Ada Issue 00394 Redundant Restriction Identifiers and completing Ravenscar definition

!standard D.07(09) 05-03-01 AI95-00394/03
!standard D.07(10)
!standard D.07(15)
!standard D.13.1(01)
!standard H.4(2)
!standard H.4(9)
!standard H.4(16)
!standard J.13(1)
!class amendment 05-01-20
!status Amendment 200Y 05-02-25
!status ARG Approved 9-0-1 05-02-13
!status work item 05-01-20
!status received 05-01-20
!priority High
!difficulty Easy

!subject Redundant Restriction Identifiers and completing Ravenscar definition

!summary

This AI moves existing restriction identifiers to Annex J, and modifies the definition of the Ravenscar profile. It also completes the definition of the Ravenscar profile by including new restrictions that concern new features of the language.

!problem

The new restriction identifier, No_Dependence, has made existing identifiers redundant and allows the Ravenscar profile to be defined without the need to introduce further new identifiers. It also makes AI-353 redundant, and a number of new restriction identifiers in AI-305.

To complete the definition of Ravenscar it is necessary to restrict the use of the following new features:

1. CPU Timers
2. Group Timers
3. Specific handlers for task termination.

The first two are restricted due to their incompatibility with Ravenscar; the events they define identify dynamic actions that Ravenscar cannot easily utilize. The required restrictions are obtained by the use of restriction identifier No_Dependence.

As tasks in Ravenscar are not intended to terminate (and are all at the library level) it is deemed adequate for there to be only a fall-back handler. The run-time will be simpler if specific handlers are not supported. To define this restriction a new restrictions identifier is defined: No_Specific_Termination_Handlers.

Finally, to make it clear that if a Ravenscar task does terminate the fall-back handler (if there is one) will be executed; the definition of No_Task_Termination is modified.

!proposal

(See wording.)

!wording

Delete (the original) D.7(10), H.4(9), and H.4(16).

Delete No_Calendar and No_Task_Attributes_Package from the wording changes of AI-305.

AI-353 should be dropped from the Amendment.

Add new Section in Annex J:

J.13 Dependence Restrictions Identifiers

The following restrictions involve dependence on specific language-defined units. The more general No_Dependence restriction should be used for this purpose.

Static Semantics

The following restrictions_identifiers exist:

No_Asynchronous_Control
  There are no semantic dependences on package Asynchronous_Task_Control.

No_Unchecked_Conversion
  Semantic dependence on the predefined generic Unchecked_Conversion is not allowed.

No_Unchecked_Deallocation
  Semantic dependence on Unchecked_Deallocation is not allowed.

Add the following to the end of H.4(2):

The following uses of restriction identifier No_Dependence apply in this Annex: No_Dependence => Ada.Unchecked_Deallocation and No_Dependence => Ada.Unchecked_Conversion.

Add the following new static restriction_identifier after D.7(10):

No_Specific_Termination_Handlers
  There are no calls to the following subprograms in Task_Termination: Set_Specific_Handler and Specific_Handler.

Modify the definition of restriction_identifier in AI-305 to the following:

No_Task_Termination

All tasks are non-terminating. It is implementation-defined what happens if a task attempts to terminate. If there is a fall-back handler (see C.7.3) set for the partition it should be called when the first task attempts to terminate.

The static semantic section of the Ravenscar profile definition (in AI-249) becomes:

Static Semantics
The run-time profile Ravenscar is equivalent to the following set of pragmas:

```
pragma Task_Dispatching_Policy (FIFO_Within_Priorities);

pragma Locking_Policy (Ceiling_Locking);

pragma Detect_Blocking;

pragma Restrictions (
        No_Abort_Statements,
        No_Dynamic_Attachment,
        No_Dynamic_Priorities,
        No_Implicit_Heap_Allocations,
        No_Local_Protected_Objects,
        No_Local_Timing_Events,
        No_Protected_Type_Allocators,
        No_Relative_Delay,
        No_Requeue_Statements,
        No_Select_Statements,
        No_Specific_Termination_Handlers,
        No_Task_Allocators,
        No_Task_Hierarchy,
        No_Task_Termination,
        Simple_Barriers,
        Max_Entry_Queue_Length => 1,
        Max_Protected_Entries => 1,
        Max_Task_Entries => 0,
        No_Dependence => Ada.Asynchronous_Task_Control,
        No_Dependence => Ada.Calendar,
        No_Dependence => Ada.Execution_Time.Group_Budget,
        No_Dependence => Ada.Execution_Time.Timers,
        No_Dependence => Ada.Task_Attributes);
```

!discussion

The identifier No_IO covers a set of library packages and thus was not moved to Annex J.

The identifier No_Dynamic_Priorities was extended by AI-327 to cover uses of the Priority attribute for protected objects as well as dependencies on Ada.Dynamic_Priorities. Thus it was not moved to Annex J.

