

git_comments:

git_commits:

1. **summary:** PARQUET-1255: Fix error message when PARQUET_TEST_DATA isn't defined
message: PARQUET-1255: Fix error message when PARQUET_TEST_DATA isn't defined Author: Antoine Pitrou <antoine@python.org> Closes #448 from pitrou/PARQUET-1255 and squashes the following commits: b162511 [Antoine Pitrou] PARQUET-1255: Fix error message when PARQUET_TEST_DATA isn't defined Change-Id: I8f900b59e7fa6e41171e004c61dfa247201047ef

github_issues:

github_issues_comments:

github_pulls:

1. **title:** ARROW-719: [GLib] Release source archive
body: I don't know about his approach is good but it will help you to consider what approach is better.

github_pulls_comments:

1. It seems Apache Thrift (which also has a GLib-based C implementation) makes source releases requiring autotools. Could you tell us more about the use case? One issue with creating a signed release archive for arrow-glib is that it's an extra step to validate the additional artifact when making releases. The `make dist` source release could be thought of as a sort of "binary release" because it is not strictly a source release as it includes files generated by autotools, so it's possible we could publish the arrow-glib tarball as a separate artifact _after_ the main source release happens. @ryanblue @tomwhite sorry to pull you into this thread, but could you advise on what other ASF projects have done in these kinds of situations? Thanks
2. > It seems Apache Thrift (which also has a GLib-based C implementation) makes source releases requiring autotools. Really? <https://www.apache.org/dist/thrift/0.10.0/thrift-0.10.0.tar.gz> includes `configure` (= GNU Autotools isn't required for users). `configure` isn't included in <https://github.com/apache/thrift/>. > Could you tell us more about the use case? Sure. I should add a link to JIRA: <https://issues.apache.org/jira/browse/ARROW-719> If we bundle generated `configure` and related files, users don't need autoconf-archive. The autoconf-archive dependency may confuse users like you did at <https://github.com/apache/arrow/pull/418#issuecomment-288415708> If user's platform is old and packaged GNU Autotools is old, the user may fail to generate `configure`. If we bundle generated `configure` and related files, we can avoid the case. I hope that this description helps you.
3. > Really? <https://www.apache.org/dist/thrift/0.10.0/thrift-0.10.0.tar.gz> includes configure oops, sorry about that! I was looking in lib/c_glib. Is there a way we can generate only a single tarball in the source release, then, like Thrift?
4. > Is there a way we can generate only a single tarball in the source release, then, like Thrift? How about the following? ``text git archive ... | tar xf - ... cd apache-arrow-XXX/c_glib make dist mv apache-arrow-glib-XXX.tar.gz ../ cd - rm -rf apache-arrow-XXX/ git archive ... | tar xf - tar xf apache-arrow-glib-XXX.tar.gz rm -rf apache-arrow-XXX/c_glib mv apache-arrow-glib-XXX apache-arrow-XXX/c_glib tar czf apache-arrow-XXX.tar.gz apache-arrow-XXX ``
5. **body:** That seems fine to me -- do you want to put this in the 02-source.sh? @xhochy? If cross-platform release becomes an issue we can always add a Dockerfile, I don't think it's an issue for the moment since my understanding is that autotools does not generate a platform specific configure file
label: code-design
6. **body:** That's fine, put in in 02 and if it's really a problem, I'll spent the 10min to add a Docker option if really necessary.
label: code-design
7. I've force pushed commit that uses 02 instead of creating 03.

github_pulls_reviews:

1. Not sure how other feel about this but I would like to have this in some kind of reproducible environment (e.g. a docker container) so that releases are independent of the OS they're made from.
2. I think so too.

jira_issues:

1. **summary:** [C++] Exceptions thrown in some tests

description: Some tests (not all) throw a basic_string exception. Example: {code} \$./debug/reader-test
Running main() from gtest_main.cc [=====] Running 11 tests from 4 test cases. [-----] Global test environment set-up. [-----] 7 tests from TestAllTypesPlain [RUN]
TestAllTypesPlain.NoopConstructDestruct unknown file: Failure C++ exception with description "basic_string::_S_construct null not valid" thrown in SetUp(). [FAILED]
TestAllTypesPlain.NoopConstructDestruct (0 ms) [RUN] TestAllTypesPlain.TestBatchRead unknown file: Failure C++ exception with description "basic_string::_S_construct null not valid" thrown in SetUp(). [FAILED] TestAllTypesPlain.TestBatchRead (0 ms) [RUN]
TestAllTypesPlain.TestFlatScannerInt32 unknown file: Failure C++ exception with description "basic_string::_S_construct null not valid" thrown in SetUp(). [FAILED]
TestAllTypesPlain.TestFlatScannerInt32 (0 ms) [RUN] TestAllTypesPlain.TestSetScannerBatchSize unknown file: Failure C++ exception with description "basic_string::_S_construct null not valid" thrown in SetUp(). [FAILED] TestAllTypesPlain.TestSetScannerBatchSize (0 ms) [RUN]
TestAllTypesPlain.DebugPrintWorks unknown file: Failure C++ exception with description "basic_string::_S_construct null not valid" thrown in SetUp(). [FAILED]
TestAllTypesPlain.DebugPrintWorks (0 ms) [RUN] TestAllTypesPlain.ColumnSelection unknown file: Failure C++ exception with description "basic_string::_S_construct null not valid" thrown in SetUp(). [FAILED] TestAllTypesPlain.ColumnSelection (0 ms) [RUN]
TestAllTypesPlain.ColumnSelectionOutOfRange unknown file: Failure C++ exception with description "basic_string::_S_construct null not valid" thrown in SetUp(). [FAILED]
TestAllTypesPlain.ColumnSelectionOutOfRange (0 ms) [-----] 7 tests from TestAllTypesPlain (0 ms total) [-----] 2 tests from TestLocalFile [RUN] TestLocalFile.FileClosedOnDestruction unknown file: Failure C++ exception with description "basic_string::_S_construct null not valid" thrown in SetUp(). [FAILED] TestLocalFile.FileClosedOnDestruction (0 ms) [RUN]
TestLocalFile.OpenWithMetadata unknown file: Failure C++ exception with description "basic_string::_S_construct null not valid" thrown in SetUp(). [FAILED]
TestLocalFile.OpenWithMetadata (0 ms) [-----] 2 tests from TestLocalFile (0 ms total) [-----] 1 test from TestFileReaderAdHoc [RUN] TestFileReaderAdHoc.NationDictTruncatedDataPage unknown file: Failure C++ exception with description "basic_string::_S_construct null not valid" thrown in the test body. [FAILED] TestFileReaderAdHoc.NationDictTruncatedDataPage (1 ms) [-----] 1 test from TestFileReaderAdHoc (1 ms total) [-----] 1 test from TestJSONWithLocalFile [RUN]
TestJSONWithLocalFile.JSONOutput unknown file: Failure C++ exception with description "basic_string::_S_construct null not valid" thrown in the test body. [FAILED]
TestJSONWithLocalFile.JSONOutput (0 ms) [-----] 1 test from TestJSONWithLocalFile (0 ms total) [-----] Global test environment tear-down [=====] 11 tests from 4 test cases ran. (1 ms total) [PASSED] 0 tests. [FAILED] 11 tests, listed below: [FAILED]
TestAllTypesPlain.NoopConstructDestruct [FAILED] TestAllTypesPlain.TestBatchRead [FAILED]
TestAllTypesPlain.TestFlatScannerInt32 [FAILED] TestAllTypesPlain.TestSetScannerBatchSize [FAILED] TestAllTypesPlain.DebugPrintWorks [FAILED] TestAllTypesPlain.ColumnSelection [FAILED] TestAllTypesPlain.ColumnSelectionOutOfRange [FAILED]
TestLocalFile.FileClosedOnDestruction [FAILED] TestLocalFile.OpenWithMetadata [FAILED]
TestFileReaderAdHoc.NationDictTruncatedDataPage [FAILED] TestJSONWithLocalFile.JSONOutput
11 FAILED TESTS {code}

jira_issues_comments:

1. This is the magic way of telling you to set {{PARQUET_TEST_DATA}} to the correct path to the {{data/}} folder than contains the reference files.
2. Hmm, yes, of course, it works once I set PARQUET_TEST_DATA... :-/
3. pitrou opened a new pull request #448: PARQUET-1255: Fix error message when PARQUET_TEST_DATA isn't defined URL: <https://github.com/apache/parquet-cpp/pull/448> -----
----- This is an automated message from the Apache Git Service. To

respond to the message, please log on GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org

4. xhochy closed pull request #448: PARQUET-1255: Fix error message when PARQUET_TEST_DATA isn't defined URL: <https://github.com/apache/parquet-cpp/pull/448> This is a PR merged from a forked repository. As GitHub hides the original diff on merge, it is displayed below for the sake of provenance: As this is a foreign pull request (from a fork), the diff is supplied below (as it won't show otherwise due to GitHub magic):

```
diff --git a/src/parquet/arrow/arrow-reader-writer-test.cc b/src/parquet/arrow/arrow-reader-writer-test.cc
index 8051ff17..dae2e8f9 100644 --- a/src/parquet/arrow/arrow-reader-writer-test.cc
+++ b/src/parquet/arrow/arrow-reader-writer-test.cc
@@ -39,6 +39,8 @@ #include
"parquet/file_writer.h" + #include "parquet/util/test-common.h" + #include "arrow/api.h" #include
"arrow/test-util.h" #include "arrow/type_traits.h" @@ -2098,8 +2100,7 @@
TEST(TestImpalaConversion, NanosecondToImpala) { TEST(TestArrowReaderAdHoc,
Int96BadMemoryAccess) { // PARQUET-995 - const char* data_dir =
std::getenv("PARQUET_TEST_DATA"); - std::string dir_string(data_dir); + std::string
dir_string(test::get_data_dir()); std::stringstream ss; ss << dir_string << "/" << "alltypes_plain.parquet";
@@ -2119,7 +2120,7 @@ class TestArrowReaderAdHocSpark std::tuple<std::string,
std::shared_ptr<::DataType>>> {}}; TEST_P(TestArrowReaderAdHocSpark, ReadDecimals) { -
std::string path(std::getenv("PARQUET_TEST_DATA")); + std::string path(test::get_data_dir());
std::string filename; std::shared_ptr<::DataType> decimal_type; diff --git a/src/parquet/reader-test.cc
b/src/parquet/reader-test.cc
index c536fdbcb..d628f472 100644 --- a/src/parquet/reader-test.cc
+++ b/src/parquet/reader-test.cc
@@ -30,6 +30,7 @@ #include "parquet/file_reader.h" #include
"parquet/printer.h" #include "parquet/util/memory.h" + #include "parquet/util/test-common.h" using
std::string; @@ -37,10 +38,8 @@ namespace parquet { using ReadableFile = ::arrow::io::ReadableFile; -
const char* data_dir = std::getenv("PARQUET_TEST_DATA"); - std::string alltypes_plain() { -
std::string dir_string(data_dir); + std::string dir_string(test::get_data_dir()); std::stringstream ss; ss <<
dir_string << "/" << "alltypes_plain.parquet"; @@ -48,7 +47,7 @@ std::string alltypes_plain() { }
std::string nation_dict_truncated_data_page() { - std::string dir_string(data_dir); + std::string
dir_string(test::get_data_dir()); std::stringstream ss; ss << dir_string << "/" << "nation.dict-
malformed.parquet"; @@ -171,7 +170,7 @@ TEST_F(TestAllTypesPlain,
ColumnSelectionOutOfRange) { class TestLocalFile : public ::testing::Test { public: void SetUp() { -
std::string dir_string(data_dir); + std::string dir_string(test::get_data_dir()); std::stringstream ss; ss <<
dir_string << "/" diff --git a/src/parquet/util/test-common.h b/src/parquet/util/test-common.h
index
ebf48516..22b748e7 100644 --- a/src/parquet/util/test-common.h
+++ b/src/parquet/util/test-common.h
