freq

COLLABORATORS

	TITLE : freq	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		July 29, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

freq

1.1 S0

```
File Requester Plug-ins -- Stuart Ferguson 1/20/95

1  Introduction

2  Server Interface
    2.1  Process
    2.2  Activation Parameters
       (1)  Public types
    2.3  File Type Conversion
       (2)  Public types
```

1.2 s1

File requesters are in the tools library as part of the system-generic interface components. File requesters will work the same way on all systems, with the default mode being to use the local standard file requester provided by the host system. In addition, the file requester interface here is defined as a plug-in server class, so in the future more and different types of file requesters may be provided.

1.3 s2

The file requester server class is "FileRequester" and the version is 1.

```
      2.1  Process
      2.2  Activation Parameters
      2.3  File Type Conversion
```

1.4 s2.1

The activation function for a file requester server performs the entire operation at once. The host passes a local structure defining the request, the server opens a requester and gets a reply from the user and returns that result back to the calling host.

A file requester should respect the current directory of the caller when processing the request and should leave it unchanged. It should also the user to specify relative path names as well as fully qualified ones. (Both of these issues have to be specially coded when using the Windows common file dialogue.)

1.5 s2.2

The host sets up the local data with the type of request and the initial filename and path, if any. The 'reqType' request type is one of the FREQ_* values defined below. GENERIC might be either a load or a save. A possible title for this request is passed in 'title'. The 'fileType' string is set by the host as a hint about the type of file being requested. Some possible values might be "Scenes," "Objects," "Images," etc. A null pointer indicates no particular file type. The initial path is set in the 'path' string and the initial filename is set in 'baseName', either of which may be empty strings. All three character pointers must point to buffers at least 'bufLen' characters long, a value the host must also set.

When the activation function returns, the user has made a choice. If they canceled the request, 'result' will be false. If not, the full file name is in 'fullName' and path and base names have been updated to reflect the full name.

(1) Public types

```
typedef struct {
  int              reqType, result;
  const char      *title;
  const char      *fileType;
  char            *path;
  char            *baseName;
  char            *fullName;
  int              bufLen;
} FileReq_Local;

#define FREQ_GENERIC     0
#define FREQ_LOAD        1
#define FREQ_SAVE        2
. . .
```

1.6 s2.3

The host should provide a global function to get filename patterns
from file type strings, returned by ID "File Type Pattern". This
function takes a file type string and returns a pattern used to match
files of the given type. If the type is unknown, or the string is
null, a generic wildcard is returned to match all files on the system
which will varry depending on the host filesystem.

    (2) Public types
  . . .
  typedef const char   *FileTypeFunc (const char *);