// Defines the text format for including per-op API definition and

// overrides for client language op code generators.

syntax = "proto3";

package tensorflow;

import "tensorflow/core/framework/attr\_value.proto";

option cc\_enable\_arenas = true;

option java\_outer\_classname = "ApiDefProtos";

option java\_multiple\_files = true;

option java\_package = "org.tensorflow.framework";

option go\_package = "github.com/tensorflow/tensorflow/tensorflow/go/core/framework/api\_def\_go\_proto";

// Used to specify and override the default API & behavior in the

// generated code for client languages, from what you would get from

// the OpDef alone. There will be a set of ApiDefs that are common

// to all client languages, and another set per client language.

// The per-client-language ApiDefs will inherit values from the

// common ApiDefs which it can either replace or modify.

//

// We separate the API definition from the OpDef so we can evolve the

// API while remaining backwards compatible when interpreting old

// graphs. Overrides go in an "api\_def.pbtxt" file with a text-format

// ApiDefs message.

//

// WARNING: Be \*very\* careful changing the API for any existing op --

// you can change the semantics of existing code. These changes may

// need to wait until a major release of TensorFlow to avoid breaking

// our compatibility promises.

message ApiDef {

// Name of the op (in the OpDef) to specify the API for.

string graph\_op\_name = 1;

// If this op is deprecated, set deprecation message to the message

// that should be logged when this op is used.

// The message should indicate alternative op to use, if any.

string deprecation\_message = 12;

// Major version when the op will be deleted. For e.g. set this

// value to 2 if op API should be removed in TensorFlow 2.0 and

// deprecated in versions before that.

int32 deprecation\_version = 13;

enum Visibility {

// Normally this is "VISIBLE" unless you are inheriting a

// different value from another ApiDef.

DEFAULT\_VISIBILITY = 0;

// Publicly visible in the API.

VISIBLE = 1;

// Do not include this op in the generated API. If visibility is

// set to 'SKIP', other fields are ignored for this op.

SKIP = 2;

// Hide this op by putting it into an internal namespace (or whatever

// is appropriate in the target language).

HIDDEN = 3;

}

Visibility visibility = 2;

// If you specify any endpoint, this will replace all of the

// inherited endpoints. The first endpoint should be the

// "canonical" endpoint, and should not be deprecated (unless all

// endpoints are deprecated).

message Endpoint {

// Name should be either like "CamelCaseName" or

// "Package.CamelCaseName". Client-language-specific ApiDefs may

// use a snake\_case convention instead of CamelCase.

string name = 1;

// Set if this endpoint is deprecated. If set to true, a message suggesting

// to use a non-deprecated endpoint instead will be printed. If all

// endpoints are deprecated, set deprecation\_message in ApiDef instead.

bool deprecated = 3;

// Major version when an endpoint will be deleted. For e.g. set this

// value to 2 if endpoint should be removed in TensorFlow 2.0 and

// deprecated in versions before that.

int32 deprecation\_version = 4;

}

repeated Endpoint endpoint = 3;

message Arg {

string name = 1;

// Change the name used to access this arg in the API from what

// is used in the GraphDef. Note that these names in `backticks`

// will also be replaced in the summary & description fields.

string rename\_to = 2;

// Note: this will replace any inherited arg doc. There is no

// current way of modifying arg descriptions (other than replacing

// them entirely) as can be done with op descriptions.

string description = 3;

}

repeated Arg in\_arg = 4;

repeated Arg out\_arg = 5;

// List of original in\_arg names to specify new argument order.

// Length of arg\_order should be either empty to keep current order

// or match size of in\_arg.

repeated string arg\_order = 11;

// Description of the graph-construction-time configuration of this

// Op. That is to say, this describes the attr fields that will

// be specified in the NodeDef.

message Attr {

string name = 1;

// Change the name used to access this attr in the API from what

// is used in the GraphDef. Note that these names in `backticks`

// will also be replaced in the summary & description fields.

string rename\_to = 2;

// Specify a new default value to use for this attr. This default

// will be used when creating new graphs, as opposed to the

// default in the OpDef, which will be used when interpreting old

// GraphDefs.

AttrValue default\_value = 3;

// Note: this will replace any inherited attr doc, there is no current

// way of modifying attr descriptions as can be done with op descriptions.

string description = 4;

}

repeated Attr attr = 6;

// One-line human-readable description of what the Op does.

string summary = 7;

// Additional, longer human-readable description of what the Op does.

string description = 8;

// Modify an existing/inherited description by adding text to the beginning

// or end.

string description\_prefix = 9;

string description\_suffix = 10;

}

message ApiDefs {

repeated ApiDef op = 1;

}