



METAS VNA Tools - Data Formats V2.7.0

Michael Wollensack

March 2023

Contents

1	Introduction	4
1.1	S-Parameter Data	4
1.2	VNA Data	5
1.3	Data Collections	5
1.3.1	S-Parameter Data Collection	5
1.3.2	VNA Data Collection	6
1.4	Overview	6
2	sdatb File Specification	7
2.1	Binary Structure Version 1	7
2.1.1	Uncertainty Numbers	7
2.2	Binary Structure Version 2	8
2.2.1	Flat Vector of Uncertainty Numbers	8
2.3	Binary Structure Version 3	9
2.3.1	VNA Port Description	9
2.4	Binary Structure Version 4	10
2.4.1	Frequency Conversion Subset	10
2.5	Binary Structure Version 5	10
2.5.1	Frequency Conversion	11
2.6	Example MATLAB Code	11
3	sdatx File Specification	15
3.1	XML Schema	15
3.1.1	Frequency and Port Lists	15
3.1.2	Uncertainty Numbers	17



METAS VNA Tools - Data Formats V2.7.0

4	sdatcv File Specification	19
4.1	Header Lines	19
4.1.1	Port Assignment	19
4.1.2	Reference Impedance	19
4.1.3	Data Column Description	19
4.2	Data Lines	20
4.2.1	1-Port Example	20
4.2.2	2-Port Examples	21
4.3	Comment Lines	22
5	Touchstone V1.x snp File Specification	23
5.1	Examples	23
6	Touchstone V2.0 ts File Specification	24
6.1	Examples	24
7	vdatb File Specification	25
7.1	Binary Structure Version 1	25
7.1.1	VNA Parameter Data	25
7.1.2	VNA Parameter	25
7.1.3	VNA Parameter Examples	26
7.2	Binary Structure Version 2	26
7.3	Binary Structure Version 3	27
7.4	Binary Structure Version 4	27
7.5	Binary Structure Version 5	28
7.6	Example MATLAB Code	28
8	vdatx File Specification	32
8.1	XML Schema	32
8.1.1	VNA Parameter Description	32
9	vdatcv File Specification	34
9.1	Header Lines	34
9.1.1	Port Assignment	34
9.1.2	Reference Impedance	34
9.1.3	Data Column Description	34
9.2	Data Lines	35
9.2.1	Example	35
9.3	Comment Lines	36
10	CITI File Specification	37
10.1	Examples	37
11	scolb File Specification	39



METAS VNA Tools - Data Formats V2.7.0

12 scolcv File Specification	39
12.1 Header Lines	39
12.1.1 Number	40
12.1.2 Name	40
12.1.3 Port Assignment	40
12.1.4 Reference Impedance	40
12.1.5 Data Column Description	40
12.2 Data Lines	40
12.2.1 Example	40
12.3 Comment Lines	41
13 vcolb File Specification	42
14 vcolcv File Specification	42
14.1 Header Lines	42
14.1.1 Number	43
14.1.2 Name	43
14.1.3 Port Assignment	43
14.1.4 Reference Impedance	43
14.1.5 Data Column Description	43
14.2 Data Lines	43
14.2.1 Example	44
14.3 Comment Lines	45
A ZIP File with Embedded Data Files	46
B PDF File with Embedded Data Files	46
B.1 Example	46
C PTB DCC XML File with Embedded Data Files	47



METAS VNA Tools - Data Formats V2.7.0

1 Introduction

Touchstone and CITI are two well known file formats for storing S-parameters, but both of them don't support uncertainties with dependencies. Uncertainties with dependencies are a method with low memory consumption for taking correlations into account, see METAS UncLib. The solution is to define new data formats and file types which support uncertainties with dependencies.

The in the following described data and file formats were developed for VNA Tools. Including uncertainties increases the file size drastically. Several file formats which include compression (ZIP) are proposed for this reason.

A file format is a mapping of a data format. Thus the two data formats which are used in VNA Tools are described first. The following sections contain descriptions of the files formats which are derived from the data formats.

1.1 S-Parameter Data

S-Parameter Data is a data format. Most of the calculations in VNA Tools are done with the S-Parameter Data type. It is the main data format in VNA Tools. S-Parameter Data contains the following properties:

- Frequency List (1d array of double)
- Port Assignment (1d array of VnaPortDescription)
- Port Impedance (1d array of ComplexUncNumber) ¹
- Frequency Conversion (1d array of FrequencyConversion) ²
- Data (3d array of ComplexUncNumber)
 - Index 0: Frequency
 - Index 1: Receiver Port
 - Index 2: Source Port

As can be seen from the properties, this data format is well suited for storing S-parameters. S-Parameter Data supports the following file types:

METAS sdatb is a binary file format which contains the full information,

METAS sdatx is an XML file format which contains the same information as sdatb,

METAS sdatcv is an ASCII text file format which contains only a subset of the information (no correlation between frequency points and different sdatcv files),

Touchstone snp, ts is an ASCII text file format which doesn't contain uncertainty information.



METAS VNA Tools - Data Formats V2.7.0

1.2 VNA Data

VNA Data is another data format in VNA Tools. It is used in the visualization part of VNA Tools to display receiver values and to be compatible with old file formats. VNA Data contains the following properties:

- Frequency List (1d array of double)
- Port Assignment (1d array of VnaPortDescription)
- Port Impedance (1d array of ComplexUncNumber) ¹
- Frequency Conversion (1d array of FrequencyConversion) ²
- Parameter Data (1d array of VnaParameterData)
 - Parameter (VnaParameter)
 - Data (1d array of ComplexUncNumber)

As can be seen from the properties, this data format is suited for storing arbitrary receiver ratios as well as receiver values. It's more general than S-Parameter Data. VNA Data supports the following file types:

METAS vdatb is a binary file format which contains the full information,

METAS vdatx is an XML file format which contains the same information as vdatb,

METAS vdatcv is an ASCII text file format which contains only a subset of the information (no correlation between frequency points and different files),

CITI is an ASCII text file format which contains only a subset of the information (no correlation and no port impedance).

1.3 Data Collections

A single Data Collection file can contain either multiple S-Parameter Data files or multiple VNA Data files.

1.3.1 S-Parameter Data Collection

S-Parameter Data Collection supports the following file types:

METAS scolb is a zip file format which contains multiple sdatb files,

METAS scolcv is an ASCII text file format which contains multiple standards (correlation between different standards in the same scolcv file) but only a subset of the information (no correlation between frequency points and different scolcv files).

¹The port impedance is the complex reference impedance. It can be different for each port but not for each frequency. Complex reference impedance in function of frequency needs to be re-normalized. For changing the complex reference impedance see the appendix 'Transmission Line Junction' of the METAS VNA Tools - Math Reference.

²The frequency conversion property is optional. It can be different for each port.



METAS VNA Tools - Data Formats V2.7.0

1.3.2 VNA Data Collection

VNA Data Collection supports the following file types:

METAS vcolb is a zip file format which contains multiple vdatb files,

METAS vcolcv is an ASCII text file format which contains multiple standards (correlation between different standards in the same vcolcv file) but only a subset of the information (no correlation between frequency points and different vcolcv files).

1.4 Overview

Table 1 shows an overview of the different file formats. Each file format can be converted into

Table 1: List of file formats

Format	Type	Port Assignment	Port Impedance	S-Parameter	Mixed-Mode P.	Freq. Conversion	Receiver Values	Arbitrary Ratios	Collection	Uncertainties	Correlation	Dependencies	ZIP Compression	No Redundancy	File Size	Speed
METAS sdatb V1	binary	☺	☺	☺						☺	☺	☺	☺		☺	☺
METAS sdatb V2	binary	☺	☺	☺						☺	☺	☺		☺	☺	☺☺
METAS sdatb V3	binary	☺	☺	☺	☺					☺	☺	☺		☺	☺	☺☺
METAS sdatb V4	binary	☺	☺	☺	☺	☺				☺	☺	☺		☺	☺	☺☺
METAS sdatb V5	binary	☺	☺	☺	☺	☺				☺	☺	☺		☺	☺	☺☺
METAS sdatx	xml	☺	☺	☺	☺	☺				☺	☺	☺	☺		☺	☺☺
METAS sdatcv	text	☺	☺	☺	☺					☺	☺	☺			☺	☺
Touchstone V1.x snp	text	☹	☺	☺						☹	☹	☹			☺	☺
Touchstone V2.0 ts	text	☺	☺	☺	☺					☹	☹	☹			☺	☺
METAS vdatb V1	binary	☺	☺	☺			☺	☺		☺	☺	☺	☺		☺	☺
METAS vdatb V2	binary	☺	☺	☺			☺	☺		☺	☺	☺		☺	☺	☺☺
METAS vdatb V3	binary	☺	☺	☺	☺		☺	☺		☺	☺	☺		☺	☺	☺☺
METAS vdatb V4	binary	☺	☺	☺	☺	☺	☺	☺		☺	☺	☺		☺	☺	☺☺
METAS vdatb V5	binary	☺	☺	☺	☺	☺	☺	☺		☺	☺	☺		☺	☺	☺☺
METAS vdatx	xml	☺	☺	☺	☺	☺	☺	☺		☺	☺	☺	☺		☺	☺☺
METAS vdatcv	text	☺	☺	☺	☺		☺	☺		☺	☺	☺			☺	☺
CITI	text	☺	☹	☺			☺	☺		☺	☹	☹			☺	☺
METAS scolb	zip	☺	☺	☺	☺	☺			☺	☺	☺	☺	☺		☺	☺
METAS scolcv	text	☺	☺	☺	☺				☺	☺	☺	☺			☺	☺
METAS vcolb	zip	☺	☺	☺	☺	☺	☺	☺	☺	☺	☺	☺	☺		☺	☺
METAS vcolcv	text	☺	☺	☺	☺		☺	☺	☺	☺	☺	☺			☺	☺

another from this table.



2 sdatb File Specification

The sdatb-file format is a binary file type developed by METAS. The file can be written in a GZIP file stream to reduce the file size. The byte ordering is little-endian.

2.1 Binary Structure Version 1

Version 1 of sdatb uses a GZIP file stream to reduce the file size. The following enumeration describes the binary structure of a sdatb-file:

1. Header (string), value: '%SDATA'
2. Version (int32), value: 1
3. Number of Frequencies (int32)
4. Number of Ports (int32)
5. Frequency List (double[]), size: number of frequencies
6. Ports (int32[]), size: number of ports
7. Port Impedance (ComplexUncNumber[]), size: number of ports
8. Data (ComplexUncNumber[, ,]), size dim 0: number of frequencies, size dim 1 and 2: number of ports.

