



IBM Power Systems - IBM i

# Modernisation, développement d'applications et DB2 sous IBM i *Technologies, outils et nouveautés 2012-2013*

8 et 9 avril 2013 – IBM Client Center Paris, Bois-Colombes

## **S07 - RPG et IFS**

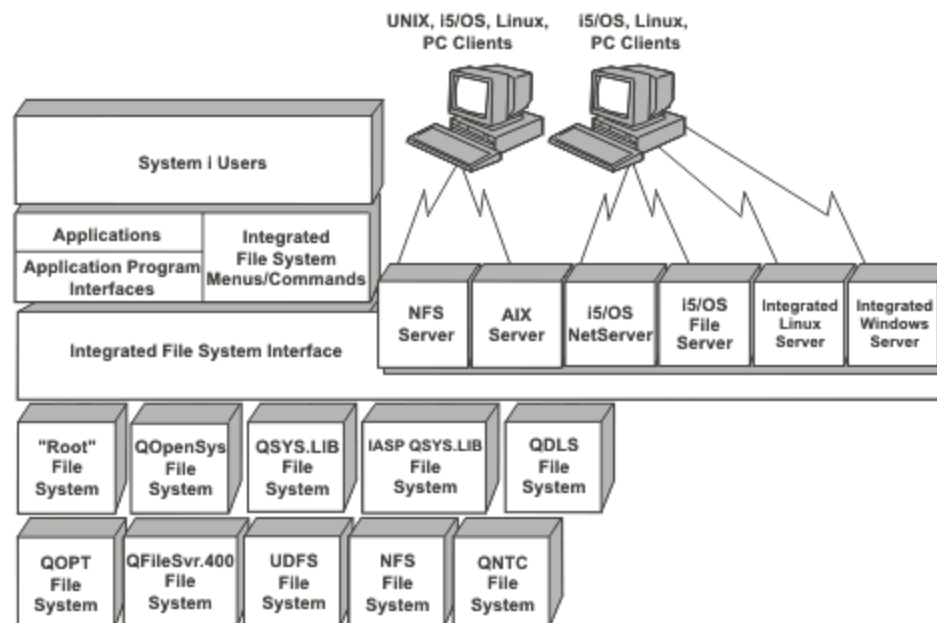
*Lundi 8 avril – 16h00-17h30*

Nathanaël BONNET – GAIA Mini Systèmes

# IFS

## ■ Integrated File Systems

- Fait partie de l'i5/OS
- Supporte des « flux d'entrée/sortie » pour l'ensemble des données du système
- Supporte différents types de systèmes de fichiers
  - « Root » ou /
    - DOS, OS/2
  - QOpenSys
    - UNIX, POSIX
  - QSYS.LIB
    - i5/OS
  - ...



RV01N21-3

# IFS – comparatif des différents systèmes de fichiers

Capability	"root" (/)	QOpenSys	QSYS.LIB <sup>16</sup>	QDLS	QNTC
Standard part of i5/OS®	Yes	Yes	Yes	Yes	Yes
Type of file	Stream	Stream	Record <sup>12</sup>	Stream	Stream
File size limit	T2=1 TB; T1=128 GB	T2=1 TB; T1=128 GB	<a href="#">Database file sizes</a>	4 GB	Varies <sup>17</sup>
Access through i5/OS file server	Yes	Yes	Yes	Yes	Yes
Direct access through file server I/O processor <sup>1</sup>	No	No	No	No	Yes
Comparative speed for open/close	Medium <sup>2</sup>	Medium <sup>2</sup>	Low <sup>2</sup>	Low <sup>2</sup>	Medium <sup>2</sup>
Case-sensitive name search	No	Yes	No <sup>4</sup>	No <sup>5</sup>	No
Maximum length of each component in path name	255 characters <sup>3</sup>	255 characters <sup>3</sup>	10/6 characters <sup>6</sup>	8/3 characters <sup>7</sup>	255 characters <sup>3</sup>
Maximum length of path name <sup>8</sup>	16MB	16MB	55 – 66 characters <sup>4</sup>	82 characters	255 characters
Maximum length of extended attributes for an object	2GB	2GB	Varies <sup>9</sup>	32KB	0 <sup>18</sup>
Maximum levels of directory hierarchy within file system	No limit <sup>10</sup>	No limit <sup>10</sup>	3	32	127
Maximum links per object <sup>11</sup>	Varies <sup>15</sup>	Varies <sup>15</sup>	1	1	1
Supports symbolic links	Yes	Yes	No	No	No
Object or file can have owner	Yes	Yes	Yes	Yes	No
Supports integrated file system commands	Yes	Yes	Yes	Yes	Yes
Supports integrated file system APIs	Yes	Yes	Yes	Yes	Yes
Supports hierarchical file system (HFS) APIs	No	No	No	Yes	No
Threadsafe <sup>13</sup>	Yes	Yes	Yes	No	Yes
Supports object journaling	Yes	Yes	Yes <sup>14</sup>	No	No

