

从入门到放弃

冰冻三尺，非一日之寒

目录视图



摘要视图

RSS 订阅

0



(0)

收藏

举报

个人资料



langb2014

原创	粉丝	喜欢	评论
349	1300	285	362



等级： 博客 7

访问量：150万+

积分：1万+

排名：645

Tensorflow的Bazel编程(三)

2017年01月10日 16:18:28

1666人阅读

分类： TF (19)

版权声明：本文为博主原创文章，未经博主允许不得转载。 <http://blog.csdn.net/langb2014/article/details/54314501>

里面的一些规则简单了解一下：

BUILD的一般规则：

Attribute

Description



酒店式公寓出租



戴牙套的年龄



在职研究生取消



手板模型

友情链接

楼燚(yì)航
Rachel Zhang
handong1587
大胖
视觉机器人新浪
视觉机器人
wjmishuai
Deep_Learner
山在岭就在
我爱机器学习
Augusdi
菜鸡一枚
hjimce
machinelearningmastery
liumaolincycle
凌风探梅
无涯
xyy199201
iamzhangzhuping

文章分类

Computer Vision (30)
Machine Learning (33)

	Description
features	<p>List of <i>features</i>. Default is the empty list.</p> <p>Features on a rule modify the <i>features</i> currently enabled on the <code>package</code> level via the <i>features</i> attribute. For example, if the features ['a', 'b'] are enabled on the package level, and a rule's <i>features</i> attribute contains ['a', 'c'], the features enabled for the rule will be 'b' and 'c'.</p>
licenses	<p>List of strings; optional</p> <p>A list of license-type strings to be used for this particular build rule. Overrides the <code>BUILD</code>-file scope defaults defined by the <code>licenses()</code> directive.</p>
data	<p>List of <code>labels</code>; optional</p> <p>The list of files needed by this rule at runtime.</p> <p>Targets named in the <code>data</code> attribute will appear in the <code>*.runfiles</code> area of this rule. This may include data files needed by a binary or library, or other programs needed by it. See the <code>data dependencies</code> section for more information about how to depend on and use data files.</p> <p>Almost all rules permit a <code>data</code> attribute, but where this attribute is not allowed, the specific rule is documented under the specific rule.</p>
visibility	<p>List of <code>labels</code>; optional; default <code>default_visibility</code> from <code>package</code> if specified, else <code>private</code></p> <p>The <code>visibility</code> attribute on a rule controls whether the rule can be used by other packages. Rules are always visible to other rules declared in the same package.</p> <p>There are five forms (and one temporary form) a visibility label can take:</p>

Deep Learning (39)

caffe (96)

数学知识 (13)

C++ (STL) (55)

C++设计模式 (25)

C++(Boost) (12)

Mxnet (1)

Keras (1)

TF (20)

CUDA编程 (18)

Matlab点滴 (46)

图像处理 (10)

UFLDL (1)

Linux (48)

C/C++ (82)

内存篇 (18)

openCV (21)

MatConvNet (1)

MFC小项目 (18)

Python (52)

论文阅读 (16)

gnuplot (画图) (1)

windows (11)

LaTeX (1)

问题集锦 (53)

XML (2)

阅读排行

linux记录系统中常用重要的lo... (23378)

Description

- `["//visibility:public"]`: Anyone can use this rule.
- `["//visibility:private"]`: Only rules in this package can use this rule. Rules in `java/tests/foo/bar` can always use rules in `java/foo/bar`.
- `["//some/package:__pkg__", "//other/package:__pkg__"]`: Only rules in `some/package` and `other/package` (defined in `some/package/BUILD` and `other/package/BUILD`) have access to this rule. Note that sub-packages do not have access to the rule; for example, `some/package/foo:bar` or `//other/package/testing:bla` wouldn't have access. `__pkg__` is a special target and must be used verbatim. It represents all of the rules in the package.
- `["//project:__subpackages__", "//other:__subpackages__"]`: Only rules in packages `project` or `other` or in one of their sub-packages have access to this rule. For example, `//project:rule`, `//project/library:lib` or `//other/testing/internal:munge` are allowed to depend on this rule (but not `//independent:evil`).
- `["//some/package:my_package_group"]`: A `package group` is a named set of package names. Package groups can also grant access rights to entire subtrees, e.g. `//myproj/...`

The visibility specifications of `//visibility:public` and `//visibility:private` can not be combined with any other visibility specifications. A visibility specification may contain a combination of package labels (i.e. `//foo:__pkg__`) and `package_groups`.

