
**User's
Manual**

**Model 707712
WVF File Access API**

Product Registration

Thank you for purchasing YOKOGAWA products.

YOKOGAWA provides registered users with a variety of information and services.

Please allow us to serve you best by completing the product registration form accessible from our homepage.

<http://tmi.yokogawa.com/>

Foreword

Thank you for purchasing the WVF File Access API (Model 707712).

This User's Manual contains information about the installation, programming model, and functions of the WE Control API. To ensure correct use, please read this manual thoroughly before operation.

Notes

- **The contents of this manual describe the WVF File Access API Ver. 1.0.1. If you are using another version of the API, the information given in this manual may differ from that of API that you are using.**
- The contents of this manual are subject to change without prior notice as a result of continuing improvements to the instrument's performance and functions.
- Every effort has been made in the preparation of this manual to ensure the accuracy of its contents. However, should you have any questions or find any errors, please contact your nearest YOKOGAWA dealer.
- Copying or reproducing any or all of the contents of this manual without YOKOGAWA's permission is strictly prohibited.

Trademarks

- Microsoft, Windows, and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
- Adobe and Acrobat are trademarks of Adobe Systems Incorporated.
- All other company and product names used in this manual are trademarks or registered trademarks of their respective companies.

Revisions

1st Edition: July 2003

Terms and Conditions of the Software License

NOTICE - PLEASE READ CAREFULLY BEFORE USE

Thank you very much for purchasing this medium containing a software program and related documentation provided by Yokogawa Electric Corporation (hereinafter called "Yokogawa"), and the program contained, embedded, inserted or used in the medium (hereinafter called the "Yokogawa Software Program").

By opening this package or plastic wrapping (hereinafter called "Package") enclosing the Yokogawa Software Program, you acknowledge that you understand and agree to the "Terms and Conditions of the Software License" (hereinafter called "Terms and Conditions") which is written in the documentation and separately attached. Accordingly, the Terms and Conditions bind you.

The Yokogawa Software Program and its related documentation including ownership of copyright shall remain the exclusive property of Yokogawa or those third parties from whom sublicensed software in the Yokogawa Software Program is licensed.

Yokogawa hereby grants you permission to use the Yokogawa Software Program on the conditions that you agree to the Terms and Conditions before you open the Package and/or install it in or onto a computer.

IF YOU DO NOT AGREE TO THE TERMS AND CONDITIONS, YOU CANNOT OPEN THE PACKAGE, AND MUST IMMEDIATELY RETURN IT TO YOKOGAWA OR ITS DESIGNATED PARTY.

Terms and Conditions of the Software License

Yokogawa Electric Corporation, a Japanese corporation (hereinafter called "Yokogawa"), grants permission to use this Yokogawa Software Program (hereinafter called the "Licensed Software") to the Licensee on the conditions that the Licensee agrees to the terms and conditions stipulated in Article 1 hereof.

You, as the Licensee (hereinafter called "Licensee"), shall agree to the following terms and conditions for the software license (hereinafter called the "Agreement") based on the use intended for the Licensed Software.

Please note that Yokogawa grants the Licensee permission to use the Licensed Software under the terms and conditions herein and in no event shall Yokogawa intend to sell or transfer the Licensed Software to the Licensee.

Licensed Software Name: Model 707712 WVF File Access API

Number of License: 1

Article 1 (Scope Covered by these Terms and Conditions)

- 1.1 The terms and conditions stipulated herein shall be applied to any Licensee who purchases the Licensed Software on the condition that the Licensee consents to agree to the terms and conditions stipulated herein.
- 1.2 The "Licensed Software" herein shall mean and include all applicable programs and documentation, without limitation, all proprietary technology, algorithms, and know-how such as a factor, invariant or process contained therein.

Article 2 (Grant of License)

- 2.1 Yokogawa grants the Licensee, for the purpose of single use, non-exclusive and non-transferable license of the Licensed Software with the license fee separately agreed upon by both parties.
- 2.2 The Licensee is, unless otherwise agreed in writing by Yokogawa, not entitled to copy, change, sell, distribute, transfer, or sublicense the Licensed Software.
- 2.3 The Licensed Software shall not be copied in whole or in part except for keeping one (1) copy for back-up purposes. The Licensee shall secure or supervise the copy of the Licensed Software by the Licensee itself with great, strict, and due care.
- 2.4 In no event shall the Licensee dump, reverse assemble, reverse compile, or reverse engineer the Licensed Software so that the Licensee may translate the Licensed Software into other programs or change it into a man-readable form from the source code of the Licensed Software. Unless otherwise separately agreed by Yokogawa, Yokogawa shall not provide the Licensee the source code for the Licensed Software.
- 2.5 The Licensed Software and its related documentation shall be the proprietary property or trade secret of Yokogawa or a third party which grants Yokogawa the rights. In no event shall the Licensee be transferred, leased, sublicensed, or assigned any rights relating to the Licensed Software.
- 2.6 Yokogawa may use or add copy protection in or onto the Licensed Software. In no event shall the Licensee remove or attempt to remove such copy protection.
- 2.7 The Licensed Software may include a software program licensed for re-use by a third party (hereinafter called "Third Party Software", which may include any software program from affiliates of Yokogawa made or coded by themselves.) In the case that Yokogawa is granted permission to sublicense to third parties by any licensors (sub-licensor) of the Third Party Software pursuant to different terms and conditions than those stipulated in this Agreement, the Licensee shall observe such terms and conditions of which Yokogawa notifies the Licensee in writing separately.
- 2.8 In no event shall the Licensee modify, remove or delete a copyright notice of Yokogawa and its licensor contained in the Licensed Software, including any copy thereof.

Article 3 (Restriction of Specific Use)

- 3.1 The Licensed Software shall not be intended specifically to be designed, developed, constructed, manufactured, distributed or maintained for the purpose of the following events:
 - a) Operation of any aviation, vessel, or support of those operations from the ground;
 - b) Operation of nuclear products and/or facilities;
 - c) Operation of nuclear weapons and/or chemical weapons and/or biological weapons; or
 - d) Operation of medical instrumentation directly utilized for humankind or the human body.
- 3.2 Even if the Licensee uses the Licensed Software for the purposes in the preceding Paragraph 3.1, Yokogawa has no liability to or responsibility for any demand or damage arising out of the use or operations of the Licensed Software, and the Licensee agrees, on its own responsibility, to solve and settle the claims and damages and to defend, indemnify or hold Yokogawa totally harmless, from or against any liabilities, losses, damages and expenses (including fees for recalling the Products and reasonable attorney's fees and court costs), or claims arising out of and related to the above-said claims and damages.

Article 4 (Warranty)

- 4.1 The Licensee shall agree that the Licensed Software shall be provided to the Licensee on an "as is" basis when delivered. If defect(s), such as damage to the medium of the Licensed Software, attributable to Yokogawa is found, Yokogawa agrees to replace, free of charge, any Licensed Software on condition that the defective Licensed Software shall be returned to Yokogawa's specified authorized service facility within seven (7) days after opening the Package at the Licensee's expense. As the Licensed Software is provided to the Licensee on an "as is" basis when delivered, in no event shall Yokogawa warrant that any information on or in the Licensed Software, including without limitation, data on computer programs and program listings, be completely accurate, correct, reliable, or the most updated.
- 4.2 Notwithstanding the preceding Paragraph 4.1, when third party software is included in the Licensed Software, the warranty period and terms and conditions that apply shall be those established by the provider of the third party software.
- 4.3 When Yokogawa decides in its own judgement that it is necessary, Yokogawa may from time to time provide the Licensee with Revision upgrades and Version upgrades separately specified by Yokogawa (hereinafter called "Updates").

- 4.4 Notwithstanding the preceding Paragraph 4.3, in no event shall Yokogawa provide Updates where the Licensee or any third party conducted renovation or improvement of the Licensed Software.
- 4.5 THE FOREGOING WARRANTIES ARE EXCLUSIVE AND IN LIEU OF ALL OTHER WARRANTIES OF QUALITY AND PERFORMANCE, WRITTEN, ORAL, OR IMPLIED, AND ALL OTHER WARRANTIES INCLUDING ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE ARE HEREBY DISCLAIMED BY YOKOGAWA AND ALL THIRD PARTIES LICENSING THIRD PARTY SOFTWARE TO YOKOGAWA.
- 4.6 Correction of nonconformity in the manner and for the period of time provided above shall be the Licensee's sole and exclusive remedy for any failure of Yokogawa to comply with its obligations and shall constitute fulfillment of all liabilities of Yokogawa and any third party licensing the Third Party Software to Yokogawa (including any liability for direct, indirect, special, incidental or consequential damages) whether in warranty, contract, tort (including negligence but excluding willful conduct or gross negligence by Yokogawa) or otherwise with respect to or arising out of the use of the Licensed Software.