# Usage

- Ce que nous appelons IFS par abus de langage
  - Root ou /
    - Contient lui-même les autres systèmes de fichiers
- Nous utilisons l'IFS
  - QSYS fait partie de l'IFS !
  - De nombreuses fonctionnalités de l'OS sont hébergées dans l'IFS
    - PASE, Machine virtuelle Java, PHP ...
    - Serveurs Web Apache
    - Serveurs Java WebSphere, Lotus Domino, IWS ...
    - Client de services web
    - Certificats SSL
    - XML ToolKit
    - ...
  - Et de nombreuses fonctionnalités nouvelles font appel à PASE

# Usage

- Stream file
  - Fichier flux
  - Fichier plat (non BD, non zoné)
- Exemple
  - Fichier XML, HTML, log, texte, image ...
- Performance
  - Meilleure que DB2
    - L'usage n'est pas le même, comparons ce qui est comparable !
- Modernisation d'application
  - Capacité à traiter des flux XML, csv, log ...
    - Lecture
    - Ecriture
    - Mise à jour

## Mise en œuvre

- RPG ne sait manipuler que des fichiers BD
- Root ou QOpenSys sont des systèmes de fichiers issus d'UNIX
  - Utilisables en langage C
  - Utilisables via les API Unix de l'IBM i
- C'est grâce à l'ILE que nous pouvons utiliser les API Unix dans des programmes RPG
  - Le compilateur RPG ILE utilise le \*BNDDIR QC2LE, contenant les API Unix, ainsi que la bibliothèque standard C ...

# Différence entre UNIX et RPG

## ■ Fichier

- RPG : un fichier possède un format, c`ad une structure (colonnes)
- Unix : un fichier est une suite d'octets
  - Pas de notion d'enregistrement
  - Eventuellement des caractères sp`eciaux : CRLF (Retour Charriot / Saut de ligne)
- RPG : un fichier est manipul`e par son nom ou son nom de format (nom interne RPG `eventuellement)
- Unix : un fichier est manipul`e par un « handler »
  - C'est un nombre d`etermin`e `a l'ouverture du fichier
  - A transmettre `a chaque op`eration suivante sur le fichier

# Utilisation des API Unix en RPG

- Une API Unix est une procédure
  - Style C
    - Types de paramètres exprimés avec les types C
    - Chaînes de caractères à terminaison nulle
    - Pointeur ...
- Nécessite de prototyper l'API en RPG
- Où trouver les API à utiliser
  - Infocentre
    - API finder
      - UNIX API -> Integrated File System APIs  
(<http://publib.boulder.ibm.com/infocenter/iseres/v7r1m0/index.jsp?topic=%2Fapis%2Funix2.htm>)



# API courantes

API	Description
Open	Ouvrir ou créer un fichier
Opendir	Ouvrir un répertoire (pour lister les fichiers)
Close	Fermer un fichier
Closedir	Fermer un répertoire
Read	Lire un nombre d'octets
Readdir	Lire une entrée de répertoire
Write	Ecrire un nombre d'octets
Lseek	Déplacement dans le fichier
Stat	Statut du fichier (information)
Access	Le fichier est-il accessible pour l'opération demandée ?

# Comment prototyper les API en RPG

- Depuis le prototype C

```

open()--Open File

Syntax

#include <fcntl.h>

int open(const char *path, int oflag, . . .);
  
```

- En RPG

```

D open                PR                10I 0  EXTPROC ( *CWIDEN : 'open' )
D path                *                VALUE OPTIONS (*STRING)
D oflag               10I 0  VALUE
D* ...
  
```

- Où

- int : valeur de retour, le « handle » du fichier
- open : nom de la procédure. Sensible à la casse dans « EXTPROC »
- Const char \*path : pointeur sur le nom du fichier
- Int oflag : mode d'ouverture : création, lecture ...

# Comment prototyper les API en RPG

## ■ Pour prototyper

- Infocentre
- Converting form C prototypes to RPG prototypes (Barbara Morris) :  
<http://www.scottklement.com/rpg/callc.html>

## ■ Google

- Extproc open
- Scott Klement

Environ 56 700 résultats (0,11 secondes)

Conseil : [Recherchez des résultats uniquement en français](#). Vous pouvez indiquer votre langue de recherche sur la page [Préférences](#).

### [The Basics of Stream Files](#)

[www.scottklement.com/rpg/.../basics.htm...](http://www.scottklement.com/rpg/.../basics.htm...) - Traduire cette page

Therefore, when we make our RPG prototype, we'll use the `EXTPROC()` keyword to refer to **open** as an all-lowercase procedure name. (3): This is where we ...

### [Easy IFS file to RPG var pgm « aaronbartell](#)

[mowyourlawn.com/blog/?p=161](http://mowyourlawn.com/blog/?p=161) - Traduire cette page

8 sept. 2009 – D open pr 10i 0 `extproc('open')` D filename \* value options(\*string) D openflags 10i 0 value. D mode 10u 0 value options(\*nopass) D codepage ...

## ■ IBM

- Nécessite l'option 13 du SS1 « System Openness Includes »
  - Gratuit, utile, productif, donc fortement recommandé !
- Contient les prototypes pour chaque API, dans QSYSINC
  - Pas pour toutes les API !

