GDPR : Protection et traçabilité renforcées des données sur IBM i

Session – Université IBM i 2018 – IBM Client Center Paris

Animateur:

Guy MARMORAT

Senior Director of Product Management

16 MAI 2018

AGENDA - GDPR : Protection et traçabilité renforcées des données sur IBM i

1. Identification des données
2. Identification des processus
3. Sécurité (niveau Objet)
4. Demo — Audit des changements (niveau Objet)
5. Audit des changements (niveau Donnée)
6. Audit des accès
7. Demo — Sécurité (niveau Donnée)
8. Demo — Contrôle d'accès (niveau Objet)
9. Fonctions avancées

1. Identification des données

Identification des données

DB2 IFS

Comment collecter les informations ?

Vous avez un ERP ?
Demandez à l'Editeur !

Vous utilisez des Outils ?
Cross-références, ALM, Modernisation de la Base de données ?

Vous faites du développement « maison »?
Demandez aux développeurs !

Vous avez un administrateur base de données ?
Intégrez-le/la dans cette phase d'identification !

Identification des données

Faites le vous-même !

Comment chercher directement des noms de champs ou de tables?

DB2

SELECT SYSTEM_TABLE_SCHEMA library, SYSTEM_TABLE_NAME table, SYSTEM_COLUMN_NAME field, length, data_type, column_text,column_heading FROM **qsys2.syscolumns**

WHERE length >= 20 and (lower(column_text) like '%name%' or lower(column_heading) like '%name%') and table_schema in ('LIB1','LIB2','LIB3')

ORDER BY library,table,field ;

select * from **qsys2.systables** where (lower(table_text) like '%clien%' or lower(table_text) like '%custom%') and table_schema in ('LIB1','LIB2','LIB3') ;

Comment chercher directement dans l'IFS?

IFS

cl: **RTVDIRINF** DIR('/') INFFILEPFX(RTVDIRINF) INFLIB(IJRNDEMOEX) OMIT('/QSYS.LIB' '/QIBM' '/QOPT' '/QOpenSys' '/QTCPTMM' '/QJRN400');

SELECT * from ijrndemoex.rtvdirinfd where lower(qezdirnam1) like '%client%' or lower(qezdirnam1) like '%customer%' ;

SELECT * from ijrndemoex.rtvdirinfo where lower(qezobjnam) like '%client%' or lower(qezobjnam) like '%customer%' ;

Identification des données

DB2

Comment chercher directement des noms de champs ou de tables avec un dictionnaire?

Et encore Mieux !

SELECT SYSTEM_TABLE_SCHEMA library, SYSTEM_TABLE_NAME table, SYSTEM_COLUMN_NAME field, length, data_type, column_text,column_heading FROM **qsys2.syscolumns**

Join **gm.gdprdic**

dic1 on lower(column_text concat column_heading) like '%' concat dic1.searchfor concat '%' and length >= dic1.len

where table_schema in ('LIB1','LIB2','LIB3') ORDER BY library,table,field ;

SEARCHFOR	LEN
name	20
address	20
zip	10
mail	20
city	20
country	20
tel	15
size of shoes	2
favorite chocolate	20
customer	8
client	6

Identification des données

IFS **Comment chercher directement dans l'IFS avec un dictionnaire?**

Et encore Mieux !

```
SELECT
QEZDIRNAM1,QEZOBJNAM,QEZOBJTYPE,QEZDTASIZE,QEZOWN,QEZCRTTIM,QEZACCTIM,QEZCHGTIMD
FROM ijrndemoex.rtvdirinfo o

left join ijrndemoex.rtvdirinfd d on o.QEZDIRIDX = d.QEZDIRIDX

join gm.gdprdic dic1 on lower(QEZDIRNAM1 concat QEZOBJNAM) like '%' concat dic1.searchfor concat
'%' ;
```

SEARCHFOR	LEN
name	20
address	20
zip	10
mail	20
city	20
country	20
tel	15
size of shoes	2
favorite chocolate	20
customer	8
client	6

2. Identification des processus

Identification des processus

Comment analyser des références croisées ?

De façon statique:

cl: DSPPGMREF PGM(ERPPGM/*ALL) OUTPUT(*OUTFILE) OBJTYPE(*ALL) OUTFILE(QTEMP/PGMREF);

select * from qtemp.pgmref where WHSNAM = 'GLFCLIEN‘;

Limitations/pièges: override, SQL dynamique, client-serveur, fichiers logiques

DB2

De façon dynamique: depuis le journal système, pour les fichiers sous audit *ALL (commandes CHGOBJAUD/CHGAUD)

```
select current_user current_user, job_name, program_library, program_name, remote_address, count(*) count from
table(qsys2.Display_Journal('QSYS','QAUDJRN', Journal_Codes => 'T')) as x
where object like '%GLFCLIEN%' and JOURNAL_ENTRY_TYPE in ('ZC', 'ZR')
group by current_user, job_name, program_library, program_name, remote_address
order by current_user, job_name, program_library, program_name, remote_address;
```

DB2

```
select current_user current_user, job_name, program_library, program_name, remote_address, count(*) count from
table(qsys2.Display_Journal('QSYS','QAUDJRN', Journal_Codes => 'T')) as x
where path_name like '/Customer_Info/%' and JOURNAL_ENTRY_TYPE in ('ZC', 'ZR')
group by current_user, job_name, program_library, program_name, remote_address
order by current_user, job_name, program_library, program_name, remote_address;
```

IFS

Faites le vous-même !

3. Sécurité (niveau Objet)

Sécurité (Niveau Objet)

Vérifier les autorisations assignées aux objets

DSPOBJD OBJ(ERPFILE/*ALL) OBJTYPE(*ALL) OUTPUT(*OUTFILE) OUTFILE(QTEMP/OBJ) OUTMBR(*FIRST *ADD)

Loop: DSPOBJAUT OBJ(ERPFILE/&ODOBNM) OBJTYPE(*FILE) OUTPUT(*OUTFILE) OUTFILE(QTEMP/AUT) OUTMBR(*FIRST *ADD)

SELECT oaname, oausr,oaobja,oaown,oaanam FROM aut

where oaown <> 'ERPOWNER' or oaanam <> 'ERPAUTL' or (oausr = '*PUBLIC' and oaobja <> '*EXCLUDE') or oausr not in ('ERPOWNER','*PUBLIC')

Faites le vous-même !

