Galactic Universal Data Interface Specification

Document History:

10-21-94	SCS	Original version (GRAMSUI.TXT).
05-18-95	JHD/BMG	Add NMR log text extensions.
07-13-95	JHD	Add AIA log text extensions.
09-18-96	JHD/PEH	Update NMR log text extensions, reformat to split DRIVER commands from converter log text entries, add sections for other techniques (NIR & UV/VIS), add new DRIVER commands for UV/VIS instruments
10-9-96	JHD/DEA/PEH	Incorporated changes from DEA & PEH to NIR section and DRIVER commands sections respectively.

Table of Contents

Introduction

The Universal Interface Specification is a set of conventions which allow trace data and Array Basic programs to be more easily shared between a wide variety of instruments. A set of Universal Parameters with fixed text names (herein “Log Text Key Names”) is defined for use in the Galactic SPC data file log text block and in setting up for data acquisition. A set of Universal Commands for the DRIVER statement is defined for use in Galactic-derived collect programs (primarily for GRAMS/386 and GRAMS/32, but may extend to future products as well) so that independent Array Basic programs can be written to collect data from a variety of instruments.

Note that the DRIVER Universal Parameter specifications are almost entirely optional and Galactic data acquisition drivers need not support all possible universal DRIVER commands or all possible parameter settings. Galactic Array Basic programs calling DRIVER commands or using the log text information should be written to work even if commands or parameters are not supported unless absolutely required for proper representation of the data. However, the goal of universal data interchange requires that as much information as possible be preserved in the SPC file log text file block.

Note that one goal of this specification is to make Array Basic applications that make use of DRIVER commands for acquiring data portable between instruments from different vendors. This requires that a minimal set of DRIVER commands be supported in all compliant drivers.

In addition, this document defines a set of standard log text key names for data files that may require technique specific parameters to more completely describe the data contained within the Galactic SPC file. This is primarily for use with Galactic-written data file converters, but may be placed in the log text block by software from other companies who write/convert data files to the Galactic SPC file format or by Galactic Array basic programs. Note that most of these parameters are optional, and Galactic file converters need not fill in all possible parameters in the log text block unless otherwise noted by the technique (i.e. FT-IR, NIR, UV/VIS, NMR, etc.) specification in this document. This set of key names is designed to take care of the following situations in an attempt to maintain data file “completeness” when it is brought into the Galactic software environment:

1. Data acquisition parameters used on the instrument to collect the data; examples are the Universal Parameters used for the DRIVER statement in the above case, or similar parameters from a foreign file imported with a Galactic-written file converter.
2. Data processing parameters required by Array Basic program to properly manipulate and/or display the data (i.e. NMR spectrum shift reference).
3. Other relevant data that may exist in a foreign file for which there is no corresponding structure in the Galactic SPC file format for which access via Array Basic programs is required.
4. Other relevant data that may exist in a foreign file which may be required if the file is to be exported back to the exact same foreign format.

Galactic SPC File Log Text Completeness Goal

Galactic data acquisition and data converters should fill in as much information in the SPC file header as they can. They should use standard Galactic SPC header items where available (i.e. date, time, memo) and log text settings otherwise. The goal of universal data interchangability can only be achieved when Galactic software does not lose information when data is created or converted.

Universal Parameters

Universal Parameters are text names which have an associated value. The value is either a number or a text string. A particular parameter always uses either a numeric value or a quoted text value (although in log text, numbers may optionally be followed by text after a space). Most parameters use numbers. Text parameters have descriptions below which specify whether they are names or text and which specify their possible text string values within quotes.

Note that parameters defined as "names", "text" or "strings" require string values. Also, parameters with multiple choices where the choices are shown as quoted text require string values. Examples are NAME, MEMO, NAMEBG, SPEED, DET, SRC, BSPLITTER, USER, COMMENT, MODEL, IVER, APOD, PHASE, etc. All other parameters require numeric values. Text parameters may still have numeric values as long as the numbers are represented as text.

Other parameter names are possible, and not everyone need use any or all of the Universal Parameters, but the Universal Parameter names should never be used for other than their defined meanings.

Universal Parameters are used in trace file header log text to specify data characteristics which may not be supported in the standard header. The standard trace header remembers the X limits, number of points, number of sub-files, the axis types, resolution text, memo text, and source text. These same settings can optionally be duplicated in the log text with Universal Parameters. Moreover, other settings such as the number of scans, scan speed, laser frequency, and so on can only be recorded in the log text.

```
Scans = 10
Speed = 15
LWN = 15799.7
```

Universal Parameters are also used by data acquisition programs for the DRIVER "IGET" and DRIVER "IPUT" Array Basic interfaces. Again, the DRIVER statement may support additional parameters and need not support all of them, but those that it does should use the defined meanings. However, any unsupported command within the document should return an error code in the _V variable.

Note that there are many potential ways to express the limits for an equally-spaced Y-only trace. Each seems more natural in certain situations. This is due to the fact that there are four relevant settings BEGX (the X position or frequency of the first point), ENDX (X position of last point), INCX (X increment per point), and NPTS (number of points). However only three are independent and the fourth can be derived from the others. For example, BEGX, INCX, and NPTS is used internally in GRAMS/386 and GRAMS/32. However it is important that the Universal Parameters always conform to a single system: BEGX, ENDX, and NPTS. To derive the increment use:

```
INCX = (ENDX-BEGX)/(NPTS-1).
```

Numeric parameter quantities have inherent units. However, the units are not always expressed. In the log text, units are optional but must be separated from the number by a space character:

```
BEGX = 4000 cm-1
```

When setting data acquisition parameters with DRIVER, the units cannot be included:

```
DRIVER "IPUT BEGX", 4000
```

General Header Parameters

The following parameter names may optionally end in the digits "1" through "9". Normally, though, they are used without an ending digit (which should be thought of as the implied "0" primary data). Note that these are duplicates of parameters that are already part of the standard SPC file header. These are primarily provided for DRIVER “IPUT” commands to set up data acquisition from Array Basic programs and do not normally appear in the log text block.