# Comment trouver le prototype dans QSYSINC

- Depuis l'infocentre

## open()--Open File

### Syntax

```
#include <fcntl.h>
```

```
int open(const char *path, int oflag, . . .);
```

Service Program Name: QPOLLIB1

Default Public Authority: \*USE

Threadsafe: Conditional; see [Usage Notes](#).

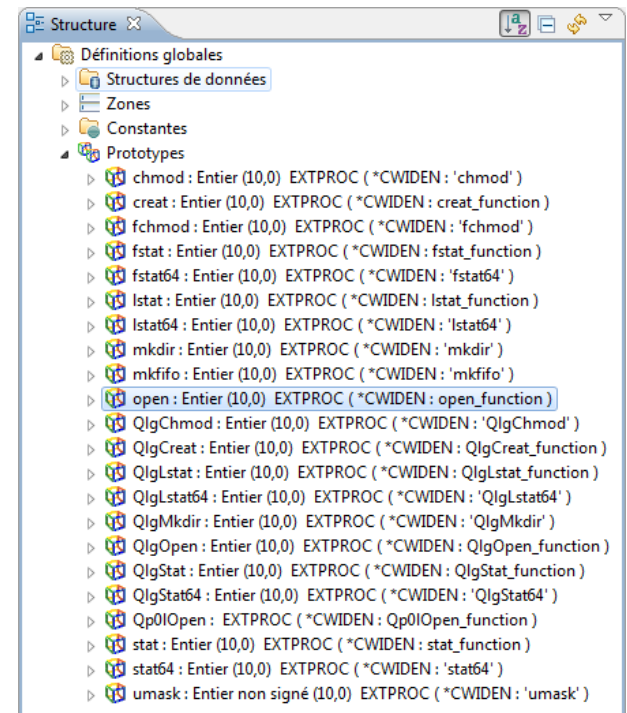
- Dans QSYSINC
  - Fichier QRPGLSRC
  - membre FCNTL

```

/IF DEFINED(LARGE_FILES)
D open_function C 'open64'
/ELSE
D open_function C 'open'
/ENDIF
D open PR 10I 0 EXTPROC(*CWIDEN : open_function)
D path * VALUE OPTIONS(*STRING)
D oflag 10I 0 VALUE
D mode VALUE LIKE(mode_t)
D OPTIONS(*NOPASS)
D conversion_id...
D 10U 0 VALUE
D OPTIONS(*NOPASS)
D text_file_creation_conversion_id...
D 10U 0 VALUE
D OPTIONS(*NOPASS)

```

- membre IFS pour toutes les API IFS



# Cas pratique

## ■ Agenda

- Ecrire des données dans un fichier texte
- Remarques sur les API write et open
- Lire les données d'un fichier
- Ligne à ligne
- Lister le contenu d'un répertoire
- Tester l'existence d'un fichier

# Ecrire des données

## ■ Cinématique

- Ouverture du fichier
  - API open
- Ecriture des données
  - API write
- Fermeture du fichier
  - API close

```
// int open(const char *path, int oflag, . . .);
d open          pr          10i 0 extproc( *CWIDEN : 'open' )
d path          *          value options(*string)
d oflag         10i 0 value

// ssize_t write(int file_descriptor, const void *buf, size_t nbytes);
d write        pr          10i 0 extproc( *CWIDEN : 'write' )
d filedesc     10i 0 value
d buf          *          value
d nbytes       10i 0 value

// int close(int fildes);
d close        pr          10i 0 extproc( *CWIDEN : 'close' )
d fildes       10i 0 value
```

- Eléments supplémentaires pour open

```
// Flags pour open
* Open for writing only
d O_WRONLY     C          2
* Create file if it doesn't exist
d O_CREAT      C          8
* Truncate flag
d O_TRUNC      C          64
```

# Ecrire des données

```
// descripteur de fichier
d fd          s          10i 0 inz
// données à écrire
d data        s          128a  inz varying

/free

// création ou écrasement du fichier :
fd = open( '/home/bonnet/session7.log'
           :O_WRONLY + O_CREAT + O_TRUNC ) ;
if ( fd < 0 ) ;
  dsply 'Ouverture "/home/bonnet/session7.log" impossible' ;
  return ;
endif ;

// écriture de données :
data = 'Bienvenue dans session RPG & IFS !' ;
if ( write( fd
           :%addr( data : *data )
           :%len( data ) ) < 0 ) ;
  dsply 'Ecriture dans "/home/bonnet/session7.log" impossible' ;
  return ;
endif ;

// fermeture
if ( close( fd ) < 0 ) ;
  dsply 'Erreur à la fermeture du fichier' ;
  return ;
endif ;

dsply 'Fichier "/home/bonnet/session7.log" créé' ;
return ;
/end-free
```

## Open

- Renvoie un « file descripteur »
  - un entier qui représente le fichier et permet par la suite de le manipuler
- Nécessite le nom réel du fichier
- Utilise des « flags » :  
O\_WRONLY, O\_CREAT et O\_TRUNC
  - On en reparle

## Write, nécessite :

- Le file descriptor (cf open)
- Les données (pointeur)
- La longueur des données

## Close, nécessite :

- Le file descriptor (cf open)