Article 5 (Infringement)

- 5.1 If and when any third party should demand injunction, initiate a law suit, or demand compensation for damages against the Licensee under patent right (including utility model right, design patent, and trade mark), copy right, and any other rights relating to any of the Licensed Software, the Licensee shall notify Yokogawa in writing to that effect without delay.
- 5.2 In the case of the preceding Paragraph 5.1, the Licensee shall assign to Yokogawa all of the rights to defend the Licensee and to negotiate with the claiming party. Furthermore, the Licensee shall provide Yokogawa with necessary information or any other assistance for Yokogawa's defense and negotiation. If and when such a claim should be attributable to Yokogawa, subject to the written notice to Yokogawa stated in the preceding Paragraph 5.1, Yokogawa shall defend the Licensee and negotiate with the claiming party at Yokogawa's cost and expense and be responsible for the final settlement or judgment granted to the claiming party in the preceding Paragraph 5.1.
- 5.3 When any assertion or allegation of the infringement of the third party's rights defined in Paragraph 5.1 is made, or when at Yokogawa's judgment there is possibility of such assertion or allegation, Yokogawa will, at its own discretion, take any of the following countermeasures at Yokogawa's cost and expense.
- a) To acquire the necessary right from a third party which has lawful ownership of the right so that the Licensee will be able to continue to use the Licensed Software;
 - b) To replace the Licensed Software with an alternative one which avoids the infringement; or
 - c) To remodel the Licensed Software so that the Licensed Software can avoid the infringement of such third party's right.
- 5.4 If and when Yokogawa fails to take either of the countermeasures as set forth in the preceding subparagraphs of Paragraph 5.3, Yokogawa shall indemnify the Licensee only by paying back the price amount of the Licensed Software which Yokogawa has received from the Licensee. THE FOREGOING PARAGRAPHS STATE THE ENTIRE LIABILITY OF YOKOGAWA AND ANY THIRD PARTY LICENSING THIRD PARTY SOFTWARE TO YOKOGAWA WITH RESPECT TO INFRINGEMENT OF THE INTELLECTUAL PROPERTY RIGHTS INCLUDING BUT NOT LIMITED TO, PATENT AND COPYRIGHT.

Article 6 (Liabilities)

- 6.1 If and when the Licensee should incur any damage relating to or arising out of the Licensed Software or service that Yokogawa has provided to the Licensee under the conditions herein due to a reason attributable to Yokogawa, Yokogawa shall take actions in accordance with this Agreement. However, in no event shall Yokogawa be liable or responsible for any special, incidental, consequential and/or indirect damage, whether in contract, warranty, tort, negligence, strict liability, or otherwise, including, without limitation, loss of operational profit or revenue, loss of use of the Licensed Software, or any associated products or equipment, cost of capital, loss or cost of interruption of the Licensee's business, substitute equipment, facilities or services, downtime costs, delays, and loss of business information, or claims of customers of Licensee or other third parties for such or other damages. Even if Yokogawa is liable or responsible for the damages attributable to Yokogawa and to the extent of this Article 6, Yokogawa's liability for the Licensee's damage shall not exceed the price amount of the Licensed Software or service fee which Yokogawa has received. Please note that Yokogawa shall be released or discharged from part or all of the liability under this Agreement if the Licensee modifies, remodels, combines with other software or products, or causes any deviation from the basic specifications or functional specifications, without Yokogawa's prior written consent.
- 6.2 All causes of action against Yokogawa arising out of or relating to this Agreement or the performance or breach hereof shall expire unless Yokogawa is notified of the claim within one (1) year of its occurrence.
- 6.3 In no event, regardless of cause, shall Yokogawa assume responsibility for or be liable for penalties or penalty clauses in any contracts between the Licensee and its customers.

Article 7 (Limit of Export)

Unless otherwise agreed by Yokogawa, the Licensee shall not directly or indirectly export or transfer the Licensed Software to any countries other than those where Yokogawa permits export in advance.

Article 8 (Term)

This Agreement shall become effective on the date when the Licensee receives the Licensed Software and continues in effect unless or until terminated as provided herein, or the Licensee ceases using the Licensed Software by itself or with Yokogawa's thirty (30) days prior written notice to the Licensee.

Article 9 (Injunction for Use)

During the term of this Agreement, Yokogawa may, at its own discretion, demand injunction against the Licensee in case that Yokogawa deems that the Licensed Software is used improperly or under severer environments other than those where Yokogawa has first approved, or any other condition which Yokogawa may not permit.

Article 10 (Termination)

Yokogawa, at its sole discretion, may terminate this Agreement without any notice or reminder to the Licensee if the Licensee violates or fails to perform this Agreement. However, Articles 5, 6, and 11 shall survive even after the termination.

Article 11 (Jurisdiction)

Any dispute, controversies, or differences between the parties hereto as to interpretation or execution of this Agreement shall be resolved amicably through negotiation between the parties upon the basis of mutual trust. Should the parties fail to agree within ninety (90) days after notice from one of the parties to the other, both parties hereby irrevocably submit to the exclusive jurisdiction of the Tokyo District Court (main office) in Japan for settlement of the dispute.

Article 12 (Governing Law)

This Agreement shall be governed by and construed in accordance with the laws of Japan. The Licensee expressly agrees to waive absolutely and irrevocably and to the fullest extent permissible under applicable law any rights against the laws of Japan which it may have pursuant to the Licensee's local law.

Article 13 (Severability)

In the event that any provision hereof is declared or found to be illegal by any court or tribunal of competent jurisdiction, such provision shall be null and void with respect to the jurisdiction of that court or tribunal and all the remaining provisions hereof shall remain in full force and effect.

Contents

Foreword	1
Terms and Conditions of the Software License	2
1. Overview	1-1
2. Installation	2-1
3. File Operation Functions	3-1
3.1 Basic Model	3-1
3.2 File Format	3-1
3.3 Model When Performing Write Operation of Measured Data	3-2
3.4 Model When Performing Read Operation of Measured Data	3-4
3.5 Common Information Structure and Channel Information Structure	3-8
3.6 The List of Functions	3-10
3.7 Single File Access	3-11
WeDPHeaderReadS	3-11
WeDPDataRead	3-11
WeDPHeaderWriteS	3-12
WeDPDataWrite	3-13
3.8 Sequential File Access	3-15
WeDPHeaderCsReadS	3-15
WeDPCsRead	3-15
WeDPHeaderCsWriteS	3-16
WeDPCsWrite	3-17
3.9 Access the Specified Item of the Header File	3-19
WeDPHeaderItemRead	3-19
WeDPHeaderItemWrite	3-19
3.10 Data Operation	3-21
WeDPGetSampleChNum	3-21
WeDPGetBlockNum	3-21
WeDPInitializeAcqInfo	3-22
3.11 The List of C Language Interface	3-24
4. Error Cord	4-1
Index	Index-1

1

2

3

4

Index

1. Overview

This user's manual describes the interface functions (WVF File Access API) used to access the waveform data files (WVF files).

The WVF File Access API is a dynamic link library (DLL) for Microsoft Windows. The DLL can be used in Win32 program development platforms such as Microsoft Visual C++ and Microsoft Visual Basic.

A List of Files Included

File Name	Description
WeAscii.bas	File containing definitions of constants for Visual Basic
WeDP.bas	File access function API definition file for Visual Basic
*.dll	DLL files for the WVF File Access API
WeAscii.h	File containing definitions of constants for Visual C++
WeDP.h	File access function API definition file for Visual C++
WeDP.pdf	WVF File Access Online Manual (Adobe Acrobat Reader 3.0 or later required for opening the file.)
WeFileAccess	DLL file for the .Net development platform

The definition files above contain definitions of symbols, data structures, and other elements that are used by the API. Be sure to include them in the user application programs.

By default, the files listed above excluding the DLL files are copied to the C:\Program Files\WE7000\API directory. The DLL files are copied to directories indicated below in addition to the directory indicated above.

On Windows 95/98/Me: Windows\System

On Windows NT 4.0, 2000 Pro, XP Professional/Home Edition: Windows\System32

Supported OSs

Microsoft Windows 95/98/Me, Windows NT 4.0, Windows 2000 Pro or Windows XP Professional/Home Edition

Supported Development Platforms

Microsoft Visual Basic 5.0, 6.0, and .Net

Microsoft Visual C++6.0 and .Net

Microsoft Visual C#.Net

Applicable Models

WE7000

DL708, DL708E, DL716, and DL750

DL1720 and DL1740

DL1600 Series

DL1500 Series

DL7100 and DL7200

DL7400 Series

Note

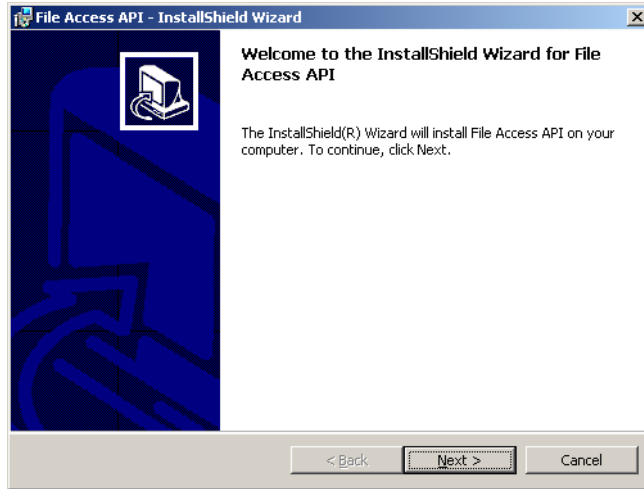
Class names and function names of the .Net-compatible class library

- The namespace is WeFileAccess. The class name is WeFile.
- The names of the .Net functions are obtained by removing "WeDP" from the "WeDPxxx" function names.

