

DNW / ONERA-GMT

Technical Report

Wind tunnel Data Format for test results presentation

ATA-TR-001-2002

DNW / ONERA - GMT

Technical Report ATA-TR-001-2002

Wind tunnel Data Format for test results presentation

Abstract

The development of new measuring techniques requires new data formats and data structures. Due to the rigidity of the twenty years old formats and structures adaptation to the new requirements are impossible. Moreover the availability of modern software tools gives a lot more possibilities.

As a preliminary joint exercise to the forthcoming Aero Testing Alliance (ATA), it was decided by DNW and ONERA-GMT to develop a common wind tunnel data format WDF, available in all the facilities of ATA.

The definition of a common wind tunnel data format (WDF) for classical data and a coherent file structure for all concerned wind tunnels, creates the opportunity for DNW and ONERA to provide the customer with a common look-and-feel. The customer gets, independent of which wind tunnel is used, the results in a standard manner. The common WDF results will be contained in a binary direct access format that is platform independent. To reach this target the standard already in use in the atmospheric research community is adopted. Routines to handle this standard, called Network Common Data Format (netCDF), are available for free. The format definition is identical for all Onera-GMT and DNW wind tunnels, the programs for reading and visualising the data from this format are also identical for all wind tunnels. Customers of Onera-GMT and DNW wind tunnels are provided with data on the WDF media which contains the results that are used for further processing on their computing facilities. Customers use one and only one program to transfer the data to their home facilities. Onera-GMT and DNW will provide their customers with support, including the initial use of the necessary programs and libraries.

Written by

Reinier van der DRAAI, DNW

Detlef KRACK, DNW

Bernard LALLEMENT, ONERA

Paul REGIS, ONERA

Distribution list of DNW / ONERA-GMT report ATA-TR-001-2002

DNW

Mr. G. Eitelberg
Mr. K.W. Bock
Mr. J. Kooi
Mr. C. Joosen
Mr. D. Eckert
Mr. R.K. v d Draai
Mr. D. Krack
Mr. H.G. Pagendarm

ONERA

DGMT Mr. X. Bouis
DGMT Mr. M. Bazin
DTEX Mr. P. Wagner
DTEX Mr. G. Grenat
DTEX Mrs M. Lyonnet
DTEX Mr. P. Régis
DTEX Mr. P. Vallon
DSMA Mr. J.P. Bècle
DSFM Mr. J.M. Carrara
DSFM Mr. A. Mignosi
DSFM Mr. J. Seret
DSFM Mr. D. Cassoudealle
DRIS Mr. F. Mescam

HISTORY

Version

1.0	May 2002	First common document between DNW and ONERA-GMT
1.1	May 2002	Re-organisation of the document.
1.2	June 2002	Attribute long_name
1.3	June 2002	Distribution list
1.4	July 2002	Reference to DNW / ONERA-GMT
1.5	September 2002	References to DNW formats and new WdPlot application.
1.55	September 2004	Dots accepted in variable names.
2.0	May 2006	File naming rules, organisation of the medium.
2.1	November 2006	Use of standard font (Helvetica) for WDF documents.

TABLE OF CONTENTS

1. INTRODUCTION	6
2. WINDTUNNEL DATA FORMAT.....	6
2.1. DEFINITIONS :	7
2.2. SPECIFICATIONS OF WDF FORMAT :	7
2.3. BENEFITS FOR THE CLIENT	9
2.4. SOFTWARE DEVELOPMENTS :	10
2.4.1. <i>WDF Files Viewer: WdView</i>	10
2.4.2. <i>WDF Files Plotter: WdPlot</i>	10
2.4.3. <i>Medium Index Creator</i> :.....	11
2.4.4. <i>Gfo-to-Wdf converter</i> :	11
2.4.5. <i>DNW-to-WDF converter</i>	11
2.4.6. <i>Fortran WDF Library</i>	11
2.4.7. <i>C WDF Library</i>	12
2.4.8. <i>Java WDF Library</i>	12
2.5. DEMONSTRATOR	12
3. ORGANISATION OF THE MEDIUM	14
3.1. DESCRIPTION OF THE MEDIUM	14
3.2. FILE FORMATS AVAILABLE ON THE MEDIUM.....	16
4. REFERENCES	17
5. SYMBOLS AND ABBREVIATIONS.....	17
6. APPENDIX A : DIFFERENCES BETWEEN PROVIDERS.....	17
7. APPENDIX B – WDF FORTRAN LIBRARY.....	18
7.1. FILE ACCESS	18
7.2. VARIABLES ACCESS.....	19
7.3. RECORD READING UTILITY ROUTINES	20
7.4. RECORD ACCESS.....	21
7.5. EXTRACTION ROUTINES	22
7.6. UTILITY ROUTINES	24
7.7. DISPLAY ROUTINES	25

1. INTRODUCTION

The format for the numerical data, the so-called classical data has been fixed for most DNW and ONERA-GMT facilities since 1980's. Since that time new data sources were added to the wind tunnel data flow that had to be delivered separate of the steady-state data. Twenty years later, it is time to define a new format for the classical data, and a new structure for all data. The new structure is open to data of new sources to come. This new Wind Tunnel Data Format (WDF) for classical data is based on 'netCDF', the standard that is already in use for years in the atmospheric research community. This format is platform independent. Its content is fully self-describing and ready for adoption of many known, and also not yet known data sources.

It should be noted that, upon customer request, the formats described in the reports of ONERA [6], DNW [8] and NLR [7] with regard to former data exchange output definitions, remain possible output formats. Furthermore in the case of DNW formats, this means the three main formats currently in use, DNW-LLF results.nnn format, DNW-HST TOUT format and DNW-TWG DeAs format, are replaced.

