Amazon Mechanical Turk

Command Line Reference

API Version 2012-03-25

Amazon Web Services

Amazon Mechanical Turk: Command Line Reference

Amazon Web Services

Welcome

This is the *Amazon Mechanical Turk Command Line Reference*. This guide provides descriptions and samples and other resources related to the Amazon Mechanical Turk command line interface. The major sections of this guide are described in the following table.

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, for example, recognizing objects in photos. For more information about this product go to Amazon Mechanical Turk.

To download the tools go to Amazon Mechanical Turk Command Line Tools.

Commands (p. 2)	Alphabetical list of all Amazon Mechanical Turk commands.
Files Used by the Command Line Tools (p. 53)	Description of the files that the Command Line Tools use.
Regions and Endpoints	Itemized regions and endpoints for all AWS products.

Command Line Reference

Topics

This section describes the commands available in the Command Line Tools.

approveQualificationRequests

Description

The `approveQualificationRequests` command approves a list of Qualification requests. You can obtain the list from a call to getQualificationRequests (p. 24).

For information about Qualifications and Qualification requests, see the Amazon Mechanical Turk Developer Guide.

Arguments

The following table describes the arguments for the `approveQualificationRequests` command.

Name	Description	Required
`-approvefile [filename]`	Specifies a tab delimited text file that contains the list of Qualification requests to approve. For information about this file, see *The Qualification approve file* in Files Used by the Command Line Tools (p. 53). Condition: Required if the *`qualRequest`* argument is not specified. Example: `-approvefile qualification_requests_toapprove.txt`	Conditional
`-force`	Specifies *not* to prompt for manual confirmation before approving the requests. Only advanced developers should use this argument. Example: `-force`	No
`-help` or *`-h`*	Displays the help for this operation. Example: `-help`	No
`-qualRequest [Qualification request ID]`	The Qualification request IDs to approve. Multiple Qualification request IDs are comma separated. Condition: Required if the *`approvefile`* argument is not specified. Example: `-qualRequest TA3ZJBYP2Y7ZJSX2BBN0TZ8ZTM4F6H4ZVQ4DE8FZ`	Conditional
`-sandbox`	Runs this command in the Amazon Mechanical Turk sandbox for testing. This argument takes precedence even if you specify the production web site in your `mturk.properties` file. Example:`-sandbox`	No
`-score [value]`	The default score to assign for each approved Qualification request. Any scores defined in the *`approvefile`* override this default. Example: `-score 100`	No

Example

The following examples for Unix and Windows show how to use the `approveQualificationRequests` command. The examples use a file named `qualifications.txt` that contains 10 requests. Each request in the file gets a score of 100.

Unix

The following example demonstrates how to call this command from Unix.

```
./approveQualificationRequests.sh -approvefile qualifications.txt -score 100
```

Windows

The following example demonstrates how to call this command from Microsoft Windows.

```
approveQualificationRequests -approvefile qualifications.txt -score 100
```

Output

These examples produce output similar to the following, but all 10 requests are listed.

```
You are about to grant 10 qual request(s).
To confirm this operation, please press ENTER (or press Ctrl+C to abort):

[TA3ZJBYP2Y7ZJSX2BBN0TZ8ZTM4F6H4ZVQ4DE8FZ] QualRequest successfully approved
with value (100)
```

Related Commands

- rejectQualificationRequests (p. 35)
- revokeQualification (p. 43)

approveWork

Description

The `approveWork` command approves assignments Workers have submitted to Amazon Mechanical Turk. To specify the assignments to approve you can:

- Obtain the assignment IDs from a call to getResults (p. 26) then call the `approveWork` command and use the *`assignment`* argument to list the IDs. When you use this argument, you are prompted to provide optional comments that the Worker can see in the **Status** section of the web site.
- Use the file that a call the `getResults` returns as the *`approvefile`* argument. You can provide optional comments for Workers in this file. For information about the format of this file, see Files Used by the Command Line Tools (p. 53).
- Provide a path to the `.success` file that a call to loadHITs (p. 30) returns. The `approveWork` command attempts to approve all assignments for all HITs in the `.success` file.

When you use this command, it initiates two payments from your Requester account. Amazon Mechanical Turk pays the reward specified in the HIT to the Worker who submitted the assignment, and also debits your account for any fees. If your Requester account does not have adequate funds for these payments, this command returns an error.

Arguments

The following table describes the arguments for the `approveWork` command.

Name	Description	Required
`-approvefile [filename]`	The name of a text file that contains a list of assignment IDs and, optionally, approval comments. For information about this file, see *The approve file* in Files Used by the Command Line Tools (p. 53). Condition: Required if neither the *`-assignment`* argument nor the *`-successfile`* argument is specified. Example: `-approvefile helloworld_approve.txt`	Conditional
`-assignment [assignment ID]`	The assignment to approve. Multiple assignments IDs are comma separated. Condition: Required if neither the *`-approvefile`* argument nor the *`-successfile`* argument is specified. Example: `-assignment SYSZH6HTMKFG2ZDECWS0`	Conditional
`-force`	Specifies *not* to prompt for manual confirmation before performing the operation. Only advanced developers should use this argument. Example: `-force`	No
`-help` or *`-h`*	Displays the help for this operation. Example: `-help`	No

Name	Description	Required
-sandbox	Runs this command in the Amazon Mechanical Turk sandbox for testing. This argument takes precedence even if you specify the production web site in your `mturk.properties` file. Example: `-sandbox`	No
-successfile [path]	The path to the `.success` file that a call to loadHITs (p. 30) returned. This argument attempts to approve all assignments for all HITs in the file. For information about this file, see *The success file* in Files Used by the Command Line Tools (p. 53). Condition: Required if neither the *-approvefile* argument nor the *-assignment* argument is specified. Example: `-approvefile helloworld_approve.txt`	Conditional

Example

The following examples for Unix and Windows show how to use the `approveWork` command. The examples approve one assignment in the file `approval.txt`.

Unix

The following example demonstrates how to call this command from Unix.

```
./approveWork.sh -approvefile approval.txt
```

Windows

The following example demonstrates how to call this command from Microsoft Windows.

```
approveWork -approvefile approval.txt
```

Output

These examples produce the following output.

```
--- Starting approval ---
[62145TS44X94HHYGW0PZK2CP0E1K9T1PR8Z42WEZ] Assignment successfully approved
for HIT 62145TS44X94HHYGW0PZ
--- Finished approval ---
  1 assignments approved.
  0 assignments failed to be approved.
```

Related Commands

- rejectWork (p. 37)

assignQualification

Description

The `assignQualification` command assigns a Qualification to a Worker without the Worker requesting the Qualification. There are two ways to use this operation:

- **Bulk Operation—** If you want to assign multiple Qualifications, use this command with a file that contains the Qualification Type ID and the list of Workers to be assigned the Qualfication.
- **Single Assignmentn—** You can specify the Qualification Type ID, the Worker, and the assigned Qualification score on the command line.

Arguments

The following table describes the arguments for the `assignQualification` command.

Name	Description	Required
-donotnotify	Specifies not to notify Workers when you have assigned them the Qualification. Example: `-donotnotify`	No
-help or *-h*	Displays the help for this operation. Example: `-help`	No
-input [filename]	Specifies the file that contains the Qualification ID you want to assign to Workers. This file contains a single Qualification ID. You can find this ID in the `.success` file generated after you create the Qualification with the `createQualificationType (p. 13)` operation. For information about this file, see *The Qualification ID file* in Files Used by the Command Line Tools (p. 53). Condition: Required if the *qualtypeid* argument is not specified. Example: `-input qualification.properties.success`	Conditional
-qualtypeid [Qualification Type ID]	Specifies the Qualification Type ID of the Qualification you want to assign to the Worker. This ID can be found in the `.success` file generated after you create the Qualification with the > operation. Condition: Required if the *input* argument is not specified. Example: `-qualtypeid RWFZTKZ55ZPZXN1C8TDZ`	Conditional
-sandbox	Runs this command in the Amazon Mechanical Turk sandbox for testing. This argument takes precedence even if you specify the production web site in your `mturk.properties` file. Example:`-sandbox`	No

Name	Description	Required
-score [integer]	Specifies the default Qualification score to assign to Workers. This argument can be used with both the bulk operation and the single assignment. If this argument is not specified, and the score has not been defined in the scorefile, a default score of `1` is used. Example: `-score 100`	No
-scorefile [filename]	Specifies a file which contains a tab delimited list of Worker IDs and optional Qualification scores to assign to each Worker. If the score is not specified, a default value of `1` is assigned. For information about this file, see *The score file* in Files Used by the Command Line Tools (p. 53). Condition: Required if the *-workerid* argument is not specified. Example: `-scorefile workerstoassign.txt`	Conditional
-workerid [Worker ID]	Specifies the ID of the Worker you want to assign the Qualification to. Condition: Required if the *scorefile* argument is not specified. Example: `-workerid A3C4G8DMXFG5PQ`	Conditional

Example

The following examples for Unix and Windows show how to use the `AssignQualification` command. The examples demonstrate a single assignment.

