

Linked Network Actor - Zero Coupling Messages

Dr. Holger Brand, RBEE, GSI Helmholtzzentrum für Schwerionenforschung

Linked Network Actor

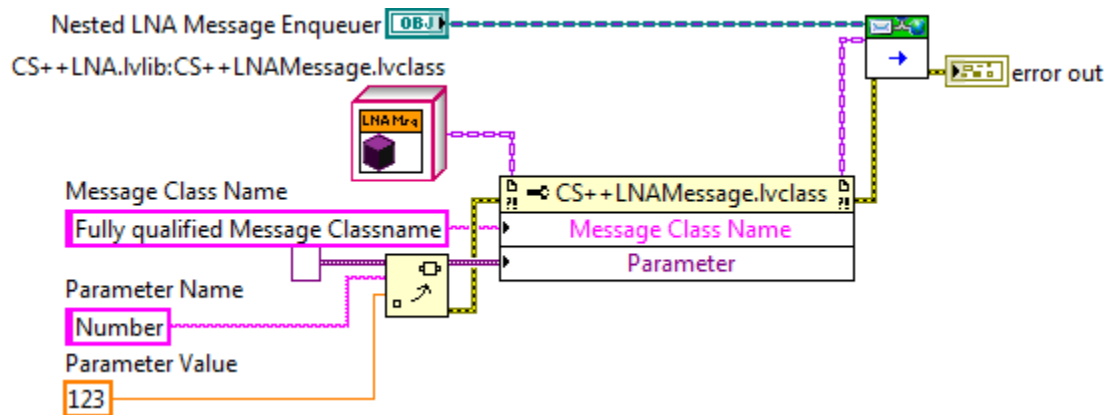
The *Linked Network Actor* library for NI's Actor Framework was published by *niACS* (nickname) at <https://decibel.ni.com/content/docs/DOC-24051>. It allows sending objects of *Message.lvclass* to remote LNA's using *Network Streams*. This is working fine in principle. You set the attribute of the message and send it to the remote LNA using *Linked Network Actor.lvclass:Transmit Network Message.vi*.

Problem

Typically a derived class of *Message.lvclass* calls in its Do.vi a public VI of an actor. So, the corresponding SubVI-Hierarchy is loaded into you project respectively the application. If the VI-Hierarchy contains a VI that is not supported on the current development platform or a necessary license is not available the VI-Hierarchy breaks and is not executable anymore.

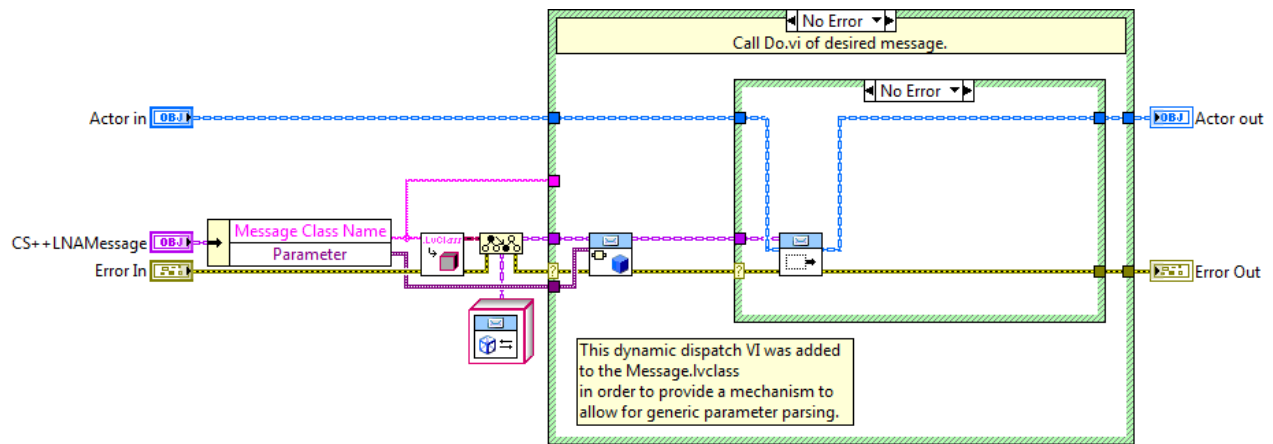
Solution Proposal

A possible solution is not to send a concrete message object, but a message containing the name of the desired message and necessary parameters for the corresponding public SubVI in its attributes. Such a class is provided *CS++LNA.lvlib:CS++LNAMessage.lvclass*.

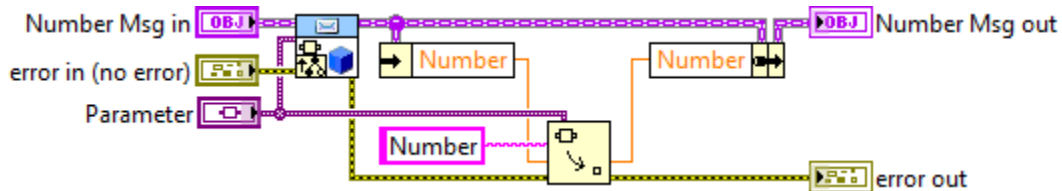


Sending a zero coupling Linked Network Actor message.

The method *CS++LNA.lvlib:CS++LNAMessage.lvclass.Do.vi* takes the message class name, and creates a *class default object* of the desired class type and casts it to the abstract *Message.lvclass*, which needs to be extended with an dynamic dispatch VI *Variant to Attribute.vi* to enable for generic parsing the variant attributes to message object's attributes. Derived classes will implement the concrete parsing.