This report describes the specifications of the Common Wind Tunnel Data Format and the medium organisation.

The description of the specifications gives the reader insight in the structure of the data files.

The organisation on the medium is hierarchical of nature and gives easy to understand access to data of wind tunnel related results, such as the so-called classical data, photos, videos, PIV, PSP,...

2. WINDTUNNEL DATA FORMAT

The product netCDF, used to create WDF files, provides the following standard features:

- Binary storage of the data, enabling less wasted space on the medium.
- Direct access to the data.
- Data is stored in an eXternal Data Representation (XDR) format (independent of the platform used)
- Software libraries exist for different languages: C, C++, Fortran, Perl and Java.
- netCDF is a self-describing format: The file contains the description of its contents.
- netCDF is a free software: without licensing.

However, the netCDF product doesn't provide any record concept, and the default value of the maximum number of variables is 2000. So, deviations of the netCDF standard have to be implemented to permit record concept to be available for WDF files, as well as up to 5000 different variables accessible in one netCDF file, which must have a unique name along one netCDF file. This netCDF file is called a netCDF dataset.

More information about netCDF can be gained by visiting the web site <http://www.unidata.ucar.edu/packages/netcdf/>. More technical information upon netCDF can be found in reference [4]: netCDF Fortran Guide.

2.1. Definitions :

Variable : a variable is a named value, measured or computed or defined, single or matrix.

Record : a record is a set of homogeneous variables, i.e. measurements acquired at the same time, or results computed from the same raw data record, except for an “average” record. A record is identified by a unique name.

DPN : a Data Point is a processed set of channel readings pertaining to a particular measurement and model configuration and taken at stationary or semi-stationary conditions. A Data Point can be constituted with several different records or variables.

Polar : a Polar is a set of successive data points, pertaining to a particular measurement and model configuration, taken at stationary or semi-stationary testing conditions except for one independent controlled parameter.

Run : a Run is a set of successive polars, pertaining to a particular measurement and model configuration preceded by some sets of control data and a Data Point under no wind conditions. The run will be terminated by a defined set of control data , possibly preceded by a Data Point under no wind conditions. During a run, no changes can be made manually to the model. The Run usually begins when the wind starts, and ends after the wind has stopped.

Series : a Series is a group of runs, pertaining to a particular measurement and model configuration.

Test : a Test is a collection of runs or series made for a particular customer, usually confined to a specific model of which several model configurations are to be examined.

2.2. Specifications of WDF format :

One WDF file will be a netCDF dataset : variables with their names, attributes of each variables, dimensions of arrays..., as well as global attributes associated to the contents of the WDF file. The present version “v3.5” of the netCDF software library will be used.

The different kinds of variables accepted by WDF are:

- character (8 bits text),
- Byte (8 bits integer),
- Short integer (16 bits),
- Integer (32b),
- float (32bits standard IEEE),
- double (64bits),
- Multidimensional arrays

Associated to each variable, attributes may be defined:

- *long_name*: contains the name of the variable (mandatory only if RecordConcept is “FULL”, else optional).
- *units*: This standard attribute exists and may be used in the future.

Associated with the WDF file, global attributes are used to add information to the contents of the file. Especially with ISO 9001, the following global attributes are mandatory.

- Version (integer) Version number of the WDF software library.
- CreationDate (character) Creation Date and Time of the file.
- Provider (character) DNW / ONERA.
- Windtunnel (character) Wind tunnel used for the test.
- TestTitle (character) Description of the Test
- RunTitle (character)
- RunNumber (character) number of run, or list of runs, or range of runs
- PolarNumber (character) number of polar, or list of polars, or range of polars

- RecordConcept (Character) if present (= "FULL or SHORT"), means that the record concept is available, else if missing, record concept is not available.

The following global attributes are optional:

- AcquisitionProgramName (character)
- AcquisitionConstantName (character)
- AcquisitionDate (character) « yyyy-mm-dd hh :mm :ss »
- DataReductionProgramName (character)
- DataReductionConstantName (character)
- DataReductionDate (character)
- URL (character) used to link to other files

This list is pretty open and can be extended according to available information.

The minimum addressable file unit is a polar file. A WDF file can contain, dependent on the provider:

- One or more POLARs pertaining to the same RUN
- One or more RUNs pertaining to the same SERIES
- One or more SERIES pertaining to the same TEST
- A TEST

WDF files will have the ".wdf" extension.

The name of a WDF file may take the following form, with respect to the "LONG NAMES" rule: alphabetic or numeric characters, plus minus and underscore, without spaces, maximum 64 characters, not including the path name, within bracket means optional:

[<nature>_]C<number>[_<suffix>].wdf where :

<nature> may indicate the nature of the content (PRESSURE, INTERPOL), optional

C indicates : T for a single TEST
 S for a single SERIES
 R for a single RUN
 P for a single POLAR

<number> is the number of the SERIES, the RUN or the POLAR.

Five digit are required, left padded with "0"

<suffix> optional information, free

Specifications for the names of the variables:

- There is no limitation about the name length of each variable: just a suggested limit: up to 64 characters.
- Only alphabetic and numeric character plus minus, dots and underscore are allowed.
- Names of the variables are case sensitive ("Mach" is different from "mach")
- Maximum number of variables in one WDF file: 5000.
- The name of the variable follow the rule, depending on the state of *RecordConcept*:
 - If *RecordConcept* is equal to "FULL", the name of the variable is equal to "VariableName-RecordName-" (called qualified name)
 - If *RecordConcept* is missing, or present with the value "SHORT", when existing, the *long_name* attribute must be equal to the name of each variable. If the optional *long_name* attribute doesn't exist, the reading software will assume that it is equal to the variable name for internal use.
- Due to the possible missing of the record concept in a WDF file, the reading software must always use the qualified name to access a variable, and use the *long_name* attribute to display the names of the variables.