2.1.1 Uncertainty Numbers

The following enumeration describes the binary structure of 'ComplexUncNumber':

1. Version (int32), value: 1
2. Real (UncNumber)
3. Imag (UncNumber)

The following enumeration describes the binary structure of 'UncNumber':

1. Version (int32), value: 1
2. Value (double)
3. Version2 (int32), value: 4
4. Number of Dependencies (int32)
5. Dependencies (DependsOn[]), size: number of dependencies

The following enumeration describes the binary structure of 'DependsOn':

1. Number of Id Bytes (int32)
2. Input Id (byte[]), size: number of id bytes



METAS VNA Tools - Data Formats V2.7.0

3. Input Description (string)
4. Input IDof (double)
5. Jacobi (double)

2.2 Binary Structure Version 2

Version 2 of sdatb is not using a GZIP file stream. The redundancy of the data is removed by storing uncertainty inputs in a look up table. The uncompressed file size of version 2 is comparable to the GZIP compressed file size of version 1 and about four times smaller than the uncompressed file size of version 1. Avoiding the GZIP step and reducing the uncompressed file size speeds up loading and saving of files. The following enumeration describes the binary structure of a sdatb-file:

1. Header (string), value: '%SDATA'
2. Version (int32), value: 2
3. Number of Frequencies (int32)
4. Number of Ports (int32)
5. Frequency List (double[]), size: number of frequencies
6. Ports (int32[]), size: number of ports
7. Flat Vector (UncNumber[]), size: $2N_{Ports} + 2N_{Freq}N_{Ports}N_{Ports}$

The last element contains the port impedance and the data mapped to a flat vector of uncertainty numbers.

2.2.1 Flat Vector of Uncertainty Numbers

The following enumeration describes the binary structure of 'FlatVectorUncNumbers':

1. Version (int32), value: 1
2. Length (7-bit encoded int)
3. Values (double[]), size: length
4. Number of Inputs (7-bit encoded int)
5. Inputs (UnclInput[]), size: number of inputs
6. Dependencies (UncDependencies[]), size: length

The following enumeration describes the binary structure of 'UnclInput':

1. Temp (byte), bit 0: same id size, bit 1: empty description, bit 2: zero idof, bit 3-7: 0
2. Id Size (7-bit encoded int), field only present if not same id size



METAS VNA Tools - Data Formats V2.7.0

3. Input Id (byte []), size: id size
4. Input Description (string), field only present if not empty description
5. Input IDof (double), field only present if not zero idof

The following enumeration describes the binary structure of 'UncDependencies'

1. Number of Dependencies (7-bit encoded int)
2. Dependencies (UncDependency[]), size: number of dependencies, pointer to inputs set to 0.

The following enumeration describes the binary structure of 'UncDependency'

1. Relative Pointer to Inputs (7-bit encoded int)
2. Jacobi (double)

2.3 Binary Structure Version 3

Version 3 of sdatb is an extension to version 2 which adds support for mixed-mode S-parameters and port indices. The following enumeration describes the binary structure of a sdatb-file:

1. Header (string), value: '%SDATA'
2. Version (int32), value: 3
3. Number of Frequencies (int32)
4. Number of Ports (int32)
5. Frequency List (double[]), size: number of frequencies
6. Ports (VnaPortDescription[]), size: number of ports
7. Flat Vector (UncNumber[]), size: $2N_{Ports} + 2N_{Freq}N_{Ports}N_{Ports}$

The last element contains the port impedance and the data mapped to a flat vector of uncertainty numbers.

For the definition of 'Flat Vector of Uncertainty Numbers' see section 2.2.1.

2.3.1 VNA Port Description

The following enumeration describes the binary structure of 'VnaPortDescription':

1. Port (int32)
2. Mode (VnaPortMode)
3. Index (int16)

'VnaPortMode' is an enumeration represented by a 16-bit integer where 0 is single-ended 's', 1 is differential mode 'd' and 2 is common mode 'c'.



METAS VNA Tools - Data Formats V2.7.0

2.4 Binary Structure Version 4

Version 4 of sdatb is an extension to version 3 which adds support for frequency converting S-parameters. The following enumeration describes the binary structure of a sdatb-file:

1. Header (string), value: '%SDATA'
2. Version (int32), value: 4
3. Number of Frequencies (int32)
4. Number of Ports (int32)
5. Frequency List (double[]), size: number of frequencies
6. Ports (VnaPortDescription[]), size: number of ports
7. Frequency Conversion List (FrequencyConversionSub[]), size: number of ports
8. Flat Vector (UncNumber[]), size: $2N_{Ports} + 2N_{Freq}N_{Ports}N_{Ports}$

The last element contains the port impedance and the data mapped to a flat vector of uncertainty numbers.

For the definition of 'VnaPortDescription' see section 2.3.1. For the definition of 'Flat Vector of Uncertainty Numbers' see section 2.2.1.

2.4.1 Frequency Conversion Subset

The following enumeration describes the binary structure of 'FrequencyConversionSub':

1. Numerator (double)
2. Denominator (double)
3. Offset (double)

2.5 Binary Structure Version 5

Version 5 of sdatb is an extension to version 4 which adds support for frequency converting S-parameters, where the receiver frequency is not equal to the source frequency. The following enumeration describes the binary structure of a sdatb-file:

1. Header (string), value: '%SDATA'
2. Version (int32), value: 5
3. Number of Frequencies (int32)
4. Number of Ports (int32)
5. Frequency List (double[]), size: number of frequencies
6. Ports (VnaPortDescription[]), size: number of ports



METAS VNA Tools - Data Formats V2.7.0

7. Frequency Conversion List (FrequencyConversion[]), size: number of ports
8. Flat Vector (UncNumber[]), size: $2N_{Ports} + 2N_{Freq}N_{Ports}N_{Ports}$

The last element contains the port impedance and the data mapped to a flat vector of uncertainty numbers.

For the definition of 'VnaPortDescription' see section 2.3.1. For the definition of 'Flat Vector of Uncertainty Numbers' see section 2.2.1.

2.5.1 Frequency Conversion

The following enumeration describes the binary structure of 'FrequencyConversion':

1. Test Receiver (FrequencyConversionSub)
2. Reference Receiver (FrequencyConversionSub)
3. Source (FrequencyConversionSub)

For the definition of 'FrequencyConversionSub' see section 2.4.1.

2.6 Example MATLAB Code

The following code shows how to load a sdatb-file in MATLAB:

```
1 function d = LoadSDATB(filepath)
2 % Loads VNA Tools II SParamData (*.sdatb) file
3 % Michael Wollensack METAS - 22.04.2022
4
5 d = {};
6 f3 = OpenFile(filepath);
7 % Type
8 type = char(f3.ReadString());
9 % Version
10 version = int32(f3.ReadInt32());
11 disp(['Type: ' type ', Version: ' num2str(version)])
12 if (strcmp(type, '%SDATA') && 1 <= version && version <= 5)
13     % Number of Frequencies
14     nfreq = f3.ReadInt32();
15     % Number of Ports
16     nports = f3.ReadInt32();
17     % Init
18     d.Frequency = zeros(1, nfreq);
19     d.Ports = cell(1, nports);
20     d.PortZr = LinProp(zeros(1, nports));
21     d.FrequencyConversions = cell(1, nports);
22     d.Data = LinProp(zeros([nfreq, nports, nports]));
23     % Frequency (Hz)
24     for i = 1:nfreq
25         d.Frequency(i) = f3.ReadDouble();
26     end
27     % Ports
28     ModeType = {'', 'd', 'c'};
29     IndexType = {'', ':I', ':II', ':III', ':IV', ':V', ':VI', ':VII',
30                 ':VIII', ':IX', ':X', ':XI', ':XII'};
```



METAS VNA Tools - Data Formats V2.7.0

```
30     for i = 1:nports
31         if (version < 3)
32             d.Ports{i} = num2str(f3.ReadInt32());
33         else
34             d.Ports{i} = [num2str(f3.ReadInt32()) ModeType{f3.ReadInt16()}
35                 + 1} IndexType{f3.ReadInt16() + 1}];
36         end
37     end
38     % Frequency Conversions
39     for i = 1:nports
40         if (version < 4)
41             d.FrequencyConversions{i}.TestReceiver.Numerator = 1;
42             d.FrequencyConversions{i}.TestReceiver.Denominator = 1;
43             d.FrequencyConversions{i}.TestReceiver.Offset = 0;
44             d.FrequencyConversions{i}.ReferenceReceiver.Numerator = 1;
45             d.FrequencyConversions{i}.ReferenceReceiver.Denominator = 1;
46             d.FrequencyConversions{i}.ReferenceReceiver.Offset = 0;
47             d.FrequencyConversions{i}.Source.Numerator = 1;
48             d.FrequencyConversions{i}.Source.Denominator = 1;
49             d.FrequencyConversions{i}.Source.Offset = 0;
50         elseif (version == 4)
51             numerator = f3.ReadDouble();
52             denominator = f3.ReadDouble();
53             offset = f3.ReadDouble();
54             d.FrequencyConversions{i}.TestReceiver.Numerator = numerator;
55             d.FrequencyConversions{i}.TestReceiver.Denominator =
56                 denominator;
57             d.FrequencyConversions{i}.TestReceiver.Offset = offset;
58             d.FrequencyConversions{i}.ReferenceReceiver.Numerator =
59                 numerator;
60             d.FrequencyConversions{i}.ReferenceReceiver.Denominator =
61                 denominator;
62             d.FrequencyConversions{i}.ReferenceReceiver.Offset =
63                 offset;
64             d.FrequencyConversions{i}.Source.Numerator = numerator;
65             d.FrequencyConversions{i}.Source.Denominator = denominator;
66             d.FrequencyConversions{i}.Source.Offset = offset;
67         else
68             d.FrequencyConversions{i}.TestReceiver.Numerator =
69                 f3.ReadDouble();
70             d.FrequencyConversions{i}.TestReceiver.Denominator =
71                 f3.ReadDouble();
72             d.FrequencyConversions{i}.TestReceiver.Offset =
73                 f3.ReadDouble();
74             d.FrequencyConversions{i}.ReferenceReceiver.Numerator =
75                 f3.ReadDouble();
76             d.FrequencyConversions{i}.ReferenceReceiver.Denominator =
77                 f3.ReadDouble();
78             d.FrequencyConversions{i}.ReferenceReceiver.Offset =
79                 f3.ReadDouble();
80             d.FrequencyConversions{i}.Source.Numerator = f3.ReadDouble();
81             d.FrequencyConversions{i}.Source.Denominator = f3.ReadDouble();
82             d.FrequencyConversions{i}.Source.Offset = f3.ReadDouble();
83         end
84     end
85     if (version == 1)
86         % Port Zr
87         for i = 1:nports
```



METAS VNA Tools - Data Formats V2.7.0

```
77         d.PortZr(i) = ReadComplexLinProp(f3);
78     end
79     % Data
80     for i1 = 1:nfreq
81         for i2 = 1:nports
82             for i3 = 1:nports
83                 d.Data(i1, i2, i3) = ReadComplexLinProp(f3);
84             end
85         end
86     end
87     elseif (2 <= version || version <= 5)
88         % Flat Vector
89         v = ReadComplexFlatVectorLinProp(f3);
90         index = 1;
91         % PortsZr
92         for i = 1:nports
93             d.PortZr(i) = v(index); index = index + 1;
94         end
95         % Data
96         for i1 = 1:nfreq
97             for i2 = 1:nports
98                 for i3 = 1:nports
99                     d.Data(i1, i2, i3) = v(index); index = index + 1;
100                 end
101             end
102         end
103     end
104 end
105 f3.Close();
106 end
107
108 function f3 = OpenFile(filepath)
109 % Open File
110 NET.addAssembly('System');
111 % File Stream
112 f1 = System.IO.FileStream(filepath, System.IO.FileMode.Open);
113 % Try if Stream is GZIP compressed
114 try
115     f2 = System.IO.Compression.GZipStream(f1,
116         System.IO.Compression.CompressionMode.Decompress);
117     f2.ReadByte();
118     f1.Position = 0;
119     f2 = System.IO.Compression.GZipStream(f1,
120         System.IO.Compression.CompressionMode.Decompress);
121     disp('GZIP compressed file')
122 catch
123     f1.Position = 0;
124     f2 = f1;
125     disp('Uncompressed file')
126 end
127 % Binary Reader
128 f3 = System.IO.BinaryReader(f2);
129 end
130
131 function c = ReadComplexLinProp(f3)
132 % Read ComplexLinProp using METAS UncLib
```



