syntax = "proto3";

package tensorflow.error;

option cc\_enable\_arenas = true;

option java\_outer\_classname = "ErrorCodesProtos";

option java\_multiple\_files = true;

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

option go\_package = "github.com/tensorflow/tensorflow/tensorflow/go/core/protobuf/for\_core\_protos\_go\_proto";

// The canonical error codes for TensorFlow APIs.

//

// Warnings:

//

// - Do not change any numeric assignments.

// - Changes to this list should only be made if there is a compelling

// need that can't be satisfied in another way. Such changes

// must be approved by at least two OWNERS.

// - These error codes must match gRPC and protobuf error codes (except for

// DO\_NOT\_USE\_RESERVED\_FOR\_FUTURE\_EXPANSION\_USE\_DEFAULT\_IN\_SWITCH\_INSTEAD\_).

//

// Sometimes multiple error codes may apply. Services should return

// the most specific error code that applies. For example, prefer

// OUT\_OF\_RANGE over FAILED\_PRECONDITION if both codes apply.

// Similarly prefer NOT\_FOUND or ALREADY\_EXISTS over FAILED\_PRECONDITION.

enum Code {

// Not an error; returned on success

OK = 0;

// The operation was cancelled (typically by the caller).

CANCELLED = 1;

// Unknown error. An example of where this error may be returned is

// if a Status value received from another address space belongs to

// an error-space that is not known in this address space. Also

// errors raised by APIs that do not return enough error information

// may be converted to this error.

UNKNOWN = 2;

// Client specified an invalid argument. Note that this differs

// from FAILED\_PRECONDITION. INVALID\_ARGUMENT indicates arguments

// that are problematic regardless of the state of the system

// (e.g., a malformed file name).

INVALID\_ARGUMENT = 3;

// Deadline expired before operation could complete. For operations

// that change the state of the system, this error may be returned

// even if the operation has completed successfully. For example, a

// successful response from a server could have been delayed long

// enough for the deadline to expire.

DEADLINE\_EXCEEDED = 4;

// Some requested entity (e.g., file or directory) was not found.

// For privacy reasons, this code \*may\* be returned when the client

// does not have the access right to the entity.

NOT\_FOUND = 5;

// Some entity that we attempted to create (e.g., file or directory)

// already exists.

ALREADY\_EXISTS = 6;

// The caller does not have permission to execute the specified

// operation. PERMISSION\_DENIED must not be used for rejections

// caused by exhausting some resource (use RESOURCE\_EXHAUSTED

// instead for those errors). PERMISSION\_DENIED must not be

// used if the caller can not be identified (use UNAUTHENTICATED

// instead for those errors).

PERMISSION\_DENIED = 7;

// The request does not have valid authentication credentials for the

// operation.

UNAUTHENTICATED = 16;

// Some resource has been exhausted, perhaps a per-user quota, or

// perhaps the entire file system is out of space.

RESOURCE\_EXHAUSTED = 8;

// Operation was rejected because the system is not in a state

// required for the operation's execution. For example, directory

// to be deleted may be non-empty, an rmdir operation is applied to

// a non-directory, etc.

//

// A litmus test that may help a service implementor in deciding

// between FAILED\_PRECONDITION, ABORTED, and UNAVAILABLE:

// (a) Use UNAVAILABLE if the client can retry just the failing call.

// (b) Use ABORTED if the client should retry at a higher-level

// (e.g., restarting a read-modify-write sequence).

// (c) Use FAILED\_PRECONDITION if the client should not retry until

// the system state has been explicitly fixed. E.g., if an "rmdir"

// fails because the directory is non-empty, FAILED\_PRECONDITION

// should be returned since the client should not retry unless

// they have first fixed up the directory by deleting files from it.

// (d) Use FAILED\_PRECONDITION if the client performs conditional

// REST Get/Update/Delete on a resource and the resource on the

// server does not match the condition. E.g., conflicting

// read-modify-write on the same resource.

FAILED\_PRECONDITION = 9;

// The operation was aborted, typically due to a concurrency issue

// like sequencer check failures, transaction aborts, etc.

//

// See litmus test above for deciding between FAILED\_PRECONDITION,

// ABORTED, and UNAVAILABLE.

ABORTED = 10;

// Operation tried to iterate past the valid input range. E.g., seeking or

// reading past end of file.

//

// Unlike INVALID\_ARGUMENT, this error indicates a problem that may

// be fixed if the system state changes. For example, a 32-bit file

// system will generate INVALID\_ARGUMENT if asked to read at an

// offset that is not in the range [0,2^32-1], but it will generate

// OUT\_OF\_RANGE if asked to read from an offset past the current

// file size.

//

// There is a fair bit of overlap between FAILED\_PRECONDITION and

// OUT\_OF\_RANGE. We recommend using OUT\_OF\_RANGE (the more specific

// error) when it applies so that callers who are iterating through

// a space can easily look for an OUT\_OF\_RANGE error to detect when

// they are done.

OUT\_OF\_RANGE = 11;

// Operation is not implemented or not supported/enabled in this service.

UNIMPLEMENTED = 12;

// Internal errors. Means some invariant expected by the underlying

// system has been broken. If you see one of these errors,

// something is very broken.

INTERNAL = 13;

// The service is currently unavailable. This is a most likely a

// transient condition and may be corrected by retrying with

// a backoff.

//

// See litmus test above for deciding between FAILED\_PRECONDITION,

// ABORTED, and UNAVAILABLE.

UNAVAILABLE = 14;

// Unrecoverable data loss or corruption.

DATA\_LOSS = 15;

// An extra enum entry to prevent people from writing code that

// fails to compile when a new code is added.

//

// Nobody should ever reference this enumeration entry. In particular,

// if you write C++ code that switches on this enumeration, add a default:

// case instead of a case that mentions this enumeration entry.

//

// Nobody should rely on the value (currently 20) listed here. It

// may change in the future.

DO\_NOT\_USE\_RESERVED\_FOR\_FUTURE\_EXPANSION\_USE\_DEFAULT\_IN\_SWITCH\_INSTEAD\_ = 20;

}