NAME

git-check-attr - Display gitattributes information

SYNOPSIS

git check-attr [--source <tree-ish>] [-a | --all | <attr>...] [--] <pathname>...
git check-attr --stdin [-z] [--source <tree-ish>] [-a | --all | <attr>...]

DESCRIPTION

For every pathname, this command will list if each attribute is *unspecified*, *set*, or *unset* as a gitattribute on that pathname.

OPTIONS

-a, --all

List all attributes that are associated with the specified paths. If this option is used, then *unspecified* attributes will not be included in the output.

--cached

Consider **.gitattributes** in the index only, ignoring the working tree.

--stdin

Read pathnames from the standard input, one per line, instead of from the command-line.

-z

The output format is modified to be machine-parsable. If **--stdin** is also given, input paths are separated with a NUL character instead of a linefeed character.

--source=<tree-ish>

Check attributes against the specified tree-ish. It is common to specify the source tree by naming a commit, branch or tag associated with it.

--

Interpret all preceding arguments as attributes and all following arguments as path names.

If none of **--stdin**, **--all**, or **--** is used, the first argument will be treated as an attribute and the rest of the arguments as pathnames.

OUTPUT

The output is of the form: <path> COLON SP <attribute> COLON SP <info> LF

unless **-z** is in effect, in which case NUL is used as delimiter: <path> NUL <attribute> NUL <info> NUL

<path> is the path of a file being queried, <attribute> is an attribute being queried and <info> can be either:

unspecified
: when the attribute is not defined for the path.

unset
: when the attribute is defined as false.

set
: when the attribute is defined as true.

<value>
: when a value has been assigned to the attribute.

Buffering happens as documented under the **GIT_FLUSH** option in **git**(1). The caller is responsible for avoiding deadlocks caused by overfilling an input buffer or reading from an empty output buffer.

EXAMPLES

In the examples, the following *.gitattributes* file is used:

```
*.java diff=java -crlf myAttr
NoMyAttr.java !myAttr
README caveat=unspecified
```

- a single attribute:

```
$ git check-attr diff org/example/MyClass.java
org/example/MyClass.java: diff: java
```

- multiple attributes for a file:

```
$ git check-attr crlf diff myAttr -- org/example/MyClass.java
org/example/MyClass.java: crlf: unset
org/example/MyClass.java: diff: java
org/example/MyClass.java: myAttr: set
```

- all attributes for a file:

```
$ git check-attr --all -- org/example/MyClass.java
org/example/MyClass.java: diff: java
org/example/MyClass.java: myAttr: set
```

- an attribute for multiple files:

```
$ git check-attr myAttr -- org/example/MyClass.java org/example/NoMyAttr.java
org/example/MyClass.java: myAttr: set
org/example/NoMyAttr.java: myAttr: unspecified
```

- all values are equally unambiguous:

```
$ git check-attr caveat README
README: caveat: unspecified
```

SEE ALSO

gitattributes(5).

GIT

Part of the **git**(1) suite