language (DDL).
This objective may include but is not limited to: understanding how T-SQL can be used to create database objects such as tables and views

2. **Creating Database Objects**
   2.1. Choose data types.
   This objective may include but is not limited to: understanding what data types are, why they are important, and how they affect storage requirements

   2.2. Understand tables and how to create them.
   This objective may include but is not limited to: purpose of tables; creating tables in a database by using proper ANSI SQL syntax

   2.3. Create views.
   This objective may include but is not limited to: understanding when to use views and how to create a view by using T-SQL or a graphical designer.

   2.4. Create stored procedures and functions.
   This objective may include but is not limited to: selecting, inserting, updating, or deleting data

3. **Manipulating Data**
   3.1. Select data.
   This objective may include but is not limited to: utilizing SELECT queries to extract data from one table; extracting data by using joins; combining result sets by using UNION and INTERSECT

   3.2. Insert data.
   This objective may include but is not limited to: understanding how data is inserted into a database; how to use INSERT statements

   3.3. Update data.
   This objective may include but is not limited to: understanding how data is updated in a database and how to write the update data to the database by using the appropriate UPDATE statements; update by using a table

   3.4. Delete data.
   This objective may include but is not limited to: deleting data from single or multiple tables; ensuring data and referential integrity by using transactions

4. **Understanding Data Storage**
   4.1. Understand normalization.
This objective may include but is not limited to: understanding the reasons for normalization, the five most common levels of normalization, how to normalize a database to third normal form.

4.2. Understand primary, foreign, and composite keys.
This objective may include but is not limited to: understanding the reason for keys in a database, choosing appropriate primary keys, selecting appropriate data type for keys, selecting appropriate fields for composite keys, understanding the relationship between foreign and primary keys.

4.3. Understand indexes.
This objective may include but is not limited to: understanding clustered and non-clustered indexes and their purpose in a database.

5. Administering a Database

5.1. Understand database security concepts.
This objective may include but is not limited to: understanding the need to secure a database, what objects can be secured, what objects should be secured, user accounts, and roles.

5.2. Understand database backups and restore.
This objective may include but is not limited to: understanding various backup types, such as full and incremental, importance of backups, how to restore a database.
Exam Design

The Basics
This exam is designed to assess candidates’ knowledge of fundamental Windows Server administration concepts. MTA is a new certification under the Microsoft Certification Program that validates the foundational knowledge needed to begin building a career in Microsoft technologies. It can also serve as a stepping stone to the Microsoft Certified Technology Specialist exams. Successful candidates for this exam will earn an MTA certification as well as access to benefits of the Microsoft Certification Program. The primary target audience for the MTA certification is students attending high schools and two-year colleges.

The following anatomy is required of each knowledge-level item in this exam:
- Question Statement (ex: What should you do?)
- Answer Choices (Preferably Multiple Choice, choose 1, with 3 distracters)
  - Ex: Which of the following is a valid ASP.NET variable name? A. _foo; B. &foo; C. foo#; D. foo 1

The following anatomy is required of each application-level item in this exam:
- Concise scenario, including any constraints/requirements necessary to make distracter answers 100% incorrect
- Goal Statement (You need to ...)
- Question Statement (ex: What should you do?)
- Answer Choices (Preferably Multiple Choice, choose 1, with 3 distracters)

Target Audience
Candidates for this exam are seeking to prove Windows Server administration knowledge and skills. Before taking this exam, candidates should have a solid foundational knowledge of the topics outlined in this preparation guide. It is recommended that candidates become familiar with the concepts and the technologies described here by taking relevant training courses. Candidates are expected to have some hands-on experience with Windows Server, Windows-based networking, Active Directory, account management, and system recovery tools and concepts.
Objective Domain

1. Understanding Server Installation
   1.1. Understand device drivers.
   This objective may include but is not limited to: installation; removal; disabling; update/upgrade; rollback; troubleshooting; Plug and Play (PnP); IRQ; interrupts; driver signing
   1.2. Understand services.
   This objective may include but is not limited to: what services are; which statuses a service can be in; startup types; recovery options; delayed startup; Run As settings for a service; stopping or pausing a service; service accounts, dependencies
   1.3. Understand server installation options.
   This objective may include but is not limited to: choosing correct OS version; partitioning; F8 options; server core vs. full; interactive install; unattended install; automated install using Windows Deployment Service (WDS); upgrade vs. clean install; firmware updates including BIOS

