Connector Protector

Component Design Document

1 Description

This is a generic component that can be used to protect (as in protected object) the call to the input connector. The component serves as a multi-tasking safe synchronization point for multiple callers. When the T_Recv_Sync connector is called, the component immediately calls the T_Send connector, passing through any arguments. The T_Send connector is called from within a protected object, and so its execution is automic with respect to other callers of T_Recv_Sync. That is, the call to T_Send will finish before another task is allowed to invoke T_Recv_Sync (and thus T_Send). The protection mechanism effectively makes all downstream connector calls of this component thread-safe. The advantage of this component is that deploying it appropriately in an assembly can provide thread-safety to components which are not designed to be thread-safe in and of themselves.

2 Requirements

No requirements have been specified for this component.

3 Design

3.1 At a Glance

Below is a list of useful parameters and statistics that give a quick look into the makeup of the component.

- **Execution** - *passive*
- **Number of Connectors** - 2
- **Number of Invokee Connectors** - 1
- **Number of Invoker Connectors** - 1
- **Number of Generic Connectors** - *None*
- **Number of Generic Types** - 1
- **Number of Unconstrained Arrayed Connectors** - *None*
- **Number of Commands** - *None*
- **Number of Parameters** - *None*
- **Number of Events** - *None*
- **Number of Faults** - *None*
- **Number of Data Products** - *None*
- **Number of Data Dependencies** - *None*
- **Number of Packets** - *None*

3.2 Diagram

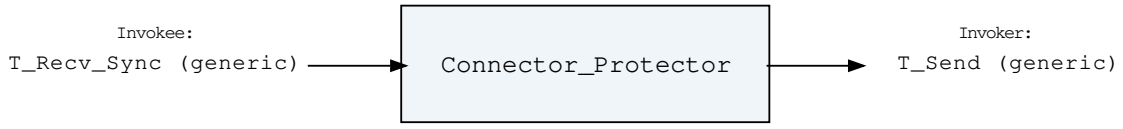

Figure 1: Connector Protector component diagram.

3.3 Connectors

Below are tables listing the component's connectors.

3.3.1 Invokee Connectors

The following is a list of the component's *invokee* connectors:

Table 1: Connector Protector Invokee Connectors

Name	Kind	Type	Return_Type	Count
T_Recv_Sync	recv_sync	T (generic)	-	1

Connector Descriptions:

- **T_Recv_Sync** - The generic invokee connector.

3.3.2 Invoker Connectors

The following is a list of the component's *invoker* connectors:

Table 2: Connector Protector Invoker Connectors

Name	Kind	Type	Return_Type	Count
T_Send	send	T (generic)	-	1

Connector Descriptions:

- **T_Send** - The generic invoker connector. Calls originating from this connector are contained within a protected object and thus downstream operations are atomic and thread-safe.

3.4 Interrupts

This component contains no interrupts.

3.5 Initialization

Below are details on how the component should be initialized in an assembly.

3.5.1 Generic Component Instantiation

The is generic component in that it can be instantiated to protect a connector of any type at compile time. This component contains generic formal types. These generic formal types must be instantiated with a valid actual type prior to component initialization. This is done by specifying types for the following generic formal parameters:

Table 3: Connector Protector Generic Formal Types

Name	Formal Type Definition
T	`type T is private;`

Generic Formal Type Descriptions:

- **`T`** - The generic type of data passed in and out of the component.

3.5.2 Component Instantiation

This component contains no instantiation parameters in its discriminant.

3.5.3 Component Base Initialization

This component contains no base class initialization, meaning there is no `init_Base` subprogram for this component.

3.5.4 Component Set ID Bases

This component contains no commands, events, packets, faults or data products that need base indentifiers.

3.5.5 Component Map Data Dependencies

This component contains no data dependencies.

3.5.6 Component Implementation Initialization

This component contains no implementation class initialization, meaning there is no `init` subprogram for this component.

3.6 Commands

The Connector Protector component has no commands.

3.7 Parameters

The Connector Protector component has no parameters.

3.8 Events

The Connector Protector component has no events.

3.9 Data Products

The Connector Protector component has no data products.

3.10 Packets

The Connector Protector component has no packets.

4 Unit Tests

The following section describes the unit test suites written to test the component.

4.1 *Connector_ Protector_ Tests* Test Suite

This is a unit test suite for the Connector Protector.

Test Descriptions:

- **`Test_Protected_Call`** - This unit test invokes the protected connector and makes sure the arguments are passed through synchronously, as expected.

5 Appendix

5.1 Preamble

This component contains no preamble code.

5.2 Packed Types

The following section outlines any complex data types used in the component in alphabetical order. This includes packed records and packed arrays that might be used as connector types, command arguments, event parameters, etc..

No complex types found in component.

5.3 Enumerations

No enumerations found in component.