# Ecrire des données

- Appel

```
call s7_1
DSPLY Fichier "/home/bonnet/session7.log" créé
```

- dspf '/home/bonnet/session7.log'

```
Browse : /home/bonnet/session7.log
Record : _____1 of _____1 by _18
Control : _____

.....1.....2.....3.....4.....+..
*****Beginning of data*****
Bienvenue dans session RPG & IFS !
*****End of Data*****
```



# write

## ■ write

- Nécessite un pointeur sur les données
  - Ce n'est pas très naturel en RPG
- Il est possible de prototyper d'une façon différente

```
// ssize_t write(int file_descriptor, const void *buf, size_t nbytes);  
d write          pr          10i 0 extproc( *CWIDEN : 'write' )  
d filedesc      10i 0 value  
d buf           a   len( 16000000 ) const  
d               options( *varsize )  
d nbytes       10i 0 value
```

- Ce qui permet des appels plus « naturels »

```
// écriture de données :  
data = 'Bienvenue dans session RPG & IFS !' ;  
if ( write( fd  
          :data  
          :%len( data ) ) < 0 ) ;
```

```
if ( write( fd  
          :'Bienvenue dans session RPG & IFS !'  
          :%len( 'Bienvenue dans session RPG & IFS !' ) ) < 0 ) ;
```

# open

## ■ open

- C'est une API qui offre de nombreuses possibilités

```

D open          PR          10I 0  EXTPROC(*CWIDEN : open_function)
D path         *          VALUE OPTIONS(*STRING)
D oflag        10I 0  VALUE
D mode         VALUE LIKE(mode_t)
D              OPTIONS(*NOPASS)
D conversion_id...
D              10U 0  VALUE
D              OPTIONS(*NOPASS)
D text_file_creation_conversion_id...
D              10U 0  VALUE
D              OPTIONS(*NOPASS)

```

- oflag
  - Mode d'ouverture du fichier
    - Création/écrasement, lecture seule, lecture/écriture, écriture seule, fichier texte, CCSID, ouverture en mode partagé, lecture synchrone/asynchrone ...
- mode
  - Pour les fichiers en création (cf oflag)
    - Permissions sur le fichier créé
- conversion\_id
  - CCSID ou code page des données à lire/écrire depuis/vers le fichier
- text\_file\_creation\_conversion\_id
  - CCSID ou code page des données manipulés dans le RPG

# open

## ■ open

### – oflag

- Les valeurs possibles sont définies par des constantes (puissance de 2)
- Peuvent se combiner (certaines combinaisons invalides)

Nom	Rôle	Valeur
O_RDONLY	Lecture seule	(2**0) 1
O_WRONLY	Ecriture seule	(2**1) 2
O_RDWR	Lecture et écriture	(2**2) 4
O_CREAT	Créer si inexistant	(2**3) 8
O_EXCL	Avec O_CREAT, provoque une erreur si le fichier existe	(2**4) 16
O_CCSID O_CODEPAGE	Indique un CCSID ou un code page pour le fichier	(2**5) 32 (2**23)
O_TRUNC	Avec O_RDWR ou O_WRONLY, écrase le fichier	(2**6) 64
O_APPEND	Ajout en fin de fichier	(2**8) 256
O_TEXTDATA	Les données sont du texte	(2**24)
O_TEXT_CREAT	Permet les conversions de CCSID	(2**25)
...		

# open

## ■ open

### – mode

- Les valeurs possibles sont définies par des constantes (puissance de 2)
- Peuvent se combiner

Nom	Rôle	Valeur
S_IRUSR	Lecture pour le propriétaire	(2**8) 256
S_IWUSR	Ecriture pour le propriétaire	(2**7) 128
S_IXUSR	Exécution pour le propriétaire	(2**6) 64
S_IRWXU	Lecture, écriture, exécution =S_IRUSR+S_IWUSR+S_IXUSR	448
S_IRGRP	Idem pour le groupe	(2**5) 32
S_IWGRP		(2**4) 16
S_IXGRP		(2**3) 8
S_IRWXG		56
S_IROTH	Idem pour « OTHER »	(2**2) 4
S_IWOTH		(2**1) 2
S_IXOTH		(2**0) 1
S_IRWXO		7

# open

## ■ open

- conversion\_id
  - CCSID ou code page des données à lire/écrire depuis/vers le fichier
  - Nécessite O\_CCSID ou O\_CODEPAGE dans oflag
- text\_file\_creation\_conversion\_id
  - CCSID ou code page des données manipulés dans le RPG
  - Nécessite O\_TEXT\_CREAT (éventuellement O\_TEXTDATA et/ou O\_CREAT, O\_CSSID, O\_CODEPAGE)
  - Par défaut (si non indiqué) le CCSID du job en cours
- Remarques
  - IBM recommande l'utilisation des CCSID plutôt que des codes pages
  - C'est l'API iconv qui est utilisée pour les conversions

# CCSID

- Voyons le fichier créé précédemment

```
Attributs d'affichage
Objet . . . . . : /home/bonnet/session7.log
Type . . . . . : STMF
Propriétaire . . . . . : BONNET
Système de l'objet . . . . . : Local
Pool de mémoire secondaire . . . . . : 1
  Objet en dépassement . . . . . : Non
ID de jeu de caractères codés . . . . . : 297
Fichier cache . . . . . : Non
Fichier système PC . . . . . : Non
Lecture seule . . . . . : Non
```

- Le job utilisé est en 297 (FR EBCDIC)
- Il peut être utile sur l'IFS de lire ou écrire des fichiers dans différents CCSID
  - Usage web, fichier fournit par d'autres applications ...

