NeXus Instrument Definitions for Muon Data

This paper presents Revision 8 of a new Muon NeXus Instrument Definition (Version 2), with the original version included for comparison.

This revision was made following discussions with PSI, considering the prototype version 2 definition in anticipation of using the definition as a muon exchange format.

Stephen Cottrell
24 December 2011

Table of Contents

Muon Instrument Definition: Version 1

This data format evolved as a student project with the following outcome:

- CONVERT_NEXUS written to convert MCS raw files to NeXus using this Instrument Definition.
- TMOGGER written to plot temperature and data logs with the same time axis.
- Read Routines provided in Fortran 77 and C for inclusion in user code.
- UDA and WIMDA adapted to read the data format.
- Format written automatically by MuSR and EMU on a central PC disk as an alternative file format.
- Defined in RAL Technical Report: D. Flannery, S.P. Cottrell, P.J.C. King, RAL-TR-2001-029, August 2001.

NXfile

RE	Name	Attribute	Type	Value	Description
	NXfile			Top level class	
		NeXus_version	NX_CHAR	Version of NeXus API used to create file	
		user	NX_CHAR	Scientist who performed experiment	
	run		NXentry		First (and only) entry

NXentry

RE	Name	Attribute	Type	Value	Description
	NXentry			Name of entry	
	IDF_version		NX_INT32	version of IDF that NeXus file conforms to	
	program_name		NX_CHAR	name of creating program – 'MCS'	
		version	NX_CHAR	version of creating program	
	number		NX_INT32	run number	
	title		NX_CHAR	string containing sample, temperature and field	
	notes		NX_CHAR	comment from MCS file	
	analysis		NX_CHAR	type of muon experiment - "muonTD" (muon, time differential)	
	lab		NX_CHAR	origin of experiment – "ISIS" (collected at the ISIS facility)	
	beamline		NX_CHAR	particular beamline used for experiment	
	start_time		NX_CHAR	start time and date of measurement	
	end_time		NX_CHAR	stop time and date of measurement	
	duration		NX_CHAR	calculated duration	
	switching_states		NX_INT32	'1' – Normal data collection, '2' – Red/Green mode	
	user		NXuser		Details of PI
	sample		NXsample		Details of the sample under investigation
	instrument		NXinstrument		Details of the instrument used
	histogram_data_1		NXdata		The data collected
	temperature_log		NXlog		Log of temperature values during run
	event_log		NXlog		Log of events captured during run

NXuser

RE	Name	Attribute	Type	Value	Description
	NXuser			name of user	
	name		NX_CHAR	scientist(s) name	
	experiment_number		NX_CHAR	RB number	

NXlog

RE	Name	Attribute	Type	Value	Description
	NXlog			name of log	
	name		NX_CHAR	name of log	e.g. ‘sample temperature’
		available	NX_INT32	number of values	
	values		NX_FLOAT	log of values	e.g. ‘temperature values obtained from the ‘TLOG’ file
		units	NX_CHAR	units of values	e.g. ‘kelvin’
	time		NX_FLOAT	time stamp of values	from start of run
		units	NX_CHAR	units of time stamps	e.g. ‘seconds’

NXsample

RE	Name	Attribute	Type	Value	Description
	NXsample			sample	
	name		NX_CHAR	sample name	
	temperature		NX_FLOAT	temperature setting	Desired temperature
		units	NX_CHAR	units for temperature	e.g. ‘kelvin’
	magnetic_field		NX_FLOAT	Field setting	Desired magnetic field
		units	NX_CHAR	units for field	e.g. ‘gauss’
	shape		NX_CHAR	sample orientation	
	magnetic_field_state		NX_CHAR	mode	e.g. ‘TF’
	magnetic_field_vector		NX_FLOAT[]	vector describing magnetic field orientation	
		coordinate_system	NX_CHAR	coordinate system	e.g. ‘cartesian’
		units	NX_CHAR	units for field	e.g. ‘gauss’
		available	NX_INT32	available	attribute ‘1’ if field vector is available, otherwise ‘0’
	environment		NX_CHAR	rig	e.g.’CCR’
	temperature_log		NXlink	link to temperature log	

NXinstrument

RE	Name	Attribute	Type	Value	Description
	NXinstrument			name	
	name		NX_CHAR	instrument name	
	detector		NXdetector		
	collimator		NXcollimator		
	beam		NXbeam		

NXdetector

RE	Name	Attribute	Type	Value	Description
	NXdetector			detector name	
	number		NX_INT32	number of detectors	
	orientation		NX_CHAR	detector arrangement, ‘Longitudinal (“l”)’ or ‘Transverse (“t”)’	
	angles		NX_FLOAT32[ ][]	2D array defining detector positions	*(see Note 1)*
		coordinate_system	NX_CHAR	coordinate_sytem	e.g. ‘cartesian’
		available	NX_INT32	available	attribute ‘1’ if angles are available, otherwise ‘0’
	deadtimes		NX_FLOAT32[ ]	1D array of detector deadtime values	*(see Note 2)*
		units	NX_CHAR	‘microseconds’	
		available	NX_INT32	available	attribute ‘1’ if deadtimes are available, otherwise ‘0’

NXcollimator

RE	Name	Attribute	Type	Value	Description
	NXcollimator			collimator name	
	type		NX_CHAR	'slits'	
	aperture		NX_CHAR	slit setting	not defined by MCS

NXbeam

RE	Name	Attribute	Type	Value	Description
	NXbeam			beamline	
	event_log_1		NXlink	link to event log	
	total_counts		NX_FLOAT32	counts	total number of counts
		units	NX_CHAR	'Mev'	
	daereads		NX_INT32	daereads	number of readouts from the DAE
	frames		NX_INT32	frames	ISIS frames collected on each DAE read

NXdata

RE	Name	Attribute	Type	Value	Description
	NXdata			dataset name	*(see note 3)*
	counts		NX_INT32[][]	raw data counts	2D array of counts: (detector number * switching_states) vs. bin
		units	NX_CHAR	'counts'	required by NeXus
		signal	NX_INT32	'1'	attribute to indicate signal to be plotted
		number	NX_INT32	number of histograms	attribute, number of histograms in **NXdata** group *(see Note 4)*
		length	NX_INT32	length of histograms	attribute, length of histograms in bins
		t0_bin	NX_INT32	bin for time zero	attribute, t0 bin value for histograms
		first_good_bin	NX_INT32	first good bin	attribute, first good bin values for histograms
		last_good_bin	NX_INT32	last good bin	attribute, last good bin values for histograms
		offset	NX_FLOAT32	offset	attribute giving offset to centre of 1st bin - 0.5*histogram_resolution
	resolution		NX_INT32		histogram resolution, set to zero if not applicable
		units	NX_CHAR	'picoseconds'	attribute to describe resolution units
	time_zero		NX_INT32		time zero for muon measurements (*see Note 5*)
		units	NX_CHAR	'microseconds'	attribute to describe time_zero units
		available	NX_INT32	available	Attribute, time zero ("0" if not available)
	raw_time		NX_FLOAT32[]	raw time	scale for time axis (raw time) in microseconds
		axis	NX_INT32	'1'	fastest varying index
		primary	NX_INT32	'1'	raw time is the defaul
		units	NX_CHAR	'microseconds'	attribute to describe time units
	corrected_time		NX_FLOAT32[]	corrected time	scale for time axis (corrected for 'histogram_timezero')
		axis	NX_INT32	'1'	fastest varying index (2nd axis)
		units	NX_CHAR	'microseconds'	attribute to describe time units
	grouping		NX_INT32[ng/2]	grouping	1D array defining grouping for histograms in **NXdata** *(see Note 6)*
		available	NX_INT32	available	attribute, number of groups ("0" if

					information not available)
	alpha		NX_FLOAT32[ng/2][3]	available	2D array, alpha, for pairs of groups defined in ‘grouping’ *(see Note 7)*
		available	NX_INT32	available	attribute, number of pairs for which alpha is defined (“0” if not in use)