2. Understanding Server Roles
   2.1. Identify application servers.
   This objective may include but is not limited to: mail servers; database servers; collaboration servers; monitoring servers; threat management
   2.2. Understand Web services.
   This objective may include but is not limited to: IIS, WWW, and FTP; separate worker processes; adding components; sites; ports; SSL; certificates
   2.3. Understand remote access.
   This objective may include but is not limited to: remote assistance; remote administration tools; Remote Desktop Services; licensing; RD Gateway; VPN; application virtualization; multiple ports
   2.4. Understand file and print services.
   This objective may include but is not limited to: local printers; network printers; printer pools; Web printing; Web management; driver deployment; file, folder, and share permissions vs. rights; auditing; print job management
   2.5. Understand server virtualization.
   This objective may include but is not limited to: virtualization modes; VHDs; virtual memory; virtual networks; snapshots and saved states; physical to virtual; virtual to physical

3. Understanding Active Directory
   3.1. Understand accounts and groups.
   This objective may include but is not limited to: domain accounts; local accounts; user profiles; group types; group scopes; group nesting
   3.2. Understand organizational units (OUs) and containers.
   This objective may include but is not limited to: purpose of OUs; purpose of containers; delegation; default
3.3. Understand Active Directory infrastructure.
This objective may include but is not limited to: domain controllers; forests; operation masters roles; domain vs. workgroup; child domains; trusts; functional levels; namespace; sites; replication
3.4. Understand group policy.
This objective may include but is not limited to: group policy processing; Group Policy Management Console; computer policies; user policies; local policies

4. Understanding Storage
4.1. Identify storage technologies.
This objective may include but is not limited to: advantages and disadvantages of different storage types; local (SATA, SCSI, IDE); NAS; SAN; fibre channel; iSCSI; NFS; FC HBA and FC switches; iSCSI hardware
4.2. Understand RAID.
This objective may include but is not limited to: RAID 0, RAID 1, RAID 5, RAID 10 and combinations; hardware and software RAID
4.3. Understand disk types.
This objective may include but is not limited to: basic disk; dynamic disk; mount points; file systems; mounting a virtual hard disk; distributed file systems; optical disks

5. Understanding Server Performance Management
5.1. Identify major server hardware components.
This objective may include but is not limited to: memory; disk; processor; network; 32 and 64 bits; removable drives; graphic cards; cooling; power usage; ports
5.2. Understand performance monitoring.
This objective may include but is not limited to: methodology; procedures; creating a baseline; perfmon; Resource Monitor; Task Manager; performance counters
5.3. Understand logs and alerts.
This objective may include but is not limited to: purpose of performance logs and alerts

6. Understanding Server Maintenance
6.1. Identify steps in the startup process.
This objective may include but is not limited to: BIOS; bootsector; bootloader; MBR; boot.ini; bcdedit; POST; Safe Mode
6.2. Understand business continuity.
This objective may include but is not limited to: backup and restore; disaster recovery; clustering; Active Directory restore; folder redirection; data redundancy; uninterruptible power supply (UPS)
6.3. Understand updates.
This objective may include but is not limited to: software; driver; operating systems; applications; Windows Update; Windows Server Update Services (WSUS)
6.4. Understand troubleshooting methodology.
This objective may include but is not limited to: processes; procedures; best practices; systematic vs. specific approach; perfmon; Event Viewer; Resource Monitor; Information Technology Infrastructure Library; central logging; event filtering; default logs
Exam Design

The Basics
This exam is designed to assess candidates’ knowledge of fundamental networking concepts. MTA is a new certification under the Microsoft Certification Program that validates the foundational knowledge needed to begin building a career in Microsoft technologies. It can also serve as a stepping stone to the Microsoft Certified Technology Specialist exams. Successful candidates for this exam will earn an MTA certification as well as access to benefits of the Microsoft Certification Program. The primary target audience for the MTA certification is students attending high schools and two-year colleges.

We are specifying an item pool of 75 unique multiple-choice items, which will be used on 1 form. For more information about cognitive levels, refer to the Cognitive Domain in Bloom’s Taxonomy.

Categories in the cognitive domain of Bloom's Taxonomy (Anderson & Krathwohl, 2001)

The following anatomy is required of each knowledge-level item in this exam:
- Question Statement (ex: What should you do?)
- Answer Choices (Preferably Multiple Choice, choose 1, with 3 distracters)
  - Ex: Which of the following is a valid ASP.NET variable name? A. _foo; B. &foo; C. foo#; D. foo 1

The following anatomy is required of each application-level item in this exam:
- Concise scenario, including any constraints/requirements necessary to make distracter answers 100% incorrect
- Goal Statement (You need to ...)
- Question Statement (ex: What should you do?)
Answer Choices (Preferably Multiple Choice, choose 1, with 3 distracters)

Target Audience

Candidates for this exam are seeking to prove fundamental networking knowledge and skills. Before taking this exam, candidates should have a solid foundational knowledge of the topics outlined in this preparation guide. It is recommended that candidates become familiar with the concepts and the technologies described here by taking relevant training courses. Candidates are expected to have some hands-on experience with Windows Server, Windows based networking, network management tools, DNS, TCP/IP, names resolution process, and network protocols and topologies.

