

## 5.4. CIFTBX: FORTRAN TOOLS FOR MANIPULATING CIFs

the input CIF as a direct-access file (*i.e.* for random access to parts of the CIF). `init_` is a *logical* function that is always returned with a value of `.true.`. The default device numbers for these files are 1, 2, 3 and 6.

`dict_` is an optional command for opening a dictionary `fname` and initiating various optional data checks, `checks`. The choices of checks to perform are given by a string of blank-separated five-character ‘check codes’, such as `valid` or `dtype`, which turn on checking for the validity of tags or types of values, respectively. `dict_` is a *logical* function which is returned as `.true.` if the named dictionary was opened and if the check codes are recognizable.

## 5.4.4. Read commands

These commands are used to read data from an existing CIF. Since CIF data are order-independent, most applications would work from a known list of data names (tags) and extract the desired values from the CIF in the order specified. However, some applications need to browse a CIF in the order of presentation. In *CIFTbx*, a blank name has the meaning of the next name in the file.

```
logical function ocif_ (fname)
  character fname*(*)
logical function data_ (name)
  character name*(*)
logical function bkmrk_ (mark)
  integer mark
logical function find_ (name, type, strg)
  character name*(*), type*(*), strg*(*)
logical function test_ (name)
  character name*(*)
logical function name_ (name)
  character name*(*)
logical function numb_ (name, numb, sdev)
  character name*(*)
  real numb, sdev
logical function numd_ (name, numb, sdev)
  character name*(*)
  double precision numb, sdev
logical function char_ (name, strg)
  character name*(*), strg*(*)
logical function cmnt_ (strg)
  character strg*(*)
subroutine purge_
```

`ocif_` requests the named CIF `fname` to be opened. The *logical* function is returned as `.true.` if the CIF can be opened.

`data_` specifies the data block `name` containing the data to be read from the CIF. The *logical* function is returned as `.true.` if the data block is found.

`bkmrk_` is a bookmark function that saves or restores the current position in the CIF so that data can be accessed nonsequentially if need be. The *logical* function is returned as `.true.` if there is space to store the current position or if the restored bookmark number is valid.

`find_` finds the requested item in the current data block. The *logical* function is returned as `.true.` if the item is found.

`test_` provides the data attributes of a data item in the current data block. The *logical* function is returned as `.true.` if the item is found. The data attributes are returned in the common-block variables `list_`, `type_`, `dictype_`, `diccat_` and `dicname_`.

`name_` identifies the next data name in the current data block. The *logical* function is returned as `.true.` if another data name exists in the data block and `.false.` if the end of the data block is reached. The name is returned in the function argument, `name`.

`numb_` returns the number `numb` and its standard uncertainty `sdev` (if appended) of a named data item `name`. The *logical* function is returned as `.true.` if the item is present and is a number. If the

item is either absent or cannot be recognized as a valid number, the function is returned as `.false.` and the original numeric argument values are not changed.

`numd_` returns the number `numb` and its standard uncertainty `sdev` (if appended) as double-precision variables of a named data item `name`. The *logical* function is returned as `.true.` if the item is present and is a number. If the item is either absent or cannot be recognized as a valid number, the function is returned as `.false.` and the original numeric argument values are not changed.

`char_` returns character or text strings, `strg`, of the named data item `name`. The *logical* function is returned as `.true.` if the item is present. If text lines are being read, this function is called repeatedly until the *logical variable* `text_is` is `.false.`.

`cmnt_` returns the next comment, `strg`, in the current data block. The *logical* function is returned as `.true.` if a comment is present. The initial comment character ‘#’ is not included in the returned string and a completely blank line is treated as a comment.

`purge_` closes all attached data files and clears all tables and pointers. This is a subroutine call.

## 5.4.5. Write commands

The following commands are available for writing data to a new CIF.

```
logical function pfile_ (fname)
  character fname*(*)
logical function pdata_ (name)
  character name*(*)
logical function ploop_ (name)
  character name*(*)
logical function pnumb_ (name, numb, sdev)
  character name*(*)
  real numb, sdev
logical function pnumd_ (name, numb, sdev)
  character name*(*)
  double precision numb, sdev
logical function pchar_ (name, string)
  character name*(*), string*(*)
logical function pcmnt_ (string)
  character string*(*)
logical function ptext_ (name, string)
  character name*(*), string*(*)
logical function prefix_ (strg, lstrg)
  character strg*(*)
  integer lstrg
subroutine close_
```

`pfile_` creates a new file with the specified file name `fname`. The *logical* function is returned as `.true.` if the file is opened. The value will be `.false.` if the file already exists.

`pdata_` puts the string `data_name` from the argument `name` into the output CIF. The *logical* function is returned as `.true.` if the block is created. The value will be `.false.` if the block name already exists. This command inserts the string `save_name` instead of the data-block name if the variable `saveo_` is set to `.true.`. If the prior block was a save frame, the necessary terminal `save_` is written for that block before the new block is started.

`ploop_` puts the specified data name `name` into the output CIF. On the first invocation of this command for a given loop, a `loop_` string is placed before the data name. The *logical* function is returned as `.true.` if the name passes any requested dictionary validation checks. Once a series of data names for a `loop_` header has been declared by calls to this function, all calls to `pchar_`, `ptext_`, `pnumb_` or `pnumd_` for the associated data values must be made with *blank* data names or the `loop_` will be terminated. (At the very least, the first character of these data names must be blank.)

`pchar_` puts the specified data name `name` and *character string* into the output CIF. If the data name is blank, only the