The minimum requirement for task termination when No_Task_Termination is in force is to ensure that any fall-back handler is executed at least once. Given that no tasks are meant to terminate this would seem to be sufficient.

!corrigendum D.7(10)
!comment The AI-305 changes are made in the conflict text only.
@drepl
@xhang<@xterm<No_Asynchronous_Control>
There are no semantic dependences on the package Asynchronous_Task_Control.>
@dby
@xhang<@xterm<No_Specific_Termination_Handlers>
There are no calls to the Set_Specific_Handler and Specific_Handler subprograms in Task_Termination.

!corrigendum D.7(15)

@drepl
@i<This paragraph was deleted>
@dby
The following @I<restriction_>@fa<identifier>s are language defined:

@xhang<@xterm<No_Task_Termination>
All tasks are non-terminating. It is implementation-defined what happens if a task attempts to terminate. If there is a fall-back handler (see C.7.3) set for the partition it should be called when the first task attempts to terminate.>

!corrigendum D.13.1(01)

@dinsc

This clause defines the Ravenscar profile.

@i<@s8<Legality Rules>>

The @i<profile_>@fa<identifier> Ravenscar names a run-time profile.
For run-time profile Ravenscar, there shall be no
@i<profile_>@fa<pragma_argument_association>s.

@i<@s8<Static Semantics>>

The run-time profile Ravenscar
is equivalent to the following set of pragmas:

@xcode<@b<pragma> Task_Dispatching_Policy (FIFO_Within_Priorities);
@b<pragma> Locking_Policy (Ceiling_Locking);
@b<pragma> Detect_Blocking;
@b<pragma> Restrictions (
No_Abort_Statements,
No_Dynamic_Attachment,
No_Dynamic_Priorities,
No_Implicit_Heap_Allocations,

No_Local_Protected_Objects,
No_Local_Timing_Events,
No_Protected_Type_Allocators,
No_Relative_Delay,
No_Requeue_Statements,
No_Select_Statements,
No_Specific_Termination_Handlers,
No_Task_Allocators,
No_Task_Hierarchy,
No_Task_Termination,
Simple_Barriers,
Max_Entry_Queue_Length =@> 1,
Max_Protected_Entries =@> 1,
Max_Task_Entries =@> 0,
No_Dependence =@> Ada.Asynchronous_Task_Control,
No_Dependence =@> Ada.Calendar,
No_Dependence =@> Ada.Execution_Time.Group_Budget,
No_Dependence =@> Ada.Execution_Time.Timers,
No_Dependence =@> Ada.Task_Attributes);>

@xindent<@s9<NOTES@hr
37 The effect of the Max_Entry_Queue_Length =@> 1 restriction applies only to protected entry queues due to the accompanying restriction of Max_Task_Entries =@> 0.>>

!corrigendum H.4(2)

@drepl
The following restrictions, the same as in D.7, apply in this Annex:
No_Task_Hierarchy, No_Abort_Statement, No_Implicit_Heap_Allocation,
Max_Task_Entries is 0, Max_Asynchronous_Select_Nesting is 0, and Max_Tasks is
0. The last three restrictions are checked prior to program execution.

@dby
The following restrictions, the same as in D.7, apply in this Annex:
No_Task_Hierarchy, No_Abort_Statement, No_Implicit_Heap_Allocation,
Max_Task_Entries is 0, Max_Asynchronous_Select_Nesting is 0, and Max_Tasks is
0. The last three restrictions are checked prior to program execution.
The following uses of restriction identifier No_Dependence apply in this
Annex: No_Dependence =@> Ada.Unchecked_Deallocation and No_Dependence =@>
Ada.Unchecked_Conversion.

!corrigendum H.4(9)

@ddel
@xhang<@xterm<No_Unchecked_Deallocation>
Semantic dependence on Unchecked_Deallocation is not allowed.>

!corrigendum H.4(16)

@ddel
@xhang<@xterm<No_Unchecked_Conversion>

Semantic dependence on the predefined generic Unchecked_Conversion is not allowed.>

!corrigendum J.13(1)

@dinsc

The following @fa<restrictions> involve dependence on specific language-defined units. The more general restriction No_Dependence (see 13.12.1) should be used for this purpose.

@i<@s8<Static Semantics>>

The following @i<restrictions_>@fa<identifiers> exist:

@xhang<@xterm<No_Asynchronous_Control>
There are no semantic dependences on package Asynchronous_Task_Control.>

@xhang<@xterm<No_Unchecked_Conversion>
Semantic dependence on the predefined generic Unchecked_Conversion is not allowed.>

@xhang<@xterm<No_Unchecked_Deallocation>
Semantic dependence on Unchecked_Deallocation is not allowed.>

!ACATS test

This AI should be covered by tests for the Ravenscar profile.