About This Exam

Objective Domain

1. Understanding Network Infrastructures
   1.1. Understand the concepts of the Internet, intranet, and extranet.
       This objective may include but is not limited to: VPN, security zones, firewalls
   1.2. Understand local area networks (LANs).
       This objective may include but is not limited to: perimeter networks; addressing; reserved address ranges for local use (including local loopback ip), VLANs; wired LAN and wireless LAN
   1.3. Understand wide area networks (WANs).
       This objective may include but is not limited to: leased lines, dial-up, ISDN, VPN, T1, T3, E1, E3, DSL, and cable and their characteristics (speed, availability)
   1.4. Understand wireless networking.
       This objective may include but is not limited to: types of wireless networking standards and their characteristics (802.11A, B, G, N including different Ghz ranges), types of network security (for example, WPA/WEP/802.1X ), point-to-point (P2P) wireless, wireless bridging
   1.5. Understand network topologies and access methods.
       This objective may include but is not limited to: star, mesh, ring

2. Understanding Network Hardware
   2.1. Understand switches.
       This objective may include but is not limited to: transmission speed; number and type of ports; number of uplinks; speed of uplinks; managed or unmanaged switches; VLAN capabilities; Layer 2 and Layer 3 switches, security options; hardware redundancy; support; backplane speed; switching types, MAC table; understanding capabilities of hubs vs. switches
   2.2. Understand routers.
       This objective may include but is not limited to: transmission speed considerations, directly connected routes, static routing, dynamic routing (routing protocols), default routes; routing
table and how it selects best routes; routing table memory, NAT, software routing in Windows Server

2.3. Understand media types.
This objective may include but is not limited to: cable types and their characteristics, including media segment length and speed; fibre optic; twisted pair shielded or non-shielded; cabling, wireless; susceptibility to external interference (for example, machinery, power cables); susceptibility to electricity (for example, lightning), susceptibility to interception

3. Understanding Protocols and Services

3.1. Understand the OSI model.
This objective may include but is not limited to: OSI model; TCP model; examples of devices, protocols, and applications and which OSI/TCP layer they belong to; TCP and UDP; well-known ports for most-used purposes (not necessarily Internet); packets and frames

3.2. Understand IPv4.
This objective may include but is not limited to: addressing, subnetting; NAT, static IP, gateway; APIPA; network classes, classful/classless IP addressing; reserved address ranges for local use (including local loopback ip)

This objective may include but is not limited to: subnetting; IPconfig; why use IPv6; addressing; ipv4toipv6 tunneling protocols to ensure backwards compatibility; dual ip stack; subnetmask; gateway; ports; packets; reserved address ranges for local use (including local loopback ip)

3.4. Understand names resolution.
This objective may include but is not limited to: DNS, WINS, steps in the name resolution process

3.5. Understand networking services.
This objective may include but is not limited to: DHCP, IPsec, remote access

3.6. Understand TCP/IP.
This objective may include but is not limited to: tools such as ping; tracert; pathping; Telnet; IPconfig; netstat, reserved address ranges for local use (including local loopback ip); protocols
Exam Design

The Basics
This exam is designed to assess candidates’ knowledge of fundamental security concepts. MTA is a new certification under the Microsoft Certification Program that validates the foundational knowledge needed to begin building a career in Microsoft technologies. It can also serve as a stepping stone to the Microsoft Certified Technology Specialist exams. Successful candidates for this exam will earn an MTA certification as well as access to benefits of the Microsoft Certification Program. The primary target audience for the MTA certification is students attending high schools and two-year colleges.

The following anatomy is required of each knowledge-level item in this exam:
- Question Statement (ex: What should you do?)
- Answer Choices (Preferably Multiple Choice, choose 1, with 3 distracters)
  - Ex: Which of the following is a valid ASP.NET variable name? A. _foo; B. &foo; C. foo#; D. foo 1

The following anatomy is required of each application-level item in this exam:
- Concise scenario, including any constraints/requirements necessary to make distracter answers 100% incorrect
- Goal Statement (You need to ...)
- Question Statement (ex: What should you do?)
- Answer Choices (Preferably Multiple Choice, choose 1, with 3 distracters)

