Amazon Mechanical Turk

Getting Started Guide

API Version 2013-11-15

Amazon Mechanical Turk: Getting Started Guide

Table of Contents

Welcome 1
- How Do I...? 1

Introduction to Amazon Mechanical Turk 3
- Overview of Amazon Mechanical Turk 3
  - Features 3
- Key Amazon Mechanical Turk Concepts 4
  - Requester 4
  - Human Intelligence Task 4
  - Worker 4
  - Assignment 4
  - Reward 4
- What's Next? 5

Setting Up Accounts and Tools 6
- Sign Up for an AWS Account 6
  - AWS Security Credentials 6
- Create a Requester Account 7
- Prepay for Your HITs 7
  - The Sandbox Testing Environment 8
- Download the Appropriate Amazon Mechanical Turk SDK and Tools 8
  - Installing Command Line Tools 8
  - Install C# Tools 9
  - Install Java Tools 9
  - Install Perl Tools 9
  - Install Ruby Tools 9
  - Creating a HIT Tutorial 10

Creating a HIT 11
- Workflow 11
- Working with the Examples 12
  - Showing Your Preferred Programming Language 12
  - Creating a HIT Using the Command Line Interface 12
  - Creating a HIT Using the API 14
- What's Next? 26

Implementing Amazon Mechanical Turk 27
- Interfaces 27
- Considerations 28
  - Considerations for Writing a HIT 28
- Common Use Scenarios 28
  - Photo and Video Processing 29
  - Data Verification and Clean-up 29
  - Information Gathering 29
  - Data Processing 29
- Coding Resources 29
- Advanced Functionality 30
- Reference Resources 31

Document History 32

Welcome

This is the *Amazon Mechanical Turk Getting Started Guide*. This guide provides a conceptual overview of Amazon Mechanical Turk for developers who want to write scripts or software applications using the Amazon Mechanical Turk SDKs. The major sections of this guide are described in the following table. For information on using a graphical user interface to access Amazon Mechanical Turk functionality, go to the Amazon Mechanical Turk Requester User Interface Guide.

Amazon Mechanical Turk is a web service that provides an on-demand, scalable, human workforce to complete jobs that humans can do better than computers, such as recognizing objects in photographs. For more information about this product, go to Amazon Mechanical Turk.

How Do I...?

How do I...?	Relevant Topics
Get a general product overview of Mechanical Turk	Introduction to Amazon Mechanical Turk (p. 3)
Set up the accounts and tools needed to use Mechanical Turk	Setting Up Accounts and Tools (p. 6)
Use the Mechanical Turk SDKs to write and publish a Human Intelligence Task (HIT)	Creating a HIT (p. 11)
Learn more about implementing Mechanical Turk	Implementing Amazon Mechanical Turk (p. 27)
Find answers to common questions about Mechanical Turk	Amazon Mechanical Turk Technical FAQ
Learn more about programmatically interacting with Mechanical Turk	Amazon Mechanical Turk Developer Guide
Find detailed information about Mechanical Turk operations	Amazon Mechanical Turk API Reference

How do I...?	Relevant Topics
Find information about the latest release of Mechanical Turk	Amazon Mechanical Turk Release Notes

Introduction to Amazon Mechanical Turk

The following topics provide a high-level overview of the Amazon Mechanical Turk web service. After reading these topics, you should understand the basics you need to work through the examples in this guide.

Topics

- Overview of Amazon Mechanical Turk (p. 3)
- Key Amazon Mechanical Turk Concepts (p. 4)
- What's Next? (p. 5)

Overview of Amazon Mechanical Turk

Amazon Mechanical Turk provides an on-demand, scalable, human workforce to complete jobs that humans can do better than computers. Amazon Mechanical Turk software formalizes job offers to the thousands of Workers willing to do piecemeal work at their convenience. The software also retrieves work performed and compiles it for you, the Requester, who pays the Workers for satisfactory work (only). Optional qualification tests enable you to select competent Workers.

The kinds of tasks humans can complete better than computers includes finding objects in photos, writing reviews of restaurants, movies, or businesses, translating text passages into foreign languages, getting the hours of operation of the business center within a hotel, determining if a hotel is family-friendly, or telling you the most relevant search results for a given phrase.

This guide presents a very slim slice of the Amazon Mechanical Turk API. For a complete description of the entire API, go to the Amazon Mechanical Turk API Reference. For more information about using the API, go to the Amazon Mechanical Turk Developer Guide.

Features

The following list describes the features of Amazon Mechanical Turk highlighted by the tutorial in this guide.

- **On-demand workforce—**Amazon Mechanical Turk provides access to a virtual community of Workers.

- **Create jobs that Workers perform over the Internet—**Advertise your job to the thousands of Amazon Mechanical TurkWorkers around the world
  You prescribe the job (HIT) that Workers complete using their computer, and pay them for their work.
- **Test and publish your jobs—**Test your applications in the Amazon Mechanical Turk sandbox
  Test your jobs in the Amazon Mechanical Turk sandbox and publish the revised jobs to the outside world.

Amazon Mechanical Turk provides SDKs and command line tools to make it easier to build solutions leveraging Amazon Mechanical Turk.

Key Amazon Mechanical Turk Concepts

This section describes the concepts and terminology you need to understand to use Amazon Mechanical Turk effectively.

Requester

A *Requester* is a company, organization, or person that creates and submits tasks (HITs) to Amazon Mechanical Turk for Workers to perform. As a Requester, you can use a software application to interact with Amazon Mechanical Turk to submit tasks, retrieve results, and perform other automated tasks. You can use the Requester website to check the status of your HITs, and manage your account.

Human Intelligence Task

A *Human Intelligence Task (HIT)* is a task that a Requester submits to Amazon Mechanical Turk for Workers to perform. A HIT represents a single, self-contained task, for example, "Identify the car color in the photo." Workers can find HITs listed on the Amazon Mechanical Turk website. For more information, go to the Amazon Mechanical Turk website.

Each HIT has a lifetime, specified by the Requester, that determines how long the HIT is available to Workers. A HIT also has an assignment duration, which is the amount of time a Worker has to complete a HIT after accepting it.

Worker

A *Worker* is a person who performs the tasks specified by a Requester in a HIT. Workers use the Amazon Mechanical Turk website to find and accept assignments, enter values into the question form, and submit the results. The Requester specifies how many Workers can work on a task. Amazon Mechanical Turk guarantees that a Worker can work on each task only one time.