BEGX	X position of first trace point
ENDX	X position of last trace point
NPTS	Number of trace points (0=XY pairs)
BEGZ	Z position of first subfile
ENDZ	Z position of last subfile
NSUBS	Number of subfiles
XTYPE	X axis type number (as defined in SPC.INC)
YTYPE	Y axis type number (as defined in SPC.INC)
ZTYPE	Z axis type number (as defined in SPC.INC)
RES	X Resolution (not the point spacing if FTIR) (-1 = automatic or unknown)
NAME	Trace File Name (optionally including path and extension)
MEMO	Primary Comment text (up to 129 bytes)

Instrument & Acquisition Parameters

The remaining parameters cannot normally have ending digits. Note that multiple choice parameters have text choices which are not case sensitive so that they can be easily viewed from log text. The possible choices are shown below in quotes. The order of the choices is significant since "IGET" returns the choice number (0=first, 1=second, etc.) in the specified variable.

SCANS	Number of co-added scans
SCANSBG	Background scans ratioed against (number of scans in backgd)
NAMEBG	Background or baseline spectrum filename
NAMESTD	Standard reference correction filename
NAMEZR	Zero reference (dark reference) correction filename
GAIN	Detector Gain Factor
SPEED	Scan Speed or Velocity text (may be text like "Slow")
VELOCITY	Scan Speed or Velocity number (-1 = automatic or unknown)
LWN	Laser Wavenumber (normally 15799.7 cm-1)
RAMANFREQ	Raman Laser Frequency (normally in X axis units)
RAMANPWR	Raman Laser Power
JSTOP	J-Stop aperture width
BSTOP	B-Stop aperture width
SLIT1	Slit aperture width (-1 = automatic or unknown)
SLIT2	Slit aperture width (-1 = automatic or unknown)
DET	Detector type name
SRC	Source type name
BSPLITTER	Beam Splitter name
PURGE	Purge Delay before scanning
USER	User's or Analyst's name
COMMENT	Secondary comment (use MEMO first)
MODEL	The unique instrument name without spaces. This is used in log text and with "IGET" to determine the type of instrument or data acquisition or the file converter name. By convention "MODEL=name" is placed at the beginning of log text for a file.
IVER	Instrument or Instrument Software Version text
DVER	Driver version number or file converter version
GVER	GRAMS/386 or GRAMS/32 EXE version number
ZFF	Zero Filling Factor (FTIR)
IRMODE	FTIR Laser Sampling Frequency and Raman Mode 2 = Mid-IR mode 1 = Near-IR mode -2 = Raman-shift (negative => Raman)

APOD	Apodization function type name: "None" (boxcar apodization) "Triangular" "Weak" Norton-Beer "Medium" Norton-Beer "Strong" Norton-Beer "Happ-Genzel" "Bessel" "Cosine" "Filler" "Kaiser" Bessel "Gaussian" "Tradezoidal"
PHASE	Type of Phase Correction type name "Self" "Magnitude" "Background" "Stored" Phase "None"
PHASEPTS	Number of phase correction interferogram points
IGRAMSIDE	Interferogram mode: "Double" sided "Single" sided "Left" single sided "Right" single sided
IGRAMDIR	Interferogram collection direction "Bi" interferometer scans in both direction "Uni" interferometer scans in one direction "Forward" uni-directional "Reverse" uni-directional
POLARIZER	Polarizer Angle
LOWPASS	Low Pass Filter Cutoff
HIGHPASS	High Pass Filter Cutoff
FILTER	Optical Filter Name
SMOOTH	Number of averaged readings per data point.
AVGTIME	Time for A/D averaging of each point reading (msec). If the A/D has a fixed time and points readings are averaged, then this time should be divided by the approximate A/D time to get the number of readings.
BEAM	Single beam mode (FT-IR): "Sample" (Does not replace existing background). “Background" for future ratioing with sample.
BEAM	Beam mode (UV/VIS/NIR): “Double” “Single Front” “Single Rear” “Double Reverse” “Dual Single”
DEMO	Set to 1 only when data file created by demo driver
BDELAY	Begin Delay in seconds
SDELAY	Scan delay in seconds
PMT	Photo multiplier Tube voltage (-1 = automatic or unknown)
DETCHG	Detector changeover (NIR to UV/VIS) wavelength in nm
DETCOR	Detector correction 0 = off

	1 = on
SRCCHG	Source changeover wavelength in nm
GRTCHG	Grating changeover wavelength in nm
INDEPENDENT	Spectral range independent collection: 0 = off, use same instrument parameters for whole spectrum 1 = allow different parameters for UV/VIS and NIR range
SCANMODE	Scan type: “Spectrum” (0) “Time scan” (1) “Multi-wl time” scan (2)
SIGNOISE	Signal to noise processing 0 = no S/N processing 1 = use S/N processing
SNLEVEL	Acceptable S/N ratio
SNTIMEOUT	Time out (in sec) for S/N processing in scan
NIR_RES	NIR data interval (INDEPENDENT = 1 only)
NIR_AVERAGING	NIR averaging time (INDEPENDENT = 1 only)
NIR_SBW	NIR slit width (INDEPENDENT = 1 only)
NIR_ENERGY	NIR energy level (INDEPENDENT = 1 only)
NIR_SLITHT	NIR slit height (INDEPENDENT = 1 only): 0 = Full 1 = 1/3 height
CORRECTION	Sample baseline correction: 0 = off 1 = on
CORR_TYPE	Type of correction: “Normal” (0) “Zero line” (1) “Zero SRA” (2) “Zero*Std.Ref.” (3) (uses NAMESTD file)
SCANTYPE	Scan type: “Sample” (0) “Baseline” (1) “Zero line” (2)

DRIVER Command Specific Parameters

The following parameters may be used by DRIVER statements and are not normally seen in the log text block:

