

## tf.train.GlobalStepWaiterHook

### Contents

Class `GlobalStepWaiterHook`

### Methods

`__init__`

`after_create_session`

## Class `GlobalStepWaiterHook`

Inherits From: `SessionRunHook`

Defined in `tensorflow/python/training/basic_session_run_hooks.py`.

See the guide: [Training > Training Hooks](#)

Delays execution until global step reaches `wait_until_step`.

This hook delays execution until global step reaches to `wait_until_step`. It is used to gradually start workers in distributed settings. One example usage would be setting `wait_until_step=int(K*log(task_id+1))` assuming that `task_id=0` is the chief.

## Methods

### `__init__`

```
__init__(wait_until_step)
```

Initializes a `GlobalStepWaiterHook`.

Args:

- `wait_until_step`: an `int` shows until which global step should we wait.

### `after_create_session`

```
after_create_session(  
    session,  
    coord  
)
```

Called when new TensorFlow session is created.

This is called to signal the hooks that a new session has been created. This has two essential differences with the situation in which `begin` is called:

- When this is called, the graph is finalized and ops can no longer be added to the graph.
- This method will also be called as a result of recovering a wrapped session, not only at the beginning of the overall session.

Args:

- `session` : A TensorFlow Session that has been created.
- `coord` : A Coordinator object which keeps track of all threads.

## after\_run

```
after_run(
    run_context,
    run_values
)
```

Called after each call to `run()`.

The `run_values` argument contains results of requested ops/tensors by `before_run()`.

The `run_context` argument is the same one send to `before_run` call. `run_context.request_stop()` can be called to stop the iteration.

If `session.run()` raises any exceptions then `after_run()` is not called.

Args:

- `run_context` : A `SessionRunContext` object.
- `run_values` : A `SessionRunValues` object.

## before\_run

```
before_run(run_context)
```

## begin

```
begin()
```

## end

```
end(session)
```

Called at the end of session.

The `session` argument can be used in case the hook wants to run final ops, such as saving a last checkpoint.

If `session.run()` raises exception other than `OutOfRangeError` or `StopIteration` then `end()` is not called. Note the difference between `end()` and `after_run()` behavior when `session.run()` raises `OutOfRangeError` or `StopIteration`. In that case `end()` is called but `after_run()` is not called.

Args:

- `session` : A TensorFlow Session that will be soon closed.

---

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see our [Site Policies](#). Java is a registered trademark of Oracle and/or its affiliates.

Last updated November 2, 2017.

## Stay Connected

[Blog](#)

[GitHub](#)

[Twitter](#)

## Support

[Issue Tracker](#)

[Release Notes](#)

[Stack Overflow](#)

**English**

[Terms](#) | [Privacy](#)