Unix

The following example demonstrates how to call this command from Unix.

```
./assignQualification.sh -qualtypeid RWFZTKZ55ZPZXN1C8TDZ -workerid
A3C4G8DMXFG5PQ -score 100
```

Windows

The following example demonstrates how to call this command from Microsoft Windows.

```
assignQualification -qualtypeid RWFZTKZ55ZPZXN1C8TDZ -workerid A3C4G8DMXFG5PQ
-score 100
```

Output

These examples produce the following output.

```
Assigned qualification RWFZTKZ55ZPZXN1C8TDZ to A3C4G8DMXFG5PQ with value 100
```

Related Commands

- createQualificationType (p. 13)

blockWorker

Description

The `blockWorker` command blocks a Worker from working on your HITs.

Arguments

The following table describes the arguments for the `blockWorker` command.

Name	Description	Required
-help or *-h*	Displays the help for this operation. Example: `-help`	No
-reason [string]	The reason why the Worker is being blocked. This reason is logged in our system for auditing purposes and can be used to determine if corrective action against the Worker is necessary. Enclose the reason string in quotation marks. Example: `-reason "After several warnings, the Worker continued to submit answers without reading the instructions carefully."`	Yes
-sandbox	Runs this command in the Amazon Mechanical Turk sandbox for testing. This argument takes precedence even if you specify the production web site in your `mturk.properties` file. Example: `-sandbox`	No
-workerid [Worker ID]	The ID of the Worker you want to block. Example: `-workerid A3C4G8DMXFG5PQ`	Yes

Example

The following examples for Unix and Windows show how to use the `blockWorker` command.

Unix

The following example demonstrates how to call this command from Unix. You must write this command on a single line. It is divided into multiple lines in this example for readability.

```
./blockWorker.sh -workerid A3C4G8DMXFG5PQ -reason "After several warnings, the
 Worker continued to
submit answers without reading the instructions carefully."
```

Windows

The following example demonstrates how to call this command from Microsoft Windows. You should write this command on a single line. It is divided into multiple lines in this example for readability.

```
blockWorker -workerid A3C4G8DMXFG5PQ -reason "After several warnings, the
Worker continued to
submit answers without reading the instructions carefully."
```

Output

These examples produce the following output.

```
Blocked A3C4G8DMXFG5PQ with reason: After several warnings, the Worker continued
 to submit answers without reading the instructions carefully.
```

Related Commands

- unblockWorker (p. 45)

createQualificationType

Description

The `createQualificationType` command creates a Qualification that can be used for your HITs. You can use the arguments to specify files that contain the Qualification test and the answers for the test. You can also create a Qualification that does not require a test. For more information about Qualifications, see the Amazon Mechanical Turk Developer Guide.

Arguments

The following table describes the arguments for the `createQualificationType` command.

Name	Description	Required
-answer [filename]	Specifies the file that contains the answers for the Qualification test. If this argument is omitted, you create a Qualification that does not require a test. For information about this file, see *The Qualification answer file* in Files Used by the Command Line Tools (p. 53). Example: `-answer qualification.answers`	No
-help or *-h*	Displays the help for this operation. Example: `-help`	No
-noretry	Specifies that Workers can only request the Qualification once. Example: `-noretry`	No
-properties [filename]	Specifies the Qualification properties file. For information about this file, see *The Qualification properties file* in Files Used by the Command Line Tools (p. 53). Example: `-properties qualification.properties`	Yes
-question [filename]	Specifies the Qualification question file that contains the Qualification test. For information about this file, see *The Qualification question file* in Files Used by the Command Line Tools (p. 53). Example:`-question qualification.question`	No
-sandbox	Runs this command in the Amazon Mechanical Turk sandbox for testing. This argument takes precedence even if you specify the production web site in your `mturk.properties` file. Example:`-sandbox`	No

Example

The following examples for Unix and Windows show how to use the `createQualificationType` command. The examples use the property file `qualification.properties` and the question file `qualification.question`. These examples create the qualification in the test environment.

Unix

The following example demonstrates how to call this command from Unix.

```
.createQualificationType.sh -properties qualification.properties -question
qualification.question -noretry -sandbox
```

Windows

The following example demonstrates how to call this command from Microsoft Windows.

```
createQualificationType -properties qualification.properties -question qualific
ation.question -noretry -sandbox
```

Output

If this command completes successfully, it creates a `.success` file with the name specified in the *`-properties`* argument. For this example, the file is named `qualification.properties.success`. This file contains the Qualification Type ID of the newly created Qualification. This command also produces output similar to the following.

```
Created qualification type: KYJ4GZZ5G3M6TDCEWYF0
You can take the test here:
http://workersandbox.mturk.com/mturk/requestqualification?qualifica
tionId=KYJ4GZZ5G3M6TDCEWYF0
```

Related Commands

- getQualificationRequests (p. 24)

deleteHITs

Description

The `deleteHITs` command deletes your HITs from Amazon Mechanical Turk. Use the arguments to specify how to handle assignments that have not been approved or are still available to Workers. You specify the HITs to delete in a file, which you generate as the output of getResults (p. 26).

Arguments

The following table describes the arguments for the `deleteHITs` command.

Name	Description	Required
`-approve`	Automatically approves any assignments that have been submitted and have not been approved or rejected. Assignments that are in the review or reviewing states cannot be deleted. Example: `-approve`	No
`-expire`	Automatically expires any HITs that are still available to Workers on Amazon Mechanical Turk. Live HITs cannot be deleted. Example: `-expire`	No
`-force`	Specifies *not* to prompt for manual confirmation before deleting the HITs. Only advanced developers should use this argument. Example: `-force`	No
`-help` or *`-h`*	Displays the help for this operation. Example:`-help`	No
`-sandbox`	Runs this command in the Amazon Mechanical Turk sandbox for testing. This argument takes precedence even if you specify the production web site in your `mturk.properties` file. Example:`-sandbox`	No
`-successfile [filename]`	Specifies the success file that contains the HITs to be deleted. For information about this file, see *The success file* in Files Used by the Command Line Tools (p. 53). Example: `-successfile helloworld.success`	Yes

Example

The following examples for Unix and Windows show how to use the `deleteHITs` command. This example deletes five assignments in the file `survey.success`. If any HITs in the file have been submitted, this example approves them. If any HITs are still available, this example expires them.

Windows

The following example demonstrates how to call this command from Microsoft Windows.

```
deleteHITs -successfile survey.success -approve -expire
```

Unix

The following example demonstrates how to call this command from Unix.

```
./deleteHITs.sh -successfile survey.success -approve -expire
```

Output

These examples produce the following output.

```
--- Starting to delete HITs ---
You are about to delete 5 HITs.
To confirm this operation, please press ENTER (or press Ctrl+C to abort):

[X1F6ZRZ8GW1ETS561XR0] Successfully deleted HIT (1/5)
[8X3GV2YWKAZZR9ZCEYB0] Successfully deleted HIT (2/5)
[YA3ZKSYWVWMWZ053TYKZ] Successfully deleted HIT (3/5)
[DYBZQP51T3VEKHXXFWR0] Successfully deleted HIT (4/5)
[F09PWRZ81WDEW5Z2FZJZ] Successfully deleted HIT (5/5)
--- Finished to delete HITs ---
  5 HITs have been deleted or were deleted previously.
  0 errors occurred.
```

evaluateQualificationRequests

Description

The `evaluateQualificationRequests` command evaluates the answers submitted by Workers so you can approve or reject the Qualification requests. You can use the *`preview`* argument to view the results before you approve them. If you run this command without the *`preview`* argument, the request is approved or rejected.

