File Specification for Bio-Rad 1sc files

The information in this document has been gleaned from many hours of detective work and examination of 1sc files.

Many thanks to the team of the Open Microscopy Environment. Their software package Bio-Formats and their implementation of "Bio-Rad Gel 1sc file format" gave me the start in investigating the structure of this file.

Overall File Structure

All known files are little-endian. Seems to be what "Intel Format" at the top of the file is indicating.

The file is made up of a file header from bytes 0 through 4139, followed by 11 contiguous Data Blocks.

File Header

File bytes	Numbers or ASCII	Description
0-1	Numbers	0xAF, 0xAF
2-31	ASCII	`Stable File Version 2.0` \r\n ` ` \r\n
32-55	ASCII	`Intel Format` \r\n
56-95	ASCII	`Bio-Rad Scan File - ID` <17-digit number>
96-135	ASCII	<38 spaces>\r\n
136-139	Numbers	0xC8, 0x00, 0x00, 0x00 (uint16 0x000000C8 = 200)
140-143	Numbers	0x03, 0x00, 0x00, 0x00 (uint32 0x00000003 = 3)
144-147	Numbers	0x00, 0x00, 0x00, 0x00 (uint32 0x00000000 = 0)
148-151	Numbers	0x2C, 0x10, 0x00, 0x00 (uint32 0x0000012C = 300)
152-155	Numbers	uint32 <length of file - 4140> Number of bytes from start of Data Block 0 to End Of File.
156-159	Numbers	0x00, 0x00, 0x01, 0x00 (uint32 0x10000 = 4096)
160-379	Numbers	Data Fields Describing Data Blocks 11x 20-byte Fields
380-4139	Numbers	3760 bytes of 0x00
4140-	Mixed	Start of Data Block 0

Within each Data Block are a series of Data Fields. (See Sections **Field Structure** and **Field Types** for descriptions)

Fields can contain references to other fields, by using a uint32 Data ID to refer to other fields. Each referenceable field has its own unique Data ID recorded in its Field Header.

After byte 4140, the entire file can be parsed as a series of contiguous Data Fields, with special parsing for Field Type 0 (End Of Data Block). If parsing the entire file at once, (and not each Data Block in isolation,) one can use the following method when encountering Field Type 0:

1. Parse the End Of Data Block Field
    - Field Type: 0
    - Field Len: 8

- Field ID: 0

2. Parse the Data Block Footer
    1. Keep reading groups of 7x uint16 values as long as the values satsify: (non-zero, any, 0, any, 0, non-zero, 0)

3. Parse the next Data Block Header
    1. Read 2x uint32 values.

Data Block Structure

Note: Data Block 10, the "Image Data" Data Block, has no Data Block Header, no Data Block Footer, and no Data Fields. It only consists of image data.

All other Data Blocks follow the structure described below.

Data Block Header

The start of each Data Block starts with 2x uint32 numbers.

The first number is the length in bytes of this Data Block Header and all the following Data Block fields, (including the last field, Field Type 0.) It does **not** include the Data Block Footer.

The second number is currently of unknown significance. It has been observed to be one of: 1, 2, 4, 7, 8.

Bytes	Type	Description
0-3	uint32	Data Block Length in Bytes (Header + Fields)
4-7	uint32	Unknown (1, 2, 4, 7, or 8)

Data Block Fields

Following the bytes of the Data Block Header, the fields inside the Data Block are parsed contiguously as normal.

The last field of the Data Block fields is Field Type 0. Field Type 0, Field Len 8 signifies End Of Data Block. This field is only a Field Header--the length of 8 bytes only allows for the length of a Field Header.

Data Block Footer

The data after this Field Type 0 until the end of the Data Block is considered the Data Block Footer.

The footer is an summary of information about the fields seen in this Data Block. It is composed of groups of 14 bytes. Each group summarizes information on a particular Field Type. The groups are in the following format:

Bytes	Type	Description
0-1	uint16	Item 0 Field Type
2-5	uint32	Item 0 Num. Occurrences A
6-9	uint32	Item 0 Num. Occurrences B
10-13	uint32	Item 0 Unknown
14-15	uint16	Item 1 Field Type
...	...	...

"Occurrences A" and "Occurrences B" sum to the total number of occurrences of the Field Type in the Data Block. They must refer to different types of occurrences, but in which way is unknown.

The Unknown field may be the number of times a given Field Type has been referenced in the Data Block.