Dispatching of a zero coupling Linked Network Actor message.



Example for parsing variant attribute to message object's attribute.

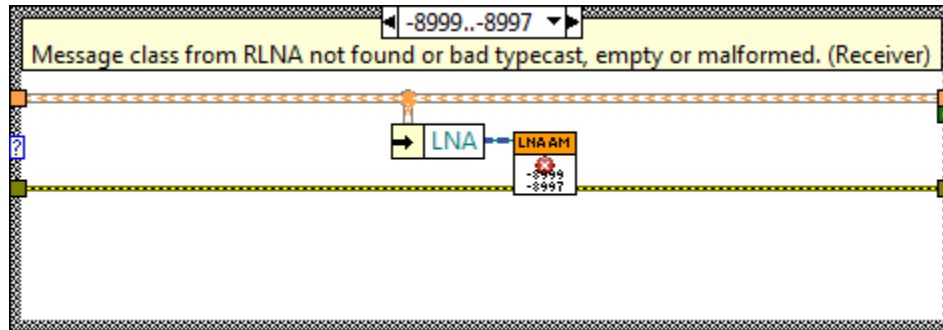
Error Handling

Possible known errors are converted to -8999 to -8996 and can be handled in LNA callers actor *Handle Error.vi*. Two SubVI are provided to handle these errors.

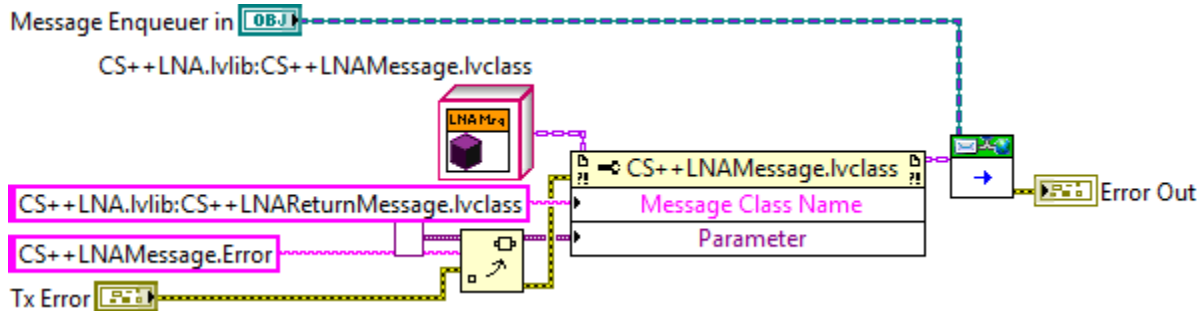
- -8999: Class received from Remote LNA not found.
- -8998: Bad typecast of messaged received from Remote LNA. Message.lvclass is expected.
- -8997: Message returned from Remote LNA. Message class name is empty string or malformed.
- -8996: Message returned from Remote LNA. Message class not found or bad typecast.

The following illustrates a possible error handling. As usual with *Linked Network Actor*, a derived actor class using a nested LNA needs to provide a derived class of *Linked Network Actor.lvlib:LNA Return Message Msg.lvclass* in order to handle returned messages. **No extra message classes are necessary compared to the ordinary LNA usage!**

- *CS++LNA:Handle Error -8999-8997.vi* can be used to return error information to the remote sender LNA using the zero coupling *CS++LNA.lvlib:CS++LNAMessage.lvclass* and *CS++LNA.lvlib:CS++LNAReturnMessage.lvclass* carries the error information.

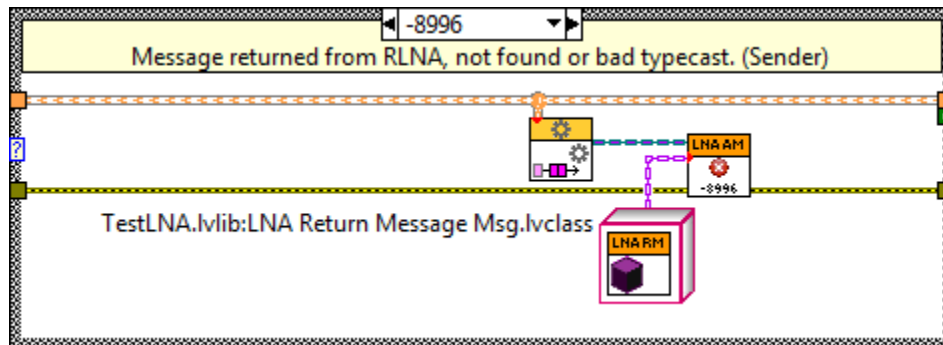


Excerpt of caller actors Handle Error.vi.

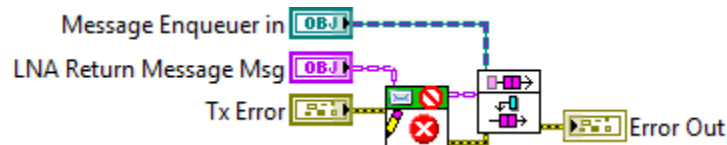


CS++LNA:Handle Error -8999-8997.vi

- *CS++LNA:Handle Error -8996.vi* can be used to return error information from the remote LNA to self by dispatching an caller actor class specific derived *LNA Return Message Msg.lvclass* using the self-enqueuer.



Excerpt of caller actors Handle Error.vi.



CS++LNA:Handle Error -8996.vi

The *CS++LNA.lvlib* is published under EUPL v1.1 on https://github.com/HB-GSI/CSPP_LNA.