Target Audience

Candidates for this exam are seeking to prove fundamental security knowledge and skills. Before taking this exam, candidates should have a solid foundational knowledge of the topics outlined in this preparation guide. It is recommended that candidates become familiar with the concepts and the technologies described here by taking relevant training courses. Candidates are expected to have some hands-on experience with Windows Server, Windows based networking, Active Directory, Anti-Malware products, firewalls, network topologies and devices, and network ports.
Objective Domain

1. **Understanding Security Layers**
   1.1. Understand core security principles.
   This objective may include but is not limited to: confidentiality; integrity; availability; how threat and risk impact principles; principle of least privilege; social engineering; attack surface
   1.2. Understand physical security.
   This objective may include but is not limited to: site security; computer security; removeable devices and drives; access control; mobile device security; disable Log On Locally; keyloggers
   1.3. Understand Internet security.
   This objective may include but is not limited to: browser settings; zones; secure Web sites
   1.4. Understand wireless security.
   This objective may include but is not limited to: advantages and disadvantages of specific security types; keys; SSID; MAC filters

2. **Understanding Operating System Security**
   2.1. Understand user authentication.
   This objective may include but is not limited to: multifactor; smart cards; RADIUS; Public Key Infrastructure (PKI); understand the certificate chain; biometrics; Kerberos and time skew; using Run As to perform administrative tasks; password reset procedures
   2.2. Understand permissions.
   This objective may include but is not limited to: file; share; registry; Active Directory; NTFS vs. FAT; enabling or disabling inheritance; behavior when moving or copying files within the same disk or onto another disk; multiple groups with different permissions; basic permissions and advanced permissions; take ownership; delegation
   2.3. Understand password policies.
   This objective may include but is not limited to: password complexity; account lockout; password length; password history; time between password changes; enforce by using group policies; common attack methods
   2.4. Understand audit policies.
   This objective may include but is not limited to: types of auditing; what can be audited; enabling auditing; what to audit for specific purposes; where to save audit information; how to secure audit information
   2.5. Understand encryption.
   This objective may include but is not limited to: EFS; how EFS-encrypted folders impact moving and copying files; BitLocker (To Go); Trusted Platform Module (TPM); software-based encryption; MAIL encryption and signing and other uses; VPN; public key and private key; encryption algorithms; certificate properties; certificate services; PKI/certificate services infrastructure; token devices
   2.6. Understand malware.
   This objective may include but is not limited to: buffer overflow; worms; Trojans; spyware
3. **Understanding Network Security**
   
   3.1. Understand dedicated firewalls.
   This objective may include but is not limited to: types of hardware firewalls and their characteristics; when to use a hardware firewall instead of a software firewall; stateful vs. stateless inspection
   
   3.2. Understand Network Access Protection (NAP).
   This objective may include but is not limited to: purpose of NAP; requirements for NAP
   
   3.3. Understand network isolation.
   This objective may include but is not limited to: VLANs; routing; honeypot; DMZ; NAT; VPN; IPsec; Server and Domain Isolation
   
   3.4. Understand protocol security.
   This objective may include but is not limited to: protocol spoofing; IPsec; tunneling; DNSsec; network sniffing; common attack methods

4. **Understanding Security Software**
   
   4.1. Understand client protection.
   This objective may include but is not limited to: anti-virus; User Account Control (UAC); keeping client operating system and software updated; encrypting offline folders; software restriction policies
   
   4.2. Understand e-mail protection.
   This objective may include but is not limited to: anti-spam; anti-virus; spoofing, phishing, and pharming; client vs. server protection; SPF records; PTR records
   
   4.3. Understand server protection.
   This objective may include but is not limited to: separation of services; hardening; keeping server updated; secure dynamic DNS updates; disabling unsecure authentication protocols; Read-Only Domain Controllers; separate management VLAN; Microsoft Baseline Security Analyzer (MBSA)
Audience

Microsoft Technology Associate

The Microsoft Technology Associate (MTA) is a new Microsoft Certification program that validates the foundational knowledge needed to begin building a career by using Microsoft technologies.

Successful candidates earn MTA certificates as well as access to benefits on the Microsoft Certification member site.

This program:

- is targeted primarily at students attending high schools or two-year colleges.
- provides an appropriate entry point to a future career in technology.
- assumes some hands-on experience or training but does not assume on-the-job experience.