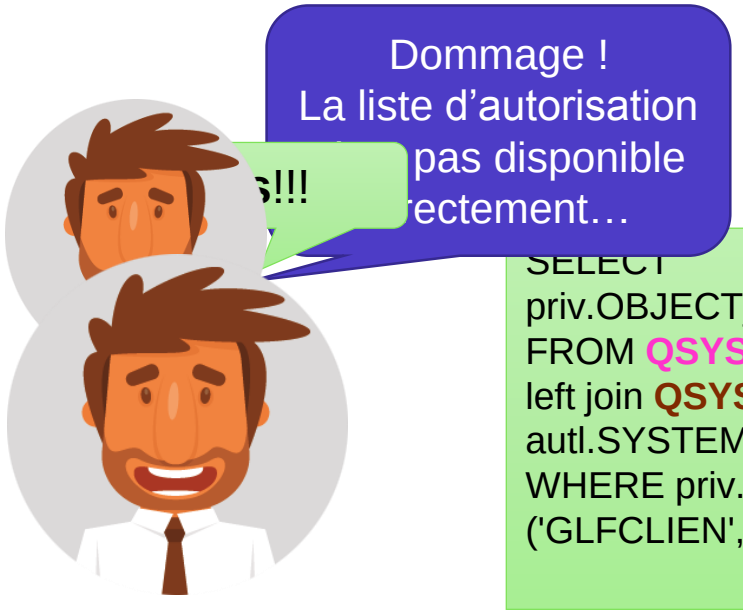
```
                          Edit Object Authority

Object . . . . . :    GLFCLIEN          Owner . . . . . :    ERPOWNER
  Library  . . . . :      ERPFILE         Primary group . . . :  *NONE
Object type  . . . :    *FILE             ASP device . . . . . :  *SYSBAS
Row or column access control  . . . . . . . . . . . . . . . . :  Active
Type changes to current authorities, press Enter.

  Object secured by authorization list  . . . . . . . . . . . .  ERPAUTL

                          Object
User          Group       Authority
*PUBLIC                   *EXCLUDE
ERPOWNER                  *ALL
```

Sécurité (Niveau Objet)

Faites le vous-même !

Vérifier les autorisations assignées aux objets (alternative)

api QUSLOBJ or SELECT * FROM TABLE(QSYS2.OBJECT_STATISTICS('ERPFILE', '*FILE')) as x
then, api QSYRTVUA "Retrieve Users Authorized to an Object"
SELECT oaname, oausr,oaobja,oaown,oaanam FROM aut
where oaown <> 'ERPOWNER' or oaanam <> 'ERPAUTL'
 or (oausr = '*PUBLIC' and oaobja <> '*EXCLUDE')
 or oausr not in ('ERPOWNER','*PUBLIC')



Depuis IBM 7.3 TR2 et 7.2 TR6:

SELECT * FROM **QSYS2.OBJECT_PRIVILEGES** WHERE SYSTEM_OBJECT_SCHEMA = 'ERPFILE' and OBJECT_NAME in ('GLFCLIEN','GLFCUENTA') and OBJECT_TYPE = '*FILE';

```
SELECT
priv.OBJECT_SCHEMA,priv.OBJECT_NAME,AUTHORIZATION_NAME,AUTHORIZATION_LIST,OBJECT_AUTHORITY
FROM QSYS2.OBJECT_PRIVILEGES priv
left join QSYS2.AUTHORIZATION_LIST_INFO autl on priv.SYSTEM_OBJECT_SCHEMA =
autl.SYSTEM_OBJECT_SCHEMA and priv.SYSTEM_OBJECT_NAME = autl.SYSTEM_OBJECT_NAME
WHERE priv.SYSTEM_OBJECT_SCHEMA = 'ERPFILE' and priv.SYSTEM_OBJECT_NAME in
('GLFCLIEN','GLFCUENTA') and priv.OBJECT_TYPE = '*FILE';
```

Sécurité (Niveau Objet)

Faites le vous-même !

Vérifier les programmes permettant d'obtenir une autorité forte

Par adoption de droits:

DSPPGMADP USRPRF(QSECOFR) OUTPUT(*OUTFILE) OUTFILE(QTEMP/PGMADP)

Par permutation de droits (swap):

DSPPGMREF PGM(ERPPGM/*ALL) OUTPUT(*OUTFILE) OBJTYPE(*ALL) OUTFILE(QTEMP/PGMREF)

SELECT * FROM pgmref where whfnam in ('QSYGETPH','QWTSETP','QSYRLSPH')

QAUDJRN – related entry types

AP Obtaining adopted authority
PA Program changed to adopt authority
PS Profile swap

Sécurité (Niveau Objet)

Vérifier les utilisateurs

Faites le vous-même !

Review *ALLOBJ users

```
SELECT AUTHORIZATION_NAME, STATUS, NO_PASSWORD_INDICATOR, PREVIOUS_SIGNON,
TEXT_DESCRIPTION
FROM QSYS2.USER_INFO
WHERE SPECIAL_AUTHORITIES LIKE '%*ALLOBJ%‘ OR AUTHORIZATION_NAME IN (SELECT USER_PROFILE_NAME
FROM QSYS2.GROUP_PROFILE_ENTRIES
WHERE GROUP_PROFILE_NAME IN (
SELECT AUTHORIZATION_NAME
FROM QSYS2.USER_INFO
WHERE SPECIAL_AUTHORITIES like '%*ALLOBJ%'))
ORDER BY AUTHORIZATION_NAME;
```

Checks inheritance from groups (PRTUSRPRF does not work)

Review default passwords

```
Select authorization_name  from user_info
where USER_DEFAULT_PASSWORD = 'YES'
```

ANZDFTPWD
- **Requires *SECADM and *ALLOBJ**
- **produces file QASECPWD that includes any user**
- **has an action option *DISABLE / *PWDEXP**

User_info

Only *USRPRF objects that the user has *OBJOPR and *READ authority to will be returned.

Review group profiles and associated users

```
SELECT CAST(GROUPNAME AS CHAR(10)) AS GROUP,
       CAST(USERNAME AS CHAR(10)) AS USER
  FROM QSYS2.GROUP_PROFILE_ENTRIES
```

Supplemental group profiles added to USER_INFO

https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20i%20Technology%20Updates/page/QSYS2.USER_INFO%20catalog

Sécurité (Niveau Objet)

Vérifier les utilisateurs

Review users without limited capabilities

Select * from qsys2.user_info where LIMIT_CAPABILITIES = '*YES'

PLEASE REMEMBER
PuTTY, RmtCmd, ODBC allows running commands even for limited users

Can be solved using Exit Programs

Faites le vous-même !

Review users with Password attempts

SELECT * FROM QSYS2.USER_INFO WHERE
SIGN_ON_ATTEMPTS_NOT_VALID > 0

Review users not used within 90 days

```
SELECT AUTHORIZATION_NAME, STATUS, LAST_USED_TIMESTAMP
  FROM QSYS2.USER_INFO
  WHERE LAST_USED_TIMESTAMP < CURRENT TIMESTAMP – 90 DAYS
    AND AUTHORIZATION_NAME <> 'QSECOFR'
    AND STATUS <> '*DISABLED'
    AND AUTHORIZATION_NAME NOT LIKE 'Q%'
  ORDER BY 3 DESC;
```

Review users with command auditing

Select * from QSYS2.user_info where USER_ACTION_AUDIT_LEVEL like '%*CMD %'

CHGUSRAUD USRPRF(XX) AUDLVL(*CMD)

Review objects *USRPRF that can be used by other users

```
SELECT * FROM QSYS2.OBJECT_PRIVILEGES
where OBJECT_TYPE = '*USRPRF' and
((AUTHORIZATION_NAME = '*PUBLIC' and OBJECT_AUTHORITY <> '*EXCLUDE')
or (AUTHORIZATION_NAME <> owner       and AUTHORIZATION_NAME <>
System_object_name and AUTHORIZATION_NAME <> '*PUBLIC' and owner <>
'QSYS' ))
```

Submitting commands under other users

4. Audit des changements (niveau Objet)

Audit des changements (niveau Objet) – Généralités journaux

Command & Keyword	Possible Values
CRTJRN, CHGJRN **Fixed length data . . . . FIXLENDTA**	Single Values *SAME *JOBUSRPGM Other Values *JOB *USR *PGM *PGMLIB *SYSSEQ *RMTADR *THD *LUW *XID
CRTJRN, CHGJRN **Manage receivers . . . . MNGRCV**	*SYSTEM *USER
CRTJRN, CHGJRN **Delete receivers . . . . DLTRCV**	*NO *YES

*JOBUSRPGM still the default value - not allowed for QAUDJRN
*RMTADR and *PGMLIB are useful;

Detachment based on the journal receiver threshold and at each IPL

HA solutions are able to keep a specific amount of receivers online based on the size, the number, the date, etc…
Your minimum retention period is at least 3 days so you can investigate on Monday on any situation that happened during the weekend

Audit des changements (niveau Objet) – Généralités journal système

How to find journal codes and journal entry types?

