

# 【C++】 Day29

▼ Class	C++
📅 Date	@December 31, 2021
🔗 Material	
# Series Number	
☰ Summary	

## 【Ch8】 The IO Library

### 8.2.2 File Modes

Each stream has an associated file mode that represents how the file may be used.

The following table lists the file modes and their meanings.

**Table 8.4. File Modes**

in	Open for input
out	Open for output
app	Seek to the end before every write
ate	Seek to the end immediately after the open
trunc	Truncate the file
binary	Do IO operations in binary mode

We can supply a file mode whenever we open a file-either when we call `open` or when we indirectly open the file when we initialize a stream from a file name. The modes that we can specify have the following restrictions:

- `out` may be set only for an `ofstream` or `fstream` object
- `in` may be set only for an `ifstream` or `fstream` object
- `trunc` may be set only when `out` is also specified
- `app` mode may be specified so long as `trunc` is not.

If `app` is specified, the file is always opened in output mode, even if `out` was not explicitly specified

- By default, a file opened in `out` mode is truncated even if we do not specify `trunc`.  
To preserve the contents of a file opened with `out`, either we must also specify `app`, in which case we can write only at the end of the file, or we must also specify `in`, in which case the file is open for both input and output.
- The `ate` and `binary` modes may be specified on any file stream object type and in combination with any other file modes.

Each file stream type defines a default file mode that is used whenever we do not otherwise specify a mode.

Files associated with an `ifstream` are opened in `in` mode; files associated with an `ofstream` are opened in `out` mode; and files associated with an `fstream` are opened with both `in` and `out` mode.

### *Opening a File in out Mode Discards Existing Data*

By default, when we open an `ofstream`, the contents of the file are discarded. The only way to prevent an `ostream` from emptying the given file is to specify `app`:

```
//file1 is truncated in each of these cases
ofstream out("file1"); //out and trunc are implicit
ofstream out2("file1", ofstream::out); //trunc is implicit
ofstream out3("file31", ofstream::out | ofstream::trunc);

//to preserve the file's contents, we must explicitly specify app mode
ofstream app("file2", ofstream::app); //out is implicit
ofstream app2("file2", ofstream::out | ofstream::app);
```

*Warning: The only way to preserve the existing data in a given file is to explicitly specify app mode.*

### *File Mode is Determined Each Time open is Called*

The file mode of a given stream may change each time a file is opened.

```
ofstream out; //no file mode is set
out.open("scratchpad"); //mode implicitly out and trunc
out.close(); //close out so we can use it for a different file
```

```
out.open("precious", ofstream::app); //mode is out and app  
out.close();
```

*Note: Any time open is called, the file mode is set, either explicitly or implicitly. Whenever a mode is not specified, the default value is used.*