

# nestedSequencer3 Report

4th October 2016

## 1 ID Files

### 1.1 MissionIds

**section** *MissionIds* **parents** *scj\_prelude*, *MissionId*

*MainMissionMID : MissionID*

*NestedMissionAMID : MissionID*

*NestedMissionBMID : MissionID*

---

*distinct*(*nullMissionId*, *MainMissionMID*, *NestedMissionAMID*,  
*NestedMissionBMID*)

## 1.2 SchedulablesIds

**section** *SchedulableIds* **parents** *scj\_prelude, SchedulableId*

*mainSequencerSID : SchedulableID*

*NestedMissionSequencerSID : SchedulableID*

*MT1SID : SchedulableID*

*MT2SID : SchedulableID*

*distinct*(*nullSequencerId, nullSchedulableId, mainSequencerSID,*  
*NestedMissionSequencerSID, MT1SID,*  
*MT2SID*)

## 2 Network

### 2.1 Network Channel Sets

```
section NetworkChannels parents scj_prelude, MissionId, MissionIds,  
    SchedulableId, SchedulableIds, MissionChan, TopLevelMissionSequencerFWChan,  
    FrameworkChan, SafeletChan, AperiodicEventHandlerChan, ManagedThreadChan,  
    OneShotEventHandlerChan, PeriodicEventHandlerChan, MissionSequencerMethChan  
  
channelset TerminateSync ==  
    { schedulables_terminated, schedulables_stopped, get_activeSchedulables }  
  
channelset ControlTierSync ==  
    { start_toplevel_sequencer, done_toplevel_sequencer, done_safeletFW }  
  
channelset TierSync ==  
    { start_mission . MainMission, done_mission . MainMission,  
      done_safeletFW, done_toplevel_sequencer }  
  
channelset MissionSync ==  
    { done_safeletFW, done_toplevel_sequencer, register,  
      signalTerminationCall, signalTerminationRet, activate_schedulables, done_schedulable,  
      cleanupSchedulableCall, cleanupSchedulableRet }  
  
channelset SchedulablesSync ==  
    { activate_schedulables, done_safeletFW, done_toplevel_sequencer }  
  
channelset ClusterSync ==  
    { done_toplevel_sequencer, done_safeletFW }  
  
channelset SafeltAppSync  $\hat{=}$   
    { getSequencerCall, getSequencerRet, initializeApplicationCall, initializeApplicationRet, end_safelet_app }  
  
channelset MissionSequencerAppSync ==  
    { getNextMissionCall, getNextMissionRet, end_sequencer_app }  
  
channelset MissionAppSync ==  
    { initializeCall, register, initializeRet, cleanupMissionCall, cleanupMissionRet }  
  
channelset AppSync ==  
    { SafeltAppSync, MissionSequencerAppSync, MissionAppSync,  
      MTEAppSync, OSEHSync, APEHSync, PEHSync,  
      { getSequencer, end_mission_app, end_managedThread_app,  
        setCeilingPriority, requestTerminationCall, requestTerminationRet, terminationPendingCall,  
        terminationPendingRet, handleAsyncEventCall, handleAsyncEventRet } }  
  
channelset ThreadSync ==  
    { raise_thread_priority, lower_thread_priority, isInterruptedCall, isInterruptedRet, get_priorityLevel }  
  
channelset LockingSync ==  
    { lockAcquired, startSyncMeth, endSyncMeth, waitCall, waitRet, notify, isInterruptedCall, isInterruptedRet,  
      interruptedCall, interruptedRet, done_toplevel_sequencer, get_priorityLevel }  
  
channelset Tier0Sync ==  
    { done_toplevel_sequencer, done_safeletFW,  
      start_mission . NestedMissionA, done_mission . NestedMissionA,  
      initializeRet . NestedMissionA, requestTermination . NestedMissionA . mainSequencer,  
      start_mission . NestedMissionB, done_mission . NestedMissionB,  
      initializeRet . NestedMissionB, requestTermination . NestedMissionB . mainSequencer }
```

## 2.2 Locking

**section** *NetworkLocking* **parents** *scj\_prelude, GlobalTypes, FrameworkChan, MissionId, MissionIds, ThreadIds, NetworkChannels, ObjectFW, ThreadFW*

**process** *Threads*  $\hat{=}$   
(**Skip**)

**process** *Objects*  $\hat{=}$   
(**Skip**)

**process** *Locking*  $\hat{=}$  *Threads* [ *ThreadSync* ] *Objects*

## 2.3 Program

**section** *Program* **parents** *scj\_prelude, MissionId, MissionIds, SchedulableId, SchedulableIds, MissionChan, SchedulableMethChan, MissionFW, SafeletFW, TopLevelMissionSequencerFW, NetworkChannels, ManagedThreadFW, SchedulableMissionSequencerFW, PeriodicEventHandlerFW, OneShotEventHandlerFW, AperiodicEventHandlerFW, ObjectFW, ThreadFW, MyAppApp, mainSequencerApp, MainMissionApp, NestedMissionSequencerApp, NestedMissionAApp, MT1App, NestedMissionBApp, MT2App*