Security Reference Guide, Appendix “Layout of audit journal entries”
(only for Journal Code = T)

Model files in QSYS (example : QASYCPJ5 for entry type CP)
(only for Journal Code = T)

Fields you may encounter in many entries for Journal Code T

```
XXETYP      TYPE OF ENTRY                A      1
XXONAM      Object Name                  A     10
XXOLIB      Library name                 A     10
XXOTYP      Object type                  A      8
XXPNM       Path name                    A   5000
```

Website:

http://www.ibm.com/support/knowledgecenter/en/ssw_ibm_i_72/rzaru/rzarufinder.htm

Are ZC entry types still used for replication?

Very common to see ZC entries occupying a huge portion of the journal receivers.

ZC are triggered by changing the auditing value of an object to *CHANGE
CHGOBJAUD OBJ(ERPFILE/GLFCLIEN) OBJTYPE(*FILE) OBJAUD(*CHANGE)

Back before V5R2, there was no other way to detect a change in the file structure.

An entry ZC can correspond to different events for a file:

- Simply opening the file in update mode
- Alter table
- CHGPF
- RGZPFM
- ADDPFCST
- …..

Using the entries D in the database journal don't cause this pollution and generate dedicated entry types.

Entry Type	Description
CG	Change file
CT	Create database file
DC	Remove referential integrity constraint
DF	File was deleted
DH	File saved
DJ	Change journaled object attribute
DZ	File restored
EF	Journaling for a physical file ended (ENDJRNPF)
FN	File renamed (RNMOBJ)
GO	Change owner
GT	Grant authority
JF	Journaling for a physical file started (STRJRNPF (JRNPF))
TD	Remove trigger
ZB	Change object attribute

Audit des changements (niveau Objet) – Généralités journal système

Being more specific with what event have to be recorded

QAUDLVL/2 Security auditing level

*JOBDTA ==> *JOBBAS *JOBCHGUSR

*NETCMN ==> *NETBAS *NETCLU *NETFAIL *NETSCK

*SECURITY ==> *SECCFG *SECDIRSRV *SECIPC *SECNAS *SECRUN *SECSCKD *SECVFY *SECVLDL

How to record full commands?

2 ways:

Record all commands for a specific user:
CHGUSRAUD USRPRF(XXX) AUDLVL(*CMD)

Record specific commands for any user:
CHGOBJAUD OBJ(QSYS/UPDDTA) OBJTYPE(*CMD) OBJAUD(*ALL)

One CD entry is generated in QAUDJRN per command run
An additional entry may be recorded for the proxy command.

Another way is to register a program to the exit point QIBM_QCA_RTV_COMMAND with the qualified command as a parameter

What's new in 7.2?

New entries:
- ⇨ ***AX*** - Row and column access control
- ⇨ ***PF*** - PTF operations
- ⇨ ***PU*** - PTF object changes
- ⇨ ***X2*** - Query manager profile changes

Some existing entries now have previous value fields:
- ⇨ ***AD*** - Auditing changes
- ⇨ ***AU*** - Attribute changes
- ⇨ ***CA*** - Authority changes
- ⇨ ***CP*** - User profile changed, created, or restored
- ⇨ ***GR*** - Generic record
- ⇨ ***PA*** - Program changed to adopt authority
- ⇨ ***PG*** - Change of an object's primary group
- ⇨ ***RJ*** - Restoring job description with user profile specified

Command Auditing improved
- ⇨ ***CD*** - Command string audit
  new values for CDCLP
  also available in 7.1 through PTF SI44865

What's new in 7.3?

The CP (User Profile Changes) security audit journal entry contains fields for all the Create User Profile (CRTUSRPRF) command parameters except TEXT and AUT and all the Change User Profile (CHGUSRPRF) command parameters except TEXT.

Audit des changements (niveau Objet)

Mieux que DSPJRN

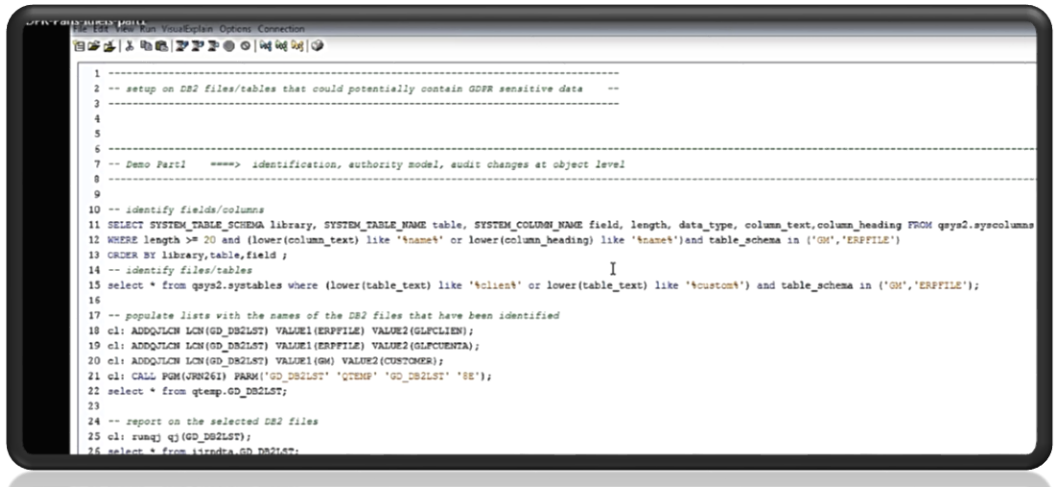
Querying the System Audit Journal

7.2 +

```
select JOURNAL_ENTRY_TYPE, current_user current_user, job_name, program_library,
program_name, remote_address, count(*) count from
table(qsys2.Display_Journal('QSYS','QAUDJRN', Journal_Codes => 'T')) as x
where object like '%GLFCLIEN%'
group by JOURNAL_ENTRY_TYPE, current_user, job_name, program_library, program_name,
remote_address
order by JOURNAL_ENTRY_TYPE, current_user, job_name, program_library, program_name,
remote_address;
```

