DOCUMENTATION

```
1. --

2. --   Uwe R. Zimmer, Australia, September 2011

3. --

4.

5. package body Generic_Router_LinksTo API docTo spec is

6.

7.    function To_Router_PortsTo API docTo spec (Task_IdTo API docTo spec :
   Router_RangeTo API docTo spec; LinksTo API docTo spec : Ids_To_LinksTo API
   docTo spec) return Connected_Router_PortsTo API docTo spec is

8.

9.       No_of_Links : Natural := 0;

10.

11.   begin

12.      for Router_Id in Router_RangeTo                        p

13.         if Task_IdTo API docTo spec                    en LinksTo API docTo
   spec (Router_Id) /= Null_ReferenceTo API docTo spec then

14.            No_of_Links := No_of_Links + 1;

15.         end if;

16.      end loop;

17.      declare

18.         Channels   : Connected_Router_PortsTo API docTo spec (1 ..
   No_of_Links);

19.         Channel_Ix : Positive range Channels&apos;Range :=
   Channels&apos;First;

20.      begin

21.         for Router_Id in Router_RangeTo API docTo spec loop

22.            if Task_IdTo API docTo spec /= Router_Id and then LinksTo API
   docTo spec (Router_Id) /= Null_ReferenceTo API docTo spec then

23.               Channels (Channel_Ix) := (Id => Router_Id, Link => LinksTo API
   docTo spec (Router_Id));
```

```
24.              if Channel_Ix < Channels'Last then

25.                 Channel_Ix := Channel_Ix + 1;

26.              end if;

27.          end if;

28.       end loop;

29.       return Channels;

30.    end;

31.  end To_Router_PortsTo API docTo specTo body;

32.

33.end Generic_Router_LinksTo API docTo specTo body;
```