FILES	Number of separate trace files acquired in a run. This is used in techniques such as GCIR which produce more than one data set in a run. The main data (normally a multi-file spectrum) uses parameters without ending numbers. The other data sets (often chromatograms) use ending numbers as in BEGX1, ENDX1, NAME1, etc.
WAIT	0 = Continue with next statement after starting run 1 = Wait for run to end before next statement (default)
HIDE	0 = Monitor the run in the view (default) 1 = Do not monitor the run (leave hidden in background) 2 = Run instrument "detached" collecting to its own memory.
WARN	0 = Do not warn about existing file with same name 1 = Show file overwrite warning if needed (default) 2 = Increment or modify name if needed to avoid overwrite
SAVE	0 = Ask user at end of run 1 = Save to disk 2 = Keep in memory 3 = Both Save to disk and keep in memory
GRATING	0 = Use primary grating (emission for fluorescence) 1 = Use secondary grating (excitation for fluorescence)
POSTAB	Specifies an Array Basic program name (or null for none) to be started at the end of an "ISCAN" run. The program is run after all normally-required processing is complete for the data (but before it is saved to disk for non-multifile data). If null (with zero in the first character) then no extra special program is run. When a POSTAB program is defined, most drivers use it even after normal runs not initiated via DRIVER.
REGION	Instrument's internal storage region.
STATUS	Instrument status: 0 = instrument ready 1 = instrument busy -1 = communications error -2 = abort current operation, discard data (IPUT only) -3 = stop current operation, save data (IPUT only) WARNING: The -3 "stop" is superseded by DRIVER "IHALT"!!
LAMP1	Turn on/off main lamp (or UV lamp): 0 = off 1 = on
LAMP2	Turn off/on 2nd lamp (or VIS lamp)
LAMP3	Turn off/on 3rd lamp (or NIR lamp)
YMIN, YMAX, XMIN, XMAX	Set default display limits for monitoring scan during collect
AUTOSCALE	Autoscale in Y direction during collection: 0 = off 1 = on NOTE: When AUTOSCALE is “on”, then values of YMIN, YMAX are ignored.

Note that most FTIR instruments allow for acquisition of Absorbance, Transmission, Kubelka-Munk, Single-Beam, and Interferogram data. To control this, YTYPE should be set to 2, 128, 3, 0, and 1 respectively. 130 should also be accepted for Single-Beam files. To acquire background data (which is normally single-beam), use BACKG=1. However, the some drivers may use different settings according to the menu choices available in the collect setup dialog. All efforts should be taken to insure that the common GRAMS enumerated types used if possible.

There may be up to ten traces collected during a single run in some systems. For example, GCIR may collect spectra, a GS chromatogram, and various windowed chromatograms. Therefore there are ten versions of the first group of parameters. Normally, they are used as is. For example, BEGX=4000. However the parameters for the second through tenth traces have the digits "1" through "9" appended to their names. For example, BEGX1=4000. Note that the extra versions are used only for data acquisition "IPUT" and "IGET". Log header text for the second trace would use only the short names without following digits.

By convention, the primary data for the experiment has no ending digit. By convention, the second trace (with "1") is used in GCIR for the chromatogram which controls peak picking and integration.

Fluorescence instrument are assumed to scan the emission grating for the X axis and the excitation grating for the Z axis. To scan the other way, set GRATING to 1.

Galactic Enumerated Unit Labels

The SPC.INC & SPC.H files define the axis type numbers. However, for convenience, they are duplicated below. Be aware that this list is NOT used in compiling code and the ONLY source of the currently defined unit strings is maintained in these source code header files.

Possible settings for XTYPE and ZTYPE and the corresponding unit label:

0	Arbitrary
1	Wavenumber (cm-1)
2	Micrometers
3	Nanometers
4	Seconds
5	Minutes
6	Hertz
7	Kilohertz
8	Megahertz
9	Mass (M/z)
10	Parts per million
11	Days
12	Years
13	Raman Shift (cm-1)
15	X,Y,Z text labels in fcatxt (old 4Dh version only)
255	Double interferogram (no display labels)

Possible settings for YTYPE and the corresponding unit label:

-1	Reference Arbitrary Energy
0	Arbitrary Intensity
1	Interferrogram
2	Absorbance
3	Kubelka-Munk
4	Counts
5	Volts
6	Degrees
7	milliamps
8	millimeters
9	millivolts
10	Log (1/R)
11	Percent
128	Transmission (ALL TYPES >=128 ARE ASSUMED TO HAVE INVERTED PEAKS!)
129	Reflectance
130	Arbitrary or Single Beam with Valley Peaks

Universal DRIVER Commands

Data acquisition programs normally support the DRIVER statement to allow user-written Array Basic programs to acquire data. This section describes a set of universal commands which should be implemented if possible. A program may choose to implement any or all of these commands and any or all of the parameter settings. They may also implement other commands, parameters, and syntax. The only requirement is that the commands and parameters should never be implemented as anything other than their universal meanings and that they always issue an "anticipated" error (with _V=-1 and ERRORESC 1) for commands and parameters they do not support.

It is strongly recommended that programs which support "ISCAN" also support "ILOCK" and "IMETHOD <name>". Support for running a "IPUT POSTAB" program defined by the caller is also highly recommended. This program should be run after any normal post-processing required by the driver but while the data is still open in memory. Normally this progam is CHAIN'ed from the end of [CQUIT] program.

DRIVER "ILOCK", lock

The lock value should be 1 to grab the DRIVER lock so that the caller has exclusive access to the instrument via the use of DRIVER commands and locks out access to launching other menu items during a collect. There is an error if the driver is already locked such that _V=-5 (operation not allowed) and ERRORESC 1 is issued. The lock value should be 0 to release the lock. For instruments which maintain their own settings outside of GRAMS/386 and which take parameters only in groups (as in a SCAN command), the "ILOCK 1" also serves to get the initial settings in the PC so that various "IPUT..." commands can change them before using "ISCAN" to send them back to the instrument. For such instruments, using "ILOCK" is mandatory and so the caller which is built to run any instrument must always bracket their runs with "ILOCK 1" and "ILOCK 0" even if it is not concerned with sharing issues.