Notes:

Coordinate systems: Coordinate axes are taken with the z-axis pointing along the beam direction and the x-axis defined as vertically upwards, the origin is taken to be at the centre of the spectrometer (the sample position). Cartesian, cylindrical and spherical polar coordinate systems may be used with angles θ (rotation about the beam axis) and ϕ (rotation about an axis perpendicular to the beam direction) increasing for clockwise rotations viewed from the origin of the coordinate system.

Note 1: 2D array describing detector positions and solid angles. The index of the 1st rank represents the detector number and the first three elements of the 2nd rank define the position in the specified coordinate system. The fourth element of the 2nd rank defines the detector solid angle.

Note 2: The deadtime values for each detector. The values, read from the default deadtime file 'DT(E)PAR.DAT', are stored as a 1D array where the index represents the detector number. The attribute 'deadtimes_available' should be set to "1" if this information is read successfully, otherwise "0".

Note 3: MCS can generate datasets where individual histograms can have different resolutions, lengths, t0 bins, first good bins and last good bins! In practice this feature is rarely used. For efficiency, the muon NeXus instrument definition specifies that sets of histograms where these five parameters are common are grouped into a single NXdata group. Most ISIS muon NeXus files will therefore contain a single NXdata group.

Note 4: The value of the attribute 'histogram_number' should be set equal to the total number of histograms in the NXdata group. The value should reflect any histogram doubling resulting from the Red/Green data collection mode.

Note 5: Not defined in the MCS file, the value should be read from the file 'BASETIME.UDA' if this is available in the current directory. The attribute 'available' should be set accordingly.

Note 6: If there is a grouping file ('long.uda' for longitudinal data and 'trans.uda' for transverse data) in the directory where the NeXus file is being created then this should be used. If this is not available, the default grouping file corresponding to the current instrument orientation should be used. A check should be made that the total number of histograms listed in the grouping file corresponds to the value of 'histogram_number' in the NeXus file, with a mismatch indicating an error. The attribute 'grouping_available' should be set equal to the total number of groups unless grouping information is not available or there is an error condition, when it should be set to "0". The grouping is represented as a 1D array with the index representing the histogram number and the value corresponding to the group into which it should be placed.

Note 7: The balance parameter, alpha, is given for pairs of groups defined in 'grouping'. Values are stored in a 2D array together with the numbers for the forward and backward groups. The attribute 'alpha_available' should be set to the number of alpha values defined or zero if not in use.

Note 8: NXLOG groups can be added as required for any variables logged during an experiment. Logged groups should contain a values array and units and a corresponding time array and units.

Muon Instrument Definition: Version 2 – 'muonTD'

A revision of the Instrument Definition is proposed for the following reasons:

- To correct misunderstandings and misinterpretations of the NeXus format before the definition moves into wider use (PSI are considering adopting NeXus as their RAW file format, and NeXus is a candidate for an exchange file format for sharing muon data between facilities).
- To bring the muon definition closer to that proposed for the neutron instruments at ISIS, from which many of the ideas in this document are taken.
- Where possible, to incorporate ideas and standards set by the NeXus International Advisory Committee (NIAC) (see http://www.nexusformat.org/Main_Page for current discussion). The new definition, however, still requires class definitions not currently defined and/or ratified by the NIAC; these are listed in Appendix 1 in preparation for ratification where applicable.
- To provide a RAW file format suitable for storing data written by PC DAE2 and that, in particular, provides the facility for handling unlimited periods in an efficient fashion.
- To provide a definition that is better able to adapt to the wide range of specialist experiments that now run at RAL. The new definition has not attempted to define all entries required for all possible experiments (e.g. RF, laser, etc), it is anticipated that the basic definition (together with the necessary 'read' routines) will be extended to accommodate the additional information as required. Guidelines are provided to assist the extension of the basic definition in a compatible way.

The Instrument Definition described in this document is uniquely identified by the data entries 'idf_version' ('2') and 'definition' ('muonTD') stored within the 'NXentry' group, and the attribute 'creator' ('ISIS') in the group 'NXroot'. It is specifically designed for the ISIS muon instruments (EC and RIKEN-RAL) for storing raw data files, although it should be equally applicable to other pulsed muon sources (with appropriate localisation). For continuous sources, modifications appear inevitable; however, it is hoped that the majority of the meta-data can be written in a compatible fashion. It is vital for reading programs that any modification to entries in the existing definition should be marked by new values written for the 'definition', 'version' and 'creator' entries. A simple extension of the definition need not yield a new version number, although reading programs would then need to use other methods to discover the additional data. The NXuif group provides a convenient area to store user/experiment specific information.

NXroot

RE	Name	Attribute	Type	Value	Description
	NXroot				top level class
1		**file_name**	NX_CHAR		file name of original data file to assist identification if the external name has been changed
1		**file_time**	ISO 8601		date and time of file creation
0/1		Initial_file_format	NX_CHAR	'HDF(4)' | 'HDF(5)' | 'XML'	Format used when creating initial NeXus file
0/1		file_update_time	ISO 8601		date and time of last file update at close
0/1		NeXus_version	NX_CHAR		version of nexus API used in writing the file
0/1		HDF_version	NX_CHAR		version of HDF library used by nexus to create file
0/1		HDF5_version	NX_CHAR		version of HDF5 library used by nexus to create file
0/1		XML_version	NX_CHAR		version of XML library used to create file
1		**creator**	NX_CHAR		facility or program where file originated
1+	***{raw_data_*}***		NXentry		Entries holding the raw data should follow the defined naming convention and be numbered; entries may also be written containing analysed data etc using a locally defined naming scheme.

