

## REVIEW LESSON

MTA Course: 10754 Microsoft .NET Fundamentals

Lesson name: Microsoft .NET Fundamentals 3.1

Topic: Understand the fundamentals of Microsoft Intermediate Language (MSIL) and Common Language Infrastructure (CLI) (One 50-minute class period)

File name: 10754\_Msft.NET\_RL\_3.1

### Lesson Objective

**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.

### Preparation Details

#### Prerequisite student experiences and knowledge:

This MTA Certification Exam Review lesson is written for students who have learned about application programming with the Microsoft .NET Framework. Students who do not have the prerequisite knowledge and experiences cited in the objective will find additional learning opportunities using resources such as those listed in the “Resources” section at the end of this review lesson.

#### Instructor preparation activities:

- Before starting this lesson, students must have a simple, working application that they can build and execute. It is recommended that they use a project from a previous assignment. As an alternative, they could create a simple “Hello World” application.
- Modify step 1 of the Student Activity document 10754\_Msft.NET\_SA\_3.1 to indicate the project that students will use to complete the activity.

#### Resources, software, and additional files needed for this lesson:

- 10754\_Msft.NET\_PPT\_3.1
- 10754\_Msft.NET\_SA\_3.1

- Microsoft Visual Studio® 2010; students may also use Microsoft Visual C#® 2010 Express or Microsoft Visual Basic® 2010 Express, available at <http://www.microsoft.com/express/Windows/>.

### **Teaching Guide**

#### **Essential Vocabulary**

**Just-In-Time (JIT) compiler**—a compiler that converts MSIL into native code as the application is executing.

**Microsoft Intermediate Language (MSIL)**—a CPU-independent set of instructions that can be converted efficiently to native code. MSIL is created when a developer builds a project.

**native code**—instructions designed to be executed directly by a CPU without any additional translation or conversion. Native code is specific to a CPU or architecture.

**portable executable (PE)**—a file created by the .NET compiler that contains MSIL code as well as metadata about the application.

### **Lesson Sequence**

#### **Activating prior knowledge/lesson staging (5 minutes):**

1. As indicated on the “Anticipatory Set” slide in the presentation, discuss the managed execution process used by the .NET Framework.

#### **Lesson activity (20 minutes):**

1. Use the Microsoft PowerPoint® presentation to review MSIL, JIT compilers, and managed execution.

#### **Assessment/lesson reflection (25 minutes):**

1. Direct students to complete the MSIL disassembly assignment in the Student Activity document 10754\_Msft.NET\_SA\_3.1.
2. After the students have completed the activity and responded to the prompt at the end of the worksheet, lead a discussion of their experiences. Although the details of MSIL are beyond the scope of this lesson, it is valuable for students to look at this important step in the execution process.

**Resources:**

- **MSDN®: Compiling MSIL to Native Code**  
*<http://msdn.microsoft.com/en-us/library/ht8ecch6.aspx>*
- **MSDN: Compiling to MSIL**  
*<http://msdn.microsoft.com/en-us/library/c5tkafs1.aspx>*
- **MSDN: The Microsoft Shared Source CLI Implementation**  
*<http://msdn.microsoft.com/en-us/library/ms973879.aspx>*
- **MSDN: Managed Execution Process**  
*<http://msdn.microsoft.com/en-us/library/k5532s8a.aspx>*