DRIVER "ISCAN"

This performs a data acquisition run with the current parameter settings. Before starting the run, the caller may set parameters using the "IPUT", "IMETHOD", "ISETUP", or "IDEFAULTS" commands. It should use DRIVER "ILOCK",1 before setting parameters (and unlock after the run) so that other callers cannot interfere with its operation. Note that most programs monitor the run in the view and do not return from DRIVER until the run is complete. This may be modified with the HIDE and WAIT parameters.

DRIVER "IDEFAULTS"

Initializes all instrument scan parameters to the defaults dictated by the specific acquisition driver being controlled.

DRIVER "ISETUP"

Allows the user to set parameters in one or more dialogs before returning. This command should launch the actual instrument scan/setup dialog for the instrument being controlled. Sometimes the dialogs have action buttons that are not implemented internally. If so, the _V variable can return 1 to request that a run be started immediately. Normally, _V is set to 0 for OK or Cancel.

DRIVER "IMETHOD <name>"

This sets some or all parameters from a named method. The program can map method names to sets of parameters via any mechanism it desires. The <name> is not always a file name (but it must not have blanks within it). The program should document how methods are set up. Most callers are written to work with a variety of instruments by making up a method for each rather than by setting all of the needed parameters with "IPUT". This allows for instrument idiosyncrasies and settings which are not defined as universal parameters. For most instruments, method is stored in a GRAMS/386 .SET file loaded from the home directory. This allows the calling program to ship with a set of <name>.SET files which it installs into the \GRAMS directory.

DRIVER "IMETHOD", 0 'Download current parameters to instrument
DRIVER "IMETHOD", 1 'Upload parameters from instrument

Some (but not all) drivers may implement these extensions to IMETHOD if they maintain parameters both in PC memory and in the instrument. However, they violate the assumed model that IGET and IPUT control the "instrument" and that differences between the instrument and the PC parameters are automatically handled and synchronized by the driver.

DRIVER "IMOVE", X [,Z]

This moves the grating(s) to a specified location. It is not normally implemented on fast scanning instruments such as FTIR. The caller can then use DRIVER "IREAD",value to read the current Y value at the X or X,Z location.

DRIVER "IREAD", variable

This reads the current instrument detector or other single value. It is typically supported only by grating spectrometers and other instruments (like balances, PH meters, etc) which produce only one value per "run".

DRIVER “IREAD_X”
DRIVER “IREAD_Y”

These commands read the current grating position and response of the instrument. These commands cannot be used during a scan, and will return an error if called while monitoring mode (IMONITOR) is turned on. Returns the values in the units of the current settings for the XTYPE and YTYPE commands.

DRIVER “IZERO”

This command causes the instrument to self-zero. This effectively sets the response value at the current grating position to represent perfect transmission of energy in the beam path.

DRIVER “IMON_X”, variable
DRIVER “IMON_Y”, variable

These commands read the current grating position and response for a scanning instrument. Must call DRIVER “IMONITOR”, 1 first to place the driver in monitoring more. These commands return an error if the driver monitoring more is set to off. Returns the values in the units of the current settings for the XTYPE and YTYPE commands.

DRIVER “IMONITOR”, 0 ‘Monitoring off
DRIVER “IMONITOR”, 1 ‘Monitoring on

Places scanning instrument into monitoring mode where the grating is kept at one position and the response is read as quickly as possible. This command locks out all other DRIVER commands except for IMON_X, IMON_Y and IMOVE. These commands are use to read the current grating position and response respectively.

DRIVER "IPUT <name>", value

This changes the named parameter to the value. Some parameters take numeric values and some take "strings". The <name> is normally one of the Universal Parameters described above (but can also be a custom parameter specific to the instrument). The <name> should be separated from IPUT by a space. If the parameter is not supported, an "ERRORESC 1" with the _V static variable set to 1 is issued. Note that the accepted parameters may still not be used from the menu command interface on some instruments. They are only guaranteed to effect other DRIVER commands such as "ISCAN", "IGET", "IREAD", "IMETHOD", etc.

DRIVER "IGET <name>", variable

This gets the value of a parameter and assigns it to the variable to be returned to the caller. The <name> is normally one of the Universal Parameters described above (but can also be a custom parameter specific to the instrument). The <name> should be separated from IPUT by a space. If the parameter is not supported, an "ERRORESC 1" with the _V static variable set to 1 is issued. For supported multiple choice parameters with text values, the variable must be an array which gets the value text and the _V static variable is set to the value number (0 for first, 1 for second, and so on).

DRIVER "IDIAG", mode

Runs instrument or driver dependent diagnostics as specified by the mode number.

DRIVER "IHALT"

Ends the current operation as soon as possible. This usually means that the data already collected is saved normally but no more data is collected.

DRIVER Implementation and Errors

Note that the DRIVER statement is not obligated to support all of the commands and all of the parameters as long as it follows the error conventions below. The caller must be able to detect whether a command or parameter is supported by trying to use it and then trapping the error if it is not available.

The DRIVER statement is equivalent to a CHAIN into MENU.ABM. It runs a handler program at a label specified by PORTOUT -72,label (which is normally issued in the label 10 initialization program).

To protect the caller's variables, the DRIVER handler in MENU.ABM begins a PROCEDURE statement.The PROCEDURE arguments must be general enough to handle all cases. Arguments which are sometimes numbers and sometimes strings should be specified with ARG(*). The NPTS(ARG) can be used to determine whether a single number was passed on. If NPTS(ARG)>1 then DRIVER normally assumes that the argument is a string.

The implementation should support "anticipated" errors so that the caller can check whether a feature is supported by trying to use it without having an error message shown to the user. Anticipated errors use "ERRORESC 1" which does not present an error dialog to the user when the caller has ONESC enabled.

In order to pass back unsolicited, infrequently used information, DRIVER sets the _V static variable (which, unlike non-static variables, is not preserved through the PROCEDURE). For warnings, _V is set to a negative error code but no ERRORESC is issued. For anticipated errors, _V is set to a negative error code and "ERRORESC 1" is issued. For multiple choice "IGET" parameters, _V is set to the choice number (0=1st) and no ERRORESC is issued. Otherwise, _V is left unchanged. Error code 1 is used for anticipated errors since it does not produce an error dialog. The caller can use ONESC to intercept all errors (but not warnings).

