How to use the Commentary Panel

Patrick H. Winston

Revision of December 23, 2014

The Commenary Panel

When developing a new capability, you will often want to look at commentary. Yes, you can print to the console, and should, using the `Mark.say` mechanism while debugging. Later, however, you will want to show off your miracles, so you should use the `Commentary` panel.

Suppose, for example, you are working on the `QuestionExpert` and you want to write to the `Commentary` panel. The first thing you do is connect a port, say COMMENTARY, on the `QuestionExpert` box to the `Commentary` panel. Thus, the following line will appear in the `QuestionExpert`:

```
public static final String COMMENTARY = "commentary";
```

Then, in your private copy of Genesis or, if your work has graduated out of your private package into the system, the following lines will be somewhere in `GenesisPlugBoarLower`:

```
Connections.wire(QuestionExpert.COMMENTARY,
                 getQuestionExpert(),
                 getCommentaryContainer());
```

Next, you put statements into your `QuestionExpert` box to transmit information to the `Commentary` panel. The information is packaged in a `BetterSignal` wrapper that includes the name of a tab and the message to be printed:

```
String tab = "Test";
String message = "Hello World";
BetterSignal signal = new BetterSignal(tab, message);
Connections.getPorts(this).transmit(QuestionExpert.COMMENTARY, signal);
```

You can use the buttons labeled with three vertical bars in the GUI windows to select the `Commentary` panel for display.

Optionally, you may want to cause the `Commentary` panel to appear programmatically. If so, you include an argument in the `BetterSignal` that says where the panel should appear. You have three choices: `GenesisConstants.LEFT`, `GenesisConstants.RIGHT`, and `GenesisConstants.BOTTOM`.

```
String tab = "Test";
String message = "Hello World";
BetterSignal signal = new BetterSignal(GenesisContants.RIGHT, tab, message);
Connections.getPorts(this).transmit(QuestionExpert.COMMENTARY, signal);
```

You can also control what panels are displayed using a user-interface idiom in your stories; see the tutorial, “How to Speak Genesese.”

The Html formatting class

The strings you write to the Commentary panel can contain HTML markup. To do the markup, you may find it convenient to use some syntactic sugaring methods in the `Html` class. For example, if you want part of your string to be bold, you could write:

```
String message = "Hello <b>wonderful</b> world";
```

Alternatively you can write:

```
String message = "Hello" + Html.bold("wonderful") + "world";
```

Using the `Html` methods is a big help when the messages get complex.