Module: tf.compat.v1.saved\_model / tf.saved\_model

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model#top_of_page)
* [Modules](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model#modules)
* [Classes](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model#classes)
* [Functions](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model#functions)
* [Other Members](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model#other_members)

Public API for tf.saved\_model namespace.

Modules

[builder](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/builder) module: SavedModel builder.

[constants](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/constants) module: Constants for SavedModel save and restore operations.

[experimental](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/experimental) module: Public API for tf.saved\_model.experimental namespace.

[loader](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/loader) module: Loader functionality for SavedModel with hermetic, language-neutral exports.

[main\_op](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/main_op) module: SavedModel main op.

[signature\_constants](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/signature_constants) module: Signature constants for SavedModel save and restore operations.

[signature\_def\_utils](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/signature_def_utils) module: SignatureDef utility functions.

[tag\_constants](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/tag_constants) module: Common tags used for graphs in SavedModel.

[utils](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/utils) module: SavedModel utility functions.

Classes

[class Builder](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/Builder): Builds the SavedModel protocol buffer and saves variables and assets.

Functions

[build\_signature\_def(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/build_signature_def): Utility function to build a SignatureDef protocol buffer.

[build\_tensor\_info(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/build_tensor_info): Utility function to build TensorInfo proto from a Tensor. (deprecated)

[classification\_signature\_def(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/classification_signature_def): Creates classification signature from given examples and predictions.

[contains\_saved\_model(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/contains_saved_model): Checks whether the provided export directory could contain a SavedModel.

[get\_tensor\_from\_tensor\_info(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/get_tensor_from_tensor_info): Returns the Tensor or SparseTensor described by a TensorInfo proto. (deprecated)

[is\_valid\_signature(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/is_valid_signature): Determine whether a SignatureDef can be served by TensorFlow Serving.

[load(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/load): Loads the model from a SavedModel as specified by tags. (deprecated)

[load\_v2(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/saved_model/load): Load a SavedModel from export\_dir.

[main\_op\_with\_restore(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/main_op_with_restore): Returns a main op to init variables, tables and restore the graph. (deprecated)

[maybe\_saved\_model\_directory(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/contains_saved_model): Checks whether the provided export directory could contain a SavedModel.

[predict\_signature\_def(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/predict_signature_def): Creates prediction signature from given inputs and outputs.

[regression\_signature\_def(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/regression_signature_def): Creates regression signature from given examples and predictions.

[save(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/saved_model/save): Exports the Trackable object obj to [SavedModel format](https://github.com/tensorflow/tensorflow/blob/master/tensorflow/python/saved_model/README.md).

[simple\_save(...)](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/simple_save): Convenience function to build a SavedModel suitable for serving. (deprecated)

Other Members

* ASSETS\_DIRECTORY = 'assets'
* ASSETS\_KEY = 'saved\_model\_assets'
* CLASSIFY\_INPUTS = 'inputs'
* CLASSIFY\_METHOD\_NAME = 'tensorflow/serving/classify'
* CLASSIFY\_OUTPUT\_CLASSES = 'classes'
* CLASSIFY\_OUTPUT\_SCORES = 'scores'
* DEFAULT\_SERVING\_SIGNATURE\_DEF\_KEY = 'serving\_default'
* GPU = 'gpu'
* LEGACY\_INIT\_OP\_KEY = 'legacy\_init\_op'
* MAIN\_OP\_KEY = 'saved\_model\_main\_op'
* PREDICT\_INPUTS = 'inputs'
* PREDICT\_METHOD\_NAME = 'tensorflow/serving/predict'
* PREDICT\_OUTPUTS = 'outputs'
* REGRESS\_INPUTS = 'inputs'
* REGRESS\_METHOD\_NAME = 'tensorflow/serving/regress'
* REGRESS\_OUTPUTS = 'outputs'
* SAVED\_MODEL\_FILENAME\_PB = 'saved\_model.pb'
* SAVED\_MODEL\_FILENAME\_PBTXT = 'saved\_model.pbtxt'
* SAVED\_MODEL\_SCHEMA\_VERSION = 1
* SERVING = 'serve'
* TPU = 'tpu'
* TRAINING = 'train'
* VARIABLES\_DIRECTORY = 'variables'
* VARIABLES\_FILENAME = 'variables'

# tf.saved\_model.contains\_saved\_model

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/saved_model/contains_saved_model#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/saved_model/contains_saved_model#aliases)

Checks whether the provided export directory could contain a SavedModel.

