BNL-102260-2014-TECH

RHIC/AP/155;BNL-102260-2013-IR

SXF (Standard exchange Format) Definitions, syntax, examples

H. Grote

June 1998

Collider Accelerator Department

Brookhaven National Laboratory

U.S. Department of Energy

USDOE Office of Science (SC)

DISCLAIMER

RHIC/AP/155

SXF (Standard eXchange Format): definition, syntax, examples

H.Grote, J.Holt, N.Malitsky, F.Pilat, R.Talman, C.G.Trahern

An accelerator lattice as a single sequence

An SXF lattice description is an ascii listing that contains one named, “flat”, ordered list of elements, delimited as {...}, with one entry for each element. The list resembles a MAD “sequence” describing the entire machine. The syntax is supposed to be adapted for ease of reading by human beings and for ease of parsing by LEX and YACC.

Grammatical rules

Reserved words *rw* are lower case. The allowed data types are **names** (which are case sensitive, with alphanumerics, upper and lower case, periods, dashes, colons and underscores all allowed), scalar **numbers** v, (integer or decimal, E or e, D or d, exponential notation, or decimal point alone, all allowed), **arrays** a (which are dynamically-sized ordered collections of elements and are only used as 1-dimensional vectors here) delimited as [v1 v2], or **associative arrays** (which are unordered sequences of key-value pairs and are also known as hashes or dictionaries) delimited as {*rw1*=v1 *rw2*=v2 *rw3*=v3}. The separation characters are blanks or end-of-line characters. Everything is case-sensitive. C++ style comments are allowed; also, lines beginning with # are regarded as comments. Blank lines are ignored. Unrecognized words in locations where a reserved word is expected leads to fatal error.

Lattice description rules:

- Element types (like *quadrupole*) and their attributes correspond to MAD-8 types unless they have been redefined. A complete table is given below.
- The TILT and ANGLE attributes are **not allowed**.
- The syntax for individual attributes is like “*l*=3.5” but parameters that are naturally grouped (for example multipole coefficients) are given in the form of a fixed-order, variable-length arrays like “*kl*=[0.3 0.7]” having an explicit entry for each element up to and including the last non-vanishing element; all subsequent elements are zero by default. This is just an application of the principle (for which there a few natural exceptions) that attribute values not given **default to zero.**
- Length-integrated (length*strength) products (like *kl*) are given rather than strengths (like *k*) alone. Field integrals are defined as in MAD, i.e. *kl*(SXF) = *kl*(MAD).
- Element longitudinal structure is described in the form of **primary** attribute values within *entry*{ }, *body*{ }, and *exit*{ }, all optional, and deviations from those values within *entry.dev*{ }, *body.dev*{ }, and *exit.dev*{ }. This structure is provided for convenience in changing a few values while holding the rest fixed and (with important exceptions to be discussed below) it is only the sums of **primary** values and **deviations** that constitute the “flat” lattice description.
- For backward compatibility with MAD, pole-face angles *e1, e2* and similar attributes can be

included as body attributes (see below) but this is deprecated since the data are more consistently treated in *entry* and *exit*.

- **Intentional** displacements from the **primary** closed orbit are listed within *align*. **Deviations** from these values are in *align.dev*. Correct use of these features requires considerable care which amounts to understanding and respecting the requirements discussed below in the section "Closed orbit and linear optics".
- Every element *must* have a **unique element name** (which is probably a laboratory-wide name such as the RHIC SiteWideName) and an **element type** specified. All other entries are optional with defined defaults (almost always zero).
- The *tag* attribute is an optional name satisfying the same rules as the unique element name. Its purpose is provide the option of retaining the name of a "parent" from which the element was previously derived. There is **no inheritance** of attribute values implied by this however.

Example

An SXF sequence with three elements (a quadrupole, a dipole and a marker) look like:

```
// SXF version 1.0
ThreeElementExample sequence {
qd13.r2 quadrupole { tag=qd at=100.0 l=3.5
        body =  { kl=[0, 0.005] }
        align = { al=[0.001 0.001 0 0 0.005] }
        body.dev =  { kl=[0.0001 ...] kls=[0.0001 ...] }
        entry.dev = { kl=[0.0001 ...] kls=[0.0001 ...] }
        exit.dev =  { kl=[0.0001 ...] kls=[0.0001 ...] }
        align.dev = { al=[0.0005 0.0005 0 0 0.001] }
        aperture =  { ap=[1 0.03 0.02 0.001 0.001] }
};
bh.r1   sbend { tag=bh at=200.0 l=10.0
        body = { kl=[0.0045 0.001 0.0002 0.045] fint=0.01
                 e1=0.001 e2=0.002 h1=0.1 h2=0.02 hgap=0.002 }
        align = { al=[0.001 0.001 0 0 0.0005] }
        body.dev =  { kl=[0.0001 ...] kls=[0.0001, ...] }
        entry.dev = { kl=[0.0001 ...] kls=[0.0001 ...] }
        exit.dev =  { kl=[0.0001 ...] kls=[0.0001 ...] }
        align.dev = { al=[0.0005 0.0005 0 0 0.001] }
        aperture =  { ap=[1 0.03 0.02 0.001 0.001] }
};
m1 marker {
};
endsequence at=300.0
}
// SXF end
```

