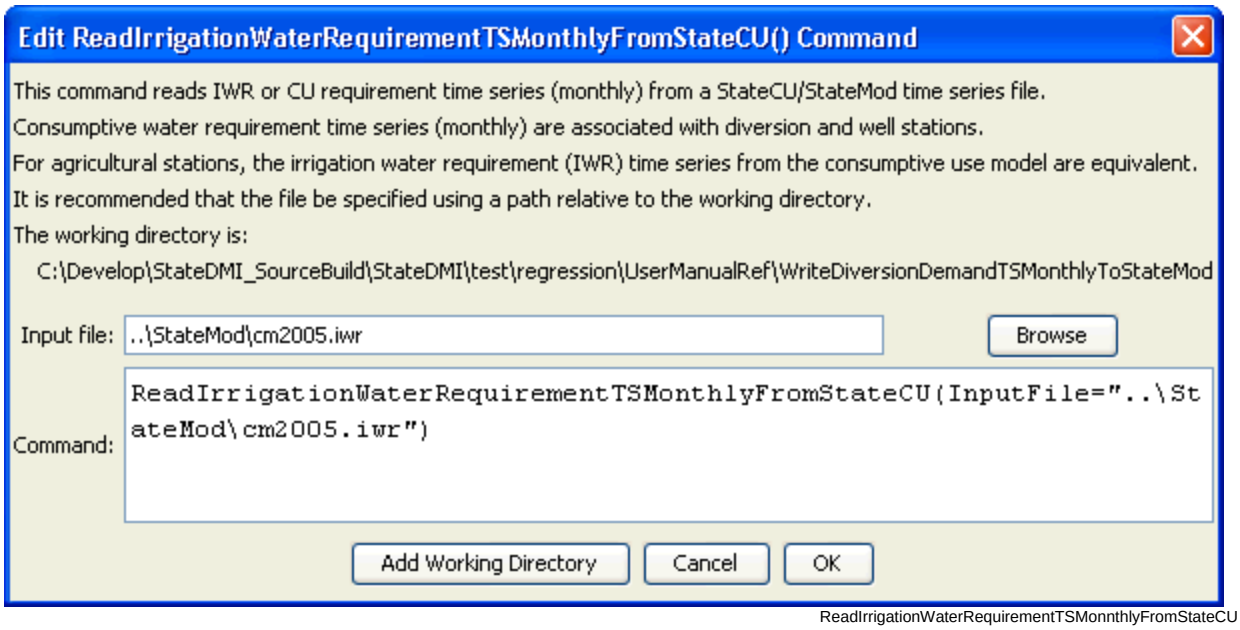
Command Reference: ReadIrrigationWaterRequirementTSMonthlyFrom StateCU()

Read irrigation water requirement time series data from a StateCU file

StateMod Command
Version 3.09.01, 2010-02-01

The `ReadIrrigationWaterRequirementTSMonthlyFromStateCU()` command reads irrigation water requirement time series data from a StateCU irrigation water requirement time series file and defines the data in memory. Currently this command is meant to read the IWR time series for use in estimating average efficiencies and demands for StateMod – it is not supported in StateCU commands (e.g., to read and modify the time series file). All time series are read, whether or not they match the list of diversion stations. The following dialog is used to edit the command and illustrates the syntax of the command.



ReadIrrigationWaterRequirementTSMonnthlyFromStateCU

ReadIrrigationWaterRequirementTSMonthlyFromStateCU() Command Editor

The command syntax is as follows:

```
ReadIrrigationWaterRequirementTSMonthlyFromStateCU(Parameter=Value,…)
```

Command Parameters

Parameter	Description	Default
`InputFile`	The name of the StateCU irrigation water requirement file (StateMod time series format) to read.	None – must be specified.

The following abbreviated command file illustrates how irrigation water requirement time series can be processed into average demand time series:

```
StartLog(LogFile="Cddm.commands.StateDMI.log")
# Cddm.commands.StateDMI
#
#  StateDMI command file to create the Calculated demand file
#
#
#  Step 1 - set the output period, used to compute averages...
#
SetOutputPeriod(OutputStart="10/1908",OutputEnd="09/2005")
SetOutputYearType(OutputYearType=Water)
#
#  Step 2 - read historical diversion file -defines structures for *.ddm file
#           plus read *.ddh file
#
ReadDiversionStationsFromStateMod(InputFile="..\StateMod\cm2005.dds")
ReadDiversionHistoricalTSMonthlyFromStateMod(InputFile="..\StateMod\cm2005.ddh")
#
#  Step 3 - read StateCU *.iwr and *.def files (irrigation requirements and average efficiencies)
#
ReadIrrigationWaterRequirementTSMonthlyFromStateCU(InputFile="..\StateMod\cm2005.iwr")
# calculateDiversionStationEfficiencies(ID="*",EffMin=0,EffMax=60,
  EffCalcStart=10/1974,EffCalcEnd=9/2004,LEZeroInAverage=False)
SetDiversionStationsFromList(ListFile="cm2005.def",IDCol="1",EffMonthlyCol="2",
  Delim="Space",MergeDelim=True)
#
#  Step 4 - determine calculated demand = iwr/efficiency
#         - take max of calculated demand and historical diversion
#
CalculateDiversionDemandTSMonthly(ID="*")
CalculateDiversionDemandTSMonthlyAsMax(ID="*")
#
#  Step 5 - set carriers nodes demand to 0, set full demand and summary demand nodes
#
#  set carrier "transbasin" diversion to Divide Creek to "0", use operating rules to satisfy demand
SetDiversionDemandTSMonthlyConstant(ID="724721",Constant=0)
# place summary demand at the Moffat Tunnel, zero out collection points
SetDiversionDemandTSMonthly(ID="514655",TSID="514655..DivTotal.Month~StateMod~514655.stm")
… similar commands omitted…
#
#   Step 6 - set calculated demand to historic for structures whose historical acreage is
#             different from current
#
SetDiversionDemandTSMonthly(ID="360687",TSID="360687..DivTotal.MONTH~StateMod~..\StateMod\cm2005H.ddm")
SetDiversionDemandTSMonthly(ID="360725",TSID="360725..DivTotal.MONTH~StateMod~..\StateMod\cm2005H.ddm")
…similar commands omitted…
#
#  Set Ute WCD demand node structure and set other structures to zero
SetDiversionDemandTSMonthly(ID="950020",TSID="950020..DivTotal.Month~StateMod~950020.stm")
SetDiversionDemandTSMonthlyConstant(ID="950030",Constant=0)
… similar commands omitted…
#
#  Set Orchard Mesa Check
SetDiversionDemandTSMonthly(ID="950003",TSID="950003..DivTotal.MONTH~StateMod~..\StateMod\cm2005H.ddm")
#
# Set Excess HUP node demands for Homestake, Dillon, Williams Fork, and Wolford Reservoirs
SetDiversionDemandTSMonthlyConstant(ID="954516D",Constant=999999)
…similar commands omitted…
#  Step 7 - write out calculated demand file
#
WriteDiversionDemandTSMonthlyToStateMod(OutputFile="..\StateMod\cm2005C.ddm")
#
# Check the results
CheckDiversionDemandTSMonthly(ID="*")
WriteCheckFile(OutputFile="Cddm.commands.StateDMI.check.html")
```