### Aliases:

* tf.compat.v2.saved\_model.contains\_saved\_model
* tf.saved\_model.contains\_saved\_model

tf.saved\_model.contains\_saved\_model(export\_dir)

Defined in [python/saved\_model/loader\_impl.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/saved_model/loader_impl.py).

Note that the method does not load any data by itself. If the method returns false, the export directory definitely does not contain a SavedModel. If the method returns true, the export directory may contain a SavedModel but provides no guarantee that it can be loaded.

#### Args:

* **export\_dir**: Absolute string path to possible export location. For example, '/my/foo/model'.

#### Returns:

True if the export directory contains SavedModel files, False otherwise.

# tf.saved\_model.load

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/saved_model/load#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/saved_model/load#aliases)
* [Used in the guide:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/saved_model/load#used_in_the_guide)

Load a SavedModel from export\_dir.

### Aliases:

* tf.compat.v1.saved\_model.load\_v2
* tf.compat.v2.saved\_model.load
* tf.saved\_model.load

tf.saved\_model.load(  
    export\_dir,  
    tags=None  
)

Defined in [python/saved\_model/load.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/saved_model/load.py).

### Used in the guide:

* [Using the SavedModel format](https://www.tensorflow.org/beta/guide/saved_model)

Signatures associated with the SavedModel are available as functions:

imported = tf.saved\_model.load(path)  
f = imported.signatures["serving\_default"]  
print(f(x=tf.constant([[1.]])))

Objects exported with [tf.saved\_model.save](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/saved_model/save) additionally have trackable objects and functions assigned to attributes:

exported = tf.train.Checkpoint(v=tf.Variable(3.))  
exported.f = tf.function(  
    lambda x: exported.v \* x,  
    input\_signature=[tf.TensorSpec(shape=None, dtype=tf.float32)])  
tf.saved\_model.save(exported, path)  
imported = tf.saved\_model.load(path)  
assert 3. == imported.v.numpy()  
assert 6. == imported.f(x=tf.constant(2.)).numpy()

Loading Keras models

Keras models are trackable, so they can be saved to SavedModel. The object returned by [tf.saved\_model.load](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/saved_model/load) is not a Keras object (i.e. doesn't have .fit, .predict, etc. methods). A few attributes and functions are still available: .variables, .trainable\_variables and .\_\_call\_\_.

model = tf.keras.Model(...)  
tf.saved\_model.save(model, path)  
imported = tf.saved\_model.load(path)  
outputs = imported(inputs)

Use [tf.keras.models.load\_model](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/keras/models/load_model) to restore the Keras model.

Importing SavedModels from TensorFlow 1.x

SavedModels from [tf.estimator.Estimator](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/estimator/Estimator) or 1.x SavedModel APIs have a flat graph instead of [tf.function](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/function) objects. These SavedModels will have functions corresponding to their signatures in the .signatures attribute, but also have a .prune method which allows you to extract functions for new subgraphs. This is equivalent to importing the SavedModel and naming feeds and fetches in a Session from TensorFlow 1.x.

imported = tf.saved\_model.load(path\_to\_v1\_saved\_model)  
pruned = imported.prune("x:0", "out:0")  
pruned(tf.ones([]))

See [tf.compat.v1.wrap\_function](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/wrap_function) for details. These SavedModels also have a .variablesattribute containing imported variables, and a .graph attribute representing the whole imported graph. For SavedModels exported from [tf.saved\_model.save](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/saved_model/save), variables are instead assigned to whichever attributes they were assigned before export.

#### Args:

* **export\_dir**: The SavedModel directory to load from.
* **tags**: A tag or sequence of tags identifying the MetaGraph to load. Optional if the SavedModel contains a single MetaGraph, as for those exported from [tf.saved\_model.load](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/saved_model/load).

#### Returns:

A trackable object with a signatures attribute mapping from signature keys to functions. If the SavedModel was exported by [tf.saved\_model.load](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/saved_model/load), it also points to trackable objects and functions which were attached to the exported object.

#### Raises:

* **ValueError**: If tags don't match a MetaGraph in the SavedModel.

# tf.saved\_model.save

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/saved_model/save#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/saved_model/save#aliases)
* [Used in the guide:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/saved_model/save#used_in_the_guide)

Exports the Trackable object obj to [SavedModel format](https://github.com/tensorflow/tensorflow/blob/master/tensorflow/python/saved_model/README.md).