Normally, ONESC interception shows the error message before going to the ONESC label in the caller's program. However ERRORESC 1 shows no error dialog and just goes to the caller's ONESC handler. In general, calls using the “IGET” command should always supply an argument or argument list for returning values. The _V variable should be primarily reserved for returning error codes.

The _V error codes are as follows:
-1 = Not Supported
-2 = Bad Value
-3 = Passive Setting (Normally Warning)
-4 = Value Out of Range (Normally Warning)
-5 = Operation Not Currently Allowed
-6 = Operation Failed (Normally warning).

The _V=-3 "passive setting warning" can be issued by instruments which use the value only to reflect a manual instrument setting but not to control the instrument.

The _V=-4 "out of range warning" can be issued when a setting value is incorrect but has been "moved in" to the closest possible valid setting.

The _V=-5 "not allowed" is issued when ILOCK cannot grab the driver lock and in other driver-defined situations.

Note that programs using ABC can use ?(argument,"string1 string2 ...) to decode argument text names. For commands which have two names (such as "IGET <name>", "IPUT <name>", and "IMETHOD <name>") can use @PA(3) to skip to the <name> text.

Converting Data To and From Galactic SPC Format

When data is converted from foreign file formats into Galactic SPC format, all attempts should be made to include relevant information that is available in the foreign file. This includes filling in the standard SPC header items (i.e. X units, Y units, Date, Time, Resolution Text, Memo Text) as well as all instrumental parameters possible in the log text block.

When data is exported from a Galactic SPC file to a foreign file format, the converter should look for any and all relevant items in the log text block to determine common parameters that can be exported. Note that some foreign formats will not support fields for all the data that may exist in the log text block, thus those parameters should be ignored on export. In addition, some foreign file formats will support many more parameter fields than are available in the parameters in the log text block. In this case, the export must fill in these fields with reasonable defaults that will allow the foreign software to load the data without failure. Finally, when exporting data from a Galactic SPC file with no log text block, or no log text entries that match required parameters in the foreign file format, reasonable defaults should be used for these fields that will again allow the foreign software to at least load the data without failure.

In some cases, it will not be possible to properly export data from a Galactic SPC file to a foreign file format. There are many possible reasons for this:

1. Critical information required for constructing the foreign file is not available in the Galactic SPC file and there are no reasonable default values that can be universally used for all possible data files.
2. The foreign file format is not a “single file” format, but uses complex structures (i.e. multiple directories and fixed file names) to split the information into many different files.
3. The documentation on the foreign file format is not complete enough to determine reasonable default values for required parameters.

There are some parameters that are not required by the Galactic Universal Parameter specification as they are specific only to one technique. In general these are defined only for data that is imported into the Galactic SPC format and will not be used as part of a DRIVER command interface. The format for the main header and binary data is as defined in SPC.INC and SPC.H. The information here only defines the additional information stored in the log text and log binary blocks in Galactic SPC files converted from foreign formats. In addition, all future Galactic-written converters are to follow the GCDLL.H converter specification.

Galactic FT-IR Data Files

Almost all information to be found in foreign FT-IR data file formats is covered by the Universal Parameters “Instrument & Acquisition Parameters” list. As Galactic finds new required parameters, the Universal Parameter list should be extended, rather than adding FT-IR specific Log Text Key Names for converted data only.

Galactic NIR Multifile Data

When at all possible, import/export converters for NIR data should use the Universal Parameters above when applicable. However, many NIR instrument software packages use multi-record file formats where each record contains a spectrum and additional information and/or data related to the individual sample. Since the Galactic SPC multifile format does not allow for easy storage of this information within the subfile records, the extra information should be imported according to the following specifications.

NIR Parameters

NODE	Identifies the instrument connection node number
PPROCxx	Text description of additional processing applied to entire data set. There can be multiple entries allowed and they are NOT required to be in numerical order, and are NOT required to have a complete sequence (i.e. file could have only PPROC2 and PPROC8). Some possible values are, (although others are allowed): "Constant Multiplication" "File Multiplication" "Constant Division" "File Division" "Constant Addition" "File Multiplication" "Constant Subtraction" "File Subtraction" "Back Correlation" "Normalize" "Mean Center" “Variance Scaled” “Smooth” "SG Smooth" "First Derivative" "Second Derivative" "Log Y" “MSC” (Multiplicative Scatter Correction) “SG 1st Derivative” “SG 2nd Derivative” “SNV” (Standard Normal Variate) “SNV + Detrend” “Divide by Wavelength (xxx)” (xxx is wavelength value) “FFT”
CHANNEL	Channel used for acquisition.
SENSORCNT	If the "SENSORCNT=" value is non-zero, each "SUBFILEx=" line is followed by a "SENSORVALx=" line where "x" is the subfile number beginning with “1” for the first subfile.
SENSORNAMEx	Sensor names. If the sensor count is non-zero, then the "SENSORCNT=" label is followed by "SENSORNAME1=" “SENSORNAME2=", etc. up to the number of sensors identified by SENSORCNT.
SUBFILEx	Extra text information for subfile number “x”. This is a set of comma delimited strings in the following format: MM/DD/YY, HH:MM:SS, “Memo text” Example: SUBFILE43= 04/03/1996, 22:31:45, "BEER"
SENSORVALx	Sensor values for subfile number “x” as comma delimited floating point values. Must be SENSORCNT number of values.

There can be (optional) at least one subfile label section for each subfile in the file. If used, the subfile section starts with the label "SUBFILEx=" (where "x" is the subfile number beginning with “1”) followed by the subfile date and time. The date and time are fixed width fields of the form MM/DD/YYYY HH:MM:SS separated by commas. The time should be in 24 hour format. The last item on the "SUBFILEx=" label line is the subfile description in double quotes. If no description is available, "" is used.
If the "SENSORCNT=" value is non-zero, each "SUBFILEx=" line is followed by a "SENSORVALx=" line where "x" is the again subfile number. Following the label are the sensor values as comma delimited text of floating point values.

