

---

# Appendix: DIADvisor Input Type – DRAFT

2004-05-27, Acrobat Distiller

## Overview

This document is in draft form – DIADvisor support is being evaluated.

DIADvisor is a product developed by Diad, Incorporated (now OneRain). DIADvisor is a data collection product for real-time environmental monitoring. The DIADvisor database consists of an operational database (e.g., named *Swatch2.mdb* when used with Microsoft Access), and an archive database (e.g., named *archiveH.mdb* or *archive.mdb* when used with Microsoft Access). The operational database contains tables to define sites, sensors, time series, and alarms, and time series data are typically saved for one month. The archive database primarily contains tables to store time series data that roll off of the operational tables.

The following summarizes the important tables and fields as they apply to time series:

*SiteDef* (operational database only)

<i>Site Name</i>	Descriptive name for a site (might be more descriptive than the <i>SensorDef.Sensor</i> name, especially with respect to location).
<i>Site ID</i>	Numerical, unique, site identifier.
<i>Other fields</i>	Not described here because they are used by DIADvisor but not necessarily important for time series.

*SensorDef* (operational database only)

With respect to time series, the *SensorDef* table can be thought of as an index for available time series.

<i>Sensor ID</i>	Unique numerical key for the sensor.
<i>Site ID</i>	Foreign key to site identifier. Multiple sensors can be installed at a site.
<i>Type</i>	Indicates the type of sensor (e.g., Analog, Rain, Battery). Although this can indicate the data type, <b>Group</b> is better suited to indicate a data type (e.g., Analog is not as clear as Stage or Temperature, and can be ambiguous).
<i>Group</i>	Indicates the data type group (e.g., Rain, Stage, Temperature). Although there are preferred values, groups can vary from system to system.
<i>Description</i>	Descriptive name for a sensor - for time series, the description should ideally include both the location and data type (e.g., XYZ River below TownA Streamflow).
<i>Rating Type</i>	If None, only one data value is available at this sensor ( <i>DataChron.Data Value</i> ). Otherwise, this field indicates how the second value ( <i>DataChron.Data Value 2</i> ) is to be determined from the first – Equation or Table.
<i>Display Units</i>	Units for <i>DataChron.Data Value</i> .
<i>Display Units 2</i>	Units for <i>DataChron.Data Value 2</i> .
<i>Decimal</i>	Number of digits of precision after the decimal point for <i>DataChron.Data Value</i> displays.

<i>Decimal 2</i>	Number of digits of precision after the decimal point for <i>DataChron.Data Value 2</i> displays.
<i>Most Recent Time</i>	Date and time of the most recent value in <i>DataChron</i> .
<i>Other fields.</i>	Not described here because they are used by DIADvisor but not necessarily important for time series.

*DataChron* (operational and archive databases)

The *DataChron* table holds the raw data for measurements and can be thought of as containing irregular interval time series.

<i>Date/Time</i>	Date and time that a sensor measurement was recorded, to the second.
<i>Sensor ID</i>	Foreign key to <i>SensorDef.Sensor ID</i> .
<i>Count</i>	Number of units measured for the sensor.
<i>Data Type</i>	Flag that indicates whether data are good, suspect, etc (need more information from DIAD).
<i>Source</i>	How values are determined (need more information from DIAD).
<i>Data Value</i>	Data value corresponding to the sensor's measuring device (e.g., stage).
<i>Data Value 2</i>	Secondary data value: <ul style="list-style-type: none"> <li>• Computed/engineering value determined using the method in <i>SensorDef.Rating Type</i>. If the rating type is not NONE, then <i>Data Value 2</i> will be computed by DIADvisor. For example, for streamflow, stage is recorded in <i>Data Value</i> and streamflow is recorded in <i>Data Value 2</i> using a rating table or equation (it would be helpful to have a full list of <i>Data Value</i> and corresponding <i>Data Value 2</i> that commonly occur, for documentation).</li> <li>• If the <i>SensorDef.Type</i> for the sensor has a value of RAIN, then <i>Data Value 2</i> contains the cumulative rain (for what period? until the next tip? system interval?).</li> </ul>
<i>Other fields</i>	Not described here because they are used by DIADvisor but not necessarily important for time series.

*IntervalTS, DayTS, HourTS* (operational and archive databases)

These tables contain regular interval data computed from the *DataChron* data by DIADvisor. It appears that these tables only contain values when *SensorDef.Group* is Rain (need DIAD to confirm).

