

3.1. GENERAL CONSIDERATIONS WHEN DEFINING A CIF DATA ITEM

associated with the group of three values in some well defined format (e.g. comma separated, as h, k, l) could have been defined instead. However, this would require a parser to understand the internal structure of the value so that it could parse out the separate values for h, k and l .

On the other hand, there are many examples of data values that are stored as string values parsable into distinct components. An extreme example is the reference list mentioned in Section 3.1.7.1. More common are dates (`_audit_creation_date`), chemical formulae (e.g. `_chemical_formula_moiety`), symmetry operations (`_symmetry_equiv_pos_as_xyz`) or symmetry transformation codes (`_geom_bond_site_symmetry_1`). There is no definitive answer as to which approach is preferred in a specific case. In general, the separation of the components of a compound value is preferred when a known application will make use of the separate components individually. For instance, applications may list structure factors according to a number of ordering conventions on individual Miller indices. As an extreme example of separating the components of a compound value, the mmCIF dictionary defines data names for the standard uncertainty values of most of the measurable quantities it describes, while the core dictionary just uses the convention that a standard uncertainty is specified by appending an integer in parentheses to a numeric value.

When compound values are left as parsable strings, the parsing rules for individual data items need to be made known to applications. The DDL1 attribute `_type_construct` was envisaged as a mechanism for representing the components of a data value with a combination of regular expressions and reference to primitive data items, but this has not been implemented in existing CIF dictionaries (or in dictionary utility software). An alternative approach used in DDL2-based dictionaries defines within the dictionaries a number of extended data types (expressed in regular-expression notation through the attribute `_item_type_list.code`).

A related problem is how to handle data names that describe an indeterminate number of parameters. For example, in the modulated structures dictionary an extra eight Miller indices are defined to span a reciprocal space of dimension up to 11. In principle, the dimensionality could be extended without limit. According to the practice of defining a unique data name for each modulation dimension, new data names would need to be defined as required to describe higher-dimensional systems. Beyond a certain point this will become unwieldy, as will the set of data names required to describe the n^2 components of the W matrix for a modulated structure of dimensionality n (`_cell_subsystem_matrix_W_1_1` etc.).

The modulated structures dictionary was constrained to define extended Miller indices in this way for compatibility with the core dictionary. Data names describing new quantities that are subject to similar unbounded extensibility should perhaps refer to values that are parsable into vector or matrix components of arbitrary dimension.

3.1.7.5. Consistency of abbreviations

One further consideration when constructing a data name is the use of consistent abbreviations within the components of the data name. This is of course a matter of style, since if a data name is fully defined in a dictionary with a machine-readable attribute set, the data name itself can be anything. Nonetheless, to help to find and group similar data names it is best to avoid too many different abbreviations.

Table 3.1.7.1 lists the abbreviations used in the current public dictionaries. Note that there are already cases where different abbreviations are used for the same term.

3.1.8. Management of multiple dictionaries

So far this chapter has discussed the mechanics of writing dictionary definitions and of assembling a collection of definitions in a single global or local dictionary file. In practice, the set of data names in a CIF data file may include names defined in several dictionary files. A mechanism is required to identify and locate the dictionaries relevant to an individual data file. In addition, because dictionaries are suitable for automated validation of the contents of a data file, it is convenient to be able to overlay the attributes listed in a dictionary with an alternative set that permit validation against modified local criteria. This section describes protocols for identifying, locating and overlaying dictionary files and fragments of dictionary files.

3.1.8.1. Identification of dictionaries relevant to a data file

A CIF data file should declare within each of its data blocks the names, version numbers and, where appropriate, locations of the global and local dictionaries that contain definitions of the data names used in that block. For DDL1 dictionaries, the relevant identifiers are the items `_audit_conform_dict_name`, `_audit_conform_dict_version` and `_audit_conform_dict_location`, defined in the core dictionary. DDL2 dictionaries are identified by the equivalent items `_audit_conform.dict_name`, `*.dict_version` and `*.dict_location`. For convenience, the DDL1 versions will be used in the following discussion.

The values of the items `_audit_conform_dict_name` and `_audit_conform_dict_version` are character strings that match the values of the `_dictionary_name` and `_dictionary_version` identifiers in the dictionary that defines the relevant data names. Validation against the latest version of a dictionary should always be sufficient, since every effort is made to ensure that a dictionary evolves only by extension, not by revising or removing parts of previous versions of the dictionary. Nevertheless, including `_audit_conform_dict_version` is encouraged: it can be useful to confirm which version of the dictionary the CIF was initially validated against.

The data item `_audit_conform_dict_location` may be used to specify a file name or uniform resource locator (URL). However, a file name on a single computer or network will be of use only to an application with the same view of the local file system, and so is not portable. A URL may be a better indicator of the location of a dictionary file on the Internet, but can go out of date as server names, addresses and file-system organization change over time. The preferred method for locating a dictionary file is to make use of a dynamic register, as described in Section 3.1.8.2. Nevertheless, `_audit_conform_dict_location` remains a valid data item that may be of legitimate use, particularly in managing local applications.

The following example demonstrates a statement of dictionary conformance in a data file describing a powder diffraction experiment with some additional local data items:

```
loop_
  _audit_conform_dict_name
  _audit_conform_dict_version
  _audit_conform_dict_location
  cif_core.dic      2.3.1 .
  cif_pd.dic       1.0.1 .
  cif_local_my.dic 1.0
                  /usr/local/dics/my_local_dictionary
```

3. CIF DATA DEFINITION AND CLASSIFICATION

Table 3.1.7.1. *Abbreviations in CIF data names*

Terms for which abbreviations are defined are sometimes found unabbreviated.

