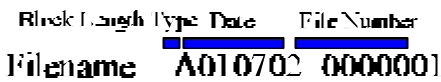
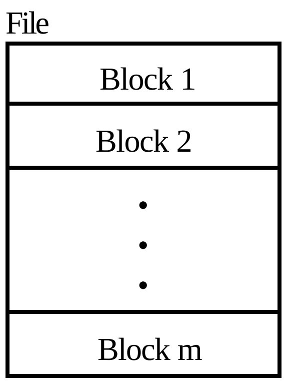
**RSMS-4 File Format**
Version 0.4
04/05/02

## I. General RSMS-4 File Format Information

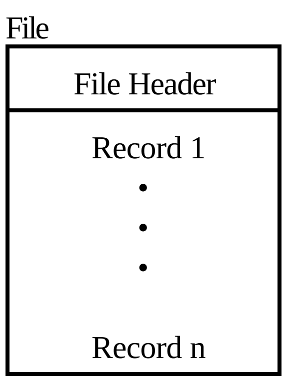


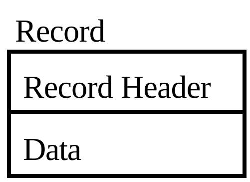

**<u>Imposed Rules</u>**: 1) Software must start a new file when a new calibration is recorded.
2) Software must start a new file when the file header information changes.

**<u>Physical File Structure</u>**:
The file is a random access file that consists of fixed-length blocks[1].




**<u>Logical File Structure</u>**:
File consists of a File Header followed by a variable number of records[2] up to some maximum number of records (in DA the maximum was 128, in RSMS-4 the maximum is user specified). Each record consists of a Record Header and Data.




---

[1]The term “block” is used for a physical record in the description of the RSMS-4 file format.

[2]The term “record” is used for a logical record in the description of the RSMS-4 file format.

## Use, Control, and Documentation of Types

The file format makes use of “flags” (called “types” in this document) to designate portions of the file structure that can be easily changed in the future or that can allow for multiple instances. These types behave like a class in object-oriented implementation in that multiple objects can be instantiated from the class. The types that are used in this file format are the: Block Length Type, File Header Type, User Input Form Type, Record Header Type, Measurement Header Type, Measurement Data Structure Type, and Component Type . The specific instances of these types that are known at this time are described in detail in this document. As new instances of these types are defined, they will be added to the relevant sections of this document. It is imperative that the specific instances of the types that are defined are very tightly controlled and that their usage is absolutely consistent. Once an instance of a type is defined, use of this instance MUST be constrained to only refer to this instance. If one wants to make changes to an instance of a type, a new instance of that type MUST be defined, approved, and then its detailed description added to this document. As an example, Measurement Data Structure Type 501 might define data to be an array of 601 power values from a spectrum analyzer. Once this is defined, approved, and its detailed description added to this document, Measurement Data Structure Type 501 MUST ONLY be used to refer to this specific Measurement Data Structure. Changes to this Measurement Data Structure will require that a new Measurement Data Structure Type be defined (for example, 502).

## II. File Header Type

The File Header Type permits different file headers to be defined if needed or desired. The initial File Header Type is designated as 001 and is described in detail below. Subsequent File Header Types when defined will be included following the description of File Header Type 001 in this document.

File Header Type 001

| |
|---|
| Block Length Type (1 byte character)<br>Block Length (integer)<br>File Header Type (integer)<br>User Input Form Type (integer)<br>Maximum # of Records in File (long)<br>Actual # of Records in File (long)<br># of Blocks of File Header (long)<br># of Records of Calibration Data in File (integer)<br># of Blocks of $n^{th}$ Record (long)<br>**{User Input Form Information}**[3] (To be defined) |
| Block # of Start of $1^{st}$ Record (long)<br>•<br>•<br>•<br>Block # of Start of $n^{th}$ Record (long) |

Block Length Type (1 byte character)

Defined in Section IV of this document. Section IV includes a description of the specific Block Length Types also.

Block Length (integer)

Number of bytes in a block that is specified with the Block Length Type.

File Header Type (integer)

Defined above.

User Input Form Type (integer)

Defined in Section V of this document. Section V includes a description of the specific User Input Form Types also.

Maximum # of Records in File (long)

Allow the user to specify this in RSMS-4 (it is not limited by Visual Basic). DA and SPAM impose a limit of 128 records in a file.