Field Structure

Each field in the file is composed of an 8-byte Header, followed by data in the Payload.

Field IDs can be different for the same string in different files. They are not consistent across files.

Header

Bytes	Type	Description
0-1	uint16	Field Type
2-3	uint16	Field Length in bytes (including Header bytes) Value of 1 indicates Field Length of 20
4-7	uint32	Field ID

Payload

Bytes	Type	Description
8 - <End Of Field>	byte or uint16 or uint32 or mix	Payload Data

Field Types

Field Referencing Hierarchy

Root types: 102, 1000, 1004, 1015

A single Field Type 16 can be repeatedly referenced by multiple fields.

```
102  ->  101 ->  100 ->  16
    \->  16 \->  16

1015 -> 1008 -> 1007 -> 16
    \-> 1024 -> 1022 -> 16
    \-> 2

1000 -> 1020 -> 1011 -> 1010 -> 1040 -> 131  -> 16
    |       |               |       |       \-> 1000 -> ...
    |       |               |       \-> 1000 -> ...
    |       |               \-> 1000 -> ...
    |       \-> 1000 -> ...
    \-> 1030 -> 1040 -> ...
    |       \-> 1000 -> ...
    \-> 1000 -> ...
    \-> 16
```

NOP Fields

Field Type	Contains References to types	Is Referenced by types	Notes
0	**None**	**None**	End Of Data Block field_id = 0 Data Block Footer and next Data Block Header follows.
2	**None**	1015	nop field? - payload is all 0's, otherwise normal header field_id = one of { 0x1099c4a8, 0x10b9d4a8, 0x10d9e4a8, 0x11e4a4a8, 0x128944a8, 0x144144a8} field_len = 208

Data Block Info Fields

Structure

All Data Block Info Fields have the following structure:

- **NO** references to other fields
- **NOT** referenced by other field
- Field ID = 0
- Field Len = 20 (bytes 2-3 in header uint16 = 1)

Field bytes	Number Format	Description
0-1	uint16	Field Type
2-3	uint16	0x0001 = 1 Field Len of 20
4-7	uint32	0x0000 = 0 Field ID of 0
8-11	uint32	Data Block start Byte offset from start of file.
12-15	uint32	Data Block length Number of bytes in Data Block.
16-17	uint16?	Data Block number? (except 11 for Data Block 0 Info)
18-19	uint16?	Unknown

Field Types

Field Type	Notes
142	Data Block 0 info
143	Data Block 1 info
132	Data Block 2 info
133	Data Block 3 info
141	Data Block 4 info
140	Data Block 5 info
126	Data Block 6 info
127	Data Block 7 info
128	Data Block 8 info
129	Data Block 9 info
130	Data Block 10 info (image data)

String Field

Field Type	Contains References to types	Is Referenced by types	Notes
16	**None**	100, 101, 102, 131, 1000	Previous data fields reference this via Field ID Null-terminated string. (0x00 is always last byte of payload) Field ID: most significant 16-bits are usually one of: 0x0085, 0x0086, 0x0087, 0x0088, 0x008a, 0x014a, 0x014c, 0x014d, 0x0919, 0x091b, 0x1004, 0x1043, 0x1045, 0x107b, 0x107d, 0x1083, 0x1097, 0x1099, 0x10b9, 0x10d9, 0x11e4, 0x1289, 0x1441

Data Container Fields

Field Type 100

Every 36 bytes is a data item, starting at beginning of Field Payload, until end of field. Field ID references are to String Fields later in file.

Num Words, Pointer Byte Offset, and Word Size refer to the payload of a future Field Type 1000 (and possibly other types?) It is unclear how to determine which specific field these pointers refer to.

Field Type	Contains References to types	Is Referenced by types
100	16	101

Field bytes	Number Format	Description
8-11	uint16	Item 0 Unknown0
12-15	uint32	Item 0 Num Words
16-19	uint32	Item 0 Pointer Byte Offset
20-23	uint32	Item 0 Reference to Field Type 16 string
24-27	uint16	Item 0 Unknown1
28-31	uint32	Item 0 Word Size (bytes)
32-33	uint16	Item 0 Unknown2
34-35	uint16	Item 0 Unknown3
36-39	uint16	Item 0 Unknown4
40-43	uint16	Item 0 Unknown5
44-47	uint16	Item 1 Unknown0
...	...	...

Field Type 101

Every 20 bytes is data item until end of field.