Constituent Concentration Data

Many NIR instrument file formats store concentration values and constituent name information that is used to create quantitative calibrations from the spectral data. The Galactic SPC file has no room for this information as the number of constituents and thus the size of the extra data is completely dynamic. While it is possible that the Log Text Key Name specifications may be extended in the future to include this information as part of the Galactic SPC file, the current mechanism is to write the information into a Galactic ASCII text Calibration File List (CFL) file. There are currently two formats allowed for the CFL file, however all future converters should only import to the “new” format. When importing the concentration data, the base name of the CFL file should match the base name of the output Galactic SPC data file (i.e. importing the foreign file “SAMPLE.DAT” would generate a Galactic multifile of the spectral data called “SAMPLE.SPC”, and a Galactic CFL file called “SAMPLE.CFL” of the concentration data).

When exporting data to a foreign NIR data file format, the Galactic-written converters should check for the existence of a matching CFL file. If there is a CFL file that matches the base name of the Galactic SPC file being exported, the conversion routine MUST export both the spectral data and concentration data. Also note that export converters will need to support BOTH CFL file formats as there is no guarantee that the CFL file is always in the new format and the user is not expected to manually translate it.

Descriptions of the Galactic CFL file formats follow.

The old CFL format is from previous versions of Galactic's quantitative Add-On applications and is still supported for import. However, this format is no longer used and new foreign file converters and Array Basic applications are only expected to save concentration data into the new CFL format. However, file converters must support the both the old and new CFL file formats on export.

Old CFL Format Example

```
3   =NUMBER OF COMPONENTS
8   =NUMBER OF SAMPLES
Sample Calibration file for PLSplus tutorial - Synthetic test mixtures
A
B
C
1       2       3       4       5       6       7       8       9       10
GOOD #1
.1      .35     .55
GOOD #2
.15     .4      .45
GOOD #3
.2      .45     .35
GOOD #4
.25     .5      .25
GOOD #5
.3      .1      .6
GOOD #6
.35     .15     .5
GOOD #7
.4      .2      .4
GOOD #8
.45     .25     .3
```

In the old CFL file format it is very important that there are NO BLANK LINES, especially at the top of the file.

Line 1 Contains the number of components (M). A comment may follow, however there must be AT LEAST ONE SPACE between the comment and the number.

Line 2 Contains the number of samples (N). A comment may follow, however there must be AT LEAST ONE SPACE between the comment and the number.

Line 3 Contains a comment line.

Line 4 Contains the name of the first component. Continue typing the name of each component on a separate line up to the number of components (M) listed in Line 1. Component names may be up to 15 characters long.

Line 4+M The next line simply labels 10 columns for visual examination.

Line 5+M The rest of the lines contain the sample file name (with NO FILE EXTENSION!), and on the following line(s), the corresponding concentration data. Follow the example as shown. The values may be delimited by a space, comma or tab. If you have more than 10 components in the unknown, the concentration data may continue on the next line but the name of the next file must start on the following line. There must be N (from Line 2) sample name, concentration line pairs in the file.

The new PLSplus/IQ CFL format was designed specifically for users that create calibration lists using text editor or spreadsheet programs. It uses a comma delimited format to store the spectral file names, and concentration values (if applicable). Therefore, it is not necessary to have the number of components and number of samples in the file header.

New CFL Format

```
Sample Calibration file for PLSplus/IQ tutorial - Synthetic test mixtures
File/Comp,Constit A,Constit B,Constit C
GOOD.SPC #1,         0.10,    0.35,    0.55
GOOD.SPC #2,         0.15,    0.40,    0.45
GOOD.SPC #3,         0.20,    0.45,    0.35
GOOD.SPC #4,         0.25,    0.50,    0.25
GOOD.SPC #5,         0.30,    0.10,    0.60
GOOD.SPC #6,         0.35,    0.15,    0.50
GOOD.SPC #7,         0.40,    0.20,    0.40
GOOD.SPC #8,         0.45,    0.25,    0.30
```

In the new CFL file format, it is very import that there are NO BLANK LINES!. The number of samples (N) is determined by counting the number of non-blank lines after the component list line (Line 2). The count is stopped when the first blank line, or a line that has fewer concentration values than the first sample is encountered.

Line 1 File comment (MUST NOT BE BLANK!).

Line 2 Must start with "File/Comp", then list constituent names using commas as separators. All characters between the commas are considered significant, thus ", Stuff ," will result in a constituent name " Stuff ". The number of constituents (M) is determined by the number of comma separated constituent names found on this line. If there are no names following the "File/Comp" leader, then the rest of the file is assumed to have NO CONCENTRATION DATA and there must be no concentration values following the comma after the file name.

Line 3 File name of 1st sample spectrum, followed by the comma separated concentration values of the constituents listed in line 2. Files can optionally have the extension, but if it is not present, then “.SPC” is assumed. When reading concentration values any non-numeric characters (i.e. spaces) between the commas are ignored.

Line N+2 File name and concentrations for last sample spectrum.

Galactic NMR Data Files

Most NMR parameters are entirely unique. The following list of parameters have been defined as an extension for imported NMR data. Note that some parameters are REQUIRED, and must be present in the log text in order for Galactic to properly process the data. Also, some parameters may only be present in processed NMR spectra, and not FID data.

NOTE: These parameters have been defined for 1D NMR data only. This specification will need to be extended if there is a decision made to accommodate 2D NMR data in the future.

NOTE: All the following parameters <u>MUST</u> appear as the first set of information in the Galactic file text header record. In addition, the last item in the NMR part of the log text parameters must be "NMREND=NMREND". This is used by the Galactic NMR processing routines to recognize if additional processing has been applied to the data file since it was last processed. It is also used to flag that there is no more NMR-related information in the log text after that point.

Complex (Imaginary) Data Storage - Binary Log Block

Most NMR instruments generate complex data (data that has arrays of both “real” and “imaginary” numbers). In order to meet the dual requirements that the data must display the “real” part of the spectrum properly for plotting, general manipulation and publication, but retain the “imaginary’ part for future NMR-related manipulations (such as re-phasing), Galactic has designed a split file format.