This command uses an answer key file to automatically review Workers' test results and determine if the Qualification request should be approved or rejected.

> **Tip**
> This allows you to specify free text answers that can be automatically processed. This overcomes a limitation of the Amazon Mechanical Turk system where FreeTextAnswer type answers cannot be auto-graded.

All questions must be answered correctly in order for the Qualification request to be approved. A Qualification score of 100 is assigned for approvals.

For more information about Qualifications, see the Amazon Mechanical Turk Developer Guide.

Arguments

The following table describes the arguments for the `evaluateQualificationRequests` command.

Name	Description	Required
`-answers [filename]`	Specifies the file that contains the answer key. For information about this file, see *The Qualification answer key file* in Files Used by the Command Line Tools (p. 53). Example: `-answers qualification.answerkey`	Yes
`-help` or *`-h`*	Displays the help for this operation. Example: `-help`	No
`-input [filename]`	Specifies the file that contains the Qualification ID you want to assign to Workers. This file contains a single Qualification ID. This ID is found in a `.success` file generated after you create the Qualification with the createQualificationType (p. 13) operation. For information about this file, see *The Qualification ID file* in Files Used by the Command Line Tools (p. 53). Condition: Required if the *`qualtypeid`* argument is not specified. Example: `-input qualification.properties.success`	Conditional
`-preview`	Specifies that you want to preview the outcome of the Qualification request evaluations before submitting the decisions to Amazon Mechanical Turk. Example: `-preview`	No

Name	Description	Required
-qualtypeid [Qualification Type ID]	Specifies the Qualification Type ID for which to evaluate the Qualification requests. Condition: Required if the *input* argument is not specified. Example: `-qualtypeid RWFZTKZ55ZPZXN1C8TDZ`	Conditional
-sandbox	Runs this command in the Amazon Mechanical Turk sandbox for testing. This argument takes precedence even if you specify the production web site in your `mturk.properties` file. Example:`-sandbox`	No

Example

The following examples for Unix and Windows show how to use the `evaluateQualificationRequests` command. These examples preview the results.

Unix

The following example demonstrates how to call this command from Unix.

```
./evaluateQualificationRequests.sh -answers qualification.answerkey -qualtypeid
 RWFZTKZ55ZPZXN1C8TDZ -preview
```

Windows

The following example demonstrates how to call this command from Microsoft Windows.

```
evaluateQualificationRequests -answers qualification.answerkey -qualtypeid RW
FZTKZ55ZPZXN1C8TDZ -preview
```

Output

These examples produce the following output.

```
Preview flag is set. Qualification requests will not be approved or rejected.
---[Worker A3C4G8DMFSG5PQ]---------------------------------------------------
----------
Question question1:CORRECT [The answer key 'false' matches the given answer
'false']
Question question2:CORRECT [The answer key 'true' matches the given answer
'true']
Question question3:CORRECT [The answer key '1' matches the given answer '1']
Worker A3C4G8DMFSG5PQ has PASSED your test and scored 100
----------------------------------------------------------------------------
```

```
- - - - - - - - -
```

Related Commands

- getQualificationRequests (p. 24)

extendHITs

Description

The `extendHITs` command extends the expiration date or increases the maximum number of assignments for all HITs in the specified *`-successfile`*. If you extend the expiration date, and the HIT has not expired, the new expiration date is the existing date plus the amount of time specified. If the HIT has already expired, the new expiration date is the current time plus the amount of time specified. If you add additional assignments, you must be sure that you have enough funds to pay for the assignments.

Arguments

The following table describes the arguments for the `extendHITs` command.

Name	Description	Required
`-assignments [integer]`	The number of assignments to add to the HITs. Example: `-assignments 12`	No
`-help` or *`-h`*	Displays the help for this operation. Example: `-help`	No
`-hours [integer]`	The amount of time, in hours, by which to extend the expiration date of the HITs. Example: `-hours 12`	No
`-sandbox`	Runs this command in the Amazon Mechanical Turk sandbox. This argument takes precedence even if you specify the production web site in your `mturk.properties` file. Example:`-sandbox`	No
`-successfile [filename]`	The path to the `.success` file that contains the HITs to extend. This is the file that loadHITs (p. 30) returns. For information about this file, see *The success file* in Files Used by the Command Line Tools (p. 53). Example:`-successfile ..\mysurvey\mysurvey.success`	Yes

Example

The following examples for Unix and Windows show how to use the `extendHITs` command. These examples add four assignments and three hours to the five HITs in the `.success` file.

Unix

The following example demonstrates how to call this command from Unix.

```
./extendHITs.sh -successfile ..\mysurvey\mysurvey.success -assignments 4 -hours
```

```
3
```

Windows

The following example demonstrates how to call this command from Microsoft Windows.

```
extendHITs -successfile ..\mysurvey\mysurvey.success -assignments 4 -hours 3
```

Output

This example produces output similar to the following.

```
--- Starting to extend HITs ---
[0YFZ2TYJF3HZPGZV4Z40] Successfully extended HIT (1/5)
[4GMZSHZKKK9WT9M9XWA0] Successfully extended HIT (2/5)
[XYTZY0YK1W2ZTCZM9Z80] Successfully extended HIT (3/5)
[RZGZZ4Z6GXKTV5DX81B0] Successfully extended HIT (4/5)
[M26ZN61JMT9E4MG0M94Z] Successfully extended HIT (5/5)
--- Finished to extend HITs ---
  5 HITs have been extended (added 4 assignment(s), 3 hour(s))
  0 HITs failed to be extended.
```

getBalance

Description

The `getBalance` command retrieves the available balance in your Amazon Mechanical Turk account. This amount is your current balance minus any outstanding payments, fees, or bonuses you owe.

Arguments

The following table describes the arguments for the `getBalance` command.

Name	Description	Required
-help or *-h*	Displays the help for this operation. Example: `-help`	No
-sandbox	Runs this command in the Amazon Mechanical Turk sandbox and gets your sandbox account balance. This amount is always $10000.00. This argument takes precedence even if you specify the production web site in your `mturk.properties` file. Example:`-sandbox`	No

Example

The following examples for Unix and Windows show how to use the `getBalance` command.

Unix

The following example demonstrates how to call this command from Unix.

```
./getBalance.sh
```

Windows

The following example demonstrates how to call this command from Microsoft Windows.

```
getBalance
```

Output

This example produces output similar to the following.

```
Your account balance: $819.45
```

getQualificationRequests

Description

The `getQualificationRequests` command retrieves the Qualification requests from Workers for your Qualifications. For more information about Qualifications and Qualification requests see the Amazon Mechanical Turk Developer Guide.

Arguments

The following table describes the arguments for the `getQualificationRequests` command.

Name	Description	Required
`-help` or *`-h`*	Displays the help for this operation. Example: `-help`	No
`-outputfile [filename]`	Specifies the output file. This is a tab delimited text file that contains the details of Workers' Qualification. You can then approve or reject the Qualification Requests by using approveQualificationRequests (p. 3) or rejectQualificationRequests (p. 35) command. For information about this file, see *The Qualification request file* in Files Used by the Command Line Tools (p. 53). Example: `-outputfile qualification_requests.txt`	Yes
`-qualtypeid [Qualification Type ID]`	Specifies the Qualification Type ID of the Qualification you want to retrieve requests for. This ID is in a `.success` file generated after you create the Qualification with the `createQualificationType (p. 13)` command. If this ID is not specified, all Qualification requests for Qualifications you own are included in the output file. Example:`-qualtypeid RWFZTKZ55ZPZXN1C8TDZ`	No
`-sandbox`	Runs this command in the Amazon Mechanical Turk sandbox for testing. This argument takes precedence even if you specify the production web site in your `mturk.properties` file. Example:`-sandbox`	No

Example

The following examples for Unix and Windows show how to use the `getQualificationRequests` command. These examples write the requests for all Qualification types to the file `qualrequests.txt`.