METAS VNA Tools - Data Formats V2.7.0

```
131 n = NET.createGeneric('Metas.UncLib.Core.Complex',  
    {'Metas.UncLib.LinProp.UncNumber'});  
132 n.BinarySetDataFrom(f3);  
133 c = LinProp(n);  
134 end  
135  
136 function v = ReadComplexFlatVectorLinProp(f3)  
137 % Read ComplexFlatVectorLinProp using METAS UncLib  
138 list = Metas.UncLib.LinProp.UncList();  
139 list.BinarySetDataFrom(f3);  
140 n = NET.createGeneric('Metas.UncLib.Core.Ndims.RealNArray',  
    {'Metas.UncLib.LinProp.UncNumber'});  
141 n.Init1dData(list.data);  
142 r = LinProp(n);  
143 v = r(1:2:end-1) + 1i.*r(2:2:end);  
144 end
```



3 sdatx File Specification

The sdatx-file format is an XML file type developed by METAS. It's described using an XML schema. See https://www.w3schools.com/xml/schema_intro.asp for more details about XML schemas. The file can be written in a GZIP file stream to reduce the file size.

3.1 XML Schema

The following listing shows the XML schema for 'SParamData':

```
1 <!-- definition of SParamData -->
2 <xs:element name="SParamData">
3   <xs:complexType>
4     <xs:sequence>
5       <xs:element ref="FrequencyList" />
6       <xs:element ref="PortList" />
7       <xs:element ref="PortZrList" />
8       <xs:element ref="FrequencyConversionList" minOccurs="0"
9         maxOccurs="1" />
10      <xs:element name="Data">
11        <xs:complexType>
12          <!-- Index 0: Frequency -->
13          <xs:sequence>
14            <xs:element maxOccurs="unbounded" name="Frequency">
15              <xs:complexType>
16                <!-- Index 1: Receiver Port -->
17                <xs:sequence>
18                  <xs:element maxOccurs="unbounded" name="ReceiverPort">
19                    <xs:complexType>
20                      <!-- Index 2: Source Port -->
21                      <xs:sequence>
22                        <xs:element maxOccurs="unbounded"
23                          name="SourcePort"
24                          type="ComplexUncNumberType" />
25                      </xs:sequence>
26                    </xs:complexType>
27                  </xs:element>
28                </xs:sequence>
29              </xs:complexType>
30            </xs:element>
31          </xs:sequence>
32        </xs:complexType>
33      </xs:element>
```

3.1.1 Frequency and Port Lists

The following listing shows the XML schema for 'FrequencyList':

```
1 <!-- definition of FrequencyList -->
2 <xs:element name="FrequencyList">
3   <xs:complexType>
4     <xs:sequence>
```



METAS VNA Tools - Data Formats V2.7.0

```
5         <xs:element maxOccurs="unbounded" name="Frequency"
           type="xs:double" />
6     </xs:sequence>
7 </xs:complexType>
8 </xs:element>
```

The following listing shows the XML schema for 'PortList':

```
1 <!-- definition of PortList -->
2 <xs:element name="PortList">
3     <xs:complexType>
4         <xs:sequence>
5             <xs:element maxOccurs="unbounded" name="Port"
                       type="VnaPortDescriptionType" />
6         </xs:sequence>
7     </xs:complexType>
8 </xs:element>
```

The following listing shows the XML schema for 'VnaPortDescriptionType':

```
1 <!-- definition of VnaPortDescriptionType -->
2 <xs:simpleType name="VnaPortDescriptionType">
3     <xs:restriction base="xs:string">
4         <xs:pattern
           value="[0-9]+[sdc]??:?M{0,3}(C[MD]|D?C{0,3})(X[CL]|L?X{0,3})(I[XV]|V?I{0,3})"/>
5     </xs:restriction>
6 </xs:simpleType>
```

The following listing shows the XML schema for 'PortZrList':

```
1 <!-- definition of PortZrList -->
2 <xs:element name="PortZrList">
3     <xs:complexType>
4         <xs:sequence>
5             <xs:element maxOccurs="unbounded" name="PortZr"
                       type="ComplexUncNumberType" />
6         </xs:sequence>
7     </xs:complexType>
8 </xs:element>
```

The following listing shows the XML schema for 'FrequencyConversionList':

```
1 <!-- definition of FrequencyConversionList -->
2 <xs:element name="FrequencyConversionList">
3     <xs:complexType>
4         <xs:sequence>
5             <xs:element maxOccurs="unbounded" name="FrequencyConversion"
                       type="FrequencyConversionType" />
6         </xs:sequence>
7     </xs:complexType>
8 </xs:element>
```

The following listing shows the XML schema for 'FrequencyConversionType':

```
1 <!-- definition of FrequencyConversionType -->
2 <xs:complexType name="FrequencyConversionType">
3     <xs:sequence>
4         <xs:element name="TestReceiver" type="FrequencyConversionSubType" />
5         <xs:element name="ReferenceReceiver"
                       type="FrequencyConversionSubType" />
6         <xs:element name="Source" type="FrequencyConversionSubType" />
7     </xs:sequence>
8 </xs:complexType>
```




METAS VNA Tools - Data Formats V2.7.0

The following listing shows the XML schema for 'FrequencyConversionSubType':

```
1 <!-- definition of FrequencyConversionSubType -->
2 <xs:complexType name="FrequencyConversionSubType">
3   <xs:sequence>
4     <xs:element name="Numerator" type="xs:double" />
5     <xs:element name="Denominator" type="xs:double" />
6     <xs:element name="Offset" type="xs:double" />
7   </xs:sequence>
8 </xs:complexType>
```

3.1.2 Uncertainty Numbers

The following listing shows the XML schema for 'ComplexUncNumberType':

```
1 <!-- definition of ComplexUncNumberType -->
2 <xs:complexType name="ComplexUncNumberType">
3   <xs:sequence>
4     <xs:element name="Real" type="UncNumberType" />
5     <xs:element name="Imag" type="UncNumberType" />
6   </xs:sequence>
7 </xs:complexType>
```

The following listing shows the XML schema for 'UncNumberType':

```
1 <!-- definition of UncNumberType -->
2 <xs:complexType name="UncNumberType">
3   <xs:sequence>
4     <xs:element name="Value" type="xs:double" />
5     <xs:element name="Dependencies">
6       <xs:complexType>
7         <xs:sequence>
8           <xs:element minOccurs="0" maxOccurs="unbounded"
9             name="DependsOn">
10            <xs:complexType>
11              <xs:sequence>
12                <xs:element name="Input">
13                  <xs:complexType>
14                    <xs:sequence>
15                      <xs:element name="Id" type="UncInputIdType" />
16                      <xs:element name="Description" type="xs:string" />
17                      <xs:element name="IDof" type="xs:double" />
18                    </xs:sequence>
19                  </xs:complexType>
20                </xs:element>
21                <xs:element name="Jacobi" type="xs:double" />
22              </xs:sequence>
23            </xs:complexType>
24          </xs:element>
25        </xs:sequence>
26      </xs:complexType>
27    </xs:element>
28  </xs:sequence>
29 </xs:complexType>
```

The following listing shows the XML schema for 'UncInputIdType':

```
1 <!-- definition of UncInputIdType -->
2 <xs:simpleType name="UncInputIdType">
```



METAS VNA Tools - Data Formats V2.7.0

```
3     <xs:restriction base="xs:string">
4         <xs:pattern value="([0-9a-fA-F][0-9a-fA-F][\-\-]?)+"/>
5     </xs:restriction>
6 </xs:simpleType>
```



4 sdatcv File Specification

The sdatcv-file format is an ASCII text file type developed by METAS. sdatcv-files consist of a header block followed by one or more sets of S-parameter data. For each frequency there is one set of data. It contains the values of the S-parameters and their covariance matrix. There are some general rules for sdatcv-files:

1. sdatcv-files contain only ASCII characters and the evaluation of sdatcv-files is case-insensitive.
2. Individual entries in a header or data line are separated by tabulator.
3. Header and data lines are terminated by a newline character (CR or CR/LF combination).
4. The decimal symbol for floating point values is the point (.) and not the comma (,), e.g.: 1.234567e-08. Note that digit-grouping symbols like (') are not allowed.
5. By convention, sdatcv-filenames use the file extension 'sdatcv'.

4.1 Header Lines

Each sdatcv-file must contain a header block. The header block is formatted as follows:

```
1 SDATCV
2 Ports
3 1
4 Zr [1]re   Zr [1]im
5 50.0      0.0
6 Freq      S [1,1]re  S [1,1]im  CV [1,1]   CV [2,1]   CV [1,2]   CV [2,2]
```

Here the first header line defines that it is a sdatcv-file. The other five header lines are described in the following subsections.

4.1.1 Port Assignment

The keyword 'Ports' in header line 2 initiates the port assignment. Header line 3 describes the used ports by a list of VNA port descriptions. A VNA port description consists of an integer port number and an optional letter which describes the port-mode. No letter or 's' denotes single-ended, 'd' is differential mode and 'c' is common mode.

4.1.2 Reference Impedance

The reference impedance is described in header lines 4 and 5. For each port the reference impedance in Ohm is formatted as a pair of values (real-imaginary).

4.1.3 Data Column Description

Header line 6 describes the data columns. The first column is the frequency column followed by the S-parameter data columns. These are formatted as pairs of values (real-imaginary).



METAS VNA Tools - Data Formats V2.7.0

After the S-parameter columns follow the covariance columns. They are as well formatted as pairs of values. It's possible to specify only certain parts of the covariance matrix. For completing partially given covariance matrices, it's assumed that the matrix is symmetric. Values which can not be deduced from symmetry are set to zero. The following table describes the order of the S-parameters of a n -port in the covariance matrix:

Table 2: Order of the S-parameters in the covariance matrix

Parameter	Index	Parameter	Index	...	Parameter	Index
$S_{1,1re}$	1	$S_{1,2re}$	$2n + 1$...	$S_{1,nre}$	$2n^2 - 2n + 1$
$S_{1,1im}$	2	$S_{1,2im}$	$2n + 2$...	$S_{1,nim}$	$2n^2 - 2n + 2$
$S_{2,1re}$	3	$S_{2,2re}$	$2n + 3$...	$S_{2,nre}$	$2n^2 - 2n + 3$
$S_{2,1im}$	4	$S_{2,2im}$	$2n + 4$...	$S_{2,nim}$	$2n^2 - 2n + 4$
\vdots	\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
$S_{n,1re}$	$2n - 1$	$S_{n,2re}$	$4n - 1$...	$S_{n,nre}$	$2n^2 - 1$
$S_{n,1im}$	$2n$	$S_{n,2im}$	$4n$...	$S_{n,nim}$	$2n^2$

E.g., the covariance of the real part of $S_{1,1}$ and the imaginary part of $S_{2,2}$ would be CV[1,8] for a two port device.