# CCSID

## ■ Créer un fichier en UTF-8

- Depuis des données au format du job (297 ici)

```
// création ou écrasement du fichier, avec encodage fichier UTF-8 :
fd = open( '/home/bonnet/session7-UTF8.log'
           :O_WRONLY + O_CREAT + O_TRUNC + O_TEXTDATA + O_CCSID +
           O_TEXT_CREAT
           :S_IRWXU + S_IRWXG + S_IRWXO
           :1208 ) ;
```

- Cette fois le fichier est bien UTF-8

```
Attributs d'affichage
Objet . . . . . : /home/bonnet/session7-UTF8.log
Type . . . . . : STMF
Propriétaire . . . . . : BONNET
Système de l'objet . . . . . : Local
Pool de mémoire secondaire . . . . . : 1
  Objet en dépassement . . . . . : Non
ID de jeu de caractères codés . . . . . : 1208
Fichier caché . . . . . : Non
Fichier système PC . . . . . : Non
Lecture seule . . . . . : Non
```

- Et les droits

Utilisat	Droits sur données	---Droits sur objet---			
		Exist	Gest	Modif	Réf
*PUBLIC	*RWX	X	X	X	X
BONNET	*RWX	X	X	X	X

# Lire des données

## ■ Cinématique

- Ouverture du fichier
  - API open
- Lecture des données
  - API read
- Fermeture du fichier
  - API close

```

// int open(const char *path, int oflag, . . .);
d open          pr          10i 0 extproc( *CWIDEN : 'open' )
d path          *          value options(*string)
d oflag         10i 0 value
d mode         10i 0 value options(*nopass)
d conversion_id...
d              10u 0 value options(*nopass)
d text_file_creation_conversion_id...
d              10u 0 value options(*nopass)

// ssize_t read(int file_descriptor, void *buf, size_t nbytes);
d read         PR          10i 0 extproc(*CWIDEN : 'read')
d fildes       10i 0 value
d buf          *          value
d nbytes       10u 0 value

// int close(int fildes);
d close        pr          10i 0 extproc( *CWIDEN : 'close' )
d fildes       10i 0 value

```

- Éléments supplémentaires pour open

```

* Open for reading only
d O_RDONLY    C          1
* text data flag
D O_TEXTDATA  C          16777216
* CCSID
D O_CCSD      C          32

```



# Lire des données

```
// descripteur de fichier
d fd          s          10i 0 inz
// données à lire
d data        s          128a  inz
// nb d'octets lus
d nbread      s          10u 0 inz
// dsply
d Msg         s          52a  inz

/free

// ouverture du fichier en lecture, fichier en UTF-8 :
fd = open( '/home/bonnet/session7-UTF8.log'
           :O_RDONLY + O_TEXTDATA + O_CCSD
           :0      // sans importance en lecture
           :0 ) ; // convertir vers le CCSID du job
if ( fd < 0 ) ;
  dsply 'Ouverture ".../session7-UTF8.log" impossible' ;
  return ;
endif ;

// lecture des données :
nbread = read( fd : %addr( data ) : %size( data ) ) ;
if ( nbread < 0 ) ;
  dsply 'Erreur lecture ".../session7-UTF8.log"' ;
  return ;
endif ;
Msg = %subst( data : 1 : nbread ) ;
dsply msg ;

// fermeture
if ( close( fd ) < 0 ) ;
  dsply 'Erreur à la fermeture du fichier' ;
  return ;
endif ;
```

## Open

- O\_TEXTDATA + O\_CCSD
  - Permet la conversion automatique du CCSID du fichier (UTF-8) vers le CCSID du job

## read, nécessite :

- Le file descriptor (cf open)
- Une variable où stocker les données lues (pointeur)
- La longueur max. des données à recevoir

## Close, nécessite :

- Le file descriptor (cf open)

# Lire des données

- Appel

```
call s7_5
DSPLY  Bienvenue dans session RPG & IFS !
```

- Si l'on ne tient pas compte des CCSID

```
call s7_6
DSPLY  âÑÁ>ÎÁ>ÍÁÀ/>ËËÁËËÑ?>ê&ãñäë
```

- Résultat produit par

```
// ouverture du fichier en lecture, fichier en UTF-8 :
fd = open( '/home/bonnet/session7-UTF8.log'
           :O_RDONLY ) ;
```

## Ligne à ligne

- Bien souvent, on ne cherche pas à lire ou écrire x caractères
  - Mais une ligne
- Une ligne est définie par la présence de caractères spéciaux
  - Non affichables
  - Présents dans les données
  - Différentes conventions
    - UNIX : LF (Line Feed)
    - Apple : CR (Carriage Return)
    - Windows : CR/LF (2 caractères)
- En EBCDIC
  - CR : x'0D', LF : x'25'
  - Définis par les constantes `STREAM_LINE_FEED` et `STREAM_CARRIAGE_RETURN`