If the concept of record is available, the global attribute “*RecordConcept*” must exist with the value “SHORT” or “FULL” and the following five extra variables are mandatory:

- RecordNames (character*16 array) Names array of the records
- RecordCounts (integer array) Counts array of each record
- RecordStart (integer array) Index of the first variable belonging to each record
- RecordEnd (integer array) Index of the last variable belonging to each record
- RecordOrder (integer array) Order of the records.

If the global attribute “*RecordConcept*” doesn’t exist, the reading application will use the following default values:

- RecordNames = “RECORD”
- RecordCounts(1) = -1 not known
- RecordStart(1) = 1 first variable
- RecordEnd(1) = N number of variables
- RecordOrder(*) = 1 always the same.

A WDF file can contain a Test, Series, Runs, or Polars. Information to filter on specific series, run, polar, or data point number will be provided by adding the following extra variables:

- SERIES
- RUN
- POLAR
- DPN

These extra variables are mandatory, of integer type with the default value “-1” if meaningless.

2.3. Benefits for the client

The primary objective of the WDF project is to provide our clients with all the tests results together, and modern tools to analyse these results. To reach this goal, software provided will be free of charge, and everything will be made to permit installation of software as easy as possible. Applications will be developed with graphical user interface so that their use will be more friendly.

This WDF common format is proposed by both DNW and ONERA, which permits the client to receive a medium with always the same data structure : results, documentation and software tools. However, documentation and software will be available only on the first medium in case of several media existing. The client will receive the same data format coming from DNW or ONERA. A converter from WDF format to current formats will permit the client to delay the implementation of WDF.

The client will get software libraries and installation procedures for different platforms and for different languages for netCDF and WDF :

- Fortran and C with VMS.
- Fortran and C with UNIX.
- Java

These software libraries will permit the client to develop its own software more easily.

Well known limitations of the existing current formats will be overcome, such as the number of record, the number of variables per record, variables not supported or even accuracy.

File structure of WDF is open, global and variable attributes can be added upon request of the clients, according to available information.

Thanks to the binary storage, WDF file will be about three times smaller than ASCII files.

The direct or random access to variables permits to read average values or results or records before the values themselves.

A WDF viewer programme will be provided to the client. This application will be developed in JAVA to permit platform independence. This application will also be able to convert WDF files into files readable by spreadsheet software or Tecplot or Matlab software. The conversion to the old DNW formats or GMT files, (well known as BMSTD format) will continue to exist, but only if the DNW or BMSTD old requirements are satisfied. It should be noticed that this option does not include the advantages of the WDF format.

A WDF plotting application will be provided to the client as well. Based on the Gsharp commercial product, this tool will permit the client to plot WDF files with graphs usually used. The product will be free since the license will be embedded in the application. This application will be available for different platforms: Windows, Linux, UNIX and OpenVMS.

2.4. Software developments :

2.4.1. WDF Files Viewer: WdView

This application will be the central application of the WDF project. WdView will be written in JAVA language to permit portability. The features available with WdView will be:

- Read and display WDF files contents
- Allow filtering to permit access to
 - SERIES
 - RUNs
 - POLARs
 - DPNs if applicable
 - Records if applicable
 - Variables
- Allow renaming of records or variables
- Creation and recording of such filters, including renaming information to permit reusability.
- Allow creation of output formats
 - CSV files, readable by a spreadsheet programme, or XMGRACE for Unix table oriented programmes.
 - GMT files, well known as BMSTD format, only if BMSTD requirements are satisfied.
 - MATLAB files to import results in the MATLAB application.
 - TECPLOT files
 - DNW files, only if DNW files requirements are met
 - WDF files themselves
 - Others to be defined later ...
- Implementation of a test bench to verify that WDF works correctly on the used platform, that means, that the software is correctly installed, and conversions are made correctly.

2.4.2. WDF Files Plotter: WdPlot

This WDF plotting application uses the graphical package Gsharp from AVS Company as a basis. The multi platform graphical application will contain the following features in a first version:

- Menu structure
- Any components that come with Gsharp
- Output to common image formats

- Saving and loading layouts
- Read multiple WDF files
- Select x and y data
- Multiple curves within same axes
- Axis limits auto-calculated or user specified
- Legend showing polar and data point number, also possible global attributes, variable names
- User defined title

Install scripts will support installation on customer computer for WINxx, Linux, UNIX and OVMS operating systems. A run-time license is embedded in the application.

2.4.3. Medium Index Creator:

As can be seen with the description of the hierarchy on the medium in paragraph §3, the hierarchy permits an access to individual files from the top of the hierarchy down to the bottom, called vertical access to data. An application will be developed to permit access "horizontally", that means permit access to everything belonging to the same SERIES or RUN or POLAR or even DPN. This application will be able to produce an index.html file describing the whole test results so that technical staff can add documentation on the test easily.

This will permit the client to access the medium as a web site.

2.4.4. Gfo-to-Wdf converter :

This application is needed by ONERA to convert his internal GFO files and create WDF files. Due to the availability of GFO library only in Fortran, this application will be written in Fortran. This application will allow filtering to permit access to:

- Runs
- Polars
- Records
- Variables

The application will also allow creation of the hierarchy defined in paragraph §3. That means Pxxx.wdf under Rxxx directories, Rxxx.dir under DATA directoy. This will be fully automated.