Target Audience

Candidates for this exam are seeking to prove knowledge and skills on Microsoft .NET fundamentals. Before taking this exam, candidates should have a solid foundational knowledge of the topics outlined in this preparation guide. It is recommended that candidates be familiar with the concepts of and have hands-on experience with the technologies described here either by taking relevant training courses or by working with tutorials and samples available on MSDN and in Microsoft Visual Studio. Candidates are expected to have some experience with a .NET language such as C# or Microsoft Visual Basic .NET.
Candidates for this exam are in the process of expanding their knowledge and skills in the following areas:

- .NET namespace and class organizations
- core .NET knowledge
- managed code theory
- memory management in .NET
- language parity

**Objective Domain**

1. **Understanding .NET Framework Concepts**
   1.1. Understand basic application settings.
       This objective may include but is not limited to: understanding app.config and web.config
   1.2. Understand events and event handling in the .NET Framework.
       This objective may include but is not limited to: understanding the event-driven programming model and event handlers, raising events, and implementing delegates
   1.3. Understand structured exception handling in the .NET Framework.
       This objective may include but is not limited to: understanding error handling concepts, exceptions, and exception types

2. **Understanding Namespaces and Classes in the .NET Framework**
   2.1. Understand .NET class hierarchies.
       This objective may include but is not limited to: understanding system classes, classifications of classes, and logical organization of classes
   2.2. Understand Object Oriented Concepts in the .NET Framework.
       This objective may include but is not limited to: understanding how inheritance works in .NET, polymorphism, and interfaces
   2.3. Understand .NET namespaces.
       This objective may include but is not limited to: understanding the reason for namespaces, the organization of namespaces in the class library, and how to use namespaces in an application
   2.4. Understand and create class libraries.
       This objective may include but is not limited to: understanding the logical grouping of classes and the logic behind class libraries (what they are, why they are important)
2.5. Understand and use different data types in the .NET Framework.
   This objective may include but is not limited to: understanding intrinsic data types, values
types, reference types, boxing, unboxing, and .NET collection classes

2.6. Understand generics.
   This objective may include but is not limited to: understanding generics infrastructure,
generic interfaces, generic delegates, contravariant and covariant generic type arguments,
generic methods, verifiability, and constraints

3. Understanding .NET Code Compilation
   3.1. Understand the fundamentals of Microsoft Intermediate Language (MSIL) and Common
        Language Infrastructure (CLI).
        This objective may include but is not limited to: understanding what MSIL is, what the CLI is,
        how source code is compiled into MSIL, and how code is Just-in-Time (JIT) compiled

   3.2. Understand the use of strong naming.
        This objective may include but is not limited to: understanding why strong naming is used,
        how to sign assemblies to support strong naming, and Global Assembly Cache (GAC)

   3.3. Understand version control.
        This objective may include but is not limited to: understanding how .NET applications are
        versioned and how to run different versions on the same computer

   3.4. Understand assemblies and metadata.
        This objective may include but is not limited to: understanding .NET assemblies, shared
        assemblies, and metadata in .NET

4. Understanding I/O Classes in the .NET Framework
   4.1. Understand .NET file classes.
        This objective may include but is not limited to: understanding read/write file classes and
        stream readers and writers

   4.2. Understand console I/O.
        This objective may include but is not limited to: understanding System.Console classes for
        input and output

   4.3. Understand XML classes in the .NET Framework.
        This objective may include but is not limited to: understanding XMLReader, XMLWriter, and
        XML Schemas

5. Understanding Security
   5.1. Understand the System Security namespace.
        This objective may include but is not limited to: understanding permissions and what
        cryptography is

   5.2. Understand authentication and authorization.
        This objective may include but is not limited to: understanding code access security, access
control, and policies

6. Understanding .NET Languages
   6.1. Understand language interoperability.
       This objective may include but is not limited to: calling code written in one language from another language, understanding .NET language parity
   6.2. Understand type safety.
       This objective may include but is not limited to: understanding memory type safety, type safety in classes, strong types, and security policies

7. Understanding Memory Management
   7.1. Understand resource allocation.
       This objective may include but is not limited to: understanding garbage collection and memory allocation, understanding stack versus heap
   7.2. Understand the difference between managed and unmanaged applications.
       This objective may include but is not limited to: understanding why managed code is called managed code, understanding the differences between coding in managed versus unmanaged code
98-373
MTA: Mobile Development Fundamentals

About this Exam

The Microsoft Technology Associate (MTA) is a new Microsoft Certification program that validates the foundational knowledge needed to begin building a career using Microsoft technologies.

Successful candidates earn MTA certificates as well as access to benefits on the Microsoft Certification member site.

This program:

- is targeted primarily at students who attend high schools and two-year colleges.
- provides an appropriate entry point to a future career in technology.
- assumes some hands-on experience or training but does not assume on-the-job experience.

This exam is designed to provide candidates with an assessment of their knowledge of fundamental mobile development concepts. It can also serve as a stepping stone to the Microsoft Certified Technology Specialist exams.