Assignment

An *assignment* specifies how many people can submit completed work for your HIT. When a Worker accepts a HIT, Amazon Mechanical Turk creates an assignment to track the work to completion. The assignment belongs exclusively to the Worker and guarantees that the Worker can submit results and be eligible for a reward until the time the HIT or assignment expires.

Reward

A *reward* is the money you, as a Requester, pay Workers for satisfactory work they do on your HITs.

What's Next?

The next section explains how to sign up for AWS and for Amazon Mechanical Turk, and how to set up your development environment.

Setting Up Accounts and Tools

The following topics describe the tasks you need to perform before you can use Amazon Mechanical Turk.

Topics

- Sign Up for an AWS Account (p. 6)
- Create a Requester Account (p. 7)
- Prepay for Your HITs (p. 7)
- Download the Appropriate Amazon Mechanical Turk SDK and Tools (p. 8)

Sign Up for an AWS Account

To develop solutions using the Amazon Mechanical Turk web service, you must first sign up for an AWS account. An AWS account is an Amazon.com account that enables you to use services from AWS. Although you can use the login name and password of your personal Amazon.com account, you might want to sign up for a separate Amazon.com account to use for your AWS activity.

If you already have an AWS account, you can skip to the next section. For more information, see Create a Requester Account (p. 7).

To sign up for an AWS account

1. Go to http://aws.amazon.com, and then click **Sign Up**.
2. Follow the on-screen instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a PIN using the phone keypad.

Using your AWS account, you can view your AWS account activity and usage reports and manage your security credentials.

AWS Security Credentials

AWS uses special identifiers called access keys to help protect your data. For your AWS account, you'll have two keys, an Access Key ID, and a Secret Access Key.

- Sample Access Key ID: AKIAIOSFODNN7EXAMPLE
- Sample Secret Access Key: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

You use access keys to sign programmatic requests that you make to Amazon Mechanical Turk, and to work with command line interface. You can use these access identifiers in both the sandbox and the production environment.

> **Important**
> Amazon Mechanical Turk does not use AWS Identity and Access Management (IAM) credentials.

To manage your AWS security credentials

1. Go to the Amazon Web Services website at http://aws.amazon.com/security-credentials.
2. Sign in.
3. Click **Continue to Security Credentials** (Amazon Mechanical Turk does not use IAM credentials).

Create a Requester Account

Before you can use Amazon Mechanical Turk, you must have an Amazon Mechanical Turk Requester account.

To create and register a Requester account

1. Go to http://requester.mturk.amazon.com.
2. Click **Create an Account**.
3. Enter your email address.
4. Follow the prompts to complete your Requester account registration.
   When prompted, you must enter your mailing address and you must accept the **Amazon Mechanical Turk Participation Agreement**.

Prepay for Your HITs

The Creating a HIT (p. 11) tutorial uses the Amazon Mechanical Turk sandbox in which you can execute Amazon Mechanical Turk operations without having to pay Workers for working on your HITs. We recommend that you use the sandbox to test your applications before moving them to the production system. Before you use the production version of Amazon Mechanical Turk you need to prepay for the HITs you create. Otherwise, you can't post your HITs to Workers. For more information about the sandbox, see The Sandbox Testing Environment (p. 8).

To post your HITs to Workers, you must have money in your Prepaid HIT Balance to prepay for all of your HITs. You can provide banking information, credit card or debit card information, or an Amazon Payments account to prepay for the HITs.

If you use a bank account, it can take up to one week for your transfer to be approved. During that time you can test your HITs in the Amazon Mechanical Turk Developer sandbox, but you cannot create HITs on the Amazon Mechanical Turk production system. If you use a credit card, debit card, or Amazon Payments account to prepay, you can use the Amazon Mechanical Turk production system immediately.

For instructions on how to prepay for your HITs, go to the Requester website.

The Sandbox Testing Environment

You should test your HITs in the Amazon Mechanical Turk sandbox testing environment to make sure they work as expected before publishing your HITs in the Mechanical Turk marketplace. The sandbox provides a testing environment where you can publish and work on HITs to try them out before publishing them in the Amazon Mechanical Turk Marketplace. The sandbox consists of a Requester sandbox website and a Worker sandbox website.

You will need to create a Requester account on the Requester sandbox website, which is located at https://requestersandbox.mturk.com. You'll also need to create a Worker account on the Worker sandbox website located at https://workersandbox.mturk.com to view your sandbox created HITs as a Worker. There is no charge for using the Mechanical Turk sandbox sites.

When testing in the sandbox, test each HIT in all the major Internet browsers to make sure your HIT works the same in each browser. If you have links to pictures or videos in your HITs, make sure the links work. Also, your testing should include verifying that the format of the submitted answers is acceptable.

Download the Appropriate Amazon Mechanical Turk SDK and Tools

Amazon Mechanical Turk provides SDKs and command line tools designed to make it easier for you to build solutions leveraging Amazon Mechanical Turk. These SDKs and tools hide the complexity and simplify using the Amazon Mechanical Turk APIs.

The sections below provide instructions for installing the SDKs and command line tools. You must install one of the SDKs or the command line tools to run the examples in this guide.

Installing Command Line Tools

You must install and configure the command line tools correctly before you can use the command line to work through the examples in this guide.

If you use Unix, use the following procedure.

To install the Amazon Mechanical Turk command line tools for Unix users

1. Go to Amazon Mechanical Turk Command Line Tools.
2. Go to the `[`*`Command Line Tools Installation Directory`*`]\` directory and open the file `Overview.html`.
3. Follow the instructions in the section "Installing the command line tools" in the `Overview.html` file to install and configure the command line tools.
   You need your Access Key ID and Secret Access Key for this step.

If you use Windows, use the following procedure.

To install the Amazon Mechanical Turk command line tools for Windows Users

1. Go to Amazon Mechanical Turk Command Line Tools and download the command line tools.
2. Run `mech-turk-setup.exe` and follow the instructions.

   This setup wizard installs and configures the command line tools. You need your Access Key ID and Secret Access Key for this step.