2.4.5. DNW-to-WDF converter

These applications are needed by DNW to convert the internal results files and create WDF files. These applications will allow filtering to permit access to:

- Series
- Runs
- Polars
- Variables

2.4.6. Fortran WDF Library

This library will permit the client to develop its own software providing him the way to read WDF files and write the results into its own database.

The following procedures will be provided.

2.4.6.1. File access procedures

Integer Function WDF_OpenFile (Filename)
Integer Function WDF_GetVariablesInfos ()
Integer Function WDF_GetStructureInfos ()

2.4.6.2. Variables access procedures

Integer Function WDF_GetVar_Ch (Varnam,Charray,Lenstr,Size)
 Integer Function WDF_GetVar_I2 (Varnam,I2array,Size)
 Integer Function WDF_GetVar_I4 (Varnam,I4array,Size)
 Integer Function WDF_GetVar_R4 (Varnam,R4array,Size)
 Integer Function WDF_GetVar_R8 (Varnam,R8array,Size)

2.4.6.3. Record reading procedures

Integer Function WDF_GetPolars (PolarList,PolarStart,PolarEnd,PolarNumber)
 Integer Function WDF_GetRuns (RunList,RunStart,RunEnd,RunNumber)
 Integer Function WDF_GetSeries (SeriesList,SeriesStart,SeriesEnd,SeriesNumber)
 Integer Function WDF_StartRead (Range,Number)
 Integer Function WDF_EndRead (Range,Number)

2.4.6.4. Record access procedures

Integer Function WDF_GetRecordInfos (ReclIndex,RecName,Nbvar)
 Integer Function WDF_GetRecord (Ie,RecordName,RecordIndex,Buffer,Nbvar,RecordSize)
 Integer Function WDF_GetSelectedRecord (Ie, SelectList, NbSelect, RecordName,
 RecordIndex, Buffer, Nbvar, RecordSize)

2.4.6.5. Extraction procedures

Integer Function WDF_GetVariableInfos (VarIndex, ReclIndex, Varname, VarType, Dims,
 Nbdims)
 Integer Function WDF_GetI2FromRecord (Iv,RecordIndex,Buffer,Tab,TabSize)
 Integer Function WDF_GetI4FromRecord (Iv,RecordIndex,Buffer,Tab,TabSize)
 Integer Function WDF_GetR4FromRecord (Iv,RecordIndex,Buffer,Tab,TabSize)
 Integer Function WDF_GetR8FromRecord (Iv,RecordIndex,Buffer,Tab,TabSize)
 Integer Function WDF_GetStringFromRecord (Iv,Buffer,Tab,TabLen,TabSize)

2.4.6.6. Utility procedures

Integer Function WDF_SetIoUnit (Unit)
 Subroutine HANDLE_ERR (Io,STATUS,LEVEL)
 Subroutine BYTE2BYTE (Source,Destination,Size)
 Subroutine BYTE2CHAR (Source,Destination,Size)
 Subroutine CHAR2BYTE (Source,Destination,Size)
 Subroutine Suptrbl (Chain, Lchain)

2.4.6.7. Display procedures

Integer Function WDF_DisplayDimensions ()
 Integer Function WDF_DisplayGatts ()
 Integer Function WDF_DisplayVnames ()

2.4.7. C WDF Library

Every Fortran procedure defined in paragraph §2.4.6 will be available in the C language.

2.4.8. Java WDF Library

Classes developed for WdView should be reusable for other software.

2.5. Demonstrator

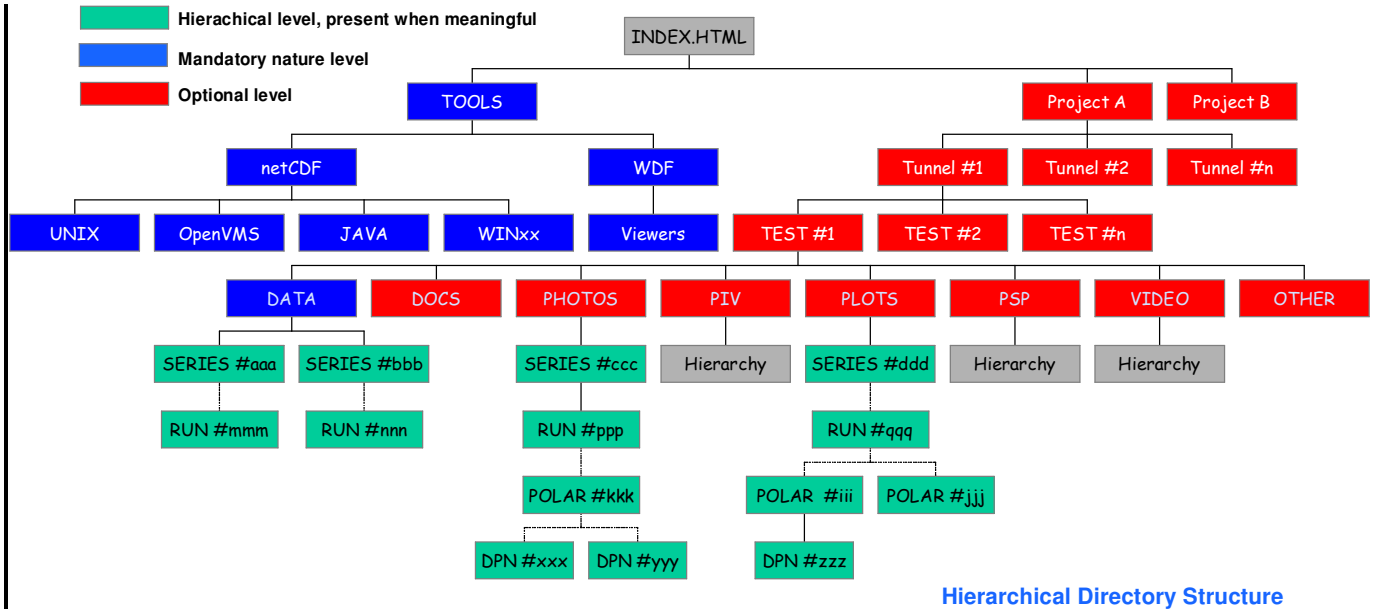
Provided with this documentation, comes an example piece of software and two WDF files are available to show what a WDF file is.