NXentry

RE	Name	Attribute	Type	Value	Description
	NXentry				name of entry
1	**idf_version**		NX_INT	'2'	version of IDF that NeXus file confirms to
1	**definition**		NX_CHAR	'muonTD'	the template (DTD name) on which the entry was based, e.g. 'muonTD' (muon, time differential). It's suggested that muon definitions always use the prefix 'muon', with a subsequent sequence of capitals defining the unique function of the definition.
0/1		version	NX_CHAR		DTD version number
0/1		URL	NX_CHAR		URL of XML DTD or schema appropriate for file
0/1	definition_local		NX_CHAR		a template (DTD name) on which an extension to the base definition is based
0/1		version	NX_CHAR		DTD version number
0/1		URL	NX_CHAR		URL of XML DTD or schema appropriate for file
0/1	program_name		NX_CHAR	'SECI' | 'MCS' | 'CONVERT_NEXUS'	name of creating program
0/1		version	NX_CHAR		version of creating program
0/1		configuration	NX_CHAR		configuration of software e.g. SECI configuration
1	**run_number**		NX_INT		run number
1	**title**		NX_CHAR		extended title for the entry, e.g. string containing sample, temperature and field
0/1	notes		NX_CHAR		log of useful stuff about the experiment, supplied by the user
1	**start_time**		ISO 8601		start time and date of measurement
1	**end_time**		ISO 8601		end time and date of measurement
0/1	duration		NX_INT		duration of measurement i.e. (end-start)
0/1		units	NX_CHAR	'second'	
0/1	collection_time		NX_FLOAT		duration of data collection, taking out periods when collection was suspended (e.g.

					because of a beam off or run control veto)
0/1		units	NX_CHAR	‘second’	
0/1	total_counts		NX_INT		total number of detector events
0/1	good_frames		NX_INT		number of proton pulses used (not vetoed)
0/1	raw_frames		NX_INT		number of proton pulses to target
0/1	proton_charge		NX_FLOAT		
0/1		units	NX_CHAR	microAmp*hour	
1	**experiment_identifier**		NX_CHAR		experiment number, for ISIS, the RB number
0/1	run_cycle		NX_CHAR		ISIS cycle
1	user_1		NXuser		details of representative user
0/1+	*{experiment team}*		NXuser		details of members of experiment team, numbered in sequence
0/1	runlog		NXrunlog		container for log(s) of run parameters, inevitably specific to each facility. See Note 0
0/1	selog		NXselog		container for log(s) of sample environment parameters, that may be specific to each experiment.
0/1	periods		NXperiod		log of period parameters, inevitably specific to each facility. See Note 0
1	sample		NXsample		details of the sample under investigation
1	**instrument**		NXinstrument		details of the instrument used
1+	***{detector_*}***		NXdata		the data collected *(see Note 1)*
0/1+	*{characterization}*		NXcharacterization		contains a link calibration data, e.g. silver runs, spot size measurements.
0/1	uif		NXuif		group containing user supplied information contents of group not defined in definition

NXuser

RE	Name	Attribute	Type	Value	Description
	NXuser				name of user
1	name		NX_CHAR		full name of user
0/1		role	NX_CHAR		role of user e.g. ‘PI’, ‘Contact’ etc, multiple roles are allowed.
0/1	affiliation		NX_CHAR		institute
0/1	address		NX_CHAR		address
0/1	telephone_number		NX_CHAR		telephone number
0/1	fax_number		NX_CHAR		FAX
0/1	email		NX_CHAR		email
0/1	facility_user_id		NX_CHAR		unique facility based identifier, at ISIS the User Number

NXrunlog

RE	Name	Attribute	Type	Value	Description
	NXrunlog				
Class to contain action and acquisition logs arising from activities of the data acquisition electronics (DAE) during the run – entries will be facility specific.					

NXselog

RE	Name	Attribute	Type	Value	Description
	NXselog				
Class to contain sample environment logs arising from activities of the experiment control software – entries will be facility specific.					

NXperiod

RE	Name	Attribute	Type	Value	Description
	NXperiod				
1	**number**		NX_INT	‘np’	number of periods used
1	**type**		NX_INT[np]	‘1’ | ‘2’	function of period: ‘1’ – DAQ, ‘2’ – DWELL
1	frames_requested		NX_INT[np]		frames collected in each period before switching, ‘0’ for unlimited frames
1		frame_type	NX_CHAR	‘good’ | ‘raw’	type of frame for period switching
0/1	output		NX_INT[np]		output bit pattern on period card. If not known, write ‘0’ … ‘np’-1 into array
0/1	labels		NX_CHAR		list of period names, separated by character given as attribute. May use a 2D array of NX_CHAR - TBC
0/1		separator	NX_CHAR		separator character for label list
0/1	raw_frames		NX_INT[np]		raw frames collected for each period
0/1	good_frames		NX_INT[np]		good frames collected for each period
0/1	sequences		NX_INT[np]		number of times data collection took place in each period
0/1	counts		NXlog		counts collected in each periods
Additional entries may be stored to clarify Period activity.					

NXsample

RE	Name	Attribute	Type	Value	Description
	NXsample				sample
1	name		NX_CHAR		sample name
0/1	chemical_formula		NX_CHAR		element symbols to be arranged in 'Hill System' order: C, H, then other elements alphabetically
0/1	description		NX_CHAR		description of sample
0/1	type		NX_CHAR	'sample' | 'calibration sample' | 'normalisation sample' | 'simulated data' | 'none' | 'sample environment'	type of sample
0/1	situation		NX_CHAR	'atmosphere' | 'vacuum'	
0/1	preparation_date		ISO 8601		date of preparation of sample
0/1	sample_holder		NX_CHAR		description of sample holder
0/1	flypast		NX_INT	'0' | '1'	flypast (suspended sample) mounting: '0' – No, '1' – Yes
0/1	geometry		NXgeometry		sample size
0/1	sample_component		NX_CHAR[n_comp]		name of each sample componenet
0/1	thickness		NX_FLOAT[n_comp]		sample thickness, may be multiple components
0/1		units	NX_CHAR	'milli.metre'	
0/1	mass		NX_FLOAT[n_comp]		sample mass, may be multiple components
0/1		units	NX_CHAR	'milli.gram'	
0/1	density		NX_FLOAT[n_comp]		sample density, may be multiple components
0/1		units	NX_CHAR	'milli.gram.centimetre-3'	
0/1	temperature		NX_FLOAT		linked to most representative sample temperature (to help cataloguing programs)
0/1	magnetic_field		NX_FLOAT[]		linked to most representative magnetic field (to help cataloguing programs)
0/1	magnetic_field_state		NX_CHAR	'LF' | 'TF' | 'ZF'	current field operating mode
0/1+	temperature_*		NX_FLOAT		temperature *(see Note 2)*
0/1+		units	NX_CHAR	'kelvin'	units for temperature
0/1+		role	NX_CHAR	'control' | 'sample'	function of temperature
0/1+		value	NX_CHAR	'nominal' | 'derived'	.eq.
0/1+	temperature_*_env		NXenvironment		details of associated hardware

0/1+	temperature_*_log		NXlog		temperature log
0/1+	magnetic_field_*		NX_FLOAT[]	field setting	magnetic field, may be a vector
0/1+		units	NX_CHAR	‘gauss’	units for field
0/1+		role	NX_CHAR	‘active’ | ‘inactive’	field status
0/1+		direction	NX_CHAR	‘x’ | ‘y’ | ‘z’	field direction
0/1+	magnetic_field_*_env		NXenvironment		details of associated hardware
0/1+	magnetic_field_*_log		NXlog		log of field values – extension to NeXus to enable log of vector quantities