Install C# Tools

To install and configure the Amazon Mechanical Turk SDK for .Net

1. Go to Amazon Mechanical Turk SDK for .NET and download the Amazon Mechanical Turk SDK for .NET.
2. Open the file `[SDK Installation Directory]\Overview.html`.
3. Verify that you meet the prerequisites listed in the "Prerequisites" section of the `Overview.html` file.
4. Follow the instructions in the section "Installing the Amazon Mechanical Turk SDK for .NET" of the `Overview.html` file to install and configure the SDK.
   You need your Access Key ID and Secret Access Key for this step.

You must configure the SDK correctly before you can use the examples in this guide.

Install Java Tools

To install the Amazon Mechanical Turk SDK for Java

1. Go to Amazon Mechanical Turk SDK for Java and download the SDK for Java.
2. Open the file `[SDK Installation Directory]\Overview.html`.
3. Verify that you meet the prerequisites listed in the "Prerequisites" section of the `Overview.html` file.
4. Follow the instructions in the section "Installing the SDK" of the `Overview.html` file to install and configure the SDK.
   You need your Access Key ID and Secret Access Key for this step.

You must configure the SDK correctly before you can use the examples in this guide.

Install Perl Tools

To install the Amazon Mechanical Turk SDK for Perl

1. Go to Amazon Mechanical Turk SDK for Perl and download the SDK for Perl.
2. Open the file `[SDK Installation Directory]\README`.
3. Verify that you meet the prerequisites listed in the "Prerequisites" section of the `README` file.
4. Follow the instructions in the "Installation" section of the `README` file to install and configure the SDK.
   You need your Access Key ID and Secret Access Key for this step.

You must configure the SDK correctly before you can use the examples in this guide.

Install Ruby Tools

To install the Ruby Libraries for Amazon Web Services

1. Go to Ruby Libraries for Amazon Web Services and download the Ruby libraries.
2. Open the file `[SDK Installation Directory]\README`.
3. Verify that you meet the prerequisites listed in the "Prerequisites" section of the `README` file.

4. Follow the instructions in the "Installation" section of the README file to install and configure the libraries.
   You need your Access Key ID and Secret Access Key for this step.

You must configure the SDK correctly before you can use the examples in this guide.

Creating a HIT Tutorial

The Creating a HIT (p. 11) tutorial in the next section takes you step-by-step through using Amazon Mechanical Turk to create a HIT. The tutorial is written procedurally so you should follow it from beginning to end. After completing the tutorial, you should have a good feel for the major tasks you can complete using Amazon Mechanical Turk.

Alternately, you can skip the **Creating a HIT** tutorial and jump right to the last section that provides links to code samples, application examples, forums, and other resources designed to help you learn Amazon Mechanical Turk. For more information, see Implementing Amazon Mechanical Turk (p. 27).

Creating a HIT

In the previous section you set up your AWS account and installed the programming tools you need to use Amazon Mechanical Turk. The following topics describe how to write, test, publish, and manage a Human Intelligence Task (HIT).

Topics

- Workflow (p. 11)
- Working with the Examples (p. 12)
- What's Next? (p. 26)

Workflow

The following procedure gives you an overview of creating, testing, publishing, and managing a HIT.

Workflow for Requesters

1	Write your HIT. Write the question or task for Workers to answer or perform.
2	Test your HIT. Publish your HIT on the *Amazon Mechanical Turk Developer Sandbox*. The *Amazon Mechanical Turk Developer Sandbox* is a simulated environment that allows you to view your HIT as it would appear to Workers. For more information about the *Amazon Mechanical Turk Developer Sandbox* and how to use it, go to the Amazon Mechanical Turk Developer Sandbox.
3	Publish your HIT on the Amazon Mechanical Turk production system. This step makes your HIT available to Workers.
4	Workers accept your HIT and complete the assignment. You can view the status of your HITs. For more information, go to the Requester website.
5	Process the assignment results. When a Worker completes an assignment, you can view the results, output the results to a file, and accept or reject the work. Accepting the work means that you agree to pay the Worker.

6	Manage your HIT. You can extend the completion time for your HIT, expire the HIT early, add additional assignments, modify the HIT properties, or block Workers whose work does not meet your standards.

The examples in this section provide instructions for performing tasks 1, 2, and 3. For information about tasks 5 and 6, see Implementing Amazon Mechanical Turk (p. 27).

Working with the Examples

The following examples use SDK operations or the command line tools to show you how to create a survey that asks the question "How many movies have you seen this month?" The samples create a HIT and print the HIT ID to the console. These examples demonstrate how to publish the HIT to the *Amazon Mechanical Turk Developer Sandbox* for testing and then how to publish the HIT on the Amazon Mechanical Turk production system.

The following sections show you how to create and publish a HIT using the command line interface or the API and a programming language:

- Creating a HIT Using the Command Line Interface (p. 12)
- API and C# (p. 15)
- API and Java (p. 17)
- API and Perl (p. 20)
- API and Ruby (p. 23)

Showing Your Preferred Programming Language

If you are viewing the HTML version of this guide, you can hide the text in this section that does not pertain to the programming language you are using. There is a **Filter View:** drop-down menu in the upper-right corner of pages with language-specific text. Select a computer language to show the examples in only that computer language, or select **All** to show the examples in all available computer languages.

You can also select **Command Line Tools**, which hides all code samples and leaves only information related to the command line tools.

Creating a HIT Using the Command Line Interface

Topics

- How to Create a HIT (p. 12)
- How to Test Your HIT (p. 14)
- How to Publish Your HIT (p. 14)

This example uses the commands in the *Amazon Mechanical Turk command line tools* to create a survey. You must have the command line tools installed and configured correctly to run this example.

How to Create a HIT

The following procedure describes how to create the HIT, "MovieSurvey."

To create the HIT

1. Create an input file, `moviesurvey.input`, a tab-delimited text file, and populate it with the variable data for the HIT.

```
question
How many movies have you seen this month?
```

2. Create a question template file, `moviesurvey.question`, and add the text of your HIT template. This XML file contains placeholders for the fields defined in the input file and conforms to the QuestionForm schema.

```
<?xml version="1.0" encoding="UTF-8"?>
<QuestionForm xmlns="http://mechanicalturk.amazonaws.com/AWSMechanicalTurk
DataSchemas/2005-10-01/QuestionForm.xsd">
  <Question>
   <QuestionIdentifier>1</QuestionIdentifier>
   <QuestionContent>
     <Text>${question}</Text>
   </QuestionContent>
   <AnswerSpecification>
     <FreeTextAnswer/>
   </AnswerSpecification>
  </Question>
</QuestionForm>
```

3. Create a HIT properties file, `moviesurvey.properties`, and populate it with the properties of the HIT (*`title`*, *`description`*, etc.)

```
######################################
##  HIT Properties
######################################

title:Movie Survey
description:This is a survey to find out how many movies you have watched
recently.
keywords:movie, survey
reward:0.05
assignments:100
annotation:sample#command

######################################
## HIT Timing Properties
######################################

# this Assignment Duration value is 60 * 60 = 1 hour
assignmentduration:3600

# this HIT Lifetime value is 60*60*24*3 = 3 days
hitlifetime:259200
```

```
# this Auto Approval period is 60*60*24*15 = 15 days
autoapprovaldelay:1296000
```

4. Navigate to the [*Command Line Tools Installation Directory*]\bin directory to run the commands.

How to Test Your HIT

To test your HIT, publish it in the sandbox by including the -sandbox argument with the loadHITs command.