Field Type	Contains References to types	Is Referenced by types
101	16, 100	102

Field bytes	Number Format	Description
8-11	uint16	Item 0 Unknown0
12-15	uint16	Item 0 Unknown1
16-19	uint32	Item 0 Reference to Field Type 100
20-23	uint16	Item 0 Unknown2
24-27	uint32	Item 0 Reference to Field Type 16 string
28-31	uint16	Item 1 Unknown0
...	...	...

Field Type 102

A **Root Field** of hierarchy.

Every 16 bytes is data item until end of field. Often (always?) only one data item.

Field Type	Contains References to types	Is Referenced by types
102	16, 101	**None**

Field bytes	Number Format	Description
8-11	uint16	Item 0 Unknown0
12-15	uint16	Item 0 Unknown1
16-19	uint32	Item 0 Reference to Field Type 101
20-23	uint32	Item 0 Reference to Field Type 16 string
24-27	uint16	Item 1 Unknown0
...	...	...

Field Type 131

Every 12 bytes is data item. Bytes 4-7 are uint32 Field ID reference

Field Type	Contains References to types	Is Referenced by types
131	16, 1000	1040

Field bytes	Number Format	Description
8-11	uint32	Item 0 Reference to Field Type 1000
12-15	uint32	Item 0 Reference to Field Type 16 string
16-19	uint16	Item 0 Unknown
20-23	uint32	Item 1 Reference to Field Type 1000
...	...	...

Field Type 1000

A **Root Field** of hierarchy. Can be involved in **circular references** that eventually refer back to self. Data in this field pointed to from data in Field Type 100 (and other types?) Is format fixed based on which data block?

At least in one instance, it seems the last 24 bytes of this Field are not pointed to by other fields (??)

Field Type	Contains References to types	Is Referenced by types
1000	16, 1000, 1020, 1030,	131, 1000, 1010, 1020, 1030, 1040

Other Fields

Field Type	Contains References to types	Is Referenced by types	Notes
1004	**None**	**None**	Root field of hierarchy. payload is all 0's, otherwise normal header NOP field?
1007	16	1008	
1008	1007	1015	
1010	1000, 1040	1011	
1011	1010	1020	
1015	1008, 1024, 2	**None**	
1020	1000, 1011	1000	
1022	16	1024	No data items, only Field ID tags? 4 uints in payload, first 3 uints are Field ID tags. Every 4 bytes is data item, last 4 bytes are not used (??) Bytes 0-3 are uint32 Field ID tag
1024	1022	1015	
1030	1000, 1040	1000	
1040	131, 1000	1010, 1030	

List of Data Blocks

Data Block 0

Field Types: 16, 100, 101, 102

```
Strings (field_type=16):
    eType, color, where, parentIndex, start, end, startArrow, endArrow,
    rotationAngle, orientation, runs, alignment, bkgColor, bTransparentBkg,
    volumeDataPtr, lassoPtr, OverlaySave, x, y, OverImgloc, first, last,
    OverImgbox, array, avail, used, regressionType, OverlaySaveArray,
    string, font, fontFace, fontSize, color, scriptStyle, isBold, isItalic,
    isUnderlined, OverTextRun, array, avail, used, OverTextRunArray,
    sumTotal, sumBorders, numPixels, numPixelsBorders, minPixelValue,
    maxPixelValue, stdDeviation, concentration, type, hasUserLabel, string,
    overlaySavePtr, OverVolumeData, start, bounds, nsteps, swused, swavail,
    steps, integden, pixcnt, maxpix, minpix, OverLasso, Overlay Header
```