NXenvironment

RE	Name	Attribute	Type	Value	Description
	NXenvironment				
1	**name**		NX_CHAR		name of apparatus
0/1	short_name		NX_CHAR		name displayed on DAE software
0/1	type		NX_CHAR		short code
0/1	description		NX_CHAR		long description
0/1	program		NX_CHAR		version of driver used to collect data, e.g. VI name and version
0/1+	*{hardware log}*		NXlog		log of hardware parameter relating to apparatus, e.g. temperature controller parameters, etc
0/1+	*{sensor name}*		NXsensor		

NXinstrument

RE	Name	Attribute	Type	Value	Description
	NXinstrument				name
1	**name**		NX_CHAR		instrument name
1	**source**		NXsource		details of the muon source used
1	**beamline**		NXbeamline		beamline description
1+	***{detector_*}***		NXdetector		details of detector used
0/1	dae		NXdae		Details of the DAE used. See note 0.
0/1+	*{aperture}*		NXlink		link to aperture in NXbeamline (naming must be identical) apertures used to collimate beam (only link those that the user is interested in)

NXdata

RE	Name	Attribute	Type	Value	Description
	NXdata				
1	**counts**		NX_INT[ns] | NX_INT[ns][ntc] | NX_INT[np][ns][ntc]		linked to detector counts in NXdetector
1		**signal**	NX_INT	'1'	
1		**axes**	NX_CHAR	'[period_index, spectrum_index, raw_time]'	axes definitions
1		**long_name**	NX_CHAR	'positron counts' | 'electron counts'	
0/1	raw_time		NX_FLOAT[ntc+1]		linked to 'raw_time' in NXdetector
0/1		units	NX_CHAR	'micro.second'	
0/1		long_name	NX_CHAR	'time'	
0/1	spectrum_index		NX_INT[ns]		linked to 'spectrum_index' in NXdetector *(see Note 3)*
0/1		long_name	NX_CHAR	'spectrum number'	
0/1	spectrum_labels		NX_CHAR		linked to semicolon separated list of spectrum names
0/1	period_index		NX_INT[np]		link to 'period_index' in NXdetector
0/1		long_name	NX_CHAR	'period number'	
0/1	period_output		NX_INT[np]		linked to 'period_output' in NXperiod - usually integer representing output bit pattern on period card
0/1	period_labels		NX_CHAR		linked to semicolon separated list of period names

NXcharacterization

RE	Name	Attribute	Type	Value	Description
	NXcharacterization			*{type of characterization}*	name of group should reflect type of characterization
0/1	source		NX_CHAR		source file, if missing then use current file
1	location		NX_CHAR		path to data in file
0/1	mime_type		NX_CHAR		if missing, the source file is NAPI readablee
0/1	definition		NX_CHAR		definition data conforms to
0/1		version	NX_CHAR		version of definition

NXsensor

RE	Name	Attribute	Type	Value	Description
	NXsensor				
0/1	model		NX_CHAR		sensor model number, e.g. ‘RhFe 11074’
1	**name**		NX_CHAR		name
0/1	short_name		NX_CHAR		name displayed on DAE software, e.g. SECI block name
0/1	attached_to		NX_CHAR		where sensor is attached
0/1	measurement		NX_CHAR	‘temperature | pH | magnetic_field | electric field | conductivity | resistance | voltage | pressure | flow | stress | strain | shear | surface_pressure’	type of measurement
0/1	type		NX_CHAR	e.g. ‘Rh/Fe | Cernox | Hall | NMR’	type of probe making the measurement
0/1	run_control		NX_BOOLEAN		is data collection controlled by sensor value
0/1	high_trip_value		NX_FLOAT		if using run control, upper bound
0/1		units	NX_CHAR		
0/1	low_trip_value		NX_FLOAT		if using run control, lower bound
0/1		units	NX_CHAR		
0/1	value		NX_FLOAT[]		sensor value
0/1		units	NX_CHAR		
0/1	value_log		NXlog		log of sensor value

NXsource

RE	Name	Attribute	Type	Value	Description
	NXsource				name of source
1	**name**		NX_CHAR		facility name
1	**type**		NX_CHAR	‘pulsed muon source’ | ‘low energy muon source’	
1	**probe**		NX_CHAR	‘positive muons’ | ‘negative muons’	
0/1	source_frequency		NX_FLOAT		accelerator frequency, note that some frames may be ‘missing’ at target
0/1		units	NX_CHAR	‘hertz’	
0/1	source_frequency_log		NXlog		log of source frequency during run
0/1	source_frame_pattern		NX_INT[rep_len]		frame pattern: ‘1’ frame to target, ‘0’ frame missing at ‘frequency’, e.g. ISIS target 1 with TS2: ‘1,1,1,1,0’, with a ‘rep_len’ of ‘5’ and ‘period’ 100ms.
0/1		rep_len	NX_INT		repetition length of frame pattern in terms of frames to target
0/1		period	NX_FLOAT		period of repetition of frame pattern, e.g. 100ms at ISIS target 1, with TS2
0/1		units	NX_CHAR	‘milli.second’	units of period
0/1		pulses_per_frame	NX_FLOAT		number of pulses for each accelerator frame, e.g. ‘2’ at ISIS
0/1	source_energy		NX_FLOAT		source energy at target
0/1		units	NX_CHAR	‘mega.electronvolt’	
0/1	source_current		NX_FLOAT | NXlog		source current - this could be an average source current for the run, or logged values
0/1		units	NX_CHAR	‘micro.amp’	
0/1	source_current_log		NXlog		log of source current during run
0/1	source_pulse_width		NX_FLOAT		source pulse width
0/1		units	NX_CHAR		
0/1	target_material		NX_CHAR		e.g. ‘carbon’
0/1	target_thickness		NX_FLOAT		thickness of target
0/1		units	NX_CHAR	‘milli.metre’	
0/1	pion_momentum		NX_FLOAT		pion momentum
0/1		units	NX_CHAR	‘mega.electronvolt.c^-1’	
0/1	muon_energy		NX_FLOAT		muon energy
0/1		units	NX_CHAR	‘electronvolt’	
0/1	muon_momentu		NX_FLOAT		muon momentum

	m				
0/1		units	NX_CHAR	‘mega.electronvolt.c^-1’	
0/1	muon_pulse_pattern		NX_FLOAT[rep_len]		pulse pattern – ‘n’ number of pulses to instrument each frame, e.g. ISIS target 1 with TS2: ‘2,2,2,2,0’, with a ‘rep_len’ of ‘5’ and ‘period’ 100ms, assuming no muon kicker.
0/1		rep_len	NX_INT		repetition length of pulse pattern in terms of frames to target
0/1		period	NX_FLOAT		period of repetition of pulse pattern, e.g. 100ms at ISIS target 1, with TS2
0/1		units	NX_CHAR	‘milli.second’	units of period
0/1	muon_pulse_width		NX_FLOAT[]		pulse width for each pulse in frame, e.g. 80ns at ISIS
0/1		units	NX_CHAR	‘nano.second’	
0/1	muon_pulse_separation		NX_FLOAT[]		separation of consecutive pulses in frame, e.g. 300ns at ISIS
0/1		units	NX_CHAR	‘nano.second’	
0/1	notes		NX_CHAR		source related messages or announcements, e.g. MCR messages

