[NTGSUI-209048] [ELVIS] [Stability] System Reset after switch HU On (IGN_OFF) and select System Information Created: 15/Jul/20 Updated: 23/Jul/20

Status:	In Progress
Project:	NTG6-UI Software
Component/s:	01 Widgets
Affects Version/s:	UI2020 E020.2
Fix Version/s:	None

Type:	Elvis Bug	**Priority:**	Blocker
Reporter:	Oleksandr Babenko	**Assignee:**	Thomas Victor
Resolution:	Unresolved	**Votes:**	0
Labels:	FUP2_TOP, Stability, new150720		
Remaining Estimate:	1 day, 1 hour		
Time Spent:	Not Specified		
Original Estimate:	1 day, 1 hour		

Attachments:	abstractModel_MBProxyEntryModel.png frame_17_locals.png frame_17_locals.png frame_17_this_vptr.png frame_23_this.png frame_24_locals.png fullStackTrace.log stack.txt
Issue Links:	**Cloners** clones MMXXSYHA-30943 [Stability] System Reset after switch... In Progress
Rank:	2|i0cwlz:
Release Package:	UI2020 FreshUp2
Release Build:	none
ELVIS-ID:	2980399

Description

Reported By = EBernatovic
System = HU7 Premium TV ECE_STAR3_M658
Functiongroup = Stability
TOP Flag = FUP2_TOP
Stability Category = Reset

[SW Label] PL_NTG7_E020.201_20285AC3
[Timestamp] 2020-07-13 04:07:58 PM
[Date] 07/13/2020
[Time] 16:07
[Location] DAI Dummy
[Contact Info] Eugen.Bernatovic@harman.com

[Precondition(s)]

- Test car W223-2149 (BB OR 8607) parked in AFO CarLab
- IGN_OFF;
- HU_OFF;
- System language: German;
- FG Application: Navigation Map Display;
- Profile "2149 withMusic PW1234" was active;
- Door on driver side is open;

[Action(s)]

- Switch HU ON by ON/OFF button;
- Switch to Home menu;
- Select Settings -> Info -> System Information (to check installed SW version)

[Observed]

- Suddenly System Reset occurs;
- HU restarted and system trigger was written;

[Expected]

- No system resets;

[Recovery] Recover without interaction

[Notes] Probably reset was caused by short change HU OFF- ON and Profile log in/out.

[Customer Impact] Visible for normal user
Fails/#Tests 1/1
[Occurence in > 1 LC] No

[Versions]
HU Versions:
HW Sample: D5
HW Variant: M658
Navi DB: DB_MBW223SO_ECE_ECE_R_1120_E70.0301
SW Label: PL_NTG7_E020.201_20285AC3
SW Version: E020.2
System: HU Premium TV ECE
Test Location: Böblingen
Traces for this session are uploaded here: \\tdt-traces.harman.com\TDT_DAI\NTG7\SysTest\2020-07\14-Tue\Car\DAI Dummy_EBernatovic_10-57

HTA Reloaded Internal ID: 07/14/2020 13:39:22_HIFIP120_EBernatovic_6363
Ticket created from ELVIS API v1.20

Comments

Comment by Dmitry Kosolapov [16/Jul/20]

Tuan Tu , could you please check the core dump.

and please show me how you handle this file:

core_20200714-102953(core.504.1016.1014.6.1594649050.appman.lz4)_

Comment by Tuan Tu [16/Jul/20]

Hi Valerii Shaferman, could you please check this from Media?
I cannot fully extract the core file. But below is what I can find out:

1. In DLT, before crash is a message from "QmlChunkedMediaList::setAvailableRange"

```
313455 2020/07/13 16:06:09.541194 2941.4178 191 ECU1 UI IF1 504 log info verbose 1 [daimler.if1verbose] 2987 MediaList#44444:EV :
availableRangeChanged(range: AvailableRange(left: 0, right: 0), clientID: 0) [unknown:0]
313456 2020/07/13 16:06:09.541194 2941.4179 45 ECU1 UI MPL 504 log info verbose 1 [daimler.hf.mediaplayer] void
entertainment::player::QmlChunkedMediaList::setAvailableRange(const entertainment::player::AvailableRange&) 0 , 0 [unknown:0]
313466 2020/07/13 16:06:09.541194 2941.4204 46 ECU1 UI MPL 504 log info verbose 1 [daimler.hf.mediaplayer] getItems offset= 0 length=
1 [unknown:0]
313467 2020/07/13 16:06:09.541194 2941.4207 192 ECU1 UI IF1 504 log info verbose 1 [daimler.if1verbose] 150273 MediaList#44444:RQ :
getItems(offset: 0, length: 1, clientID: 1) [unknown:0]
313481 2020/07/13 16:06:09.541194 2941.4234 193 ECU1 UI IF1 504 log info verbose 1 [daimler.if1verbose] 2988
CurrentPlaylistControl#1:EV : playbackStatusChanged(status: PLAYING, clientID: 0) [unknown:0]
313484 2020/07/13 16:06:09.541194 2941.4236 194 ECU1 UI IF1 504 log info verbose 1 [daimler.if1verbose] 2990 MediaList#44444:EV :
availableRangeChanged(range: AvailableRange(left: 0, right: 2), clientID: 0) [unknown:0]
313485 2020/07/13 16:06:09.541194 2941.4236 47 ECU1 UI MPL 504 log info verbose 1 [daimler.hf.mediaplayer] void
entertainment::player::QmlChunkedMediaList::setAvailableRange(const entertainment::player::AvailableRange&) 0 , 2 [unknown:0]
315320 2020/07/13 16:06:10.020662 2941.8580 22 ECU1 UI GEN 504 log fatal verbose 1 [] *** process /opt/qt-5.12/bin/appman (504)
crashed *** [unknown:0]
315321 2020/07/13 16:06:10.020662 2941.8581 23 ECU1 UI GEN 504 log fatal verbose 1 [] > why: uncaught signal 11 (Segmentation fault)
[unknown:0]
```

2. In the trace fullStackTrace.log , there is stack related to "QAbstractItemModel::endInsertRows". This may be related to above media list (just my guess).

3.The last track is RSU_RemoteUpdateIF2Broker, but we don't use this IF2 at all.

Therefore, I would like to ask you to check from Media side.
Thanks!!

Comment by Oleksii Kondratenko [17/Jul/20]

Radhika Jammalamadaka please check from widgets side, the crash happened in MBProxyEntryModel:

We use our QmlChunkedMediaList model to set a source for MBProxyEntryMode:

gui/apps/Media/AppViews/Media/MediaCoverFlow.qml:118
model: _internal.isCurrentPlaylistNotEmpty ? currentPlaylist : 1

Model is defined here:

gui/apps/Entertainment/components/CoverFlow/CoverFlowPathView.qml:158

MBProxyEntryModel {

id: _proxyModel
delegate: CoverFlowEntry
{ data: model && (model.modelData !== undefined ? model.modelData : model.data !== undefined ? model.data : model) }

}

And that is the model where is actual crash happened, see frame 16 in stack.txt , also on the below screenshot it is clear that parent of this model is CoverFlowPathView

Actual reason is that in frame 16:

QAbstractItemModelPrivate::rowsInserted
{
QVector<QPersistentModelIndexData *> persistent_moved = persistent.moved.pop();

persistent.moved is a stack and it is already empty at the moment when pop was called, pop function should not be called on empty stack, it will not directly cause crash on Release build, but further initialization of QVector caused a crash.

Please check why the above mentioned stack was empty.
In QmlChunkedMediaList we do calls like this:

models/src/Entertainment/Player/qmlchunkedmedialist.cpp:100

beginInsertRows(QModelIndex(), m_availableRange.Right() + 1 - m_availableRange.Left(), availableRange.Right() - m_availableRange.Left());
...
endInsertRows();

Comment by Thomas Victor [21/Jul/20]

The crash appears to be happening directly in Qt code as the result of operations on QmlChunkedMediaList model. Possibly due to back to back insertions in the setAvailableRange method:

```
// Extension. The new range is guaranteed to be wider than existing one.
        if (m_availableRange.Right() != availableRange.Right()) {
            // Update right part of the list.
            beginInsertRows(QModelIndex(), m_availableRange.Right() + 1 - m_availableRange.Left(), availableRange.Right() -
m_availableRange.Left());
            increaseSize(availableRange.Right() - m_availableRange.Right());
            m_availableRange.setRight(availableRange.Right());
            endInsertRows();
        }
        if (m_availableRange.Left() != availableRange.Left()) {
            // Update left part of the list.
            beginInsertRows(QModelIndex(), 0, m_availableRange.Left() - availableRange.Left() - 1);
            m_availableRange.setLeft(availableRange.Left());
            endInsertRows();
            emit firstEntryIndexChanged();
        }
```