Explanation

- *ThreeElementExample sequence {* : names the sequence and opens its definition

- *endsequence at=300.0* : closes the sequence definition. The *at=* value is **required** (since it fixes the length of a possible final drift section).
- *qd13.r2, bh.r1* : the **unique element name (repetition not allowed)**
- *qd, bh* : generic element name (zero or more repetitions allowed). Here these names are treated only as optional *tag*'s with **no inheritance implied.**
- *quadrupole, sbend* :the **element type,** as listed below (**abbreviation not allowed**).
- *at :* defines the longitudinal position of the element center relative to the start of the sequence (which is assumed to be *at =0*). If *at* is missing the element is assumed to be butted up against the previous element.
- *l* : the magnetic length of the element along the design orbit. this definition is qualified in the section below entitled "Closed orbit and linear optics".
- *flipped* : since the multipole coefficients in SXF are to be expressed in the frame of reference in which the magnet has been measured, a magnet can have the attribute *flipped* = 0, 1, 2 or 3. 0 means the lattice frame and measurement frame are identical and 1, 2, 3 mean, respectively, that the frames are flipped by 180 degrees around the x, y or z axis.
- *body* : contains **primary** design attributes (such as the length*strength product of a quadrupole) of the element not including end effects. *kl* = [ ] is a vector up to dimension 20 that holds normal integrated strengths. *kls* = [ ] is a vector up to dimension 20 that holds skew integrated strengths. the first entry is "dipole order", the second "quadrupole order" and so on up to the last non-vanishing order.
- align : intentional spatial displacement of the element in the local reference system, The entries of al[0,1,2,3,4,5] are (Δx, Δy, Δs, $\Delta\alpha_x$, $\Delta\alpha_y$,$\Delta\alpha_s$) which are (horizontal displacement, vertical displacement, longitudinal displacement, small rotation about the local x-axis, small rotation about the local y-axis, small rotation about the local s-axis)
- *aperture* : *shape* = 1,2,3 for ellipse, rectangle, diamond respectively. Width and height are contained in *size*[ ], horizontal and vertical displacements from local design orbit are contained in *al*[ ].

Closed orbit and linear optics

Since the leading purpose of SXF is for transferring lattice descriptions with the aim of comparing or combining the results of different codes it is extremely important that the **primary** closed orbit be specified unambiguously to permit high precision preliminary checks. One way to reduce ambiguity is to reduce redundancy and that is why the ANGLE attribute has been **disallowed** or, if one prefers, has been redefined as the *kl*[0] value contained within the *body*{ } attribute list. Often this *body*{*kl*[0]} entry will have come directly from the lattice "ideal design" but to avoid this supposition the term **primary** has been introduced to describe the data in lists not having the "*.dev*" suffix. Hence, in the case of *body*{*kl*[0]} it will be conventional within SXF that the intended **primary** closed orbit be obtained by suppressing any non-zero value of *body.dev*{*kl*[0]}. If this entry is in fact non-zero, after confirming the primary orbit, it will be necessary to make another closed orbit calculation to find the perturbed closed orbit of the lattice with all deviations included.

Much the same statements can be made about the linear optics. With accelerator lattice optics being so highly sensitive to small changes it is also important and difficult to guarantee agreement of linear optical functions to high precision. For that reason it will be conventional within SXF that the intended **primary** linear optics be obtained by simultaneously suppressing any non-zero values of *body.dev*{*kl*[0]} and *body.dev*{*kl*[1]} and this will facilitate precise preliminary comparison of linear optics.

Similar comparisons are possible at higher orders. For full SXF compliance all such comparisons are to be meaningful. These are to be performed using the **primary** lattice which has all "*.dev*" entries zeroed out and can be used to check agreement between codes at the sending and receiving ends. Since it may not always be convenient to respect this convention, this requirement can be dropped to yield a non-fully-compliant SXF in which it is not valid to treat separately the two terms in the sum *body*{*kl*[*i*]}+*body.dev*{*kl*[*i*]}. In its conceptually purest form it is only these sums that constitute the fully flat SXF lattice description.

Other than ANGLE the important contributor to the global machine geometry is the length of individual elements. For ease of locating elements longitudinally it is extremely convenient if all arc lengths within elements can simply be added and SXF **requires** this feature.

SXF is designed to be free of redundancy if deprecated features (present for temporary backward compatibility) are avoided. Unfortunately there is one respect in which this is hard to guarantee. This has to do with arc length along the **primary** closed orbit and primarily with bending magnets. For *sbend*'s (that is "sector" bends) there is no problem since in all accelerator codes the length parameter *L* or *l* stands for arc length along the nominal (circular) orbit. Furthermore this is a "magnetic length", which is to say that multiplying it by the nominal dipole field yields the correct length-integrated length-strength product. But for *rbend*'s (that is "rectangular" bends) some codes assume that *L* or *l* stands for the end-to-end length along a straight line---this is normally a magnetic length so at least that is not an issue.