Unix

The following example demonstrates how to call this command from Unix:

```
./getQualificationRequests.sh -outputfile qualrequests.txt
```

Windows

The following example demonstrates how to call this command from Microsoft Windows:

```
getQualificationRequests -outputfile qualrequests.txt
```

Output

These examples retrieved four Qualfication requests and produced the following output.

```
Retrieved 4 Qualification Requests
Answers successfully saved to file: qualrequests.txt
```

Related Commands

- evaluateQualificationRequests (p. 17)
- approveQualificationRequests (p. 3)
- rejectQualificationRequests (p. 35)

getResults

Description

The `getResults` command retrieves the results of HITs submitted to Amazon Mechanical Turk. You must supply the name of a file that contains the IDs of the HITs. You can get this file from a call to loadHITs (p. 30).

Arguments

The following table describes the arguments for the `getResults` command.

Name	Description	Required
`-help` or *`-h`*	Displays the help for this operation. Example: `-help`	No
`-namevaluepairs`	Outputs the results as name-value pairs instead of column format. Example: `-namevaluepairs`	No
`-outputfile [filename]`	Specifies the file in which the results are saved. For information about this file, see *The output file* in Files Used by the Command Line Tools (p. 53). Example: `-outputfile helloworld.results`	Yes
`-sandbox`	Runs this command in the Amazon Mechanical Turk sandbox for testing. This argument takes precedence even if you specify the production web site in your `mturk.properties` file. Example:`-sandbox`	No
`-successfile [filename]`	Specifies the path and name of the `.success` file that loadHITs (p. 30) returns. For information about this file, see *The success file* in Files Used by the Command Line Tools (p. 53). Example: `-successfile helloworld.success`	Yes

This command creates an output file that contains the results of submitted HITs. The file contains all data related to the submitted HITs and assignments. Each HIT in this file has a link to your **Manage HITs** page on the Requester website. Use these links to manually reject assignments, pay bonuses, or send emails to Workers.

Example

The following examples for Unix and Windows show how to use the `getResults` command.

Unix

The following example demonstrates how to call this command from Unix.

```
./getResults.sh -successfile survey.success -outputfile survey.results
```

Windows

The following example demonstrates how to call this command from Microsoft Windows.

```
getResults -successfile survey.success -outputfile survey.results
```

Output

This example writes the results to the survey.results file and produces output similar to the following:

```
--[Retrieving Results]----------
Retrieved HIT 1/10, 832TY7YE5HKWS1H10YR0
Retrieved HIT 2/10, EW2ZHA4R3R7Z4WY5XZAZ
Retrieved HIT 3/10, 5YJ0T51KASD63A4J5YW0
Retrieved HIT 4/10, 0X5PSKYBVXNZPXZHCY0Z
Retrieved HIT 5/10, FWDE79ST7Y6A025QVXHZ
Retrieved HIT 6/10, X3YPN7HBFRXJ1KYMPGWZ
Retrieved HIT 7/10, YZJEZWZ3QW6Z28D9DG90
Retrieved HIT 8/10, 2J5Z5MZ91A06JFGRYYGZ
Retrieved HIT 9/10, PYMPVMRPPWVZTJN9RXE0
Retrieved HIT 10/10, N9PZZ0YW0Y9ZZQZPQH3Z
--[Done Retrieving Results]----------

Results have been written to file 'survey.results'.

Assignments completed: 30/30 (100%)
         Time elapsed: 0:05:16 (h:mm:ss)
  Average submit time: 13.7 seconds
```

Related Commands

- approveWork (p. 5)
- rejectWork (p. 37)

grantBonus

Description

The `grantBonus` command issues a payment from your account to a Worker. This payment happens separately from the reward you pay to the Worker when you approve the Worker's assignment. You must have enough funds in your account to pay for the bonus.

Arguments

The following table describes the arguments for the `grantBonus` command.

Name	Description	Required
-amount [decimal]	The amount of bonus to give the Worker. Example: `-amount 10.50`	Yes
-assignment [assignment ID]	The ID of the assignment associated with this bonus. Example: `-assignment ZYJZWSCAT0DZRFY5KYP00S0ZS8Y5H0NZR9YAMY1Z`	Yes
-help or *-h*	Displays the help for this operation. Example: `-help`	No
-reason [text]	The reason for the bonus. You must enclose the string in quotation marks. Example: `-reason "You did the most work last week."`	Yes
-sandbox	Runs this command in the Amazon Mechanical Turk sandbox. This argument takes precedence even if you specify the production web site in your `mturk.properties` file. Example:`-sandbox`	No
-workerid [workerID]	The ID of the Worker who gets the bonus. Example: `-workerid A3C4G8DMXFG5PQ`	Yes

Example

The following examples for Unix and Windows show how to use the `grantBonus` command.

Unix

The following example demonstrates how to call this command from Unix.

```
./grantBonus.sh -workerid A3C4G8DMXFG5PQ -assignment 0YFZ2TYJF3HZPGZV4Z40EZD4YZZF
DSTZ0YG78W2Z -reason "Your answers are very accurate." -amount 5.00
```

Windows

The following example demonstrates how to call this command from Microsoft Windows.

```
grantBonus -workerid A3C4G8DMXFG5PQ -assignment 0YFZ2TYJF3HZPGZV4Z40EZD4YZZFD
STZ0YG78W2Z-reason "Your answers are very accurate." -amount 5.00
```

Output

These examples produce output similar to the following.

```
Granted bonus to A3C4G8DMXFG5PQ
```

loadHITs

Description

The `loadHITs` command loads HITs into Amazon Mechanical Turk. Before you can use this command, you must create an `input` file, a `question` file, and a `properties` file. If you are loading a single HIT, your `question` file can contain the question information. The `question` file is an XML file that conforms to the QuestionForm Data Structure. If you are loading multiple HITs, you should create a question template file. This file has the same XML format as the `question` file, but it contains placeholders for the fields defined in the `input` file. For each row in the `input` file, the system inserts the the fields into the placeholders of the template file. The number of rows in the `input` file determines how many HITs are created. For more information about these files, see Files Used by the Command Line Tools (p. 53).

If the HITs load successfully, this operation creates a `.success` file with the same name as your `.input` file. That is, if your input file is called `MyHITs.input`, then the `.success` file will be `MyHITs.success`. If the HITs do not load successfully, this operation creates a `.failure` file with the same name as your `.input` file. If you want to call this command multiple times with the same input file, you can use the *`label`* argument to change the name of the created file. For more information about the `.success` file and the `.failure` file see Files Used by the Command Line Tools (p. 53).

Arguments

The following table describes the arguments for the `loadHITs` command.

Name	Description	Required
`-help` or *`-h`*	Displays the help for this operation Example: `-help`	No
`-input [file name]`	Specifies the input file. For information about this file, see *The input file* in Files Used by the Command Line Tools (p. 53). Example: `-input helloworld.input`	Yes
`-label [filename]`	Specifies the name to use for the `.success` and the `.failure` files. Example: `-label survey_2`	No
`-maxhits`	Specifies the maximum number of HITs to create. Use this for testing if you do not want to load the entire input file. Example: `-maxhits 1`	No
`-preview`	Creates an HTML preview of the HIT instead of loading it into Amazon Mechanical Turk. The name of the file is specified by the argument. If the *`-previewfile`* argument is not specified, this operation creates a file named `preview.html` in the current directory. Example: `-preview`	No
`-previewfile [file name]`	Specifies the name of the html preview file. This argument is used only if the *`preview`* argument is specified. Example: `-previewfile helloworld.html`	No

Name	Description	Required
-properties [file name]	Specifies the HIT properties file. For information about this file, see *The HIT properties file* in Files Used by the Command Line Tools (p. 53). Example: `-properties helloworld.properties`	Yes
-question [file name]	Specifies the question file. For information about this file, see *The question file* in Files Used by the Command Line Tools (p. 53). Example: `-question helloworld.question`	Yes
-sandbox	Runs this command in the Amazon Mechanical Turk sandbox for testing. This argument takes precedence even if you specify the production web site in your `mturk.properties` file. Example:`-sandbox`	No

Example

The following examples for Unix and Windows show how to use the `loadHITs` command. This example loads five HITs using the information found in the `survey.input`, `survey.question`, and `survey.properties` files found in the directory `..\survey`.