### Aliases:

* tf.compat.v1.saved\_model.experimental.save
* tf.compat.v1.saved\_model.save
* tf.compat.v2.saved\_model.save
* tf.saved\_model.save

tf.saved\_model.save(  
    obj,  
    export\_dir,  
    signatures=None  
)

Defined in [python/saved\_model/save.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/saved_model/save.py).

### Used in the guide:

* [Using the SavedModel format](https://www.tensorflow.org/beta/guide/saved_model)

#### Example usage:

class Adder(tf.Module):  
  
  @tf.function(input\_signature=[tf.TensorSpec(shape=None, dtype=tf.float32)])  
  def add(self, x):  
    return x + x + 1.  
  
to\_export = Adder()  
tf.saved\_model.save(to\_export, '/tmp/adder')

The resulting SavedModel is then servable with an input named "x", its value having any shape and dtype float32.

The optional signatures argument controls which methods in obj will be available to programs which consume SavedModels, for example serving APIs. Python functions may be decorated with@tf.function(input\_signature=...) and passed as signatures directly, or lazily with a call to get\_concrete\_function on the method decorated with @tf.function.

If the signatures argument is omitted, obj will be searched for @tf.function-decorated methods. If exactly one @tf.function is found, that method will be used as the default signature for the SavedModel. This behavior is expected to change in the future, when a corresponding[tf.saved\_model.load](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/saved_model/load) symbol is added. At that point signatures will be completely optional, and any @tf.function attached to obj or its dependencies will be exported for use with load.

When invoking a signature in an exported SavedModel, Tensor arguments are identified by name. These names will come from the Python function's argument names by default. They may be overridden by specifying a name=... argument in the corresponding [tf.TensorSpec](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/TensorSpec) object. Explicit naming is required if multiple Tensors are passed through a single argument to the Python function.

The outputs of functions used as signatures must either be flat lists, in which case outputs will be numbered, or a dictionary mapping string keys to Tensor, in which case the keys will be used to name outputs.

Signatures are available in objects returned by [tf.saved\_model.load](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/saved_model/load) as a .signatures attribute. This is a reserved attribute: [tf.saved\_model.save](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/saved_model/save) on an object with a custom .signaturesattribute will raise an exception.

Since [tf.keras.Model](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/keras/Model) objects are also Trackable, this function can be used to export Keras models. For example, exporting with a signature specified:

class Model(tf.keras.Model):  
  
  @tf.function(input\_signature=[tf.TensorSpec(shape=[None], dtype=tf.string)])  
  def serve(self, serialized):  
    ...  
  
m = Model()  
tf.saved\_model.save(m, '/tmp/saved\_model/')

Exporting from a function without a fixed signature:

class Model(tf.keras.Model):  
  
  @tf.function  
  def call(self, x):  
    ...  
  
m = Model()  
tf.saved\_model.save(  
    m, '/tmp/saved\_model/',  
    signatures=m.call.get\_concrete\_function(  
        tf.TensorSpec(shape=[None, 3], dtype=tf.float32, name="inp")))

[tf.keras.Model](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/keras/Model) instances constructed from inputs and outputs already have a signature and so do not require a @tf.function decorator or a signatures argument. If neither are specified, the model's forward pass is exported.

x = input\_layer.Input((4,), name="x")  
y = core.Dense(5, name="out")(x)  
model = training.Model(x, y)  
tf.saved\_model.save(model, '/tmp/saved\_model/')  
# The exported SavedModel takes "x" with shape [None, 4] and returns "out"  
# with shape [None, 5]

Variables must be tracked by assigning them to an attribute of a tracked object or to an attribute of obj directly. TensorFlow objects (e.g. layers from [tf.keras.layers](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/keras/layers), optimizers from [tf.train](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/train)) track their variables automatically. This is the same tracking scheme that [tf.train.Checkpoint](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/train/Checkpoint)uses, and an exported Checkpoint object may be restored as a training checkpoint by pointing [tf.train.Checkpoint.restore](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/train/Checkpoint#restore) to the SavedModel's "variables/" subdirectory. Currently variables are the only stateful objects supported by [tf.saved\_model.save](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/saved_model/save), but others (e.g. tables) will be supported in the future.

