MODULE *JustInTimePaxos*

EXTENDS *Naturals*, *Sequences*, *FiniteSets*, *TLC*

The set of *Paxos* replicas
CONSTANT *Replicas*

The set of *Paxos* clients
CONSTANT *Clients*

An empty value
CONSTANT *Nil*

Client request/response types+
CONSTANTS
MWriteRequest,
MWriteResponse,
MReadRequest,
MReadResponse

Server request/response types
CONSTANTS
MRepairRequest,
MRepairResponse,
MAbortRequest,
MAbortResponse,
MViewChangeRequest,
MViewChangeResponse,
MStartViewRequest

Replica roles
CONSTANTS
SNormal,
SAborting,
SViewChange

Entry types
CONSTANTS
TValue,
TNoOp

VARIABLE *replicas*

$globalVars \triangleq \langle replicas \rangle$

VARIABLE *messages*

$messageVars \triangleq \langle messages \rangle$

VARIABLE $cTime$

VARIABLE $cViewID$

VARIABLE $cSeqNum$

VARIABLE $cResps$

VARIABLE $cWrites$

VARIABLE $cReads$

$clientVars \triangleq \langle cTime,\ cViewID,\ cSeqNum,\ cResps,\ cWrites,\ cReads \rangle$

VARIABLE $rStatus$

VARIABLE $rLog$

VARIABLE $rViewID$

VARIABLE $rSeqNum$

VARIABLE $rLastView$

VARIABLE $rViewChanges$

VARIABLE $rAbortSeqNum$

VARIABLE $rAbortResps$

$replicaVars \triangleq \langle rStatus,\ rLog,\ rViewID,\ rSeqNum,\ rLastView,\ rViewChanges,\ rAbortSeqNum,\ rAbortResps \rangle$

VARIABLE $transitions$

$vars \triangleq \langle globalVars,\ messageVars,\ clientVars,\ replicaVars,\ transitions \rangle$

Helpers

RECURSIVE $SeqFromSet(\_)$
$SeqFromSet(S) \triangleq$
  IF $S = \{\}$ THEN $\langle\rangle$
  ELSE LET $x \triangleq$ CHOOSE $x \in S$: TRUE
       IN $\langle x \rangle \circ SeqFromSet(S \setminus \{x\})$

$Max(s) \triangleq$ CHOOSE $x \in s : \forall\, y \in s : x \geq y$

$IsQuorum(s) \triangleq Cardinality(s) * 2 \geq Cardinality(Replicas)$

$Quorums \triangleq \{r \in$ SUBSET $Replicas : IsQuorum(r)\}$

$Primary(v) \triangleq replicas[(v\%Len(replicas)) + (\text{IF } v \geq Len(replicas) \text{ THEN } 1 \text{ ELSE } 0)]$

$IsPrimary(r) \triangleq Primary(rViewID[r]) = r$

$Replace(l,\, i,\, x) \triangleq [j \in 1 \,..\, Max(\{Len(l),\, i\}) \mapsto \text{IF } j = i \text{ THEN } x \text{ ELSE } l[j]]$

Messaging helpers

$Sends(ms) \triangleq messages' = messages \cup ms$

$Send(m) \triangleq Sends(\{m\})$

$Replies(req,\, resps) \triangleq messages' = (messages \cup resps) \setminus \{req\}$

$Reply(req,\, resp) \triangleq Replies(req,\, \{resp\})$

$Discard(m) \triangleq messages' = messages \setminus \{m\}$

$Write(c) \triangleq$
- $\wedge\ cTime' = cTime + 1$
- $\wedge\ cSeqNum' = [cSeqNum \text{ EXCEPT } ![c] = cSeqNum[c] + 1]$
- $\wedge\ Sends(\{[src \mapsto c,\ dest \mapsto r,\ type \mapsto MWriteRequest,\ viewID \mapsto cViewID[c],\ seqNum \mapsto cSeqNum'[c],\ timestamp \mapsto cTime'] : r \in Replicas\})$
- $\wedge$ UNCHANGED $\langle globalVars,\, replicaVars,\, cViewID,\, cResps,\, cWrites,\, cReads\rangle$

