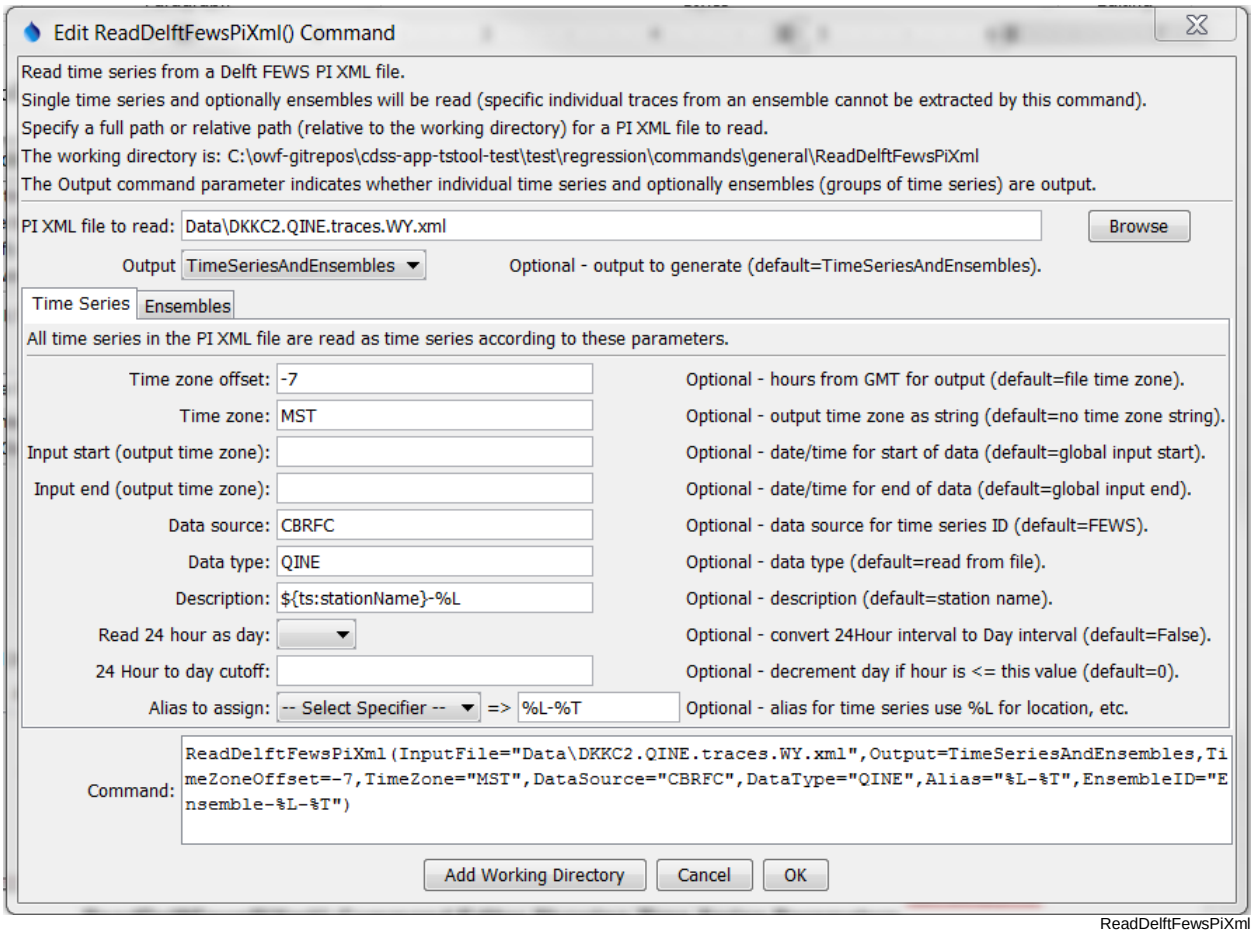
# Command Reference: ReadDelftFewsPiXml()

