



Elsys AG
Mellingerstrasse 12
CH – 5443 Niederrohrdorf
www.elsys.ch

TPC5 and TPS5 File Specification

(TransAS 3)

1.5

Change Log:

Version 1.5 08.10.2012	Minor changes
Version 1.4 18.01.2011	New Attribute: "deviceName" in „channels“ group. Organization of calculated curves in TPC5 Organization of spectral curves (TPS5-Format)

Contents

1	Basics	3
2	File organization	4
2.1	Measured data	4
2.1.1	Root group	4
2.1.2	Group: measurements	4
2.1.3	Group: channels	4
2.1.4	Group: blocks	5
2.1.5	DataSets	5
2.2	Calculated Data	6
2.2.1	Group: channels	6
2.3	Spectrum files (TPS5)	7
2.3.1	Group: channels	7
2.3.2	Group: blocks	7
2.3.3	DataSets	8
3	Interpretation of measured values	9
3.1	Scaling measured values	9
3.2	Signal- / marker-bits allocation	9
4	Additional Measurement Information	9
4.1	Dataset "autosequence"	9
4.2	Dataset "formulaeditor"	9
4.3	Dataset "hw-settings"	9
4.4	Dataset "layout"	9
4.5	DataSet "settings"	9
5	Programming	11
5.1	Using the library	11
6	Utilities	12

1 Basics

The TPC5 files are based on the HDF5 file format which was developed at the University of Illinois. The HDF5 specifications and the libraries source codes are open.

See also : <http://www.hdfgroup.org/HDF5/index.html>

An HDF5 file is composed of groups and datasets to store and organize signal traces. Recording information such as the recording time is saved as attributes which then can be added to the groups.

To ensure an accelerated display view even with large recordings in TransAS, additional signal traces with reduced data sets are included in the file.

Since TransAS version 3.1.5.46 the following settings can be included; the hardware settings, layout configuration, formulas and the used auto sequence

HDF5 provides also to possibility to compress the data with the gzip compression algorithm. This option can be selected in TransAS.

Further information to the here listed expressions can be found in the TpcAccess API documentation.

2 File organization

2.1 Measured data

Image 1 shows the internal file organization of a TPC5 file. The folder symbols represent HDF5 groups, the table symbols represent HDF5 datasets. Folders which are marked with an 'A' contain attributes. Folders with numeric names represent a measurement, a channel or a block. They are all numbered starting from 1.

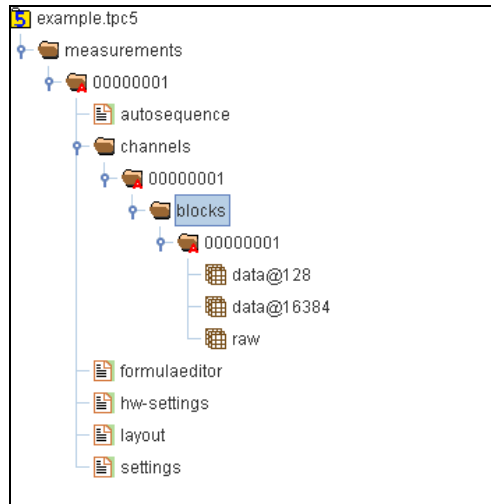


Image 1 - File Organisation

2.1.1 Root group

The top level in a TPC5 file is the root group `/'`. You are also able to set attributes for the root group.

Attribute name	Attribute value	HDF5 File type
filetype	TransAsData	H5T_C_S1, H5T_VARIABLE
format	1	H5T_STD_I32LE
compatible-format	1	H5T_STD_I32LE
Compression	X	H5T_STD_I32LE
creator	TransAS xxxxxx	H5T_C_S1, H5T_VARIABLE

- *format*: Defines the file format version
- *compatible-format*: Defines to which version the actual format is compatible
- The value of the attribute *creator* can be chosen freely

2.1.2 Group: measurements

This group represents a measurement. At the moment it is not possible to store several measurements into one file, therefore only one subgroup of 'measurements' with the name '00000001' is displayed. Following attributes can be set for this subgroup.

Attribute name	Attribute value	HDF5 File type
Name	M1	H5T_C_S1, H5T_VARIABLE

2.1.3 Group: channels

In this group, a channel group (in the form 'xxxxxxx') will be created for each available channel. The channel names are numbered starting from 1. If not all channels are stored, the numbering is nevertheless gapless. If there is more than one board, the numbering starts at board A and continues along the other boards.