$Read(c) \triangleq$
- $\wedge\ Sends(\{[src \mapsto c,\ dest \mapsto r,\ type \mapsto MReadRequest,\ viewID \mapsto cViewID[c]] : r \in Replicas\})$
- $\wedge$ UNCHANGED $\langle globalVars,\, replicaVars,\, cTime,\, cSeqNum,\, cResps,\, cWrites,\, cReads\rangle$

$HandleWriteResponse(c,\, r,\, m) \triangleq$
- $\wedge\ \vee\ \wedge\ m.viewID = cViewID[c]$
  - $\wedge$ IF $m.seqNum \notin$ DOMAIN $cResps[c][r]$ THEN
    - $cResps' = [cResps \text{ EXCEPT } ![c] = [cResps[c] \text{ EXCEPT } ![r] = cResps[c][r] \,@@\, (m.seqNum :> m)]]$
    - ELSE
    - $cResps' = [cResps \text{ EXCEPT } ![c] = [cResps[c] \text{ EXCEPT } ![r] = [cResps[c][r] \text{ EXCEPT } ![m.seqNum]$
  - $\wedge$ LET
    - $allResps \triangleq \{cResps[c][r][r1] : r1 \in \{r2 \in Replicas : r2 \in \text{DOMAIN } cResps[c][r]\}\}$
    - $isCommitted \triangleq \{r1.src : r1 \in \{r2 \in allResps : r2.succeeded\}\} \in Quorums$

```
        IN
            ∧ ∨ ∧ isCommitted
                  ∧ cWrites' = [cWrites EXCEPT ![c] = cWrites[c] ∪ {m}]
              ∨ ∧ ¬isCommitted
                  ∧ UNCHANGED ⟨cWrites⟩
              ∧ UNCHANGED ⟨cViewID, cSeqNum⟩
          ∨ ∧ m.viewID > cViewID[c]
            ∧ cViewID'  = [cViewID EXCEPT ![c] = m.viewID]
            ∧ cSeqNum'  = [cSeqNum EXCEPT ![c] = 0]
            ∧ cResps'   = [cResps  EXCEPT ![c]  = [i ∈ Replicas ↦ {}]]
            ∧ UNCHANGED ⟨cWrites⟩
          ∨ ∧ m.viewID < cViewID[c]
            ∧ UNCHANGED ⟨cWrites⟩
      ∧ Discard(m)
      ∧ UNCHANGED ⟨globalVars, replicaVars, cTime, cSeqNum, cReads⟩

HandleReadResponse(c, r, m) ≜
      ∧ ∨ ∧ m.viewID = cViewID[c]
            ∧ cReads' = [cReads EXCEPT ![c] = cReads[c] ∪ {m}]
            ∧ UNCHANGED ⟨cViewID, cSeqNum⟩
          ∨ ∧ m.viewID > cViewID[c]
            ∧ cViewID'  = [cViewID EXCEPT ![c] = m.viewID]
            ∧ cSeqNum'  = [cSeqNum EXCEPT ![c] = 0]
            ∧ UNCHANGED ⟨cReads⟩
          ∨ ∧ m.viewID < cViewID[c]
            ∧ UNCHANGED ⟨cViewID, cSeqNum, cReads⟩
      ∧ Discard(m)
      ∧ UNCHANGED ⟨globalVars, replicaVars, cTime, cSeqNum, cResps, cWrites⟩
```

Server request/response handling

```
Repair(r, c, m) ≜
      ∧ Replies(m, {[src      ↦ r,
                     dest     ↦ d,
                     type     ↦ MRepairRequest,
                     viewID   ↦ rViewID[r],
                     client   ↦ c,
                     seqNum ↦ rSeqNum[r][c] + 1] : d ∈ Replicas})

Abort(r, c, m) ≜
      ∧ IsPrimary(r)
      ∧ rStatus[r]      = SNormal
      ∧ rStatus'        = [rStatus      EXCEPT ![r]    = SAborting]
      ∧ rAbortResps'    = [rAbortResps  EXCEPT ![r]    = [rAbortResps[r] EXCEPT ![c] = {}]]
```

$\land\ rAbortSeqNum' = [rAbortSeqNum \text{ EXCEPT } ![r] = [rAbortSeqNum[r] \text{ EXCEPT } ![c] = m.seqNum]]$
$\land\ Replies(m, \{[src \mapsto r,$
$dest \mapsto d,$
$type \mapsto MAbortRequest,$
$viewID \mapsto rViewID[r],$
$client \mapsto c,$
$seqNum \mapsto m.seqNum] : d \in Replicas\})$