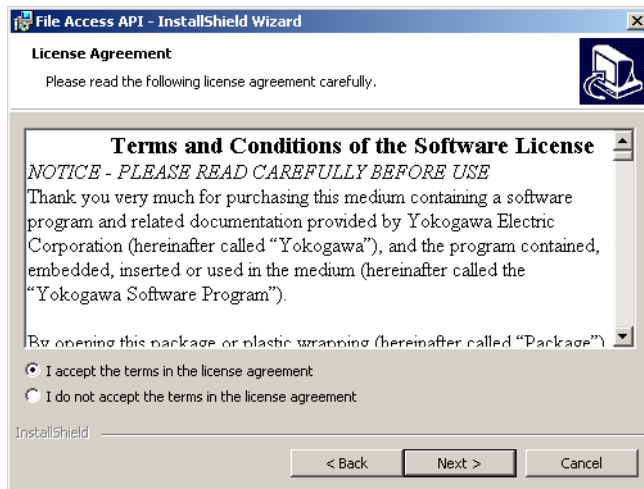
2. Installation

This chapter describes the installation procedure for the “WVF File Access API.”

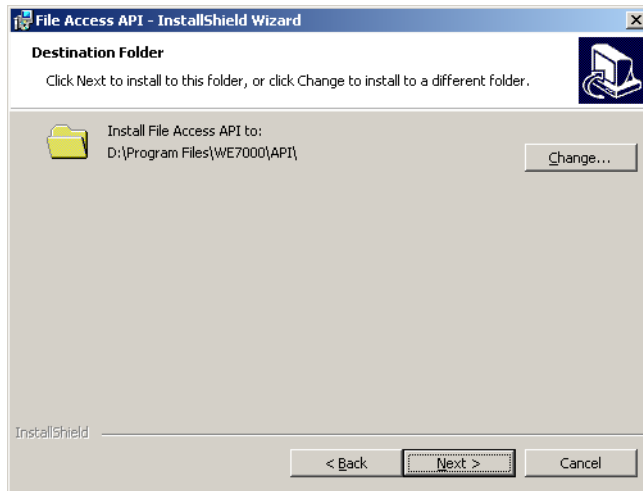
1. Insert the WVF File Access API Setup Disk into your PC's CD-ROM drive. An installer automatically starts and the following dialog box opens. Click **Next**.



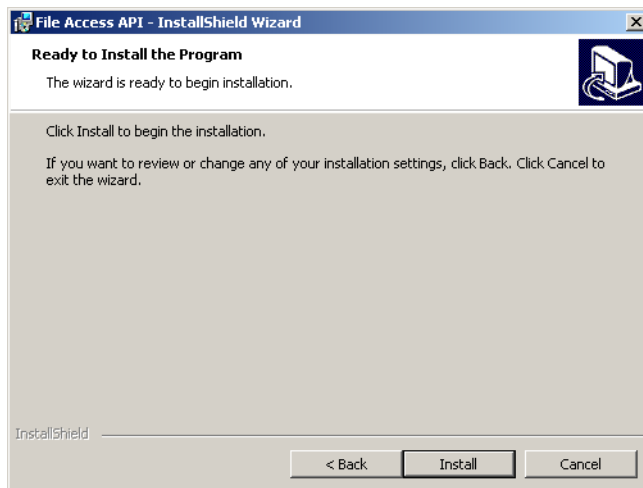
2. The following dialog box opens containing license agreement information. Confirm the license agreement, click the I accept the terms in the license agreement option button, and click **Next**.



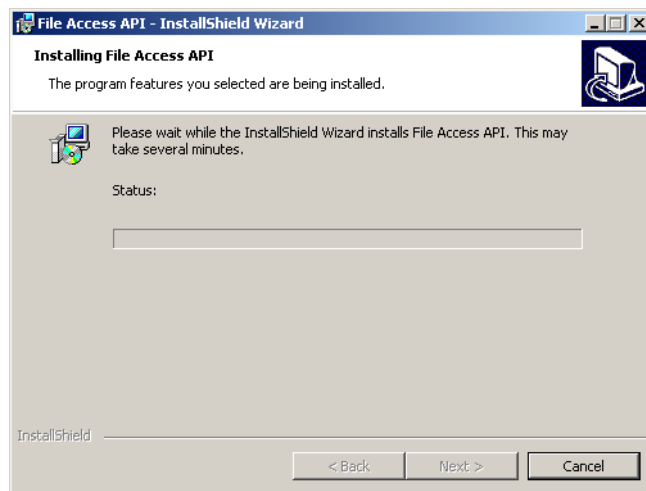
3. Specify the installation destination. The default installation destination is set to “C:\Program Files\WE7000\API\.” If this is OK, click **Next**. To change the installation directory, click **Change**, select the directory, and click **Next**.



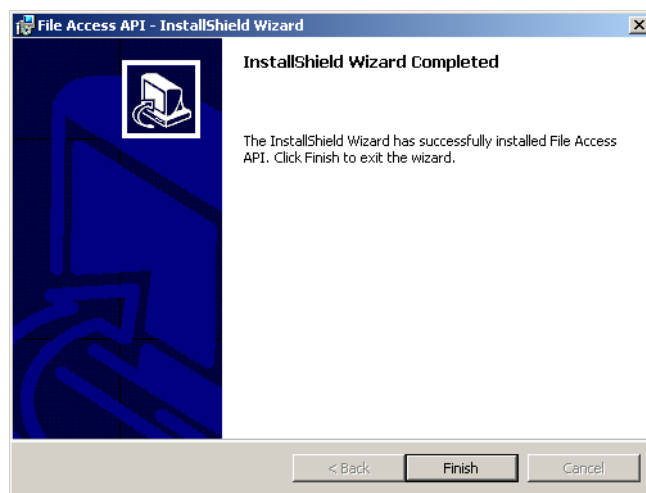
4. A dialog box appears confirming that you wish to begin installation. Click **Install**.



- The installation starts and the following dialog box opens indicating the progress of the installation.



- A dialog box opens notifying you that the installation has been completed. Click **Finish**.



3. File Operation Functions

3.1 Basic Model

For the write operation of measured data, functions are provided for the following two assumed models: one in which data blocks are added in order to a single file (Single File Model) and another in which a file is created for each data block (Sequential File Model).

For the read operation of measured data, functions are provided for the following two assumed models: one in which data is read by specifying a block from a file in which data is stored in multiple blocks (Single File Model) and another in which data is read by specifying the number of samples from multiple files (Sequential File Model).

3.2 File Format

The files are in YOKOGAWA's proprietary format (the same format as the measured data that is saved in binary format using the WE7000 Control Software (.wvf file)).

There are three types of storage formats of .wvf files: block type, trace type, and scan type. This API can be used to read the measured data without being aware of the storage format of .wvf files.