Unix

The following example demonstrates how to call this command from Unix.

```
./loadHITs.sh -input ..\survey\survey.input -question ..\survey\survey.question
 -properties ..\survey\survey.properties
```

Windows

The following example demonstrates how to call this command from Microsoft Windows.

```
loadHITs -input ..\survey\survey.input -question ..\survey\survey.question -
properties ..\survey\survey.properties
```

Output

These examples create a `.success` file with the same path and name as the file specified by the`-`*`input`* argument. In these examples, the returned file is `survey.sucess` in the `..\survey` directory. The command produces the following output.

```
--[Initializing]----------
 Input: ..\survey\survey.input
 Properties: ..\survey\survey.properties
 Question File: ..\survey\survey.question
 Preview mode disabled
--[Loading HITs]----------
  Start time: Fri Dec 14 16:23:28 PST 2007
Created HIT 1: HITId=GX9P62Y4EXAZSZYRQ8EZ
Created HIT 2: HITId=2WAZRKZHC27ZV25HYXD0
Created HIT 3: HITId=R8TZ2QYQKXMZ6X1NYKN0
Created HIT 4: HITId=TA6CVRDAQBWPTNN013W0
Created HIT 5: HITId=R0AZ1KZSVWFZS75QZRF0

You may see your HIT(s) with HITTypeId 'YTXZ13ZWEYFZRBZ4V120' here:

  http://www.mturk.com/mturk/preview?groupId=YTXZ13ZWEYFZRBZ4V120

  End time: Fri Dec 14 16:23:32 PST 2007
--[Done Loading HITs]----------
  Total load time: 4 seconds.
  Successfully loaded 5 HITs.
```

Related Commands

- getResults (p. 26)

makeTemplate

Description

The makeTemplate command creates a copy of a sample application. After you run the sample applications that the Amazon Mechanical Turk Command Line Tools provide, you might want to create your own application, probably one that is similar to one of the sample applications. The makeTemplate command makes it easy for you to replicate one of the samples in a different directory.

You can make a template of a HIT-generating sample, such as the Hello World sample or the Best Image sample. For these templates, run the makeTemplate command from the [Command Line Tools installation directory]\hits directory.

You can also make templates of Qualification-generating samples, such as the Assign Qualification sample. For these templates, run the makeTemplate from the Command Line Tools installation directory]\qualifications directory.

Arguments

The following table describes the arguments for the makeTemplate command.

Name	Description	Required
-help or *-h*	Displays the help for this operation. Example: -help	No
-target [directory name]	Specifies the name of the directory to create. This is the directory where the new files will be copied. This name is also used for the files that are created in this directory. Example: -target weathersurvey	Yes
-template [directory name]	The name of the sample directory you are making the template from. Example:-template image_category	Yes

When you run this command for HIT-type samples, the Command Line Tools creates the new directory and copies the .input, .question, and .properties files from the specified sample. It creates these files with the name specified in the *-target* option. The Command Line Tools also creates new run.cmd, getResults.cmd, and approveAndDeleteResults.cmd files.

When you run this command for qualification-type samples, the Command Line Tools creates the new directory and copies the .answer,.question, and .properties files from the specified sample. It creates these files with the name specified in the *-target* option. The Command Line Tools also creates new createQualification.cmd, updateQualification.cmd, and deactivateQualification.cmd files.

Example

The following examples for Unix and Windows show how to use the makeTemplate command.

Unix

The following example demonstrates how to call this command from Unix.

```
./makeTemplate.sh -target my_survey -template best_image
```

Windows

The following example demonstrates how to call this command from Microsoft Windows.

```
makeTemplate -target my_survey -template best_image
```

Output

These examples produce the following output.

```
Copying resource file: ..\hits\my_survey\my_survey.question
Copying resource file: ..\hits\my_survey\my_survey.properties
Copying resource file: ..\hits\my_survey\my_survey.input
Generating script: ..\hits\my_survey\run.cmd
Generating script: ..\hits\my_survey\getResults.cmd
Generating script: ..\hits\my_survey\approveAndDeleteResults.cmd
```

rejectQualificationRequests

Description

The `rejectQualificationRequests` command rejects Workers' Qualification requests.

Arguments

The following table describes the arguments for the `rejectQualificationRequests` command.

Name	Description	Required
-force	Specifies *not* to prompt for manual confirmation before performing the reject operation. Only advanced developers should use this argument. Example: `-force`	No
-help or *-h*	Displays the help for this operation. Example: `-help`	No
-qualRequest [Qualification request ID]	Specifies the Qualification request ID(s) to reject. You can specify multiple Qualification request IDs in a comma-separated list. The list (or single ID) must be enclosed in quotation marks. Conditions: Required if the *rejectfile* argument is not specified. Example: `-qualRequest "789RVWYBAZW00EXAMPLE951RVWYBAZW00EXAMPLE"`	Conditional
-rejectfile [filename]	Specifies a file that contains the list of Qualification requests to reject. For information about this file, see *The Qualification reject file* in Files Used by the Command Line Tools (p. 53). Conditions: Required if the *qualRequest* argument is not specified. Example: `-rejectfile qualification_requests_toreject.txt`	Conditional
-sandbox	Runs this command in the Amazon Mechanical Turk sandbox for testing. This argument takes precedence even if you specify the production web site in your `mturk.properties` file. Example:`-sandbox`	No

Example

The following examples for Unix and Windows show how to use the `rejectQualificationRequests` command. These examples reject one Qualification request listed in the file `toreject.txt`.

Unix

The following example demonstrates how to call this command from Unix.

```
./rejectQualificationRequests.sh -rejectfile toreject.txt
```

Windows

The following example demonstrates how to call this command from Microsoft Windows.

```
rejectQualificationRequests -rejectfile toreject.txt
```

Output

These examples produce the following output.

```
You are about to reject 1 qualification request(s). Are you sure?
To confirm this operation, please press ENTER (or press Ctrl+C to abort):

If you would like to supply a comment to the worker(s), please type it below
then press ENTER.
If not, just hit ENTER:

[789RVWYBAZW00EXAMPLE951RVWYBAZW00EXAMPLE] Qualification request successfully
rejected
```

rejectWork

Description

The rejectWork command rejects assignments submitted by Workers. You can reject single assignments, or you can specify a file that contains the assignments to reject.

Arguments

The following table describes the arguments for the rejectWork command.

Name	Description	Required
-assignment [assignment IDs]	Specifies the assignment ID to reject. For multiple assignments, separate each assignment ID with a comma. Condition: Required if the *rejectfile* argument is not specified. Example: -assignment 0YFZ2TYJF3HZPGZV4Z40EZD4YZZFDSTZ0YG78W2Z	Conditional
-force	Specifies *not* to prompt for manual confirmation before performing the reject operation. Only advanced developers should use this argument. Example: -force	No
-help or *-h*	Displays the help for this operation. Example: -help	No
-rejectfile [filename]	Specifies a text file that contains a list of assignment IDs and optional rejection comments. For information about this file, see *The reject file* in Files Used by the Command Line Tools (p. 53). Condition: Required if the *assignment* argument is not specified. Example: -rejectfile helloworld_reject.txt	Conditional
-sandbox	Runs this command in the Amazon Mechanical Turk sandbox for testing. This argument takes precedence even if you specify the production web site in your mturk.properties file. Example:-sandbox	No

Example

The following examples for Unix and Windows show how to use the rejectWork command. These examples reject the specified assignment with no additional comments.