$HandleWriteRequest(r, c, m) \triangleq$
$\land\ rStatus[r] = SNormal$
$\land\ \lor\ \land\ m.viewID = rViewID[r]$
$\land$ LET
$isSequential \triangleq m.seqNum = rSeqNum[r][c] + 1$
$isLinear \triangleq \forall\, i \in \text{DOMAIN } rLog[r] : \forall\, e \in rLog[r][i] : m.timestamp > e.timestamp$
IN
$\lor\ \land\ isSequential$
$\land\ isLinear$
$\land\ rLog' = [rLog \text{ EXCEPT } ![r] = [$
$rLog[r] \text{ EXCEPT } ![c] =$
$Append(rLog[r][c], [type \mapsto TValue,$
$value \mapsto m.value,$
$timestamp \mapsto m.timestamp])]]$
$\land\ rSeqNum' = [rSeqNum \text{ EXCEPT } ![r] = [rSeqNum[r] \text{ EXCEPT } ![c] = m.seqNum]]$
$\land\ Reply(m, [src \mapsto r,$
$dest \mapsto c,$
$type \mapsto MWriteResponse,$
$seqNum \mapsto m.seqNum,$
$viewID \mapsto rViewID[r],$
$succeeded \mapsto \text{TRUE}])$
$\lor\ \land\ \lor \neg isSequential$
$\lor \neg isLinear$
$\land\ \lor\ \land\ IsPrimary(r)$
$\land\ Abort(r, c, m)$
$\lor\ \land \neg IsPrimary(r)$
$\land\ Reply(m, [src \mapsto r,$
$dest \mapsto c,$
$type \mapsto MWriteResponse,$
$seqNum \mapsto m.seqNum,$
$viewID \mapsto rViewID[r],$
$succeeded \mapsto \text{FALSE}])$
$\land$ UNCHANGED $\langle rLog \rangle$
$\lor\ \land\ m.viewID < rViewID[r]$
$\land\ Reply(m, [src \mapsto r,$
$dest \mapsto c,$
$type \mapsto MWriteResponse,$

```
                        seqNum    ↦ m.seqNum,
                        viewID    ↦ rViewID[r],
                        succeeded ↦ FALSE])
          ∧ UNCHANGED ⟨rLog⟩
    ∧ UNCHANGED ⟨globalVars, clientVars, rStatus, rViewID, rLastView, rViewChanges⟩

HandleReadRequest(r, c, m) ≜
    ∧ rStatus[r] = SNormal
    ∧ Len(rLog[r]) > 0
    ∧ Reply(m, [src       ↦ r,
                dest      ↦ c,
                type      ↦ MReadResponse,
                viewID    ↦ rViewID[r],
                primary   ↦ IsPrimary(r),
                index     ↦ Len(rLog[r]),
                checksum  ↦ rLog[r][Len(rLog[r])].checksum,
                succeeded ↦ TRUE])
    ∧ UNCHANGED ⟨globalVars, clientVars, rStatus, rLog, rViewID, rLastView, rViewChanges⟩

HandleRepairRequest(r, s, m) ≜
    ∧ m.viewID = rViewID[r]
    ∧ IsPrimary(r)
    ∧ rStatus[r] = SNormal
    ∧ LET index ≜ Len(rLog[r][m.client]) + 1 − (rSeqNum[r] − m.seqNum)
      IN
        ∧ ∨ ∧ index ≤ Len(rLog[r][m.client])
            ∧ Reply(m, [src       ↦ r,
                        dest      ↦ s,
                        type      ↦ MRepairResponse,
                        viewID    ↦ rViewID[r],
                        client    ↦ m.client,
                        seqNum    ↦ m.seqNum])
            ∧ UNCHANGED ⟨rStatus, rAbortResps, rAbortSeqNum⟩
          ∨ ∧ index = Len(rLog[r][m.client]) + 1
            ∧ Abort(r, m.client, m)
    ∧ UNCHANGED ⟨globalVars, clientVars⟩

HandleRepairResponse(r, s, m) ≜
    ∧ HandleWriteRequest(r, m.client, [m EXCEPT !.src = m.client])

HandleAbortRequest(r, s, m) ≜
    ∧ m.viewID = rViewID[r]
    ∧ rStatus[r] ∈ {SNormal, SAborting}
    ∧ LET index ≜ Len(rLog[r][m.client]) + 1 − (rSeqNum[r] − m.seqNum)
      IN
        ∧ index ≤ Len(rLog[r][m.client]) + 1
```

