# tf.TextLineReader

Contents



## Class `TextLineReader`

Inherits From: `ReaderBase`

Defined in `tensorflow/python/ops/io_ops.py`.

See the guides: Inputs and Readers > Readers, Reading data > Reading from files

A Reader that outputs the lines of a file delimited by newlines.

Newlines are stripped from the output. See ReaderBase for supported methods.

## Properties

### `reader_ref`

Op that implements the reader.

### `supports_serialize`

Whether the Reader implementation can serialize its state.

## Methods

### `__init__`

```
__init__(
    skip_header_lines=None,
    name=None
)
```

Create a TextLineReader.

Args:

- `skip_header_lines`: An optional int. Defaults to 0. Number of lines to skip from the beginning of every file.
- `name`: A name for the operation (optional).

### num_records_produced

```
num_records_produced(name=None)
```

Returns the number of records this reader has produced.

This is the same as the number of Read executions that have succeeded.

Args:

- `name`: A name for the operation (optional).

Returns:

An int64 Tensor.

### num_work_units_completed

```
num_work_units_completed(name=None)
```

Returns the number of work units this reader has finished processing.

Args:

- `name`: A name for the operation (optional).

Returns:

An int64 Tensor.

### read

```
read(
    queue,
    name=None
)
```

Returns the next record (key, value) pair produced by a reader.

Will dequeue a work unit from queue if necessary (e.g. when the Reader needs to start reading from a new file since it has finished with the previous file).

Args:

- `queue`: A Queue or a mutable string Tensor representing a handle to a Queue, with string work items.
- `name`: A name for the operation (optional).

Returns:

A tuple of Tensors (key, value). `key`: *A string scalar Tensor.* `value`: A string scalar Tensor.

### `read_up_to`

```
read_up_to(
    queue,
    num_records,
    name=None
)
```

Returns up to num_records (key, value) pairs produced by a reader.

Will dequeue a work unit from queue if necessary (e.g., when the Reader needs to start reading from a new file since it has finished with the previous file). It may return less than num_records even before the last batch.

Args:

- `queue`: A Queue or a mutable string Tensor representing a handle to a Queue, with string work items.
- `num_records`: Number of records to read.
- `name`: A name for the operation (optional).

Returns:

A tuple of Tensors (keys, values). `keys`: *A 1-D string Tensor.* `values`: A 1-D string Tensor.

### `reset`

```
reset(name=None)
```

Restore a reader to its initial clean state.

Args:

- `name`: A name for the operation (optional).

Returns:

The created Operation.

### `restore_state`

```
restore_state(
    state,
    name=None
)
```

Restore a reader to a previously saved state.

Not all Readers support being restored, so this can produce an Unimplemented error.

Args:

- `state`: A string Tensor. Result of a SerializeState of a Reader with matching type.
- `name`: A name for the operation (optional).

Returns:

The created Operation.

## `serialize_state`

```
serialize_state(name=None)
```

Produce a string tensor that encodes the state of a reader.

Not all Readers support being serialized, so this can produce an Unimplemented error.

Args:

- `name`: A name for the operation (optional).

Returns:

A string Tensor.

---