Attribute name	Attribute value	HDF5 File type
analogMask		H5T_STD_I32LE
markerMask		H5T_STD_I32LE

binToVoltFactor		H5T_IEEE_F64LE
binToVoltConstant		H5T_IEEE_F64LE
name	Custom channel name	H5T_C_S1, H5T_VARIABLE
deviceName	Device Name	H5T_C_S1, H5T_VARIABLE
physicalUnit	X	H5T_C_S1, H5T_VARIABLE
voltToPhysicalConstant		H5T_IEEE_F64LE
voltToPhysicalFactor		H5T_IEEE_F64LE
markerNames	Name1;Name2	H5T_C_S1, H5T_VARIABLE
rangeMin	X	H5T_IEEE_F64LE
rangeMax	X	H5T_IEEE_F64LE
ChannelName	Physical channel name	H5T_STD_I32LE
ChannelType	None, FFT, AVG	H5T_C_S1, H5T_VARIABLE
UniqueInputID	Reserved	H5T_C_S1, H5T_VARIABLE
color	ARGB	H5T_STD_I32LE

- rangeMin, rangeMax: Min. and Max. Values of the input amplifier (scaled to physical units)

For further information about the usage of these attributes see : 3.1 Scaling measurement values

2.1.4 Group: blocks

In this group, a block group (in the form 'xxxxxxx') will be created for each available block of the measurement. The block group names are numbered starting from 1. If not all blocks from a measurement are stored, the numbering is nevertheless gapless. There is always at least one block available (scope mode, continuous mode).

The following attributes can be set for the block groups.

Attribute name	Attribute value	HDF5 File type
sampleRateHertz		H5T_IEEE_F64LE
startTime	YYYY-MM-DDThh:mm:ss.pppppppp	H5T_C_S1,H5T_VARIABLE
triggerSample		H5T_STD_I64LE
triggerTimeSeconds		H5T_IEEE_F64LE
relativeDivisor		H5T_STD_I32LE

- *startTime* contains the time when the measurement was started in TransAS.
- *triggerSample* represents the number of samples which have been recorded due to a trigger event.
- *triggerTimeSeconds* stores the time period in s between the measurement start and the trigger event.
- *relativeDivisor*: data compression factor between the single datasets.

2.1.5 DataSets

The structure and format of the datasets are depending on whether the data source is from a physical channel or calculated by the Formula Editor. Physical data are stored in the format UINT16 including the marker bits. Calculated curves are stored in the 32-bit floating-point format and are scaled in the physical unit.

Using the dataset ,raw', the ADC measured data will be stored as UINT16 format. In the dataset 'data@xxxx' the envelopes are stored at different data compression rates. The envelope is computed from X samples of the minimal and maximal values. These value pairs are stored in the dataset. The value after the @ sign denotes the number of samples from which the min/max value pair is calculated. The number of value pairs corresponds therefore to the blocksize divided by X. Depending on the block size, more or less envelope datasets will be generated. X is multiplied by 128 until the number of samples equals 1. This doesn't apply to the continuous and ECR mode recordings. Instead, the last min/max pair is calculated from the remaining samples which occur at a division by 128 (and multiples thereof).

Example – Physical channel data:

Data Set Name	Description	Size
data@128	Min and Max values from 128 samples	2x number of samples / 128

data@16384	Min and Max values from 16384 samples	2x number of samples / 16384
data@2097152	Min and Max values from 2 ²¹ samples	2x number of samples / 2 ²¹
data@268435456	Min and Max values from 2 ²⁸ samples	2x number of samples / 2 ²⁸
raw	Raw files as 16 bit unsigned integer	Number of samples

Example – Calculated trace data:

Data Set Name	Description	Size
data	Calculated values in volt	Number of samples
data@128	Min and Max values from 128 samples	2x number of samples / 128
data@16384	Min and Max values from 16384 samples	2x number of samples / 16384

2.2 Calculated Data

Calculated curves from the TransAS formula editor are stored in the TPC5 file slightly different than measured data. Only the difference to the file structure explained above is mentioned here.

Image 2 shows the data structure of a calculated curve. In the opposite of a measured curve, there is no raw data. All data sets are already in 32-bit floating-point format and scaled to the physical unit.

