Manual Test Script for Actor Framework Project Providers

The test script described herein tests most of the features of the Actor Framework Project Providers. Developers working on the Providers should step through this sequence to confirm existing functionality remains intact.

**Versions**

Unless otherwise noted, this test script should be used to verify baseline functionality for any open source release of the Actor Framework Project Providers. As of this writing, two versions are available for testing:

actor\_framework\_project\_providers\_2024\_for\_2020-1.0.0.4.vip

actor\_framework\_project\_providers\_2024\_for\_2020-1.1.0.10.vip

Additional test scripts will be required for any build that introduces new functionality.

**Dependencies**

To run this script, you must install any version of one of these packages:

1. “Interface for Actor INI” package from Zyah Solutions, available on VIPM.io

or

1. “Actor Framework 2024 for 2020”, available in the Open Source Actor Framework repository

**The Script**

1. Open and save a new project.
2. Test the **Add Actor** provider.
   1. Right click on **My Computer**, and select New » Actor.
   2. Name the actor “Bob” and click **OK**.
3. Test the **Message Maker** provider “Send Message” menu item.
   1. Create a new method called **Do Something.vi.**
   2. Add a numeric control to its front panel, and connect it to a connector pane terminal.
   3. Give the VI a meaningful icon, and save the VI
   4. Right click on the VI, then select Actor Framework » Create Message
   5. Confirm that Do Something Msg was created and added to the library, and that Send Do Something.vi takes a numeric value as input and has the correct icon.
4. Test the **Actor Message Maker** provider “Create Abstract Message for Caller” menu item.
   1. Right click on Bob.lvclass
   2. Select Actor Framework » Create Abstract Message for Caller.
   3. Select **Continue.**
   4. Name the message “Announce Data”.
   5. Add a numeric control to the “Add Attributes” dialog and select **OK.**
   6. Confirm that Announce Data Msg was created and added to Bob.lvlib, and that Send Announce Data.vi takes the numeric as input.
5. Test the **Message Maker** provider “Create Child of Abstract Message” menu item.
   1. Create a new actor, called “Alice”.
   2. Create a method called **Respond.vi** that takes a numeric as input. This numeric should have the same name and data type as the numeric created in Step 4e, above. Save the method.
   3. Right click on the method and select Actor Framework » Create Child of Abstract Message.
   4. In the file dialog, navigate to Bob » Bob Messages » Announce Data Message, and select Announce Data Msg.lvclass.
   5. Confirm that Response Msg.lvclass has been added to Alice.lvlib, and that it contains Do.vi.
6. Test the **Actor Message Maker** provider “Create Messages for This Actor” menu item.
   1. Create a new actor, called “Charlie”.
   2. Create a new method for this actor, called “Trigger.vi”.
   3. Right click on Charlie.lvclass and select Actor Framework » Create Messages for Actor.
   4. Confirm that Trigger Msg.lvclass has been added to Charlie.lvlib.
7. Test the **Actor Message Maker** provider “Create Messages for This Actors” menu item.
   1. Create two new actors, “Denise” and “Ellen”.
   2. For Denise, create a new method called “A Thing.vi”. Add a string control to its front panel, and connect the control to the VI’s connector pane.
   3. For Ellen, create a new method called “Toggle.vi”. Add a Boolean control to its front panel, and connect the control to the VI’s connector pane.
   4. In the project, select both actors. Then right click on either actor and select Actor Framework » Create Messages for Actors.
   5. Confirm that A Thing Msg.lvclass has been added to Denise.lvlib.
   6. Confirm that Toggle Msg.lvclass has been added to Ellen.lvlib.
8. Test the **Add Actor Interface** provider.
   1. Right Click on **My Computer** and select New » Interface for Actor.
   2. Name the new interface “Frodo”.
   3. Add a new method “Discard the Ring.vi” to the interface.
   4. Right click on the method, and confirm that Actor Framework » Create Message exists.
   5. Right click on Frodo.lvclass, and select Actor Framework » Create Messages for Actor.
   6. Confirm that the message “Discard the Ring Msg.lvclass” has been added to Frodo.lvlib.
9. Test the **Message Rescripter** provider.
   1. Right click on any message and select Actor Framework » Rescript Message.
   2. Confirm that the message was rescripted. (Feel free to first modify the target method in some way, but this is optional.)
   3. Repeat for a few other messages in this test.