4.2 Data Lines

After the header lines follow the data sets. They contain the S-parameter data. Each data set starts with the frequency in Hz and ends with a newline character (CR or CR/LF combination). After the frequency follow the S-parameter and covariance data. These are formatted as pairs of values (real-imaginary). Each data set has to have as many entries as defined in the data column description. The data set have to be arranged in increasing order of frequency.

4.2.1 1-Port Example

The following example shows a sdatcv-file of a 1-port with a complete covariance matrix (correlation between real and imaginary parts):

```

1 SDATCV
2 Ports
3 1
4 Zr [1] re    Zr [1] im
5 50.0         0.0
6 Freq        S [1,1] re  S [1,1] im  CV [1,1]    CV [2,1]    CV [1,2]    CV [2,2]
7 1.00e+9     -9.16e-1    3.91e-1    1.39e-6    3.56e-7    3.56e-7    2.05e-6
8 2.00e+9     -6.90e-1    7.17e-1    1.98e-6    2.47e-7    2.47e-7    1.96e-6
9 3.00e+9     -3.55e-1    9.29e-1    2.58e-6    3.88e-7    3.88e-7    1.74e-6

```

CV[1,1] is the variance of the real part of $S_{1,1}$.

CV[2,1] and CV[1,2] describe the covariance between the real and imaginary parts of $S_{1,1}$.

CV[2,2] is the variance of the imaginary part of $S_{1,1}$.



METAS VNA Tools - Data Formats V2.7.0

4.2.2 2-Port Examples

The following example shows a sdatcv-file of a 2-port with a reduced covariance matrix (correlation between real and imaginary parts, but no correlation between different S-parameters):

```
1 SDATCV
2 Ports
3 1      2
4 Zr[1]re Zr[1]im Zr[2]re Zr[2]im
5 50.0    0.0    50.0    0.0
6 Freq    S[1,1]re S[1,1]im S[2,1]re S[2,1]im S[1,2]re S[1,2]im
   S[2,2]re S[2,2]im CV[1,1] CV[2,1] CV[2,2] CV[3,3] CV[4,3]
   CV[3,4] CV[4,4] CV[5,5] CV[6,5] CV[5,6] CV[6,6] CV[7,7]
   CV[8,7] CV[7,8] CV[8,8]
7 1.00e+9 -3.72e-3 5.39e-3 2.35e-1 -2.13e-1 2.35e-1 -2.14e-1
   -3.90e-3 6.39e-3 8.00e-8 -1.32e-9 7.86e-8 4.48e-8 2.69e-8
   2.69e-8 4.98e-8 4.50e-8 2.70e-8 2.70e-8 5.00e-8 8.46e-8
   4.22e-11 4.22e-11 8.55e-8
8 2.00e+9 -4.99e-4 9.12e-3 3.05e-2 -3.15e-1 3.05e-2 -3.15e-1
   1.82e-3 8.80e-3 8.14e-8 -5.05e-10 7.97e-8 6.69e-8 4.46e-9
   4.46e-9 2.12e-8 6.74e-8 4.38e-9 4.38e-9 2.15e-8 8.06e-8
   9.99e-10 9.99e-10 8.25e-8
9 3.00e+9 3.81e-3 1.16e-2 -1.89e-1 -2.54e-1 -1.89e-1 -2.54e-1
   7.37e-3 7.74e-3 1.46e-7 6.52e-10 1.45e-7 4.72e-8 -1.88e-8
   -1.88e-8 3.59e-8 4.72e-8 -1.89e-8 -1.89e-8 3.59e-8 1.51e-7
   -7.87e-10 -7.87e-10 1.51e-7
```

CV[1,1] is the variance of the real part of $S_{1,1}$.

CV[2,1] and CV[1,2] describe the covariance between the real and imaginary parts of $S_{1,1}$.

CV[2,2] is the variance of the imaginary part of $S_{1,1}$.

CV[3,3] is the variance of the real part of $S_{2,1}$.

CV[4,3] and CV[3,4] describe the covariance between the real and imaginary parts of $S_{2,1}$.

CV[4,4] is the variance of the imaginary part of $S_{2,1}$.

CV[5,5] is the variance of the real part of $S_{1,2}$.

CV[6,5] and CV[5,6] describe the covariance between the real and imaginary parts of $S_{1,2}$.

CV[6,6] is the variance of the imaginary part of $S_{1,2}$.

CV[7,7] is the variance of the real part of $S_{2,2}$.

CV[8,7] and CV[7,8] describe the covariance between the real and imaginary parts of $S_{2,2}$.

CV[8,8] is the variance of the imaginary part of $S_{2,2}$.

The following example shows a sdatcv-file of a 2-port with a complete covariance matrix (correlation between real and imaginary parts of all S-parameters):



METAS VNA Tools - Data Formats V2.7.0

```

1 SDATCV
2 Ports
3 1      2
4 Zr[1]re  Zr[1]im  Zr[2]re  Zr[2]im
5 50.0     0.0      50.0     0.0
6 Freq      S[1,1]re  S[1,1]im  S[2,1]re  S[2,1]im  S[1,2]re  S[1,2]im
   S[2,2]re  S[2,2]im  CV[1,1]   CV[2,1]   CV[3,1]   CV[4,1]   CV[5,1]
   CV[6,1]   CV[7,1]   CV[8,1]   CV[1,2]   CV[2,2]   CV[3,2]   CV[4,2]
   CV[5,2]   CV[6,2]   CV[7,2]   CV[8,2]   CV[1,3]   CV[2,3]   CV[3,3]
   CV[4,3]   CV[5,3]   CV[6,3]   CV[7,3]   CV[8,3]   CV[1,4]   CV[2,4]
   CV[3,4]   CV[4,4]   CV[5,4]   CV[6,4]   CV[7,4]   CV[8,4]   CV[1,5]
   CV[2,5]   CV[3,5]   CV[4,5]   CV[5,5]   CV[6,5]   CV[7,5]   CV[8,5]
   CV[1,6]   CV[2,6]   CV[3,6]   CV[4,6]   CV[5,6]   CV[6,6]   CV[7,6]
   CV[8,6]   CV[1,7]   CV[2,7]   CV[3,7]   CV[4,7]   CV[5,7]   CV[6,7]
   CV[7,7]   CV[8,7]   CV[1,8]   CV[2,8]   CV[3,8]   CV[4,8]   CV[5,8]
   CV[6,8]   CV[7,8]   CV[8,8]
7 1.00e+9   -3.72e-3  5.39e-3   2.35e-1   -2.13e-1  2.35e-1   -2.14e-1
   -3.90e-3  6.39e-3   8.00e-8   -1.32e-9  -9.15e-10 -2.38e-10 -1.30e-9
   5.48e-11  -2.13e-8  -4.74e-8  -1.32e-9  7.86e-8   -1.66e-9  -2.15e-9
   -1.91e-9  -2.48e-9  4.47e-8   -2.42e-8  -9.15e-10 -1.66e-9  4.48e-8
   2.69e-8   3.45e-8   2.79e-8   -1.49e-10 -7.61e-9  -2.38e-10 -2.15e-9
   2.69e-8   4.98e-8   2.80e-8   3.97e-8   3.21e-9   -7.84e-9  -1.30e-9
   -1.91e-9  3.45e-8   2.80e-8   4.50e-8   2.70e-8   5.68e-10 -7.06e-9
   5.48e-11  -2.48e-9  2.79e-8   3.97e-8   2.70e-8   5.00e-8   2.55e-9
   -7.22e-9  -2.13e-8  4.47e-8   -1.49e-10 3.21e-9   5.68e-10 2.55e-9
   8.46e-8   4.22e-11 -4.74e-8  -2.42e-8  -7.61e-9  -7.84e-9  -7.06e-9
   -7.22e-9  4.22e-11  8.55e-8
8 2.00e+9   -4.99e-4  9.12e-3   3.05e-2   -3.15e-1  3.05e-2   -3.15e-1
   1.82e-3   8.80e-3   8.14e-8   -5.05e-10 -1.21e-9  -2.87e-10 -1.58e-9
   -9.18e-10 -5.13e-8  2.08e-10  -5.05e-10 7.97e-8   -9.83e-10 2.11e-10
   -4.86e-10 -3.19e-10 -4.38e-9  -5.22e-8  -1.21e-9  -9.83e-10 6.69e-8
   4.46e-9   5.78e-8   4.67e-9  -2.97e-9  -4.73e-9  -2.87e-10 2.11e-10
   4.46e-9   2.12e-8   4.58e-9  1.01e-8   1.03e-9  -3.86e-10 -1.58e-9
   -4.86e-10 5.78e-8   4.58e-9  6.74e-8   4.38e-9  -2.53e-9  -4.32e-9
   -9.18e-10 -3.19e-10 4.67e-9  1.01e-8   4.38e-9  2.15e-8   5.67e-10
   4.68e-11 -5.13e-8  -4.38e-9  -2.97e-9  1.03e-9  -2.53e-9 5.67e-10
   8.06e-8   9.99e-10 2.08e-10  -5.22e-8  -4.73e-9  -3.86e-10 -4.32e-9
   4.68e-11 9.99e-10  8.25e-8
9 3.00e+9   3.81e-3  1.16e-2  -1.89e-1  -2.54e-1  -1.89e-1  -2.54e-1
   7.37e-3   7.74e-3   1.46e-7   6.52e-10 -9.51e-10 1.55e-9   -6.48e-10
   2.26e-9   -4.75e-8  2.02e-8   6.52e-10 1.45e-7   -1.75e-9  6.72e-10
   -2.51e-9  8.33e-10 -2.38e-8  -5.19e-8  -9.51e-10 -1.75e-9  4.72e-8
   -1.88e-8  3.74e-8   -1.98e-8  4.16e-9  -7.01e-9  1.55e-9  6.72e-10
   -1.88e-8  3.59e-8   -1.98e-8  2.55e-8  7.48e-10 6.13e-9  -6.48e-10
   -2.51e-9  3.74e-8   -1.98e-8  4.72e-8  -1.89e-8 3.44e-9  -6.72e-9
   2.26e-9  8.33e-10 -1.98e-8  2.55e-8  -1.89e-8 3.59e-8  3.21e-10
   5.44e-9  -4.75e-8  -2.38e-8  4.16e-9  7.48e-10 3.44e-9  3.21e-10
   1.51e-7   -7.87e-10 2.02e-8  -5.19e-8  -7.01e-9 6.13e-9  -6.72e-9
   5.44e-9  -7.87e-10 1.51e-7

```

4.3 Comment Lines

One can add comments to a sdatcv-file. Comments are always preceded by a percent sign (%). A comment can be the only entry on a line or can follow the data on any line.



