

## Die Bibliothek SysLibFileStream.lib

Diese Bibliothek unterstützt Funktionen, die die entsprechenden ANSI C Funktionen für Dateioperationen nachbilden. Die Abarbeitung erfolgt synchron.

**Hinweis:** Zur Benutzung der in der Version SysLibFileStream23.lib enthaltenen Funktionen muss das Laufzeitsystem mit der zusätzlichen Komponente SysFileStream ausgestattet sein.

Die Funktionen:

SysLibFileStream Funktion	ANSI C Funktion	Datentyp	Bemerkungen
SysFileStreamFOpen	*fopen( char *filename, char *mode );	DWORD	Datei mit dem Namen <i>filename</i> als wird als Stream geöffnet; mögliche Werte für inputvariable <i>Mode</i> : 'w' (write), 'r' (read), 'a' (append), '+', 'b', 't'
SysFileStreamClearerr	clearerr(FILE* pFile);	DINT	interner Fehlerstatus von <i>pFile</i> wird gelöscht; liefert immer 1
SysFileStreamFClose	fclose( FILE *pFile );	DINT	alle offenen Streams werden geschlossen (außer <i>stdin</i> , <i>stdout</i> , <i>stderr</i> ). Liefert EOF im Fehlerfall, sonst 0.
SysFileStreamFEOF	*feof( FILE* pFile );	DINT	liefert !=0, falls für <i>pFile</i> das Dateieinde erreicht ist
SysFileStreamFError	ferror( FILE* pFile );	DINT	liefert !=0, falls für <i>pFile</i> ein Fehler aufgetreten ist
SysFileStreamFFlush	fflush( FILE *pFile );	DINT	Ausgeben von noch intern gepufferten Zeichen
SysFileStreamRemove	remove( char* filename);	BOOL	Datei wird gelöscht; liefert 1 wenn OK, 0 im Fehlerfall
SysFileStreamRename	rename( char* filename);	BOOL	Umbenennen einer Datei; liefert 1 wenn OK, 0 im Fehlerfall
SysFileStreamRewind	rewind( FILE* pFile);	DINT	setzt die Dateiposition auf den Anfang und löscht internen Fehlerstatus; liefert immer 1
SysFileStreamFGetC	fgetc( FILE *pFile );	DINT	liefert das nächste Zeichen aus dem Strom (0--255), SYSFILESTREAM_EOF bei Fehler
SysFileStreamFGetPos	*fgetpos( FILE* pFile, fpos_t * ptr );	DINT	speichert die aktuelle Dateiposition von pFile nach <i>ptr</i> ; <i>fpos_t</i> ist hier als "unsignedlong" (32 bits) definiert
SysFileStreamFSetPos	fsetpos( FILE* pFile, fpos_t * ptr );	DINT	setzt die Dateiposition von Stream pFile entsprechend <i>ptr</i> ; <i>fpos_t</i> ist hier als "unsignedlong" (32 bits) definiert; <i>pFPos</i> :DWORD; (* pointer !!*)
SysFileStreamFGetS	* fgets( char * str, int n, FILE * pFile );	POINTER TO STRING	liest maximal n--1 Zeichen nach s (Abschluß automatisch mit 0); Abbruch bei '\n', das '\n' wird nach s übernommen; Rückgabewert: s bzw. 0 (bei Dateieinde oder Fehler)

<b>SysLibFileStream Funktion</b>	<b>ANSI C Funktion</b>	<b>Datentyp</b>	<b>Bemerkungen</b>
SysFileStreamFPrintf_Int	fprintf( FILE* pFile, char* szFormat, intnArg);	DINT	formatierte Ausgabe in den Strom pFile; Einschränkung gegenüber C: nur 1 Argument des Typs INT/DINT etc. kann gedruckt werden szFormat sollte sein: z.B. '%d'
SysFileStreamFPrintf_Real	fprintf( FILE* pFile, char* szFormat, float fArg);	DINT	formatierte Ausgabe in den Strom pFile; Einschränkung gegenüber C: nur 1 Argument des Typs REAL etc. kann gedruckt werden szFormat sollte sein: z.B. '%f'
SysFileStreamFPrintf_String	fprintf( FILE* pFile, char* szFormat, char *pcArg);	DINT	formatierte Ausgabe in den Strom pFile; Einschränkung gegenüber C: nur 1 Argument des Typs STRING etc. kann gedruckt werden szFormat sollte sein: z.B. '%s'
SysFileStreamFPutC	fputc( int c, FILE *pFile );	DINT	Schreiben des Zeichens (unsignedchar)c in den Stream pFile Rückgabewert: c (nach DINT konvertiert) oder SYSFILESTREAM_EOF im Fehlerfall
SysFileStreamFPutS	fputs( char* str, FILE * pFile );	DINT	Schreiben der Strings s in den Stream pFile Rückgabewert: str (Zeiger auf String) oder SYSFILESTREAM_EOF im Fehlerfall
SysFileStreamFRead	fread( void* ptr, size_t size, size_t nobj, FILE* pFile );	DWORD	nobj Objekte der Größe size werden von pFile nach ptr gelesen Rückgabewert: Anzahl der tatsächlich gelesenen Objekte
SysFileStreamFWrite	fwrite( void* ptr, size_t size, size_t nobj, FILE* pFile );	DWORD	nobj Objekte der Größe size werden von ptr nach pFile geschrieben Rückgabewert: Anzahl der tatsächlich gelesenen Objekte
SysFileStreamFScanf_Int	fscanf( FILE* pFile, char* szFormat, int * pnArg);	DINT	formatierte Eingabe aus Stream pFile; Einschränkung gegenüber C: nur 1 DINT Argument kann eingelesen werden; szFormat sollte sein: z.B. '%d'
SysFileStreamFScanf_String	fscanf( FILE* pFile, char* szFormat, char *pcArg);	DINT	formatierte Eingabe aus Stream pFile; Einschränkung gegenüber C: nur 1 STRING Argument kann eingelesen werden; szFormat sollte sein: z.B. '%s'
SysFileStreamFScanf_Real	fscanf( FILE* pFile, char* szFormat, float* pfArg);	DINT	formatierte Eingabe aus Stream pFile; Einschränkung gegenüber C: nur 1 REAL Argument kann eingelesen werden; szFormat sollte sein: z.B. '%f'

<b>SysLibFileStream Funktion</b>	<b>ANSI C Funktion</b>	<b>Datentyp</b>	<b>Bemerkungen</b>
SysFileStreamFSeek	fseek( FILE* pFile, long offset, int origin);	DINT	setzt die Dateiposition auf offset Bytes relativ zu origin; Werte für origin: SEEK_SET=Dateianfang, SEEK_CUR=aktuelle Position; SEEK_END=Dateiende; 0=OK
SysFileStreamFTell	ftell( FILE* pFile);	DINT	liefert die aktuelle Dateiposition (vom Anfang) in Bytes (-1 bei Fehler)