**process** *ControlTier*  $\hat{=}$   

$$\left( \begin{array}{l} \text{SafeletFW} \\ \llbracket \text{ControlTierSync} \rrbracket \\ \text{TopLevelMissionSequencerFW}(\text{mainSequencer}) \end{array} \right)$$

**process** *Tier0*  $\hat{=}$   

$$\left( \begin{array}{l} \text{MissionFW}(\text{MainMissionID}) \\ \llbracket \text{MissionSync} \rrbracket \\ (\text{OneShotEventHandlerFW}(\text{NestedMissionSequencerID})) \end{array} \right)$$

**process** *Tier1*  $\hat{=}$   

$$\left( \begin{array}{l} \text{MissionFW}(\text{NestedMissionAID}) \\ \llbracket \text{MissionSync} \rrbracket \\ (\text{ManagedThreadFW}(\text{MT1ID})) \\ \llbracket \text{ClusterSync} \rrbracket \\ \text{MissionFW}(\text{NestedMissionBID}) \\ \llbracket \text{MissionSync} \rrbracket \\ (\text{ManagedThreadFW}(\text{MT2ID})) \end{array} \right)$$

**process** *Framework*  $\hat{=}$   

$$\left( \begin{array}{l} \text{ControlTier} \\ \llbracket \text{TierSync} \rrbracket \\ \left( \begin{array}{l} \text{Tier0} \\ \llbracket \text{Tier0Sync} \rrbracket \end{array} \right) \\ \text{Tier1} \end{array} \right)$$

**process** *Application*  $\hat{=}$   

$$\left( \begin{array}{l} \text{MyAppApp} \\ ||| \\ \text{mainSequencerApp} \\ ||| \\ \text{MainMissionApp} \\ ||| \\ \text{NestedMissionSequencerApp} \\ ||| \\ \text{NestedMissionAApp} \\ ||| \\ \text{MT1App} \\ ||| \\ \text{NestedMissionBApp} \\ ||| \\ \text{MT2App} \end{array} \right)$$

**process** *Program*  $\hat{=}$   $(\text{Framework} \llbracket \text{AppSync} \rrbracket \text{Application}) \llbracket \text{LockingSync} \rrbracket \text{Locking}$

### 3 Safelet

**section** *MyAppApp* **parents** *scj\_prelude, SchedulableId, SchedulableIds, SafeletChan, MethodCallBindingChannels*

**process** *MyAppApp*  $\hat{=}$  **begin**

*InitializeApplication*  $\hat{=}$   
 $\left( \begin{array}{l} \textit{initializeApplicationCall} \longrightarrow \\ \textit{initializeApplicationRet} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

*GetSequencer*  $\hat{=}$   
 $\left( \begin{array}{l} \textit{getSequencerCall} \longrightarrow \\ \textit{getSequencerRet} \text{! } \textit{mainSequencerSID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

*Methods*  $\hat{=}$   
 $\left( \begin{array}{l} \textit{GetSequencer} \\ \square \\ \textit{InitializeApplication} \end{array} \right); \textit{Methods}$

•  $(\textit{Methods}) \triangle (\textit{end\_safelet\_app} \longrightarrow \mathbf{Skip})$

**end**

## 4 Top Level Mission Sequencer

**section** *mainSequencerApp* **parents** *TopLevelMissionSequencerChan*,  
*MissionId*, *MissionIds*, *SchedulableId*, *SchedulableIds*, *mainSequencerClass*, *MethodCallBindingChannels*

**process** *mainSequencerApp*  $\hat{=}$   
*name* : *String* • **begin**

<i>State</i> <i>this</i> : <b>ref</b> <i>mainSequencerClass</i>
--

**state** *State*

<i>Init</i> <i>State'</i>
<i>this'</i> = <b>new</b> <i>mainSequencerClass</i> ()

*GetNextMission*  $\hat{=}$  **var** *ret* : *MissionID* •  
 $\left( \begin{array}{l} \textit{getNextMissionCall} . \textit{mainSequencerSID} \longrightarrow \\ \textit{ret} := \textit{this} . \textit{getNextMission}(); \\ \textit{getNextMissionRet} . \textit{mainSequencerSID} ! \textit{ret} \longrightarrow \\ \textbf{Skip} \end{array} \right)$

*Methods*  $\hat{=}$   
 $(\textit{GetNextMission}) ; \textit{Methods}$

•  $(\textit{Init} ; \textit{Methods}) \triangle (\textit{end\_sequencer\_app} . \textit{mainSequencerSID} \longrightarrow \textbf{Skip})$