Actual # of Records in File (long)

# of Blocks of File Header (long)

# of Records of Calibration Data in File (integer)

Calibration data is stored in records in the file just like measured data. Calibration data is stored in the first set of records in a file before any of the measured data is stored. The number of records of

[3]In this document, items that are in **bold** and within **{ }** are **NOT** entries in the file, they are headings that describe entries in the file.

calibration data in a file is equivalent to the number of calibrations stored in the file since there is one and only one calibration stored within a given record. There is only one calibration stored within a record because each calibration can have its own path and measurement parameters associated with it.

# of Blocks of $n^{th}$ Record (long)

**{User Input Form Information}**

To be defined.

Block # of Start of $1^{st}$ Record (long)

Pointer (in terms of block #) to the beginning of Record 1

•

•

•

Block # of Start of $n^{th}$ Record (long)

n is the maximum number of records in a file

---

## III. Record Header Type

The Record Header Type permits different record headers to be defined if needed or desired. The initial Record Header Type is designated as 001 and is described in detail below. Subsequent Record Header Types when defined will be included following the description of Record Header Type 001 in this document.

Record Header Type 001

Record Header Type (integer)
Measurement Header Type (integer)
Measurement Data Structure Type (integer)
Total # of Bytes of Measurement Data in this Record (long)
Total # of Nodes in Path (integer)
Total # of Components in Path = M (integer)

**{Measurement Error List}**
Error Code for 1st Error (integer)
•
•
•
Error Code for 20th Error (integer)

# of Calibrations Pertinent to the Data in this Record [L] (integer)
**{Pointer to Calibration Data 1}**
Record # in this Data File (integer)
•
•
•
**{Pointer to Calibration Data L}**
Record # in this Data File (integer)

Scheduler File Name (64 byte string)
Event File Name (64 byte string)
Description of Measurement Data Structure (256 byte string)
Comments (384 byte string)
**{Measurement Header}**

**{Measurement System Architecture}**
**{Component 1}**
Comp. #, Comp. Category, Comp. Type, Model #, Serial #, Input Node, Output Node, Aux Node 1, Aux Node 2
•
•
•
**{Component M}**
Comp. #, Comp. Category, Comp. Type, Model #, Serial #, Input Node, Output Node, Aux Node 1, Aux Node 2

| | |
|---|---|
| **{Component/ Instrument Configuration}** | (128 bytes are allocated for the component/ instrument configuration for each Component 1 to M) |
| **{Component 1}**<br>•<br>•<br>•<br>**{Component M}** | |

Record Header Type (integer)

Defined above.

Measurement Header Type (integer)

Defined in Section VI of this document. Section VI includes a description of the specific Measurement Header Types also.

Measurement Data Structure Type (integer)

Defined in Section VII of this document. Section VII includes a description of the specific Measurement Data Structure Types also.

Total # of Bytes of Measurement Data in this Record (long)

(Recall that Measurement Data can be Calibration Data or Measured Data)

Total # of Nodes in Path (integer)

See description of **{Measurement System Architecture}** below.

Total # of Components in Path = M (integer)

See description of **{Measurement System Architecture}** below.

**{Measurement Error List}**

A maximum of 20 measurement errors that occurred associated with the data stored in this record are listed here. This is a fixed sized array of 20 integers. Errors are listed by their integer error code. Only non-fatal measurement errors are listed here. An error code of 00 represents the condition of no errors.

Error Code for 1$^{st}$ Error (integer)

•

•

•

Error Code for 20$^{th}$ Error (integer)

# of Calibrations Pertinent to the Data in this Record [L] (integer)

This is the total number of calibrations that are useful for the data in this record. (If the data in this record are calibration data, then this value is set to zero.) Calibrations of different types can be present as well as multiple instances of the same type of calibration.

**{Pointer to Calibration Data 1}**

Record # in this Data File (integer)

Each calibration is stored in an individual record. The record where the calibration is stored provides all of the necessary information about the calibration. The record number serves as a pointer to the record where the calibration data is stored.