Contactez-nous pour obtenir une **démo!**

Contact-Cilasoft@syncsort.com



```
1  ------------------------------------------------------------------------
2  -- setup on DB2 files/tables that could potentially contain GDPR sensitive data    --
3  ------------------------------------------------------------------------
4
5
6  ------------------------------------------------------------------------------------------------------------------------
7  -- Demo Part1    ====>  identification, authority model, audit changes at object level
8  ------------------------------------------------------------------------------------------------------------------------
9
10 -- identify fields/columns
11 SELECT SYSTEM_TABLE_SCHEMA library, SYSTEM_TABLE_NAME table, SYSTEM_COLUMN_NAME field, length, data_type, column_text,column_heading FROM qsys2.syscolumns
12 WHERE length >= 20 and (lower(column_text) like '%name%' or lower(column_heading) like '%name%')and table_schema in ('GM','ERPFILE')
13 ORDER BY library,table,field ;
14 -- identify files/tables
15 select * from qsys2.systables where (lower(table_text) like '%client%' or lower(table_text) like '%custom%') and table_schema in ('GM','ERPFILE');
16
17 -- populate lists with the names of the DB2 files that have been identified
18 cl: ADDQULCH LCH(GD_DB2LST) VALUE1(ERPFILE) VALUE2(GLFCLIEN);
19 cl: ADDQULCH LCH(GD_DB2LST) VALUE1(ERPFILE) VALUE2(GLFCUENTA);
20 cl: ADDQULCH LCH(GD_DB2LST) VALUE1(GM) VALUE2(CUSTOMER);
21 cl: CALL PGM(JRN261) PARM('GD_DB2LST' 'QTEMP' 'GD_DB2LST' '8E');
22 select * from qtemp.GD_DB2LST;
23
24 -- report on the selected DB2 files
25 cl: runqj qj(GD_DB2LST);
26 select * from izrndta.GD_DB2LST;
```

5. Audit des changements (niveau Donnée)

Audit des changements (niveau Donnée) – Généralités database journal

CRTJRN, CHGJRN
Minimize entry specific data . .
MINENTDTA

Single Values
*SAME *NONE

Other Values
*FILE *FLDBDY *DTAARA

***FILE cannot be used for auditing, everybody agrees.**
IBM promotes *FLDBDY but…look at these screen shots.

Only the changed values are displayed. To get values of additional fields, you have to create an SQL index and journalize it.
Not so easy…

Audit des changements (niveau Donnée) – Généralités database journal

Secure journals and receivers

Correct Authorities on journals & receivers

Protecting the file while leaving the journal data exposed

DISPLAY _JOURNAL() handles correctly RCAC rules. **Not the case for DSPJRN**
IBM i 7.3 SF99703 Level 3 and IBM i 7.2 SF99702 Level 14

https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20i%20Technology%20Updates/page/DISPLAY_JOURNAL%20%28easier%20searches%20of%20Audit%20Journal%29

STRJRN,STRJRNPF, STRJRNLIB
Record images . . . . . IMAGES

*AFTER *BOTH

*BOTH for important files

STRJRN,STRJRNPF, STRJRNLIB
Journal entries to be omitted . OMTJRNE

*NONE *OPNCLO

*NONE for files that require open auditing

CHGJRNOBJ OBJ((ERPFILE/GLFCLIEN *FILE))
ATR(*IMAGES) IMAGES(*BOTH)

Changes journaling attributes without the need to end and restart journaling for the object.
Introduced in V5R3 !

Audit des changements (niveau Donnée)

Mieux que DSPJRN

7.2 +

Querying the Database Journal

```
Select entry_timestamp, JOURNAL_ENTRY_TYPE, current_user current_user, job_name, program_library,
program_name, remote_address, cast(entry_data as char(200)) Data
from table(Display_Journal('IJRNDTA','ERPJRN', starting_receiver_name => '*CURCHAIN')) as x
where object like  '% GLFCLIEN %' and journal_code = 'R';
```

6. Audit des accès

Audit des accès

- **Tracer au niveau objet – Qui a ouvert ce fichier ?**
- **Tracer au niveau enregistrement – Qui a lu cet enregistrement ?**

Au niveau Objet

- System audit journal - Auditing value *ALL generates ZC & ZR entries
- Database journal - Parameter OMTJRNE(*NONE) generates OP entries
- (exit point) QIBM_QDB_OPEN intercepts in real time the openings of files under audit

Au niveau enregistrement

- Application (ex: send “user entries” to a journal for specific reads) ➔ incomplete
- Field procedures (7.1) ➔ gives the value of the field, not the entire record
- Read triggers ➔ it works, with limitations (not compatible with RCAC)

L'impact sur la performance est une préoccupation majeure

Options alternatives : Tokenisation, Encryption et depuis IBM i 7.2: RCAC

Audit des accès

DB2: CHGOBJAUD OBJ(erpfile/glfclien) OBJTYPE(*file) OBJAUD(*ALL)
IFS: CHGAUD OBJ('/customer_info/*') OBJAUD(*ALL) SUBTREE(*ALL)

Faites le vous-même !

```
select entry_timestamp, JOURNAL_ENTRY_TYPE,
current_user current_user, job_name, program_library,
program_name, remote_address from
table(qsys2.Display_Journal('QSYS','QAUDJRN',
Journal_Codes => 'T')) as x
where object like '%GLFCLIEN%' and
journal_entry_type in ('ZC','ZR');
```

Le Journal Système

Numeric codes for access types

This table lists the access codes used for object auditing journal entries in files QASYYCJE/J4/J5, QASYYRJE/J4/J5, QASYZCJE/J4/J5, and QASYZRJE/J4/J5.

Table 240. Numeric codes for access types

Code	Access type	Code	Access type	Code	Access type
1	Add	26	Load	51	Send
2	Activate Program	27	List	52	Start
3	Analyze	28	Move	53	Transfer
4	Apply	29	Merge	54	Trace
5	Call or TFRCTL	30	Open	55	Verify
6	Configure	31	Print	56	Vary
7	Change	32	Query	57	Work
8	Check	33	Reclaim	58	Read/Change DLO Attribute
9	Close	34	Receive	59	Read/Change DLO Security
10	Clear	35	Read	60	Read/Change DLO Content
11	Compare	36	Reorganize	61	Read/Change DLO all parts
12	Cancel	37	Release	62	Add Constraint
13	Copy	38	Remove	63	Change Constraint
14	Create	39	Rename	64	Remove Constraint
15	Convert	40	Replace	65	Start Procedure

Audit des accès

Le Journal Base de Données

Faites le vous-même !

DB2: STRJRNPF FILE(ERPFILE/GLFCLIEN) JRN(IJRNDTA/ERPJRN) IMAGES(*BOTH) OMTJRNE(*NONE)
CHGJRNOBJ OBJ((ERPFILE/GLFCLIEN *FILE)) ATR(*OMTJRNE) OMTJRNE(*NONE)

```
select entry_timestamp, JOURNAL_ENTRY_TYPE, current_user
current_user, job_name, program_library, program_name,
remote_address from
table(qsys2.Display_Journal('IJRNDTA','ERPJRN', Journal_Codes
=> 'F')) as x
where object like '%GLFCLIEN%' and journal_entry_type = 'OP';
```

7. Sécurité (niveau Donnée)

Sécurité (niveau Donnée)

RCAC

- Fully data-centric, not dependent upon specific interfaces, not only for SQL
- Works for any row operation (read, update, delete, insert). These SQL statements transparently contain the condition exactly like a WHERE clause.
- Relatively easy to implement
- “Silent” mechanism (no messages indicating rows are not permitted)
- Once activated, access to the rows is denied by default. Default permission with condition 0=1
- Operations on permissions and ALTER TABLE require an exclusive lock on the table.
- RCAC is processed after the object level security. But RCAC prevents a user with authority on the table (even *ALLOBJ) to see a specific subset of data.
- RCAC can impact performance, it has to be tested.