If a rule does not specify the visibility attribute, the `default_visibility` attribute of the `package` statement in the BUILD file containing the rule is used (except `exports_files`).

If the default visibility for the package is not specified, the rule is private.

Example:

File `//frobber/bin/BUILD`:

```
# This rule is visible to everyone
cc_binary(
  name = "executable",
  visibility = ["//visibility:public"],
  deps = [":library"],
)
```

python的Tqdm模块	(20471)
OpenCV笔记大集锦	(20047)
Ubuntu14.04安装PyCharm	(18184)
梳理caffe代码layer(五)	(17175)
python中的self理解	(15293)
matlab的Regionprops详解	(14086)
梳理caffe代码math_functions(一)	(14063)
梳理caffe代码blob(三)	(13922)
梳理caffe代码im2col (十七)	(13622)

最新评论

Deconvolutional N...

langb2014 : [reply]qq_33083519[/reply] 已发送

opencv的dnn解析

caiyic : [reply]zhangruyu[/reply] 请问，这个问题有答案了嘛？

Deconvolutional N...

我搬的不是砖而是瓦 : 楼主，求代码。373805562@qq.com

caffe中HDF5层及数据生成

xiaoyangsichuan : 博主请问一下，你有已经转好了的modelnet10的.h5的文件吗，急求，谢谢了

Deconvolutional N...

langb2014 : [reply]u013422403[/reply] 请查收

Deconvolutional N...

langb2014 : [reply]TK9194[/reply] 请查收

Deconvolutional N...

langb2014 : [reply]u013502138[/reply] 请查收

Deconvolutional N...

langb2014 : [reply]peiheng90[/reply] 请查收

Deconvolutional N...

langb2014 : [reply]To_The_One[/reply] 请查收

Deconvolutional N...

Description

```
# This rule is visible only to rules declared in the same package
cc_library(
    name = "library",
    visibility = ["//visibility:private"],
)
```



```
# This rule is visible to rules in package //object and //noun
```

```
cc_library(
    name = "subject",
    visibility = [
        "//noun:__pkg__",
        "//object:__pkg__",
    ],
)
```



```
# See package group "//frobber:friends" (below) for who can
# access this rule.
```

```
cc_library(
    name = thingy,
    visibility = ["//frobber:friends"],
)
```

File **//frobber/BUILD**:

```
# This is the package group declaration to which rule
# //frobber/bin:thingy refers.
```

```
#
```

```
# Our friends are packages //frobber, //fribber and any
# subpackage of //fribber.
```

```
package_group(
    name = "friends",
```

To_Be_The_One : [reply]u011713358[/reply] 你好，文章中的链接失效，能将 Deconvol...

联系我们






请扫描二维码联系客服
webmaster@csdn.net
400-660-0108
QQ客服 客服论坛

关于 招聘 广告服务 百度
©1999-2018 CSDN版权所有
京ICP证09002463号

经营性网站备案信息
网络110报警服务
中国互联网举报中心
北京互联网违法和不良信息举报中心

Description	
	<pre>packages = ["//fribber/...", "//frobber",],)</pre>
c o m p a t i b l e — w i t h d i s t r i b s	<div><div></div><div>0</div><div></div><div></div></div> <p>List of labels; optional</p> <p>The list of environments this rule can be built for, in addition to default-supported environments.</p> <p>This is part of Bazel's soft-launched constraint system, which lets users declare which rules can and cannot depend on each other. For example, externally deployable binaries shouldn't depend on libraries with company-secret code. See ConstraintSemantics for details.</p>
	<p>List of strings; optional</p> <p>A list of distribution-method strings to be used for this particular build rule. Overrides the BUILD-file scope defaults defined by the <code>distribbs()</code> directive.</p>

	Description
dependencies	<p>List of labels; optional</p> <p>A list of dependencies of this rule.</p> <p>The precise semantics of what it means for this rule to depend on another using deps are specific to the kind of this rule, and the rule-specific documentation below goes into more detail. At a minimum, though, the targets named via deps will appear in the *.runfiles area of this rule, if it has one.</p> <p>Most often, a deps dependency is used to allow one module to use symbols defined in another module written in the same programming language and separately compiled. Cross-language dependencies are also permitted in many cases: for example, a java_library rule may depend on C code in a cc_library rule, by declaring the latter in the deps attribute. See the definition of dependencies for more information.</p> <p>Almost all rules permit a deps attribute, but where this attribute is not allowed, this fact is documented under the specific rule.</p>
deprecation	<p>String; optional</p> <p>An explanatory warning message associated with this rule. Typically this is used to notify users that a rule has become obsolete, or has become superseded by another rule, is private to a package, or is perhaps considered harmful for some reason. It is a good idea to include some reference (like a webpage, a bug number or example migration CLs) so that one can easily find out what changes are required to avoid the message. If there is a new target that can be used as a drop in replacement, it is a good idea to just migrate all users of the old target.</p> <p>This attribute has no effect on the way things are built, but it may affect a build tool's diagnostic output. The build tool issues a warning when a rule with a deprecation attribute is depended upon by another rule.</p> <p>Intra-package dependencies are exempt from this warning, so that, for example, building the tests of a deprecated rule does not encounter a warning.</p> <p>If a deprecated rule depends on another deprecated rule, no warning message is issued.</p> <p>Once people have stopped using it, the package can be removed.</p>