@@ -23,6 +23,7 @@ #include <random> #include <vector> + #include "parquet/exception.h" #include
"parquet/types.h" using std::vector; @@ -35,6 +36,20 @@ typedef ::testing::Types<BooleanType,
Int32Type, Int64Type, Int96Type, FloatType DoubleType, ByteArrayType, FLBAType> ParquetTypes;
+class ParquetTestException : public parquet::ParquetException { + using
ParquetException::ParquetException; +}; + +const char* get_data_dir() { + const auto result =
std::getenv("PARQUET_TEST_DATA"); + if (!result || !result[0]) { + throw ParquetTestException( +
"Please point the PARQUET_TEST_DATA environment " + "variable to the test data directory"); + } +
return result; +} + template <typename T> static inline void assert_vector_equal(const vector<T>& left,
const vector<T>& right) { ASSERT_EQ(left.size(), right.size()); -----
----- This is an automated message from the Apache Git Service. To respond to the message,
please log on GitHub and use the URL above to go to the specific comment. For queries about this
service, please contact Infrastructure at: users@infra.apache.org
```
5. Issue resolved by pull request 448 [<https://github.com/apache/parquet-cpp/pull/448>]
6. majetideepak commented on issue #448: PARQUET-1255: Fix error message when PARQUET_TEST_DATA isn't defined URL: <https://github.com/apache/parquet-cpp/pull/448#issuecomment-377202205> @xhochy I tried to merge this commit yesterday and followed the instructions in `dev/committers-guide.md` at `2. Add your gpg key to the Apache Parquet {format,mr,cpp} KEYS file:` I get the following error: `` \$ git clone git@github.com:apache/parquet-cpp.git Cloning into 'parquet-cpp'... Permission denied (publickey). fatal: Could not read from remote repository. Please make sure you have the correct access rights and the repository exists. `` Can you help me? Thanks! ----- This is an automated message from the Apache Git Service. To respond to the message, please log on GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org
7. xhochy commented on issue #448: PARQUET-1255: Fix error message when PARQUET_TEST_DATA isn't defined URL: <https://github.com/apache/parquet-cpp/pull/448#issuecomment-377208606>

@majetideepak Make sure your ASF and github account is correctly linked: <https://gitbox.apache.org/> (you will need to activate 2 factor auth in Github). -----

This is an automated message from the Apache Git Service. To respond to the message, please log on GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org

8. majetideepak commented on issue #448: PARQUET-1255: Fix error message when PARQUET_TEST_DATA isn't defined URL: <https://github.com/apache/parquet-cpp/pull/448#issuecomment-377234307> I missed this step! Thank you! On Thu, Mar 29, 2018 at 7:40 AM, Uwe L. Korn <notifications@github.com> wrote: > @majetideepak <<https://github.com/majetideepak>> Make sure your ASF and > github account is correctly linked: <https://gitbox.apache.org/> (you will > need to activate 2 factor auth in Github). > > — > You are receiving this because you were mentioned. > Reply to this email directly, view it on GitHub > <<https://github.com/apache/parquet-cpp/pull/448#issuecomment-377208606>>, > or mute the thread > <<https://github.com/notifications/unsubscribe-auth/AE2vV4L6IqMoBeE2q5Rbw2vh6O2uth8nks5tjMgugaJpZM4S87n4>> > . > -- regards, Deepak Majeti ----- This is an automated message from the Apache Git Service. To respond to the message, please log on GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org