

EXTENDS *Naturals, Sequences, TLC, Messages*

**Message type constants**

CONSTANT

*E2Setup,*  
*E2SetupResponse,*  
*E2SetupFailure*

CONSTANT

*ResetRequest,*  
*ResetResponse*

CONSTANT

*RICSubscriptionRequest,*  
*RICSubscriptionResponse,*  
*RICSubscriptionFailure*

CONSTANT

*RICSubscriptionDeleteRequest,*  
*RICSubscriptionDeleteResponse,*  
*RICSubscriptionDeleteFailure*

CONSTANT

*RICControlRequest,*  
*RICControlResponse,*  
*RICControlFailure,*  
*RICServiceUpdate*

CONSTANT

*E2ConnectionUpdate,*  
*E2ConnectionUpdateAcknowledge,*  
*E2ConnectionUpdateFailure*

CONSTANT

*E2NodeConfigurationUpdate,*  
*E2NodeConfigurationUpdateAcknowledge,*  
*E2NodeConfigurationUpdateFailure*

**Failure cause constants**

CONSTANT

*MiscFailureUnspecified,*  
*MiscFailureControlProcessingOverload,*  
*MiscFailureHardwareFailure,*  
*MiscFailureOMIntervention*

CONSTANT

*ProtocolFailureUnspecified,*  
*ProtocolFailureTransferSyntaxError,*  
*ProtocolFailureAbstractSyntaxErrorReject,*  
*ProtocolFailureAbstractSyntaxErrorIgnoreAndNotify,*  
*ProtocolFailureMessageNotCompatibleWithReceiverState,*

*ProtocolFailureSemanticError,*  
*ProtocolFailureAbstractSyntaxErrorFalselyConstructedMessage*  
 CONSTANT  
*RICFailureUnspecified,*  
*RICFailureRANFunctionIDInvalid,*  
*RICFailureActionNotSupported,*  
*RICFailureExcessiveActions,*  
*RICFailureDuplicateAction,*  
*RICFailureDuplicateEvent,*  
*RICFailureFunctionResourceLimit,*  
*RICFailureRequestIDUnknown,*  
*RICFailureInconsistentActionSubsequentActionSequence,*  
*RICFailureControlMessageInvalid,*  
*RICFailureCallProcessIDInvalid*  
 CONSTANT  
*RICServiceFailureUnspecified,*  
*RICServiceFailureFunctionNotRequired,*  
*RICServiceFailureExcessiveFunctions,*  
*RICServiceFailureRICResourceLimit*  
 CONSTANT  
*TransportFailureUnspecified,*  
*TransportFailureTransportResourceUnavailable*

VARIABLE *sbConn*

VARIABLE *sbConnId*

---

*InitSBVars*  $\triangleq$   
 $\wedge sbConn = [c \in \{\} \mapsto [e2node \mapsto Nil, ricnode \mapsto Nil, messages \mapsto \langle \rangle]]$

---

*SBSend*(*c*, *m*)  $\triangleq sbConn' = [sbConn \text{ EXCEPT } ![c] = [sbConn[c] \text{ EXCEPT } !.messages = Append(sbConn[c].messages, m)]]$   
*SBReceive*(*c*)  $\triangleq sbConn' = [sbConn \text{ EXCEPT } ![c] = [sbConn[c] \text{ EXCEPT } !.messages = SubSeq(sbConn[c].messages, 1)]]$   
*SBReply*(*c*, *m*)  $\triangleq sbConn' = [sbConn \text{ EXCEPT } ![c] = [sbConn[c] \text{ EXCEPT } !.messages = Append(SubSeq(sbConn[c].messages, 1), m)]]$

---

*SBConnect*(*e2node*, *ricnode*)  $\triangleq$   
 $\wedge sbConnId' = sbConnId + 1$   
 $\wedge sbConn' = sbConn @@ (sbConnId' :> [e2node \mapsto e2node, ricnode \mapsto ricnode, messages \mapsto \langle \rangle])$   
*SBDisconnect*(*id*)  $\triangleq$   
 $\wedge sbConn' = [c \in \{v \in \text{DOMAIN } sbConn : v \neq id\} \mapsto sbConn[c]]$

---


$$SBNext \triangleq$$

$$\vee \exists c \in \text{DOMAIN } sbConn : SBDisconnect(c)$$


---

\\* Modification History  
\\* Last modified *Tue Aug 03 18:57:12 PDT 2021* by *jordanhalterman*  
\\* Created *Mon Jul 26 10:01:02 PDT 2021* by *jordanhalterman*