5 Touchstone V1.x snp File Specification

The Touchstone snp-file format is an ASCII text file type developed by the EIA/IBIS Open Forum. For the Touchstone snp file specification see https://ibis.org/connector/touchstone_spec11.pdf.

5.1 Examples

The following example shows a Touchstone s1p-file of a 1-port:

```
1 # Hz S RI R 50.0
2 1.00e+9 -9.16e-1 3.91e-1
3 2.00e+9 -6.90e-1 7.17e-1
4 3.00e+9 -3.55e-1 9.29e-1
```

The following example shows a Touchstone s2p-file of a 2-port:

```
1 # Hz S RI R 50.0
2 1.00e+9 -3.72e-3 5.39e-3 2.35e-1 -2.13e-1 2.35e-1 -2.14e-1 -3.90e-3 6.39e-3
3 2.00e+9 -4.99e-4 9.12e-3 3.05e-2 -3.15e-1 3.05e-2 -3.15e-1 1.82e-3 8.80e-3
4 3.00e+9 3.81e-3 1.16e-2 -1.89e-1 -2.54e-1 -1.89e-1 -2.54e-1 7.37e-3
7.74e-3
```



6 Touchstone V2.0 ts File Specification

The Touchstone ts-file format is an ASCII text file type developed by the EIA/IBIS Open Forum. For the Touchstone ts file specification see https://ibis.org/touchstone_ver2.0/touchstone_ver2_0.pdf.

6.1 Examples

The following example shows a Touchstone ts-file of a 1-port:

```
1 [Version] 2.0
2 # Hz S RI R 50.0
3 ! Metas.Vna.Tools, 2.1.6907.29753
4 ! Metas.Vna.Data, 2.1.6907.29469
5 ! Created: UTC 2018.11.30 07:25:08
6 [Number of Ports] 1
7 [Number of Frequencies] 3
8 [Reference]
9 50.0
10 [Network Data]
11 ! FREQ      re:S1,1    im:S1,1
12   1.00e+9   -9.16e-1   3.91e-1
13   2.00e+9   -6.90e-1   7.17e-1
14   3.00e+9   -3.55e-1   9.29e-1
15 [End]
```

The following example shows a Touchstone ts-file of a 2-port:

```
1 [Version] 2.0
2 # Hz S RI R 50.0
3 ! Metas.Vna.Tools, 2.1.6907.29753
4 ! Metas.Vna.Data, 2.1.6907.29469
5 ! Created: UTC 2018.11.30 07:25:20
6 [Number of Ports] 2
7 [Two-Port Data Order] 21_12
8 [Number of Frequencies] 3
9 [Reference]
10 50.0 50.0
11 [Network Data]
12 ! FREQ      re:S1,1    im:S1,1    re:S2,1    im:S2,1    re:S1,2    im:S1,2
13   re:S2,2    im:S2,2
14   1.00e+9   -3.72e-3   5.39e-3   2.35e-1   -2.13e-1   2.35e-1   -2.14e-1
15   -3.90e-3   6.39e-3
16   2.00e+9   -4.99e-4   9.12e-3   3.05e-2   -3.15e-1   3.05e-2   -3.15e-1
17   1.82e-3   8.80e-3
18   3.00e+9   3.81e-3   1.16e-2   -1.89e-1   -2.54e-1   -1.89e-1   -2.54e-1
19   7.37e-3   7.74e-3
20 [End]
```




7 vdatb File Specification

The vdatb-file format is a binary file type developed by METAS. The file can be written in a GZIP file stream to reduce the file size. The byte ordering is little-endian.

7.1 Binary Structure Version 1

Version 1 of vdatb uses a GZIP file stream to reduce the file size. The following enumeration describes the binary structure of a vdatb-file:

1. Header (string), value: '%VDATA'
2. Version (int32), value: 1
3. Number of Frequencies (int32)
4. Number of Ports (int32)
5. Number of Parameters (int32)
6. Frequency List (double[]), size: number of frequencies
7. Ports (int32[]), size: number of ports
8. Port Impedance (ComplexUncNumber[]), size: number of ports
9. Data (VnaParameterData[]), size: number of parameters

For the definition of 'ComplexUncNumber' see section 2.1.1.

7.1.1 VNA Parameter Data

The following enumeration describes the binary structure of 'VnaParameterData':

1. Parameter (VnaParameter)
2. Data (ComplexUncNumber[]), size: number of frequencies

7.1.2 VNA Parameter

The following enumeration describes the binary structure of 'VnaParameter':

1. Numerator Receiver (ReceiverType)
2. Numerator Port (int32)
3. Denominator Receiver (ReceiverType)
4. Denominator Port (int32)
5. Source Port (int32)

'ReceiverType' is an enumeration represented by an integer where 0 is '1', 1 is the test receiver 'b' and 2 is the reference receiver 'a'.



METAS VNA Tools - Data Formats V2.7.0

7.1.3 VNA Parameter Examples

Table 3 shows some examples of VNA Parameters. Where W1 is the switch term of port 1

Table 3: VNA Parameter

	S11	S21	S12	S22	W1	W2
Numerator Receiver	'b'	'b'	'b'	'b'	'a'	'a'
Numerator Port	1	2	1	2	1	2
Denominator Receiver	'a'	'a'	'a'	'a'	'b'	'b'
Denominator Port	1	1	2	2	1	2
Source Port	1	1	2	2	2	1

and W2 is the switch term of port 2

7.2 Binary Structure Version 2

Version 2 of vdatb is not using a GZIP file stream. The redundancy of the data is removed by storing uncertainty inputs in a look up table. The uncompressed file size of version 2 is comparable to the GZIP compressed file size of version 1 and about four times smaller than the uncompressed file size of version 1. Avoiding the GZIP step and reducing the uncompressed file size speeds up loading and saving of files. The following enumeration describes the binary structure of a vdatb-file:

1. Header (string), value: '%VDATA'
2. Version (int32), value: 2
3. Number of Frequencies (int32)
4. Number of Ports (int32)
5. Number of Parameters (int32)
6. Frequency List (double[]), size: number of frequencies
7. Ports (int32[]), size: number of ports
8. VNA Parameters (VnaParameter[]), size: number of parameters
9. Flat Vector (UncNumber[]), size: $2N_{Ports} + 2N_{Parameters}N_{Freq}$

The last element contains the port impedance and the data mapped to a flat vector of uncertainty numbers.

For the definition of 'VnaParameter' see section 7.1.2. For the definition of 'Flat Vector of Uncertainty Numbers' see section 2.2.1.



METAS VNA Tools - Data Formats V2.7.0

7.3 Binary Structure Version 3

Version 3 of vdatb is an extension to version 2 which adds support for mixed-mode parameters and port indices. The following enumeration describes the binary structure of a vdatb-file:

1. Header (string), value: '%VDATA'
2. Version (int32), value: 3
3. Number of Frequencies (int32)
4. Number of Ports (int32)
5. Number of Parameters (int32)
6. Frequency List (double[]), size: number of frequencies
7. Ports (VnaPortDescription[]), size: number of ports
8. VNA Parameters (VnaParameter[]), size: number of parameters
9. Flat Vector (UncNumber[]), size: $2N_{Ports} + 2N_{Parameters}N_{Freq}$

The last element contains the port impedance and the data mapped to a flat vector of uncertainty numbers.

For the definition of 'VnaPortDescription' see section 2.3.1. For the definition of 'Vna-Parameter' see section 7.1.2. For the definition of 'Flat Vector of Uncertainty Numbers' see section 2.2.1.

7.4 Binary Structure Version 4

Version 4 of vdatb is an extension to version 3 which adds support for frequency converting parameters. The following enumeration describes the binary structure of a vdatb-file:

1. Header (string), value: '%VDATA'
2. Version (int32), value: 4
3. Number of Frequencies (int32)
4. Number of Ports (int32)
5. Number of Parameters (int32)
6. Frequency List (double[]), size: number of frequencies
7. Ports (VnaPortDescription[]), size: number of ports
8. Frequency Conversion List (FrequencyConversionSub[]), size: number of ports
9. VNA Parameters (VnaParameter[]), size: number of parameters
10. Flat Vector (UncNumber[]), size: $2N_{Ports} + 2N_{Parameters}N_{Freq}$



METAS VNA Tools - Data Formats V2.7.0

The last element contains the port impedance and the data mapped to a flat vector of uncertainty numbers.

For the definition of 'VnaPortDescription' see section 2.3.1. For the definition of 'FrequencyConversionSub' see section 2.4.1. For the definition of 'VnaParameter' see section 7.1.2. For the definition of 'Flat Vector of Uncertainty Numbers' see section 2.2.1.

7.5 Binary Structure Version 5

Version 5 of vdatb is an extension to version 4 which adds support for frequency converting parameters, where the receiver frequency is not equal to the source frequency. The following enumeration describes the binary structure of a vdatb-file:

1. Header (string), value: '%VDATA'
2. Version (int32), value: 5
3. Number of Frequencies (int32)
4. Number of Ports (int32)
5. Number of Parameters (int32)
6. Frequency List (double[]), size: number of frequencies
7. Ports (VnaPortDescription[]), size: number of ports
8. Frequency Conversion List (FrequencyConversion[]), size: number of ports
9. VNA Parameters (VnaParameter[]), size: number of parameters
10. Flat Vector (UncNumber[]), size: $2N_{Ports} + 2N_{Parameters}N_{Freq}$

The last element contains the port impedance and the data mapped to a flat vector of uncertainty numbers.

For the definition of 'VnaPortDescription' see section 2.3.1. For the definition of 'FrequencyConversion' see section 2.5.1. For the definition of 'VnaParameter' see section 7.1.2. For the definition of 'Flat Vector of Uncertainty Numbers' see section 2.2.1.