Unix

The following example demonstrates how to call this command from Unix.

```
./rejectWork.sh -assignment 0YFZ2TYJF3HZPGZV4Z40EZD4YZZFDSTZ0YG78W2Z
```

Windows

The following example demonstrates how to call this command from Microsoft Windows.

```
rejectWork -assignment 0YFZ2TYJF3HZPGZV4Z40EZD4YZZFDSTZ0YG78W2Z
```

Output

These examples produce the following output.

```
You are about to reject 1 assignment(s).
To confirm this operation, please press ENTER (or press Ctrl+C to abort):

If you would like to supply a comment to the worker(s), please type it below
then press ENTER.
If not, just hit ENTER:

[0YFZ2TYJF3HZPGZV4Z40EZD4YZZFDSTZ0YG78W2Z] Assignment successfully rejected
with comment ()
```

resetAccount

Description

The `resetAccount` command deletes all of your existing HITs from Amazon Amazon Mechanical Turk. If Workers are still working on a HIT, it is not deleted. Amazon Mechanical Turk approves any assignments that are submitted but not yet approved.

arguments

The following table describes the arguments for the `resetAccount` command.

Name	Description	Required
-force	Specifies *not* to prompt for manual confirmation before performing the operation. Only advanced developers should use this argument. Example: `-force`	No
-help or *-h*	Displays the help for this operation. Example: `-help`	No
-sandbox	Runs this command in the Amazon Mechanical Turk sandbox. This argument takes precedence even if you specify the production web site in your `mturk.properties` file. Example:`-sandbox`	No

Example

The following examples for Unix and Windows show how to use the `resetAccount` command.

Unix

The following example demonstrates how to call this command from Unix.

```
./resetAccount.sh
```

Windows

The following example demonstrates how to call this command from Microsoft Windows.

```
resetAccount
```

Output

These examples produce output similar to the following.

```
PLEASE READ THIS CAREFULLY

You are about to delete ALL your HITs from Amazon Mechanical Turk.
To confirm this operation, please press ENTER (or press Ctrl+C to abort):

--- Starting to reset account (10 HITs) ---
[2AQZM8YF9BTZYSHPKX30] Successfully deleted HIT (1/10)
[4AQZPBYWF1NZNBGZMGZZ] Successfully deleted HIT (2/10)
[2XAZ439GBWGZ57M0KX80] Successfully deleted HIT (3/10)
[4SPZGWZ41XDZS9YGTXM0] Successfully deleted HIT (4/10)
[4YKZ528SWH0TQTZ0JRPZ] Successfully deleted HIT (5/10)
[4WYZWAYD7AYJSCYT6J70] Successfully deleted HIT (6/10)
[38QGWV8SW2SZ3TY0KY5Z] Successfully deleted HIT (7/10)
[4XTZGSX1PR1ZGRSTCY0Z] Successfully deleted HIT (8/10)
[5WGMVNYC9RC6Q8W8K9WZ] Successfully deleted HIT (9/10)
--- Finished to reset account ---
  9 HITs have been deleted or were deleted previously.
  0 HITs failed to delete.
  1 HITs could not be deleted because they are currently being worked on.
  1 HITs are left in your account
```

reviewResults

Description

The `reviewResults` command allows you to approve or reject multiple assignments from a file. You use the output file from getResults (p. 26) to specify which assignments to reject. All other assignments are approved.

To use the reviewResults command

1. Run getResults (p. 26).
2. Open the output file.
3. For each assignment you want to reject, type any character in the "reject" column for that assignment.
4. Specify the file in the *`-resultsfile`* argument for the `reviewResults` command.
5. Run the `reviewResults` command.

Important
Any mark in the "reject" column for an assignment causes the assignment to be rejected. Any unmarked assignments are accepted.

Arguments

The following table describes the arguments for the `reviewResults` command.

Name	Description	Required
`-help` or *`-h`*	Displays the help for this operation. Example: `-help`	No
`-resultsfile [filename]`	Specifies the output file from getResults (p. 26) in which you have marked the assignments to reject. All unmarked assignments are approved. Example: `-resultsfile helloworld.results`	Yes
`-sandbox`	Runs this command in the Amazon Mechanical Turk sandbox for testing. This argument takes precedence even if you specify the production web site in your `mturk.properties` file. Example:`-sandbox`	No

Example

The following examples for Unix and Windows show how to use the `reviewResults` command.

Unix

The following example demonstrates how to call this command from Unix.

```
./reviewResults.sh  -resultsfile survey.results
```

Windows

The following example demonstrates how to call this command from Microsoft Windows.

```
reviewResults -resultsfile survey.results
```

Output

This example produces output similar to the following:

```
[BWZZVVYNNYKZ6QZT4V00QHRZNS93BANZ7ZZ2AW50] Assignment successfully approved
[5W48WR9T5X2ZYVZYNS7Z92QZ2VYWNZ1EYSWY3X4Z] Assignment successfully approved
[JYVZ4GXZ0XYZNEZ29350XXX0VR986J4YP6ZH7WZ0] Assignment successfully approved
[M0ZZXFC0J8KP56YVNWM0A0KZSCYA3YFZPJRNGR8Z] Assignment successfully approved
[1Z0ZHS5GGYQWWA83XVE05KZ4RSY233CZ0G91XAJZ] Assignment successfully approved

Assignments approved: 5/5 (100%)
Assignments rejected: 0/5 (0%)
Assignments failed: 0/5 (0%)
```

Related Commands

- approveWork (p. 5)
- rejectWork (p. 37)

revokeQualification

Description

The revokeQualification command revokes a Qualification from a Worker.

Arguments

The following table describes the arguments for the revokeQualification command.

Name	Description	Required
-help or *-h*	Displays the help for this operation. Example: -help	No
-qualtypeid [Qualification Type ID]	The Qualification Type ID of the Qualification to revoke from the Worker. Example: -qualtypeid RWFZTKZ55ZPZXN1C8TDZ	Yes
-reason [text]	The reason you are revoking the Qualification. The text must be enclosed in quotation marks. Example: -reason "I'm sorry but your work quality is below acceptable standards."	No
-sandbox	Runs this command in the Amazon Mechanical Turk sandbox for testing. This argument takes precedence even if you specify the production web site in your mturk.properties file. Example:-sandbox	No
-workerid [Worker ID]	The Worker ID of the Worker from whom to revoke the Qualification. Example: -workerid A3C4G8DMXFG5PQ	Yes

Example

The following examples for Unix and Windows show how to use the revokeQualification command. These examples revoke a Worker's qualification and provide a reason.

Unix

The following example demonstrates how to call this command from Unix.

```
./revokeQualification.sh -qualtypeid RWFZTKZ55ZPZXN1C8TDZ -workerid
A3C4G8DMXFG5PQ -reason "I'm sorry but your work quality is below acceptable
standards."
```

Windows

The following example demonstrates how to call this command from Microsoft Windows.

```
revokeQualification -qualtypeid RWFZTKZ55ZPZXN1C8TDZ -workerid A3C4G8DMXFG5PQ
-reason "I'm sorry but your work quality is below acceptable standards."
```

Output

These examples produce the following output.

```
Revoked qual RWFZTKZ55ZPZXN1C8TDZ from A3C4G8DMXFG5PQ with reason:
I'm sorry but your work quality is below acceptable standards.
```

unblockWorker

Description

The `unblockWorker` command unblocks a Worker who has been blocked from working on your HITs.

Arguments

The following table describes the arguments for the `unblockWorker` command.

Name	Description	Required
-help or *-h*	Displays the help for this operation. Example: `-help`	No
-reason	The reason why you are unblocking the Worker. This reason is logged in our system for auditing purposes. Enclose this string in quotation marks. Example: `-reason "Made a mistake. Blocked the wrong Worker ID"`	Yes
-sandbox	Runs this command in the Amazon Mechanical Turk sandbox for testing. This argument takes precedence even if you specify the production web site in your `mturk.properties` file. Example: `-sandbox`	No
-workerid [worker ID]	The ID of the Worker to unblock. Example: `-workerid A3C4G8DMXFG5PQ`	Yes

Example

The following examples for Unix and Windows show how to use the `unblockWorker` command. These examples unblock a specified Worker.