Data Block 1

Field Types: 1004

Data Block 2

Field Types: 16, 100, 101, 102

```
Strings (field_type=16):
    file_ver, stripe, notes, nt_used, nt_avail, stdname, stdunits, stdtype,
    blotrows, blotcols, smplwidth, bkgden, bkgtype, calcflags, nbacklog,
    backlog, tdisp_md, lbkg_md, lbkg_disk, lbkg_window, sensitivity,
    min_peak, noise_filter, shoulder_sens, size_scale, normalize,
    use_bandlimit, shadow, lbkg_flags, bandlimit, tolerance, match_flags,
    qcused, qcavail, calcurves, qtyunits, vntr_ambig, flank, repeat,
    vntr_flags, sim_flags, sim_tolerance, sim_required, asl_used, asl_avail,
    as_links, allele_set_code, db_name, db_path, db_filename, db_id,
    mod_time, taglist, db_gelnum, db_unit, mobilmap, db_update, db_type,
    adb_gelnum, adb_unit, adb_taglist, flags, bstyle, difdsp, lanes, lnused,
    lnavail, nxties, nyties, nties, ties, Gel, dens, denused, denavail,
    bkgbox, minimum, average, maximum, Stripe, name, nyties, crossings,
    segtrace, segused, segavail, bands, bandused, bandavail, gpk, gaussused,
    gaussavail, dentrace, stdlanenum, right_stdlanenum, right_frac,
    smplwidth, lanenum, flags, calcflags, sumden, sumd_bands, lbkg_disk,
    lbkg_window, lbkg_flags, dtparm, db_sample, db_band_set, db_standard,
    dmt_used, dmt_avail, db_mobil, db_bset_flags, adb_band_set, adb_sample,
    lbkg_md, Lane, lane pointer, Lane Pointer, dvused, dvavail, dvals,
    srcstrace, navg, min, max, avg, bkdvals, gaussdvused, gaussdvavail,
    gaussdvals, Trace, diag, xaxis, yaxis, data, srctrace, dsttrace,
    lanenum, datawidth, firstden, max, Tdiag, name, sumden, rf, stdval,
    quality, norm_den, calnum, qty, this, first, peak, last, maxpix, minpix,
    lasso, db_btp_code, db_btp_flags, adb_btp_code, adb_btp_flags,
    stdsource, flags, qtysource, Band, band pointer, Band Pointer, start,
    bounds, nsteps, swused, swavail, steps, integden, pixcnt, maxpix,
    minpix, Lasso, lanenum, Bandnum, Band Link, x, y, Imgloc, first, last,
    Imgbox, unowned band pointer, Band Pointer, name, desc, from, cbused,
    cbavail, calbands, ninterp, intps, slope, intercept, corr_coef, calnum,
    mcode, model, extrapolate, status, type, named, Calcurve,
    calcurve pointer, Calcurve Pointer, band, measure, qty, reldev,
    dilution, dilution_txt, qtysource, relstat, Calband, measure, qty,
    Calintp, left, ax, Crosstie, x, y, Crdloc, a, r, Stretcloc, rf,
    mobility, bst_idx, btp_code, MobilTie, name, id_safety, allele_set,
    als_item, AlleleSetLink, sensitivity, min_peak, noise_filter,
    shoulder_sens, size_scale, normalize, use_bandlimit, shadow, bandlimit,
    UserDetect, type, minden, maxden, BackLog, head, tail, text_start, text,
    flags, Note, pr_code, vl_code, tag, used, avail, tags, taglist, std,
    mobility, StandardTie, lanenum, used, stdties, MobilMap, mode, ratio,
    differ, DifDsp Layout, center, sigma, height, gauerr, lolim, hilim,
    GaussPeak, gspk pointer, GaussPeak Pointer, Q1 Description,
```

Data Block 3

Field Types: 16, 1000

```
Strings (field_type=16):
    Mol. Wt.
    KDa
    <filename1>
    <full_path_directory>
    <filename2>
```