$\land\ rLog' = [rLog \text{ EXCEPT } ![r] = [rLog[r] \text{ EXCEPT } ![m.client] = Replace(rLog[r][m.client], index, [typ$
$\land\ \lor\ \land\ m.seqNum > rSeqNum[r][m.client]$
$\land\ rSeqNum' = [rSeqNum \text{ EXCEPT } ![r] = [rSeqNum[r] \text{ EXCEPT } ![m.client] = m.seqNum]]$
$\lor\ \land\ m.seqNum \leq rSeqNum[r][m.client]$
$\land\ \text{UNCHANGED } \langle rSeqNum \rangle$
$\land\ Replies(m, \{[src \mapsto r,$
$dest \mapsto Primary(rViewID[r]),$
$type \mapsto MAbortResponse,$
$viewID \mapsto rViewID[r],$
$seqNum \mapsto m.seqNum],$
$[src \mapsto r,$
$dest \mapsto Primary(rViewID[r]),$
$type \mapsto MWriteResponse,$
$viewID \mapsto rViewID[r],$
$seqNum \mapsto m.seqNum,$
$succeeded \mapsto \text{FALSE}]\})$
$\land\ \text{UNCHANGED } \langle globalVars, clientVars, rStatus, rViewID, rLastView, rViewChanges \rangle$

$HandleAbortResponse(r, s, m) \triangleq$
$\land\ rStatus[r] = SAborting$
$\land\ m.viewID = rViewID[r]$
$\land\ IsPrimary(r)$
$\land\ m.seqNum = rAbortSeqNum[r][m.client]$
$\land\ rAbortResps' = [rAbortResps \text{ EXCEPT } ![r] = [rAbortResps[r] \text{ EXCEPT } ![m.client] = rAbortResps[r][m.cl$
$\land\ \text{LET } resps \triangleq \{res.src : res \in \{resp \in rAbortResps'[r][m.client] :$
$\land\ resp.viewID = rViewID[r]$
$\land\ resp.seqNum = rAbortSeqNum[r][m.client]\}\}$
$isQuorum \triangleq r \in resps \land resps \in Quorums$
IN
$\lor\ \land\ isQuorum$
$\land\ rStatus' = [rStatus \text{ EXCEPT } ![r] = [rStatus[r] \text{ EXCEPT } ![m.client] = SNormal]]$
$\lor\ \land\ \neg isQuorum$
$\land\ \text{UNCHANGED } \langle rStatus \rangle$
$\land\ \text{UNCHANGED } \langle globalVars, clientVars \rangle$

$ChangeView(r) \triangleq$
$\land\ Sends(\{[src \mapsto r,$
$dest \mapsto d,$
$type \mapsto MViewChangeRequest,$
$viewID \mapsto rViewID[r] + 1] : d \in Replicas\})$
$\land\ \text{UNCHANGED } \langle globalVars, clientVars, replicaVars \rangle$

$HandleViewChangeRequest(r, s, m) \triangleq$
$\land\ rViewID[r] < m.viewID$
$\land\ rViewID' = [rViewID \text{ EXCEPT } ![r] = m.viewID]$
$\land\ rStatus' = [rStatus \text{ EXCEPT } ![r] = SViewChange]$

$\land\ rViewChanges' = [rViewChanges \text{ EXCEPT } ![r] = \{\}]$
$\land\ Reply(m, [src \mapsto r,$
$\quad dest \mapsto Primary(m.viewID),$
$\quad type \mapsto MViewChangeResponse,$
$\quad viewID \mapsto m.viewID,$
$\quad lastViewID \mapsto rLastView[r],$
$\quad logs \mapsto rLog[r]])$
$\land$ UNCHANGED $\langle globalVars, clientVars, rLog, rSeqNum, rAbortSeqNum, rAbortResps, rLastView\rangle$

