



LXI Reference Design



It's about YOUR time.

www.lxistandard.org

Introduction

- Peter Plazotta, CEO of TSEP
- About TSEP:
 - 20 members staff (engineers, software developers, software developer trainees)
 - Development of complete software solutions for Measuring Devices
 - Development of testsystem for different companies, used world wide (Germany, Spain, China, Japan)
 - Systemsoftware development for customers (Drivers etc.)
 - LXI Member

Why a LXI Reference Design?

(From the LXI Consortium side)

- Lowers the boundary for vendors to use the standard
- Clarification of the standard
- Easier and wider distribution of the standard
- Shows vendors that the LXI Standard is ready to use

Why a LXI Reference Design?

(From the vendors side)

- Reduces the implementation costs for the standard dramatically
- Lower costs and efforts for vendors if the standard changes
- Easier certificating for vendors
- Compatible with other vendors (i.e LAN Messages)

Steps for LXI Reference Design and Implementation

1. Requirements

Steps for LXI Reference Design and Implementation

1. Requirements ⇨ done

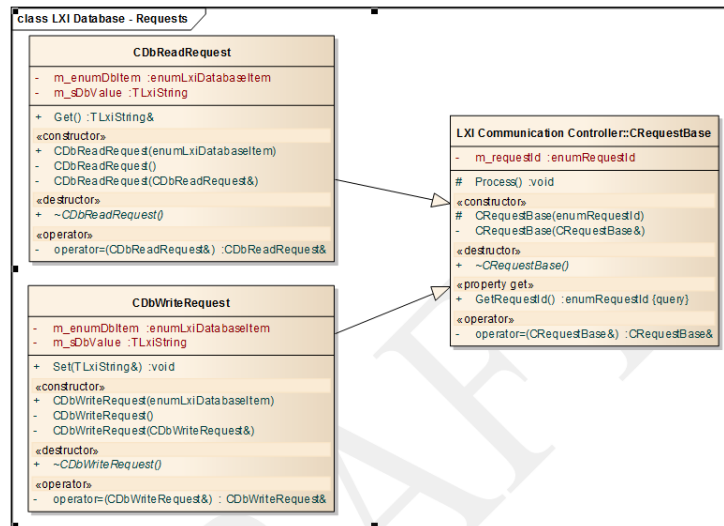
	A	B	C	D	E	F
1	Req-ID	LXI Standard Chapter	Type	In Reference Implementation	LXI Standard Chapter Name	Comments
2	REQ-STD-1-4-1	1.4.1	Rule	Yes	Application Version of Document	
3	REQ-STD-1-4-4-2	1.4.4.2	Rule	Yes	Conformance requirements	
4	REQ-STD-1-4-4-2-1	1.4.4.2.1	Rule	Yes	LXI Core 2011 Device Conformance Requirements	
5	REQ-STD-1-4-4-2-2	1.4.4.2.2	Permission	Yes	LXI Extended Functions	
6	REQ-STD-1-4-4-2-3	1.4.4.2.3	Rule	No	LXI Wired Trigger Bus Conformance Requirements	Currently not part of the reference implementation, because extended function bases on specific hardware
7	REQ-STD-1-4-4-2-4	1.4.4.2.4	Rule	Yes	LXI Event Messaging Conformance Requirements	
8	REQ-STD-1-4-4-2-5	1.4.4.2.5	Rule	No	LXI Clock Synchronization Conformance Requirements	Currently not part of the reference implementation, because extended function bases on specific hardware