Since, except for end effects differing only by their pole face rotation parameters, the trajectories through *rbend*'s and *sbend*'s are identical, it has always been redundant to have two separate types. This is even more true when, as in SXF, the end effect have been broken out from the body effects. Furthermore it is not simple to split *rbend*'s longitudinally to obtain finer granularity. It is for these reasons the use of *rbend*'s is deprecated.

But in case *rbend*'s continue to be used (as we fear may be the case) we have introduced a new attribute *arc* which is arc length along the circular orbit. For *sbend*'s *arc* and *l* are equal so *arc* need never appear. If it does it will be ignored. For *rbend*'s *arc* and *l* are redundant since they satisfy

$$arc = l \times \frac{kl[0]/2}{\sin(kl[0]/2)}$$

Here *kl*[0] is the entry appearing within *body*{ }. If *l* is given it will be converted to *arc* within SXF using this formula. If both *arc* and *l* are given *l* is ignored. Recapitulating, if (in spite of its being deprecated) an *rbend* is used and *l* appears (in spite of *arc* being its preferred length designator) SXF (like MAD) will interpret *l* as a straight line magnetic length.

None of this has in any way broached the subject of the appropriate multipole expansion to be applied to bending magnets. The standard multipole expansion is strictly valid only for bend-free elements. Normal practice is to ignore this and apply the standard expansion to bend elements and SXF continues this tradition. At some level of accuracy this is not correct, but at least this issue makes no distinction between *rbend*'s and *sbend*'s even if the multipoles have been measured using a full length, necessarily straight rotating coil. If the multipoles have been measured with a "mole" that follows an *sbend* it seems likely (though not guaranteed) that this error will be reduced, which is yet another reason for avoiding *rbend*'s in superconducting accelerators.)

When, at some later date, this issue is better understood or an elegant procedure has been found for transforming to an expansion tailored to circular orbits, SXF will have to be modified appropriately.

element type	*body* attributes	remarks
marker		
drift		*at* not allowed
rbend	*kl*[ ], *kls*[ ], *fint*, *hgap* *e1*, *e2*	*kl*[0]=ANGLE, type is deprecated deprecated
sbend	*kl*[ ], *kls*[ ], *fint*, *hgap* *e1*, *e2*	*kl*[0]=ANGLE deprecated
quadrupole *sextupole* *octupole*	*kl*[ ], *kls*[ ] *kl*[ ], *kls*[ ] *kl*[ ], *kls*[ ]	
multipole	*kl*[ ], *kls*[ ]	
solenoid	*ks*	
hkicker *vkicker* *kicker*	*kl*[ ], *kls*[ ] *kl*[ ], *kls*[ ] *kl*[ ], *kls*[ ]	
rfcavity	*volt*, *lag*, *harmon*	unit of *volt* is MV/m
elseparator	*el* *e*	unit of *e* is MV/m deprecated
hmonitor *vmonitor* *monitor*		
instrument		
ecollimator *rcollimator*	*xsize* = [ ], *ysize*[ ] *xsize* = [ ], *ysize*[ ]	
beambeam	*size* = [ ], *npart*, *charge*	*align* (including α_s, not necessarily small) is offset of "other" beam. *size* is r.m.s.

Table 1: Allowed element types and their primary attributes. All element types have attribute *l* that defaults to zero and all but *drift* have *at* which defaults to "butted elements". All element have *align* attributes expressed as *align* = {*al* =[ ]}. All element have *aperture* attributes that default to "ignore aperture" and are subdivided as *aperture*{*shape* = integer *size* = [ ] *al* = [ ]}. All units are MKS except *volt.*

Comments

The format allows a consistent expandability; things like parameter maxima and minima can be added the way apertures have been here.

Even though *drift*'s are implicitly included by means of the *at* = mechanism they are allowed to be included explicitly as well.

The European/American multipole indexing convention disagreement is finessed by using the array syntax.

The *tag* = mechanism permits the inclusion of a generic name; this has been retained as an option on the principle that potentially useful information should not be thoughtlessly discarded.

The presence of a semi-colon terminating each element is expected to be helpful for repairing files that have been slightly corrupted while being hand-edited.

Angular misalignments are given in form judged most convenient for comparison with actual survey measurement. They must be kept small (like the angles they replace in existing lattice descriptions) since they do not commute. This smallness requirement can be dropped, as for example in *beambeam*, if only one angle is allowed.

All redundant constructs are deprecated. For example, *hkick* and *vkick* elements can both be represented by kick elements and are hence deprecated.

For ease of reading, the listing should be "pretty-printed" with simple assignments listed first, followed by array assignments in the order body{ }, entry{ }, exit{ }, body.dev{ }, entry.dev{ }, align{ }, align.dev{ }, aperture{ }. This not required however. Standard indentation should also be applied, for example as in the example.