Data Block 4

Field Types: 16, 100, 101, 102

```
Strings (field_type=16):
    pr_code, vl_code, tag, used, avail, tags, taglist, references, decode,
    tag_value, prompt, references, used, avail, values, tagdef, used, avail,
    tagdefs, tagdef_list, quality, std_value, norm_den, btp_code, flags,
    peak, band, bands_used, bands_avail, bands, sample_code, bst_code,
    flags, dentrace, dmt_used, dmt_avail, db_mobil, lane, path, filename,
    id, name, description, cre_time, mod_time, update, lanes_used,
    lanes_avail, lanes, taglist, mobilmap, lanewidth, detection, unit, gidx,
    stdtype, lbkg_md, lbkg_disk, lbkg_status, layout, gel, gel pointer,
    gel pointer, name, cre_time, description, taglist, idx_used, idx_avail,
    indices, flags, sample, sample pointer, sample pointer, name, btp_code,
    index, gidx, lanenum, low_std, ideal_std, high_std, low_sf, ideal_sf,
    high_sf, band_type, name, cre_time, mod_time, idx_used, idx_avail,
    index, comment, id, tolerance, bst_idx, bt_used, bt_avail, bt_valid,
    band_types, taglist, tagdefs, unit, norm_btp_code, gidx, lanenum,
    method, modified, code_style, display_names, report_names, type,
    unit_change, model_vers, band set, band set pointer, band set pointer,
    name, description, cre_time, mod_time, id, pathname, gels_used,
    gels_avail, gels, gel_sorting, gel_sort_tag, gel_count, gtpl_used,
    gtpl_avail, gtpl_count, gel_templates, smpl_used, smpl_avail, samples,
    sample_sorting, sample_count, bst_used, bst_avail, band_sets,
    bst_sorting, bst_count, srch_used, srch_avail, srch_count, searches,
    tagdef_list, layouts, units_used, units_avail, units, pop_used,
    pop_avail, pop_count, pop_links, seg_map, db_type, base, sum, gel_list,
    sample_detail, sample_list, gel_detail, bset, srch, odrep, dbp, difdsp,
    detect, layouts, sel_name, sel_date_from, sel_date_to, sel_tag1,
    sel_tag2, sort_by, lst_pr_code, dbpos, gel_list_layout, tagdefs, dbpos,
    sample_detail_layout, sel_tagdef1, sel_tagdef2, lst_tagdef1,
    lst_tagdef2, sort_by, dbpos, sample_list_layout, gel_tagdef1,
    gel_tagdef2, sample_tagdef1, sample_tagdef2, sort_by, flags, dbpos,
    geldet_layout, unit, tagdefs, default_bset, lg_dbpos, sm_dbpos,
```

```
    bset_layout, longname, shortname, unitname, interp, order, flags, unit,
    unit pointer, unit pointer, gidx, lanenum, bst_idx, reference lane,
    name, smplname, date_from, date_to, taglist, tagdefs, match, ref_smpl,
    match_percent, nlanes, ref_lanes, srchnum, search_by, compare,
    sim_method, weighting, edited, include, useGaussModelsIfPresent, search,
    search pointer, search pointer, match_percent, srchnum, tagdefs,
    sim_method, include, weighting, dbpos, search layout, gidx, lanenum,
    bst_idx, lane index, name, plidx, dir_block, data_block, pop link,
    poplink pointer, pop link pointer, first, nsegs, segs, segment map,
    type, value, dbp_pr_coldata_fields, ref_lnum, cols_used, coldata, flags,
    font, pr layout, style, lg_dbpos, sm_dbpos, sum layout, x, y, imgloc,
    x, y, imgres, loc, size, flags, ddb position, dp_pos, method,
    dbp ptree layout, dp_pos, dbp pca layout, dp_pos, dbp popfrm layout,
    popfrm, pr, ptree, pca, irp, dbp layouts, cols_used, coldata, ref,
    order, active, style, pg_layout, show_btypes, ruler, ref_lnum,
    irp layout, od_types, odrep layout, lanenum, used, stdties, mobilmap,
    std, mobility, standardtie, mode, ratio, differ, DifDsp Layout, userdet,
    screenloc, lane_width, manual, style, valid, detect layout, sensitivity,
    min_peak, noise_filter, shoulder_sens, size_scale, normalize,
    use_bandlimit, shadow, bandlimit, userdetect, dvused, dvavail, dvals,
    srctrace, navg, min, max, avg, bkdvals, gaussdvused, gaussdvavail,
    gaussdvals, gaussmax, gaussmin, dentrace, first, last, imgbox, rf,
    mobility, bst_idx, btp_code, db_mobil, DDB Description,
```

Data Block 5

Field Types: 2, 16, 10007, 1008, 1015, 1022, 1024

```
Strings (field_type=16):
    <full_path_directory>
    <filename>
    <filename_without_extension> (Raw 1-D I
    Base Pairs, Base Pairs, BP,
    Isoelectric Point, Iso. Pt., pI,
    Isoelectric Point, Iso. Pt., pI
    Molecular Weight, Mol. Wt., KDa
    Normalized Rf, Norm. Rf, NRf
```