Limitations:
- Apply only to externally described files
- Read triggers not supported
- Legacy QRY/400 queries may have a different behavior

How to do an inventory:

select * from syscontrols ➔ dedicated view for RCAC
select * from syscontrolsdep ➔ dedicated view for RCAC dependencies like functions

Sécurité (niveau Donnée)

Official IBM presentation

IBM

IBM Knowledge Center

IBM i > IBM i 7.2 > Security > Security reference > Designing security > Separation of duties

http://www.ibm.com/support/knowledgecenter/ssw_ibm_i_72/rzarl/rzarlseparationofduties.htm

Separation of duties helps businesses comply with government regulations and simplifies the management of authorities. It provides the ability for administrative functions to be divided across individuals without overlapping responsibilities, so that one user does not possess unlimited authority, such as with *ALLOBJ authority. The function, QIBM_DB_SECADM, provides a user with the ability **to grant authority**, revoke authority, change ownership, or change primary group, but **without giving access to the object** or, in the case of a database table, to the data that is in the table or allowing other operations on the table.

QIBM_DB_SECADM function usage can be given only by a user with *SECADM special authority and can be given to a user or a group.

QIBM_DB_SECADM is also responsible for **administering Row and Column Access Control**.

Sécurité (niveau Donnée)

It is very important to audit/protect the changes to this function ID and usage

How to verify?

WRKFCNUSG

select * from function_usage
where function_id = 'QIBM_DB_SECADM'

⇨ list the registered users

select * from function_info
where function_id = 'QIBM_DB_SECADM'

⇨ check the Default Authority and *ALLOBJ special authority

How to audit?

Changes to Function Usage ➔ QAUDJRN/GR Action = ZC
Failures in Function Usage ➔ QAUDJRN/GR Field1 = *USAGEFAILURE

Changes to RCAC Functions ➔ QAUDJRN/AX
Changes to RCAC Functions ➔ Exit points & Database Monitor

Sécurité (niveau Donnée)

Faites le vous-même !

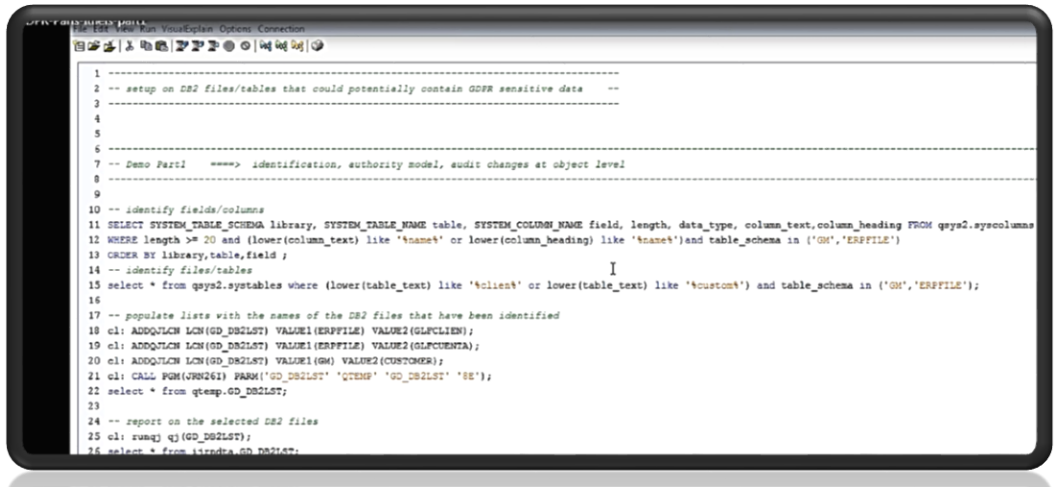
RCAC - Permissions

```
create permission erpfile.compidnot003 on erpfile.glfclien for rows where clicomp <> 003 enforced for all access enable ;
create permission erpfile.user_gm on erpfile.glfclien for rows where current_user = 'GM' enforced for all access enable ;
alter table erpfile.glfclien activate row access control ;
```

RCAC - Masks

```
create mask erpfile.mask_fax on erpfile.glfclien for column clifax2
return case
when current_user = 'GM' then clifax2
else '(Masked)'
end enable     ;
alter table erpfile.glfclien activate column access control ;
```


Contactez-nous pour obtenir une **démo!**
Contact-Cilasoft@syncsort.com



```
1  ------------------------------------------------------------------------
2  -- setup on DB2 files/tables that could potentially contain GDPR sensitive data    --
3  ------------------------------------------------------------------------
4
5
6  ------------------------------------------------------------------------------------------------------------------------------------------
7  -- Demo Part1    ====>  identification, authority model, audit changes at object level
8  ------------------------------------------------------------------------------------------------------------------------------------------
9
10 -- identify fields/columns
11 SELECT SYSTEM_TABLE_SCHEMA library, SYSTEM_TABLE_NAME table, SYSTEM_COLUMN_NAME field, length, data_type, column_text,column_heading FROM qsys2.syscolumns
12 WHERE length >= 20 and (lower(column_text) like '%name%' or lower(column_heading) like '%name%')and table_schema in ('GM','ERPFILE')
13 ORDER BY library,table,field ;
14 -- identify files/tables
15 select * from qsys2.systables where (lower(table_text) like '%clien%' or lower(table_text) like '%custom%') and table_schema in ('GM','ERPFILE');
16
17 -- populate lists with the names of the DB2 files that have been identified
18 cl: ADDQULCH LCH(GD_DB2LST) VALUE1(ERPFILE) VALUE2(GLFCLIEN);
19 cl: ADDQULCH LCH(GD_DB2LST) VALUE1(ERPFILE) VALUE2(GLFCUENTA);
20 cl: ADDQULCH LCH(GD_DB2LST) VALUE1(GM) VALUE2(CUSTOMER);
21 cl: CALL PGM(JRN261) PARM('GD_DB2LST' 'QTEMP' 'GD_DB2LST' '8E');
22 select * from qtemp.GD_DB2LST;
23
24 -- report on the selected DB2 files
25 cl: runqj qj(GD_DB2LST);
```

8. Contrôle d’Accès (niveau Objet)

Contrôle d'Accès (niveau Objet)

Pourquoi devons-nous renforcer la sécurité des objets?
(et ne pas la remplacer)

⇨ Standard object level security model:

A user who has *USE authority on a critical file can download it **using any method or protocol**

A user who has *CHANGE authority can change records in a critical file **using any method or protocol**

Adopted authority model:

- We have to trust the programs
- Does not work with the IFS

A noter :

- *ALLOBJ profiles are not controlled
- A limited user can still run commands in remote mode
- No visibility for non-5250 access, no standard log
- There is a need for contextual security

Idéal : Standard object level security + Contrôle d'Accès

Contrôle d'Accès (niveau Objet)

Parameter fields for exit point QIBM_QZDA_SQL2 format ZDAQ0200

Version 7.3

The following table shows parameter fields and their descriptions for the IBM® i database exit program called at exit point QIBM_QZDA_SQL2 with the ZDAQ0200 format.

