

# ReadStoredData

## Declaration

```
unsigned int _stdcall ReadStoredData(StationPar *St, D2Time FromTime,
D2Time ToTime, int TransId, D2Time *NextFromTime)
```

## Parameters

St	Pointer to <a href="#">StationPar</a> structure.
FromTime	Beginning of time interval that contains the required data.
ToTime	End of time interval that contains the required data.
TransId	Transaction identifier that is defined as a parameter of the function <a href="#">PointOldValue</a> .
NextFromTime	Using the parameter allows the user to split the reading historical values into several periods (their number is optional). A nonzero time value of the parameter means that the reading was finished for the values with the timestamp and older. The system will generate the next calling the function <b>ReadStoredData</b> where the value of the parameter <i>FromTime</i> is equal to the value of the parameter <i>NextFromTime</i> . Zero value of the parameter <i>NextFromTime</i> means that calling the function <b>ReadStoredData</b> is finished. The method allows to divide the reading of historical values, which takes a long time, into several intervals and to use the time between individual intervals for reading the current values.

## Description

The function **ReadStoredData** (`_ReadStoredData@28` in the *Stdcall* call convention) is optional. It is called to get the historical values that are archived in connected device (if the device allows it). The required values are within an interval defined by the parameters *FromTime* and *ToTime*. As the gaining historical values from a device could take a long time and block a transmission of current data (DLL protocol functions are called by one task, i.e. sequentially), it is possible to split the gaining of the values into several phases. Writing a new interval into the parameter *NextFromTime* will cause the generating of the next calling the function `ReadStoredData` with defined parameter *FromTime*. The sequence of callings is finished by writing 0.0 into the parameter *NextFromTime*.

Ignoring the function is performed by writing 0.0 into the parameter *NextFromTime* and the function return code `RQ_IGNORED`. The values, gained by this calling, must be sent by using the call-back function [PointOldValue](#).

## Return value

Begin time of the next interval of the reading historical values which is saved into the parameter *NextFromTime*. See also return codes of the function [ReadAllPoints](#).



### Related pages:

[D2000 KomAPI - interface functions](#)