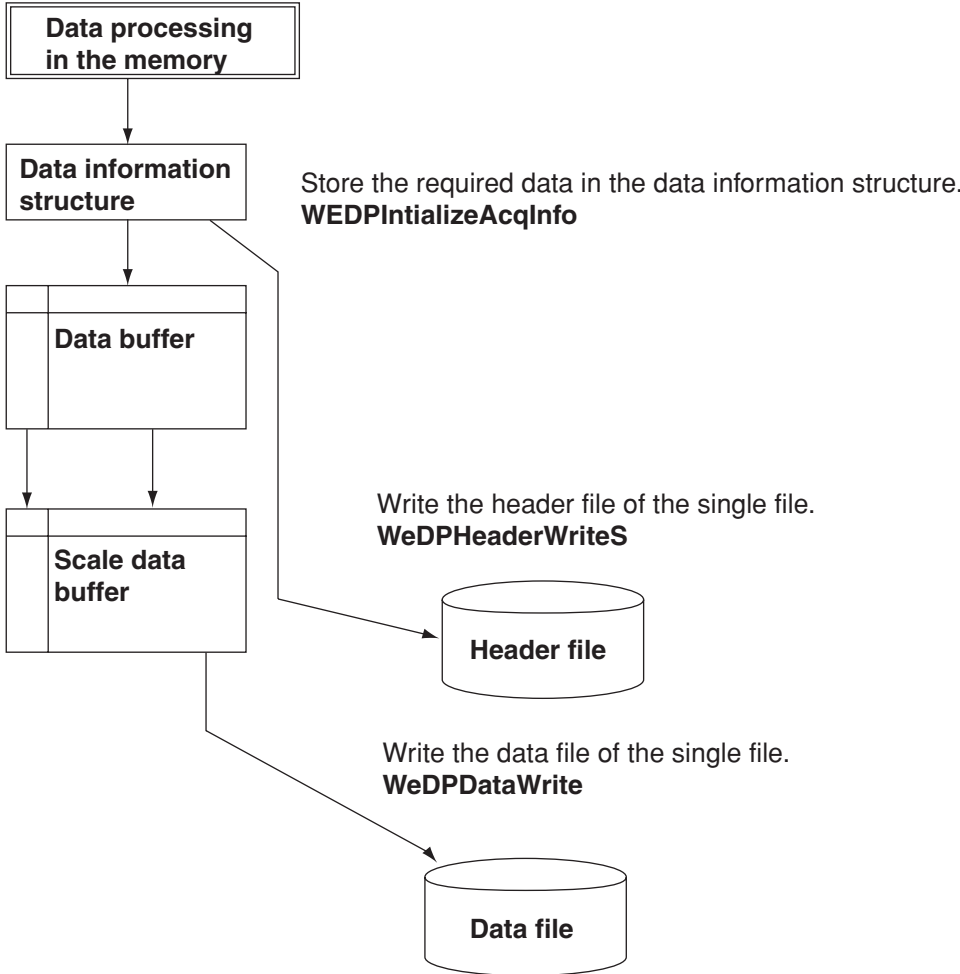
However, when reading a Scan type file, the entire data is assumed to be a single block. This API stores the data using the block type of the .wvf format.

Note

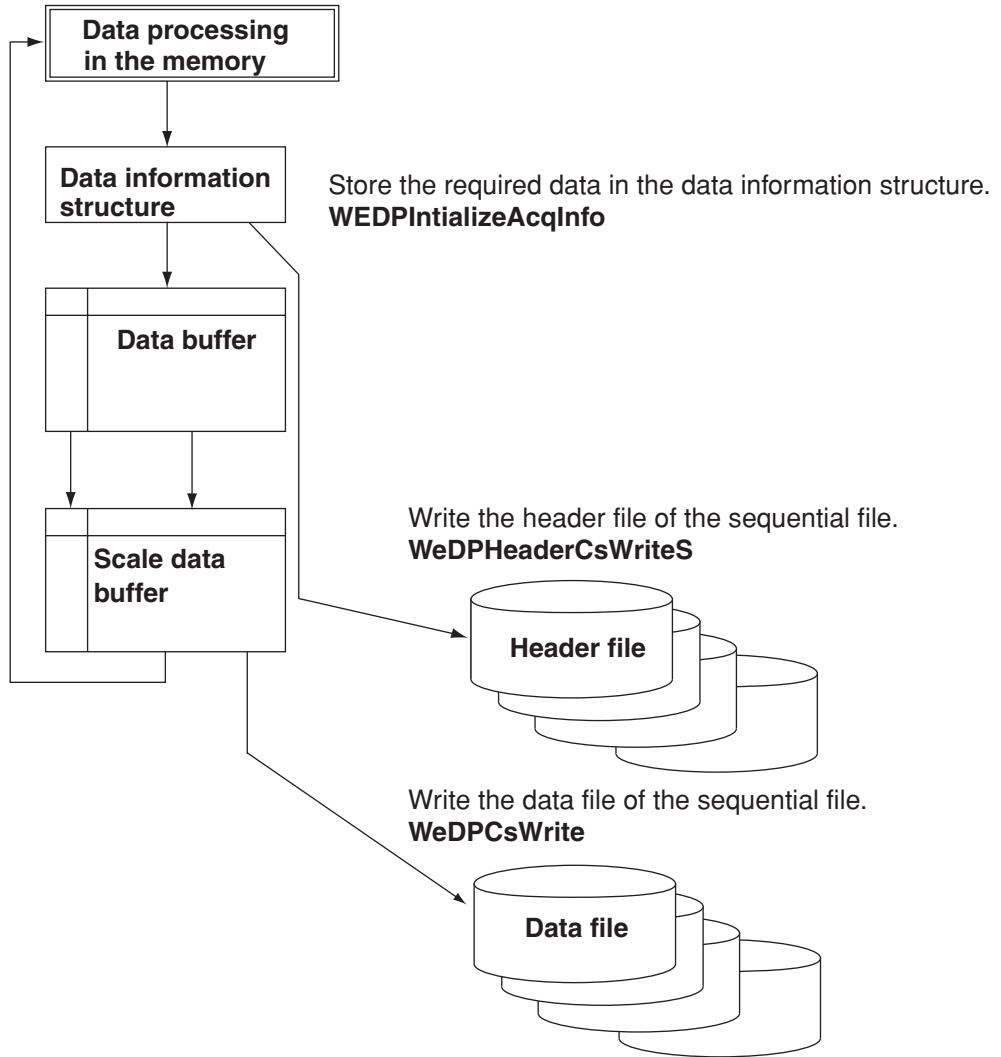
For details on .wvf files, see technical information (TI7000-21E) issued by YOKOGAWA.

3.3 Model When Performing Write Operation of Measured Data

Single File Model for Data Created Using an Application Program (Multiple blocks are stored to the same file)



Creating Sequential Files (Store data that has been created using a program to a single file for each block)



3.4 Model When Performing Read Operation of Measured Data

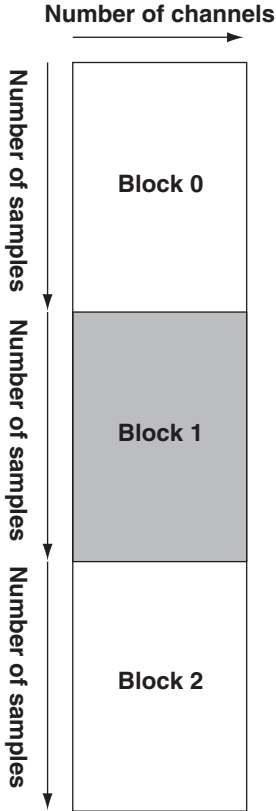
Single File Model (Read data by specifying blocks)



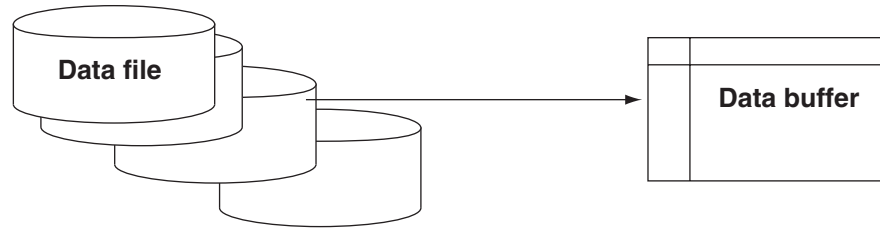
WeDPDataRead(FileName,BlockNo,ChNo,DataForm,DataBuf)

- FileName: File name (excluding the sequential number section).
- BlockNo: Block number (0 to n).
- ChNo: Channel number. Specify -1 to read all channels.
- DataForm: Data type.
- DataBuf: Pointer to the buffer in which the data is to be stored.

If you specify WeDPDataRead("FileName",1,-1,DATA_SINGLE,DataBuf), the data in the shaded section in the following figure is stored to DataBuf.



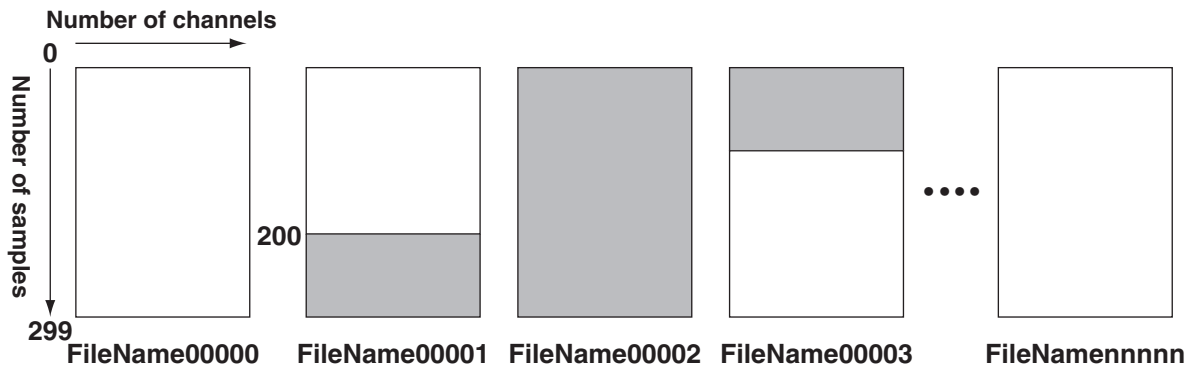
Sequential File Model (Read data by specifying the number of samples)



WeDpCsRead(FileName, SeriesNo, Start, Length, ChNo, DataForm, DataBuff)

- FileName: File name (excluding the sequential number section).
- SeriesNo: First sequence number.
- Start: Start sample number.
- Length: Number of data points per channel to be read.
- ChNo: Channel number. Specify -1 to read all channels.
- DataForm: DataBuff type.
- DataBuff: Pointer to the buffer in which the data is to be stored.