Table 1. Exit point QIBM_QZDA_SQL2 format ZDAQ0200

Offset		Type	Field	Description
Dec	Hex			
0	0	CHAR(10)	User profile name	The name of the user profile that is calling the server.
10	A	CHAR(10)	Server identifier	The value is *SQLSRV for this exit point.
20	14	CHAR(8)	Format name	The user exit format name being used. For QIBM_QZDA_SQL1, the format name is ZDAQ0100.
28	1C	BINARY(4)	Requested function	The function being performed. This field contains one of the following: ▪ X'1800' - Prepare
32	20	CHAR(18)	Statement name	Name of the statement used for the prepare or execute functions.
50	32	CHAR(18)	Cursor name	Name of the cursor used for the open function.
68	44	CHAR(2)	Prepare option	Option used for the prepare function.
70	46	CHAR(2)	Open attributes	Option used for the open function.
72	48	CHAR(10)	Extended dynamic package name	Name of the extended dynamic package.
82	52	CHAR(10)	Package library name	Name of the library for extended dyanmic SQL package.
92	5C	BINARY(2)	DRDA® indicator	▪ 0 - Connected to local RDB ▪ 1 - Connected to remote RDB
94	5E	CHAR(1)	Commitment control level	▪ 'A' - Commit *ALL ▪ 'C' - Commit *CHANGE ▪ 'N' - Commit *NONE ▪ 'S' - Commit *CS (cursor stability)
95	5F	CHAR(10)	Default SQL collection	Name of the default SQL schema used by the IBM i Database Server. If the actual default SQL schema name is greater than 10 bytes, the following special value will be passed, indicating that the default SQL schema name should be obtained from the 'Extended default SQL Schema' field: ▪ *EXTDSCHMA Note: The Extended Default SQL Schema field will always be set, even if length is less than 10. Users can always refer to that field to get the Default SQL Schema name.
105	69	CHAR(1)	Naming Mode	▪ '0' - SQL naming ▪ '1' - System naming
106	6A	CHAR(2)	Reserved	Reserved for future parameters.
108	6C	BINARY(4)	Offset to the extended cursor name	The offset in this structure to the extended cursor name
112	70	BINARY(4)	Length of the extended cursor name	Length, in bytes, of the extended cursor name
116	74	BINARY(4)	Offset to the Extended Default SQL Schema	The offset in this structure to the Extended Default SQL Schema.
120	78	BINARY(4)	Length of the Extended default SQL Schema	Length, in bytes, of the Extended Default SQL Schema.

Contrôle d'Accès (niveau Objet)

Traditional exit points

- They are connected to Host and TCP/IP servers
- They cannot be unplugged for active jobs, with the exception of TELNET
- They generally allow just one program per point
- They are unaware of port numbers
- They must reside in *SYSBAS
- They are different from each other
- Things to consider: IP Address, CCSID, authorities, activation group
- QIBM_QZDA_SQL2 is the most difficult one (potential impact on performance)
- Limitations : read carefully the documentation

Command exit points

- One entry per command & timing (before or after options)

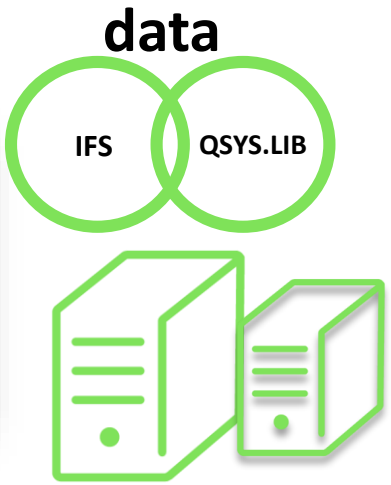
Other exit points

- Open database file
- Sockets

http://www.ibm.com/support/knowledgecenter/ssw_ibm_i_73/rzajr/rzajrmst35.htm

Access methods and exit points

TCP/IP Application Servers (STRTCPSVR)	
*FTP	QIBM_QTMF_*
*REXEC	QIBM_QTMX_*
*TELNET	QIBM_QTG_DEVINIT
*NETSVR	QIBM_QPWFS_FILE_SERV



Host Servers (STRHOSTSVR)	
*DATABASE	QIBM_QZDA_*
*FILE	QIBM_QPWFS_FILE_SERV
*RMTCMD	QIBM_QZRC_RMT

SNA	
DDM	CHGNETA DDMACC ()
Client Request	QIBM_QTF_TRANSFER

Access methods and exit points

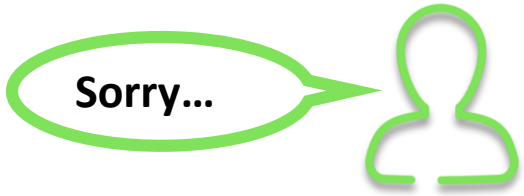
TCP/IP Application Servers (STRTCPSVR)		
*FTP	QIBM_QTMF_*	
*REXEC	QIBM_QTMX_*	
*TELNET	QIBM_QTG_DEVINIT	
*NETSVR	QIBM_QPWFS_FILE_SERV	
*SSHD	QP0ZSPW*	OpenSSH (sftp, scp, ..)
*EDRSQL		JDEwards, SAP

OTHERS		
CLI	QSQSRVR	PHP, XML Service, ...
QSQPRCED		XDA, XDN, ...
Sockets		Socket programs
Open Source		Node.js, Python, Ruby GCC, GIT, Orion, Perl...

Host Servers (STRHOSTSVR)	
*DATABASE	QIBM_QZDA_*
*FILE	QIBM_QPWFS_FILE_SERV
*RMTCMD	QIBM_QZRC_RMT

SNA		
DDM	CHGNETA DDMACC ()	
Client Request	QIBM_QTF_TRANSFER	
DRDA	QRWTSRVR	DB2 Connect

3rd Party products	
SEQUEL®	business intelligence
ShowCase®	business intelligence
Easycom®	Middleware - DB2 access



Access methods and exit points

TCP/IP Application Servers (STRTCPSVR)		
*FTP	QIBM_QTMF_*	
*REXEC	QIBM_QTMX_*	
*TELNET	QIBM_QTG_DEVINIT	
*NETSVR	QIBM_QPWFS_FILE_SERV	
*SSHD	QP0ZSPW*	OpenSSH (sftp, scp, ...)
*EDRSQL		JDEwards, SAP

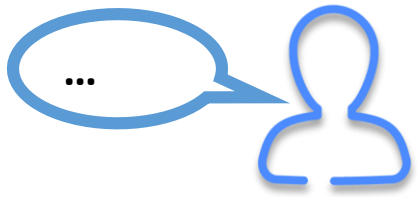
OTHERS		
CLI	QSQSRVR	PHP, XML Service, ...
QSQPRCED		XDA, XDN, ...
Sockets		Socket programs
Open Source		Node.js, Python, Ruby GCC, GIT, Orion, Perl...