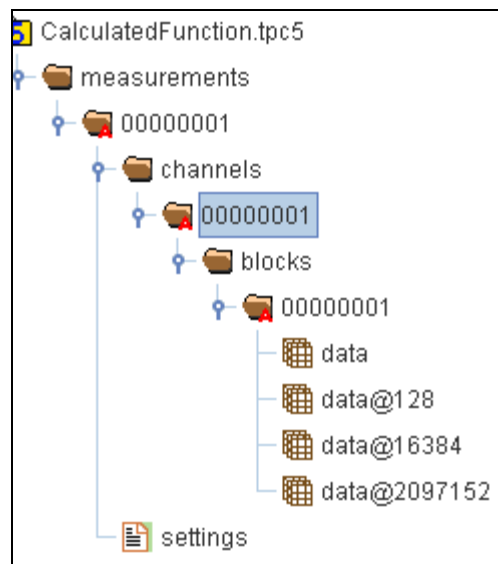


Image 2

2.2.1 Group: channels

The following attributes can be set for the group channels.

Attribute Name	Attribute Value	HDF5 Datatype
name	Channel name	H5T_C_S1, H5T_VARIABLE
deviceName	Device Name	H5T_C_S1, H5T_VARIABLE
physicalUnit		H5T_C_S1, H5T_VARIABLE
rangeMin	x	H5T_IEEE_F64LE
rangeMax	x	H5T_IEEE_F64LE

- Calculated curves are already scaled to physical unit. Therefore there is no binToVoltFactor, binToVoltConstant, voltToPhysicalFactor and voltToPhysicalConstant attribute.

2.3 Spectrum files (TPS5)

In TransAS, it is possible to save a calculated spectrum in a TPS5 file. The TPS5 is also a HDF5 file type and provide almost the same structure then TPC5 files do. Image 3 shows the file structure of a TPS5 file. Down to the “channel” level, the structure is the same as in TPC5 file. The “channel” attributes, the “block” level and the datasets are different.

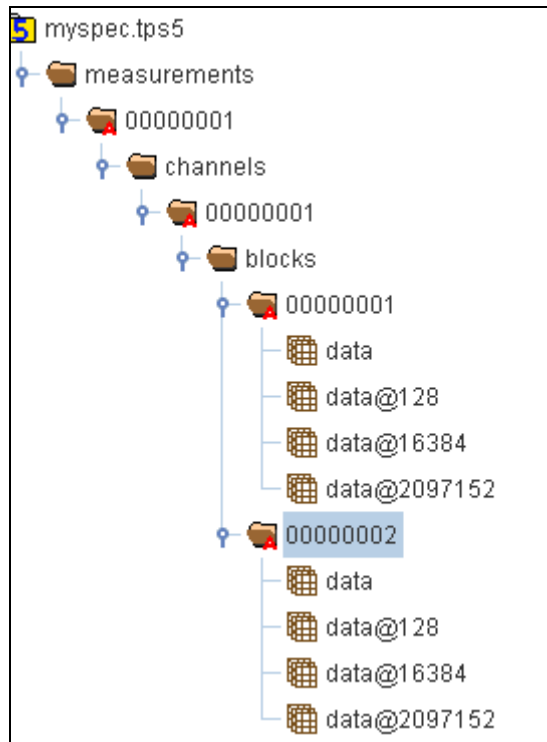


Image 3

2.3.1 Group: channels

In this group, a channel group (in the form 'xxxxxxx') will be created for each available channel. The channel names are numbered starting from 1. If not all channels are stored, the numbering is nevertheless gapless. If there is more than one board, the numbering starts at board A and continues along the other boards.

Attribute Name	Attribute Value	HDF5 Datatype
name	Custom channel Name	H5T_C_S1, H5T_VARIABLE
ChannelName	Physical channel name	H5T_STD_I32LE
ChannelType	None, FFT, AVG	H5T_C_S1, H5T_VARIABLE
UniqueInputID	Reserved	H5T_C_S1, H5T_VARIABLE
color	ARGB	H5T_STD_I32LE
rangeMin	x	H5T_IEEE_F64LE
rangeMax	x	H5T_IEEE_F64LE
physicalUnit	X	H5T_C_S1, H5T_VARIABLE
deviceName	Device Name	H5T_C_S1, H5T_VARIABLE
markerNames	Name1;Name2	H5T_C_S1, H5T_VARIABLE

2.3.2 Group: blocks

The name “blocks” is taken from the TPC5 file structure but has nothing to do with the blocks as we know from a multiblock measurement. This level is just there for having the same structure depth as in TPC5 file.