Data Block 6

Field Types: 16, 100, 101, 102

```
Strings (field_type=16):
    m_entries, m_userPool, m_descPool, m_appPool, AuditTrail, m_time,
    m_user, m_description, m_details, m_detailX1, m_detailY1, m_detailX2,
    m_detailY2, m_version, m_comment, m_filter, m_locked, AuditTrailEntry,
    AuditTrailEntryPtr, AuditTrailEntryPtr, m_mmvectorList, m_mmvectorUsed,
    m_mmvectorAvail, AuditTrailEntryPtrVector, m_pool, AuditTrailStringPool,
    m_mmvectorList, m_mmvectorUsed, m_mmvectorAvail, AuditTrailStringVector,
    x, y, Imgloc, x, y, Imgres, first, last, Imgbox, x, y, Crdloc, x, y,
    Crdres, first, last, Crdbox, x, y, Crdscale, mincon, maxcon, in, out,
    low_frac, high_frac, state, gamma, aspect, ImgState, center, scale,
    Savemap, m_x, m_y, CRealPoint, m_width, m_height, CRealSize, m_x, m_y,
    CRealDistance, m_start, m_end, CRealLine, m_top, m_left, m_right,
    m_bottom, CRealRect, m_x, m_y, CImagePoint, m_width, m_height,
    CImageSize, m_x, m_y, CImageDistance, m_start, m_end, CImageLine, m_top,
    m_left, m_right, m_bottom, CImageRect, m_x, m_y, CWindowPoint, m_width,
    m_height, CWindowSize, m_x, m_y, CWindowDistance, m_start, m_end,
    CWindowLine, m_top, m_left, m_right, m_bottom, CWindowRect, m_buffer,
    m_length, sm_string, m_buffer, m_length, mm_string, Audit Trail,
```

Data Block 7

Field Types: 16, 131, 1000, 1010, 1011, 1020, 1030, 1040

```
Strings (field_type=16):
    Scanner Name: <name_of_scanner>
    Number of Pixels: (<img_x_size_int_px> x <img_y_size_int_px>)
    Image Area: (<img_x_size_float_mm> mm x <img_y_size_float_mm> mm)
    Scan Memory Size: <img_size_float_kb> Kb
    Old file name: <filename1>
    New file name: <filename2>
    <directory>
    New Image Acquired
    Save As...
    Quantity One <Quanitity_One_version_string>
```

Data Block 8

Field Types: 16, 100, 101, 102

```
Text of and byte pointers to data in data block 9:
    filevers, creation_date, last_use_date, user_id, prog_name, scanner,
    old_description, old_comment, desc, pH_orient, Mr_orient, nxpix, nypix,
    data_fmt, bytes_per_pix, endian, max_OD, pix_at_max_OD,
    img_size_x, img_size_y, min_pix, max_pix, mean_pix, data_ceiling,
    data_floor, cal, formula, imgstate, qinf, params, history, color,
    light_mode, size_mode, norm_pix, bkgd_pix, faint_loc, small_loc,
    large_box, bkgd_box, dtct_parm_name, m_id32, m_scnId, m_imagePK, SCN,

    calfmt, dettyp, isotop, gel_run_date, cnts_loaded, xpo_start_date,
    xpo_length, ScnCalibInfo

    type, units, c_pro, c_exp, ScnFormula

    x, y, ScnImgloc

    first, last, ScnImgbox

    mincon, maxcon, in, out, low_frac, high_frac, state, gamma, aspect,
    ScnImgState

    qty_range, qty_units, blackIsZero, scanner_maxpix, scanner_units,
    scanner_bias, scanner_maxqty, calstep_count, calstep_raw, calstep_qty,
    calstep_qty_offset, gray_response_data, gray_response_len,
    gray_response_factor, ScnQtyInfo

    x, y, ScnCrdloc

    x, y, ScnCrdres

    first, last, ScnCrdbox

    resolution, scan_area, exposure_time, ref_bkg_time, gain_setting,
    light_mode, color, intf_type, size_mode, imaging_mode,
    filter_name1, filter_name2, filter_name3, filter_name4, filter_name5,
    filter_id1, filter_id2, filter_id3, filter_id4, filter_id5,
    laser_name1, laser_name2, laser_name3, laser_name4, laser_name5,
    laser_id1, laser_id2, laser_id3, laser_id4, laser_id5,
    pmt_voltage, dark_type, live_count, app_name, flat_field, ScnParams

    GR_Data, GrayResponseData, Scan Header
```

Data Block 9

Field Types: 1000

```
Data for:
    filevers, creation_date, last_use_date, user_id, prog_name, scanner,
    old_description, old_comment, desc, pH_orient, Mr_orient, nxpix, nypix,
    data_fmt, bytes_per_pix, endian, max_OD, pix_at_max_OD,
    img_size_x, img_size_y, min_pix, max_pix, mean_pix, data_ceiling,
    data_floor, cal, formula, imgstate, qinf, params, history, color,
    light_mode, size_mode, norm_pix, bkgd_pix, faint_loc, small_loc,
    large_box, bkgd_box, dtct_parm_name, m_id32, m_scnId, m_imagePK, SCN,
```

Data Block 10

Only image data, no fields