Audience Profile

Candidates for this exam are seeking to prove core mobile development skills. Before taking this exam, candidates should have solid foundational knowledge of the topics outlined in this preparation guide, including Silverlight, HTML5, and other phone operating system tools. It is recommended that candidates be familiar with the concepts of and have hands-on experience with the technologies described here either by taking relevant training courses or by working with tutorials and samples available on MSDN and in Microsoft Visual Studio.
Objective Domain

1. Work with Physical Devices
   1.1. Understand mobile device tools.
       This objective may include but is not limited to: defining the Windows Phone Capability Detection Tool and the Windows Phone Connect tool; Windows Phone Marketplace Test Kit
   1.2. Understand physical capabilities of the mobile device.
       This objective may include but is not limited to: identifying the different device sensors; describing and defining the camera capture and preview stream APIs; identifying different built-in hardware; Motion API
   1.3. Plan for physical interactions with the mobile device.
       This objective may include but is not limited to: describing and defining the differences among devices, including features, API levels, number of touch points, and networking capabilities; identifying ways to save energy; accounting for screen size/real estate when planning layout

2. Use Data with Mobile Devices
   2.1. Work with networked data.
       This objective may include but is not limited to: integrating with databases (Microsoft SQL Server, SQL Lite); describing and defining how LINQ and Microsoft ADO.NET work; implementing data binding; minimizing the data traffic for performance and cost; making use of Windows Communication Foundation (WCF) Web services and REST; describing and defining the benefits of SQL Server replication
   2.2. Use data stores.
       This objective may include but is not limited to: using different kinds of storage (for example, file and database); describing and defining the benefits of different storage locations (local, isolated, remote); integration with XML; accessing native data and functionalities (launchers, choosers); handling offline situations

3. Use a Mobile Application Development Environment
3.1. Understand design for mobile devices.

This objective may include but is not limited to: describing and defining marketplace submission rules; describing and defining mobile design concepts (for example, metro, button sizing, spacing); describing and defining globalization/localization; defining mobile optimization; defining MVVM; describing and defining object-oriented programming (OOP) and separation of concerns; describing and defining asynchronous programming/threading.

3.2. Network for mobile devices.

This objective may include but is not limited to: describing and defining the application model in relation to WCF RIA services; creating a robust server/cloud communication that can throttle between no network to mobile network to wireless network; describing and defining networking concepts in relation to multicast and HTTP requests; using Web services; describing and defining toast and other notifications.

3.3. Understand Silverlight.

This objective may include but is not limited to: describing and defining the differences between Silverlight, XNA, and HTML5 and which one to choose for a given scenario; using Silverlight and HTML5 applications; identifying Silverlight controls.

3.4. Work with developer tools.

This objective may include but is not limited to: using Microsoft Visual Studio IDE; creating the deployment package and deploying the application; using the Microsoft .NET Framework; configuring a test environment; testing and debugging mobile applications.

3.5. Code for mobile applications.

This objective may include but is not limited to: evaluating code; identifying code errors; identifying the code to use to meet requirements, distinguishing among programming languages and programs, including XNA, Microsoft Visual Basic .NET, HTML5, XAML, and C# .NET.

4. Develop Mobile Applications

4.1. Manage the application life cycle.

This objective may include but is not limited to: preserving application state information and handling activate/deactivate functions; using tombstoning.

The information contained in this document is proprietary and is not to be distributed or disclosed unless explicitly authorized by Microsoft Corporation.
balancing code between battery usage and performance; creating a responsive application with feedback of user actions; managing visible status for long-running operations; storing passwords; splash screen

4.2. Understand mobile device APIs.
This objective may include but is not limited to: NavigationServices class, mapping/GeoLocation APIs, and Forms, Canvas, and Media APIs in HTML5; describing and defining manipulation events, including ManipulationStarted and ManipulationDelta

4.3. Understand mobile device controls.
This objective may include but is not limited to: using Windows Phone controls; arranging content with panels; displaying collections of items; building custom controls; describing and defining Push/Raw/Tile notification; using tasks and choosers to enhance application functionality

4.4. Build the user interface.
This objective may include but is not limited to: creating layout with Style; designing with system theme, accent color, and screen orientation; graphic layering (transparency, borders, resizing); creating the user experience to be clean, focused, and using UI standards and guidelines; integrating images and media in an application
About This Exam

The Microsoft Technology Associate (MTA) is a new Microsoft Certification program that validates the foundational knowledge needed to begin building a career using Microsoft technologies.

Successful candidates earn MTA certificates as well as access to benefits on the Microsoft Certification member site.

This program:

- is targeted primarily at students who attend high schools and two-year colleges.
- provides an appropriate entry point to a future career in technology.
- assumes some hands-on experience or training but does not assume on-the-job experience.