7.6 Example MATLAB Code

The following code shows how to load a vdatb-file in MATLAB:

```
1 function d = LoadVDATEB(filepath)
2 % Loads VNA Tools II VnaData (*.vdatb) file
3 % Michael Wollensack METAS - 22.04.2022
4
5 d = {};
6 f3 = OpenFile(filepath);
7 % Type
8 type = char(f3.ReadString());
9 % Version
10 version = int32(f3.ReadInt32());
11 disp(['Type: ' type ', Version: ' num2str(version)])
```



METAS VNA Tools - Data Formats V2.7.0

```
12 if (strcmp(type , '%VDATA') && 1 <= version && version <= 5)
13     % Number of Frequencies
14     nfreq = f3.ReadInt32();
15     % Number of Ports
16     nports = f3.ReadInt32();
17     % Number of Parameters
18     nparams = f3.ReadInt32();
19     % Init
20     d.Frequency = zeros(1, nfreq);
21     d.Ports = cell(1, nports);
22     d.PortZr = LinProp(zeros(1, nports));
23     d.FrequencyConversions = cell(1, nports);
24     d.VnaParameter = cell(1, nparams);
25     d.Data = LinProp(zeros(nfreq, nparams));
26     % Frequency (Hz)
27     for i = 1:nfreq
28         d.Frequency(i) = f3.ReadDouble();
29     end
30     % Ports
31     for i = 1:nports
32         d.Ports{i} = ReadPort(f3, version);
33     end
34     % Frequency Conversions
35     for i = 1:nports
36         if (version < 4)
37             d.FrequencyConversions{i}.TestReceiver.Numerator = 1;
38             d.FrequencyConversions{i}.TestReceiver.Denominator = 1;
39             d.FrequencyConversions{i}.TestReceiver.Offset = 0;
40             d.FrequencyConversions{i}.ReferenceReceiver.Numerator = 1;
41             d.FrequencyConversions{i}.ReferenceReceiver.Denominator = 1;
42             d.FrequencyConversions{i}.ReferenceReceiver.Offset = 0;
43             d.FrequencyConversions{i}.Source.Numerator = 1;
44             d.FrequencyConversions{i}.Source.Denominator = 1;
45             d.FrequencyConversions{i}.Source.Offset = 0;
46         elseif (version == 4)
47             numerator = f3.ReadDouble();
48             denominator = f3.ReadDouble();
49             offset = f3.ReadDouble();
50             d.FrequencyConversions{i}.TestReceiver.Numerator = numerator;
51             d.FrequencyConversions{i}.TestReceiver.Denominator =
52                 denominator;
53             d.FrequencyConversions{i}.TestReceiver.Offset = offset;
54             d.FrequencyConversions{i}.ReferenceReceiver.Numerator =
55                 numerator;
56             d.FrequencyConversions{i}.ReferenceReceiver.Denominator =
57                 denominator;
58             d.FrequencyConversions{i}.ReferenceReceiver.Offset = offset;
59             d.FrequencyConversions{i}.Source.Numerator = numerator;
60             d.FrequencyConversions{i}.Source.Denominator = denominator;
61             d.FrequencyConversions{i}.Source.Offset = offset;
62         else
63             d.FrequencyConversions{i}.TestReceiver.Numerator =
64                 f3.ReadDouble();
65             d.FrequencyConversions{i}.TestReceiver.Denominator =
66                 f3.ReadDouble();
67             d.FrequencyConversions{i}.TestReceiver.Offset =
68                 f3.ReadDouble();
```



METAS VNA Tools - Data Formats V2.7.0

```
63         d.FrequencyConversions{i}.ReferenceReceiver.Numerator =
64             f3.ReadDouble();
65         d.FrequencyConversions{i}.ReferenceReceiver.Denominator =
66             f3.ReadDouble();
67         d.FrequencyConversions{i}.ReferenceReceiver.Offset =
68             f3.ReadDouble();
69         d.FrequencyConversions{i}.Source.Numerator = f3.ReadDouble();
70         d.FrequencyConversions{i}.Source.Denominator = f3.ReadDouble();
71         d.FrequencyConversions{i}.Source.Offset = f3.ReadDouble();
72     end
73 end
74 if (version == 1)
75     % Port Zr
76     for i = 1:nports
77         d.PortZr(i) = ReadComplexLinProp(f3);
78     end
79     % VnaParameterData
80     for i2 = 1:nparams
81         % VnaParameter
82         d.VnaParameter{i2} = ReadVnaParameter(f3, version);
83         % Data
84         for i1 = 1:nfreq
85             d.Data(i1, i2) = ReadComplexLinProp(f3);
86         end
87     end
88 elseif (2 <= version || version <= 5)
89     % VnaParameter
90     for i2 = 1:nparams
91         d.VnaParameter{i2} = ReadVnaParameter(f3, version);
92     end
93     % Flat Vector
94     v = ReadComplexFlatVectorLinProp(f3);
95     index = 1;
96     % PortsZr
97     for i = 1:nports
98         d.PortZr(i) = v(index); index = index + 1;
99     end
100     % Data
101     for i2 = 1:nparams
102         for i1 = 1:nfreq
103             d.Data(i1, i2) = v(index); index = index + 1;
104         end
105     end
106 end
107 end
108 f3.Close();
109 end
110
111 function p = ReadPort(f3, version)
112 % Read Port
113 ModeType = {'', 'd', 'c'};
114 IndexType = {'', ':I', ':II', ':III', ':IV', ':V', ':VI', ':VII', ':VIII',
115             ':IX', ':X', ':XI', ':XII'};
116 if (version < 3)
117     p = num2str(f3.ReadInt32());
118 else
```



METAS VNA Tools - Data Formats V2.7.0

```
115     p = [num2str(f3.ReadInt32()) ModeType{f3.ReadInt16() + 1}
116         IndexType{f3.ReadInt16() + 1}];
117 end
118 end
119 function p = ReadVnaParameter(f3, version)
120 % Read VNA Parameter
121 RcvType = {1, 'b', 'a'};
122 p = {};
123 p.NumRec = RcvType{f3.ReadInt32() + 1};
124 p.NumPort = ReadPort(f3, version);
125 p.DenRec = RcvType{f3.ReadInt32() + 1};
126 p.DenPort = ReadPort(f3, version);
127 p.SrcPort = ReadPort(f3, version);
128 end
129
130 function f3 = OpenFile(filepath)
131 % Open File
132 NET.addAssembly('System');
133 % File Stream
134 f1 = System.IO.FileStream(filepath, System.IO.FileMode.Open);
135 % Try if Stream is GZIP compressed
136 try
137     f2 = System.IO.Compression.GZipStream(f1,
138         System.IO.Compression.CompressionMode.Decompress);
139     f2.ReadByte();
140     f1.Position = 0;
141     f2 = System.IO.Compression.GZipStream(f1,
142         System.IO.Compression.CompressionMode.Decompress);
143     disp('GZIP compressed file')
144 catch
145     f1.Position = 0;
146     f2 = f1;
147     disp('Uncompressed file')
148 end
149 % Binary Reader
150 f3 = System.IO.BinaryReader(f2);
151 end
152
153 function c = ReadComplexLinProp(f3)
154 % Read ComplexLinProp using METAS UncLib
155 n = NET.createGeneric('Metas.UncLib.Core.Complex',
156     {'Metas.UncLib.LinProp.UncNumber'});
157 n.BinarySetDataFrom(f3);
158 c = LinProp(n);
159 end
160
161 function v = ReadComplexFlatVectorLinProp(f3)
162 % Read ComplexFlatVectorLinProp using METAS UncLib
163 list = Metas.UncLib.LinProp.UncList();
164 list.BinarySetDataFrom(f3);
165 n = NET.createGeneric('Metas.UncLib.Core.Ndims.RealNArray',
166     {'Metas.UncLib.LinProp.UncNumber'});
167 n.Init1dData(list.data);
168 r = LinProp(n);
169 v = r(1:2:end-1) + 1i.*r(2:2:end);
170 end
```



8 vdatx File Specification

The vdatx-file format is an XML file type developed by METAS. It's described using an XML schema. See https://www.w3schools.com/xml/schema_intro.asp for more details about XML schemas. The file can be written in a GZIP file stream to reduce the file size.

8.1 XML Schema

The following listing shows the XML schema for 'VnaData':

```
1 <!-- definition of VnaData -->
2 <xs:element name="VnaData">
3   <xs:complexType>
4     <xs:sequence>
5       <xs:element ref="FrequencyList" />
6       <xs:element ref="PortList" />
7       <xs:element ref="PortZrList" />
8       <xs:element ref="FrequencyConversionList" minOccurs="0"
9         maxOccurs="1" />
10      <xs:element name="ParameterDataList">
11        <xs:complexType>
12          <xs:sequence>
13            <xs:element maxOccurs="unbounded" name="ParameterData">
14              <xs:complexType>
15                <xs:sequence>
16                  <xs:element name="Parameter" type="VnaParameterType" />
17                  <xs:element name="Data">
18                    <xs:complexType>
19                      <!-- Index: Frequency -->
20                      <xs:sequence>
21                        <xs:element maxOccurs="unbounded"
22                          name="Frequency" type="ComplexUncNumberType"
23                          />
24                      </xs:sequence>
25                    </xs:complexType>
26                  </xs:element>
27                </xs:sequence>
28              </xs:complexType>
29            </xs:element>
30          </xs:sequence>
31        </xs:complexType>
32      </xs:element>
```

For the definition of 'FrequencyList', 'PortList', 'PortZrList', 'FrequencyConversionList' and 'ComplexUncNumberType' see section 3.1.1 and 3.1.2.

8.1.1 VNA Parameter Description

The following listing shows the XML schema for 'VnaParameterType':

```
1 <!-- definition of VnaParameterType -->
2 <xs:complexType name="VnaParameterType">
3   <xs:sequence>
```




METAS VNA Tools - Data Formats V2.7.0

```
4     <xs:element name="NumRcv" type="ReceiverType" />
5     <xs:element name="NumPort" type="VnaPortDescriptionType" />
6     <xs:element name="DenRcv" type="ReceiverType" />
7     <xs:element name="DenPort" type="VnaPortDescriptionType" />
8     <xs:element name="SrcPort" type="VnaPortDescriptionType" />
9     </xs:sequence>
10  </xs:complexType>
```

The following listing shows the XML schema for 'ReceiverType':

```
1  <!-- definition of ReceiverType -->
2  <xs:simpleType name="ReceiverType">
3    <xs:restriction base="xs:string">
4      <xs:enumeration value="1"/>
5      <xs:enumeration value="b"/><!-- Test Receiver -->
6      <xs:enumeration value="a"/><!-- Reference Receiver -->
7    </xs:restriction>
8  </xs:simpleType>
```

For the definition of 'VnaPortDescriptionType' see section 3.1.1.



9 vdatcv File Specification

The vdatcv-file format is an ASCII text file type developed by METAS. vdatcv-files consist of a header block followed by one or more sets of VNA parameter data. For each frequency there is one set of data. It contains the values of arbitrary receiver ratios or single receivers and their covariance matrix. There are some general rules for vdatcv-files:

1. vdatcv-files contain only ASCII characters and the evaluation of vdatcv-files is case-insensitive.
2. Individual entries in a header or data line are separated by tabulator.
3. Header and data lines are terminated by a newline character (CR or CR/LF combination).
4. The decimal symbol for floating point values is the point (.) and not the comma (,), e.g.: 1.234567e-08. Note that digit-grouping symbols like (') are not allowed.
5. By convention, vdatcv-filenames use the file extension 'vdatcv'.

9.1 Header Lines

Each vdatcv-file must contain a header block. The header block is formatted as follows:

```
1 VDATCV
2 Ports
3 1
4 Zr [1]re   Zr [1]im
5 50.0      0.0
6 Freq      a1/b1,2re a1/b1,2im CV [1,1]   CV [2,1]   CV [2,2]
```

Here the first header line defines that it is a vdatcv-file. The other five header lines are described in the following subsections.

9.1.1 Port Assignment

The keyword 'Ports' in header line 2 initiates the port assignment. Header line 3 describes the used ports by a list of VNA port descriptions. A VNA port description consists of an integer port number and an optional letter which describes the port-mode. No letter or 's' denotes single-ended, 'd' is differential mode and 'c' is common mode.

9.1.2 Reference Impedance

The reference impedance is described in header lines 4 and 5. For each port the reference impedance in Ohm is formatted as a pair of values (real-imaginary).