•

-
-

**{Pointer to Calibration Data L}**

Record # in this Data File (integer)

Scheduler File Name (64 byte string)

Each scheduler file has a sequence of events. Each event describes a particular measurement scenario. Each event is stored in a separate file. The scheduler file name and event name designate a specific measurement that was performed to obtain the data in this particular record. This specific measurement can then be repeated by obtaining the required information from the event within the scheduler file.

Event File Name (64 byte string)

File name of the event that is pertinent to the data stored in this record.

Description of Measurement Data Structure (256 byte string)

Detailed textual description of the measurement data structure in this record for a specific Measurement Data Structure Type. See Section VII of this document.

Comments (384 byte string)

**{Measurement Header}**

See Section VI of this document.

**{Measurement System Architecture}**

The measurement system signal flow path is described in a manner such as used by the SPICE circuit analysis program using components and nodes to describe the interconnection of the components.

**{Component 1}**

Components may have 1 to 4 ports. Ports are labelled as Input and Output with two auxiliary ports for 3 and 4 port components. Ports that are not connected or ports that do not exist for a component are depicted with an integer value of 999 as its node.

Component # (integer)

Component Category (24 byte string)

Component Category is the general category of equipment (for example, spectrum analyzer, oscilloscope, amplifier, etc.).

Component Type (integer)

Component Type is an integer that is used to identify the component/instrument configuration (component/ instrument specific measurement parameters) of a component/instrument. Initially, a single Component Type is defined for each Component Category. However, different Component Types may be used within the same Component Category to facilitate future changes to the component/ instrument configuration for a general category of equipment.

Model # (12 byte string)

Serial # (16 byte string)

Input Node (integer)

Output Node (integer)

Aux Node 1 (integer)

Aux Node 2 (integer)

•
•
•

**{Component M}**

Comp. #, Comp. Category, Comp.Type, Model #, Serial #, Input Node, Output Node, Aux Node 1, Aux Node 2

**{Component/ Instrument Configuration}**

Defined in Section VIII of this document. Section VIII includes a description of the instrument specific measurement parameters for each Component Type that could be used in the signal path.

**{Component 1}**

•
•
•

**{Component M}**

---

## IV. Block Length Type (1 byte character)

The Block Length Type designates a specific block length for this file. The Block Length Type is designated in the file name as well as in the file header. This permits the data analysis software to know HOW to open the file prior to actually opening the file. A longer block length would permit fewer writes to the data file. The maximum block length under Visual Basic 6 is 32,767 bytes. DA and SPAM use 128 byte block lengths. In RSMS-4, the default block length is 128 bytes (this is also the minimum permitted block length) and is designated as Block Length Type A. The software will permit a different block length to be selected (in multiples of 128 bytes only!) up to the maximum of 32,767 bytes. Specific Block Length Types are defined below.

---

Block Length Type A Y Block Length = 128 bytes

---

## V. User Input Form Type

The RSMS-4 software will provide a form in which the user will enter information about the measurements that apply to the entire file. Examples of this information might include site, location, route, receiver antenna height, transmit antenna height, transmit antenna power, etc. This User Input Form Type parameter allows for different forms to be used in the future if needed. Specific User Input Form Types are listed below:

---

User Input Form Type 001= To be determined.

---

## VI. Measurement Header Type

The Measurement Header Type identifier provides a means of changing the Measurement Header in the future. The Measurement Header will contain parameters describing how the measurement is done and the state of the measurement system. One instance of the Measurement Header (Measurement Header Type 001) uses MC Part 1 in DA so that the RSMS-4 file structure will have NO CHANCE in limiting all the functionality that was present in DA. The RSMS-4 file structure will provide provisions for adding to the capability of DA and never detracting from it. Note that MC Part 1 in DA has some reserved space for future use. As in MC Part 1 in DA, the Measurement Header will be a fixed size (1024 bytes). However, while Measurement Header Type 001 uses the single precision floating point representation for each data entry, future Measurement Header Types may use data types that are custom tailored for each data entry.

---

Measurement Header Type 001

Note: All data entries for Measurement Header Type 001 are stored as single precision floating point numbers. Therefore, Measurement Header Type 001 consists of 256 single precision floating point numbers as listed below. Since each single precision floating point number requires 4 bytes of storage, the size of Measurement Header Type 001 is 256 x 4 = 1024 bytes.