This exam is designed to provide candidates with an assessment of their knowledge of fundamental gaming development concepts. It can also serve as a stepping stone to the Microsoft Certified Technology Specialist exams.

Audience Profile

Candidates for this exam are seeking to prove core gaming development skills. Before taking this exam, candidates should have solid foundational knowledge of game design, hardware, graphics, and animation. It is recommended that candidates be familiar with the concepts of and have some hands-on experience with the technologies described here either by taking relevant training courses or working with tutorials and samples available on MSDN and in Microsoft Visual Studio.
Objective Domain

1. Understand Game Design

1.1. Differentiate among game types.
This objective may include but is not limited to: console, Xbox, MMORPG, mobile games, PC games

1.2. Differentiate among game genres.
This objective may include but is not limited to: fantasy, sports, role playing, card, board, First Person Shooter

1.3. Understand player motivation.
This objective may include but is not limited to: quests, tasks, activities, how to win, game goals

1.4. Design the user interface.
This objective may include but is not limited to: UI layout and concepts, asset management, game state, gamer services

1.5. Understand components.
This objective may include but is not limited to: differentiate between tool creation and game programming, understand artificial intelligence (AI)

1.6. Capture user data.
This objective may include but is not limited to: save and restore user data, save and restore game state, handle input states, store data, manage game state; input services

1.7. Work with XNA.
This objective may include but is not limited to: understanding the architecture of an XNA game; using built-in XNA tools; work with XNA hierarchy (initialization, update loop, drawing)

2. Understand Hardware

2.1. Choose an input device.
This objective may include but is not limited to: mouse, keyboard, Kinect, console, mobile

2.2. Choose an output device.
This objective may include but is not limited to: screen, television, hand-held devices, sound (local speakers, surround sound systems)

2.3. Work with the network.
This objective may include but is not limited to: set up Web services, TCP, UDP, basic management; plan for areas without access to Internet; notifications

2.4. Manage game performance.
This objective may include but is not limited to: CPU vs. GPU, reach vs. HiDef, graphics networking performance; frame rate

2.5. Understand the different game platforms.
This objective may include but is not limited to: console, PC, mobile; compare memory management

3. Understand Graphics

3.1. Understand rendering engines.
This objective may include but is not limited to: DirectX, video and audio compression, display initialization, resolution (full screen, Vsync, and windowed); transforms

3.2. Plan for game state.
This objective may include but is not limited to: scene hierarchy engine, gametime to handle frame rate variations, understanding games’ main loop (input/update/render), graphics pipeline; understanding the flow of a game, loading, menus, save-load, configuring options (video, audio, keyboard)

3.3. Draw objects.
This objective may include but is not limited to: using bitmaps, sprites, vector graphics, lighting, blending, text, textures, 3D geometry, parallax mapping, and different shaders; 2-D vs. 3-D; creating a sprite font

4. Understand Animation

4.1. Animate basic characters.
This objective may include but is not limited to: movement, lighting, projections, frames per second (FPS), shaders, apply filters to textures, sprite animation, generate objects from user indexed primitives, matrices, understanding keyframes, motion between keyframes

Proprietary and Confidential - © 2011 Microsoft Corporation

The information contained in this document is proprietary and is not to be distributed or disclosed unless explicitly authorized by Microsoft Corporation.
4.2. Transform objects.
   This objective may include but is not limited to: forming, deforming, moving, point distances, planes, interpolation; frames per second (FPS); translation, scale, rotation

4.3. Work with collisions.
   This objective may include but is not limited to: per pixel and rectangle collisions, collision detection, collision response, fundamentals of physics simulation
MTA Exam 98-375: HTML5 Application Development Fundamentals

About This Exam

The Microsoft Technology Associate (MTA) is a new Microsoft Certification program that validates the foundational knowledge needed to begin building a career using Microsoft technologies.

Successful candidates earn MTA certificates as well as access to benefits on the Microsoft Certification member site.

This program:

- is targeted primarily at students who attend high schools and two-year colleges.
- provides an appropriate entry point to a future career in technology.
- assumes some hands-on experience or training but does not assume on-the-job experience.

This exam is designed to provide candidates with an assessment of their knowledge of fundamental HTML5 application development concepts. It can also serve as a stepping stone to the Microsoft Certified Technology Specialist exams.