## Écriture ligne à ligne

- Il suffit d'ajouter les caractères CR/LF à la suite des données à écrire

```
/include QSYSINC/QRPGLESRC,IFS

// Plus pratique
d CRLF          c          const( x'0D25' )

// création ou écrasement du fichier :
fd = open( '/home/bonnet/agenda.csv'
           :O_WRONLY + O_CREAT + O_TRUNC + O_TEXTDATA + O_CCSID +
           O_TEXT_CREAT
           :S_IRWXU + S_IRWXG + S_IRWXO
           :1252 ) ; // PC WINDOWS 1252

// écriture de données :
data = 'Session;Titre;Intervenant' + CRLF ;
callp write( fd :%addr( data : *data ) : %len( data ) ) ;
data = 'S1;Panorama des solutions Rational pour IBM i;Philippe ' +
      'Bourgeois' + CRLF ;
callp write( fd :%addr( data : *data ) : %len( data ) ) ;
data = 'S2;DB2/SQL - Les procédures cataloguées;Pierre-Louis Berthoin' +
      CRLF ;
callp write( fd :%addr( data : *data ) : %len( data ) ) ;
data = 'S3;PHP sous IBM i : des exemples concrets;Dominique Gayte' +
      CRLF ;
callp write( fd :%addr( data : *data ) : %len( data ) ) ;
```

# Ecriture ligne à ligne

- Le fichier généré
  - CRLF a été converti
    - X'0D0A'
    - CR/LF ASCII

```

agenda.csv
1 Session;Titre;IntervenantCRLF
2 S1;Panorama des solutions Rational pour IBM i;Philippe BourgeoisCRLF
3 S2;DB2/SQL - Les procédures cataloguées;Pierre-Louis BerthoinCRLF
4 S3;PHP sous IBM i : des exemples concrets;Dominique GayteCRLF
5
  
```

agenda.csv - Microsoft Excel

Fichier Accueil Insertion Mise en page Formules Données Révision Afficha

G10 fx

	A	B	C	D
1	Session	Titre	Intervenant	
2	S1	Panorama des solutions Rational pour IBM i	Philippe Bourgeois	
3	S2	DB2/SQL - Les procédures cataloguées	Pierre-Louis Berthoin	
4	S3	PHP sous IBM i : des exemples concrets	Dominique Gayte	
5				
6				

agenda

Prêt

```

Browse : /home/bonnet/agenda.csv
Record : 1 of 4 by 18 Column : 1 64 by 131
Control :

.....1.....2.....3.....4.....5.....6.....7.....8.....
*****Beginning of data*****
Session;Titre;Intervenant
S1;Panorama des solutions Rational pour IBM i;Philippe Bourgeois
S2;DB2/SQL - Les procédures cataloguées;Pierre-Louis Berthoin
S3;PHP sous IBM i : des exemples concrets;Dominique Gayte
*****End of Data*****
  
```

# Lecture ligne à ligne

- Il n'existe pas d'API qui permette de lire une ligne
  - Il faut lire caractère par caractère jusqu'à trouver CR+LF

```

#include QSYSINC/QRPGLESRC,IFS

// Plus pratique
d CRLF          c          const( x'0D25' )

// Procédure interne de lecture d'une ligne définie par CRLF
d readline      pr          n
d line          1024a      varying

```

```

// descripteur de fichier
d fd            s          10i 0 inz
// données à lire
d data          s          1024a  inz varying
// Pour display
d msg           s          52a  inz

```

```

// Ouverture pour lecture :
fd = open( '/home/bonnet/agenda.csv'
           :O_RDONLY + O_TEXTDATA + O_CCSID
           :0      // sans importance en lecture
           :0 ) ; // convertir vers le CCSID du job
if ( fd < 0 ) ;
  dsply 'Ouverture "/home/bonnet/agenda.csv" impossible' ;
  return ;
endif ;

// lecture des données
dow ( readline( data ) ) ;
  // afficher la ligne
  msg = data ;
  dsply msg ;
enddo ;

// fermeture
callp close( fd ) ;

```

# Lecture ligne à ligne

```

preadline      b
dreadline     pi          n
d line        1024a      varying
// fin de fichier détectée
d eof         s          n   inz( *off )
// fin de ligne détectée
d eol         s          n   inz( *off )
// caractère à lire
d car         s          1a   inz
/free

```

```

// RAZ
%len( line ) = 0 ;

// Lire le caractère suivant tant que CRLF n'a pas été détecté
dow not eof and not eol ;

// lecture suivante
eof = ( read( fd : %addr( car ) : 1 ) = 0 ) ;
if ( eof ) ;
    iter ;
endif ;

// ajout du caractère lu :
line += car ;

// CRLF détecté ?
if ( %len( line ) >= 2 ) ;
    eol = %subst( line : %len( line ) - 1 : 2 ) = CRLF ;
endif ;

enddo ;

// enlever le CRLF de la chain lue
if ( eol ) ;
    %len( line ) -= 2 ;
endif ;

// fin de la lecture ?
return not eof ;

```

```

call      s7_8
DSPLY    Session;Titre;Intervenant
DSPLY    S1;Panorama des solutions Rational pour IBM i;Philip
DSPLY    S2;DB2/SQL - Les procédures cataloguées;Pierre-Louis
DSPLY    S3;PHP sous IBM i : des exemples concrets;Dominique

