*character string* is put. The *logical* function is returned as `.true.` if the data name passes any requested dictionary validation checks.

`pnumb_` puts the specified data name *`name`*, single-precision number *`numb`* and an appended standard uncertainty *`sdev`* into the output CIF. The *logical* function is returned as `.true.` if the data name passes any requested dictionary validation checks.

`pnumd_` puts the specified data name *`name`*, double-precision number *`numb`* and an appended standard uncertainty *`sdev`* into the output CIF. The *logical* function is returned as `.true.` if the data name passes any requested dictionary validation checks.

`ptext_` puts the specified data name *`name`* and text string *`string`* into the output CIF. The data name will only be inserted on the first invocation of a sequence. The *logical* function is returned as `.true.` if the data name passes any requested dictionary validation checks. This command must be invoked repeatedly until the text is finished. The terminal semicolon character '`;`' is placed in the output CIF when the next call to `pchar_`, `pnumb_` or `pnumd_` is made, or if a call is made to `ptext_` for a different data name.

`pcmnt_` puts the specified comment string *`string`* into the output CIF. The *logical* function is always returned as `.true.`. The comment character '`#`' should not be included in the string. A blank comment is presented as a blank line without the leading '`#`'. The string `char(0)//char(0)` can be used to produce an empty comment with the leading '`#`'.

`prefx_` prefixes the specified string *`strg`* of length *`lstrg`* to subsequent lines of the output CIF. The total line length is still limited to the value given by the variable `line_` (the default is 80 characters). This function is useful when embedding a CIF into another text document, such as a PDB REMARK. The *logical* function is always returned as `.true.`.

`close_` closes the output CIF only. This command *must* be used if `pfile_` is used. This a subroutine call.

### 5.4.6. Variables

The *CIFtbx* library also contains a large number of variables declared in the common blocks in the file ciftbx.cmn that provide signals to the programmer on various aspects of the data reading and writing processes. These variables are described below in four broad categories, as shown in Table 5.4.6.1: general monitor variables, general control variables, input monitor variables and output control variables.

Note that for all but special applications only the basic variables `list_`, `loop_`, `strg_`, `text_` and `type_` are usually used. These variables supplement the argument lists of the various commands, providing essential status information.

Table 5.4.6.1. *CIFtbx variables*

| General monitor | General control | Input monitor | Output control |
|---|---|---|---|
| | `alias_` | | `aliaso_` |
| | | | `align_` |
| | `append_` | | |
| | | `bloc_` | |
| | | `decp_` | `pdecp_` |
| | | `diccat_` | |
| | | `dicname_` | |
| | | `dictype_` | |
| | | `dicver_` | |
| | | | `esdlim_` |
| `file_` | | | |
| | | `glob_` | `globo_` |
| | `line_` | | |
| | | `list_` | |
| | | `long_` | |
| `longf_` | | | |
| | | `loop_` | |
| | | `lzero_` | `plzero_` |
| | `nblank_` | | `nblanko_` |
| | | `posdec_` | `pposdec_` |
| | | `posend_` | `pposend_` |
| | | `posnam_` | `pposnam_` |
| | | `posval_` | `pposval_` |
| `precn_` | | | |
| | | `quote_` | `pquote_` |
| | `recbeg_` | | |
| | `recend_` | | |
| `recn_` | | | |
| | | `save_` | `saveo_` |
| | | `strg_` | |
| | | | `tabl_` |
| | `tabx_` | | `ptabx_` |
| | | `tagname_` | |
| `tbxver_` | | | |
| | | `text_` | |
| | | `type_` | |
| | | | `xmlout_` |
| | | | `xmlong_` |

#### 5.4.6.1. General monitor variables

These variables are returned by *CIFtbx* and provide information about the general status of processing.

`file_`: *character string* containing the file name of the current input file.

`longf_`: *integer variable* containing the length of the file name in *`file`*.

`precn_`: *integer variable* containing the line number (starting from 1) of the last line written to the output CIF.

`recn_`: *integer variable* containing the line number (starting from 1) of the last line read from the input CIF.

`tbxver_`: *character*32 variable* that is the *CIFtbx* version and date in the form `'CIFtbx version` *`N.N.N DD MMM YYYY`*`'` (some older versions of *CIFtbx* use a two-digit year and have a comma after the version number).

#### 5.4.6.2. General control variables

These variables control *CIFtbx* commands. The user may accept the default values or may store new values into these variables to change the behaviour of the commands.

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

`append_`: *logical variable* to control reuse of the direct-access file. If set to `.true.`, it will cause each call to `ocif_` to append the information found to the current CIF. The default is `.false.`.

`line_`: *integer variable* to set the input/output line limit for processing a CIF. The default value is 80 characters. This limit counts the visible printable characters of the line, not the system-dependent line terminators.

`nblank_`: *logical variable* to control the treatment of input blank strings. If set to `.true.`, `char_` or `test_`, it will return the type as 'null' rather than 'char' when encountering a quoted blank.

`recbeg_`: *integer variable* to give the record number of the first record to be used. May be changed by the user to restrict access to a CIF.

`recend_`: *integer variable* to give the record number of the last record to be used. May be changed by the user to restrict access to a CIF.

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