

Test Start, 2012-09-03T14:19:22
Test Stop, 2012-09-03T14:20:50
Elapsed Time, 00:01:28
Software revision, v5.1.0.1 <6819:7397> (Jul 23 2012 10:42:34)
Script URL, https://colorkinetics.svn.beanstalkapp.com/cypher/.../MSLM%20-%20Graze%20G2%20-%20Color.xml
Script Rev, 259
Script Date, 2012-06-04 14:54:54 -0400 (Mon, 04 Jun 2012)
Script Author, ggong
Script Local Path, D:/morpheus/George/QStringListDebug/GetFileModificationDate/MSLM - SkyRibbon - Color.xml
Script Last Modified, 2012-09-21T13:09:39
Computer ID, ytech-b86168eee.
Operating System, Windows XP
Part Number, 123-000079-40
Device Name, COLORGRAZE EC POWERCORE; 1FT
Operator, 123
OVERALL RESULT, PASS

***** , ***** , *****
Application DUT Info
***** , ***** , *****

***** , ***** , *****
Results
***** , ***** , *****

***** , ***** , *****
Rejection Criteria
***** , ***** , *****

***** , ***** , *****
Test Log
***** , ***** , *****

- Comment [GG1]:** FTE uses format 2012/09/19,10:50:52. I think it will be easier to use QDateTime.toString(Qt::ISODate)
- Comment [GG2]:** I prefer Paul's use of version/SVN version/Build date to my simplified method of just date.
- Comment [GG3]:** <SVN_Url NoChecksum="true"> \$URL: \$ </SVN_Url>
- Comment [GG4]:** <SVN_Rev NoChecksum="true"> \$Rev: \$ </SVN_Rev>
- Comment [GG5]:** <SVN_Date NoChecksum="true"> \$Date: \$ </SVN_Date>
- Comment [GG6]:** <SVN_Author NoChecksum="true"> \$Author: \$ </SVN_Author>
- Comment [GG7]:** While the SVN info is good for capturing the last checked in script, it is possible that someone edits the script and does not check it in. This information will help us determine that.
QString fileName = "D:\\morpheus\\George\\QStringListDebug\\GetFileModificationDate\\MSLM - SkyRibbon - Color.xml";

QFileInfo scriptFile(fileName);

if (scriptFile.exists()) {
 qDebug() <<
 scriptFile.filePath();
 qDebug() <<
 scriptFile.lastModified().toString(Qt::ISODate);
}
- Comment [GG8]:** Useful for determining which line the information comes from.
m_computerName = QString("%1.%2").arg(QHostInfo::localHostName());
- Comment [GG9]:** QString LinearTestEngineMainApp::getOperatingSystem()
- Comment [GG10]:** From xml selected SKU entry
<SKU numSegments="1" value="123-456789-00" name="LINEAR, 1FT"/>
- Comment [GG11]:** void LinearTestEngineMainApp::getOperator()
- Comment [GG12]:** Stealing Paul's format but I want to all caps this to draw the eye to it easier.
- Comment [GG13]:** Under this section we insert application specific headers (SNs, MAC/IP/SN, etc.)
- Comment [GG14]:** Helps quickly identify why a parameter failed. I will try to add this into the linear test engine. However, to economize space, I think I would rather just display results and limits of
- Comment [GG15]:** Sometimes a screenshot is not enough. Whatever is logged in the application log window during this DUT test is copied here

Page 1: [1] Comment [GG8]	usb10532	9/21/2012 1:31:00 PM
---------------------------	----------	----------------------

Useful for determining which line the information comes from.
m_computerName = QString("%1.%2").arg(QHostInfo::localHostName())
.arg(QHostInfo::localDomainName());

Page 1: [2] Comment [GG14]	usb10532	9/21/2012 1:34:00 PM
----------------------------	----------	----------------------

Helps quickly identify why a parameter failed. I will try to add this into the linear test engine. However, to economize space, I think I would rather just display results and limits of the failures rather than parse out everything. This section should not need to be parsed by any database importer.

Page 1: [3] Comment [GG15]	usb10532	9/21/2012 1:35:00 PM
----------------------------	----------	----------------------

Sometimes a screenshot is not enough. Whatever is logged in the application log window during this DUT test is copied here. Only useful for debugging. This section should not need to be parsed by any database importer.