To test your HIT

1. Open a command prompt, navigate to the [*Command Line Tools Installation Directory*]\bin directory, and run the following command:

```
loadHITs -input c:\moviesurvey\moviesurvey.input -question  c:\moviesur
vey\moviesurvey.question -properties c:\moviesurvey\moviesurvey.properties
 -sandbox
```

2. Go to the Amazon Mechanical Turk Developer Sandbox and view your HIT.
   You can rewrite and rerun your HIT as often as you like.

How to Publish Your HIT

When you are satisfied with your HIT, publish it to the production system.

To publish your HIT to the production system

- Open a command prompt, navigate to the [*Command Line Tools Installation Directory*]\bin directory and run the following command:

```
loadHITs -input c:\moviesurvey\moviesurvey.input -question  c:\moviesur
vey\moviesurvey.question -properties c:\moviesurvey\moviesurvey.properties
```

  Your HIT is now on the production site.

Creating a HIT Using the API

Topics

- C# (p. 15)
- Java (p. 17)
- Perl (p. 20)
- Ruby (p. 23)

The following sections explain how to create a HIT using the Amazon Mechanical Turk API.

C#

The following example uses the methods of the *Amazon Mechanical Turk SDK for .Net* to create the movie survey. You must have the SDK installed and configured correctly to run this example. To download the SDK go to Amazon Mechanical Turk SDK for .NET.

How to Create a HIT

This section shows how to create a HIT.

To create the HIT

1. Add a project reference to `[`*`SDK Installation Directory`*`]\lib\Amazon.WebServices.MechanicalTurk.dll`.
2. Add the following `using` statements to your source code:

   - using Amazon.WebServices.MechanicalTurk
   - using Amazon.WebServices.MechanicalTurk.Domain

3. Copy the `app.config` file from the `[`*`SDK Installation Directory`*`]\src` directory to your project directory and add the file to your project. This file contains the configuration and the AWS identifier information for your application.
4. Create a `SimpleClient` object. This class provides a client used to invoke single operations for Mechanical Turk. Typically, you create this object once and use it throughout your application.

```
static SimpleClient client = new SimpleClient();
```

5. Use one of the `CreateHIT()` methods in the `SimpleClient` class to create a HIT. The following example uses the `CreateHIT()` method that requires values for the *`title`*, *`description`*, *`reward`*, *`question`*, and *`maxAssignments`* parameters of the method.

```
HIT hit = client.CreateHIT(
    "Movie Survey",
   "This is a survey to find out how many movies you have watched recently.",

    new decimal(0.05), "How many movies have you seen this month?", 100);
```

The following example pulls together all the code described in the preceding procedure.

```
using System;
using System.Collections.Generic;
using System.Text;
using Amazon.WebServices.MechanicalTurk;
using Amazon.WebServices.MechanicalTurk.Domain;

namespace CreateHITExample
```

```
{
    class Program
    {
        // Create a client as a class member.
        static SimpleClient client = new SimpleClient();

        static void Main(string[] args)
        {
            CreateNewHIT();
        }

        static void CreateNewHIT()
        {

            // Create a HIT with the specified parameters.
            HIT hit = client.CreateHIT(
                "Movie Survey",
                // title
                "This is a survey to find out how many movies you have watched
 recently.",    // description
                new decimal(0.05),
                // reward
                "How many movies have you seen this month?",
                // question
                100);
                // number of assignments

            // Write the new HIT Id to the console.
            Console.WriteLine("Created HIT: {0}", hit.HITId);
          Console.WriteLine("Hit Location: {0}", client.GetPreviewURL(hit.HIT
TypeId));

        }
    }
}
```

How to Test Your HIT

When you compile and run the code example, your HIT is published on the *Amazon Mechanical Turk Developer Sandbox* by default. This is a simulated environment that enables you to view your HIT as it would appear to Workers. The sandbox is a free test environment for all Requesters.

To test your HIT

1. Compile and run your project.
   If no errors occur, you see output similar to the following:

   ```
   Created HIT: 2X6T66XZQARRM98X5030
   HIT Location: http://workersandbox.mturk.com/mturk/preview?groupId=2X6T66XZQAR
   RM98X5030
   ```

2. Go to the Amazon Mechanical Turk Developer Sandbox and view your HIT. You can rewrite and recompile your HIT as often as necessary.

How to Publish Your HIT

When you are satisfied with your HIT, publish it on the Amazon Mechanical Turk production system. This makes your HIT available for Workers to complete.

To publish your HIT

1. Open the `App.config` file for your project.
2. Under element <appSettings>, find the following entry:

```
<add key="MechanicalTurk.ServiceEndpoint" value="https://mechanicalturk.sand
box.amazonaws.com?Service=AWSMechanicalTurkRequester"/>
```

3. Replace the entry with the following string:

```
<add key="MechanicalTurk.ServiceEndpoint" value="https://mechanicalturk.amazon
aws.com/?Service=AWSMechanicalTurkRequester"/>
```

4. Save and close the file.
5. Run your application.
   Your HIT is now on the production site.

Java

The following example uses the APIs of the *Amazon Mechanical Turk SDK for Java* to create the movie survey. You must have the SDK installed and configured correctly before you run this example. To download the SDK go to Amazon Mechanical Turk SDK for Java.

How to Create a HIT

This section shows how to create a HIT.