## Read all time series from a Delft FEWS PI XML File

Version 11.08.01, 2016-02-26

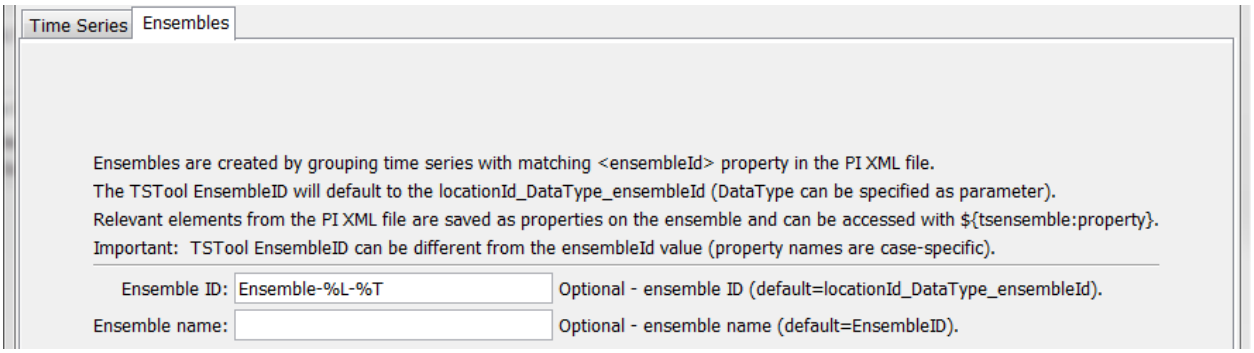
The `ReadDelftFewsPiXml()` command reads all the time series in a Delft FEWS PI XML file. See the **Delft FEWS Input Type Appendix** for information about the file format.

The following dialog is used to edit the command and illustrates the command syntax.




ReadDelftFewsPiXml

**ReadDelftFewsPiXml() Command Editor Showing Time Series Parameters**




ReadDelftFewsPiXml_Ensemble

**ReadDelftFewsPiXml() Command Editor Showing Ensemble Parameters**

The command syntax is as follows:

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

**Command Parameters**

| Parameter | Description | Default |
|---|---|---|
| `InputFile` | The name of the PI XML input file to read. Global property values can be used with the syntax `${PropertyName}`. The file can be a *.xml, *.zip or *.gz file with single compressed file. | None – must be specified. |
| `Output` | Indicate the output to be generated:<br>• `TimeSeries` – individual time series (even if in ensemble)<br>• `TimeSeriesAndEnsembles` – individual time series and ensemble | `TimeSeries AndEnsembles` |
| `TimeZone Offset` | The desired time zone offset for output. `0`=GMT, `7`=US Mountain Standard Time. | Use file time zone. |
| `TimeZone` | Time zone string to assign to output date/times, for example `MST`. No check is performed on the validity of the value. The time zone should agree with the result of applying `TimeZoneOffset`. | No time zone is assigned. |
| `InputStart` | Starting date/time to read data, in precision consistent with data. Specify as a date/time string or a processor `${Property}`. | Read all data. |
| `InputEnd` | Ending date/time to read data, in precision consistent with data. Specify as a date/time string or a processor `${Property}`. | Read all data. |
| `DataSource` | Data source to use for time series identifier, for example organization that is running FEWS. Can specify with `${ts:Property}` and time series `%` specifiers. | `FEWS` |
| `DataType` | Data type to use for time series identifier, useful because default can be long and may contain special characters. Can specify with `${ts:Property}` and time series `%` specifiers. | `<paramerId>` element from PI XML file |
| `Description` | Time series description. Can specify with `${ts:Property}` and time series `%` specifiers. | `<stationName>` element from PI XML file |
| `Read24Hour AsDay` | If `True`, read 24Hour interval time series as Day. Hour 00 values are shifted to the previous day. | `False` |
| `Read24Hour AsDayCutoff` | If the value of the 24Hour time series is <= this value, then the day will be decremented in the output time series. This is necessary because hourly data may not exactly line up with days (hour 0) and it may be appropriate to shift to the previous day depending on the data type. | `0` |
| `Alias` | The alias to assign to the time series, as a literal string or using the special formatting characters listed by the | No alias is assigned. |

| Parameter | Description | Default |
|---|---|---|
| | command editor. The alias is a short identifier used by other commands to locate time series for processing, as an alternative to the time series identifier (TSID). | |
| `EnsembleID` | Ensemble identifier to assign to output. Can specify with `${ts:Property}` and time series `%` specifiers. | `<locationId>_DataType_<ensembleId>` |
| `EnsembleName` | Ensemble name to assign to output. Can specify with `${ts:Property}` and time series `%` specifiers. | Value of `EnsembleID` |

This page is intentionally blank.