NXbeamline

RE	Name	Attribute	Type	Value	Description
	NXbeamline				
1	**beamline**		NX_CHAR		name
0/1+	diagnostics		NXdiagnostics		container for any beamline diagnostic information.
0/1+	*{beamline component}*		NXdipole_magnet | NXquadrupole_magnet | NXsolenoid_magnet | NXkicker | NXseparator | NXspinrotator | NXbeamline_collimator | NXseptum_magnet | NXsteering_magnet		beamline components

NXdiagnostics

RE	Name	Attribute	Type	Value	Description
	NXdiagnostics				diagnostics
1+	type		NX_CHAR		type of diagnostic
Class to contain entries describing diagnostic information relating to the beamline. Pictures, etc, may be included within NXnote's, while logged entries should use NXlog.					

NXdipole_magnet

RE	Name	Attribute	Type	Value	Description
	NXdipole_magnet				component name
1	**description**		NX_CHAR		name
0/1	source_distance		NX_FLOAT		flight path from production target
0/1		units	NX_CHAR	‘metres’	
0/1	set_current		NX_FLOAT		set value on PS
0/1		units	NX_CHAR		
0/1	read_current		NXlog		
0/1		units	NX_CHAR		
0/1	read_voltage		NXlog		
0/1		units	NX_CHAR		

NXquadrupole_magnet

RE	Name	Attribute	Type	Value	Description
	NXquadrupole_ magnet				component name
1	**description**		NX_CHAR		name
0/1	source_distance		NX_FLOAT		flight path from production target
0/1		units	NX_CHAR	‘metres’	
0/1	set_current		NX_FLOAT		set value on PS
0/1		units	NX_CHAR		
0/1	read_current		NXlog		read from PS
0/1		units	NX_CHAR		
0/1	read_voltage		NXlog		read from PS
0/1		units	NX_CHAR		

NXseptum_magnet

RE	Name	Attribute	Type	Value	Description
	NXseptum_magnet				component name
1	**description**		NX_CHAR		name
0/1	source_distance		NX_FLOAT		flight path from production target
0/1		units	NX_CHAR	‘metres’	
0/1	set_current		NX_FLOAT		set value on PS
0/1		units	NX_CHAR		
0/1	read_current		NXlog		value read from PS (may be logged)
0/1		units	NX_CHAR		
0/1	read_voltage		NXlog		read from PS
0/1		units	NX_CHAR		

NXsolenoid_magnet

RE	Name	Attribute	Type	Value	Description
	NXsolenoid_magnet				component name
1	**description**		NX_CHAR		name
0/1	source_distance		NX_FLOAT		distance of flight path from production target
0/1		units	NX_CHAR	‘metres’	
0/1	set_current		NX_FLOAT		set value on PS
0/1		units	NX_CHAR		
0/1	read_current		NXlog		read from PS
0/1		units	NX_CHAR		
0/1	read_voltage		NXlog		read from PS
0/1		units	NX_CHAR		

NXsteering_magnet

RE	Name	Attribute	Type	Value	Description
	NXsteering_magnet				component name
1	**description**		NX_CHAR		name
0/1	source_distance		NX_FLOAT		distance of flight path from production target
0/1		units	NX_CHAR	‘metres’	
0/1	set_current		NX_FLOAT		set value on PS
0/1		units	NX_CHAR		
0/1	read_current		NXlog		read from PS
0/1		units	NX_CHAR		
0/1	read_voltage		NXlog		read from PS
0/1		units	NX_CHAR		

NXseparator

RE	Name	Attribute	Type	Value	Description
	NXseparator				component name
1	**description**		NX_CHAR		name
0/1	source_distance		NX_FLOAT		distance of flight path from production target
0/1		units	NX_CHAR	‘metres’	
0/1	set_Bfeild_current		NX_FLOAT		set value on PS
0/1		units	NX_CHAR		
0/1	read_Bfield_current		NXlog		read from PS
0/1		units	NX_CHAR		
0/1	read_Bfield_voltage		NXlog		read from PS
0/1		units	NX_CHAR		
0/1	set_Efield_voltage		NX_FLOAT		set value on PS
0/1		units	NX_CHAR		
0/1	read_Efield_current		NXlog		read from PS
0/1		units	NX_CHAR		
0/1	read_Efield_voltage		NXlog		read from PS
0/1		units	NX_CHAR		

NXspin_rotator

RE	Name	Attribute	Type	Value	Description
	NXspin_rotator				component name
1	**description**		NX_CHAR		name
0/1	source_distance		NX_FLOAT		distance of flight path from production target
0/1		units	NX_CHAR	‘metres’	
0/1	set_Bfield_current		NX_FLOAT		set value on PS
0/1		units	NX_CHAR		
0/1	read_Bfield_current		NXlog		read from PS
0/1		units	NX_CHAR		
0/1	read_Bfield_voltage		NXlog		read from PS
0/1		units	NX_CHAR		
0/1	set_Efield_voltage		NX_FLOAT		set value on PS
0/1		units	NX_CHAR		
0/1	read_Efield_current		NXlog		read from PS
0/1		units	NX_CHAR		
0/1	read_Efield_voltage		NXlog		read from PS
0/1		units	NX_CHAR		

NXmagnetic_kicker

RE	Name	Attribute	Type	Value	Description
	NXmagnetic_kicker				component name
1	**description**		NX_CHAR		name
0/1	source_distance		NX_FLOAT		distance of flight path from production target
0/1		units	NX_CHAR	'metres'	
0/1	timing		NX_FLOAT		kicker timing, as defined by 'description' attribute
0/1		units	NX_CHAR	'nano.second'	
0/1		description	NX_CHAR		
0/1	set_voltage		NX_FLOAT		set value on PS
0/1		units	NX_CHAR		
0/1	read_voltage		NXlog		read from PS
0/1		units	NX_CHAR		
0/1	set_current		NX_FLOAT		set value on PS
0/1		units	NX_CHAR		
0/1	read_currente		NXlog		read from PS
0/1		units	NX_CHAR		

NXelectrostatic_kicker

RE	Name	Attribute	Type	Value	Description
	NXelectrostatic_kicker				component name
1	**description**		NX_CHAR		name
0/1	source_distance		NX_FLOAT		distance of flight path from production target
0/1		units	NX_CHAR	'metres'	
0/1	timing		NX_FLOAT		kicker timing, as defined by 'description' attribute
0/1		units	NX_CHAR	'nano.second'	
0/1		description	NX_CHAR		
0/1	set_voltage		NX_FLOAT		set value on PS
0/1		units	NX_CHAR		
0/1	read_voltage		NXlog		read from PS
0/1		units	NX_CHAR		
0/1	set_current		NX_FLOAT		set value on PS
0/1		units	NX_CHAR		
0/1	read_currente		NXlog		read from PS
0/1		units	NX_CHAR		

