

5.4.6.3. Input monitor variables

These variables are returned by *CIFtbx* tools and are used to decide on subsequent actions in the program. The lengths of the character strings that hold data names and block names are controlled by the parameter `NUMCHAR` in the common-block declarations.

`bloc_`: *character string* containing the current data-block name.

`decp_`: *logical variable* is `.true.` if a decimal point is present in the input numeric value.

`diccat_`: *character string* containing the category name specified in the attached dictionaries.

`dicname_`: *character string* containing the root alias data name (see Section 5.4.7) specified in the attached dictionaries or, after a call to `dict_`, the name of the dictionary.

`dictype_`: *character string* containing the data-type code specified in the attached dictionaries. These types may be more specific (e.g. ‘float’ or ‘int’) than the types given by the variable `type_` (e.g. ‘numb’).

`dicver_`: *character string* containing the version of a dictionary after a call to `dict_`.

`glob_`: *logical variable* is `.true.` if the current data block is a global block. The application is responsible for managing the relationship of global data to other data blocks.

`list_`: *integer variable* containing the sequence number of the current looped list. This value may be used by the application to identify variables that are in different lists or that are not in a list (a zero value).

`long_`: *integer variable* containing the length of the data string in `strg_`.

`loop_`: *logical variable* is `.true.` if another loop packet is present in the current looped list.

`lzero_`: *logical variable* is `.true.` if the input numeric value is of the form `[sign]0.nnnn` rather than `[sign].nnnn`.

`posdec_`: *integer variable* containing the column number (position along the line, counting from 1 at the left) of the decimal point for the last number read.

`posend_`: *integer variable* containing the column number (position along the line, counting from 1 at the left) of the last character for the last string or number read.

`posnam_`: *integer variable* containing the starting column (position along the line, counting from 1 at the left) of the last name or comment read.

`posval_`: *integer variable* containing the starting column (position along the line, counting from 1 at the left) of the last data value read.

`quote_`: *character variable* giving the quotation symbol found delimiting the last string read.

`save_`: *logical variable* is `.true.` if the current data block is a save frame, otherwise `.false.`

`strg_`: *character variable* containing the data name or string representing the data value last retrieved.

`tagname_`: *character variable* containing the data name of the current data item as it was found in the CIF. May differ from `dicname_` because of aliasing.

`text_`: *logical variable* is `.true.` if another text line is present in the current input text block.

`type_`: *character variable* containing the data-type code of the current input data item. This will be one of the four-character strings ‘null’ (for missing data, the period or the question mark), ‘numb’ (for numeric data), ‘char’ (for most character data) or ‘text’ (for semicolon-delimited multi-line character data). For most purposes the type ‘text’ is a subtype of the type ‘char’, not a distinct

data type. *CIFtbx* permits multi-line text fields to be used whenever character strings are expected.

5.4.6.4. Output control variables

These variables are specified to control the processing by *CIFtbx* commands that write CIFs.

`aliaso_`: *logical variable* to control the use of data-name aliases for output items. If set to `.true.`, preferred synonyms from the input dictionary may be output (see Section 5.4.7). The default is `.false.`

`align_`: *logical variable* to control the column alignment of data values in `loop_` lists output to a CIF. The default is `.true.`

`esdlim_`: *integer variable* to set the upper limit of appended standard uncertainty (e.s.d.) integers output by `pnumb_`. The default value is 19, which limits standard uncertainties to the range 2–19.

`globo_`: *logical variable* which if set to `.true.` will cause the output data block from `pdata_` to be written as a global block.

`nblanko_`: *logical variable* controls the treatment of output blank strings. If set to `.true.`, output quoted blank strings will be converted to an unquoted period (i.e. to a data item of type null). Recall that *CIFtbx* treats an unquoted period or question mark as being of type null.

`pdec_`: *logical variable* controls the treatment of output decimal numbers. If set to `.true.`, a decimal point will be inserted into numbers output by `pnumb_` or `pnumbd_`. If set to `.false.`, a decimal point will be output only when needed. The default is `.false.`

`plzero_`: *logical variable* controls the treatment of leading zeros in output decimal numbers. If set to `.true.`, a zero will be inserted before a leading decimal point. The default is `.false.`

`pposdec_`: *integer variable* to set the column number (position along the line, counting from 1 at the left) of the decimal point for the next number to be output.

`pposend_`: *integer variable* to set the position of the ending column for the next number or *character string* to be output. Used to pad with zeros or blanks.

`pposnam_`: *integer variable* to set the starting column of the next name or comment to be output.

`pposval_`: *integer variable* to set the position of the starting column of the next data value to be output.

`pquote_`: *character variable* containing the quotation symbol to be used for the next string written.

`saveo_`: *logical variable* is set to `.true.` for `pdata_` to output a save frame, otherwise a data block is output.

`ptabx_`: *logical variable* is set to `.true.` for tab stops to be expanded to blanks during the creation of a CIF. The default is `.true.`

`tabl_`: *logical variable* is set to `.true.` for tab stops to be used in the alignment of output data. The default is `.true.`

`xmlout_`: *logical variable* is set to `.true.` to change the output style to XML conventions. Note that this is not a CML (Murray-Rust & Rzepa, 1999) output, but a literal translation from the input CIF. The default is `.false.`

`xmlong_`: *logical variable* is set to `.true.` to change the style of XML output if `xmlout_` is `.true.` When `.true.` (the default), XML tag names are the full CIF tag names with the leading underscore, `_`, removed. When `.false.`, an attempt is made to strip the leading category name as well.

5.4.7. Name aliases

CIF dictionaries written in DDL2 permit data names to be aliased or equivalenced to other data names. This serves two purposes. First, it allows for the different data-name structures used in DDL1