In this group, for each curve a group named “00000001” and “00000002” is generated. The first group contains the magnitude datasets, the second the phase datasets from the spectrum.

For each block group the following attributes are set:

Attribute Name	Attribute Value	HDF5 Type
sampleRateHertz		H5T_IEEE_F64LE
startTime	YYYY-MM-DDThh:mm:ss.pppppppp	H5T_C_S1, H5T_VARIABLE
stopTriggerSample		H5T_STD_I64LE
timeBaseSource	Internal/External	H5T_C_S1, H5T_VARIABLE
triggerSample		H5T_STD_I64LE
triggerTimeSeconds		H5T_IEEE_F64LE
relativeDivisor		H5T_STD_I32LE

- *startTime* contains the time when the measurement was started in TransAS.
- *triggerSample* represents the number of samples which have been recorded due to a trigger event.
- *triggerTimeSeconds* stores the time period in s between the measurement start and the trigger event.
- *relativeDivisor*: data compression factor between the single datasets.

2.3.3 DataSets

Datasets of spectrum curves are saved like calculated curves in TPC5 as 32-bit floating-point numbers. There are also envelope datasets of the magnitude and phase curves.

3 Interpretation of measured values

3.1 Scaling measured values

The measured values are saved as binary word format. These values can be scaled to volt with the following formula:

$$\text{Signal [V]} = \text{Signal [binary]} * \text{binToVoltFactor} + \text{binToVoltConstant}$$

To scale the Volt values to physical values use following formula:

$$\text{Signal [phys.]} = \text{Signal [V]} * \text{voltToPhysicalFactor} + \text{voltToPhysicalConstant}$$

3.2 Signal- / marker-bits allocation

Depending on the AD converter, only a part of the 16 bit word is used for the analog signal. The remaining bits are used as markers. The signal bits are always left-aligned in a 16 bit word.

The marker bit 1 corresponds to the least-significant bit. How many and which bits are used by the analog signal can be read out from the attributes *analogmask* or *markermask*. Applying an AND disjunction to the binary value with the *markermask* results in the digital marker signals. Applying an AND disjunction to the binary value with the *analogmask* results in the analog signal.

4 Additional Measurement Information

4.1 Dataset “autosequence”

This dataset stores the used auto sequence as simple text. If no auto sequence is used this dataset is empty.

4.2 Dataset “formulaeditor”

This dataset stores the used formulas as simple text. If no formulas are used the dataset is empty.

4.3 Dataset “hw-settings”

This dataset stores the hardware settings from the control panel as XML.

4.4 Dataset “layout”

This dataset stores the layout settings from TransAS as XML.

4.5 DataSet “settings”

This dataset stores the recording attributes and recording logs which are generated by TransAS, as XML structure. The following example shows such a XML structure with 2 recording logs and 2 attributes.

```
<?xml version="1.0" encoding="utf-16"?>
<Tpc5MeasurementInternalData xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
xmlns:xsd="http://www.w3.org/2001/XMLSchema" Version="1.0"
xmlns="http://www.elsys.ch/schema/transas/tpc5filedata-1.x.xsd">
  <RecordingLog Origin="">
    <RecordingLogEntries>
      <RecordingLogEntry>
        <Time>2008-09-25T17:11:36</Time>
        <Text>RecordLog1</Text>
      </RecordingLogEntry>
      <RecordingLogEntry>
        <Time>2008-09-25T17:11:36.05</Time>
        <Text>RecordLog1</Text>
      </RecordingLogEntry>
    </RecordingLogEntries>
    <RecordingLogAttributes>
      <RecordingLogAttribute>
```

```
    <Name>ExampleAttribute1</Name>
    <Text>This is the example attribute number one.</Text>
  </RecordingLogAttribute>
  <RecordingLogAttribute>
    <Name>ExampleAttribute2</Name>
    <Text>This is just another example attribute.</Text>
  </RecordingLogAttribute>
</RecordingLogAttributes>
</RecordingLog>
</Tpc5MeasurementInternalData>
```

5 Programming

There are several libraries from the above given webpage available to write or read HDF5/TPC5 files. You may create statistic or dynamic libraries.

The High Level (HL) library from the webpage can not be used because not all options are implemented.