000 Volume
001 File Number
002 Record Number
003 Date recorded (yymmdd)
004 Time recorded (hhmmss)
005 MC Last edited Date (yymmdd)
006 MC Last edited Time (hhmmss)
007 Latitude
008 Longitude
009 Altitude
010 Antenna Bearing
011 – 016 Unassigned
017 Current Pass
018 Current Scan
019 Current Sweep
020 Current Step
021 Current Detector
022 Current Antenna
023 – 025 Unassigned
026 Current Frequency
027 Current Time
028 Current Azimuth
029 Current Elevation
030 Total Number of User Channels Measured
031 Channel Bandwidth (kHz)
032 Start Frequency (MHz)
033 Stop Frequency (MHz)
034 Bandwidth/Point (kHz)

035 Start Time
036 Stop Time
037 Incr Time
038 Start Azimuth
039 Stop Azimuth
040 Incr Azimuth
041 Start Elevation
042 Stop Elevation
043 Incr Elevation
044 – 051 Unassigned
052 Number of Passes
053 Number of Scans
054 Number of Sweeps
055 Number of Steps
056 Number of Detector
057 Number of Antenna Steps
058 – 063 Unassigned
064 Graph X Axis Minimum (IF FREQ Scan, Current Scan BF)
065 Graph X Axis Maximum (IF FREQ Scan, Current Scan EF)
066 Graph Y Axis Minimum
067 Graph Y Axis Maximum
068 Number of Points in Scan
069 Number of Series
070 Graph I.D.
071 - 095 Unassigned
096 Kernel ID
097 When Kernel is Executed
  1) On End of Step
  2) On End of Sweep
  3) On End of Scan
  4) On End of Pass
  5) On Alarm
  6) At Start of Measurement
  7) At End of Measurement
098 - 103 Unassigned
104-159 RESERVED FOR KernelS 2-8
160-169 Location (40 chars)
170-179 Test Name (40 chars)
180-189 Misc - Only AVAILABLE in Attended Mode (40 chars)
190 - 191 Unassigned
192-207 Possible processing array locations
208 Receiver Algorithm I.D.
  0) None

1) SWEPT Spectrum
2) Stepped Spectrum, Regular Step
3) Stepped Spectrum, Regular Step, DUAL Analyzer
4) List.

209 Kernel I.D.
SEE KERNTEXT PROCEDURE

210 Manual Mode I.D.
1) Spectrum Analyzer
2) Oscilloscope
3) Pulse Analyzer
4) Modulation Analyzer

211 List RA ID (0-99)

212 Measurement Mode I.D.
01) Strip
02) Sequenced
03) Single
04) Peek/POKE
05) Manual
06) Calibration

213 Measurement TYPE CAN BE THE SAME AS Datatype
214 Standard/Extended Measurement 0/1
215 - 223 Unassigned
224 Threshold #1 Units
225 Threshold #1 Maximum
226 Threshold #1 Minimum
227 Threshold #1 BOUNDS 1)BETWEEN, 2)Outside, 3)>Max, 4)<Min
228 Threshold #1 Qualify True/False ©1/0
229 Threshold #1
230 Threshold #1
231 Threshold #1
232 Threshold #2 Units
233 Threshold #2 Maximum
234 Threshold #2 Minimum
235 Threshold #2 BOUNDS 1)BETWEEN, 2)Outside, 3)>Max, 4)<Min
236 Threshold #2 Qualify True/False (1/0)
237 Threshold #2
238 Threshold #2
239 Threshold #2
240 Nominal System Noise Floor FOR Calibration Bandwidth
241 Nominal Band Gain Correction Factor
242 Nominal Noise Figure
243 Start Calibration Frequency
244 Stop Calibration Frequency

245 Calibration Date
246 Calibration Time
247 - 255 Unassigned

## VII. Measurement Data Structure Type (integer)

This specifies a type of data (with a given data structure) to be stored in the record. This can be calibration data as well as measurement data. Data structure types 1-500 are used for calibration data. Data structure types 501-999 are used for measurement data. For example, Data Structure Type 5xx is measured data and may be the traditional power values from a spectrum analyzer whereas Data Structure Type 5xy is measured data and may be an array of power values and then an array of phase values. Only one type of data with a given data structure is permitted within a given record.