```

## Lister le contenu d'un répertoire

- Un répertoire est un fichier contenant la liste d'autres fichiers (répertoires, liens)
- Pour les manipuler
  - Opendir
  - Readdir
  - Closedir

```
// DIR *opendir(const char *dirname);
d opendir          pr          *   extproc( *CWIDEN : 'opendir' )
d  dirname         *   value options( *string )

// struct dirent *readdir(DIR *dirp);
d readdir          pr          *   extproc( *CWIDEN : 'readdir' )
d  dirp            *   value

// int closedir(DIR *dirp);
d closedir         pr          10i 0 extproc( *CWIDEN : 'closedir' )
d  dirp            *   value
```

```
d dirent_t         ds          template qualified
d  d_reserved1    16a
d  d_fileno_gen_id...
d                                     10u 0
d  d_fileno        10i 0
d  d_reclen        10u 0
d  d_reserved3     10i 0
d  d_reserved4     6a
d  d_reserved5     2a
d  d_nlsinfo       likeds( qlg_nls_t )
d  d_namelen       10i 0
d  d_name          640a

// qlg_nls_t
d qlg_nls_t        ds          template qualified
d  ccsid           10i 0
d  country_id      2a
d  language_id     3a
d  nls_reserved    3a
```



# Lister le contenu d'un répertoire

```
// directory pointer
d dirp          s          *   inz
// directory entry
d dirent        ds          likeds( dirent_t )
d               ds          based( dirent_p )
// Pour display
d msg           s          52a   inz
```

```
// Ouverture du répertoire
dirp = opendir( '/home/bonnet' ) ;
if ( dirp = *null ) ;
    return ;
endif ;

// lecture de chaque poste
dirent_p = readdir( dirp ) ;
dow ( dirent_p <> *null ) ;

    // retrouver le libellé
    msg = %subst( dirent.d_name : 1 : dirent.d_namelen ) ;
    dsply msg ;

    // poste suivant
    dirent_p = readdir( dirp ) ;

enddo ;

// fermeture du répertoire
callp closedir( dirp ) ;
```

```
call    s7_10
DSPLY   .
DSPLY   ..
DSPLY   agenda.csv
DSPLY   ILEDocs
DSPLY   RSE
DSPLY   apache-tomcat-6.0.29
DSPLY   apache-tomcat-7.0.12
DSPLY   cmpsrc
DSPLY   demo
DSPLY   gaia
DSPLY   jdbc
DSPLY   projets
DSPLY   reu_tech
DSPLY   tpls
DSPLY   listemptyfile.sh
DSPLY   session7-UTF8.log
DSPLY   .atkininit
DSPLY   .ezinit
DSPLY   .icewm
DSPLY   session7.log
DSPLY   javaproject
DSPLY   tmp
DSPLY   gcc.4.1.1.tar
DSPLY   krb5ccname
DSPLY   .eclipse
DSPLY   preferences
DSPLY   wsd12rpg.sh
DSPLY   wsd12ws.sh
```

# Tester l'existence d'un fichier

- API access
  - Infocentre

## access()--Determine File Accessibility

### Syntax

```
#include <unistd.h>

int access(const char *path, int amode);
```

### *amode*

(Input) A bitwise representation of the access permissions to be checked.

The following symbols, which are defined in the `<unistd.h>` header file, can be used in *amode*:

*F\_OK*

Tests whether the file exists

*R\_OK*

Tests whether the file can be accessed for reading

*W\_OK*

Tests whether the file can be accessed for writing

*X\_OK*

Tests whether the file can be accessed for execution

- QSYSINC
  - Tout est défini
    - La procédure access
    - Les constantes F\_OK, R\_OK ...

# Tester l'existence d'un fichier

- API très simple d'utilisation

```
// -----  
// Prototype pour API IFS  
/include QSYSINC/QRPGLESRC,IFS  
  
// -----  
// variables  
  
// Pour display  
d msg          s          52a  inz  
  
// -----  
// Corps  
/free  
  
  if ( access( '/home/bonnet/agenda.csv' : F_OK ) = 0 ) ;  
    dsply 'Le fichier "/home/bonnet/agenda.csv" existe' ;  
  else ;  
    dsply 'Le fichier "/home/bonnet/agenda.csv" n''existe pas' ;  
  endif ;  
  
  return ;  
/end-free
```

```
call  s7_11  
DSPLY Le fichier "/home/bonnet/agenda.csv" existe
```

# Gestion des erreurs

## ■ Principe général

- Les API renvoient une valeur -1 en cas d'anomalie
- L'anomalie est identifiée par un n°
  - Comme un identifiant de message
- Des constantes permettent de nommer ces anomalies
  - Les rendre plus intelligibles
- Il est également possible de retrouver le texte de l'anomalie

## ■ Exemple pour open

### open()--Open File

#### Syntax

```
#include <fcntl.h>

int open(const char *path, int oflag, . . .);
```

### Return Value

*value* **open()** was successful. The value returned is the file descriptor.