This demonstrator contains:

- At least one WDF file coming from DNW.
- At least one WDF file coming from ONERA.
- the WdView application with reduced features:
 - View of the hierarchy and the contents of a file
 - Major part of the filtering capacity (Series, Run, Polar, Records and Variables)
 - Export to CSV file.
- Export to MATLAB application.

3. ORGANISATION OF THE MEDIUM

The medium containing all relevant data and access software will be organised in a hierarchical manner. A common directory hierarchy as shown in the following chart has been created. To keep track with flexible recommendations the medium organisation has been distinguished into mandatory, optional and “present when meaningful” levels. The LONG NAMES file format for CD-ROM media will be used. A **public** access web site (www.ata.aero/wdf/) containing a template structure of the medium will be created and maintained.



3.1. Description of the medium

All software contained on the medium will be usually in an English version, if meaningful, versions in French, Dutch or German will be provided. Only free of charge software or software with embedded license will be put onto the medium. Documentation will be provided in pdf format. As an general entry (main directory) the medium contains an index in html format (INDEX.HTML).

The following subdirectories are present and have the following contents:

- **TOOLS**

- Contains general documents on WDF format

- **netCDF**

- Contains all relevant files to create the platform dependent version of a netCDF library

- **UNIX**

- NetCDF.zip for HP, SGI, SUN systems containing current version of sources netCDF library inclusive ‘make’ configuration script files. The binary library contains all default nc utilities (e.g. ncdump, nctest...). Also a link to the UNIDATA url will be contained.

- **OpenVMS**
NetCDF.bck for Compaq Alpha systems running OpenVMS operating system containing last version of binary and source code netCDF library inclusive MAKE configuration command files for building the library and all nc utilities. A converter application from WDF to csv will also be found here as an example. Also a link to the UNIDATA url will be contained.
- **JAVA**
NetCDF.jar for systems running the JAVA environment. Also a link to the UNIDATA url and a “how to install” document will be contained.
- **WINxx**
NetCDF.zip for systems running a WIN32 operating system containing last version of binary netCDF library. Also a link to the UNIDATA url will be contained.
- **WDF**
 - **Viewers**
Contains all viewing/plotting applications related to WDF files.
 - WdView
The WdView (JAVA) application inclusive installation scripts for different supported operating systems will be found here.
 - WdPlot
The WdPlot (AVS) application inclusive installation scripts for different supported operating system will be found here.
- **Project (opt.)**
 - **Tunnel (opt.)** Name of ATA wind tunnel
 - **TEST (opt.)** Name/number of wind tunnel test
 - **DATA**
Contains wind tunnel results in a hierarchical structure, one directory per Series or Run whatever is relevant. Series/Run/Polar hierarchy will be present under nature level, only if relevant. If files are not related to specific Series/Run/Polar, these files will be put directly in the nature level. The minimum addressable file unit is a polar file. The data-point level will be hidden, but addressable by the viewing applications.
 - **SERIES #aaa**
 - **RUN #nnn** (when meaningful)
Contains results data of the run number #nnn
 - **DOCS**
Documentation relevant to wind tunnel test “TEST”
 - **PHOTOS**
Contains photo’s in jpeg, tiff or other relevant format.
 - **SERIES #ccc** (when meaningful)
Contains photo’s related to Series #ccc

} Relevant on
request of the
clients

- **RUN #ppp** (when meaningful)
Contains photo's related to Run #ppp
 - **POLAR #kkk** (when meaningful)
Contains photo's related to Polar #kkk
 - **DPN #xxx** (when meaningful)
Contains photo's related to data point #xxx
- **PIV**
Contains PIV results in netCDF format related to Test # A
- **PLOTS**
Contains plots related to Test #1
- **PSP**
Contains PSP results related to Test #1
- **VIDEO**
Contains videos in MPEG/mp4/5 format related to Test #1
- **OTHER** known/unknown format results, "Other" means not yet defined, this structure is open to receive new formats.

3.2. File formats available on the medium

The following files formats (dependent on the data) will be available on the medium:

- csv created by the converter from WDF to csv format to be imported by MS EXCEL
- jpg/jpeg used to store pictures in a compressed format (information loss), readable with numerous photo display applications
- mpeg video files, viewable with windows media player or Real Player.
- pdf portable documentation file format, readable with e.g Acrobat Reader
- ps postscript format files, usually plot files, can be displayed with e.g. Gsview
- tif/tiff used to store pictures in a compressed or not compressed format, not compressed format is very suitable for further editing. Readable with numerous photo editing applications.
- txt ASCII text files, readable with every text editor
- xls files in MS EXCEL format
- gmt created by the old BMSTD well known ONERA application
- dnw results data files created by the old DNW well known data processing applications
- wdf results data files in the new standard "Windtunnel Data Format"

4. REFERENCES

- [1] RT 3/03813 DTEX (October 2000) Proposal for Wind tunnel Data Formats used for test results presentation
- [2] RT 4/03813 DTEX Electronic Communication with Windtunnel's Clients.
- [3] netCDF-Cguide.pdf C documentation on netCDF
- [4] netCDF-FortranGuide.pdf Fortran documentation on netCDF
- [5] netCDF-JavaManual.pdf Java documentation on netCDF
- [6] RTI 63/1165 CK The BMSTD format specifications.
With 3 add-on april 1985, 255.406/cma april 1985, april 1986
- [7] A-87-008-U The NLR TOUT Tape
- [8] no ref. Definition of the DNW results stream from 15.10.1999

5. SYMBOLS AND ABBREVIATIONS

CDF : Common Data Format

netCDF : Network Common Data Form : An interface for array-oriented data access and a freely-distributed software library that provides implementation of the interface.