[tf.function](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/function) does not hard-code device annotations from outside the function body, instead using the calling context's device. This means for example that exporting a model which runs on a GPU and serving it on a CPU will generally work, with some exceptions. [tf.device](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/device) annotations inside the body of the function will be hard-coded in the exported model; this type of annotation is discouraged. Device-specific operations, e.g. with "cuDNN" in the name or with device-specific layouts, may cause issues. Currently a DistributionStrategy is another exception: active distribution strategies will cause device placements to be hard-coded in a function. Exporting a single-device computation and importing under a DistributionStrategy is not currently supported, but may be in the future.

SavedModels exported with [tf.saved\_model.save](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/saved_model/save) [strip default-valued attributes](https://github.com/tensorflow/tensorflow/blob/master/tensorflow/python/saved_model/README.md#stripping-default-valued-attributes) automatically, which removes one source of incompatibilities when the consumer of a SavedModel is running an older TensorFlow version than the producer. There are however other sources of incompatibilities which are not handled automatically, such as when the exported model contains operations which the consumer does not have definitions for.

#### Args:

* **obj**: A trackable object to export.
* **export\_dir**: A directory in which to write the SavedModel.
* **signatures**: Optional, either a [tf.function](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/function) with an input signature specified or the result of f.get\_concrete\_function on a @tf.function-decorated function f, in which case f will be used to generate a signature for the SavedModel under the default serving signature key. signatures may also be a dictionary, in which case it maps from signature keys to either [tf.function](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/function) instances with input signatures or concrete functions. The keys of such a dictionary may be arbitrary strings, but will typically be from thetf.saved\_model.signature\_constants module.

#### Raises:

* **ValueError**: If obj is not trackable.

#### Eager Compatibility

Not well supported when graph building. From TensorFlow 1.x,tf.compat.v1.enable\_eager\_execution() should run first. Calling tf.saved\_model.save in a loop when graph building from TensorFlow 1.x will add new save operations to the default graph each iteration.

May not be called from within a function body.

# tf.compat.v1.saved\_model.Builder

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/Builder#top_of_page)
* [Class Builder](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/Builder#class_builder)
  + [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/Builder#aliases)
* [\_\_init\_\_](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/Builder#__init__)
* [Methods](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/Builder#methods)

## Class Builder

Builds the SavedModel protocol buffer and saves variables and assets.

### Aliases:

* Class tf.compat.v1.saved\_model.Builder
* Class tf.compat.v1.saved\_model.builder.SavedModelBuilder

Defined in [python/saved\_model/builder\_impl.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/saved_model/builder_impl.py).

The SavedModelBuilder class provides functionality to build a SavedModel protocol buffer. Specifically, this allows multiple meta graphs to be saved as part of a single language-neutral SavedModel, while sharing variables and assets.

To build a SavedModel, the first meta graph must be saved with variables. Subsequent meta graphs will simply be saved with their graph definitions. If assets need to be saved and written or copied to disk, they can be provided when the meta graph def is added. If multiple meta graph defs are associated an asset of the same name, only the first version is retained.

Each meta graph added to the SavedModel must be annotated with tags. The tags provide a means to identify the specific meta graph to load and restore, along with the shared set of variables and assets.

Typical usage for the SavedModelBuilder:

...  
builder = tf.compat.v1.saved\_model.Builder(export\_dir)  
  
with tf.compat.v1.Session(graph=tf.Graph()) as sess:  
  ...  
  builder.add\_meta\_graph\_and\_variables(sess,  
                                  ["foo-tag"],  
                                  signature\_def\_map=foo\_signatures,  
                                  assets\_collection=foo\_assets)  
...  
  
with tf.compat.v1.Session(graph=tf.Graph()) as sess:  
  ...  
  builder.add\_meta\_graph(["bar-tag", "baz-tag"])  
...  
  
builder.save()

**Note:** This function will only be available through the v1 compatibility library as tf.compat.v1.saved\_model.builder.SavedModelBuilder or tf.compat.v1.saved\_model.Builder. Tensorflow 2.0 will introduce a new object-based method of creating SavedModels.

## \_\_init\_\_

\_\_init\_\_(export\_dir)

## Methods

### add\_meta\_graph

add\_meta\_graph(  
    tags,  
    signature\_def\_map=None,  
    assets\_collection=None,  
    legacy\_init\_op=None,  
    clear\_devices=False,  
    main\_op=None,  
    strip\_default\_attrs=False,  
    saver=None  
)

Adds the current meta graph to the SavedModel.

Creates a Saver in the current scope and uses the Saver to export the meta graph def. Invoking this API requires the add\_meta\_graph\_and\_variables() API to have been invoked before.