9	REQ-STD-1-4-4-2-6	1.4.4.2.6	Rule	No	LXI Timestamped Data Conformance Requirements	Currently not part of the reference implementation, because extended function bases on specific hardware
10	REQ-STD-1-4-4-2-7	1.4.4.2.7	Rule	Yes	LXI Event Log Conformance Requirements	
11	REQ-STD-1-4-5	1.4.5	Rule	Yes	Functional Declaration	
12	REQ-STD-1-4-6	1.4.6	Rule	Yes	Web Indication of Functional Declaration	
13	REQ-STD-1-4-7	1.4.7	Rule	Yes	Term Using the LXI Trademark	
14						
15	REQ-STD-2-4-5	2.4.5	Rule	Yes	LAN Configuration Initialize (button)	Needs an instrument simulation for display
16	REQ-STD-2-5-2	2.5.2	Rule	Yes	LAN Status Indicator	Needs an instrument simulation for display
17	REQ-STD-2-5-3	2.5.3	Rule	No	IEEE 1588 Clock Status Indicator	
18						
19	REQ-STD-3-2-1	3.2.1	Rule	No	Implementation of IEEE 1588 Precision Time Protocol	Currently not part of the reference implementation, because extended function bases on specific hardware
20	REQ-STD-3-2-2	3.2.2	Recommendation	No	Precision of LXI Device Clocks	Currently not part of the reference implementation, because extended function bases on specific hardware
21	REQ-STD-3-2-3	3.2.3	Recommendation	No	Use of IEEE 1588 Boundary or Transparent Clocks	Currently not part of the reference implementation, because extended function bases on specific hardware
22	REQ-STD-3-2-4	3.2.4	Recommendation	No	Traceability to UTC	Currently not part of the reference implementation, because extended function bases on specific hardware
23	REQ-STD-3-2-5	3.2.5	Rule	No	Must be able to set UTC time	Currently not part of the reference implementation, because extended function bases on specific hardware
24	REQ-STD-3-2-6	3.2.6	Rule	No	Must be able to set UTC time manually	Currently not part of the reference implementation, because extended function bases on specific hardware
25	REQ-STD-3-2-7	3.2.7	Recommendation	No	Include at least one highly stable clock	Currently not part of the reference implementation, because extended function bases on specific hardware
	REQ-STD-3-2-8	3.2.8	Rule	No	Communication of Time must use IEEE 1588 Time	Currently not part of the reference implementation, because extended function bases on specific hardware

