

<http://hp.vector.co.jp/authors/VA019876/>

Hello World

```
#include<stdio.h>
void main(){
    printf("Hello, World! ??/n");
}
```

「??/」は三文字表記で'¥'を表す。最近の AS400 では「\n」でも大丈夫みたい？

コンパイル

```
CRTCMOD
CRTPGM
```

システム A P I を使う

システム A P I で D S M 画面クリア

```
#include<stdio.h>
#include "QSYSINC/H/QSNAPI"

void main() {
    QsnClrScr(_C_Get_Ssn_Handle(), '0', NULL);
    printf("Hello, World!??/n");
}
```

ファイルの読み書き

A S / 4 0 0 の C 言語環境でファイル I / O を行うには、大きく 3 種類の方法が存在します。

RPG などと同じく、AS/400 のファイル (PF,LF,DSPF,PRTF,ICFF など) にレコード単位で I / O する

I L E - C には、そのための関数群が提供されています。これら AS/400 独自の関数を使うことにより RPG などとある程度同じイメージで、レコード単位のファイル I / O が可能です。

C から、ファイル (ストリームファイル) に I / O する標準関数を使い、AS/400 の P F , L F に出力する

C からは連続したバッファ出力を行うと、O S / 4 0 0 が物理ファイルのレコード形式にマップする。C からは連続したストリームファイルに見えるが、実際のデータは、ファイルの長さに分割されて出力される。

C から、ファイル (ストリームファイル) に I / O する標準関数を使い、I F S (統合ファイルシステム) に出力する。

I F S (統合ファイルシステム) は、ライブラリー形式以外に、標準のストリームファイル形式をサポートしています。そのため、ほぼ、他のプラットフォーム (U N I X , W I N D O S ) とコンパチの使い方が可能になります。

標準関数でアクセス

```
#include <stdio.h>

int main(void) {
    FILE *fp;
```

```

char data[100];

if (( fp = fopen ( "CLAB/CLAB1P", "ab" )) == NULL ) {
    printf ( "OPEN ERROR\n" );
    exit ( 1 );
}

memcpy(data,"1234", 4);
fwrite(data, sizeof(char), 4, fp);
fclose ( fp );
return(0);
}

```

## レコード単位でファイルへアクセス

```

#include <stdio.h>
#include <stdlib.h>
#include <recio.h>

typedef struct {
    char srcseq[6];
    char srcdat[6];
    char srcdta[80];
} SRC;

int main(void)
{
    _RFILE      *fp;
    SRC         src;

    if (( fp = _Ropen ( "CLAB/CLAB1P", "ar" )) == NULL )
    {
        printf ( "OPEN ERROR\n" );
        exit ( 1 );
    }

    memcpy(src.srcseq,"000100",6);
    memcpy(src.srcdat,"999999",6);
    memset(src.srcdta,'A',80);

    _Rwrite(fp, &src, 92);

    _Rclose ( fp );

    return(0);
}

```

コード	機能
rr	レコードの読取り専用
wr	レコードの書込み専用。データがある場合には消去された後書込み。
ar	ファイルの終りへのレコードの書込み ( 追加 )
rr+	レコードの読取り、書込み、更新
wr+	レコードの読取り、書込み、更新。データがある場合消去された後書込み。
ar+	レコードの読取りおよび書込。すべてのデータはファイルの終りに書込

## 読込時は

```

_Rreadn ( FP, C ,RECLEN ,__DFT) ; // ロック
_Rreadn ( FP, C ,RECLEN ,__NO_LOCK) ; // ロックしない

```

## DDS の定義を構造体として include する

\* SHOWUSR オプションで、コンパイルするとコンパイルリストに構造体名が書かれているのでチェックする

```
#include <stdio.h>
#include <stdlib.h>
#include <recio.h>
// #pragma mapinc(" 適当な名前 ", "QDBLIB/TABLE(*all)", "input or both")
#pragma mapinc("SRC", "CLAB1P(*all)", "both")
#include "SRC"

int main(void)
{
    _RFILE
    CLAB_CLAB1P_CLAB1P_both_t      *fp;
                                   src;

    if (( fp = _Ropen ( "CLAB/CLAB1P", "ar" )) == NULL )
    {
        printf ( "OPEN ERROR\n" );
        exit ( 1 );
    }

    memcpy(src.SRCSEQ, "000100", 6);
    memcpy(src.SRCDAT, "999999", 6);
    memset(src.SRCDTA, 'A', 80);

    _Rwrite(fp, &src, 92);

    _Rclose ( fp );

    return(0);
}
```