If you specify WeDpCsRead("FileName", 1, 200, 500, -1, DATA_SINGLE, DataBuff), the data in the shaded section in the following figure is stored to DataBuff.



This file has "4 channels × 300 samples" of data stored.
 Data is stored to the memory after scaling with VResolution and VOffset in the .hdr file.

Search Method of Sequential Files

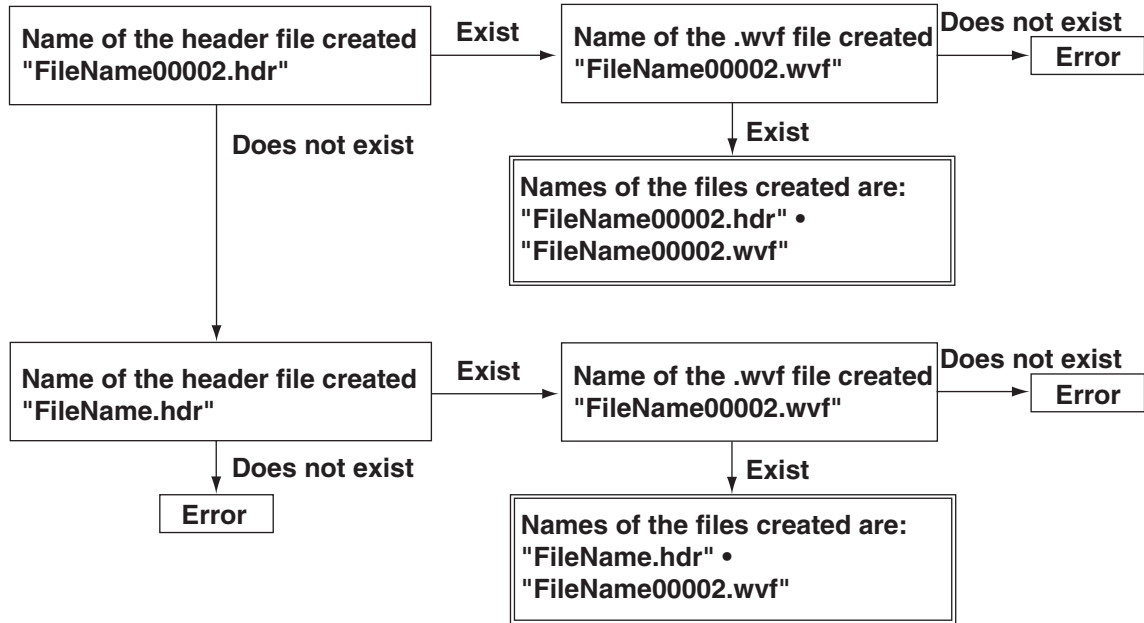
- When the sequence number is set to -1

`WeDPCsRead("FileName",-1,200,500,-1,WE_FLOAT,DataBuff)`

Names of the files created are:
"FileName.hdr" • "FileName.wvf"

- When the sequence number is set to a positive integer

`WeDPCsRead("FileName",2,200,500,-1,WE_FLOAT,DataBuff)`



Reading Header Files

- Name of the header file for single files

WeDPHeaderReadS(FileName,BlockNo,ComBuff(),ChBuff())

WeDPHeaderReadS("FileName", 2,ComBuff(),ChBuff())

FileName: "FileName"

BlockNo: 2

ComBuff(): ComBuff()

ChBuff(): When ChBuff()

Set to FileName.hdr.

- Name of the header file for sequential files

WeDPHeaderCsReadS(FileName,SeriesNo,ComBuff(),ChBuff())

WeDPHeaderCsReadS("FileName",2,ComBuff(),ChBuff())

FileName: "FileName"

SeriesNo: 2

ComBuff(): ComBuff()

ChBuff(): When ChBuff()

Set to FileName0002.hdr.

WeDPHeaderCsReadS("FileName",-1,ComBuff(),ChBuff())

FileName: "FileName"

SeriesNo: -1

ComBuff(): ComBuff()

ChBuff(): When ChBuff()

Set to FileName.hdr.

3.5 Common Information Structure and Channel Information Structure

User-Defined Type

When the X-axis data is the same on all channels, the data structure is as follows:

- Common information

```
Type CommonInf1
Comment As String*256           // Comment (set up to 255 bytes)
SampleNum As Long               // Number of samples
ChNum As Long                   // Number of channels
SamplingInterval As Double      // Sampling interval
PreTrigger As Long              // Amount of pretrigger
XUnit As String*16              // X-axis unit
Date As String*16               // Measurement date (set up to 15 bytes)
Time As String*16               // Measurement time (set up to 15 bytes)
End Type
```
- Channel information

```
Type Chanellnf1
ChName AS String*16             // Channel name (set up to 15 bytes)
ScaleA As Double                // Scaling coefficient A
ScaleB As Double                // Scaling coefficient B
Unit As String*16               // Unit (set up to 15 bytes)
End Type
```

Note

The current version does not allow handling of files in which the X-axis data is not the same for all channels.

Correspondence between User-Defined Type and Header File Items

User-Defined Type	Header File Item
Comment	Character string in the comment line
SampleNum	Sum of BlockSize for the number of blocks (information of the first channel)
ChNum	TraceTotalNumber value
SamplingInterval	HResolution value (information of the first channel)
PreTrigger	HOffset value (information of the first channel)
XUnit	HUnit value (information of the first channel)
Date	Date value (information of the first channel)
Time	Time value (information of the first channel)
ChName	TraceName value (for each channel)
ScaleA	Linear VResolution value (for each channel)
ScaleB	Linear VOffset value (for each channel)
Unit	Linear Unit value (for each channel)

Correspondence between Acquisition Data Information and Header File Items

The correspondence between acquisition items (acquisition data information that can be read using WeGetAcqDataInfo (page 6-66)) and the header file items is as follows:

Acquisition Data Information Item	Header File Item
Comment	Character string in the comment line
Channel	–
Data Type	VdataType
BlockNum	BlockNumber
StartBit	–
EffectiveBit	–
TrigActive	–
Record	BlockSize
Recordlen	–
TrigPosition	HOffset
Interval	HResolution
VResolution	VResolution
VOffset	VOffset
TrigLevel	–
TrigWidth	–
PlusOverData	VPlusOverData
MinusOverData	VMinusOverData
NonData	VIllegalData
DispMaxData	VMaxData
DispMinData	VMinData

The settings of the items of the header file that is created when writing measured data using this API are indicated in the table below. For a description of the header file of other applicable models, see the user's manual that came with the product.

Item Name	Setting
FormatVersion	1.01
Model	WE7000
Endian	Ltl
DataFormat	Block
GroupNumber	Automatically created to match the stored data when writing measured data
TraceTotalNumber	Automatically created to match the stored data when writing measured data
DataOffset	0
TraceNumber	Automatically created to match the stored data when writing measured data
BlockNumber	Automatically created to match the stored data when writing measured data
TraceName	Set to Ch1 to ChN to match the stored data when writing measured data
BlockSize	Record of acquisition data information
VResolution	VResolution of acquisition data information
VOffset	VOffset of acquisition data information
VdataType	Set from parameters when writing measured data
VUnit	–
VplusOverData	PlusOverData of acquisition data information
VminusOverData	MinusOverData of acquisition data information
VillegalData	NonData of acquisition data information
VMaxData	DispMaxData of acquisition data information
VMinData	DispMinData of acquisition data information
HResolution	SamplingInterval of CommonInf1
HOffset	PreTrigger of CommonInf1
HUnit	XUnit of CommonInf1
Date	Date of CommonInf1
Time	Time of CommonInf1
PLinearSlope	ScaleA of CommonInf1
PLinearIntercept	ScaleB of CommonInf1
PLinearUnit	Unit of CommonInf1
PTraceName	ChannelName of CommonInf1
PLinearMode	“OFF”

3.6 The List of Functions

Single File Access

Function Name	Description	Page
WeDPHeaderReadS	Read the header file of the single file.	3-11
WeDPDataRead	Read the data file of the single file.	3-11
WeDPHeaderWriteS	Write the header file of the single file.	3-12
WeDPDataWrite	Write the data file of the single file.	3-13

Sequential File Access

Function Name	Description	Page
WeDPHeaderCsReadS	Read the header file of the sequential file.	3-15
WeDPCsRead	Read the data file of the sequential file.	3-15
WeDPHeaderCsWriteS	Write the header file of the sequential file.	3-16
WeDPCsWrite	Write the data file of the sequential file.	3-17

Access the Specified Item of the Header File

Function Name	Description	Page
WeDPHeaderItemRead	Read the data of the specified item.	3-19
WeDPHeaderItemWrite	Write the data of the specified item.	3-19

Data Operation

Function Name	Description	Page
WeDPGetSampleChNum	Get the number of samples and number of channels.	3-21
WeDPGetBlockNum	Get the number of blocks.	3-21
WeDPInitializeAcqInfo	Store the required data in the data information structure.	3-22

3.7 Single File Access

WeDPHeaderReadS

Description

Reads the data from the header file by specifying the block number.