Steps for LXI Reference Design and Implementaion

1. Requirements ⇨ done
2. Reference Design

Steps for LXI Reference Design and Implementaion

1. Requirements ⇨ done
2. Reference Design ⇨ in progress

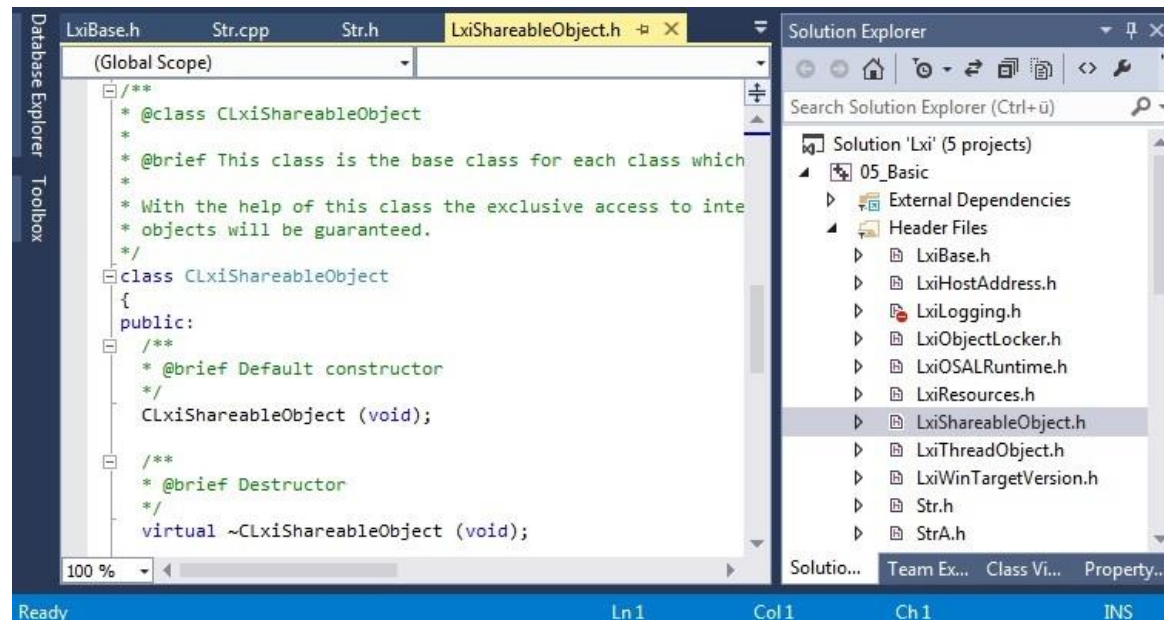


Steps for LXI Reference Design and Implementation

1. Requirements ⇨ done
2. Reference Design ⇨ in progress
3. Reference Implementation

Steps for LXI Reference Design and Implementation

1. Requirements ⇨ done
2. Reference Design ⇨ in progress
3. Reference Implementation ⇨ in progress



The screenshot displays a Visual Studio IDE window with the following components:

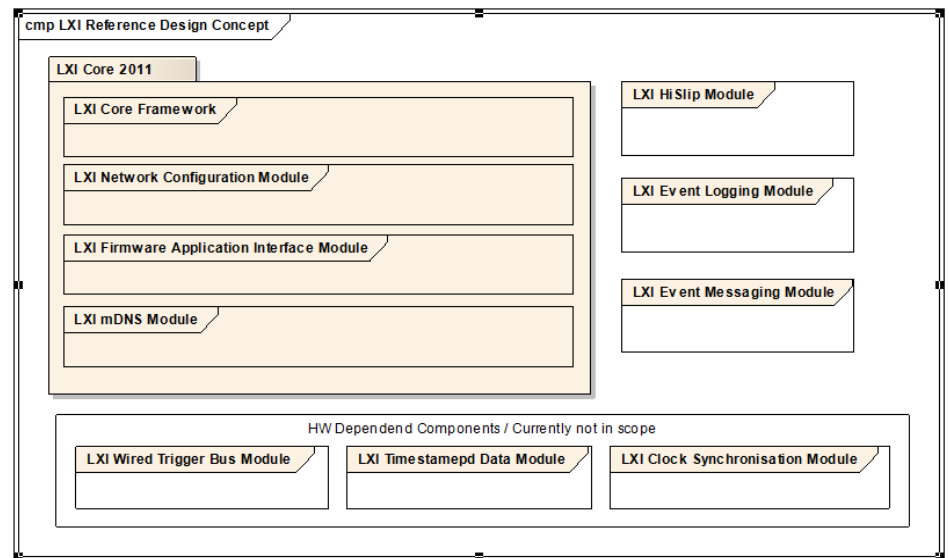
- Code Editor:** Shows the header file `LxiShareableObject.h` with the following code:

```
(Global Scope)
/**
 * @class CLxiShareableObject
 *
 * @brief This class is the base class for each class which
 *
 * With the help of this class the exclusive access to inte
 * objects will be guaranteed.
 */
class CLxiShareableObject
{
public:
    /**
     * @brief Default constructor
     */
    CLxiShareableObject (void);

    /**
     * @brief Destructor
     */
    virtual ~CLxiShareableObject (void);
};
```
- Solution Explorer:** Shows a project named "Solution 'Lxi' (5 projects)" with a folder "05_Basic" containing several header files, including `LxiShareableObject.h`.
- Status Bar:** Shows "Ready", "Ln 1", "Col 1", "Ch 1", and "INS".