Abbreviation	Term	Abbreviation	Term	Abbreviation	Term
abbrev	abbreviation	eqn	equation	oper	operation
abs	absolute (configuration, not structure)	esd	standard uncertainty (estimated standard deviation) (<i>see su</i>)	org	organism
absorpt	absorption			orient	orientation
alt	alternative	expt	experiment	origx	orthogonal coordinate matrix (PDB files)
amp	amplitude	exptl	experimental	os	operating system
AN	accession number	fom	figure of merit	param	parameter
anal	analyser	fract	fractional	pd	powder diffraction
aniso	anisotropic*	Fsqd	<i>F</i> squared	PDB	Protein Data Bank
anisotrop	anisotropic*	gen	generation	PDF	Powder Diffraction File
anom	anomalous	gen	generator	perp	perpendicular
ASTM	American Society for Testing and Materials	gen	genetic	phos	phosphate
asym	asymmetric	geom	geometric	pk	peak
atten	attenuation	H-M	Hermann–Mauguin	polarisn	polarization
au	arbitrary units	ha	heavy atom	poly	polymer
auth	author	hbond	hydrogen bond	pos	position
av	average	hist	history	prep	preparation
ax	axial	horiz	horizontal	proc	processed
B	<i>B</i> form of atomic displacement parameter (a.d.p.)	I	intensity	prof	profile
		ICSD	Inorganic Crystal Structure Database	prot	protein
backgd	background*	id	identifier	ptnr	partner
beg	begin	illum	illumination	publ	publication
bg	background*	imag	imaginary	R	agreement index
biol	biology	inc	increment	rad	radius
bkg	background*	incl	include	recd	received
bond	bonding	info	information	recip	reciprocal
Bsol	<i>B</i> form of a.d.p. for solvent	instr	instrument	ref	reference
calc	calculated	Int	international	refine	refinement
calib	calibration (pd)	ISBN	International Standard Book Number	refln	reflection
cartn	Cartesian	iso	isotropic	reflns	reflections
CAS	<i>Chemical Abstracts Service</i>	iso	isomorphous	res	resolution
char	characterization (pd)	ISSN	International Standard Serial Number	restr	restraints
chem	chemical	IUCr	International Union of Crystallography	rev	revision
chir	chirality	IUPAC	International Union of Pure and Applied Chemistry	Rmerge	agreement index of merging
clust	cluster			rms	root mean square
coef	coefficient	len	length	rot	rotation
com	common	lim	limit	S	goodness of fit
comp	component	loc	lack of closure	samp	sample
conc	concentration	ls	least squares	scat	scattering factor
conf	conformation	max	maximum	seq	sequence
config	configuration	MDF	Metals Data File	sigl	$\sigma(I)^*$
conform	conformant	meanI	mean intensity	sigmaI	$\sigma(I)^*$
conn	connectivity	meas	measured	sint	$\sin \theta$
cons	constant	mid	middle (between max and min)	sint/lambda	$\sin(\theta)/\lambda^*$
CSD	Cambridge Structural Database	min	minimum	sol	solvent
db	database	mod	modification	spec	specimen
defn	definition	mods	modifications	src	source
detc	detector	mon	monomer	std	standard
der	derivative	monochr	monochromator (pd)*	stol	$\sin(\theta)/\lambda^*$
dev	standard deviation	mono	monochromator (pd)*	struct	structure
dict	dictionary	nat	natural	su	standard uncertainty
dif	difference*	NBS	National Bureau of Standards (now National Institute of Standards and Technology)	suppl	supplementary
diff	difference*			sys	systematic
diffr	diffractometer			tbar	mean path length
diffm	diffraction	NCA	number of connected atoms	temp	temperature
displace	displacement	ncs	noncrystallographic symmetry	tor	torsion angle
dist	distance	netI	net intensity	tran	transformation*
divg	divergence	NH	number of connected hydrogen atoms	transf	transformation*
dom	domain	nha	non-hydrogen atoms	transform	transformation*
dtime	deadtime	norm	normal	tvect	translation vector (PDB files)
ens	ensemble	nst	nonstandard	vert	vertical
eq	equatorial*	nucl	nucleic acid	wR	weighted agreement index
equat	equatorial*	num	number	wt	weight
equiv	equivalent	obs	observed		

* Terms with multiple definitions.

It is clear that the location specified for the local dictionary is only meaningful for applications running on the same computer or network, and therefore the ability to validate against this local dictionary is not portable. On the other hand, it may be that the local data names used by the authors of this CIF are not intended to have meaning outside their own laboratory.

3.1.8.2. The dictionary register

COMCIFS maintains a register of dictionaries known to it, including the identifying name and version strings within those dictionaries. The register also includes the location of each dictionary, expressed at present as a URL designed to allow retrieval

by file transfer protocol (ftp) from the IUCr server. Changes in the location of a particular dictionary file can be made by modifying the entry in the register, avoiding the problem of specifying a URL in a data file that would then become outdated if the dictionary was moved. Dictionary applications can consult the register (according to a protocol outlined below) to locate and retrieve the dictionaries needed for validating data files. It is of course essential that the validation software knows how to locate the register. The location is at present given by the URL <ftp://ftp.iucr.org/pub/cifdics/cifdic.register>.

The problem of changing URLs has therefore not disappeared completely, but is at least confined to the need to maintain one single address.