<i>Sensor ID</i>	Foreign key to <i>Sensor Def.Sensor ID</i> .
<i>Start Time</i>	Date and time for the start of the interval at which a data value applies.
<i>Value</i>	Data value for the interval, processed from the <i>DataChron.Data Value</i> values.

## DIADvisor and Standard Time Series Properties

DIADvisor provides information to populate most standard time series properties, as follows:

- The location part of the time series identifier is the *SensorDef.Sensor ID* (need information from DIAD – is this what people are accustomed to using?).
- The data source part of the time series identifier is left blank, although it may be possible to pass on the original data source imported into DIADvisor.
- The data type part of the time series identifier is the *SensorDef.Group* plus a hyphen, plus *DataValue* or *DataValue2* (no spaces).
- The data interval part of the time series identifier is *Irregular* for *DataChron* time series, *SysConfig.Interval* + *Min* (e.g., *5Minute*) for *IntervalTS* time series, *Hour* for *HourTS* time series, and *Day* for *DayTS* time series.
- The scenario part of the time series identifier is *Archive* for an archive database time series and is not used for an operational database time series. Therefore, it is possible to read the same time series from both databases for comparison.
- The input type part of the time series identifier is *DIADvisor*.
- The input file part of the time series identifier is the name assigned to the database connection (for future development).
- The description for the time series is taken from *SensorDef.Description*.
- The data units are taken from *SensorDef.Display Units* for *Data Value* and *SensorDef.Display Units 2* for *Data Value 2*.
- The time series period is defined based on the query period defined by the application (e.g., *TSTool*) or is determined after querying all available data. There does not appear to be a way to quickly retrieve the start time for the data that are in the database.

## Limitations

DIADvisor has the following limitations:

- There is no explicit data type defined for time series in *DataChron*, *IntervalTS*, *HourTS*, or *DayTS*. Based on available data in *SensorDef*, one or two *DataChron* time series can be identified using *SensorID..Group-DataValue.Irregular* and *SensorID..Group-DataValue2.Irregular*. Because *Interval*, *Hour*, and *Day* time series only store the *Data Value* values, the identifiers for these time series use the pattern: *SensorID..Group-DataValue.Day*, etc.
- The *Interval*, *Hour*, and *Day* time series are only available for the Rain Group (need confirmation from DIAD – is there any way to tell in the *GroupDef* table which groups have Data Value and Data Value 2?).
- The available end of period for each *DataChron* time series can be taken from *SensorDef.Most Recent Time*. However, the start of period is not saved. The start and end are not saved for *Interval*, *Hour*, *Day* time series. Therefore, it is not possible to show the available data period (e.g., in TSTool) without querying all the time series data.
- Although the *SensorDef* table does define time series that will have values in *DataChron* (and the archive tables), there is no guarantee that the time series tables will actually have values. It is possible that a list of time series generated from *SensorDef* will result in errors when time series when an attempt is made to retrieve time series data. A check for the *SensorDef.Most Recent Time* can be made and if a non-null value is returned it can be assumed that data are available. However, if data were only collected in the past, no current data may be available (only archive database data may be available).
- In order to read data from the archive database, an operational database also needs to be opened to retrieve information about sensors. The same time series will exist in both databases, with a shorter period in the operational database. Although this is not necessarily a limitation, care must be taken to identify the connections, if, for example, the same time series is compared from the operational and archive database.
- Some table names and fields contain spaces. Although not truly a limitation, these spaces can lead to problems in SQL because [brackets] or other symbols must be used. To avoid problems in time series identifiers, spaces are removed for *Data Value* and *Data Value 2* when used in the data type. Currently the group is allowed to contain spaces but this may be changed in the future.
- The date/time for an interval in DIADvisor is the start of the interval. Although not necessarily a limitation, this convention is contrary to some other data collection and management systems – care must be taken to adjust the date/time as necessary if transferring to other systems.
- Rating tables and equations are not defined with an applicable date. Instead, only the most recent rating table is saved in the operational database. Therefore, it is impossible to reconstruct a true historical record of *Data Value 2* from raw *Data Value* data.
- Sensor out of service is only recorded for the current time. Therefore, it is impossible to construct a true historical record of missing data for *Data Value* based on sensor out of service time. (DIAD – there seem to be some “sensors” that track service?).