In all cases, whether the data is an FID or Spectrum, only the “real” portion is stored in the main data block of the SPC file. The imaginary data is stored in an additional binary block as part of the log text block at the end of the file (see SPC.H for more information). Although the main data block in an SPC file can be stored as either 32 bit-block scaled integers or 32-bit IEEE floating point values, the “imaginary” binary data block can ONLY be stored as 32-bit IEEE floats. When loading an SPC file for display of the Y values, Galactic software only reads the data from the main data block, and ignores any information in the log text block.

When creating a Galactic NMR file for SPC data, the `flogoff` value in the SPCHDR structure MUST be set to the byte offset to the beginning of the LOGSTC structure. This normally immediately follows the main binary data block:

```
typedef struct          /* log block header format */
   {
   DWORD logsizd;       /* byte size of disk block */
   DWORD logsizm;       /* byte size of memory block */
   DWORD logtxto;       /* byte offset to text */
   DWORD logbins;       /* byte size of binary area (after logstc*/
   DWORD logdsks;       /* byte size of disk area (after logbins)*/
   char logspar[44];    /* reserved (must be zero) */
   } LOGSTC;
```

The `logsizd` describes the size (in bytes) of the complete log block (LOGSTC, binary block and text data). The `logsizm` should be the `logsizd` size of the block rounded to the next larger power of 4096. The `logtxto` is the byte offset FROM THE BEGINNING OF LOGSTC to the beginning of the log text data. In other words, the binary log data should immediately follow the LOGSTC structure in the file. The `logbins` specifies the size of the binary block (in bytes) that follows LOGSTC.

Note that the size of the main data block and the binary log block MUST be exactly the same. This is required since the Galactic NMR processing routines require that the “real” and “imaginary” array have exactly the same number of data points.

Informational Parameters

These parameters are used to include extra information about the sample and source of the data in the file. These are all optional parameters and are not required for proper processing.

INSTRUM	Text description of the instrument/software
SOLVENT	Text name of the solvent used (i.e. "CDCl3", "D20", "DMSO", "Acetone", "CD30D", "MeOH", etc.)
NUCLEUS	Nucleus name for tuning frequency "1H" for Proton "13C" for Carbon 13 "19F" for Fluorine 19 General format is "AtomicnumberElement" for all others.

Acquisition Parameters

These parameters are generally related to the modes used to collect the data from the NMR instrument. Many of these are REQUIRED since Galactic uses them during data processing.

DETMODE	Detection Quadrature mode used to collect data (REQUIRED) "SIM" for simultaneous (use Complex FFT) "SEQ" for sequential (use Redfield FFT) "REAL" for real-only (use Real FFT)
NUCFREQ	Nucleus tuning frequency in MHz (REQUIRED) (aka Carrier Freq., Observed Freq., Center Freq.)
SW_HZ	Spectral band width measured in Hertz (REQUIRED if SW_ppm not used) (aka Sweep width, Spectral width)
DWELL	Dwell time in Seconds (Should equal 1/SW_Hz)
DELAY	Acquisition delay in Seconds
ACQTIME	Acquisition time in Seconds (REQUIRED) (Should be equal to NPTS*DWELL, or NPTS/SW_HZ.)
SCANS	Actual number of scans performed - as per Universal Parameters (aka Count of actual responses recorded (CT))
REQSCAN	Number of scans requested during acquisition (aka Number of transients (NT), Number of Acquisitions (NA))
FLTFREQ	Filter Frequency in Hertz
PULSWD	Pulse Width in Microseconds
SW_PPM	Spectral band width measured in PPM (REQUIRED if SW_HZ not used) (aka Sweep width, Spectral width)
AB_APPS	Text boolean - File subsequently processed by AB program "TRUE" or "FALSE"
FIDFILE	Text filename (no path!) of Galactic FID file (*.FIG) corresponding to spectrum This tag can only exist if the spectrum or FID was either processed with Galactic's NMR routines OR was read into the NMR routine. In general, only files processed using Galactic's NMR routines will have this field set.
FID	Text boolean Indicates current data type. (REQUIRED) "TRUE" for raw FID data "FALSE" for FFT'd data (REQUIRES Processing Parameters below)
SPC_REAL	Text boolean - REAL data only, no IMAG record. (REQUIRED) "TRUE" or "FALSE"

Processing Parameters

These parameters are generally added to the log text block by the Galactic NMR processing applications. In general, file converters will not be able to get this information from the foreign file format or the values will be different than what is expected by the Galactic processing routines.

DCOFF	Direct Current Offset correction. This is the data point number threshold to use when calculating the DC offset. This value indicates the number of data points, starting from the last value, and ending at the first data point, that is to be used in the DC offset calculation.
ZFF	Zero Filling Factor Text field that is used to indicate then 2^nK value by which to increase the data size by. “NONE” “2xNPTS” (i.e. 2 times the number of data points in the file) “4xNPTS” “8xNPTS” “16xNPTS” “32xNPTS”
APOD	Apodization filter function type name. (Note this is the same key for Universal Parameters, but the list of allowed strings is different.) "NONE" "MATCH_EXP" (Matched Exponential) "MATCH_GAUSS" (Matched Gaussian) "BOXCAR" (Boxcar) "COS_SQ" (Cosine) "SINE_BELL" (Sine) "TRAPEZOID" (Trapezoid) "LORENTZ-GAUSS" (Lorentz-Gauss) "OPT_RES" (Optimal Resolution)

APODP(0)
APODP(1)
	Some apodization filters require one or more fit parameter (REQUIRED if APOD present)

PHMOD	Phase mode used for spectrum (NOT SUPPORTED YET) "NORMAL" (Normal spectrum phasing using PH0,PH1,PV0&PV1) "MAGNITUDE" (Magnitude spectrum = sqrt(Real^2+Imag^2)) "POWER" (Power spectrum = (Real^2+Imag^2)) (Note if PHMOD<>"Normal" then user cannot re-phase.)
PH0	Zero order phase angle (degrees)
PH1	First order phase angle (degrees)
PV0	Zero order pivot (data point #, 0=first)
PV1	First order pivot (data point #, 0=first)
AUTOTYP	Text name of autophasing method used (NOT SUPPORTED)
SPC_REV	Text boolean - Reverse spectrum flag “TRUE” the file has been flipped “FALSE” file read in does not change
DELTA_PPM	Shift in PPM is the value that is subtracted from the full range beginning value (SW_PPM) and ending value (0) resulting in the new range in PPM

Integration Parameters

The following are used to display the integrals for the file when it is loaded into the Galactic NMR processing routines. These are entirely optional, but if one is present, they all must be.