This does not appear to be a problem with MBProxyEntryModel and the code never enters MBProxyEntryModel in the stack trace. Line 16 as mentioned in above comment is within QAbstractItemModelPrivate a Qt class that QmlChunkedMediaList indirectly extends.

Comment by Thomas Victor [21/Jul/20]

Also crash is not reproducible on my bench with above steps to reproduce, but since it seems media related, steps to reproduce were probably a coincidence.

Comment by Thomas Victor [21/Jul/20]

Oleksii Kondratenko Please reconsider from media side, stack trace does not enter MBProxyEntryModel code as explained above.

Comment by Oleksii Kondratenko [22/Jul/20]

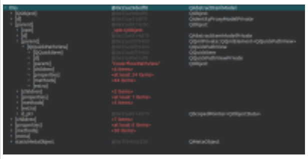
Thomas Victor sorry, but it is not true. Please look carefully into my explanations above.
QmlChunkedMediaList definitely inherits QAbstractItemModel and one of the QAbstractItemModel::endInsertRows calls in stack definitely belongs to our model, but it is frame 23:

23 QAbstractItemModel::endInsertRows qabstractitemmodel.cpp 2752 0x7f83d596b0
Pay attention to 'this' address and parent of the object:

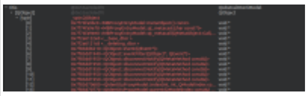
On frame 23 address is the same:

this @0x55adc0d320

And now to the frame 17:



This has different address, thus different instance. It is linked by parents to QQuickPathView which parent is CoverFlowPathView (see CoverFlowPathView.qml:120).
And beside that:



Please take a look.

Comment by Kostiantyn Cherniaiev [22/Jul/20]

Andreas Luft, FYI.

Comment by Roman Novikov [22/Jul/20]

My two cents to this investigation. Apparently the MBProxyEntryModel code is not related to this issue since the presented stack trace neither has a single invocation of methods of the MBProxyEntryModel nor any traces of its prior calls that could potentially lead to the described behaviour. The problem seems to be on the Qt side: all virtual method calls and consequent signal emits and slots invocations happen inside the QAbstractItemModel and QIdentityProxyModel (the base class of MBProxyEntryModel). One of the possible causes why the internal stack of QAbstractItemModel could be empty when endInsertRows called is that either beginInsertRows has not been called before. This may happen if the corresponding signal of theQAbstractItemModel was not connected to the slot of QIdentityProxyModel at the moment of the first call.

Comment by Thomas Victor [23/Jul/20]

I agree with Roman Novikov , I see no evidence that the stack trace ever enters into MBProxyEntryModel code. It does go into QIdentityProxyModel code, but this is due to automatic connection between QIdentityProxyModel and it's source model...

In this case the source model calls endInsertRows, and QIdentityProxyModel takes over automatically:

```
connect(sourceModel(), SIGNAL(rowsInserted(QModelIndex,int,int)),385
SLOT(_q_sourceRowsInserted(QModelIndex,int,int)));
```

However it seems as Roman is correct in order for persisent.moved stack to be empty in the call

```
QAbstractItemModelPrivate::rowsInserted
{
QVector<QPersistentModelIndexData *> persistent_moved = persistent.moved.pop();
```

Then somehow beginInsertRows call must be missed. Perhaps this is due to the model binding in MediaCoverFlow.qml

```
model: _internal.isCurrentPlaylistNotEmpty ? currentPlaylist : 1
```

Which would cause setSourceModel to be called on QIdentityProxyModel and reset signal connections, perhaps initial signal is missed due to this?

Either way since the crash is in Qt code, we either need to file a Qt bug (which will require a way to reproduce in a standalone project where we aren't doing anything obviously malicious, which seems doubtful) or we work around from media side.

If someone has better steps to reproduce please share and I will help find a workaround.