Audience Profile

Candidates for this exam are seeking to prove core HTML5 client application development skills that will run on today's touch-enabled devices (PCs, tablets, and phones). Although HTML is often thought of as a web technology that is rendered in a browser to produce a UI, this exam focuses on using HTML5, CSS3, and JavaScript to develop client applications. Before taking this exam, candidates should have solid foundational knowledge of the topics outlined in the preparation guide, including CSS and JavaScript. It is recommended that candidates be familiar with the concepts of and have some hands-on experience with the related technologies either by taking relevant training courses or by working with tutorials and samples available on MSDN and in Microsoft Visual Studio.

Objective Domain
1. Manage the Application Life Cycle
   1.1. Understand the platform fundamentals.
       This objective may include but is not limited to: packaging and the runtime environment: app package, app container, credentials/permission sets, host process, leveraging existing HTML5 skills and content for slate/tablet applications
   1.2. Manage the state of an application.
       This objective may include but is not limited to: manage session state, app state, and persist state information; understand states of an application; understand the differences between local and session storage
   1.3. Debug and test an HTML5-based touch-enabled application.
       This objective may include but is not limited to: touch gestures; understand which gestures you test on a device
   1.4. Publish an application to a store.
       This objective may include but is not limited to: understand requirements for marketplace submissions

2. Build the User Interface by Using HTML5
   2.1. Choose and configure HTML5 tags to display text content.
   2.2. Choose and configure HTML5 tags to display graphics.
       This objective may include but is not limited to: when, why, and how to use Canvas; when, why, and how to use scalable vector graphics (SVG)
   2.3. Choose and configure HTML5 tags to play media.
       This objective may include but is not limited to: video and audio tags
   2.4. Choose and configure HTML5 tags to organize content and forms.
       This objective may include but is not limited to: tables, lists, sections; semantic HTML
   2.5. Choose and configure HTML5 tags for input and validation.

Proprietary and Confidential - © 2011 Microsoft Corporation
The information contained in this document is proprietary and is not to be distributed or disclosed unless explicitly authorized by Microsoft Corporation.
3. Format the User Interface by Using CSS

3.1. Understand the core CSS concepts.

This objective may include but is not limited to: separating presentation from content – create content with HTML and style content with CSS; managing content flow - inline vs. block flow; managing positioning of individual elements – float vs. absolute positioning; managing content overflow – scrolling, visible, and hidden; basic CSS styling

3.2. Arrange user interface (UI) content by using CSS.

This objective may include but is not limited to: using flexible box and grid layouts to establish content alignment, direction, and orientation; proportional scaling and use of “free scale” for elements within a flexible box or grid; ordering and arranging content; concepts for using flex box for simple layouts and grid for complex layouts; grid content properties for rows and columns; using application templates

3.3. Manage the flow of text content by using CSS.

This objective may include but is not limited to: regions and using regions to flow text content between multiple <div> sections – content source, content container, dynamic flow, flow-into, flow-from, msRegionUpdate, msRegionOverflow, msGetRegionContent(); columns and hyphenation and using these CSS settings to optimize the readability of text; using “positioned floats” to create text flow around a floating object

3.4. Manage the graphical interface by using CSS.

This objective may include but is not limited to: graphics effects - rounded corners, shadows, transparency, background gradients, typography, and Web Open Font Format; 2D and 3D transformations – translate, scale, rotate, skew, and 3D perspective transitions and animations; SVG filter effects; Canvas

4. Code by Using JavaScript

4.1. Manage and maintain JavaScript.

This objective may include but is not limited to: creating and using functions; using Windows Library for JavaScript, jQuery, and other third-party libraries

Proprietary and Confidential - © 2011 Microsoft Corporation

The information contained in this document is proprietary and is not to be distributed or disclosed unless explicitly authorized by Microsoft Corporation.
4.2. Update the UI by using JavaScript.

This objective may include but is not limited to: locating/accessing elements; listening and responding to events; showing and hiding elements; updating the content of elements; adding elements

4.3. Code animations by using JavaScript.

This objective may include but is not limited to: using animation; manipulating the canvas; working with images, shapes, and other graphics

4.4. Access data access by using JavaScript.

This objective may include but is not limited to: sending and receiving data; transmitting complex objects and parsing; loading and saving files; App Cache; datatypes; forms; cookies; localStorage

4.5. Respond to the touch interface.

This objective may include but is not limited to: gestures, how to capture and respond to gestures

4.6. Code additional HTML5 APIs.

This objective may include but is not limited to: GeoLocation, Web Workers, WebSocket; File API

4.7. Access device and operating system resources.

This objective may include but is not limited to: Windows Runtime (WinRT); in memory resources such as contact lists and calendar; hardware capabilities such as GPS, accelerometer and camera

Proprietary and Confidential - © 2011 Microsoft Corporation

The information contained in this document is proprietary and is not to be distributed or disclosed unless explicitly authorized by Microsoft Corporation.