$HandleViewChangeResponse(r, s, m) \triangleq$
$\land\ IsPrimary(r)$
$\land\ rViewID[r] = m.viewID$
$\land\ rStatus[r] = SViewChange$
$\land\ rViewChanges' = [rViewChanges \text{ EXCEPT } ![r] = rViewChanges[r] \cup \{m\}]$
$\land$ LET $viewChanges \triangleq \{v \in rViewChanges'[r][m.client] : \land v.viewID = rViewID[r]\}$
$\quad viewSources \triangleq \{v.src : v \in viewChanges\}$
$\quad isQuorum \triangleq r \in viewSources \land viewSources \in Quorums$
$\quad lastViews \triangleq \{v.lastViewID : v \in viewChanges\}$
$\quad lastView \triangleq (\text{CHOOSE } v1 \in lastViews : \forall\, v2 \in lastViews : v2 \leq v1)$
$\quad viewLogs \triangleq [c \in Clients \mapsto \{v1.logs[c] : v1 \in \{v2 \in viewChanges : v2.lastView = lastView\}$
$\quad mergeEnts(es) \triangleq$
$\qquad$ IF $es = \{\} \lor \exists\, e \in es : r.type = TNoOp$ THEN
$\qquad\quad [type \mapsto TNoOp]$
$\qquad$ ELSE
$\qquad\quad$ CHOOSE $e \in es : e.type \neq TNoOp$
$\quad range(ls) \triangleq Max(\{Len(l) : l \in ls\})$
$\quad entries(ls, i) \triangleq \{l[i] : l \in \{k \in ls : i \leq Len(k)\}\}$
$\quad mergeLogs(ls) \triangleq [i \in 1 \,..\, range(ls) \mapsto mergeEnts(entries(ls, i))]$
IN
$\qquad \lor \land isQuorum$
$\qquad\quad \land Replies(m, \{[src \mapsto r,$
$\qquad\qquad dest \mapsto d,$
$\qquad\qquad type \mapsto MStartViewRequest,$
$\qquad\qquad viewID \mapsto rViewID[r],$
$\qquad\qquad logs \mapsto [c \in Clients \mapsto mergeLogs(viewLogs[c])]] : d \in Replicas\})$
$\qquad \lor \land \neg isQuorum$
$\qquad\quad \land Discard(m)$
$\land$ UNCHANGED $\langle globalVars, clientVars, rStatus, rViewID, rLog, rSeqNum, rAbortSeqNum, rAbortResps, r$

$HandleStartViewRequest(r, s, m) \triangleq$
$\land\ \lor rViewID[r] < m.viewID$
$\quad \lor \land rViewID[r] = m.viewID$
$\qquad \land rStatus[r] = SViewChange$
$\land\ rLog' = [rLog \text{ EXCEPT } ![r] = m.log]$
$\land\ rStatus' = [rStatus \text{ EXCEPT } ![r] = SNormal]$

$\land\ rViewID' = [rViewID \text{ EXCEPT } ![r] = m.viewID]$
$\land\ rLastView' = [rLastView \text{ EXCEPT } ![r] = m.viewID]$
$\land\ Discard(m)$
$\land$ UNCHANGED $\langle globalVars,\ clientVars,\ rViewChanges\rangle$

$InitMessageVars \triangleq$
$\land\ messages = \{\}$

$InitClientVars \triangleq$
$\land\ cTime = 0$
$\land\ cViewID = [c \in Clients \mapsto 1]$
$\land\ cSeqNum = [c \in Clients \mapsto 0]$
$\land\ cResps = [c \in Clients \mapsto [r \in Replicas \mapsto [s \in \{\} \mapsto [index \mapsto 0,\ checksum \mapsto Nil]]]]$
$\land\ cWrites = [c \in Clients \mapsto \{\}]$
$\land\ cReads = [c \in Clients \mapsto \{\}]$

$InitReplicaVars \triangleq$
$\land\ replicas = SeqFromSet(Replicas)$
$\land\ rStatus = [r \in Replicas \mapsto SNormal]$
$\land\ rLog = [r \in Replicas \mapsto [c \in Clients \mapsto \langle\rangle]]$
$\land\ rSeqNum = [r \in Replicas \mapsto [c \in Clients \mapsto 0]]$
$\land\ rAbortSeqNum = [r \in Replicas \mapsto [c \in Clients \mapsto 0]]$
$\land\ rAbortResps = [r \in Replicas \mapsto [c \in Clients \mapsto \{\}]]$
$\land\ rViewID = [r \in Replicas \mapsto 1]$
$\land\ rLastView = [r \in Replicas \mapsto 1]$
$\land\ rViewChanges = [r \in Replicas \mapsto \{\}]$

