DOCUMENTATION

```
1. --

2. --  Uwe R. Zimmer, Australia, September 2019

3. --

4.

5. with Ada.Text_IO; use Ada.Text_IO;

6. with TopologiesTo API docTo specTo body;  use TopologiesTo API docTo specTo
   body;

7.

8. package body Generic_RoutersTo API docTo spec is

9.

10.   procedur _____                    PI_doc         To spec :
   Router_Range

11.

12.   begin

13.      select

14.         Router_TasksTo API docTo spec (IdTo API docTo spec).ShutdownTo API
   docTo specTo body;

15.      or

16.         delay Command_Line_Parameters.Comms_Timeout;

17.         Put_Line ("Warning: Router" & Router_RangeTo API docTo
   spec&apos;Image (IdTo API docTo spec) & " did not respond to Shutdown call ->
   trying to abort it");

18.         abort Router_TasksTo API docTo spec (IdTo API docTo spec);

19.      end select;

20.      Router_ActiveTo API docTo spec (IdTo API docTo spec) := False;

21.   end Router_ShutdownTo API docTo specTo body;

22.

23.   procedure Global_ShutdownTo API docTo spec is

24.
```

```
25.    begin

26.        for Id in Router_Range loop

27.            if Router_Active (Id) then

28.                Router_Shutdown (Id);

29.            end if;

30.        end loop;

31.    end Global_Shutdown;

32.

33.    use Router_Link;

34.

35.begin

36.    for Router_Id in Router_Range loop

37.        declare

38.            Links : IDs_To_Links := (others => null);

39.        begin

40.            fo                                          loop

41.                if Router_Id /= Potential                    hen Nodes_Connected
   (Connection_Topology                              itive (Router_Id),
   Positive (Potential_Neighbour_Id)) the

42.                    Links (Potential_Neighbour_Id) := Router_Tasks
   (Potential_Neighbour_Id)'Access;

43.                end if;

44.            end loop;

45.            select

46.                Router_Tasks (Router_Id).Configure (Links);

47.            or

48.                delay Command_Line_Parameters.Comms_Timeout;

49.                Routers_Configured := False; exit;

50.            end select;

51.        end;

52.    end loop;

53.end Generic_Routers;
```