To create the HIT

1. Specify all the `*.jar` files in the `[SDK Installation Directory]\lib` directory and in the `[SDK Installation Directory]\lib\third-party` directory in your classpath.
2. Specify all the `*.jar` files in the `[SDK Installation Directory]\build\lib` directory in your classpath.
3. Copy the `mturk.properties` file to your project. This file contains the configuration information needed for your project.
4. Import the following classes to your application.
   - com.amazonaws.mturk.service.axis.RequesterService
   - com.amazonaws.mturk.service.exception.ServiceException
   - com.amazonaws.mturk.util.ClientConfig
   - com.amazonaws.mturk.requester.HIT
5. Create a `RequesterService` object.This class provides the configuration information for your application.

```
service = new RequesterService(new PropertiesClientConfig());
```

6. Use one of the `CreateHIT()` methods in the `RequesterService` class to create a HIT. The following example uses the `CreateHIT()` method with parameter values for the *`title`*, *`description`*, *`reward`*, and *`maxAssignments`* properties of the HIT.

```
HIT hit = service.createHIT
(
     title,
     description,
     reward,
     RequesterService.getBasicFreeTextQuestion(
     "How many movies have you seen this month?"),
     numAssignments);
```

The following example pulls together all the code described in the preceding procedure.

```
package createnewhit;

import com.amazonaws.mturk.service.axis.RequesterService;
import com.amazonaws.mturk.service.exception.ServiceException;
import com.amazonaws.mturk.util.PropertiesClientConfig;
import com.amazonaws.mturk.requester.HIT;

/**
 * The MovieSurvey sample application creates a simple HIT using the
 * Amazon Mechanical Turk SDK for Java. The file mturk.properties must be found
 in the current file path.
 */
public class MovieSurvey{

   private RequesterService service;

   // Define the properties of the HIT to be created.
   private String title = "Movie Survey";
   private String description =
    "This is a survey to find out how many movies you have watched recently.";

   private int numAssignments = 100;
   private double reward = 0.05;

   /**
    * Constructor
    */
   public MovieSurvey()
   {
      service = new RequesterService(new PropertiesClientConfig());
   }

   /**
    * Create a simple survey.
    *
    */
   public void createMovieSurvey()
   {
     try
     {
       // The createHIT method is called using a convenience static method
      // RequesterService.getBasicFreeTextQuestion() that generates the question
 format
       // for the HIT.
       HIT hit = service.createHIT
       (
          title,
          description,
          reward,
          RequesterService.getBasicFreeTextQuestion(
               "How many movies have you seen this month?"),
          numAssignments);

       // Print out the HITId and the URL to view the HIT.
       System.out.println("Created HIT: " + hit.getHITId());
       System.out.println("HIT location: ");
       System.out.println(service.getWebsiteURL() + "/mturk/preview?groupId="
+ hit.getHITTypeId());

    }
    catch (ServiceException e)
    {
       System.err.println(e.getLocalizedMessage());
    }
  }

   /**
    * Main method
    *
    * @param args
    */
   public static void main(String[] args)
   {
     // Create an instance of this class.
     MovieSurvey app = new MovieSurvey();

     // Create the new HIT.
     app.createMovieSurvey();
   }
}
```

How to Test Your HIT

This section shows how to test your HIT.

When you compile and run the code example, your HIT is published on the *Amazon Mechanical Turk Developer Sandbox*. This is a simulated environment that enables you to view your HIT as it would appear to Workers. The sandbox is a free test environment for all Requesters.

To test your HIT

1. Compile and run your project.
   If no errors occur, you see output similar to the following:

```
Created HIT: 2X6T66XZQARRM98X5030
HIT Location:
http://workersandbox.mturk.com/mturk/preview?groupId=2X6T66XZQARRM98X5030
```

2. Go to the Amazon Mechanical Turk Developer Sandbox and view your HIT. You can rewrite and recompile your HIT as often as necessary.

How to Publish Your HIT

When you are satisfied with your HIT, publish it on the Amazon Mechanical Turk production system. This makes your HIT available for Workers to complete.

To publish your HIT

1. Open the `mturk.properties` file for your project.
2. Comment out the Developer Sandbox service_url with a "#".

```
#service_url=https://mechanicalturk.sandbox.amazonaws.com/?Service=AWSMech
anicalTurkRequester
```

3. Remove the comment mark # from the Production site service_url.

```
service_url=https://mechanicalturk.amazonaws.com/?Service=AWSMechanicalTurkRe
quester
```

4. Save and close the file.
5. Compile and run your application.
   Your HIT is now on the production site.

Perl

This example creates a movie survey using the methods of the *Amazon Mechanical Turk SDK for Perl*. You must have the SDK installed and configured correctly before you can run this example. To download the SDK go to Amazon Mechanical Turk SDK for Perl.

How to Create a HIT

You can use several different methods to create a HIT. You can define the format for the question and answer in a file. The following example writes the question format and the question in the script file. The example prints out the ID and the location of the new HIT to the console.

To create the HIT

1. Add the following line to your module to use the `MechanicalTurk` class.

```
use NET::Amazon::MechanicalTurk;
```

2. Define the question format. In this example, this is a string that defines the XML format of the question.

```
my $questionXml = <<END_XML;
<?xml version="1.0" encoding="UTF-8"?>
<QuestionForm xmlns="http://mechanicalturk.amazonaws.com/AWSMechanicalTurk
DataSchemas/2005-10-01/QuestionForm.xsd">
  <Question>
    <QuestionIdentifier>1</QuestionIdentifier>
    <QuestionContent>
      <Text>$question</Text>
    </QuestionContent>
    <AnswerSpecification>
      <FreeTextAnswer/>
    </AnswerSpecification>
  </Question>
</QuestionForm>
END_XML
```

3. Create a string that specifies the question for the HIT.

```
my $question = "How many movies have you seen this month?";
```

4. Create a client by creating an instance of the `MechanicalTurk` class.

```
my $mturk = Net::Amazon::MechanicalTurk->new;
```

5. Use the `CreateHIT()` method of the `MechanicalTurk` class and specify the properties of the HIT.

```
my $result = $mturk->CreateHIT(
    Title       => 'Movie Survey',
    Description => 'This is a survey to find out how many movies you have
watched recently',
    Keywords    => 'movie, survey',
    Reward => {
        CurrencyCode => 'USD',
        Amount       => 0.05
    },
    RequesterAnnotation         => 'Movie Survey',
    AssignmentDurationInSeconds => 60 * 60 * 24,
    AutoApprovalDelayInSeconds  => 60 * 60 * 10,
    MaxAssignments              => 100,
    LifetimeInSeconds           => 60 * 60,
    Question                    => $questionXml
);
```