NXbeamline_collimator

RE	Name	Attribute	Type	Value	Description
	NXbeamline_collimator				
1	**description**		NX_CHAR		name
0/1	source_distance		NX_FLOAT		distance of flight path from production target
0/1		units	NX_CHAR	‘metres’	
0/1	shape		NXgeometry		shape of aperture
0/1	material		NX_CHAR		material from which aperture is fabricated

NXdae

RE	Name	Attribute	Type	Value	Description
	NXdae				
1	**type**		NX_CHAR	‘ISIS_MCS’ | ‘ISIS_DAE2’ | ‘RIKEN-RAL_MACS-EXP’	description of DAE hardware: form name as ‘facility’_’dae’
0/1	interface		NX_CHAR	‘CAMAC’ | ‘VME’	
Class to contain entries describing the data acquisition electronics (DAE). ‘type’ is required, other entries will be facility specific.					

NXdetector

RE	Name	Attribute	Type	Value	Description
	NXdetector			detector name	
1	**description**		NX_CHAR		description
0/1	orientation		NX_CHAR	'Longitudinal' | 'Transverse' | 'L' | 'T'	general detector arrangement -:legacy entry for MuSR spectrometer at ISIS
1	**counts**		NX_INT[ns] | NX_INT[ns][ntc] | NX_INT[np][ns][ntc]		1, 2 or 3D array of counts, [*period*, spectrum, time bin]
1		**signal**	NX_INT	'1'	
1		**axes**	NX_CHAR	'[period_index, spectrum_index, raw_time]'	axes definitions
1		**long_name**	NX_CHAR	'positron counts' | 'electron counts'	
0/1	raw_time		NX_FLOAT[ntc+1]		linked to 'raw_time' in NXdetector
0/1		units	NX_CHAR	'micro.second'	
0/1		long_name	NX_CHAR	'time'	
0/1	resolution		NX_FLOAT		histogram resolution in this NXdata group, time differential data must have one of 'resolution' or 'raw_time'
0/1		units	NX_CHAR	'nano.second'	
1	**spectrum_index**		NX_INT[ns]		list of global spectra
0/1		long_name	NX_CHAR	'spectrum number'	
0/1	spectrum_labels		NX_CHAR		list of labels for spectra, separated by character given as attribute. May use a 2D array of NX_CHAR - TBC
0/1		separator	NX_CHAR		separator character for label list
0/1	period_index		NX_INT[np]		list of period numbers
0/1		long_name	NX_CHAR	'period index'	
0/1	period_labels		NX_CHAR		linked to 'period_lables' in NXperiod
0/1	detector_index		NX_INT[ns]		'dectector_index[j]' is the location of first detector in 'spectrum_index[j]' in the array 'detector_list' *(see Note 3)*
0/1	detector_count		NX_INT[ns]		'detector_count[j]' is the total number of detectors forming spectrum in 'spectrum_index[j]' *(see Note 3)*

0/1	detector_list		NX_INT[nd]		sorted list of detector numbers *(see Note 3)*
0/1	crate		NX_INT[nd]		crate number (DAE2 ISIS parameter)
0/1	slot		NX_INT[nd]		slot number (DAE2 ISIS parameter)
0/1	input		NX_INT[nd]		input number (DAE2 ISIS parameter)
0/1	type		NX_CHAR[nd]	'scintillator'	type of detector
0/1	voltage		NX_FLOAT[nd]		PMT voltage
0/1		units	NX_CHAR	'volt'	
0/1	discriminator		NX_CHAR[nd]	'LE' | 'CF'	type of discriminator
0/1		name	NX_CHAR[nd]		Name of discriminator
0/1	threshold		NX_FLOAT32[nd]		thresholds
0/1		units	NX_CHAR	'milli.volt'	
0/1	output_width		NX_FLOAT[nd]		pulse widths
0/1		units	NX_CHAR	'nano.second'	
0/1	geometry		NXgeometry		defining detector positions *(see Note 4)*
0/1	solid_angle		NX_FLOAT[nd]		solid angle subtended by the detector at the sample
0/1		units	NX_CHAR	'steradian'	
0/1	calibrated_angles		NX_FLOAT[nd]		calibrated detector angles
0/1		units	NX_CHAR	'degree'	
0/1		calibration_date	ISO8601		date when angles were determined
0/1		calibration_run	NX_CHAR		run used to determine angles
0/1	dead_time | dead_time_coefficients		NX_FLOAT[np][nd] | NX_FLOAT[np][nd][nc]		array of detector deadtime values *(see Note 5)*
0/1		units	NX_CHAR	'micro.second'	
0/1		calibration_date	ISO8601		date when deadtime file was determined
0/1		calibration_run	NX_CHAR		run used to generate deadtime data
0/1+	grouping_*		NX_INT[np][ns]		preferred spectrum grouping - 1D array *(see Note 6)*
0/1		primary	NX_INT		define preferred grouping, if multiple entries
0/1+		number_groups	NX_INT	'ng'	number of groups
0/1+		name_groups	NX_CHAR		semicolon separated list of group names May use a 2D array of NX_CHAR - TBC
0/1+	alpha_*		NX_FLOAT[ng/2][3]		alpha, for pairs of groups defined in 'grouping' - 2D array *(see Note 7)*
0/1+		number_alpha	NX_INT		number of alpha

					values defined
0/1	time_zero		NX_FLOAT | NX_FLOAT[np][ns]		zero time for histograms - reference ‘raw_time’, may be global or individual to detectors
0/1		units	NX_CHAR	‘micro.second’	
0/1	time_zero_bin		NX_INT | NX_INT[np][ns]		bin equating to zero time, may be global or individual to detectors
0/1	first_good_time		NX_FLOAT | NX_FLOAT[np][ns]		time for first good data - reference ‘raw_time’, may be global or individual to detectors
0/1		units	NX_CHAR	‘micro.second’	
0/1	first_good_bin		NX_INT | NX_INT[np][ns]		bin containing first good data, may be global or individual to detectors
0/1	last_good_time		NX_FLOAT | NX_FLOAT[np][ns]		time for last good data - reference ‘raw_time’, may be global or individual to detectors
0/1		units	NX_CHAR	‘micro.second’	
0/1	last_good_bin		NX_INT | NX_INT[np][ns]		bin containing last good data, may be global or individual to detectors