**-1** **open()** was not successful. The *errno* global variable is set to indicate the error.

# Gestion des erreurs

## ■ Les API suivantes sont nécessaires

### – `__errno()`

- Renvoie un pointeur sur le n° d'erreur (entier sur 4 octets)
- Ce n° d'erreur correspond à un message CPEnnnn de QCPFMSG
- Définie par QSYSINC/QRPGLESRC,IFS (ERRNO)

```
D getErrnoPtr      PR          *  EXTPROC('__errno')
```

### – Ainsi que les constantes

```
* domain error in math function
D EDOM             C          3001
* range error in math function
D ERANGE           C          3002
* truncation on I/O operation
D ETRUNC           C          3003
* file has not been opened
D ENOTOPEN         C          3004
* file not opened for read
D ENOTREAD         C          3005
```

### – Pour plus de facilité

```
// Retrouver le n° de l'erreur
p errno           b
d                 pi           10i 0
// variables
d l_errno         s           10i 0 based( errno_ptr )
/free
  errno_ptr = getErrnoPtr() ;
  return l_errno ;
/end-free
p errno           e
```

# Gestion des erreurs

- strerror()
  - Renvoie un pointeur sur le libellé du message (chaîne à terminaison nulle)
  - Non définie dans QSYSINC ...

```
// char *strerror( int );  
d strerror      pr          *   extproc( *CWIDEN : 'strerror' )  
d errno        10i 0 value
```

# Gestion des erreurs

## ■ Exemple pour open

```
// Ouverture pour lecture :
fd = open( '/home/bonnet/agenda.XXX'
           :O_RDONLY + O_TEXTDATA + O_CCSD
           :0      // sans importance en lecture
           :0 ) ; // convertir vers le CCSID du job
if ( fd < 0 ) ;

// Afficher le n° de l'erreur :
errid = errno() ;
dsply ( 'N° de l'erreur : ' + %char( errid ) ) ;

// Afficher le libellée de l'erreur :
msg = %str( strerror( errno ) ) ;
dsply msg ;

// Messages à traiter
if ( errid = ENOENT ) ;
    dsply 'Le fichier n'existe pas !' ;
endif ;

return ;
endif ;
```

```
call    s7_9
DSPLY   N° de l'erreur : 3025
DSPLY   No such path or directory.
DSPLY   Le fichier n'existe pas !
```

## Et maintenant

- Vous pouvez lire et écrire des données dans l'IFS !
  - Au format texte, HTML, XML, csv ...
  - Avec des encodages compatibles dans le monde web et windows
  - Echanger des fichiers plats avec d'autres applications
  - Créer vos logs applicatives dans l'IFS, et partager les répertoires afin que les robots d'exploitation puissent les capter automatiquement
  - Modifier par programme la configuration de vos serveurs web APACHE dynamiquement
  - ...



## Mais ...

- N'hésitez pas à vous appuyer sur les travaux de Scott Klement
  - Prototypage complet des APIs
  - Fonctions utiles
  - Exemples
  - IFS eBook
  
- CGIDEV2 (Easy400)
  - Distribué (non maintenu) par IBM
  - A la base pour le développement de CGI
  - Possède de nombreux outils pour l'IFS
    - Constitution de fichiers IFS par moteur de « template »
    - Gestion des encodages
    - Fonctions utiles
    - ...

# Conseils

- **Faites vos propres procédures RPG**
  - Gèrent l'utilisation des fonctions C, les pointeurs, le mode ligne à ligne ...
  
- **Familiarisez-vous avec les commandes**
  - i5/OS : CD, WRKLNK, MKDIR, DEL, RMVLNK, ADDLNK, DSPF, EDTF ...
  - QShell / PASE : cd, pwd, ls, cat, touch, ln, rm, rmdir, mkdir ...
  
- **Une attention particulière aux CSSID, principalement si les fichiers**
  - Sont accédés au travers du NetServer
  - Sont à destination d'autres applications
    - PASE fonctionne en CCSID 37 par défaut ...
    - Serveurs web : préférez l'UTF-8
    - ...

# Informations utiles

## ■ Ressources

- <http://publib.boulder.ibm.com/infocenter/iserics/v7r1m0>
- [http://www.scottklement.com/rpg/ifs\\_ebook](http://www.scottklement.com/rpg/ifs_ebook)
- <http://www.easy400.net/easy400h/ifstool.htm>
- [http://www.volubis.fr/news/liens/AF4SRCT/IFSSRC/IFS\\_API.htm](http://www.volubis.fr/news/liens/AF4SRCT/IFSSRC/IFS_API.htm)
- <http://jplamontre.free.fr/jpltools.htm#IFStools>
- <http://www.easy400.net/cgidev2/start>

## ■ Forums

- <http://forum.xdocs400.com>
- <http://www.developpez.net>
- <http://forum.commonfr.org>
- <http://www.volubis.fr>
- <http://www.iprodeveloper.com>

# Nous contacter

## ■ Par mail

- [nbonnet@gaia.fr](mailto:nbonnet@gaia.fr)
- [contact@gaia.fr](mailto:contact@gaia.fr)

## ■ Nos sites

- [www.gaia.fr](http://www.gaia.fr)
- [www.know400.fr](http://www.know400.fr)
- [www.as400.fr](http://www.as400.fr)