ALLSEEN ALLIANCE

Core Working Group

May 5, 2016

The contents of this document, and the interfaces described, and all the information herein, are the result of collaborative discussions by the Core Working Group.
This summary documents the final consensus of the team.

Reminder:

This call is being recorded

Antitrust Compliance Notice

- AllSeen Alliance meetings involve participation by industry competitors, and it is the intention of AllSeen Alliance to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of and not participate in any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.

- Examples of types of actions that are prohibited at AllSeen Alliance meetings and in connection with AllSeen Alliance activities are described in the AllSeen Alliance Antitrust Policy. If you have questions about these matters, please contact your company counsel, or if you are a member of AllSeen Alliance, feel free to contact Lee Gesmer or Andrew Updegrove, of the firm of Gesmer Updegrove LLP, which provides legal counsel to AllSeen Alliance.

Agenda

- : qcc::String vs. std::string in Core.
  - Status
    - MSFT discussed:
      - Currently have most of the advantages from the transition; may not be worth it being transitioned further
        - Closing out the remaining effort, it is quite costly, will take a long time – not something that MSFT will do
        - Work could fall into the general clean up of Core for usability
        - Still have the issue of ASACORE-2897 that needs solving.
  - Next steps
    - If we are going to pursue the full transition the recommendation is to deprecate the existing behavior, and replace methods depending on qcc:string with new ones that use std::string
      - Proposal if we do clean all this ups would be to create over-load functions that replace the qcc:string args with std:string
    - We should not start using std:string, but continue to use qcc:string> should convert all of it at once.
    - Next week discuss what we want to do with the Sec2.0 APIs which are using std::string (only some small outliers).
- Tickets from Base Services Project

Base Services Jiras

Tickets from Base Services team

- **Critical**
- 2918 - Blocks release of Android < API 21 (we are supporting 18+ according to our original plan for Jelly Bean 4.3+) - withdrawn
- **Important**
- 2912 - Important for Security 2.0 in Base Services
  - Current state is to leave this in 16.10, if Jarrod has stonger justification in the next wek or so we can revisit.

BaSe Jiras

- **Other annoying issues**
- *Including these depends on the work effort and risk but putting them on your radar regardless*
- 2906 and 2905 - means our code is stuck with deprecated functionality as the bindings are outdated
  - Both of these need to be addressed in 16.10 as they are issues with missing binding functionality
- 2801 - API consumer needs to download a legacy version of the NDK to build
  - Issue with latest NDK → we should look at moving forward, discuss next week with Ry on phone.
- 2919 - Implicit init and de-init via constructor/destructors causing problems with Java object life-times due to garbage collection
  - The request was to file a ticket for specific issue that the BaSe team is seening, and we can look at fixing the general problem in 16.10.

Action Items

Action Items

- Proposal to outline the process for changing APIs
  - Has been handed over to Brian Rockwell
    - will craft proposal for review by Core WG and then presented to TSC
  - Expectation that this will be reviewed with WG in January.
- Arvind (MSFT) to create Wiki process page linked off of Core WG wiki
  - Remove references to java from mandatory binding list for 16.04 timeframe
  - Hope to have something by the end of Jan.
- Ry (LF): Need to have Alliance running Windows 10 when RTM'd
  - Running the unit tests on the Windows 10 cannot seem to pass
    - Only one set of tests are failing, MSFT has worked with Ry.
    - Issue appears to be the combinations of Windows 10 and the version of VM ware
- MSFT: 16.04 work items, should have a sense by the end of Jan, or early Feb
  - Sec2.0 scenarios, working on C binding for Core Sec2.0

Action Items

- Add a way in JIRA to track compatibility issues and proposals
  - Proposal is the use a label: “compatibility impact” in the Jira ticket
  - Committers should require for these sorts of change that the release notes be updated
  - **Marcello** to document this process in the Core Process wiki
- Define process to require regression/unit tests for bugs
  - **Marcello** to add description of Committers requiring unit test for bug fixes to Core Wiki
- Define process for handling “Technical Debt”
  - Example: Took a shortcut to make a release and not loosing track of this to fix the shortcut
  - Jira items to track when things are missing – e.g. feature added, but missing from Java binding
  - If shortcut is taken: could create a Jira task to track the problem

Thank You

Follow Us On

- For more information on AllSeen Alliance, visit us at: allseenalliance.org & allseenalliance.org/news/blogs