NXlog

RE	Name	Attribute	Type	Value	Description
	NXlog				name of log
0/1	description		NX_CHAR		description of log
0/1	displayname		NX_CHAR		name displayed on DAE software, e.g. SECI block name
0/1	hardware		NX_CHAR		Rig, e.g. 'CCR'
0/1	software		NX_CHAR		version of driver used to collect data - typically VI name and version
0/1	raw_value		NX_FLOAT | NX_INT		log of raw values, e.g. voltage, etc
0/1		units			
1	**value**		NX_FLOAT[] | NX_INT[] | NX_FLOAT[][] | NX_INT[][]		log of values or arrays, e.g. 'temperature values obtained from the 'TLOG' file
1		**units**	NX_CHAR		
0/1	time		NX_FLOAT[]		time stamp of values as offsets from start time
0/1		start	ISO 8601		time of run start
0/1		units	NX_CHAR		units of logged values
0/1	average_value		NX_FLOAT		average of logged values
0/1		units	NX_CHAR		
0/1	average_value_error		NX_FLOAT		standard deviation of logged values
0/1		units	NX_CHAR		
0/1	minimum_value		NX_FLOAT		minimum of logged values
0/1		units	NX_CHAR		
0/1	maximum_value		NX_FLOAT		maximum of logged values
0/1		units	NX_CHAR		

NXgeometry

RE	Name	Attribute	Type	Value	Description
	NXgeometry				name of component
0/1	{*shape*}		NXshape		shape/size information of component
0/1	{*translation*}		NXtranslation		translation of component
0/1	{*orientation*}		NXorientation		orientation of componenet
0/1	description		NX_CHAR		description
0/1	component_index		NX_INT		position of the component along the beam path - sample is at 0, components upstream have negative index, components downstream have positive index

NXshape

RE	Name	Attribute	Type	Value	Description
	NXshape				
0/1	shape		NX_CHAR		defining shape
0/1	size		NX_FLOAT[numobj,nshapepar]		extent of object along local axes – defined by 'nshapepar' parameters
0/1		units	NX_CHAR		

NXtranslation

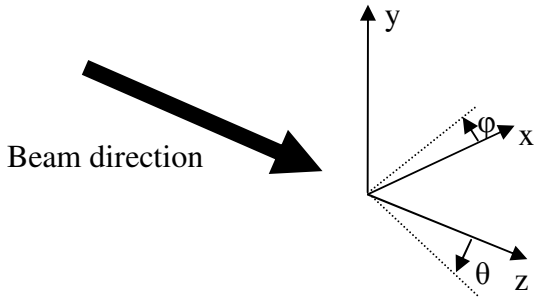
RE	Name	Attribute	Type	Value	Description
	NXtranslation				positions
0/1	distances		NX_FLOAT[numobj,3]		positions of components
0/1		units	NX_CHAR	‘m’	

NXorientation

RE	Name	Attribute	Type	Value	Description
	NXorientation				orientations
0/1	value		NX_FLOAT[numobj,6]		orientation information stored as directional cosines

Notes:

Coordinate systems: The coordinate system has been revised since Version 1. Coordinate axes are taken with the z-axis pointing along the incident beam direction, the y-axis defined as upwards perpendicular to the beam in the vertical plane and the x-axis perpendicular to the beam in the horizontal plane pointing left as seen from the source. The origin is taken to be at the centre of the spectrometer, usually about the sample position. Cartesian, cylindrical and spherical polar coordinate systems may be used with angles ϕ (rotation about the z-axis) and θ (rotation about the x-axis), increasing for clockwise rotations viewed from the origin of the coordinate system.



Conventions used in the document: The conventions used in this document are as defined in the paper 'The ISIS NeXus RAW Data File Format' by F.A. Akeroyd. In brief, to help understand this document:

- Variables in bold are mandatory, names in italics are examples and any suitable name may be substituted.
- The column headed 'RE' indicates how often an item name may occur: 0/1 – optional with a maximum of one instance, 1 – a single instance is mandatory, 0/1+ - not mandatory, but may occur more than once.
- Special variables: np – number of periods, ntc – number of time channels in spectra, ns – number of spectra (histograms), nd – number of detectors and ng – number of groups.
- For strings where the variable should only take certain values, this is indicated by 'string 1' | 'string 2'.

Note 0: The groups 'NXrunlog', 'NXselog', 'NXperiod' and NXdae are defined to store data acquisition parameters and action logs. The intention is to seek ratification of these container classes on the understanding that much of the contents are likely to be facility specific.

Note 1: In line with the latest advice from the NIAC, raw data is now moved to the NXdetector group, i.e. this contains both the parameters describing the detector and the resulting data. The NXdata group contains links to plottable data. Division of data into multiple NXdetector groups may be made according to convenience; strategies might include having an NXdetector group for each physical detector e.g. 'UP', 'DOWN', etc, or one for each bank of detectors, e.g. the 'Forward/Backward' array –

the latter strategy is adopted at ISIS. It is suggested that both the NXdetector and NXdata groups are named according to the convention 'detector_*{function}*'. The situation where a single physical detector is read using multiple data acquisition electronics should be handled by writing multiple NXdetector groups, where an underscore and instance number is appended to the functional name. For example, for a transverse detector array used to collect data using DAE2 and the DASH2 systems, there would be two NXdetector groups (called 'detector_transverse_1' and detector_transverse_2) and two NXdata groups (also called 'detector_transverse_1' and 'detector_transverse_2').

Note 2: Similar naming schemes exist for defining sample environment (temperature_*) and logged (temperature_*_log) variables, immediately showing the dependence to the variable. Sample environment variables should begin with a prefix that uniquely defines the class of sample environment they describe.

Note 3: Data is collected as a series of spectra (histograms). Usually there is a one to one mapping between the spectra and detectors, but occasionally signals from a number of detectors can be combined to form a single spectrum (muon DAE2 may have this facility in the future), and therefore the number of spectra (ns) can be less than the number of detectors (nd). The proposed detector <-> spectrum indexing scheme is suggested, that follows that proposed for neutron data.

> The detectors {i} are numbered i=[1,nd], and their output is mapped onto the spectra {j}. The global spectrum number must be unique, and this number is provided by spectrum_index[j]. The array detector_list contains a list of detector numbers {i}, but they are arranged such that detectors which map to the same spectrum appear sequentially with the start of this sequence given by detector_index[j]. The spectrum spectrum_index[j] will thus have detector_count[j] detectors mapped into it, the actually detector numbers being given by detector_list[k ... k+detector_count[j]-1], where k=detector_index[j].

For a one-to-one mapping between spectra and detectors (usual for μSR experiments), the entries 'detector_index', 'detector_count' and 'detector_list' need not be written.

Note 4: 'geometry' (given in group NXgeometry), 'shape', 'size' and 'translation' uniquely define the instrument detector array. These values will usually be 'engineering' values, with the calibrated azimuthal angles given in 'calibrated_angles'.

Note 5: Detector deadtime may be handled in two ways, either by including an array of deadtime values, one for each spectrum, or by an array of coefficients determined from a polynomial fit to the early time distortion; entries are named accordingly.

Note 6: The grouping is represented as a 1D array with the index representing the spectrum number and the value corresponding to the group into which it should be placed. The number and names of groups is defined as an attribute. Multiple groupings for a given detector bank may be written, with the preferred grouping indicated by the 'primary' attribute. The grouping entry should only be written if the values can be defined. Possible sources of the grouping information may be a default file or grouping tables derived from analysis programme, e.g. 'long.uda' for longitudinal data and 'trans.uda' for transverse data from UDA.