The following example pulls together all the code described in the preceding procedure.

```
#!/usr/bin/perl
use strict;
use warnings;
use Net::Amazon::MechanicalTurk;
use Net::Amazon::MechanicalTurk::IOUtil;
```

```
# Create the question to ask in the HIT.
my $question = "How many movies have you seen this month?";

# Create the format for the question.
my $questionXml = <<END_XML;
<?xml version="1.0" encoding="UTF-8"?>
<QuestionForm xmlns="http://mechanicalturk.amazonaws.com/AWSMechanicalTurkData
Schemas/2005-10-01/QuestionForm.xsd">
  <Question>
    <QuestionIdentifier>1</QuestionIdentifier>
    <QuestionContent>
      <Text>$question</Text>
    </QuestionContent>
    <AnswerSpecification>
      <FreeTextAnswer/>
    </AnswerSpecification>
  </Question>
</QuestionForm>
END_XML

# Create a MechanicalTurk object.
my $mturk = Net::Amazon::MechanicalTurk->new;

# Create the HIT with the properties specified.
my $result = $mturk->CreateHIT(
    Title       => 'Movie Survey',
    Description => 'This is a survey to find out how many movies you have watched
 recently',
    Keywords    => 'movie, survey',
    Reward => {
        CurrencyCode => 'USD',
        Amount       => 0.05
    },
    RequesterAnnotation         => 'Movie Survey',
    AssignmentDurationInSeconds => 60 * 60 * 24,
    AutoApprovalDelayInSeconds  => 60 * 60 * 10,
    MaxAssignments              => 100,
    LifetimeInSeconds           => 60 * 60,
    Question                    => $questionXml
);

# Print the HIT Id to the console.
printf "Created HIT:\n";
printf "HITId:     %s\n", $result->{HITId}[0];
printf "\nHIT Location: %s\n", $mturk->getHITTypeURL($result->{HITTypeId}[0]);
```

How to Test Your HIT

When you run this code example, your HIT is published on the *Amazon Mechanical Turk Developer Sandbox* by default. This is a simulated environment that enables you to view your HIT as it would appear to Workers. The sandbox is a free test environment for all Requesters.

To test your HIT

1. Run the sample to publish the HIT on the *Amazon Mechanical Turk Developer Sandbox*.
   If no errors occur, you see output similar to the following:

```
Created HIT: 2X6T66XZQARRM98X5030
HIT Location: http://workersandbox.mturk.com/mturk/preview?groupId=2X6T66XZQAR
RM98X5030
```

2. Go to the Amazon Mechanical Turk Developer Sandbox and view your HIT. You can rewrite and recompile your HIT as often as necessary.

How to Publish Your HIT

When you are satisfied with your HIT, publish it on the Amazon Mechanical Turk production system. This makes your HIT available for Workers to complete.

To publish your HIT

1. Specify the production system in the `constructor` of the `MechanicalTurk` object.
   Your code should look like the following:

```
# Create a MechanicalTurk object.
my $mturk = Net::Amazon::MechanicalTurk->new(serviceUrl=>"prod");
```

2. Save and close the file.
3. Compile and run your application.
   Your HIT is now on the production site.

Note
To go back to using the Developer Sandbox, change the `constructor` to "sandbox":

```
# Create a MechanicalTurk object.
my $mturk = Net::Amazon::MechanicalTurk->new(serviceUrl=>"sandbox");
```

Ruby

This example uses the methods in the *Ruby Libraries for Amazon Web Services* to create the movie survey. You can use several different methods to create a HIT. You can write the question format and the question in the script file. In this example we create the question and the format in a file, and then print the ID of the new HIT to the console.

You must have the *Ruby Libraries for Amazon Web Services* installed and configured correctly before you run this example. To download the Ruby Librairies go to Ruby Libraries for Amazon Web Services. The first time you run this example, or one of the examples in the *Ruby Libraries for Mechanical Turk*, you must provide your Access Key ID and Secret Access Key.

How to Create a HIT

This section shows how to create a HIT.

To write the HIT

1. Create a question file, `moviesurvey.question`, that contains the question for your HIT and the template for the question and answer.This is an XML file that conforms to the QuestionForm schema.

```
<?xml version="1.0" encoding="UTF-8"?>
<QuestionForm xmlns="http://mechanicalturk.amazonaws.com/AWSMechanicalTurk
DataSchemas/2005-10-01/QuestionForm.xsd">
  <Question>
   <QuestionIdentifier>1</QuestionIdentifier>
   <QuestionContent>
     <Text>How many movies have you seen this month?</Text>
   </QuestionContent>
   <AnswerSpecification>
     <FreeTextAnswer/>
   </AnswerSpecification>
  </Question>
</QuestionForm>
```

2. Require the ruby-aws library.

```
require 'ruby-aws'
```

3. Use the `createHIT()` method to create a new HIT.

```
def createNewHIT
   title = "Movie Survey"
   desc = "This is a survey to find out how many movies you have watched
recently."
   keywords = "movie, survey"
   numAssignments = 100
   rewardAmount = 0.05
```

The following example pulls together all the code described in the preceding procedure.

```
#!/usr/bin/env ruby

# This sample application creates a simple HIT using Libraries for Amazon Web
Services.

require 'ruby-aws'
@mturk = Amazon::WebServices::MechanicalTurkRequester.new :Host => :Sandbox

# Use this line instead if you want the production website.
#@mturk = Amazon::WebServices::MechanicalTurkRequester.new :Host => :Production


def createNewHIT
  title = "Movie Survey"
  desc = "This is a survey to find out how many movies you have watched re
cently."
  keywords = "movie, survey"
  numAssignments = 100
  rewardAmount = 0.05 # 5 cents

  # Define the location of the externalized question (QuestionForm) file.
  rootDir = File.dirname $0
  questionFile = rootDir + "/moviesurvey.question"
  # Load the question (QuestionForm) file
  question = File.read( questionFile )

  result = @mturk.createHIT( :Title => title,
    :Description => desc,
    :MaxAssignments => numAssignments,
    :Reward => { :Amount => rewardAmount, :CurrencyCode => 'USD' },
    :Question => question,
    :Keywords => keywords )

  puts "Created HIT: #{result[:HITId]}"
  puts "HIT Location: #{getHITUrl( result[:HITTypeId] )}"
  return result
end

def getHITUrl( hitTypeId )
  if @mturk.host =~ /sandbox/
    "http://workersandbox.mturk.com/mturk/preview?groupId=#{hitTypeId}"   #
Sandbox Url
  else
    "http://mturk.com/mturk/preview?groupId=#{hitTypeId}"   # Production Url
  end
end

createNewHIT
```

How to Test Your HIT

When you run this code example, your HIT is published on the *Amazon Mechanical Turk Developer Sandbox* by default. This is a simulated environment that enables you to view your HIT as it would appear to Workers. The sandbox is a free test environment for all Requesters.