INTBAS	Text boolean - All integral regions were baselined "TRUE" or "FALSE"
INTOFF	Text boolean - Integral regions were offset "TRUE" or "FALSE"
BASE	Text boolean - Interactive baseline indicator "TRUE" or "FALSE"
NORM	Region normalization value
MASK_ON	Text boolean - Use MASK regions to first excluding spectral information for calculating integrals and for display purposes "TRUE" or "FALSE"
IREGNx	Integral parameters for a single integral region. There can be any number of IREGNx log text entries (where “x” is the region number), and they must appear in sequence and in increasing order (i.e. IREGN1, IREGN2, IREGN3, etc.) The data is stored as a list of text values that are space (ASCII 32) delimited. The format is as follows: Left_Edge Right_Edge Area Normalized_Area Multiplier Offset Tilt The edge values must be specified in terms of PPM The Offset and Tilt values are the baseline correction parameters for the integral region. They are specified in terms of a fraction (0-100%) of 1% of the largest Y value in the masked real data array.
MREGNx	Integral masking parameters for a single mask region. There can be any number of MREGNx log text entries (where “x” is the region number), and they must appear in sequence and in increasing order (i.e. MREGN1, MREGN2, MREGN3, etc.)The data is stored as a list of text values that are space (ASCII 32) delimited. The format is as follows: Left_Edge Right_Edge The edge values must be specified in terms of PPM.

Peak Picking Parameters

These are used to display the peak positions for the file when it is loaded into the Galactic NMR processing routines. These are entirely optional, but if the peaktable is present, they all must be with the exception of the PEAK_MODE.

PEAK_TBL	Text filename of Galactic peak table for spectrum. Normally this will be <datafile>.PKN.
PEAK_MODE	Peak picking mode used: “MASK_PKP” presets a masked region where peaks are not to be picked, then auto picks the rest of the spectrum using the threshold and option values. If the user adds values to the table manually, both modes become obsolete.
THRESH	Peak picking threshold value (corresponds to Galactic Threshold Y Level)
SENS	Peak picking sensitivity value (corresponds to Galactic Sensitivity)

SMOOTH	Peak picking smoothing value (corresponds to Galactic Smoothing) Valid only if PICK_TYPE is set to EXTREMA (1)
PICK_TYPE	Peak picking type: 0 = Maximum 1 = Extrema REQUIRES SMOOTH value also
WINDOW	Size of peak window in data points when Adding a new peak. (Window used will actually be +/- WINDOW from the current position).
PK_MARK	Peak mark type: “None” “Arrow” “V” “Numbers”
PK_LABEL	Peak labeling type number. Corresponds to the GRAMS internal Array Basic code number for peak labeling [7 for smart labels].
LBL_FORM	Peak label digits. Integer number of digits after decimal point in peak labels. Allowed values are 1 to 4.
TR_OBJ	Trace box object number. Allowed value is 1 only
PEAK_MASK	Text boolean - Peak marking masking used: “TRUE” the mask is used to block out portions of the spectrum. This is always TRUE if PEAK_MODE=MASK_PKP. “FALSE”
PREGNx	Peak picking masking parameters for a single region. There can be any number of PREGNx log text entries (where “x” is the region number), and they must appear in sequence and in increasing order (i.e. PREGN1, PREGN2, PREGN3, etc.) The data is stored as a list of text values that are space (ASCII 32) delimited. The format is as follows: Left_Edge Right_Edge The edge values must be specified in terms of PPM.

Array Basic NMR processing programs must add the appropriate processing log text entries when manipulating data. This includes FFTing FIDs to spectra, phasing, and referencing. If the log text processing information is already present after import, but the data is FID type, then Galactic written processing retains the parameters imported from the instrument in memory and upon FFTing the FID will automatically apply those values. These values can be re-processed in the software at any point.

NMR FID data must always have the X axis in Seconds, with FFP=0 and FLP=ACQTIME [or NPTS/SW_Hz]. The maximum intensity of the data must be at the FFP end with the decay of the signal going towards the FLP end of the trace.

NMR spectra that have been FFT'd from FIDs must have X axis units in either Hertz (spanning -BANDWD/2 to +BANDWD/2) or PPM. Default is for the axis to be in PPM.

Galactic Data Files Created from AIA Data Interchange Files

There are many other data parameters which may be needed from time to time. When a setting is not available in the above list, the AIA name from the AIA FTIR or Chromatography Data Interchange specifications should be used. Alternatively, a new Galactic parameter name can be devised. However, to avoid duplications, it is important that all names used, either AIA or Galactic, be recorded in a single place (which is this document for now). A Galactic parameter should be used if possible. A new Galactic name should be created only if the equivalent AIA name has not already been used.

In order to assure proper transfer of key AIA sample information from file to file, any AIA names used as keys in the log text should appear between "AIA_info_begin" and "AIA_info_end" entries. The key names should be the AIA names as they appear in the template specification, complete with underbars as needed. For example (taken from an AIA Chromatography V1.0 file):

```
AIA_info_begin
dataset_completeness = C1+C2
aia_template_revision = 1.0
netcdf_revision = 2.0
languages = English
dataset_origin = PE Nelson Turbochrom
operator_name = JDX
company_method_name = gasoline
sample_id = 1
sample_name = Gasoline
sample_injection_volume = 1.00
sample_amount = 1.00
detection_method_name = gasoline
detector_name =
detector_unit = uV
retention_unit = time in seconds
uniform_sampling_flag = Y
autosampler_position = AIA_info_end
```