#### Args:

* **tags**: The set of tags to annotate the meta graph def with.
* **signature\_def\_map**: The map of signature defs to be added to the meta graph def.
* **assets\_collection**: Assets to be saved with SavedModel. Note that this list should be a subset of the assets saved as part of the first meta graph in the SavedModel.
* **clear\_devices**: Set to true if the device info on the default graph should be cleared.
* **init\_op**: Op or group of ops to execute when the graph is loaded. Note that when the init\_op is specified it is run after the restore op at load-time.
* **train\_op**: Op or group of opts that trains the model when run. This will not be run automatically when the graph is loaded, instead saved in a SignatureDef accessible through the exported MetaGraph.
* **saver**: An instance of tf.compat.v1.train.Saver that will be used to export the metagraph. If None, a sharded Saver that restores all variables will be used.

#### Raises:

* **AssertionError**: If the variables for the SavedModel have not been saved yet, or if the graph already contains one or more legacy init ops.

### add\_meta\_graph\_and\_variables

add\_meta\_graph\_and\_variables(  
    sess,  
    tags,  
    signature\_def\_map=None,  
    assets\_collection=None,  
    legacy\_init\_op=None,  
    clear\_devices=False,  
    main\_op=None,  
    strip\_default\_attrs=False,  
    saver=None  
)

Adds the current meta graph to the SavedModel and saves variables.

Creates a Saver to save the variables from the provided session. Exports the corresponding meta graph def. This function assumes that the variables to be saved have been initialized. For a given SavedModelBuilder, this API must be called exactly once and for the first meta graph to save. For subsequent meta graph defs to be added, the add\_meta\_graph() API must be used.

#### Args:

* **sess**: The TensorFlow session from which to save the meta graph and variables.
* **tags**: The set of tags with which to save the meta graph.
* **signature\_def\_map**: The map of signature def map to add to the meta graph def.
* **assets\_collection**: Assets to be saved with SavedModel.
* **clear\_devices**: Set to true if the device info on the default graph should be cleared.
* **init\_op**: Op or group of ops to execute when the graph is loaded. Note that when the init\_op is specified it is run after the restore op at load-time.
* **train\_op**: Op or group of ops that trains the model when run. This will not be run automatically when the graph is loaded, instead saved in a SignatureDef accessible through the exported MetaGraph.
* **strip\_default\_attrs**: Boolean. If True, default-valued attributes will be removed from the NodeDefs. For a detailed guide, see [Stripping Default-Valued Attributes](https://github.com/tensorflow/tensorflow/blob/master/tensorflow/python/saved_model/README.md#stripping-default-valued-attributes).
* **saver**: An instance of tf.compat.v1.train.Saver that will be used to export the metagraph and save variables. If None, a sharded Saver that restores all variables will be used.

### save

save(as\_text=False)

Writes a SavedModel protocol buffer to disk.

The function writes the SavedModel protocol buffer to the export directory in serialized format.

#### Args:

* **as\_text**: Writes the SavedModel protocol buffer in text format to disk. Protocol buffers in text format are useful for debugging, but parsing fails when it encounters an unknown field and so is not forward compatible. This means changes to TensorFlow may prevent deployment of new text format SavedModels to existing serving binaries. Do not deploy as\_text SavedModels to production.
* Returns:

# The path to which the SavedModel protocol buffer was written.

# tf.compat.v1.saved\_model.build\_signature\_def

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/build_signature_def#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/build_signature_def#aliases)

Utility function to build a SignatureDef protocol buffer.

### Aliases:

* tf.compat.v1.saved\_model.build\_signature\_def
* tf.compat.v1.saved\_model.signature\_def\_utils.build\_signature\_def

tf.compat.v1.saved\_model.build\_signature\_def(  
    inputs=None,  
    outputs=None,  
    method\_name=None  
)

Defined in [python/saved\_model/signature\_def\_utils\_impl.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/saved_model/signature_def_utils_impl.py).

#### Args:

* **inputs**: Inputs of the SignatureDef defined as a proto map of string to tensor info.
* **outputs**: Outputs of the SignatureDef defined as a proto map of string to tensor info.
* **method\_name**: Method name of the SignatureDef as a string.

#### Returns:

A SignatureDef protocol buffer constructed based on the supplied arguments.