Syntax

```
WeDPHeaderReadS(ByVal FileName As String,ByVal BlockNo As Long,ByRef ComBuff() As
CommonInf1,ByRef ChBuff() As ChanelInf1)
```

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

<i>FileName</i> (IN)	Header file name.
<i>BlockNo</i> (IN)	Block number. Specify the block number you wish to read.
<i>ComBuff</i> () (OUT)	Common information structure. Define using an array.
<i>ChBuff</i> () (OUT)	Channel information structure. Define using an array.

Note:

Collectively retrieves the header file information for data of which the number of X-axis data points is the same. Prepare a single-element array for the common information structure buffer. Prepare channel information structure buffer for the amount equal to the number of channels.

Example (Visual Basic)

```
Dim ret As Long
Dim FileName As String
Dim BlockNo As Long
Dim ComBuff(0) As CommonInf1
Dim ChBuff(5) As ChanelInf1
FileName = "TestData1"
BlockNo = 2
ret = WeDPHeaderReadS(FileName,BlockNo,ComBuff(),ChBuff())
If 0 <> ret Then
    MsgBox "File read error"
End If
```

Note

This function collectively retrieves a section of the data in the header file as indicated in section 3.5, "Common Information Structure and Channel Information Structure." To read unsupported items, use WeDPHeaderItemRead (page 3-19).

WeDPDataRead

Description

Reads the data from the data file by specifying the block number.

Syntax

```
WeDPDataRead(ByVal FileName As String,ByVal BlockNo As Long,ByVal ChNo As Long,ByVal
DataForm As Long,ByRef DataBuff As Any)
```

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

- FileName* (IN) Data file name. Specify the file name excluding the extension.
- BlockNo* (IN) Block number.
- ChNo* (IN) Channel number. -1 represents all channels. Counted from 1.
If you specify -1, pass a buffer defined using a two-dimensional array for DataBuff.
- DataForm* (IN) DataBuff type. Specify BYTE (char), Integer (short), Long (long), Single (float), or Double (double). No other data types can be passed. The type inside the parentheses indicates the type for the C Language.
- DataBuff* (OUT) Data storage buffer. Pass the buffer in the type specified by DataForm.

Note:

Reads the data from the data file in units of blocks. The files cannot be handled as sequential files. Specify DataForm from the following defined data.

BYTE: WE_UBYTE
Integer: WE_SWORD
Long: WE_SLONG
Single: WE_FLOAT
Double: WE_DOUBLE

Example (Visual Basic)

```
Dim FileName As String
Dim BlockNo As Long
Dim ChNo As Long
Dim DataForm As Long
Dim DataBuff(999,3) As Single
FileName = "TestData1"
BlockNo = 2
ChNo = -1
DataForm = WE_FLOAT
ret = WeDPDataRead(FileName,BlockNo,ChNo,DataForm,DataBuff(0,0))
If 0 <> ret Then
    MsgBox "File read error"
End If
```

WeDPHeaderWriteS

Description

Writes the header information at once to the header file by specifying the block.

Syntax

WeDPHeaderWriteS(ByVal *FileName* As String,ByVal *BlockNo* As Long,ByRef *ComBuff()* As CommonInf1,ByRef *ChBuff()* As ChanellInf1,ByRef *AcqInfo* As AcqDataInfEx)

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

- FileName* (IN) Header file name.
- BlockNo* (IN) Block number. Specify the block number to be written.
- ComBuff()* (IN) Common information structure. Define using an array.
- ChBuff()* (IN) Channel information structure. Define using an array.
- AcqInfo* (IN) Data information structure (AcqDataInfoEx).

Note:

Collectively writes the header file information for data of which the number of X-axis data points is the same. Prepare a single-element array for the common information structure buffer. Prepare channel information structure buffer for the amount equal to the number of channels. AcqInfo passes the data retrieved using WeAcqInfoInitialize(). Or, create the data by referring to section 3.5, “Common Information Structure and Channel Information Structure.” Data is written to the specified block.

Example (Visual Basic)

```
Dim ret As Long
Dim FileName As String
Dim BlockNo As Long
Dim CommonBuff(0) As CommonInfl
Dim ChBuff(5) As ChanelInfl
Dim AcqInfo(5) As AcqDataInfoEx
Dim InfoNum As Long
BlockNo=2
Ret = WeGetAcqDataInfoEx(hMo,-1,BlockNo,AcqInfo(),infoNum)
If 0 <> ret Then
    MsgBox "Data information read error"
EndIf
FileName = "TestData1"
ret=WeDPHeaderWriteS(FileName,BlockNo,ComBuff(),ChBuff(),AcqInfo())
If 0 <> ret Then
    MsgBox "File write error"
End If
```

WeDPDataWrite**Description**

Writes the data to the data file in units of blocks.

Syntax

WeDPDataWrite(ByVal *FileName* As String,ByVal *BlockNo* As Long,ByVal *SampleNum* As Long,ByRef *AcqInfo* As AcqDataInfoEx,ByVal *DataForm* As Long,ByRef *DataBuff* As Any)

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

<i>FileName</i> (IN)	Data file name. Specify the file name excluding the extension. A .wvf extension is added to the created file.
<i>BlockNo</i> (IN)	Block number.
<i>SampleNum</i> (IN)	Number of samples.
<i>AcqInfo</i> (IN)	Data information structure. AcqInfo passes the data retrieved using WeAcqInfoInitialize(). Or, create the data by referring to section 3.5, “Common Information Structure and Channel Information Structure.”
<i>DataForm</i> (IN)	DataBuff type. Specify BYTE (char), Integer (short), Long (long), Single (float), or Double (double). No other data types can be passed. The type inside the parentheses indicates the type for the C Language.
<i>DataBuff</i> (OUT)	Data storage buffer. Pass the buffer in the type specified by DataForm. Pass a buffer defined using a two-dimensional array.

Note:

Write data to a data file. The files cannot be handled as sequential files. Specify DataForm from the following defined data.

BYTE: WE_UBYTE
Integer: WE_SWORD
Long: WE_SLONG
Single: WE_FLOAT
Double: WE_DOUBLE

Example (Visual Basic)

```
Dim FileName As String
Dim BlockNo As Long
Dim SampleNum As Long
Dim AcqInfo(1) As AcqDataInfoEx
Dim DataForm As Long
Dim DataBuff(999,1) As Single
Dim InfoNum As Integer
BlockNo = 2
SampleNum = 1000
WeGetAcqDataInfoEx(hMo,-1,BlockNo,AcqInfo(),InfoNum)
FileName = "TestData1"
DataForm = WE_FLOAT
ret = WeDPDataWrite(FileName,BlockNo,SampleNum,AcqInfo(),DataForm,
DataBuff(0,0))
If 0 <> ret Then
    MsgBox "File write error"
End If
```

3.8 Sequential File Access

WeDPHeaderCsReadS

Description

Collectively reads the header information from a header file.

Syntax

```
WeDPHeaderCsReadS(ByVal FileName As String,ByVal SeriesNo As Long, ByRef ComBuff() As
CommonInf1,ByRef ChBuff() As ChanelInf1)
```

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

<i>FileName</i> (IN)	Header file name.
<i>SeriesNo</i> (IN)	Sequence number when creating sequence files. If you specify -1, <i>FileName</i> becomes the file name as-is.
<i>ComBuff()</i> (OUT)	Common information structure. Define using an array.
<i>ChBuff()</i> (OUT)	Channel information structure. Define using an array.

Note:

Collectively retrieves the header file information for data of which the number of X-axis data points is the same. Prepare a single-element array for the common information structure buffer. Prepare channel information structure buffer for the amount equal to the number of channels.

Example (Visual Basic)

```
Dim ret As Long
Dim FileName As String
Dim SeriesNo As Long
Dim ComBuff(0) As CommonInf1
Dim ChBuff(5) As ChanelInf1
FileName = "TestData1"
BlockNo = 2
ret = WeDPHeaderCsReadS(FileName,SeriesNo,ComBuff(),ChBuff())
If 0 <> ret Then
    MsgBox "File read error"
End If
```

Note

This function collectively retrieves a section of the data in the header file as indicated in section 3.5, "Common Information Structure and Channel Information Structure." To read unsupported items, use `WeDPHeaderItemRead` (page 3-19).