PIV : Particle Image Velocimetry

PSP : Pressure Sensitive Paint

WDF : Wind tunnel Data Format

6. APPENDIX A : DIFFERENCES BETWEEN PROVIDERS

The following list shows what is specific to each provider.

	ONERA	DNW-ASD	DNW-NOP	DNW-GUK
Availability of Series	No	Yes	Yes	Yes
Availability of Runs, Polars, DPN	Yes	Yes	Yes	Yes
Record concept available	Yes	Yes	Yes	No
Contents of one WDF file	Polar	Series	Series	Test
Global attributes available in a WDF file	Yes	Yes	Yes	Yes
Extra variables SERIES, RUN, POLAR, DPN	Yes	Yes	Yes	Yes

7. APPENDIX B – WDF FORTRAN LIBRARY

7.1. File Access

```

C *****
C Integer Function WDF_OpenFile (Filename)
C *****
C
C This routine :
C _ Opens the File (NF_OPEN)
C _ Collects global informations (NF_INQ) :
C   _ Number of Dimensions
C   _ Number of Variables
C   _ Number of Global Attributes
C   _ Unlimited dimension ID
C _ Check limites (Dimensions, Variables, Attributes)
C _ Collects dimensions infos (NF_INQ_DIM) :
C   _ Name
C   _ Size
C _ Collects global attributes infos (NF_INQ_ATTNAME, NF_INQ_ATTTYPE,
C   NF_INQ_ATTLEN) :
C   _ Name
C   _ Type
C   _ Size
C _ Collects global attributes values according to type (NF_GET_ATT_XXX)
C
C All collected values are stored in COMMON
C
C Input
C *****
C
C Filename   Character*(*)   WDF file Name
C
C *****

```

```

C *****
C Integer Function WDF_GetVariablesInfos ()
C *****
C
C This routine :
C _ Collects Variables informations (NF_INQ_VAR) :
C   _ NetCDF Name
C   _ NetCDF Type
C   _ Number of Dimensions
C   _ Dimensions ID's
C   _ Number of Attributes
C _ Collects Variable long_name attribute (if existing)
C
C All collected values are stored in COMMON, in relevant structure
C
C *****
C Integer Function WDF_GetStructureInfos ()
C *****
C
C Utility routine to collect structuring variables infos :
C
C _ RecordNames
C _ RecordCounts
C _ RecordOrder
C _ RecordStart
C _ RecordEnd
C
C Informations are stored in relevant COMMON
C
C A call to this routine must be done prior to any record access
C
C If RecordConcept global attribute has been found, structuring
c variables will be searched for. If not, default values will be used :
C RecNbr      = 1
C TotRecNbr   = Unlimited Dimension Length
C RecordNames (1) = 'RECORD '
C RecordStart (1) = 1
C RecordEnd   (1) = Nvars
C RecordCounts(1) = -1
C RecordOrder (*) = 1
C
C *****

```



7.2. Variables Access

```

C *****
C Integer Function WDF_GetVar_Ch (Varnam,Charray,Lenstr,Size)
C *****
C
C This routine returns a complete String NetCDF variable,
C retrieved by its NetCDF name. It is limited to one dimension
C variables.
C
C Input
C *****
C Varnam      C*(*) NetCDf variable name
C
C Output
C *****
C Charray     C*(1) Variable array
C Lenstr      I4 String length
C Size        I4 Array size
C
C
C
C
C *****
C Integer Function WDF_GetVar_I2 (Varnam,I2array,Size)
C *****
C
C This routine returns a complete Integer*2 NetCDF variable,
C retrieved by its NetCDF name. It is limited to one dimension
C variables.
C
C Input
C *****
C Varnam      C*(*) NetCDf variable name
C
C Output
C *****
C I2array     I2(1) Variable array
C Size        I4 Array size
C
C
C
C
C *****

```

```

C *****
C Integer Function WDF_GetVar_I4 (Varnam,I4array,Size)
C *****
C
C This routine returns a complete Integer*4 NetCDF variable,
C retrieved by its NetCDF name. It is limited to one dimension
C variables.
C
C Input
C *****
C Varnam      C*(*) NetCDf variable name
C
C Output
C *****
C I4array     I4(1) Variable array
C Size        I4 Array size
C
C
C
C
C *****
C Integer Function WDF_GetVar_R4 (Varnam,R4array,Size)
C *****
C
C This routine returns a complete Real*4 NetCDF variable,
C retrieved by its NetCDF name. It is limited to one dimension
C variables.
C
C Input
C *****
C Varnam      C*(*) NetCDf variable name
C
C Output
C *****
C R4array     R4(1) Variable array
C Size        I4 Array size
C
C
C
C
C *****

```

```

C *****
C Integer Function WDF_GetVar_R8 (Varnam,R8array,Size)
C *****
C
C This routine returns a complete Real*8 NetCDF variable,
C retrieved by its NetCDF name. It is limited to one dimension
C variables.
C
C Input
C *****
C Varnam      C*(*) NetCDf variable name
C
C Output
C *****
C R8array     R8(1) Variable array
C Size       I4  Array size
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

7.3. Record reading Utility Routines