5250	
Cmds & SQL	STRSQL, RUNSQL, RUNSQLSTM, STRQMQRY
Pgms & SQL	RPG, COBOL, CLI, QSQPRCED
PASE & QSH	QP2TERM, QSH db2
Cmds & QRY	RUNQRY, WRKQRY, QQQQry
Cmds & copy	CPY, CPYTOIMPF, SAVOBJ, CRTDUPOBJ
Apps	Bugs, Back doors

Host Servers (STRHOSTSVR)	
*DATABASE	QIBM_QZDA_*
*FILE	QIBM_QPWFS_FILE_SERV
*RMTCMD	QIBM_QZRC_RMT

SNA		
DDM	CHGNETA DDMACC ()	
Client Request	QIBM_QTF_TRANSFER	
DRDA	QRWTSRVR	DB2 Connect

3rd Party products	
SEQUEL®	business intelligence
ShowCase®	business intelligence
Easycom®	Middleware - DB2 access



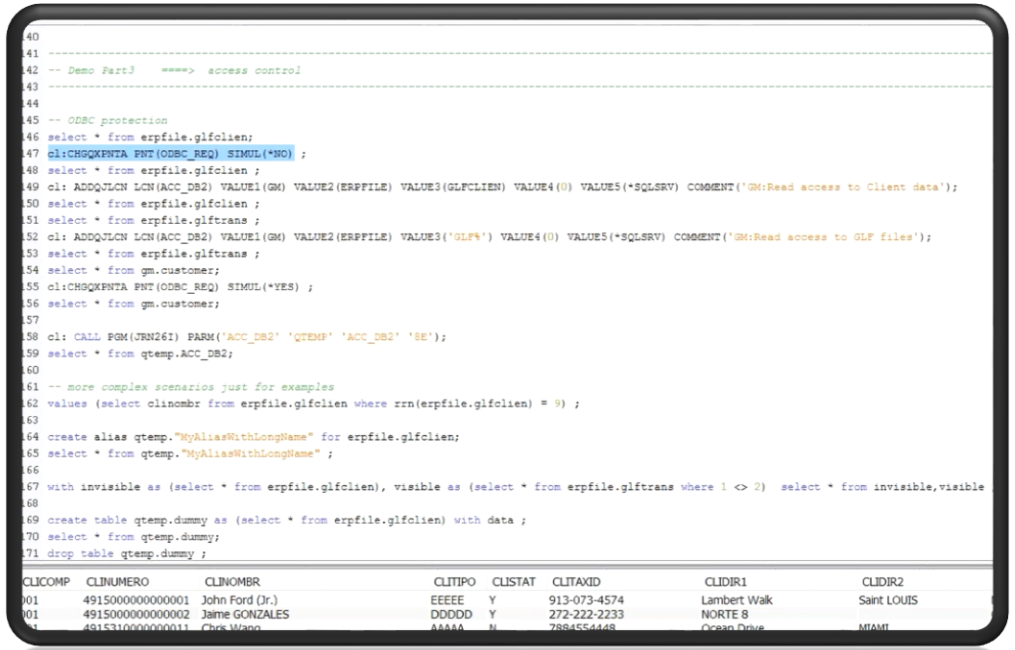
Access methods and exit points

Real life situation & thoughts to share :

- Gap between the growing number of ways to access data and the traditional exit points
- Gap between the typical IBM i administrators and the young IT people
- IBM promotes open source, which introduces new ways to access data
- SQL is growing in term of it's utilisation, power and complexity
- Exit programs add overhead and risk to production environments
- Database Monitor cannot block access and can also add overhead; it is not a tool designed for security
- There are no exit points for the Unix space, this is still based on Syslog files
- If you rely on RCAC, you still have to fully audit SQL and commands
  - alter table … deactivate row access control ; drop permission;
  - CHGFCNUSG FCNID(QIBM_DB_SECADM)

This way of protecting data is not efficient on today's systems with today's workloads.
We have to keep in mind that more than 70% of fraudulent acts are internal, which adds a huge challenge

Contactez-nous pour obtenir une **démo!**
Contact-Cilasoft@syncsort.com



```
40
41 ------------------------------------------------------------------------------------------------------------------------
42 -- Demo Part3    ====>  access control
43 ------------------------------------------------------------------------------------------------------------------------
44
45 -- ODBC protection
46 select * from erpfile.glfclien;
47 cl:CHGQXPNTA PNT(ODBC_REQ) SIMUL(*NO) ;
48 select * from erpfile.glfclien ;
49 cl: ADDQJLCN LCN(ACC_DB2) VALUE1(GM) VALUE2(ERPFILE) VALUE3(GLFCLIEN) VALUE4(0) VALUE5(*SQLSRV) COMMENT('GM:Read access to Client data');
50 select * from erpfile.glfclien ;
51 select * from erpfile.glftrans ;
52 cl: ADDQJLCN LCN(ACC_DB2) VALUE1(GM) VALUE2(ERPFILE) VALUE3('GLF%') VALUE4(0) VALUE5(*SQLSRV) COMMENT('GM:Read access to GLF files');
53 select * from erpfile.glftrans ;
54 select * from gm.customer;
55 cl:CHGQXPNTA PNT(ODBC_REQ) SIMUL(*YES) ;
56 select * from gm.customer;
57
58 cl: CALL PGM(JRN26I) PARM('ACC_DB2' 'QTEMP' 'ACC_DB2' '8E');
59 select * from qtemp.ACC_DB2;
60
61 -- more complex scenarios just for examples
62 values (select clinombr from erpfile.glfclien where rrn(erpfile.glfclien) = 9) ;
63
64 create alias qtemp."MyAliasWithLongName" for erpfile.glfclien;
65 select * from qtemp."MyAliasWithLongName" ;
66
67 with invisible as (select * from erpfile.glfclien), visible as (select * from erpfile.glftrans where 1 <> 2)  select * from invisible,visible
68
69 create table qtemp.dummy as (select * from erpfile.glfclien) with data ;
70 select * from qtemp.dummy;
71 drop table qtemp.dummy ;
```

CLICOMP	CLINUMERO	CLINOMBR	CLITIPO	CLISTAT	CLITAXID	CLIDIR1	CLIDIR2
001	4915000000000001	John Ford (Jr.)	EEEEE	Y	913-073-4574	Lambert Walk	Saint LOUIS
001	4915000000000002	Jaime GONZALES	DDDDD	Y	272-222-2233	NORTE 8	
	4915310000000011	Chris Wang	AAAAA	N	7884554448	Ocean Drive	MIAMI

9. Fonctions avancées

Fonctions avancées

How to audit changes in auditing mechanism?
Or in other words, how to garantee the integrity of the audit trail itself?

Goal / Concern	Answer
Changes to System values QAUD*	QAUDJRN - entry type SV
Changes to object auditing values	QAUDJRN - entry type AD
Changes to user auditing values	QAUDJRN - entry type AD
Changes to authority & ownership on journals and receivers	QAUDJRN - entry type CA & OW
Deleting receivers outside of the normal process	QAUDJRN - entry type DO with selection on program/program library
Changes to journal attributes (FIXLENDTA, MINENTDTA, DLTRCV)	QAUDJRN - entry type CD on command CHGJRN (CRTJRN/CHGJRN must be audited with *ALL before)
Stopping/starting journaling on DB2 files	DB Journal - code F & types EJ/JM
Changing journaling on DB2 files (IMAGES, OMTJRNE)	DB Journal - code D & types DJ
Stopping/starting journaling on IFS objects	DB Journal - code B & types ET/JT
Changing journaling on IFS objects (IMAGES, OMTJRNE)	DB Journal - code B & types JA
Commands RMVJRNCHG & APYJRNCHG	DB Journal - code D & type SR + code F & type RC …(these commands should also be audited)
Changes to the security applications	QAUDJRN & DB Journal

