

## Resource file overview

Jamie Doornbos, MAXIS, January 2000

A resource file is an interface where clients request and add blocks of memory by type and id or name. The basic interface is defined in class `iResFile`. Subclasses implement various functionalities:

### ***IFFResFile2.***

Subclass of `iResFile`. Makes calls to the file system (through `CTGFile`) to maintain a list of chunks of data within the file. Each chunk is a resource and contains the type, id, and name. A relatively recent optimization allows for an internally used map resource that has the header and position for all other resources in the file.

### ***ResolveResFile***

A subclass of `iResFile`, `ResolveResFile` implements a partial interface. It makes read-only calls to the file system (through `Resolve`) and maps each resource to a file with the id prefixed and the type suffixed. For example, a resource with type "cst\0" and id 1 would be a file named "#1#SomeName.cst". It is only meaningful to use types that end in 0. A `ResolveResFile` is a directory ending in ".dir", and the resources are files in the directory.

### ***ChainResFile***

Subclass of `iResFile`; keeps a list of `iResFile` pointers (passed in by subclass), and passes requests along in order until one of the files responds without an error. Also for each file in the chain, a representation of the allowed types is kept, specified by the subclass and defaulting to all types. For example, an SPF file allows only certain graphical types to be added, while the IFF allows all but those types.

### ***SeqResFile***

Subclass of `ChainResFile`. Generates a list of `iResFile` pointers using a mapping of file name extension to runtime type. When asked to open a file by name, it tries opening a file of each type and extension, and adds the pointer into the `ChainResFile` list if successful. For example, `StdResFile` uses this class to create a single-file binding of the files "X.iff", "X.spf", and "X.dir" when a file named "X" is requested.

### ***StdResFile***

Subclass of `SeqResfile`. Serves as a placeholder for generic clients of the resource file system to "do the right thing" so that most Sims data files can be opened with this class. The constructor sets up the opening sequence (basically IFF then SPF then DIR), and the `Open` call sets up the allowed types.

### ***ObjResFile***

Subclass of `StdResFile`. This class is used by the object system for all object files. It has been the one subclass most often changed when implementing new features and optimizations for the objects. For example, to avoid loading the same resource more than once, it presents an easy `Get/Release` interface for types specific to the object system.

### ***SlangResFile***

Subclass of `StdResFile` created just for slang. It adds the FAM extension to the opening sequence so that the tutorial house may be translated.