	Description
r e s t r i c t e d — t o	<p data-bbox="801 389 1193 421">List of labels; optional</p> <p data-bbox="801 437 1899 510">The list of environments this rule can be built for, <i>instead of</i> default-supported environments. This is part of Bazel's soft-launched constraint system. See compatible_with for details.</p> <div data-bbox="1736 375 1818 678"></div>

Description

List of arbitrary text tags. Tags may be any valid string; default is the empty list.

Tags can be used on any rule. Tags on test and `test_suite` rules are useful for categorizing the tests. Tags on non-test rules are used to control sandboxed execution of `genrules` and `Skylark` actions, and for parsing by humans and/or external tools.

Bazel modifies the behavior of its sandboxing code if it finds the following keywords in the `tags` attribute of any test rule or `genrule`, or the keys of `execution_requirements` for any `Skylark` action.

- `local` keyword results in the action or test never being run remotely or in the `sandbox`. For `genrules` and tests, marking the rule with the `local = 1` attribute has the same effect.
- `block-network` keyword blocks access to the external network from inside the sandbox. In this case, only communication with localhost is allowed.

tags

Tags on tests are generally used to annotate a test's role in your debug and release process. Typically, tags are most useful for C++ and Python tests, which lack any runtime annotation ability. The use of tags and size elements gives flexibility in assembling suites of tests based around codebase check-in policy.

Bazel modifies test running behavior if it finds the following keywords in the `tags` attribute of the test rule:

- `exclusive` keyword will force the test to be run in the "exclusive" mode, ensuring that no other tests are running at the same time. Such tests will be executed in serial fashion after all build activity and non-exclusive tests have been completed. They will also always run locally and thus without sandboxing.
- `manual` keyword will force the test target to not be included in target pattern wildcards (`...:*`, `:all`, etc); the test target will be neither built nor run. It will also be ignored by the `test_suite` rules that do not mention this test explicitly. The only way to build or run such a test is to specify it via an explicit target pattern on the command line.
- `external` keyword will force test to be unconditionally executed (regardless of `--cache_test_results` value).

	Description
testonly	<p>Boolean; optional; default 0 except as noted</p> <p>If 1, only testonly targets (such as tests) can depend on this target.</p> <p>Equivalently, a rule that is not testonly is not allowed to depend on any rule that is testonly.</p> <p>Tests (*_test rules) and test suites (test_suite rules) are testonly by default.</p> <p>This attribute is intended to mean that the target should not be contained in binaries that are released to production.</p> <p>Because testonly is enforced at build time, not run time, and propagates virally through the dependency tree, it should be applied judiciously. For example, stubs and fakes that are useful for unit tests may also be useful for integration tests involving the same binaries that will be released to production and therefore should probably not be marked testonly. Conversely, rules that are dangerous to even link in, perhaps because they unconditionally override normal behavior, should definitely be marked testonly.</p>

visibility是一个重要参数。

有几个函数：

例如：

1、load("//tools/build_rules:build_test.bzl", "build_test")

build_test(name = ...)

2、package(default_deprecation, default_testonly, default_visibility, features)

package(default_visibility = ["//foo:target"])

3、package_group(name, packages, includes)

```
package_group(  
  name = "tropical",  
  packages = [  
    "//fruits/mango",  
    "//fruits/orange",  
    "//fruits/papaya/...",  
  ],  
)
```

The following declarations specify the package groups of a fictional application:

```
package_group(  
  name = "fooapp",  
  includes = [  
    ":controller",  
    ":model",  
    ":view",  
  ],  
)  
  
package_group(  
  name = "model",  
  packages = ["//fooapp/database"],  
)  
  
package_group(  
  name = "view",  
  packages = [  
    "//fooapp/swingui",  
    "//fooapp/webui",  
  ],  
)
```



0



```
package_group(  
    name = "controller",  
    packages = ["//fooapp/algorithm"],  
)
```

4、 licenses(license_types)

restricted

Requires mandatory source distribution.

reciprocal

Allows usage of software freely in unmodified form. Any modifications must be made freely available.

notice

Original or modified third-party software may be shipped without danger nor encumbering other sources. All of the licenses in this category do, however, have an "original Copyright notice" or "advertising clause", wherein any external distributions must include the notice or clause specified in the license.