WeDPCsRead

Description

Reads the data from the data files (sequential files) by specifying the number of samples.

Syntax

```
WeDPCsRead(ByVal FileName As String,ByVal SeriesNo As Long,ByVal Start As Long,ByVal Length
As Long,ByVal ChNo As Long,ByVal DataForm As Long,ByRef DataBuff As Any)
```

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

- FileName* (IN) Data file name. Specify the file name excluding the extension.
- SeriesNo* (IN) Sequence number when reading sequential files. If you specify -1, only the file with the name specified by *FileName* is read.
- Start* (IN) Start sample number. Specify the first sample number to be retrieved using a value greater than or equal to 0.
- Length* (IN) Number of samples to be retrieved. -1 specifies all data after the sample specified by *Start*. An error occurs if the specified value is greater than the number of samples that is stored.
- ChNo* (IN) Channel number. -1 specifies all channels. Counted from 1.
- DataForm* (IN) DataBuff type. Specify BYTE (char), Integer (short), Long (long), Single (float), or Double (double). No other data types can be passed.
The type inside the parentheses indicates the type for the C Language.
- DataBuff* (OUT) Buffer for storing data. Pass the buffer in the type specified by *DataForm*.

Note:

Reads the data from the data file by specifying the number of samples.

Retrieves data of *Length* from sample *Start* in the file specified by *FileName* and *SeriesNo*.

Specify *DataForm* from the following defined data.

BYTE: WE_UBYTE
Integer: WE_SWORD
Long: WE_SLONG
Single: WE_FLOAT
Double: WE_DOUBLE

Example (Visual Basic)

```
Dim FileName As String
Dim SeriesNo As Long
Dim Start As Long
Dim Length As Long
Dim ChNo As Long
Dim DataForm As Long
Dim DataBuff(1193,3) As Single
FileName = "TestData1"
ret = WeDPCsRead(FileName,2,100,1200,-1,WE_FLOAT,DataBuff(0,0))
If 0 <> ret Then
    MsgBox "File read error"
End If
```

WeDPHeaderCsWriteS

Description

Collectively writes the header information to the header file.

Syntax

WeDPHeaderCsWriteS(ByVal *FileName* As String,ByVal *SeriesNo* As Long,ByRef *ComBuff*() As CommInf1,ByRef *ChBuff*() As ChanellInf1,ByRef *AcqInfo* As AcqDataInfoEx)

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

<i>FileName</i> (IN)	Header file name.
<i>SeriesNo</i> (IN)	Sequence number when creating sequence files. If you specify -1, <i>FileName</i> becomes the file name as-is.
<i>CommBuff()</i> (IN)	Common information structure. Define using an array.
<i>ChBuff()</i> (IN)	Channel information structure. Define using an array.
<i>AcqInfo</i> (IN)	Data information structure.

Note:

Collectively writes the header file information for data of which the number of X-axis data points is the same. Prepare a single-element array for the common information structure buffer.

Prepare channel information structure buffer for the amount equal to the number of channels.

AcqInfo passes the data retrieved using *WeAcqInfoInitialize()*. Or, create the data by referring to section 3.5, "Common Information Structure and Channel Information Structure."

If the file already exists, it is overwritten.

Example (Visual Basic)

```
Dim ret As Long
Dim FileName As String
Dim SeriesNo As Long
Dim CommonBuff(0) As CommonInfl
Dim ChBuff(5) As ChanelInfl
Dim AcqInfo(5) As AcqDataInfoEx
Dim InfoNum As Long
Ret = WeGetAcqDataInfoEx(hMo,-1,5,AcqInfo(),infoNum)
If 0 <> ret Then
    MsgBox "Data information read error"
End If
FileName = "TestData1"
SeriesNo = 2
ret = WeDPHeaderWriteS(FileName,SeriesNo,CommonBuff(),ChBuff(),
AcqInfo())
If 0 <> ret Then
    MsgBox "File write error"
End If
```

WeDPCsWrite

Description

Write data to a sequence file.

Syntax

WeDPCsWrite(ByVal *FileName* As String,ByVal *SeriesNo* As Long,ByVal *SampleNum* As Long,ByRef *AcqInfo* As Acq DataInfoEx,ByVal *DataForm* As Long,ByRef *DataBuff()* As Any)

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

<i>FileName</i> (IN)	Data file name. Specify the file name excluding the extension. A .wvf extension is added to the created file.
<i>SeriesNo</i> (IN)	Sequence number when creating sequence files. If you specify -1, <i>FileName</i> becomes the file name as-is.
<i>SampleNum</i> (IN)	Number of samples.

<i>AcqInfo</i> (IN)	Data information structure. Or, create the data by referring to section 3.5, "Common Information Structure and Channel Information Structure."
<i>DataForm</i> (IN)	DataBuff type. Specify BYTE (char), Integer (short), Long (long), Single (float), or Double (double). No other data types can be passed. The type inside the parentheses indicates the type for the C Language.
<i>DataBuff</i> (OUT)	Data storage buffer. Pass the buffer in the type specified by DataForm. Pass a buffer defined using a two-dimensional array.

Note:

Write data to a data file.

The files are handled as sequential files. SeriesNo is automatically added to the file names.

Specify DataForm from the following defined data.

BYTE: WE_UBYTE
Integer: WE_SWORD
Long: WE_SLONG
Single: WE_FLOAT
Double: WE_DOUBLE

Example (Visual Basic)

```
Dim FileName As String
Dim SeriesNo As Long
Dim AcqInfo(2) As AcqDataInfoEx
Dim DataForm As Long
Dim DataBuff(2,1000) As Single
Dim InfoNum As Integer
WeGetAcqDataInfoEx(hMo,-1,2,AcqInfo(),InfoNum)
FileName = "TestData1"
SeriesNo = 2
DataForm = WE_FLOAT
ret = WeDPCsWrite(FileName,SeriesNo,AcqInfo(),DataForm,
DataBuff(0,0))
If 0 <> ret Then
    MsgBox "File write error"
End If
```

3.9 Access the Specified Item of the Header File

WeDPHeaderItemRead

Description

Reads the information of the specified item name and specified channel from the header information of the header file.

Syntax

```
WeDPHeaderItemRead(ByVal FileName As String,ByVal ItemName As String,ByVal ChNo As Long,ByVal BlockNo As Long,ByRef DataBuff As String)
```

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

FileName (IN) Data file name. Specify the file name excluding the extension.
ItemName (IN) Item name.
ChNo (IN) Channel number. Ignored if an item unrelated to the channel number is specified.
BlockNo (IN) Block number.
DataBuff (OUT) Buffer for storing data.

Note:

Retrieves the information of the specified item and specified channel from the header file information. You must have an understanding of the structure of the .hdr file when using this function. A block number is added to the item number for data containing multiple blocks. To read such item, specify the item name with the block number. Items "GroupNumber" and "TraceNumber" cannot be retrieved.

Example (Visual Basic)

```
Dim FileName As String
Dim ItemName As String
Dim ChNo As Long
Dim DataBuff As String*256
Dim BlockNo As Long
FileName = "TestData"
ItemName = "VResolution"
ChNo = 1
BlockNo = 0
Ret = WeDPHeaderItemRead(FileName,ItemName,ChNo,BlockNo,DataBuff)
If 0 <> ret Then
    MsgBox "Error in reading a header file item"
End If
```

WeDPHeaderItemWrite

Description

Writes data to the specified item name and specified channel in the header information of the header file.

Syntax

```
WeDPHeaderItemWrite(ByVal FileName As String,ByVal ItemName As String,ByVal ChNo As Long,ByVal BlockNo As Long,ByRef DataBuff As String)
```

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

FileName (IN) Data file name. Specify the file name excluding the extension.

ItemName (IN) Item name.

ChNo (IN) Channel number.

BlockNo (IN) Block number.

DataBuff (OUT) Buffer for storing data.

Note:

Writes data to the specified item and specified channel in the header information file.

Items "FormatVersion," "Model," "Endian," "DataFormat," "GroupNumber," "TraceTotalNumber," "TraceName," "BlockNumber," and "VDataType" cannot be specified.

When you set PLinearMode, set to "0" as OFF or "1" as ON.

Example (Visual Basic)

```
Dim FileName As String
Dim ItemName As String
Dim ChNo As Long
Dim BlockNo As Long
Dim DataBuff As String
FileName = "TestData"
ItemName = "VResolution"
ChNo = 1
BlockNo = 0
DataBuff = CStr(5.42)
Ret = WeDPHeaderItemWrite(FileName,ItemName,ChNo,BlockNo,DataBuff)
If 0 <> ret Then
    MsgBox "Error in reading a header file item"
End If
```

3.10 Data Operation

WeDPGetSampleChNum

Description

Gets the number of samples and number of channels of the specified file.