# tf.compat.v1.saved\_model.build\_tensor\_info

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/build_tensor_info#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/build_tensor_info#aliases)

Utility function to build TensorInfo proto from a Tensor. (deprecated)

### Aliases:

* tf.compat.v1.saved\_model.build\_tensor\_info
* tf.compat.v1.saved\_model.utils.build\_tensor\_info

tf.compat.v1.saved\_model.build\_tensor\_info(tensor)

Defined in [python/saved\_model/utils\_impl.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/saved_model/utils_impl.py).

**Warning:** THIS FUNCTION IS DEPRECATED. It will be removed in a future version. Instructions for updating: This function will only be available through the v1 compatibility library as tf.compat.v1.saved\_model.utils.build\_tensor\_info or tf.compat.v1.saved\_model.build\_tensor\_info.

#### Args:

* **tensor**: Tensor or SparseTensor whose name, dtype and shape are used to build the TensorInfo. For SparseTensors, the names of the three constituent Tensors are used.

#### Returns:

A TensorInfo protocol buffer constructed based on the supplied argument.

#### Raises:

* **RuntimeError**: If eager execution is enabled.

# tf.compat.v1.saved\_model.classification\_signature\_def

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/classification_signature_def#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/classification_signature_def#aliases)

Creates classification signature from given examples and predictions.

### Aliases:

* tf.compat.v1.saved\_model.classification\_signature\_def
* tf.compat.v1.saved\_model.signature\_def\_utils.classification\_signature\_def

tf.compat.v1.saved\_model.classification\_signature\_def(  
    examples,  
    classes,  
    scores  
)

Defined in [python/saved\_model/signature\_def\_utils\_impl.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/saved_model/signature_def_utils_impl.py).

This function produces signatures intended for use with the TensorFlow Serving Classify API (tensorflow\_serving/apis/prediction\_service.proto), and so constrains the input and output types to those allowed by TensorFlow Serving.

#### Args:

* **examples**: A string Tensor, expected to accept serialized tf.Examples.
* **classes**: A string Tensor. Note that the ClassificationResponse message requires that class labels are strings, not integers or anything else.
* **scores**: a float Tensor.

#### Returns:

A classification-flavored signature\_def.

#### Raises:

* **ValueError**: If examples is None.

# tf.compat.v1.saved\_model.contains\_saved\_model

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/contains_saved_model#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/contains_saved_model#aliases)

Checks whether the provided export directory could contain a SavedModel.

### Aliases:

* tf.compat.v1.saved\_model.contains\_saved\_model
* tf.compat.v1.saved\_model.loader.maybe\_saved\_model\_directory
* tf.compat.v1.saved\_model.maybe\_saved\_model\_directory

tf.compat.v1.saved\_model.contains\_saved\_model(export\_dir)

Defined in [python/saved\_model/loader\_impl.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/saved_model/loader_impl.py).

Note that the method does not load any data by itself. If the method returns false, the export directory definitely does not contain a SavedModel. If the method returns true, the export directory may contain a SavedModel but provides no guarantee that it can be loaded.

#### Args:

* **export\_dir**: Absolute string path to possible export location. For example, '/my/foo/model'.

#### Returns:

True if the export directory contains SavedModel files, False otherwise.

# tf.compat.v1.saved\_model.get\_tensor\_from\_tensor\_info

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/get_tensor_from_tensor_info#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/get_tensor_from_tensor_info#aliases)

Returns the Tensor or SparseTensor described by a TensorInfo proto. (deprecated)

### Aliases:

* tf.compat.v1.saved\_model.get\_tensor\_from\_tensor\_info
* tf.compat.v1.saved\_model.utils.get\_tensor\_from\_tensor\_info

tf.compat.v1.saved\_model.get\_tensor\_from\_tensor\_info(  
    tensor\_info,  
    graph=None,  
    import\_scope=None  
)

Defined in [python/saved\_model/utils\_impl.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/saved_model/utils_impl.py).

**Warning:** THIS FUNCTION IS DEPRECATED. It will be removed in a future version. Instructions for updating: This function will only be available through the v1 compatibility library as tf.compat.v1.saved\_model.utils.get\_tensor\_from\_tensor\_info or tf.compat.v1.saved\_model.get\_tensor\_from\_tensor\_info.

#### Args:

* **tensor\_info**: A TensorInfo proto describing a Tensor or SparseTensor.
* **graph**: The tf.Graph in which tensors are looked up. If None, the current default graph is used.
* **import\_scope**: If not None, names in tensor\_info are prefixed with this string before lookup.