Unix

The following example demonstrates how to call this command from Unix.

```
./unblockWorker.sh -workerid A3C4G8DMXFG5PQ -reason "Made a mistake. Blocked
the wrong Worker ID."
```

Windows

The following example demonstrates how to call this command from Microsoft Windows.

```
unblockWorker -workerid A3C4G8DMXFG5PQ -reason "Made a mistake. Blocked the
```

```
wrong Worker ID."
```

Output

These examples produce the following output.

```
Unblocked A3C4G8DMXFG5PQ with reason:  Made a mistake. Blocked the wrong Worker
 ID.
```

updateHITs

Description

The `updateHITs` command updates properties of HITs that are already on Amazon Mechanical Turk.

All properties of a HIT can be modified:

- Title
- Description
- Keywords
- AssignmentDurationInSeconds
- AutoApprovalDelayInSeconds
- Qualification requirements

Important
The Reward property can only be updated for HITs that do not have any assignments that have been accepted, submitted, approved or rejected.

Arguments

The following table describes the arguments for the `updateHITs` command.

Name	Description	Required
`-properties [filename]`	The HIT properties file that contains the new properties for the HITs. The HITs are updated with all the values defined in this file. For information about this file, see *The HIT properties file* in Files Used by the Command Line Tools (p. 53). Example: `-properties helloworld.properties`	Yes
`-help` or *`-h`*	Displays the help for this operation. Example: `-help`	No
`-sandbox`	Runs this command in the Amazon Mechanical Turk sandbox for testing. This argument takes precedence even if you specify the production web site in your `mturk.properties` file. Example:`-sandbox`	No
`-success [filename]`	Specifies the success file that contains the IDs of the HITs to be updated. For information about this file, see *The success file* in Files Used by the Command Line Tools (p. 53). Example: `-success helloworld.success`	Yes

Example

The following examples for Unix and Windows show how to use the `updateHITs` command. These examples update four HITs in the file `..\survey\survey.success` with the new properties in the file `..\survey\survey.properties`

Unix

The following example demonstrates how to call this command from Unix.

```
./updateHITs.sh -success ..\survey\survey.success -properties ..\survey\sur
vey.properties
```

Windows

The following example demonstrates how to call this command from Microsoft Windows.

```
updateHITs -success ..\survey\survey.success -properties ..\survey\survey.prop
erties
```

Output

These examples produce the following output.

```
--[Initializing]----------
 Success File: ..\survey\survey.success
 Properties: ..\survey\survey.properties
--[Updating HITs]----------
  Start time: Fri Dec 14 16:48:03 PST 2007
  Input: 4 hitids
  New HITTypeId: SWZZPTZ7Y14ZY8WXNZH0
Updated HIT #0 (4GMZSHZKKK9WT9M9XWA0) to new HITTypeId SWZZPTZ7Y14ZY8WXNZH0
Updated HIT #1 (XYTZY0YK1W2ZTCZM9Z80) to new HITTypeId SWZZPTZ7Y14ZY8WXNZH0
Updated HIT #2 (RZGZZ4Z6GXKTV5DX81B0) to new HITTypeId SWZZPTZ7Y14ZY8WXNZH0
Updated HIT #3 (M26ZN61JMT9E4MG0M94Z) to new HITTypeId SWZZPTZ7Y14ZY8WXNZH0
  End time: Fri Dec 14 16:48:05 PST 2007
--[Done Updating HITs]----------
4 HITS were processed
4 HITS were updated
  Total load time: 2 seconds.
```

updateQualificationScore

Description

The `updateQualificationScore` command updates the Qualification scores for Workers.

Arguments

The following table describes the arguments for the `updateQualificationScores` command.

Name	Description	Required
`-help` or *`-h`*	Displays the help for this operation. Example: `-help`	No
`-input [filename]`	Specifies the file that contains the Worker IDs and desired Qualification scores to assign. For information about this file, see *The Worker ID file* in Files Used by the Command Line Tools (p. 53). Condition: Required if the *`workerid`* argument is not specified. Example: `-input worker_qual_scores.txt`	Conditional
`-qualtypeid [Qualification Type ID`	The Qualification Type ID to update the Qualification scores for. Example: `-qualtypeid RWFZTKZ55ZPZXN1C8TDZ`	Yes
`-sandbox`	Runs this command in the Amazon Mechanical Turk sandbox for testing. This argument takes precedence even if you specify the production web site in your `mturk.properties` file. Example:`-sandbox`	No
`-score [integer value]`	The score to assign to each Worker for the specified Qualification Type ID. Condition: Required if the score is not defined in the *`input`* file Example: `-score 50`	Conditional
`-workerid [Worker ID]`	The ID of the Worker you want to assign the score to. Conditions: Required if the *`input`* argument is not specified. Example: `-workerid A3C4G8DMXFG5PQ`	Conditional

Example

The following examples for Unix and Windows show how to use the `updateQualificationScore` command. These examples update the score for one Worker.

Unix

The following example demonstrates how to call this command from Unix.

```
./updateQualificationScore.sh -qualtypeid RWFZTKZ55ZPZXN1C8TDZ score 50 -workerid
 A3C4G8DMXFG5
```

Windows

The following example demonstrates how to call this command from Microsoft Windows.

```
updateQualificationScore -qualtypeid RWFZTKZ55ZPZXN1C8TDZ score 50 -workerid
A3C4G8DMXFG5
```

Output

These examples produce the following output.

```
Successfully updated A3C4G8DMXFG5 score to 50
```

updateQualificationType

Description

The `updateQualificationType` command updates the properties of your Qualification.

Arguments

The following table describes the arguments for the `updateQualificationType` command.

Name	Description	Required
-answer [filename]	Specifies the Qualification Answer file that contains the new AnswerKey XML for the Qualification test. This file allows the Amazon Mechanical Turk system to automatically evaluate and score a Qualification request. For information about this file, see *The Qualification answer file* in Files Used by the Command Line Tools (p. 53). Example: `-answer qualification.answer`	No
-help or *-h*	Displays the help for this operation. Example: `-help`	No
-properties [filename]	Specifies the Qualification properties file that contains the new properties for the Qualification. The Qualification is updated with all the properties defined in the file. For information about this file, see *The Qualification properties file* in Files Used by the Command Line Tools (p. 53). **Note** You cannot modify the title of a Qualification. Example: `-properties qualification.properties`	No
-qualtypeid	The Qualification Type ID of the Qualification you want to update. This ID is in the `.success`file generated after you create the Qualification with the `createQualificationType (p. 13)` command. Example: `-qualtypeid RWFZTKZ55ZPZXN1C8TDZ`	Yes
-question [filename]	Specifies the Qualification question file that contains the new Qualification test. For information about this file, see *The Qualification question file* in Files Used by the Command Line Tools (p. 53). Example: `-question qualification.question`	No
-status [status value]	Specifies whether the Qualification is active or inactive. Inactive Qualifications are no longer available for Workers and cannot be used for new HITs. Valid Values: Active | Inactive Example: `-status Active`	No

Name	Description	Required
-sandbox	Runs this command in the Amazon Mechanical Turk sandbox for testing. This argument takes precedence even if you specify the production web site in your `mturk.properties` file. Example: `-sandbox`	No

Example

The following examples for Unix and Windows show how to use the `updateQualificationType` command. These examples update the Qualification test.

Unix

The following example demonstrates how to call this command from Unix.

```
./updateQualificationType.sh -qualtypeid RWFZTKZ55ZPZXN1C8TDZ -question sur
vey.question
```

Windows

The following example demonstrates how to call this command from Microsoft Windows.

```
updateQualificationType -qualtypeid RWFZTKZ55ZPZXN1C8TDZ -question survey.ques
tion
```

Output

These examples produce the following output.

```
Updated qualification type RWFZTKZ55ZPZXN1C8TDZ
```

Files Used by the Command Line Tools

This section describes the files that the Command Line Tools use.