**end**

**section** *mainSequencerClass* **parents** *scj\_prelude*, *SchedulableId*, *SchedulableIds*, *SafeletChan*  
*, MethodCallBindingChannels*, *MissionId*, *MissionIds*

**class** *mainSequencerClass*  $\hat{=}$  **begin**

<b>state</b> <i>State</i> <i>notReleased</i> : $\mathbb{B}$
--

**state** *State*

<b>initial</b> <i>Init</i> <i>State</i> '
<i>notReleased</i> = <b>True</b>

**protected** *getNextMission*  $\hat{=}$  **var** *ret* : *MissionID* •

$$\left( \begin{array}{l} \text{if } notReleased = \mathbf{True} \longrightarrow \\ \quad \left( notReleased := \mathbf{False}; \right. \\ \quad \left. ret := MainMissionMID \right) \\ \quad \neg notReleased = \mathbf{True} \longrightarrow \\ \quad \quad (ret := nullMissionId) \\ \text{fi} \end{array} \right)$$

• **Skip**

**end**



## 5 Missions

### 5.1 MainMission

**section** *MainMissionApp* **parents** *scj\_prelude*, *MissionId*, *MissionIds*,  
*SchedulableId*, *SchedulableIds*, *MissionChan*, *SchedulableMethChan*, *MainMissionMethChan*,  
*MethodCallBindingChannels*

**process** *MainMissionApp*  $\hat{=}$  **begin**

<i>State</i> <i>this</i> : <b>ref</b> <i>MainMissionClass</i>
--

**state** *State*

<i>Init</i> <i>State'</i>
<i>this'</i> = <b>new</b> <i>MainMissionClass</i> ()

*InitializePhase*  $\hat{=}$   
 $\left( \begin{array}{l} \textit{initializeCall} . \textit{MainMissionMID} \longrightarrow \\ \textit{register} ! \textit{NestedMissionSequencerSID} ! \textit{MainMissionMID} \longrightarrow \\ \textit{initializeRet} . \textit{MainMissionMID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

*CleanupPhase*  $\hat{=}$   
 $\left( \begin{array}{l} \mathbf{var} \ \mathbb{B} : \textit{ret} \bullet \textit{cleanupMissionCall} . \textit{MainMissionMID} \longrightarrow \\ \textit{cleanupMissionRet} . \textit{MainMissionMID} ! \mathbf{True} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

*Methods*  $\hat{=}$   $\left( \begin{array}{c} \textit{InitializePhase} \\ \square \\ \textit{CleanupPhase} \end{array} \right) ; \textit{Methods}$

$\bullet (\textit{Init} ; \textit{Methods}) \triangle (\textit{end\_mission\_app} . \textit{MainMissionMID} \longrightarrow \mathbf{Skip})$

**end**

## 5.2 Schedulables of MainMission

**section** *NestedMissionSequencerApp* **parents** *TopLevelMissionSequencerChan*,  
*MissionId*, *MissionIds*, *SchedulableId*, *SchedulableIds*, *NestedMissionSequencerClass*, *MethodCallBindingChannels*

**process** *NestedMissionSequencerApp*  $\hat{=}$   
*name* : *String* • **begin**

*GetNextMission*  $\hat{=}$  **var** *ret* : *MissionID* •  
 $\left( \begin{array}{l} \textit{getNextMissionCall} . \textit{NestedMissionSequencerSID} \longrightarrow \\ \textit{ret} := \textit{this} . \textit{getNextMission}(); \\ \textit{getNextMissionRet} . \textit{NestedMissionSequencerSID} ! \textit{ret} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

*Methods*  $\hat{=}$   
 $(\textit{GetNextMission}) ; \textit{Methods}$

•  $(\textit{Methods}) \triangle (\textit{end\_sequencer\_app} . \textit{NestedMissionSequencerSID} \longrightarrow \mathbf{Skip})$

**end**

**section** *NestedMissionSequencerClass* **parents** *scj\_prelude*, *SchedulableId*, *SchedulableIds*, *SafeletChannels*, *MethodCallBindingChannels*, *MissionId*, *MissionIds*

**class** *NestedMissionSequencerClass*  $\hat{=}$  **begin**

---

**state** *State*  
*releases* :  $\mathbb{Z}$

---

**state** *State*

---

**initial** *Init*  
*State* '  
*releases* = 0

---

**protected** *getNextMission*  $\hat{=}$  **var** *ret* : *MissionID* •

$$\left( \begin{array}{l} \text{if } releases = 0 \longrightarrow \\ \quad \left( \begin{array}{l} releases := releases + 1; \\ ret := NestedMissionAMID \end{array} \right) \\ \quad \square \neg releases = 0 \longrightarrow \\ \quad \quad \text{if } releases = 1 \longrightarrow \\ \quad \quad \quad \left( \begin{array}{l} releases := releases + 1; \\ ret := NestedMissionBMID \end{array} \right) \\ \quad \quad \quad \square \neg releases = 1 \longrightarrow \\ \quad \quad \quad \quad (ret := nullMissionId) \\ \quad \text{fi} \\ \text{fi} \end{array} \right)$$