To test your HIT

1. Run the sample to publish the HIT on the *Amazon Mechanical Turk Developer Sandbox*.
   If no errors occur, you see output similar to the following:

```
Created HIT: 2X6T66XZQARRM98X5030
HIT Location: http://workersandbox.mturk.com/mturk/preview?groupId=2X6T66XZQAR
RM98X5030
```

2. Go to the Amazon Mechanical Turk Developer Sandbox and view your HIT. You can rewrite and recompile your HIT as often as necessary.

How to Publish Your HIT

When you are satisfied with your HIT, publish it on the Amazon Mechanical Turk production system. This makes your HIT available for Workers to complete.

To publish your HIT

1. In the sample code, find the following line and comment it out:

```
@mturk = Amazon::WebServices::MechanicalTurkRequester.new :Host => :Sandbox
```

2. In the sample code, remove the comment mark # from the following line:

```
#@mturk = Amazon::WebServices::MechanicalTurkRequester.new :Host => :Produc
tion
```

3. Save and close the file.
4. Run your application.
   Your HIT is now on the production site.

What's Next?

You now have a HIT on the Amazon Mechanical Turk product site. You've become familiar with the architecture of the system, some of its basic functionality, and the kind of responses you can expect. The following section explains how to learn more about Amazon Mechanical Turk and how to implement advanced Amazon Mechanical Turk functionality in your applications.

Implementing Amazon Mechanical Turk

The following topics describe information and additional resources you can use to implement a project using Amazon Mechanical Turk.

Topics

- Interfaces (p. 27)
- Considerations (p. 28)
- Common Use Scenarios (p. 28)
- Coding Resources (p. 29)
- Advanced Functionality (p. 30)
- Reference Resources (p. 31)

In the Creating a HIT (p. 11) tutorial you learned how to complete basic Amazon Mechanical Turk tasks. If you didn't use the tutorial, you can learn how to complete basic and advanced Amazon Mechanical Turk tasks using the *Amazon Mechanical Turk Developer Guide* and by looking at code samples. For more information, go to Amazon Mechanical Turk Developer Guide and Amazon Mechanical Turk Sample Code and Libraries, respectively.

Interfaces

Amazon Mechanical Turk offers the following interfaces:

- Command line
- API
- Requester user interface

The Amazon Mechanical Turk command line interface (CLI) makes it easy to use most of the Amazon Mechanical Turk functionality. When you wish to have a hands-on approach to using Amazon Mechanical Turk and have a relatively small number of assignments and results, the CLI is a good choice. For more information, go to Amazon Mechanical Turk Command Line Tool Reference.

When the number of assignments you have or the number of results you have is large, the Amazon Mechanical Turk API is a good choice. The API exercises all of Amazon Mechanical Turk's functionality and enables you to integrate Amazon Mechanical Turk functions programmatically. For more information, go to Amazon Mechanical Turk Developer Guide.

If you have a very large number of similar HITs, consider using the Requester user interface. It merges one question template with lots of question data to create many similar HITs. For more information, go to Amazon Mechanical Turk Requester User Interface.

Considerations

Creating a successful HIT involves more than programming. There is a certain art involved, for example, in pricing a HIT correctly, laying out the question correctly, breaking down the task into HITs, and minimizing the Worker's time spent with the HIT. For that reason, we created a best practices guide that gives detailed instructions about creating an effective HIT. For more information, go to Amazon Mechanical Turk Best Practices Guide.

Some of the HITs you create require the Workers to have special skills. You might, for example, publish a HIT that asks for a translation. Whenever you have a HIT that requires specialized skills, we recommend that you qualify the Workers. Only those who pass the test receive the opportunity to work on your HITs. For more information, go to CreateHIT and read about the `QualificationRequirement` parameter.

Once a Worker becomes qualified, you can grant them qualification status so that they no longer need to complete a qualification test before working on your HITs. For more information, go to AssignQualification.

Considerations for Writing a HIT

This guide presented a HIT for you to use. When you have to create your own, there are several things you should consider:

- What is the problem you are trying to solve?
  What questions do you want the Workers to answer? What is the best way to present the task to Workers? You need this information when you write the description and question for your HIT. You should be familiar with the format to create your question or task. For more information, go to the Amazon Mechanical Turk Developer Guide.
- How much do you want to pay Workers?
  You need to specify a reward for your HIT. Setting too low a reward discourages Workers from working on your HITs. You need to determine what a fair price is for the work you're asking the Workers to do. The best way is to look at similar tasks advertised on the Amazon Mechanical Turk website. For more information, go to the Amazon Mechanical Turk website.
- How many responses do you want?
  This is the number of assignments for your HIT. Sometimes you want only one answer per HIT. When the answer is controversial, however, you might like to get multiple answers (and thus use multiple assignments) per HIT and reach an answer by consensus.
- How much time do you want to allow to complete the task?
  Giving a Worker too long a time potentially delays getting results. Making the duration too short frustrates Workers.

Common Use Scenarios

This section describes some of the ways you can use Amazon Mechanical Turk.