```

C *****
C Integer Function WDF_GetPolars (
C   & PolarList      ! List of Polar
C   & ,PolarStart    ! Starting index of Polar
C   & ,PolarEnd      ! Ending index of Polar
C   & ,PolarNumber   ! Number of Polar
C   & )
C *****
C
C Utility routine to collect polar infos :
C
C Output
C *****
C PolarList  I4(1) List of Polar numbers
C PolarStart I4(1) Array of starting indexes of Polar
C PolarEnd   I4(1) Array of ending indexes of Polar
C PolarNumber I4  Number of Polar or 0 if POLAR doesn't
C              exist
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

```

C *****
C Integer Function WDF_GetRuns (
C   & RunList      ! List of Run
C   & ,RunStart    ! Starting index of Run
C   & ,RunEnd      ! Ending index of Run
C   & ,RunNumber   ! Number of Run
C   & )
C *****
C
C Utility routine to collect run infos :
C
C Output
C *****
C RunList    I4(1) List of Run numbers
C RunStart   I4(1) Array of starting indexes of Run
C RunEnd     I4(1) Array of ending indexes of Run
C RunNumber  I4  Number of Run or 0 if RUN doesn't
C              exist
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

```

C *****
C Integer Function WDF_GetSeries (
C   & SeriesList     ! List of Series Number
C   & ,SeriesStart   ! Starting index of Series
C   & ,SeriesEnd     ! Ending index of Series
C   & ,SeriesNumber  ! Number of Series
C   & )
C *****
C
C Utility routine to collect series infos :
C
C Output
C *****
C SeriesList  I4(1) List of Series Number
C SeriesStart I4(1) Array of starting indexes of Series
C SeriesEnd   I4(1) Array of ending indexes of Series
C SeriesNumber I4  Number of Series or 0 if SERIES doesn't
C              exist
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```



```

C          If > 0 specific record number to read
C  Output
C  *****
C  RecordName  C*(*) Name of record
C  RecordIndex I4  Index of record in structuring variables infos
C  Buffer      Bytes Untyped buffer to store record values
C  Nbvar      I4  Number of variables in record
C  RecordSize I4  Size of used buffer
C
C
C
C *****
C  Integer Function WDF_GetSelectedRecord (Ie,SelectList,NbSelect,
C  & RecordName,RecordIndex,
C  & Buffer,Nbvar,RecordSize)
C *****
C
C  Record oriented Read Access to selected record
C
C  This routine reads a complete record and store values in an untyped
C  buffer. The returned record is chosen in a list of selection
C  indexes
C
C  Input
C  Ie      R4  Record number to read :
C          If < 0 records are read in sequential order
C          from starting point (See WDF_StartRead) to
C          ending point (see WDF_EndRead)
C          If > 0 specific record number to read
C  SelectList I4(1) List of record indexes (according to
C          RecordNames structuring variable order) to
C          select. Only records existing in this list
C          will be returned. Index value of -1 allows
C          ALL records to be selected
C  NbSelect  I4  Number of items in SelectList
C
C  Output
C  *****
C  RecordName  C*(*) Name of record
C  RecordIndex I4  Index of record in structuring variables infos
C  Buffer      Bytes Untyped buffer to store record values
C  Nbvar      I4  Number of variables in record
C  RecordSize I4  Size of used buffer
C
C
C
C *****

```

7.5. Extraction Routines

```

C *****
C  Integer Function WDF_GetVariableInfos
C  & (VarIndex
C  & ,ReclIndex
C  & ,Varname
C  & ,VarType
C  & ,Dims
C  & ,Nbdims)
C *****
C
C  This routine returns informations about a specific NetCDF
C  variable identified by its index within the record, also
C  identified by its index within declared records.
C
C  Input
C  *****
C  VarIndex  I4  Index of variable within the record
C              (start at 1)
C  ReclIndex I4  Index of record to consider
C
C  Output
C  *****
C  Varname   C*(*) Variable name (from long_name attribute)
C  VarType   C*2  Variable type (I2,I4,R4,R8,C*)
C  Dims      I4(1) Dimensions values
C  Nbdims    I4  Number of dimensions (FORTRAN order)
C
C
C *****
C  Integer Function WDF_GetI2FromRecord (Iv,RecordIndex,Buffer,
C  & Tab,TabSize)
C *****
C  This routine must be used to extract Short integer value (scalar or
C  array) from an untyped buffer returned by WDF_GetRecord or
C  WDF_GetSelectedRecord.
C
C  Input
C  *****
C  Id      I4  Variable number within the record (starts at 1)

```



```

C RecordIndex I4 Index of record in structuring variables infos
C (Returned by WDF_GetRecord)
C Buffer Bytes Untyped buffer to store record values
C (Returned by WDF_GetRecord)
C Output
C *****
C Tab I2(1) Array to receive extracted values
C Tabsize I4 Number of elements in Tab (0 if incorrect type
C 1 if scalar)
C Returns
C *****
C 1 At least one value is extracted
C 0 No value extracted (type does not agree)
C
C *****
C Integer Function WDF_GetI4FromRecord (Iv,RecordIndex,Buffer,
C & Tab,TabSize)
C *****
C This routine must be used to extract integer value (scalar or
C array) from an untyped buffer returned by WDF_GetRecord or
C WDF_GetSelectedRecord.
C
C Input
C *****
C Id I4 Variable number within the record (starts at 1)
C RecordIndex I4 Index of record in structuring variables infos
C (Returned by WDF_GetRecord)
C Buffer Bytes Untyped buffer to store record values
C (Returned by WDF_GetRecord)
C Output
C *****
C Tab I4(1) Array to receive extracted values
C Tabsize I4 Number of elements in Tab (0 if incorrect type
C 1 if scalar)
C Returns
C *****
C 1 At least one value is extracted
C 0 No value extracted (type does not agree)
C *****
C *****