Calibration data can be of different types; it could include the traditional gain vs. frequency calibration from DA and could include log amp characterization, AGC characterization, preselector characterization, etc. The measurement data structure defines just how the calibration data is stored also. For example, the traditional gain vs. frequency calibration data could be stored as frequency and gain paired values for all of the calibration points or as just a list of gain values and a start and stop frequency along with a frequency interval between calibration points.

Measurement Data Structure Type 501
To be determined.

## VIII. Component/ Instrument Configuration

The component/instrument configuration describes component/ instrument specific measurement parameters for each Component Type that could be used in the signal path. As in DA and SPAM, 128 bytes are allocated for the component/ instrument configuration of each Component Type. However, unlike DA and SPAM, only the specific components used in the signal path have their component/ instrument configuration (instrument specific measurement parameters) stored in the record. The instrument specific measurement parameters in the instrument configuration in DA and SPAM used the single precision floating point representation for each data entry. The component/ instrument configuration in RSMS-4 may use data types that are custom tailored for each data entry or may use the single precision floating point representation for each data entry. This is yet to be determined.

The component/ instrument configuration for each different Component Type defined is listed below. As new Component Types are defined for use in RSMS-4, their component/ instrument configurations will be added here. The range of numbers allowed for Component Types are based upon Component Category and will be assigned according to the following table:

| **Component Category** | **Range of Numbers Permitted for Component Type** |
|---|---|

| | |
|---|---|
| Spectrum Analyzers | 001-009 |
| Oscilloscopes | 010-019 |
| Pulse Analyzers | 020-029 |
| Antenna | 030-039 |
| Amplifier | To be determined. |
| | |
| | |
| | |
| | |

---

Component Type: 001  (Component Category: Spectrum Analyzer)

Model #
Sweep Time, Dwell Time
Center Frequency
Frequency Span
Resolution Bandwidth
Video Bandwidth
Attenuation
Reference Level
Detector
1) Normal
2) POSITIVE Peak
3) Sample
4) Negative Peak
5) MXMH
6) VAVG
Y©SCALE Units (dB/DIV)
# of Internal Processing Sweeps FOR VAVG,MXMH
Annotation  On /OFF (0/1)
Trace Mode
Sweep Mode (Boolean, True=Continuous, False=Single)
Calibrated Mode (Boolean, True=Calibrated, False=Uncalibrated)
Unassigned

---

Component Type: 010  (Component Category: Oscilloscope)

Model #
Y Offset
Y Multiplier
X Increment
Wave Form ID
Number of Points
Point Offset
Point Format
X Units
Y Units
Binary Format (0 = RI, 1 = RP)
Encoding (0 = Binary, 1 = ASCII)
X Minimum
X Maximum
Y Minimum
Y Maximum
Trigger Mode
Number of Segments
Current Segment Number
Unassigned

---

Component Type: 020 (Component Category: Pulse Analyzer)

Threshold Type 0/1 Fixed/Relative
Threshold Level
Pulse Width Sort Mode 0/1/2 Disable/Between/Outside
Pulse Width Sort Min
Pulse Width Sort Max
IF Input Attenuation
IF Input SOURCE 0/1/2/3 Narrowband 25/Narrowband 75/
Wideband 500/ External
SHADOW Time
Acquisition Mode
N
Time Interval Between Acquiring Pulses
Noise Riding Threshold Integration Time
Min Amplitude Qualification
Max Amplitude Qualification
Min Frequency Qualification Limit
Max Frequency Qualification Limit
Unassigned

---

Component Type: 030 (Component Category: Antenna)

Antenna Model #
TYPE
1)Any,2)LPA,3)Horn,4)Discone,5)Biconical,6)CBS,7)YAGI,
8)Dish,9)Loop,10)Dipole,11)Monopole
Antenna Mount Type – (0)Fixed 2)Pedestal 3) Spinner)
POLARIZATION
1)Any,2)H,3)V,4)Slant,5)Dual,6)RHC,7)LHC
FMin (MHz)(3 dB)
FMax (MHz)(3 dB)
Vertical Beamwidth (Deg)(Nominal)
Horizontal Beamwidth (Deg)(Nominal)
Gain (dB)(Nominal)
FRONT TO BACK RATIO (dB)(Nominal)
SIDELOBE LEVEL (dB)(Nominal)
Antenna Factor (Ratio)(Nominal)
Unassigned
Unassigned
Current Antenna Selection Index
Unassigned