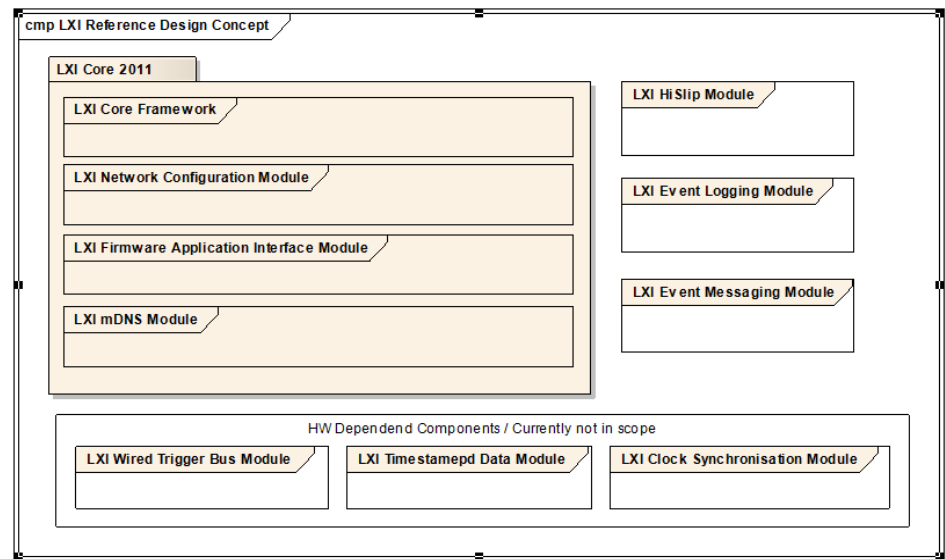
What will be included in the LXI Reference Design

- LXI Core 2011, including IPv6
- LXI Event Messaging
- LXI Event Logging
- LXI HiSlip



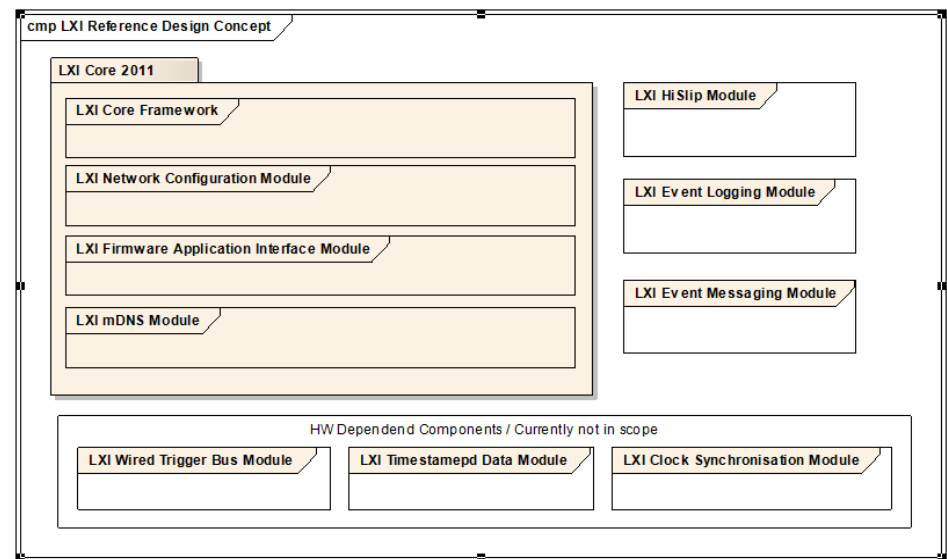
What will be excluded from the LXI Reference Design

- all modules depending on a vendor specific hardware
 - Clock Synchronisation
 - Wired Trigger Bus
 - Time Stamps
- reference hardware will not be provided



LXI Reference Design and IPv6

- IPv6 is included in the Reference Design
- IPv6 is not a standalone module
- IPv6 can be used optionally (compile time or runtime)



Requirements for the LXI Reference Design (LXI Standard)

- all rules and recommendations
⇒ qualified
- unnecessary requirements
⇒ skipped
- requirements which will be fulfilled just partly
⇒ clarified

Requirements for the LXI Reference Design (non-functional)

- development environments
 - OS
 - compiler
 - tools
- usage of third-party software
- documentation of the Reference Design and source code
- version control system and the bug tracking

Requirements for the LXI Reference Design (low-end devices)

- keep in mind the low-end devices
- „lightweight“ C++
 - no templates
 - no complex inheritance
 - no exceptions
- avoid huge objects on the stack
- decrease the amount of dynamically allocated memory

Requirements for the LXI Reference Design (low-end devices)

- do not use modern C++ libraries, such as STL or Qt
- define all data sizes by constants
- avoid new C++ standard features
- support an embedded web server

Development Tools for the LXI Reference Design

- Sparx Tool „Enterprise Architect“
i. e. an UML design and business analysis tool for object-oriented software systems
 - modeling
 - documenting
 - reverse engineering
 - building and maintaining
- customized output of the design results
- requirements included ⇨ traceability matrix

Development Tools for the LXI Reference Design



LXI Reference Implementation 2014

This draft is a work in progress and is NOT currently approved by the LXI Consortium.