• **Skip**

**end**

### 5.3 NestedMissionA

**section** *NestedMissionAApp* **parents** *scj\_prelude*, *MissionId*, *MissionIds*,  
*SchedulableId*, *SchedulableIds*, *MissionChan*, *SchedulableMethChan*, *NestedMissionAMethChan*,  
*MethodCallBindingChannels*

**process** *NestedMissionAApp*  $\hat{=}$  **begin**

*InitializePhase*  $\hat{=}$   

$$\left( \begin{array}{l} \textit{initializeCall} . \textit{NestedMissionAMID} \longrightarrow \\ \textit{register} ! \textit{MT1SID} ! \textit{NestedMissionAMID} \longrightarrow \\ \textit{initializeRet} . \textit{NestedMissionAMID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

*CleanupPhase*  $\hat{=}$   

$$\left( \begin{array}{l} \mathbf{var} \ \mathbb{B} : \textit{ret} \bullet \textit{cleanupMissionCall} . \textit{NestedMissionAMID} \longrightarrow \\ \textit{cleanupMissionRet} . \textit{NestedMissionAMID} ! \mathbf{True} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

*Methods*  $\hat{=}$   $\left( \begin{array}{c} \textit{InitializePhase} \\ \square \\ \textit{CleanupPhase} \end{array} \right) ; \textit{Methods}$

$\bullet (\textit{Init} ; \textit{Methods}) \triangle (\textit{end\_mission\_app} . \textit{NestedMissionAMID} \longrightarrow \mathbf{Skip})$

**end**

## 5.4 Schedulables of NestedMissionA

**section** *MT1App* **parents** *ManagedThreadChan, SchedulableId, SchedulableIds, MethodCallBindingChannels*

**process** *MT1App*  $\hat{=}$  **begin**

*Run*  $\hat{=}$   
$$\left( \begin{array}{l} \textit{runCall} . \textit{MT1SID} \longrightarrow \\ \mathbf{Skip}; \\ \textit{runRet} . \textit{MT1SID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

*Methods*  $\hat{=}$   
 $(\textit{Run}) ; \textit{Methods}$

•  $(\textit{Methods}) \triangle (\textit{end\_managedThread\_app} . \textit{MT1SID} \longrightarrow \mathbf{Skip})$

**end**

## 5.5 NestedMissionB

**section** *NestedMissionBApp* **parents** *scj\_prelude*, *MissionId*, *MissionIds*,  
*SchedulableId*, *SchedulableIds*, *MissionChan*, *SchedulableMethChan*, *NestedMissionBMethChan*,  
*MethodCallBindingChannels*

**process** *NestedMissionBApp*  $\hat{=}$  **begin**

*InitializePhase*  $\hat{=}$   

$$\left( \begin{array}{l} \text{initializeCall} . \text{NestedMissionBMID} \longrightarrow \\ \text{register} ! \text{MT2SID} ! \text{NestedMissionBMID} \longrightarrow \\ \text{initializeRet} . \text{NestedMissionBMID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

*CleanupPhase*  $\hat{=}$   

$$\left( \begin{array}{l} \mathbf{var} \ \mathbb{B} : \text{ret} \bullet \text{cleanupMissionCall} . \text{NestedMissionBMID} \longrightarrow \\ \text{cleanupMissionRet} . \text{NestedMissionBMID} ! \mathbf{True} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

*Methods*  $\hat{=}$   $\left( \begin{array}{c} \text{InitializePhase} \\ \square \\ \text{CleanupPhase} \end{array} \right) ; \text{Methods}$

$\bullet (\text{Init} ; \text{Methods}) \triangle (\text{end\_mission\_app} . \text{NestedMissionBMID} \longrightarrow \mathbf{Skip})$

**end**

## 5.6 Schedulables of NestedMissionB

**section** *MT2App* **parents** *ManagedThreadChan, SchedulableId, SchedulableIds, MethodCallBindingChannels*

**process** *MT2App*  $\hat{=}$  **begin**

*Run*  $\hat{=}$   
$$\left( \begin{array}{l} \textit{runCall} . \textit{MT2SID} \longrightarrow \\ \mathbf{Skip}; \\ \textit{runRet} . \textit{MT2SID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

*Methods*  $\hat{=}$   
 $(\textit{Run}) ; \textit{Methods}$

•  $(\textit{Methods}) \triangle (\textit{end\_managedThread\_app} . \textit{MT2SID} \longrightarrow \mathbf{Skip})$

**end**