Syntax

```
WeDPGetSampleChNum(ByVal FileName As String,ByVal BlockNo As Long,ByRef SampleNum As Long,ByRef ChNum As Long)
```

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

<i>FileName</i> (IN)	Data file name. Specify the file name excluding the extension.
<i>BlockNo</i> (IN)	Block number. -1 specifies all blocks.
<i>SampleNum</i> (OUT)	Number of samples.
<i>ChNum</i> (OUT)	Number of channels.

Note:

The number of samples and number of channels of the specified file are returned. For Scan type files, the total number of samples is returned regardless of the BlockNo setting.

Example (Visual Basic)

```
Dim FileName As String
Dim BlockNo As Long
Dim SampleNum As Long
Dim ChNum As Long
FileName = "TestData"
BlockNo = 0
ret = WeDPGetSampleChNum(FileName, SampleNum, ChNum)
If 0 <> ret Then
    MsgBox "Error in reading the number of samples and number of channels"
End If
```

WeDPGetBlockNum

Description

Gets the number of blocks of the specified file.

Syntax

```
WeDPGetBlockNum(ByVal FileName As String,ByRef BlockNum As Long)
```

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

<i>FileName</i> (IN)	Data file name. Specify the file name excluding the extension.
<i>BlockNum</i> (OUT)	Number of blocks.

Note:

The number of blocks of the specified file is returned.

Example (Visual Basic)

```
Dim FileName As String
Dim BlockNum As Long
FileName = "TestData"
ret = WeDPGetBlockNum(FileName,BlockNum)
If 0 <> ret Then
    MsgBox "Error in reading the number of blocks"
End If
```

WeDPInitializeAcqInfo**Description**

Stores the required data in the data information structure.

Syntax

WeDPInitializeAcqInfo(ByVal *VMaxData* As Double,ByVal *VMinData* As Double,ByVal *SampleNum* As Long,ByVal *SamplInterval* As Double,ByRef *AcqInfo()* As AcqDataInfoEx)

Return value

Returns 0 if successful. Returns an error code if unsuccessful.

Parameters

<i>VMaxData</i> (IN)	Scale Max.
<i>VMinData</i> (IN)	Scale Min.
<i>SampleNum</i> (IN)	Number of samples.
<i>SamplInterval</i> (IN)	Sampling interval (s).
<i>AcqInfo()</i> (OUT)	Data information structure.

Note:

Sets and returns the required data in the data information structure.

The data set in AcqDataInfo by this function is as follows:

Chanel	1 through n is set in the order of the array.
dataType	Data buffer type is set using the parameter of WeDPDataWrite or WeDPCsWrite.
blockNum	1
startBit	0
effectiveBit	0
trigActive	0
record	SampleNum.
recordLen	SampleNum.
time	0
trigPosition	0.0
Interval	Set to the SamplingInterval parameter.
VResolution	Set to 1.0.
VOffset	Set to 0.0.
TrigLevel	Set to 0.0.
TrigWidth	Set to 0.0.
PlusOverData	Set to the VMaxData parameter.
MinusOverData	Set to the VMinData parameter.
NonData	Set to the lost data.
DispMaxData	Set to the VMaxData parameter.
DispMinData	Set to the VMinData parameter.

Example (Visual Basic)

```
Dim VMaxData As Double
Dim VMinData As Double
Dim AcqInfo(3) As AcqDataInfoEx
VMaxData = 2.0
VMinData = -2.0
ret = WeDPInitializeAcqInfo(VMaxData,VMinData,1000,0.001,AcqInfo())
If 0 <> ret Then
    MsgBox "Error in setting data to data information structure"
End If
```

3.11 The List of C Language Interface

Single File Access

Function Name	Syntax
WeDPHeaderReadS	int WeDPHeaderReadS(char* lpName, UINT blockno, LPSAFEARRAY* ComInfo, LPSAFEARRAY* ChBuff)
WeDPDataRead	int WeDPDataRead(char* lpName, int chno, UINT blockno, UINT dataForm, LPVOID dataBuff)
WeDPHeaderWriteS	int WeDPHeaderWriteS(char* lpName, UINT blockno, LPSAFEARRAY* ComBuff, LPSAFEARRAY* ChBuff, LPSAFEARRAY* AcqInfo)
WeDPDataWrite	int WeDPDataWrite(char* lpName, UINT chNum, UINT blockno, UINT SampleNum, UINT dataForm, LPVOID buf, LPSAFEARRAY* AcqData), LPVOID DataBuff)

Sequential File Access

Function Name	Syntax
WeDPHeaderCsReadS	int WeDPHeaderCsReadS(char* lpName, int seriesNo, LPSAFEARRAY* ComBuff, LPSAFEARRAY* ChBuff)
WeDPCsRead	int WeDPCsRead(char* lpName, int seriesNo, UINT start, int length, int chno, UINT dataForm, LPVOID dataBuff)
WeDPHeaderCsWriteS	int WeDPHeaderCsWriteS(char* lpName, int seriesNo, LPSAFEARRAY* ComBuff, LPSAFEARRAY* ChBuff, LPSAFEARRAY* AcqInfo)
WeDPCsWrite	int WeDPCsWrite(char* lpName, int seriesNo, LPSAFEARRAY* AcqData, UINT dataForm, LPVOID dataBuff)

Access the Specified Item of the Header File

Function Name	Syntax
WeDPHeaderItemRead	int WeDPHeaderItemRead(char* lpName, char* itemName, UINT ChNo, UINT blockno, char* data)
WeDPHeaderItemWrite	int WeDPHeaderItemWrite(char* lpName, char* itemName, UINT ChNo, UINT blockno, char* data)

Data Operation

Function Name	Syntax
WeDPGetSampleChNum	int WeDPGetSampleChNum(char* lpName, UINT* blockno, UINT* SampleNum, UINT* ChNum)
WeDPGetBlockNum	int WeDPGetBlockNum(char* lpName, UINT* blockNo)
WeDPInitializeAcqInfo	int WeDPInitializeAcqInfo(double VMaxData, double VMinData, double SamplingInterval, LPSAFEARRAY* AcqInfo)

4. Error Cord

Symbol	Error Code (Hex)	Description
AsciiRSLT_NORMAL	0x0000	Normal completion.
AsciiRSLT_CANT_OPEN	0x0001	File open failure.
AsciiRSLT_CANT_ALLOC	0x0002	Memory allocation failure.
AsciiRSLT_FORMAT_ERROR	0x0003	The number of fields of the record is not correct.
AsciiRSLT_HANDLE_NULL	0x0004	NULL handle.
AsciiRSLT_NOT_INTEGER	0x0005	The data type is not UINT. GroupNumber, TraceTotalNumber, DataOffset, TraceNumber, BlockNumber, and BlockSize are UINT.
AsciiRSLT_POSITION_ERROR	0x0006	Description position is not correct. For example, a Group item is in PublicInfo.
AsciiRSLT_SCRIPT_ERROR	0x0007	Error in the item descriptor. For example, BlockSizeA.
AsciiRSLT_STRING_ERROR	0x0008	Error in Endian or DataFormat. Endian: Little or Big DataFormat: Trace or Block
AsciiRSLT_CANT_READ	0x0009	Failed to read. For example, "?". Vresolution, Voffset, VUnit, VPlusOverData, VminusOverData, VillegalData, VMaxData, VMinData, Hresolution, HUnit, Date, or Time
AsciiRSLT_BOUNDARY	0x000A	Range designation error.
AsciiRSLT_READ_ERROR	0x000B	Access error to WVF file.
AsciiRSLT_UNSUPPORTED_DATATYPE	0x000C	Error in DataType. Only IU1, IS1, IU2, IS2, IU4, IS4, FS4, and FS8 are supported.
AsciiRSLT_LINK_ERROR	0x000D	DLL link failure.
AsciiRSLT_UNSUPPORTED_FUNCTION	0x000E	Function not supported by the specified DLL.
AsciiRSLT_ZERO_BLOCKSIZE	0x000F	The block size is 0.
AsciiRSLT_UNSUPPORTED_SCALING	0x0010	Scaling designation error. Other than AsciiADWave, AsciiPhysicalWave, or AsciiScalingWave.
AsciiRSLT_DISK_FULL	0x0011	Insufficient free disk space.
AsciiRSLT_MODIFY_ERROR	0x0012	Error in the parameter.
AsciiRSLT_ERROR	0x00FF	Other error.

Index

T

Terms and Conditions of the Software License 2

W

WeDPCsRead 3-15
WeDPCsWrite 3-17
WeDPDataRead 3-11
WeDPDataWrite 3-13
WeDPGetBlockNum 3-21
WeDPGetSampleChNum 3-21
WeDPHeaderCsReadS 3-15
WeDPHeaderCsWriteS 3-16
WeDPHeaderItemRead 3-19
WeDPHeaderItemWrite 3-19
WeDPHeaderReadS 3-11
WeDPHeaderWriteS 3-12
WeDPInitializeAcqInfo 3-22