

```

define OroSepTarg {
  /*case a_X{
    condition wateryear==1922 .and. month<=dec
    value min(1000. + 0.5*max(S6(prevSep)-S6_WTS(prevSep) - 1000. -
DrainTar_buffer,0.), S6level5) }
  case b_X{
    condition always
    value min(1000. +
0.5*swp_allocation_final/tableA_total*max(S6(prevSep)-S6_WTS(prevSep) -
1000. - DrainTar_buffer,0.), S6level5) */
  /*case a_20{
    condition always
    value min(1000. + 0.2*max(S6(prevSep)-S6_WTS(prevSep) - 1000. -
DrainTar_buffer,0.), S6level5) */
  case a_X{
    condition wateryear==1922 .and. month<=dec
    value min(1000. + 0.5*max(S6(prevSep)-S6_WTS(prevSep) - 1000. -
DrainTar_buffer,0.), S6level5) }
  case b_X{
    condition always
    value min(1000. + 0.5 * (swp_allocation_final-tableA_loss)/SWP_TableA
      * max(S6(prevSep)-S6_WTS(prevSep) - 1000. -
DrainTar_buffer,0.), S6level5) }
  /*case a_default{
    condition always
    value min(1000. + 0.5*max(S6(prevSep)-S6_WTS(prevSep) - 1000. -
DrainTar_buffer,0.), S6level5) */
}
define OroNextMonTarg {
  value min(OroSepTarg + (max(S6(-1)-S6_WTS(-1) -
OroSepTarg,0.)/(Sep-month+1.))*(Sep-month), S6level5)
}

define S6level4 { ! An alternative method
  case JuntoSep {
    condition month >= jun
    value      OroNextMonTarg }
  case Other {
    condition always
    value      S6level5 }
}
/*
define S6level4 {
  case JunthruJul {
    condition month >= jun .and. month <= jul .and. S6(-1)-S6_WTS(-1) >
2000.0 !.and. S12(-1) < 700.0
    value      OroNextMonTarg }
  case Aug {
    condition month == aug .and. S6(-1)-S6_WTS(-1) > 2000.0 .and. S12(-
1) < 900.0
    value      OroNextMonTarg }
  case Sep {
    condition month >= sep.and. S6(-1)-S6_WTS(-1) > 2000.0 !.and. S12(-
1) < 800.0

```

```

        value      OroNextMonTarg }
    case Other {
        condition always
        value      S6level5 }

}
*/
define S6level4dv {alias S6level4 kind 'STORAGE-LEVEL' units 'TAF'}
define OroSepTarget {alias OroSepTarg kind 'STORAGE-LEVEL' units 'TAF'}
define OroPreSepStor {alias S6(prevSep) kind 'STORAGE' units 'TAF'}
define OroNextMonTarget {alias OroNextMonTarg kind 'STORAGE-LEVEL' units
'TAF'}

define S6level3 {value min(2470.,S6level4) }

define OroInf {value (I6)*cfs_taf }
define OroInfEst {sum(i=0,SEP-month,1) (I6(i))*cfs_taf(i) }
define OroDiv {value dem_d6_PWR }
define OroDivEst {sum(i=0,SEP-month,1) dem_d6_PWR(i) }

!define OroSupEst {value max(0.0,OroInfEst - OroDivEst + S6(-1)-S6_WTS(-
1) - S6level3(SEP-month))}
define OroSupEst {value max(0.0,OroInfEst - OroDivEst + S6(-1)-S6_WTS(-1)
- S6level3)}

define OroSupEstdv {alias OroSupEst kind 'FORECAST' units 'TAF'}

!define OroFlowTarg {value min((OroSupEst/cfs_taf)/(Sep-
month+1.),17000.)}
define OroFlowTarg {value (OroSupEst/cfs_taf)/(Sep-month+1.) }

define OroRuleCalc {
    case JunthruSep {
        condition month >= may .and. month <= sep
        value      OroInf - OroDiv + S6(-1)-S6_WTS(-1) -
OroFlowTarg*cfs_taf }
    case Other {
        condition always
        value      S6level3}}
define OroRuleCalcdv {alias OroRuleCalc kind 'RULE' units 'TAF'}

define S6level3adj {value min(max(OroRuleCalc,S6level2),S6level4)}
define S6level3dv {alias S6level3adj kind 'STORAGE-LEVEL' units 'TAF'}

define OroFish {
/*    case MaythruSep {
        condition month >= may .and. month <= sep
        value      s6level3adj }
    case MarApr {
        condition month == mar .or. month == apr
!        value      2470. }
        value      2000. }*/ !reducing the Oroville Mar Apr storage
target by 450 TAF, to account for releases from "river valve"
    case Other {

```

```
condition always  
value 2000.}}
```

```
define OroFishdv {alias OroFish kind 'RULE' units 'TAF'}
```

```
goal S6Zone3 {S6_3 < max(S6level3adj - S6level2,0.)}  
goal S6Zone4 {S6_4 < max(S6level4 - S6level3adj,0.)}  
goal S6Zone5 {S6_5 < max(S6level5 - S6level4,0.)}  
goal S6Zone6 {S6_6 < max(S6level6 - S6level5,0.)}
```