```

```

C *****
C Integer Function WDF_GetR4FromRecord (Iv,RecordIndex,Buffer,
C & Tab,TabSize)
C *****
C This routine must be used to extract float value (scalar or
C array) from an untyped buffer returned by WDF_GetRecord or
C WDF_GetSelectedRecord.
C
C Input
C *****
C Id I4 Variable number within the record (starts at 1)
C RecordIndex I4 Index of record in structuring variables infos
C (Returned by WDF_GetRecord)
C Buffer Bytes Untyped buffer to store record values
C (Returned by WDF_GetRecord)
C Output
C *****
C Tab R4(1) Array to receive extracted values
C Tabsize I4 Number of elements in Tab (0 if incorrect type
C 1 if scalar)
C Returns
C *****
C 1 At least one value is extracted
C 0 No value extracted (type does not agree)
C *****
C *****
C *****
C Integer Function WDF_GetR8FromRecord (Iv,RecordIndex,Buffer,
C & Tab,TabSize)
C *****
C This routine must be used to extract double real value (scalar or
C array) from an untyped buffer returned by WDF_GetRecord or
C WDF_GetSelectedRecord.
C
C Input
C *****
C Id I4 Variable number within the record (starts at 1)
C RecordIndex I4 Index of record in structuring variables infos
C (Returned by WDF_GetRecord)
C Buffer Bytes Untyped buffer to store record values
C (Returned by WDF_GetRecord)

```

```

C Output
C *****
C Tab      R8(1) Array to receive extracted values
C Tabsize  I4   Number of elements in Tab (0 if incorrect type
C           1 if scalar)
C
C Returns
C *****
C 1          At least one value is extracted
C 0          No value extracted (type does not agree)
C
C
C
C *****
C Integer Function WDF_GetStringFromRecord (lv,Buffer,Tab,TabLen,
C &                                     TabSize)
C *****
C
C This routine must be used to extract String value (scalar or
C array) from an untyped buffer returned by WDF_GetRecord or
C WDF_GetSelectedRecord.
C
C Input
C *****
C Id      I4   Variable number within the record (starts at 1)
C RecordIndex I4   Index of record in structuring variables infos
C           (Returned by WDF_GetRecord)
C Buffer   Bytes Untyped buffer to store record values
C           (Returned by WDF_GetRecord)
C
C Output
C *****
C Tab     C*(*) Array to receive extracted values
C Tabsize I4   Number of elements in Tab (0 if incorrect type
C           1 if scalar)
C Tablen  I4   Length of each element in Tab
C
C Returns
C *****
C 1          At least one value is extracted
C 0          No value extracted (type does not agree)
C
C
C

```

7.6. Utility routines

```

C *****
C Integer Function WDF_SetIoUnit (Unit)
C *****
C
C Utility routine to specify FORTRAN output unit to be used on any
C output message statement. Must be called prior to any other WDF
C routine. Assignment to unit must be done before by calling
C program
C
C Input
C *****
C Unit     I4   FORTRAN I/O unit
C
C
C *****
C Subroutine HANDLE_ERR (Io,STATUS,LEVEL)
C *****
C
C Utility routine used to display NetCDF error message associated
C to error code.
C
C Input
C *****
C Io      I4   Current Output Unit (set by WDF_SetIoUnit)
C Status  I4   NetCDF error code
C Level   I4   Index available to specify call level
C
C
C *****
C Subroutine BYTE2BYTE (Source,Destination,Size)
C *****
C
C Utility routine to move bytes from one location to another
C
C Input
C *****
C Source  Bytes   Array to move
C Destination Bytes Destination location
C Size    I4      Number of bytes to move
C
C
C

```




```

C *****
C Subroutine BYTE2CHAR (Source, Destination, Size)
C *****
C
C Utility routine to move bytes from byte location to string
C location.
C
C Input
C *****
C Source      Bytes      Array to move
C Destination C*(*)     String destination location
C Size       I4         Number of bytes to move
C
C
C
C *****
C Subroutine CHAR2BYTE (Source, Destination, Size)
C *****
C
C Utility routine to move bytes from string location to byte
C location.
C
C Input
C *****
C Source      C*(*)     String to move
C Destination Bytes     Destination location
C Size       I4         Number of characters to move
C
C
C
C *****
C Subroutine Suptrbl (Chain, Lchain)
C *****
C
C This routine returns useful length of the input string after
C removal of blank or non-printable ending characters
C
C Input
C *****
C Chain      C*(*)     String to proceed
C
C Output
C *****
C Lchain     I4         Useful length of the input string

```

CC

7.7. Display Routines

```

C *****
C Integer Function WDF_DisplayDimensions ()
C *****
C
C Utility routine to display Dimensions Infos (name and values) on
C current Output Unit (See WDF_SetloUnit)
C
C
C *****
C Integer Function WDF_DisplayGatts ()
C *****
C
C Utility routine to display Global Attributes Infos (name and
C values) on current Output Unit (See WDF_SetloUnit)
C
C
C *****
C Integer Function WDF_DisplayVnames ()
C *****
C
C Utility routine to display Variables Infos (name and dimensions
C values) on current Output Unit (See WDF_SetloUnit)
C
C
C *****

```