5.1 Using the library

Function	C API
Create file	<pre> hid_t plist = H5Pcreate(H5P_FILE_CREATE); hid_t aplist = H5Pcreate(H5P_FILE_ACCESS); hid_t hdfFile = H5Fcreate(„Filename.tpc5“,H5F_ACC_TRUNC,plist,aplist); H5Pclose(plist); H5Pclose(aplist); </pre>
Create group	<pre> hid_t linkcreatprop,groupcreatprop,groupaccessprop; linkcreatprop = H5Pcreate(H5P_LINK_CREATE); groupcreatprop = H5Pcreate(H5P_GROUP_CREATE); groupaccessprop = H5Pcreate(H5P_GROUP_ACCESS); hid_t Group = H5Gcreate2(Id,„GroupName“,linkcreatprop,groupcreatprop,groupaccessprop); H5Pclose(linkcreatprop); H5Pclose(groupcreatprop); H5Pclose(groupaccessprop); </pre>
Set attributes	<pre> // String Attribut hid_t dataspace = H5Screate(H5S_SCALAR); hid_t typeId = H5Tcopy(H5T_C_S1); H5Tset_size(typeId,H5T_VARIABLE); hid_t attrId = H5Acreate2(Id,keyString,typeId,dataspace,0,0); H5Awrite(attrId,typeId,&Value); H5Tclose(typeId); H5Aclose(attrId); H5Sclose(dataspace); // INT32 Attribut hid_t dataspace = H5Screate(H5S_SCALAR); hid_t attrId = H5Acreate2(Id,keyString,H5T_NATIVE_INT,dataspace,0,0); H5Awrite(attrId,H5T_NATIVE_UINT,&i32value); H5Aclose(attrId); H5Sclose(dataspace); //Float Attribut hid_t dataspace = H5Screate(H5S_SCALAR); hid_t attrId = H5Acreate2(Id,keyString,H5T_NATIVE_DOUBLE,dataspace,0,0); H5Awrite(attrId,H5T_NATIVE_DOUBLE,&dbvalue); H5Aclose(attrId); H5Sclose(dataspace); //INT64 Attribut hid_t dataspace = H5Screate(H5S_SCALAR); hid_t attrId = H5Acreate2(Id,keyString,H5T_NATIVE_INT64,dataspace,0,0); H5Awrite(attrId,H5T_NATIVE_UINT64,&i64value); H5Aclose(attrId); H5Sclose(dataspace); </pre>
Create dataset	<pre> hid_t cparms,linkprop,accessprop; herr_t status; hsize_t dims[1]; dims[0] = 0; hsize_t maxdims[1] = {H5S_UNLIMITED}; hsize_t chunk_dims[1] = {0x400}; hid_t dataspace; /* Create the data space with unlimited dimensions. */ dataspace = H5Screate_simple(1, dims, maxdims); /* Modify dataset creation properties, i.e. enable chunking */ cparms = H5Pcreate(H5P_DATASET_CREATE); status = H5Pset_chunk(cparms,1,chunk_dims); /* Create Link Creation Property List */ linkprop = H5Pcreate(H5P_LINK_CREATE); </pre>

	<pre> /* Create Dataset Access Property List */ accessprop = H5Pcreate (H5P_DATASET_ACCESS); /* Create a new dataset within the file using cparms creation properties. */ hid_t DataSetID = H5Dcreate2 (GroupID,"raw", H5T_NATIVE_UINT16, dataspace, linkprop, cparms, accessprop); H5Pclose(accessprop); H5Pclose(linkprop); H5Pclose(cpargs); H5Sclose(dataspace); </pre>
Write dataset	<pre> hid_t filespace,dataspace; hsize_t inputdims[1],size[1]; herr_t status; size[0] = uiNbOfSamples; inputdims[0] = uiNbOfSamples ; status = H5Dset_extent(DataSetID, size); /* Select a hyperslab */ filespace = H5Dget_space (DataSetID); hsize_t offset[1]; offset[0] = m_i64Size; status = H5Sselect_hyperslab (filespace, H5S_SELECT_SET, offset, NULL, inputdims, NULL); /* Define memory space */ dataspace = H5Screate_simple (1,inputdims , NULL); /* Write the data to the hyperslab */ status = H5Dwrite (m_DataSetID, H5T_NATIVE_UINT16, dataspace, filespace, H5P_DEFAULT, data); H5Sclose(dataspace); H5Sclose(filespace); </pre>

Note: The close file and close groupe commands are not listed here but are absolutely necessary. If the close commands are not included, the corresponding file will be locked and the allocated memory of the library will not be freed.

6 Utilities

With the ‚HDFView’ tool from the webpage above you can open TPC5 files.