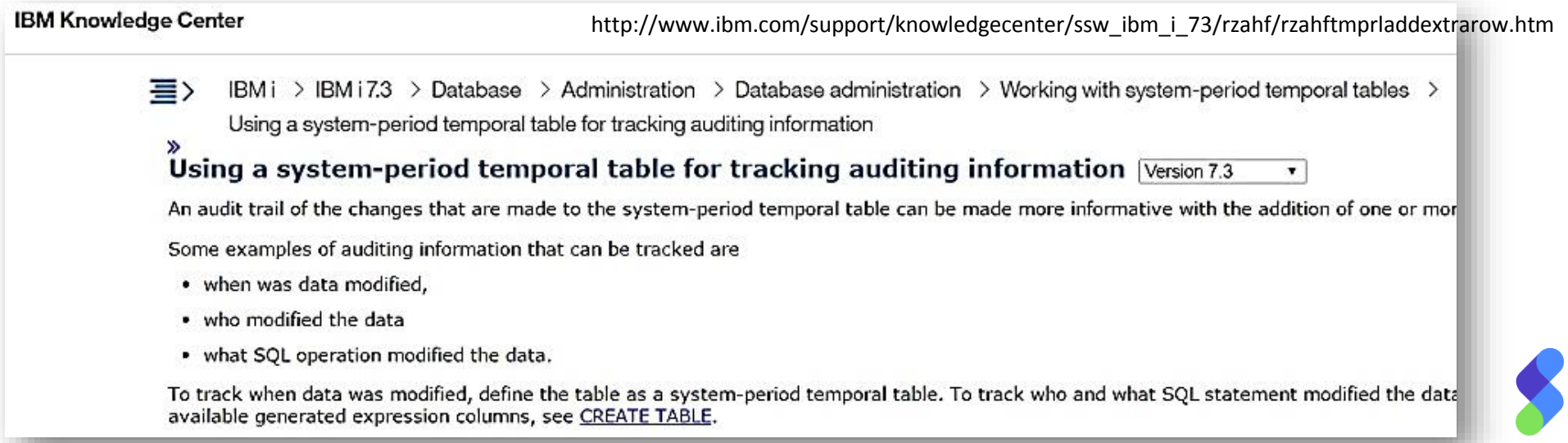
Other alternatives:
audit all these commands

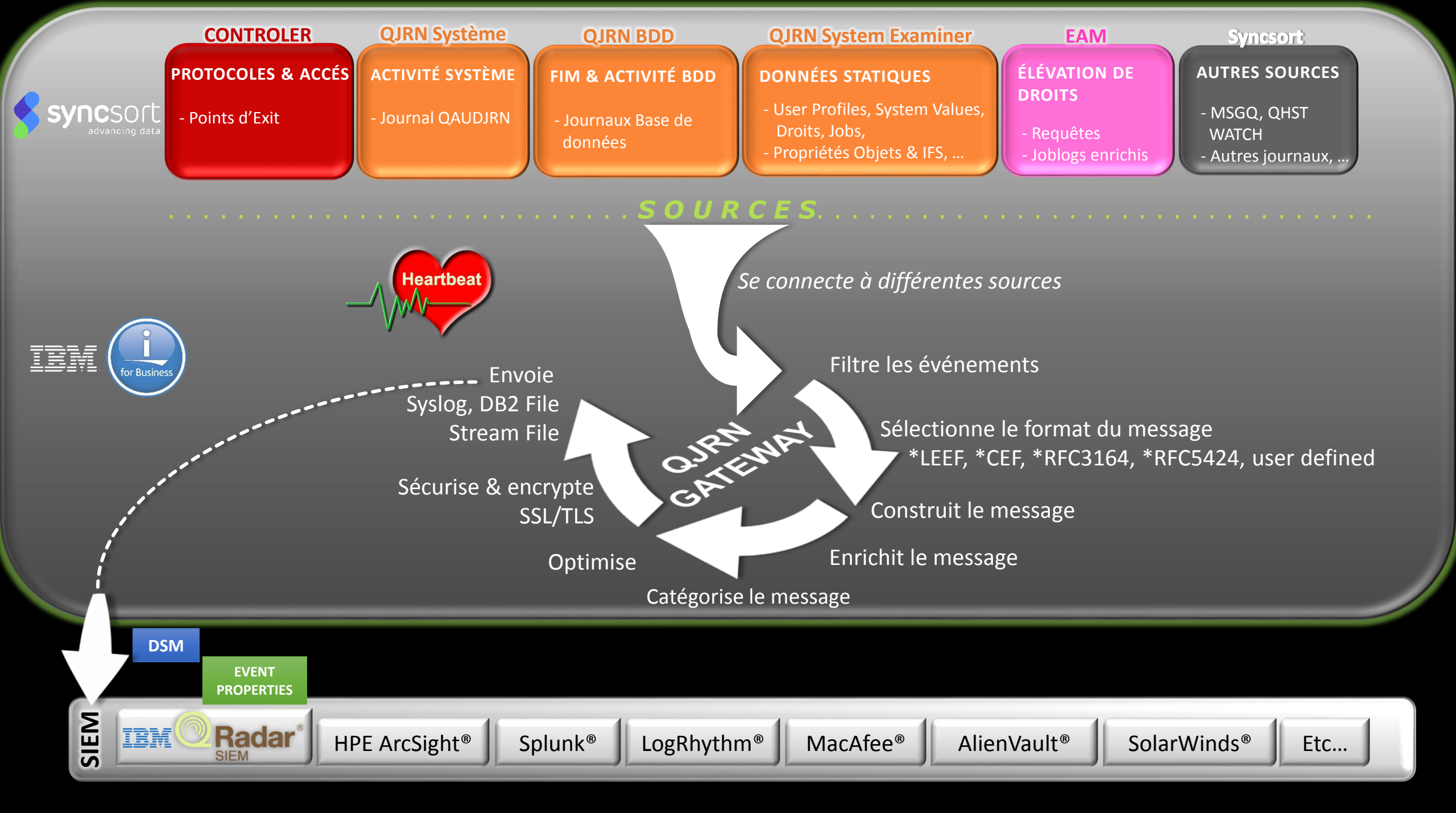
Other alternatives:
block commands using command exit point

46

Fonctions avancées

Goal / Concern	Answer	More Information
7.3: Temporal Tables	- The history file does not give the last insert/update operations as the main objective is to present the data at a certain date/time in the past. - Can be an alternative to database journal for some files - But more complex to implement than the journal - As a consequence, it cannot be applied to the entire database - Does not include operations at the file /member level like ALTER TABLE	




CONTROLER
PROTOCOLES & ACCÉS
- Points d'Exit
QJRN Système
ACTIVITÉ SYSTÈME
- Journal QAUDJRN
QJRN BDD
FIM & ACTIVITÉ BDD
- Journaux Base de données
QJRN System Examiner
DONNÉES STATIQUES
- User Profiles, System Values, Droits, Jobs,
- Propriétés Objets & IFS, …
EAM
ÉLÉVATION DE DROITS
- Requêtes
- Joblogs enrichis
Syncsort
AUTRES SOURCES
- MSGQ, QHST WATCH
- Autres journaux, …
SOURCES
Heartbeat
Se connecte à différentes sources
Filtre les événements
Sélectionne le format du message
*LEEF, *CEF, *RFC3164, *RFC5424, user defined
Construit le message
Enrichit le message
Catégorise le message
Optimise
Sécurise & encrypte
SSL/TLS
Envoie
Syslog, DB2 File
Stream File
QJRN GATEWAY
DSM
EVENT PROPERTIES
SIEM
IBM QRadar SIEM
HPE ArcSight®
Splunk®
LogRhythm®
MacAfee®
AlienVault®
SolarWinds®
Etc…

GDPR : Protection et traçabilité renforcées des données sur IBM i

Merci de votre attention !

Session – Université IBM i 2018 – IBM Client Center Paris

Animateur:

Guy MARMORAT

Senior Director of Product Management

16 MAI 2018