Photo and Video Processing

Amazon Mechanical Turk is well-suited for processing images. In the past, companies have used Mechanical Turk to:

- Tag objects found in an image for easier searching and advertising targeting
- Select from a set of images the best picture to represent a product
- Audit user-uploaded images for inappropriate content
- Classify objects found in satellite imagery

Data Verification and Clean-up

Companies with large online catalogs use Amazon Mechanical Turk to identify duplicate entries and verify item details. Examples include:

- De-duplication of yellow pages directory listings
- Identification of duplicate products in an online product catalog
- Verification of restaurant details, such as the phone number or hours of operation

Information Gathering

The Amazon Mechanical Turk workforce enables you to gather information, such as:

- Allowing people to ask questions from a computer or mobile device about any topic and have Workers return results to those questions
- Filling out survey data on a variety of topics
- Writing reviews, descriptions and blog entries for websites
- Finding specific fields or data elements in large legal and government documents

Data Processing

Use Amazon Mechanical Turk to process data, for example:

- Podcast editing and transcription
- Human powered translation services
- Search engine rating

Coding Resources

To help you code your applications, we provide the following resources:

- **Developer Resources page—** Click the **Developer** tab on the Requester website located at https://requester.mturk.com/ to get to the Developer Resources page, which has links to sample code, documentation, the sandbox, and other helpful information.
- **Sample Code and Libraries—** You can use code samples as a means of understanding how to implement the Amazon Mechanical Turk API. For more information, go to Amazon Mechanical Turk Sample Code and Libraries.

- **Customer Forum—**We recommend you look at the Amazon Mechanical Turk Forum to get an idea of what other users are doing and to benefit from the questions they've asked.
  The forum can help you understand what you can and can't do with Amazon Mechanical Turk. The forum also serves as a place for you to ask questions that other users or Amazon representatives might answer. You can use the forum to report issues with the service, the API, or the documentation. For more information, go to Amazon Mechanical Turk Discussion Forums.

Advanced Functionality

The tutorial in this guide showed how to accomplish the basic tasks of creating, testing, and publishing your HITs. The Amazon Mechanical Turk API, however, offers advanced functionality, which the following table summarizes. For more information, go to the *Amazon Mechanical Turk Developer Guide*.

Functionality	Description
Review the work submitted by Workers	You reward work well done by paying the Worker. If work is not done well, you can elect not to pay the Worker and block them from future assignments. For more information, go to GetReviewableHITs, RejectAssignment, BlockWorker, and ApproveAssignment.
Give bonuses	You might like to give a bonus to a Worker that does especially good work. This gives the Worker incentive to work on one of your other HITs. By rewarding Workers, you can create a small group of qualified people who you know do good work. They, in turn, will prioritize your HITs over others so that Workers complete your HITs in a timely manner. For more information about awarding bonuses, go to GrantBonus.
Review results programmatically	Reviewing the results manually is an option when the number of assignments is small. As the number of assignments grows, reviewing results programmatically becomes more practical. The Amazon Mechanical Turk API enables you to do that. For more information, go to GetReviewableHITs
Cancel a HIT	There are times when the results, although accurate, are unexpected and unusable by you. In that case, you have to expire the HIT, revise, and recreate it. For more information, go to ForceExpireHIT.
Update HIT properties	You can use API operations and the command line tools to update HIT properties, such as Title, Description, Reward, and Keywords. For more information, go to ChangeHITTypeOfHIT.
Extend or eliminate HITs	You can also use the API operations to extend the completion time for a HIT, expire a HIT early, or add additional assignments. For more information, go to ExtendHIT and DisposeHIT.
Communicate with Workers	You can use NotifyWorkers to send emails to specified Workers.
Return the amount of money in your account that you use to pay Workers	Before you can publish Amazon Mechanical Turk HITs, your account must contain the money required to pay for all of the assignments. You can programmatically access the amount of money in your account using GetAccountBalance.
Create your own qualifications for Workers	We recommend that you use qualification tests when completing your HITs requires specialized skills. You can create your own qualification tests using CreateQualificationType.

Reference Resources

The following list shows additional resources you can use to further your understanding of Amazon Mechanical Turk.

- Learn the pricing for Amazon Mechanical Turk.
  For more information, go to Pricing.
- Use libraries for Amazon Mechanical Turk written in Java and PHP.
  For more information, go to Sample Code and Libraries.
- The Quick Reference Card provides quick access to important Amazon Mechanical Turk operations and functionality.
  For more information, go to Quick Reference Card.
- When you have questions about developing with Amazon Mechanical Turk visit the customer forum to get your questions answered.
  A wide variety of questions have already been answered in the forum. If your question has not been answered, ask it and wait for a fellow developer or Amazon representative to help. For more information, go to Amazon Mechanical Turk Developer Forums.
- The Service Health Dashboard shows you the status of the Amazon Mechanical Turk web service. The dashboard shows you whether Amazon Mechanical Turk (and all other AWS web services) are functioning properly. For more information, go to Service Health Dashboard.
- The *Amazon Mechanical Turk Developer Guide* provides a detailed discussion of the service.
  For more information, go to Amazon Mechanical Turk Developer Guide.
- The HIT page shows the Amazon Mechanical Turk HITs available to work on.
  For more information, go to the HIT page.

Document History

This Document History describes the important changes to the documentation.

- **API version:** 2013-11-15
- **Latest documentation update:** July 17, 2014

Change	Description	Date Changed
New QualificationRequirement Comparators	The following new comparators were added to the `QualificationRequirement` data structure: *`DoesNotExist`*, *`In`*, and *`NotIn`*. For more information about these new comparators, see QualificationRequirement in the *Amazon Mechanical Turk API Reference*.	July 17, 2014
Security Credentials Update	Added information on the new AWS Security Credentials page accessed from the AWS Management Console to the Setting Up Accounts and Tools (p. 6) section of this guide.	July 30, 2013
New Mechanical Turk HITLayout	New for this release is the ability to create a HITLayout in the Amazon Mechanical Turk Requester UI that can be used with the Mechanical Turk API. A HITLayout is a reusable template used to provide Human Intelligence Task (HIT) question data for CreateHIT. For more information, see HITLayout.	March 29, 2012
Mechanical Turk Review Policies	Amazon Mechanical Turk has added Review Policies that you can use to evaluate Worker submissions against a defined set of criteria. For more information, see Review Policies.	December 1, 2011
Mechanical Turk Masters Qualifications	Amazon Mechanical Turk has added two new Mechanical Turk Masters Qualification types: Categorization Masters and Photo Moderation Masters. The new Mechanical Turk Masters role is established for an elite group of Workers who have demonstrated accuracy on specific types of HITs on the Mechanical Turk marketplace. For more information, see QualificationRequirement.	June 22, 2011

Change	Description	Date Changed
Technical documents reorganized	The API reference and the command line tool reference have been split out of the Amazon Mechanical Turk Developer Guide. Now, on the documentation landing page, Amazon Mechanical Turk Documentation you can select the document you want to view. When viewing the documents online, the links in one document will take you, when appropriate, to one of the other guides.	September 16, 2009