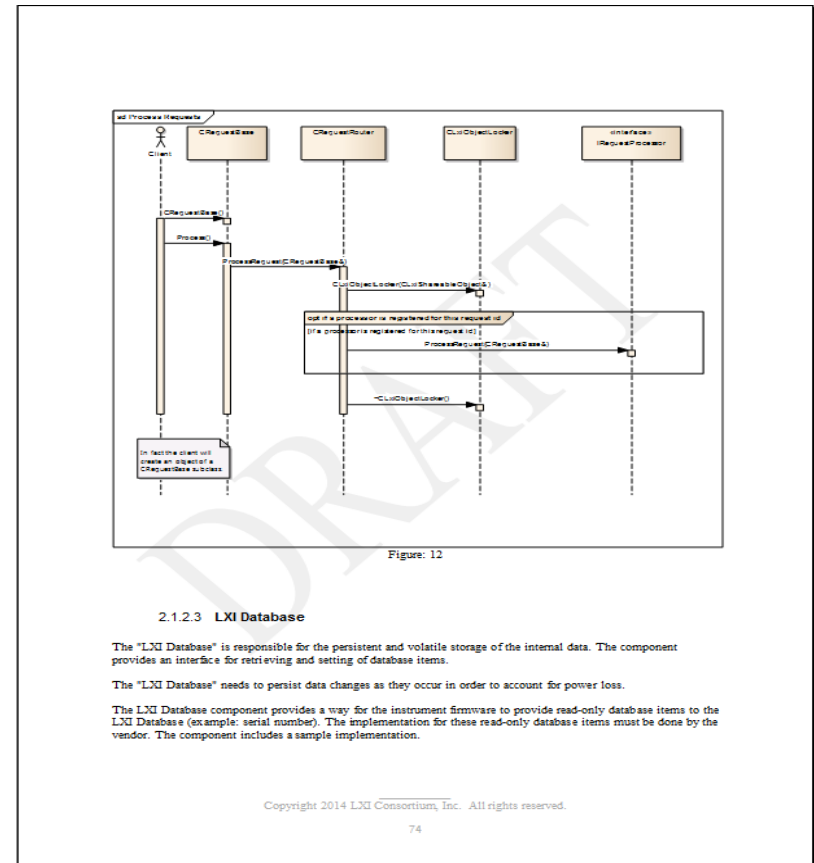
DRAFT DATE: May 15, 2014

Revision 0.13

Feb 15, 2014 Publication Date

Copyright 2014 LXI Consortium, Inc. All rights reserved.

1



Development Environment for the LXI Reference Implementation

- Platforms
 - Windows 7 (64 bit)
 - Ubuntu Linux (i386, 64 bit)
 - Ubuntu Linux Arm (32 bit)
- Windows ⇨ Visual Studio 2013
- Linux ⇨ Eclipse, GNU C++ Compiler
- solutions for compiling will be provided

Development Environment for the LXI Reference Implementation

- Linux Arm development
⇒ on CubieTruck board
- Windows and Ubuntu (i386) development
⇒ in an Oracle Virtual Box
- source code documented for doxygen

Current Status:

We are on schedule!

- all prototyping (web server etc.) ⇨ finished
- all basic classes ⇨ designed and coded
- first modules ⇨ designed
 - Core Framework, Network, Application Interface
- coding of the Core Framework ⇨ started
- mDNS and Event Logging modules ⇨ May
- HiSlip and Event Message modules ⇨ June

Thank you!