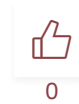
permissive

Code that is under a license but does not require a notice.

unencumbered

Public domain, free for any use.

5、 exports_files([label, ...], visibility, licenses)



```
exports_files(["golden.txt"])
```

6、 glob(include, exclude=[], exclude_directories=1)

```
java_library(  
    name = "mylib",  
    srcs = glob(["*.java"]) + [":gen_java_srcs"],  
    deps = "...",  
)
```

```
genrule(  
    name = "gen_java_srcs",  
    outs = [  
        "Foo.java",  
        "Bar.java",  
    ],  
    ...  
)
```

```
sh_test(  
    name = "mytest",  
    srcs = ["mytest.sh"],  
    data = glob(  
        ["testdata/*.txt"],  
        exclude = ["testdata/experimental.txt"],  
    ),  
)
```

```
java_library(  
    name = "mylib",  
    srcs = glob(  
        ["**/*.java"],
```



0



```
        exclude = ["**/testing/**"],
    ),
)

sh_test(
    name = "mytest",
    srcs = ["mytest.sh"],
    data = glob(["testdata/**/*.*txt"]),
)

# Conveniently, the build language supports list comprehensions.
[genrule(
    name = "count_lines_" + f[:-3], # strip ".cc"
    srcs = [f],
    outs = ["%s-linecount.txt" % f[:-3]],
    cmd = "wc -l $< >$@",
) for f in glob(["*_test.cc"])]

$ bazel query '//foo:all' | sort
//foo:count_lines_a_test
//foo:count_lines_b_test
//foo:count_lines_c_test

7、

select(
    {conditionA: valuesA, conditionB: valuesB, ...},
    no_match_error = "custom message"
)

sh_binary(
    name = "myrule",
    srcs = select({
```



0



```
    ":conditionA": ["myrule_a.sh"],  
    ":conditionB": ["myrule_b.sh"],  
    "//conditions:default": ["myrule_default.sh"]  
  })  
)
```

```
config_setting(  
  name = "windows",  
  values = {  
    "crosstool_top": "//crosstools/windows",  
  },  
)  
cc_binary(  
  name = "multiplatform_app",  
  ...  
  linkopts = select({  
    ":windows": [  
      "-Wl,windows_support1.lib",  
      "-Wl,windows_support2.lib",  
    ],  
    "//conditions:default": [],  
  },  
  ...  
)
```

8、workspace(name = "com_example_project")

- [上一篇](#) Tensorflow的Bazel编程(二)
- [下一篇](#) Tensorflow的Bazel编程(四)



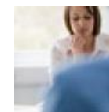
0

体验TeamCity持续集成工具



支持JAVA, .NET, C++, Python, PHP等多种语言., 即刻下载并注册

心理辅导到心语轩心理中心



高效, 专注, 保密, 隐私010-58646524, 数十年心理辅导经验, 帮助解决各种心理困扰

您还没有登录, 请[\[登录\]](#)或[\[注册\]](#)

0



bazel选项的部分翻译，希望给编译tf的童鞋参考

部分翻译了bazel的选项的含义，仅供参考



xizero00 2017年03月03日 13:48 3026

Tensorflow C++ 编译和调用图模型



rockingdingo 2017年07月19日 23:00 15414

最近在研究如何打通tensorflow线下python脚本训练建模，线上生产环境用C++代码直接调用预先训练好的模型完成预测的工作，而不需要用自己写的Inference的函数。因为目前tensorfl...

恭喜：一个公式教你秒懂天下英语

老司机教你一个数学公式秒懂天下英语

源码安装TensorFlow问题总结



idwtwt 2016年08月27日 00:08 4767

1.按教程：bazel build -c opt --jobs 1 //tensorflow/cc:tutorials_example_trainer 出现 INFO: Reading option...

Bazel入门：编译C++项目



elaine_bao

2017年11月29日 18:46

824

官网：<https://www.bazel.build> Github: <https://github.com/bazelbuild/bazel> 最近用到tensorflow的时候遇到了个新的编译工具Ba...