9.1.3 Data Column Description

Header line 6 describes the data columns. The first column is the frequency column followed by the VNA parameter data columns. These are formatted as pairs of values (real-imaginary). Table 4 describes some examples of valid VNA parameters.



METAS VNA Tools - Data Formats V2.7.0

Table 4: Examples of valid VNA parameters

Parameter	Description
'S[1,1]'	S-Parameter, reflection at port 1
'S[2,1]'	S-Parameter, transmission from port 1 to port 2
'b1,1'	Test receiver of port 1 when the source is switched to port 1
'b2,1'	Test receiver of port 2 when the source is switched to port 1
'a1,1'	Reference receiver of port 1 when the source is switched to port 1
'a1/b1,2'	Ratio of reference to test receivers of port 1 when the source is switched to port 2

After the parameter columns follow the covariance columns. They are as well formatted as pairs of values. It's possible to specify only certain parts of the covariance matrix. For completing partially given covariance matrices, it's assumed that the matrix is symmetric. Values which can not be deduced from symmetry are set to zero.

9.2 Data Lines

After the header lines follow the data sets. They contain the parameter data. Each data set starts with the frequency in Hz and ends with a newline character (CR or CR/LF combination). After the frequency follow the parameter and covariance data.

These are formatted as pairs of values (real-imaginary). Each data set has to have as many entries as defined in the data column description. The data set have to be arranged in increasing order of frequency.

9.2.1 Example

The following example shows a vdatcv-file with one parameter and its covariance:

```

1 VDATCV
2 Ports
3 1
4 Zr [1] re   Zr [1] im
5 50.0       0.0
6 Freq      a1/b1,2re a1/b1,2im CV [1,1]   CV [2,1]   CV [2,2]
7 1.00e+9   -9.16e-2  3.91e-2  1.39e-6   3.56e-7   2.05e-6
8 2.00e+9   -6.90e-2  7.17e-2  1.98e-6   2.47e-7   1.96e-6
9 3.00e+9   -3.55e-2  9.29e-2  2.58e-6   3.88e-7   1.74e-6

```

CV[1,1] is the variance of the real part of the receiver ratio $\frac{a_1}{b_1}$ when the source is switched to port 2.

CV[2,1] describe the covariance between the real and imaginary parts of the receiver ratio $\frac{a_1}{b_1}$ when the source is switched to port 2.

CV[2,2] is the variance of the imaginary part of the receiver ratio $\frac{a_1}{b_1}$ when the source is switched to port 2.



METAS VNA Tools - Data Formats V2.7.0

9.3 Comment Lines

One can add comments to a vdatcv-file. Comments are always preceded by a percent sign (%). A comment can be the only entry on a line or can follow the data on any line.



10 CITI File Specification

The CITI-file format is an ASCII text file type developed by Agilent. For the CITI file specification see:

- <https://na.support.keysight.com/pna/dbcal.html>
- <http://literature.cdn.keysight.com/litweb/pdf/ads2001/pdf/instrumentio.pdf>
- https://na.support.keysight.com/vna/help/latest/S5_Output/SaveRecall.htm#cti
- <http://hermes.eee.nott.ac.uk/teaching/ads/doc/cktsim/ck0419.html>

10.1 Examples

The following example shows a CITI-file of a 1-port:

```
1 CITIFILE A.01.01
2 NAME DATA
3 VAR FREQ MAG 3
4 DATA S[1,1] RI
5 DATA U[1,1] RI
6 VAR_LIST_BEGIN
7 1.0000000000e+009
8 2.0000000000e+009
9 3.0000000000e+009
10 VAR_LIST_END
11 BEGIN
12 -9.1600000000e-001,3.9100000000e-001
13 -6.9000000000e-001,7.1700000000e-001
14 -3.5500000000e-001,9.2900000000e-001
15 END
16 BEGIN
17 2.3579652245e-003,2.8635642127e-003
18 2.8142494559e-003,2.8000000000e-003
19 3.2124756808e-003,2.6381811917e-003
20 END
```

The following example shows a CITI-file of a 2-port:

```
1 CITIFILE A.01.01
2 NAME DATA
3 VAR FREQ MAG 3
4 DATA S[1,1] RI
5 DATA U[1,1] RI
6 DATA S[2,1] RI
7 DATA U[2,1] RI
8 DATA S[1,2] RI
9 DATA U[1,2] RI
10 DATA S[2,2] RI
11 DATA U[2,2] RI
12 VAR_LIST_BEGIN
13 1.0000000000e+009
14 2.0000000000e+009
15 3.0000000000e+009
16 VAR_LIST_END
17 BEGIN
```



METAS VNA Tools - Data Formats V2.7.0

```
18 -3.7200000000e-003,5.3900000000e-003
19 -4.9900000000e-004,9.1200000000e-003
20 3.8100000000e-003,1.1600000000e-002
21 END
22 BEGIN
23 5.6568542495e-004,5.6071380159e-004
24 5.7061365532e-004,5.6462376854e-004
25 7.6419847665e-004,7.6157671523e-004
26 END
27 BEGIN
28 2.3500000000e-001,-2.1300000000e-001
29 3.0500000000e-002,-3.1500000000e-001
30 -1.8900000000e-001,-2.5400000000e-001
31 END
32 BEGIN
33 4.2332020977e-004,4.4631815719e-004
34 5.1730068626e-004,2.9120439557e-004
35 4.3451121965e-004,3.7894590643e-004
36 END
37 BEGIN
38 2.3500000000e-001,-2.1400000000e-001
39 3.0500000000e-002,-3.1500000000e-001
40 -1.8900000000e-001,-2.5400000000e-001
41 END
42 BEGIN
43 4.2426406871e-004,4.4721359550e-004
44 5.1923019943e-004,2.9325756597e-004
45 4.3451121965e-004,3.7894590643e-004
46 END
47 BEGIN
48 -3.9000000000e-003,6.3900000000e-003
49 1.8200000000e-003,8.8000000000e-003
50 7.3700000000e-003,7.7400000000e-003
51 END
52 BEGIN
53 5.8172158289e-004,5.8480766069e-004
54 5.6780278266e-004,5.7445626465e-004
55 7.7717384603e-004,7.7717423771e-004
56 END
```



11 scolb File Specification

The METAS scolb file format is a zip file which contains multiple sdatb files, see section 2.

12 scolcv File Specification

The scolcv-file format is an ASCII text file type developed by METAS. scolcv-files consist of a header block followed by one or more sets of S-parameter data of multiple standards. For each frequency there is one set of data. It contains the values of the S-parameters of multiple standards and their covariance matrix. There are some general rules for scolcv-files:

1. scolcv-files contain only ASCII characters and the evaluation of scolcv-files is case-insensitive.
2. Individual entries in a header or data line are separated by tabulator.
3. Header and data lines are terminated by a newline character (CR or CR/LF combination).
4. The decimal symbol for floating point values is the point (.) and not the comma (,), e.g.: 1.234567e-08. Note that digit-grouping symbols like (') are not allowed.
5. By convention, scolcv-filenames use the file extension 'scolcv'.

12.1 Header Lines

Each scolcv-file must contain a header block. The header block is formatted as follows:

```
1 SCOLCV
2 -----
3 Number
4 1
5 Name
6 Standard_01
7 Ports
8 1
9 Zr[1]re   Zr[1]im
10 50.0      0.0
11 -----
12 Number
13 2
14 Name
15 Standard_02
16 Ports
17 1
18 Zr[1]re   Zr[1]im
19 50.0      0.0
20 -----
21 Freq      1:S[1,1]re      1:S[1,1]im      2:S[1,1]re
           2:S[1,1]im      CV[1,1]      CV[2,1]      CV[3,1]      CV[4,1]      CV[2,2]
           CV[3,2]      CV[4,2]      CV[3,3]      CV[4,3]      CV[4,4]
```

Here the first header line defines that it is a scolcv-file. The other header lines are described in the following subsections.



METAS VNA Tools - Data Formats V2.7.0

12.1.1 Number

The keyword 'Number' in header line $9(i - 1) + 3$ initiates the number. Header line $9(i - 1) + 4$ describes the current standard i .

12.1.2 Name

The name of the current standard is described in header lines $9(i - 1) + 5$ and $9(i - 1) + 6$.

12.1.3 Port Assignment

The keyword 'Ports' in header line $9(i - 1) + 7$ initiates the port assignment. Header line $9(i - 1) + 8$ describes the used ports of the current standard by a list of VNA port descriptions. A VNA port description consists of an integer port number and an optional letter which describes the port-mode. No letter or 's' denotes single-ended, 'd' is differential mode and 'c' is common mode.

12.1.4 Reference Impedance

The reference impedance of the current standard is described in header lines $9(i - 1) + 9$ and $9(i - 1) + 10$. For each port the reference impedance in Ohm is formatted as a pair of values (real-imaginary).

12.1.5 Data Column Description

The last header line describes the data columns. The first column is the frequency column followed by the S-parameter data columns of all standards. These are formatted as pairs of values (real-imaginary).

After the S-parameter columns follow the covariance columns. They are as well formatted as pairs of values. It's possible to specify only certain parts of the covariance matrix. For completing partially given covariance matrices, it's assumed that the matrix is symmetric. Values which can not be deduced from symmetry are set to zero.

12.2 Data Lines

After the header lines follow the data sets. They contain the S-parameter data. Each data set starts with the frequency in Hz and ends with a newline character (CR or CR/LF combination). After the frequency follow the S-parameter and covariance data. These are formatted as pairs of values (real-imaginary). Each data set has to have as many entries as defined in the data column description. The data set have to be arranged in increasing order of frequency.

12.2.1 Example

The following example shows a scolcv-file of two 1-port standards and their covariance matrix:

```
1 SCOLCV
2 -----
3 Number
4 1
```




METAS VNA Tools - Data Formats V2.7.0

```

5 Name
6 Standard_01
7 Ports
8 1
9 Zr[1]re    Zr[1]im
10 50.0      0.0
11 -----
12 Number
13 2
14 Name
15 Standard_02
16 Ports
17 1
18 Zr[1]re    Zr[1]im
19 50.0      0.0
20 -----
21 Freq      1:S[1,1]re      1:S[1,1]im      2:S[1,1]re
      2:S[1,1]im      CV[1,1]    CV[2,1]    CV[3,1]    CV[4,1]    CV[2,2]
      CV[3,2]    CV[4,2]    CV[3,3]    CV[4,3]    CV[4,4]
22 1.00E+09  -3.72E-03  5.39E-03  -3.90E-03  6.39E-03  8.00E-08  -1.32E-09
      -2.13E-08  -4.74E-08  7.86E-08  4.47E-08  -2.42E-08  8.46E-08  4.22E-11
      8.55E-08
23 2.00E+09  -4.99E-04  9.12E-03  1.82E-03  8.80E-03  8.14E-08  -5.05E-10
      -5.13E-08  2.08E-10  7.97E-08  -4.38E-09  -5.22E-08  8.06E-08  9.99E-10
      8.25E-08
24 3.00E+09  3.81E-03  1.16E-02  7.37E-03  7.74E-03  1.46E-07  6.50E-10
      -4.75E-08  2.02E-08  1.45E-07  -2.38E-08  -5.19E-08  1.51E-07  -7.87E-10
      1.51E-07

```

CV[1,1] is the variance of the real part of $S_{1,1}$ of standard 1.