Note 7: The balance parameter, alpha, is given for pairs of groups defined in 'grouping'. Association with a particular grouping entry is by name. Values are stored in a 2D array together with the numbers for the forward and backward groups.

General comments:

- The NeXus Instrument Definition has entries that define how a wide range of information should be stored in the file. Obviously most entries will be of no use to most people, and therefore their inclusion is optional. Various strategies might be required to automatically capture a comprehensive array of meta-data.

- A number of additional quantities were logged by the MuSR instrument and written to the top level of the Version 1 NeXus file. These entries were not defined by the Instrument Definition; Appendix 2 discusses how these have been incorporated into the Version 2 definition.

- Special experiments (such as RF, illumination, E-field) will require additional entries to fully describe experiment parameters. It is recommended that additional entries are placed into the NeXus file structure as follows:

    a) An addendum to the IDF should be defined to show how new entries are written within the muon IDF structure, and clarify their exact meaning. The spirit of the existing IDF should be followed wherever possible when defining new entries.

    b) In general, all new entries taken from SECI blocks will be written to the defined NXselog group (see discussion in Appendix 2).

    c) New entries should be named using a unique prefix. Where possible conventions suggested by the NIAC should be followed, otherwise e.g. ‘rf_’ for RF experiments, ‘laser_’ for laser experiments, ‘current_’ for currents. A numbered suffix could be used to allow multiple numbers of a specific device. For example, a simple IDF extension for an RF experiment might include ‘rf_frequency’, ‘rf_power’, ‘rf_delay’, ‘rf_pulselength’ and ‘rf_field’, and the definition might be made as follows:

RE	Name	Attribute	Type	Value	Description
	rf_frequency		NX_FLOAT32	rf frquency	
		units	NX_CHAR	‘MHz’	
		hardware	NX_CHAR	‘marconi’	
		display_name	NX_CHAR	name displayed on DAE software	at ISIS, typically SECI block name
		software	NX_CHAR	version of driver used to collect data	at ISIS, typically VI name and version
	rf_delay		NX_FLOAT32	rf delay	relative to extract
		units	NX_CHAR	‘us’	
		hardware	NX_CHAR	‘stanford’	
		display_name	NX_CHAR	name displayed on DAE software	at ISIS, typically SECI block name
		software	NX_CHAR	version of driver used to collect data	at ISIS, typically VI name and version
	rf_pulselength		NX_FLOAT32	rf pulse length	
		units	NX_CHAR	‘us’	
		hardware	NX_CHAR	‘CAEN Dual Timer’	
		display_name	NX_CHAR	name displayed on DAE software	at ISIS, typically SECI block name
		software	NX_CHAR	version of driver used to collect data	at ISIS, typically VI name and version
	rf_power		NX_FLOAT32	desired RF power	
		units	NX_CHAR	‘W’	
		hardware	NX_CHAR	‘AMT’	
		display_name	NX_CHAR	name displayed on DAE software	at ISIS, typically SECI block name
		software	NX_CHAR	version of driver used to collect data	at ISIS, typically VI name and version
	rf_power_log		NXlog	power log	
	rf_field		NX_FLOAT32	rf field strength	
		units	NX_CHAR	‘G’	
		rf_field_vector	NX_FLOAT[3]	vector describing orientation of RF field	
		display_name	NX_CHAR	name displayed on DAE software	at ISIS, typically SECI block name
		software	NX_CHAR	version of driver used to collect data	at ISIS, typically VI name and version

Appendix 1: Class definitions unique to 'muonTD'

The following class names are used in the definition of 'muonTD' that have not previously been defined and ratified by the NIAC. The intention is to seek ratification in order that classes and certain entry names can be reserved for muon applications.

- *NXrunlog:*
  A container for time logged parameters relating to the run.
- *NXselog:*
  A container for time logger parameters relating to the sample environment.
- *NXperiod:*
  A container for parameters relating to operation of period data acquisition.
- *NXdae:*
  A container for information about the data acquisition electronics.
- *NXbeamline:*
  Information on the beamline between target and instrument, and therefore an entry of *NXinstrument*.
- *NXdiagnostics:*
  A container for diagnostic information on beamline function (for inclusion in *NXbeamline*).
- *NXdipole_magnet:*
  (aka dipole) defined by the NIAC as *NXbending_magnet*, but change of name is requested and definition requires extension (for inclusion in *NXbeamline*).
- *NXquadrupole_magnet:*
  Class definition for a quadrupole magnet (for inclusion in *NXbeamline*).
- *NXsolenoid_magnet:*
  Class definition for a solenoid magnet (for inclusion in *NXbeamline*).
- *NXmagnetic_kicker:*
  Class definition for a magnetic beam kicker (for inclusion in *NXbeamline*).
- *NXelectrostatic_kicker:*
  Class definition for a magnetic beam kicker (for inclusion in *NXbeamline*).
- *NXseparator:*
  Class definition for a particle separator (for inclusion in *NXbeamline*).
- *NXspin_rotator:*
  Class definition for a spin rotator (for inclusion in *NXbeamline*).
- *NXseptum_magnet:*
  Class definition for a septum magnet (for inclusion in *NXbeamline*).
- *NXbeamline_collimator:*
  Class definition for a beamline collimator (for inclusion in *NXbeamline*).
- *NXuif*
  Container class for additional data/information defined by the user.

In addition, the intention is to ratify the name '*idf_version*' (for inclusion in NXentry) as a means of the muon community identifying the version of the Instrument Definition File (IDF) that the data file conforms to. It should be noted, that the recent muon data files are written containing both 'idf_version' and 'IDF_version', and therefore interrogating either case will yield a definite result.

The muon community would also value an extension to the definition of NXlog to enable information stored as a one dimensional array to be logged as a function of time. This facility is important for allowing vector fields to be stored (see entry for the magnetic field log in NXsample) and enabling, for example, traces recorded from oscilloscopes to be preserved.

Appendix 2: Incorporating SECI logged values

The ISIS Instrument Control Program (ICP), and particularly the Sample Environment Control Interface (SECI), enables the user to define logged parameters through the use of 'blocks'. Since SECI allows the user to load additional LabView VIs and associate 'blocks' with LabView values, the number of values logged by SECI may be almost unlimited. Clearly it is desirable for any value logged by SECI to be available in the data file, and indeed it is the current practice at ISIS to write all SECI defined 'blocks' when the run is completed.

Given the unlimited nature of the information that may be recorded by SECI, it is quite possible that 'blocks' may be defined that contain information that is beyond the scope of the Instrument Definition. Therefore, the following strategy has been adopted at ISIS for writing this information: a) all logged values associated with all SECI 'blocks' are written to a container group 'NXselog'; b) logged values defined throughout the Instrument Definition are actually formed by links pointing to the relevant information within the NXselog group.

It should be noted that if parameters are regularly stored outside the scope of the Instrument Definition, consideration should be given to defining extensions to the Definition (e.g. an extended description for inclusion within *NXsample*).