$Init \triangleq$
$\land\ InitMessageVars$
$\land\ InitClientVars$
$\land\ InitReplicaVars$
$\land\ transitions = 0$

The type invariant checks that no read ever reads a different value than a previous write
$Inv \triangleq$ TRUE *TODO*

$Transition \triangleq transitions' = transitions + 1$

$Next \triangleq$
$\lor \exists\, c \in Clients :$
$\land\ Write(c)$
$\land\ Transition$
$\lor \exists\, c \in Clients :$

$\quad \land \mathit{Read}(c)$
$\quad \land \mathit{Transition}$
$\lor \exists\, r \in \mathit{Replicas} :$
$\quad \land \mathit{ChangeView}(r)$
$\quad \land \mathit{Transition}$
$\lor \exists\, m \in \mathit{messages} :$
$\quad \land m.\mathit{type} = \mathit{MWriteRequest}$
$\quad \land \mathit{HandleWriteRequest}(m.\mathit{dest},\ m.\mathit{src},\ m)$
$\quad \land \mathit{Transition}$
$\lor \exists\, m \in \mathit{messages} :$
$\quad \land m.\mathit{type} = \mathit{MWriteResponse}$
$\quad \land \mathit{HandleWriteResponse}(m.\mathit{dest},\ m.\mathit{src},\ m)$
$\quad \land \mathit{Transition}$
$\lor \exists\, m \in \mathit{messages} :$
$\quad \land m.\mathit{type} = \mathit{MReadRequest}$
$\quad \land \mathit{HandleReadRequest}(m.\mathit{dest},\ m.\mathit{src},\ m)$
$\quad \land \mathit{Transition}$
$\lor \exists\, m \in \mathit{messages} :$
$\quad \land m.\mathit{type} = \mathit{MReadResponse}$
$\quad \land \mathit{HandleReadResponse}(m.\mathit{dest},\ m.\mathit{src},\ m)$
$\quad \land \mathit{Transition}$
$\lor \exists\, m \in \mathit{messages} :$
$\quad \land m.\mathit{type} = \mathit{MRepairRequest}$
$\quad \land \mathit{HandleRepairRequest}(m.\mathit{dest},\ m.\mathit{src},\ m)$
$\quad \land \mathit{Transition}$
$\lor \exists\, m \in \mathit{messages} :$
$\quad \land m.\mathit{type} = \mathit{MRepairResponse}$
$\quad \land \mathit{HandleRepairResponse}(m.\mathit{dest},\ m.\mathit{src},\ m)$
$\quad \land \mathit{Transition}$
$\lor \exists\, m \in \mathit{messages} :$
$\quad \land m.\mathit{type} = \mathit{MAbortRequest}$
$\quad \land \mathit{HandleAbortRequest}(m.\mathit{dest},\ m.\mathit{src},\ m)$
$\quad \land \mathit{Transition}$
$\lor \exists\, m \in \mathit{messages} :$
$\quad \land m.\mathit{type} = \mathit{MAbortResponse}$
$\quad \land \mathit{HandleAbortResponse}(m.\mathit{dest},\ m.\mathit{src},\ m)$
$\quad \land \mathit{Transition}$
$\lor \exists\, m \in \mathit{messages} :$
$\quad \land m.\mathit{type} = \mathit{MViewChangeRequest}$
$\quad \land \mathit{HandleViewChangeRequest}(m.\mathit{dest},\ m.\mathit{src},\ m)$
$\quad \land \mathit{Transition}$
$\lor \exists\, m \in \mathit{messages} :$
$\quad \land m.\mathit{type} = \mathit{MViewChangeResponse}$
$\quad \land \mathit{HandleViewChangeResponse}(m.\mathit{dest},\ m.\mathit{src},\ m)$
$\quad \land \mathit{Transition}$

$\lor \exists\, m \in messages :$
$\land\ m.type = MStartViewRequest$
$\land\ HandleStartViewRequest(m.dest,\ m.src,\ m)$
$\land\ Transition$

$Spec \triangleq Init \land \Box[Next]_{vars}$

\ * Modification History
\ * Last modified *Tue Sep* 22 05:04:13 *PDT* 2020 by *jordanhalterman*
\ * Created *Fri Sep* 18 22:45:21 *PDT* 2020 by *jordanhalterman*