#### Returns:

The Tensor or SparseTensor in graph described by tensor\_info.

#### Raises:

* **KeyError**: If tensor\_info does not correspond to a tensor in graph.
* **ValueError**: If tensor\_info is malformed.

# tf.compat.v1.saved\_model.is\_valid\_signature

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/is_valid_signature#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/is_valid_signature#aliases)

Determine whether a SignatureDef can be served by TensorFlow Serving.

### Aliases:

* tf.compat.v1.saved\_model.is\_valid\_signature
* tf.compat.v1.saved\_model.signature\_def\_utils.is\_valid\_signature

tf.compat.v1.saved\_model.is\_valid\_signature(signature\_def)

Defined in [python/saved\_model/signature\_def\_utils\_impl.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/saved_model/signature_def_utils_impl.py).

# tf.compat.v1.saved\_model.load

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/load#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/load#aliases)

Loads the model from a SavedModel as specified by tags. (deprecated)

### Aliases:

* tf.compat.v1.saved\_model.load
* tf.compat.v1.saved\_model.loader.load

tf.compat.v1.saved\_model.load(  
    sess,  
    tags,  
    export\_dir,  
    import\_scope=None,  
    \*\*saver\_kwargs  
)

Defined in [python/saved\_model/loader\_impl.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/saved_model/loader_impl.py).

**Warning:** THIS FUNCTION IS DEPRECATED. It will be removed in a future version. Instructions for updating: This function will only be available through the v1 compatibility library as tf.compat.v1.saved\_model.loader.load or tf.compat.v1.saved\_model.load. There will be a new function for importing SavedModels in Tensorflow 2.0.

#### Args:

* **sess**: The TensorFlow session to restore the variables.
* **tags**: Set of string tags to identify the required MetaGraphDef. These should correspond to the tags used when saving the variables using the SavedModel save() API.
* **export\_dir**: Directory in which the SavedModel protocol buffer and variables to be loaded are located.
* **import\_scope**: Optional string -- if specified, prepend this string followed by '/' to all loaded tensor names. This scope is applied to tensor instances loaded into the passed session, but it is not written through to the static MetaGraphDef protocol buffer that is returned.
* **\*\*saver\_kwargs**: Optional keyword arguments passed through to Saver.

#### Returns:

The MetaGraphDef protocol buffer loaded in the provided session. This can be used to further extract signature-defs, collection-defs, etc.

#### Raises:

* **RuntimeError**: MetaGraphDef associated with the tags cannot be found.

# tf.compat.v1.saved\_model.main\_op\_with\_restore

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/main_op_with_restore#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/main_op_with_restore#aliases)

Returns a main op to init variables, tables and restore the graph. (deprecated)

### Aliases:

* tf.compat.v1.saved\_model.main\_op.main\_op\_with\_restore
* tf.compat.v1.saved\_model.main\_op\_with\_restore

tf.compat.v1.saved\_model.main\_op\_with\_restore(restore\_op\_name)

Defined in [python/saved\_model/main\_op\_impl.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/saved_model/main_op_impl.py).

**Warning:** THIS FUNCTION IS DEPRECATED. It will be removed in a future version. Instructions for updating: This function will only be available through the v1 compatibility library as tf.compat.v1.saved\_model.main\_op\_with\_restore or tf.compat.v1.saved\_model.main\_op.main\_op\_with\_restore.

Returns the main op including the group of ops that initializes all variables, initialize local variables, initialize all tables and the restore op name.

#### Args:

* **restore\_op\_name**: Name of the op to use to restore the graph.

#### Returns:

The set of ops to be run as part of the main op upon the load operation.

# tf.compat.v1.saved\_model.predict\_signature\_def

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/predict_signature_def#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/predict_signature_def#aliases)

Creates prediction signature from given inputs and outputs.

### Aliases:

* tf.compat.v1.saved\_model.predict\_signature\_def
* tf.compat.v1.saved\_model.signature\_def\_utils.predict\_signature\_def

tf.compat.v1.saved\_model.predict\_signature\_def(  
    inputs,  
    outputs  
)

Defined in [python/saved\_model/signature\_def\_utils\_impl.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/saved_model/signature_def_utils_impl.py).

This function produces signatures intended for use with the TensorFlow Serving Predict API (tensorflow\_serving/apis/prediction\_service.proto). This API imposes no constraints on the input and output types.