File	Description	Used by
The Amazon Mechanical Turk properties file	This file is named `mturk.properties` and it must exist in the directory that you run the Command Line Tools from. This is a text file that contains specific information about your AWS identifiers as well as the Amazon Mechanical Turk service endpoint.	System
The approve file	A text file that contains a list of assignment IDs and optional approval comments. The first row must contain a column header named *assignmentIDToApprove*. The file can have an optional column named *assignmentIdToApproveComment*. Separate fields with a tab and enclose comments in quotes.	The *`-approvefile`* argument of approveWork (p. 5)
The failure file	A tab delimited text file that contains the rows from the input file that failed to load into Amazon Mechanical Turk. The system generates this file if failures occur when you use loadHITs (p. 30). You can fix the entries, rename the file, and then use it to load the HITs that failed. The system creates this file with the name `[your input filename].failure`.	Not used in command arguments.

File	Description	Used by
The HIT properties file	A text file that defines the properties of the HITs you are creating. Use the format `[property]:[value]` to list the properties in the file. Example: `reward:$0.06` For a list of properties, see HIT Properties (p. 57).	The *`-properties`* argument of the following commands: loadHITs (p. 30) updateHITs (p. 47)
The input file	A tab delimited text file that contains the dynamic fields of the HIT. This file can contain any information that you need for your HIT. Amazon Mechanical Turk merges this file into your Question template file and your HIT properties file. The first row of the file contains the field headings. Subsequent rows represent the custom field values for the HITs to be loaded. Each row represents one HIT (e.g. 1000 data rows = 1000 HITs).	The *`-input`* argument of the `loadHITs` command.
The output file	A tab delimited text file that contains the results of submitted HITs that are retrieved from Amazon Mechanical Turk. This file contains all data related to the submitted HITs and assignments. Each HIT in this file has a link to your **Manage HITs** page on the Requester web site. The last column contains a tab delimited set of question/answer pairs for each question field defined in your QuestionForm.	The *`-outputfile`* argument of getResults (p. 26)
The Qualification answer file	A text file that contains the AnswerKey XML that defines the answer key for your Qualification test. The Amazon Mechanical Turk system automatically scores the test.	The *`-answer`* argument of the following commands: createQualificationType (p. 13) updateQualificationType (p. 51)
The Qualification answer key file	A text file that contains name/value pairs of your questions and expected answers. This answer key provides another method to evaluate Qualification requests by giving you control over when requests are granted. This format also provides more flexibility by allowing you to evaluate FreeTextAnswer answers which cannot be autograded by Amazon Mechanical Turk. The format is `questionid=expected answer value`.	The *`-answers`* argument of evaluateQualificationRequests (p. 17)

File	Description	Used by
The Qualification approve file	A tab delimited text file that contains the list of Qualification requests to approve. The file must contain a column called *qualificationRequestToApprove* that contains the Qualification request IDs. The file can have an optional column called *qualificationRequestToApproveValue* that lists the Qualification score to assign for each Qualification request.	The `-approvefile` argument of approveQualificationRequests (p.3)
The Qualification ID file	A text file that contains the single Qualification ID you want to assign to workers. The file must have a column called *qualtypeid*.	The `-input` argument of the following commands: assignQualification (p. 8) evaluateQualificationRequests (p.17)
The Qualification properties file	A text file that defines the properties of the Qualification you are creating. Use the format `[property]=[value]` to list the properties in the file. Example: `autogranted=false`	The `-properties` argument of the following commands: createQualificationType (p. 13) updateQualificationType (p. 51)
The Qualification question file	A text file that contains the QuestionForm XML that defines your Qualification test. This file is similar to the question file except that templating is not applicable.	The `-question` argument of the following commands: createQualificationType (p. 13) updateQualificationType (p. 51)
The Qualification reject file	A tab delimited text file that contains the list of Qualification requests to reject and optional comments. The file must contain a column called *qualificationRequestToReject* that contains the Qualification request IDs. The file can include an optional column called *qualificationRequestToRejectComment* that contains comments you want to provide to the Worker regarding the rejection.	The `-rejectfile` argument of rejectQualificationRequests (p.35)
The Qualification request file	A tab delimited text file that contains the details of the Qualification requests made by Workers.	The `-outputfile` argument of getQualificationRequests (p. 24)

File	Description	Used by
The question file	A text file that contains the QuestionForm XML that defines your HIT. **Tip** To make your QuestionForm a template and merge it with values from your input file use the syntax `${the field name}`, where the field name is the column name from the input file. Amazon Mechanical Turk uses Apache Velocity to perform the merge. For more information about merge syntax, go to http://velocity.apache.org.	The *`-question`* argument of loadHITs (p. 30)
The reject file	A text file that contains a list of assignment ID and optional rejection comments. The first row must contain a column called *assignmentIdToReject*. The file can have an optional column called *assignmentIdToRejectComment*. Separate fields with a tab and enclose comments in quotes.	The *`-rejectfile`* argument of the rejectWork (p. 37) command
The score file	A tab delimited file that contains the Worker IDs and the Qualification scores to assign. The file must contain a column called *workerid* that contains the Worker IDs. The file can optionally contain a column called *score* that specifies the scores to assign to Workers.	The *`-scorefile`* argument of the assignQualification (p. 8) command.
The success file	A tab delimited text file that contains the HIT IDs and the HIT Type IDs that were successfully loaded into Amazon Mechanical Turk. This file is generated when you load HITs using loadHITs (p. 30) or updateHITs (p. 47). The file must contain a column called *hitid*. The system creates this file with the name `[your input filename].success`.	The *`-successfile`* argument of the following commands: approveWork (p. 5) deleteHITs (p. 15) extendHITs (p. 20) getResults (p. 26) updateHITs (p. 47)
The Worker ID file	A text file that contains a tab delimited list of Worker IDs to assign the Qualification to, and optional Qualification scores. The file must contain a column called *workerid*. The file can optionally contain a column called *score* that lists the Qualification scores to assign to each Worker.	The *`-input`* argument of the updateQualificationScore (p. 49) command

HIT Properties

The following table describes the HIT properties.

Property	Description
title	Title of the HIT
description	Description of the HIT
keywords	Keywords to associate with the HIT
reward	Reward for the HIT formatted as $0.00 USD
assignments	Maximum number of assignments available for this HIT
annotation	Value you use to identify this HIT **Tip** You can merge values from the input file into this property. To do this, use the syntax: `${the field name}`, where the field name is the column name from the input file. Amazon Mechanical Turk uses Apache Velocity to perform the merge. For more information about merge syntax, go to http://velocity.apache.org.
assignmentduration	Amount of time, in seconds, a Worker has to complete the assignment
hitlifetime	Amount of time, in seconds, the HIT is active before it expires
autoapprovaldelay	Amount of time, in seconds, before a submitted assignment is automatically approved
qualfication	Qualification requirements for your HIT. To specify the requirements, use the following syntax. • **qualification.1—**Qualification Type ID • **qualification.comparator.1—**String that specifies the kind of comparison to make • **qualification.value.1—**Integer value for the comparator to use • **qualification.locale.1—**Locale value to use • **qualification.private.1—**Boolean value that specifies whether Workers who do not meet the qualification requirements can preview the HIT To specify additional qualification requirements, increment the .1 suffix (e.g. *`qualification.value.2`*).

For more information about the qualification requirements, see QualificationRequirement Data Structure. For more information about the properties see HIT Data Structure.

Qualification Properties

The following table describes the qualification properties.

Property	Description
name	Name of the Qualification

Property	Description
description	Description of the Qualification
keywords	Keywords to associate with the Qualification
retrydelayinseconds	Minimum amount of time, in seconds, required before a Worker can re-request the Qualification
testdurationinseconds	Amount of time, in seconds, allowed for the Worker to complete the Qualification test (if a test exists)
autogranted	Specifies whether the Qualification should be autogranted upon request. This parameter is only valid if a Qualification does not have a test associated with it. Valid Values: true | false

Document History

This Document History describes the important changes to the documentation in this release of Amazon Mechanical Turk.

Relevant dates to this History

- **Current product version**-2012-03-25
- **Latest product release**-March 29, 2012
- **Latest documentation update**-March 29, 2012

Major Changes	Description	Release Date
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.	2012-03-29
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.	2011-12-01
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.	2011-06-22
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.	2009-09-16