CV[2,1] describe the covariance between the real and imaginary parts of $S_{1,1}$ of standard 1.

CV[3,1] describe the covariance between standards 1 and 2 of the real parts of $S_{1,1}$.

CV[4,1] describe the covariance between standard 1 real part and standard 2 imaginary part of $S_{1,1}$.

CV[2,2] is the variance of the imaginary part of $S_{1,1}$ of standard 1.

CV[3,2] describe the covariance between standard 1 imaginary part and standard 2 real part of $S_{1,1}$.

CV[4,2] describe the covariance between standards 1 and 2 of the imaginary parts of $S_{1,1}$.

CV[3,3] is the variance of the real part of $S_{1,1}$ of standard 2.

CV[4,3] describe the covariance between the real and imaginary parts of $S_{1,1}$ of standard 2.

CV[4,4] is the variance of the imaginary part of $S_{1,1}$ of standard 2.

12.3 Comment Lines

One can add comments to a scolcv-file. Comments are always preceded by a percent sign (%). A comment can be the only entry on a line or can follow the data on any line.



13 vcolb File Specification

The METAS vcolb file format is a zip file which contains multiple vdatb files, see section 7.

14 vcolcv File Specification

The vcolcv-file format is an ASCII text file type developed by METAS. vcolcv-files consist of a header block followed by one or more sets of VNA parameter data of multiple standards. For each frequency there is one set of data. It contains the values of arbitrary receiver ratios or single receivers of multiple standards and their covariance matrix. There are some general rules for vcolcv-files:

1. vcolcv-files contain only ASCII characters and the evaluation of vcolcv-files is case-insensitive.
2. Individual entries in a header or data line are separated by tabulator.
3. Header and data lines are terminated by a newline character (CR or CR/LF combination).
4. The decimal symbol for floating point values is the point (.) and not the comma (,), e.g.: 1.234567e-08. Note that digit-grouping symbols like (') are not allowed.
5. By convention, vcolcv-filenames use the file extension 'vcolcv'.

14.1 Header Lines

Each vcolcv-file must contain a header block. The header block is formatted as follows:

```
1 VC0LCV
2 -----
3 Number
4 1
5 Name
6 Standard_01
7 Ports
8 1
9 Zr [1] re    Zr [1] im
10 50.0        0.0
11 -----
12 Number
13 2
14 Name
15 Standard_02
16 Ports
17 1
18 Zr [1] re    Zr [1] im
19 50.0        0.0
20 -----
21 Freq          1: S [1, 1] re          1: S [1, 1] im          2: S [1, 1] re
                2: S [1, 1] im          CV [1, 1]          CV [2, 1]          CV [3, 1]          CV [4, 1]          CV [2, 2]
                CV [3, 2]          CV [4, 2]          CV [3, 3]          CV [4, 3]          CV [4, 4]
```



METAS VNA Tools - Data Formats V2.7.0

Here the first header line defines that it is a vcolcv-file. The other header lines are described in the following subsections.

14.1.1 Number

The keyword 'Number' in header line $9(i-1)+3$ initiates the number. Header line $9(i-1)+4$ describes the current standard i .

14.1.2 Name

The name of the current standard is described in header lines $9(i-1)+5$ and $9(i-1)+6$.

14.1.3 Port Assignment

The keyword 'Ports' in header line $9(i-1)+7$ initiates the port assignment. Header line $9(i-1)+8$ describes the used ports of the current standard by a list of VNA port descriptions. A VNA port description consists of an integer port number and an optional letter which describes the port-mode. No letter or 's' denotes single-ended, 'd' is differential mode and 'c' is common mode.

14.1.4 Reference Impedance

The reference impedance of the current standard is described in header lines $9(i-1)+9$ and $9(i-1)+10$. For each port the reference impedance in Ohm is formatted as a pair of values (real-imaginary).

14.1.5 Data Column Description

The last header line describes the data columns. The first column is the frequency column followed by the VNA parameter data columns of all standards. These are formatted as pairs of values (real-imaginary). Table 5 describes some examples of valid VNA parameters for multiple standards.

After the parameter data columns follow the covariance columns. They are as well formatted as pairs of values. It's possible to specify only certain parts of the covariance matrix. For completing partially given covariance matrices, it's assumed that the matrix is symmetric. Values which can not be deduced from symmetry are set to zero.

14.2 Data Lines

After the header lines follow the data sets. They contain the parameter data. Each data set starts with the frequency in Hz and ends with a newline character (CR or CR/LF combination). After the frequency follow the parameter and covariance data. These are formatted as pairs of values (real-imaginary). Each data set has to have as many entries as defined in the data column description. The data set have to be arranged in increasing order of frequency.



METAS VNA Tools - Data Formats V2.7.0

Table 5: Examples of valid VNA parameters for multiple standards

Parameter	Description
'1:S[1,1]'	S-Parameter, reflection at port 1 of standard 1
'2:S[1,1]'	S-Parameter, reflection at port 1 of standard 2
'1:S[2,1]'	S-Parameter, transmission from port 1 to port 2 of standard 1
'2:S[2,1]'	S-Parameter, transmission from port 1 to port 2 of standard 2
'1:b1,1'	Test receiver of port 1 when the source is switched to port 1 of standard 1
'2:b1,1'	Test receiver of port 1 when the source is switched to port 1 of standard 2
'1:b2,1'	Test receiver of port 2 when the source is switched to port 1 of standard 1
'2:b2,1'	Test receiver of port 2 when the source is switched to port 1 of standard 2
'1:a1,1'	Reference receiver of port 1 when the source is switched to port 1 of standard 1
'2:a1,1'	Reference receiver of port 1 when the source is switched to port 1 of standard 2
'1:a1/b1,2'	Ratio of reference to test receivers of port 1 when the source is switched to port 2 of standard 1

14.2.1 Example

The following example shows a vcolcv-file of two 1-port standards and their covariance matrix:

```

1  VC0LCV
2  -----
3  Number
4  1
5  Name
6  Standard_01
7  Ports
8  1
9  Zr[1]re    Zr[1]im
10 50.0        0.0
11 -----
12 Number
13 2
14 Name
15 Standard_02
16 Ports
17 1
18 Zr[1]re    Zr[1]im
19 50.0        0.0
20 -----
21 Freq      1:S[1,1]re    1:S[1,1]im    2:S[1,1]re
      2:S[1,1]im    CV[1,1]    CV[2,1]    CV[3,1]    CV[4,1]    CV[2,2]
      CV[3,2]    CV[4,2]    CV[3,3]    CV[4,3]    CV[4,4]
22 1.00E+09  -3.72E-03  5.39E-03  -3.90E-03  6.39E-03  8.00E-08  -1.32E-09
      -2.13E-08  -4.74E-08  7.86E-08  4.47E-08  -2.42E-08  8.46E-08  4.22E-11
      8.55E-08
23 2.00E+09  -4.99E-04  9.12E-03  1.82E-03  8.80E-03  8.14E-08  -5.05E-10
      -5.13E-08  2.08E-10  7.97E-08  -4.38E-09  -5.22E-08  8.06E-08  9.99E-10
      8.25E-08
24 3.00E+09  3.81E-03  1.16E-02  7.37E-03  7.74E-03  1.46E-07  6.50E-10
      -4.75E-08  2.02E-08  1.45E-07  -2.38E-08  -5.19E-08  1.51E-07  -7.87E-10
      1.51E-07

```



METAS VNA Tools - Data Formats V2.7.0

CV[1,1] is the variance of the real part of $S_{1,1}$ of standard 1.

CV[2,1] describe the covariance between the real and imaginary parts of $S_{1,1}$ of standard 1.

CV[3,1] describe the covariance between standards 1 and 2 of the real parts of $S_{1,1}$.

CV[4,1] describe the covariance between standard 1 real part and standard 2 imaginary part of $S_{1,1}$.

CV[2,2] is the variance of the imaginary part of $S_{1,1}$ of standard 1.

CV[3,2] describe the covariance between standard 1 imaginary part and standard 2 real part of $S_{1,1}$.

CV[4,2] describe the covariance between standards 1 and 2 of the imaginary parts of $S_{1,1}$.

CV[3,3] is the variance of the real part of $S_{1,1}$ of standard 2.

CV[4,3] describe the covariance between the real and imaginary parts of $S_{1,1}$ of standard 2.

CV[4,4] is the variance of the imaginary part of $S_{1,1}$ of standard 2.

14.3 Comment Lines

One can add comments to a vcolcv-file. Comments are always preceded by a percent sign (%). A comment can be the only entry on a line or can follow the data on any line.



A ZIP File with Embedded Data Files

Table 6 shows the supported file types for embedded data files in a ZIP file.

Table 6: File types

Description	Extension
S-Parameter Data Binary	(.sdatb)
S-Parameter Data Xml	(.sdatx)
S-Parameter Data Covariance Text	(.sdatcv)
S-Parameter Data Touchstone	(.s*p;.ts)
VNA Data Binary	(.vdatb)
VNA Data Xml	(.vdatx)
VNA Data Covariance Text	(.vdatcv)
VNA Data CITI	(.cti;.citi)

B PDF File with Embedded Data Files

The supported file types for embedded data files in a PDF file are the same like for a ZIP file, see appendix A and table 6.

B.1 Example

The following \LaTeX -code generates an example of a PDF/A-3u with two embedded data files:

```
1 \documentclass{minimal}
2
3 \usepackage[a-3u]{pdfx}
4 \usepackage{embedfile}
5 \newcommand{\datafile}[2]
6 {
7   \embedfile[
8     filespec={#2},
9     ucfilespec={#2},
10    filesystem=URL,
11    mimetype=application/octet-stream,
12    desc={#2}, stringmethod=escape]{#1#2}
13 }
14 % missing /AF entry and /AFRelationship entry, see
15 % https://tex.stackexchange.com/q/426385/139832
16
17
18 \datafile{Collection/}{Standard_01.sdatb}
19 \datafile{Collection/}{Standard_02.sdatb}
20
21 \begin{document}
22 See the embedded data files ...
23 \end{document}
```



C PTB DCC XML File with Embedded Data Files

The Digital Calibration Certificate (DCC) is an XML Schema Definition (XSD) developed by Physikalisch-Technische Bundesanstalt (PTB). For the PTB DCC development platform see <https://gitlab.com/ptb/dcc/xsd-dcc>.

The supported file types for embedded data files in a PTB DCC XML file are the same like for a ZIP file, see appendix A and table 6. The names of the embedded data files are stored under the following XPath:

```
1 /dcc:digitalCalibrationCertificate
2   /dcc:measurementResults
3     /dcc:measurementResult
4       /dcc:results
5         /dcc:result
6           /dcc:data
7             /dcc:byteData
8               /dcc:filename
```

The following XQuery is used to access the embedded data file:

```
1 /dcc:digitalCalibrationCertificate
2   /dcc:measurementResults
3     /dcc:measurementResult
4       /dcc:results
5         /dcc:result
6           /dcc:data
7             /dcc:byteData
8               /dcc:dataBase64 [../dcc:fileName='NAME ']
```