#### Args:

* **inputs**: dict of string to Tensor.
* **outputs**: dict of string to Tensor.

#### Returns:

A prediction-flavored signature\_def.

#### Raises:

* **ValueError**: If inputs or outputs is None.

# tf.compat.v1.saved\_model.regression\_signature\_def

* [**Contents**](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/regression_signature_def#top_of_page)
* [Aliases:](https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/compat/v1/saved_model/regression_signature_def#aliases)

Creates regression signature from given examples and predictions.

### Aliases:

* tf.compat.v1.saved\_model.regression\_signature\_def
* tf.compat.v1.saved\_model.signature\_def\_utils.regression\_signature\_def

tf.compat.v1.saved\_model.regression\_signature\_def(  
    examples,  
    predictions  
)

Defined in [python/saved\_model/signature\_def\_utils\_impl.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/saved_model/signature_def_utils_impl.py).

This function produces signatures intended for use with the TensorFlow Serving Regress API (tensorflow\_serving/apis/prediction\_service.proto), and so constrains the input and output types to those allowed by TensorFlow Serving.

#### Args:

* **examples**: A string Tensor, expected to accept serialized tf.Examples.
* **predictions**: A float Tensor.

#### Returns:

A regression-flavored signature\_def.

#### Raises:

* **ValueError**: If examples is None.

# tf.compat.v1.saved\_model.simple\_save

Convenience function to build a SavedModel suitable for serving. (deprecated)

tf.compat.v1.saved\_model.simple\_save(  
    session,  
    export\_dir,  
    inputs,  
    outputs,  
    legacy\_init\_op=None  
)

Defined in [python/saved\_model/simple\_save.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/saved_model/simple_save.py).

**Warning:** THIS FUNCTION IS DEPRECATED. It will be removed in a future version. Instructions for updating: This function will only be available through the v1 compatibility library as tf.compat.v1.saved\_model.simple\_save.

In many common cases, saving models for serving will be as simple as:

simple\_save(session,  
            export\_dir,  
            inputs={"x": x, "y": y},  
            outputs={"z": z})

Although in many cases it's not necessary to understand all of the many ways to configure a SavedModel, this method has a few practical implications: - It will be treated as a graph for inference / serving (i.e. uses the tag saved\_model.SERVING) - The SavedModel will load in TensorFlow Serving and supports the [Predict API](https://github.com/tensorflow/serving/blob/master/tensorflow_serving/apis/predict.proto). To use the Classify, Regress, or MultiInference APIs, please use either[tf.Estimator](https://www.tensorflow.org/api_docs/python/tf/estimator/Estimator) or the lower level [SavedModel APIs](https://github.com/tensorflow/tensorflow/blob/master/tensorflow/python/saved_model/README.md). - Some TensorFlow ops depend on information on disk or other information called "assets". These are generally handled automatically by adding the assets to the GraphKeys.ASSET\_FILEPATHS collection. Only assets in that collection are exported; if you need more custom behavior, you'll need to use the [SavedModelBuilder](https://github.com/tensorflow/tensorflow/blob/master/tensorflow/python/saved_model/builder.py).

More information about SavedModel and signatures can be found here: https://github.com/tensorflow/tensorflow/blob/master/tensorflow/python/saved\_model/README.md.

#### Args:

* **session**: The TensorFlow session from which to save the meta graph and variables.
* **export\_dir**: The path to which the SavedModel will be stored.
* **inputs**: dict mapping string input names to tensors. These are added to the SignatureDef as the inputs.
* **outputs**: dict mapping string output names to tensors. These are added to the SignatureDef as the outputs.
* **legacy\_init\_op**: Legacy support for op or group of ops to execute after the restore op upon a load.

# tf.compat.v1.saved\_model.main\_op.main\_op

Returns a main op to init variables and tables. (deprecated)

tf.compat.v1.saved\_model.main\_op.main\_op()

Defined in [python/saved\_model/main\_op\_impl.py](https://github.com/tensorflow/tensorflow/tree/r2.0/tensorflow/python/saved_model/main_op_impl.py).

**Warning:** THIS FUNCTION IS DEPRECATED. It will be removed in a future version. Instructions for updating: This function will only be available through the v1 compatibility library as tf.compat.v1.saved\_model.main\_op.main\_op.

Returns the main op including the group of ops that initializes all variables, initializes local variables and initialize all tables.

#### Returns:

The set of ops to be run as part of the main op upon the load operation.