Tensorflow的Bazel编程(一)



langb2014

20

11月09日 14:01

3107

在了解Bazel先看一下 Google Bazel原理篇：Google分布式构建软件之第一部分：访问[Google分布式构建软件之第二部分：构建系统如何工作](#) Google分布式构建软件之第三...



码农不会英语怎么行？英语文档都看不懂！

软件工程出身的英语老师，教你用数学公式读懂天下英文→



Tensorflow的Bazel编程(二)



haima1998

2017年07月26日 16:22

188

转自：<http://blog.csdn.net/langb2014/article/details/54312697> 安装官网：<https://bazel.build/versions/master>

Tensorflow的Bazel编程(二)



langb2014

2017年01月10日 13:38

4231

安装官网：<https://bazel.build/versions/master/docs/tutorial/java.html> Build Java 创建一个java项目，然后 cd /home/...

Tensorflow的Bazel编程(五)



langb2014

2017年01月10日 20:26

1495

只了解一下常用的这几种语言的Rules，这一篇是python的rules。 py_binary py_binary(name, deps, srcs, data, args, compat i...

Bazel (google自动化构建工具)



qq_15807167 2017年04月15日 15:19 676

Bazel 是Google开源出来进行自动化构建工具，相比传统的Makefile，Ant或者Maven,Bazel在速度，可伸展性，灵活性以及对不同程序语言和平台上支持都要更加出色。 Bazel ...

Window10 + tensorflow 安装 (解决Error importing tensorflow, unless you are us...

windows 10 安装 tensorflow



hit1524468 2017年03月26日 15:14 2330



tensorflow 之 bazel安装 & 使用



u010700335 2017年04月06日 19:14 25647

写在文章前面：当一个人从一个领域跨到另一个领域的时候会面临很大的改变，理论不同了，方法变换了，遇到这样挑战的时候，很多都需要长时间去适应和习惯；这种领域的转换其实有三种，一种是理论的改变，一种是方...

Hadoop大数据培训课程

hadoop大数据

百度广告

如何编译运行tensorflow的demo



moyimoyi123 2017年08月06日 14:02 1308

1.安装编译工具bazel，具体可以参照官方教程。 <https://docs.bazel.build/versions/master/install-ubuntu.html> 2. 配置tensor...

Tensorflow学习笔记：Debugging 调试Tensorflow 程序

调试Tensorflow程序 TensorFlow 调试器(tfdbg) 是专门为TensorFlow定制的调试器。在训练或推理的过程中能够让你看到流图的内部结构及状态，由于tensorflow的计...



u010312436 2017年11月30日 10:53 910

Controlling Symbol Visibility (在C/C++中控制符号的可见性)



摘要：在apple的文档中发现一篇关于符号可见性介绍的好文《Controlling Symbol Visibility》，在学习的过程中对关键部分顺带翻译下。...



delphiwcdj 2015年04月23日 20:53 2475



Tensorflow的Bazel编程(三)



langb2014 2017年01月10日 16:18 1666

里面的一些规则简单了解一下：BUILD的一般规则：Attribute Description features List of features. Default is th...

码农不会英语怎么行？英语文档都看不懂！

软件工程出身的英语老师，教你用数学公式读懂天下英文→



Bazel 安装及遇到的问题



luoyi131420 2017年11月20日 20:03 1187

bazel支持的平台有Ubuntu Linux、Mac OS X、Windows等，本文基于Ubuntu14.04下的bazel，其它平台的安装参考官网安装教程。Bazel安装 1.先安...

Tensorflow的Bazel编程(四)



langb2014 2017年01月10日 20:22 1506

C/C++ Rules cc_binary(name, deps, srcs, data, args, compatible_with, copts, defines, deprecation,...

在linux（centos）上从源码安装tensorflow



mafeiyu80 2016年05月13日 19:16 7901

我参考英文文档在centos7虚拟机上安装了tensorflow：<https://github.com/tensorflow/tensorflow/blob/master/tensorflow/g...>



0

Tensorflow在windows7下的安装



u014157632 2016年09月09日 16:16 2621

折腾了一天，终于把Tensorflow装好了。参照多篇教程，走了一些弯路，也是不容易。这次安装，得到的启示就是，还是用Linux吧说一下我的安装过程吧，我的电脑是windows7系统，64位。 1...



linux下使用bazel编译tensorflow安卓源码



ulike_MFY 2017年12月13日 10:45 156

决定学习tensorflow了，听了实验室同学的建议，安装个linux系统，然后先把安卓源码编译一下。安装AnacondaAnaconda集成了python、tensorflow、spyder(pyt...