Université IBM i 2017

17 et 18 mai – IBM Client Center de Bois-Colombes

S9 –
“ **Le top 10 des astuces pour répondre aux réglementations et à l'audit IBM i** ”

Mercredi 17 mai – 14h50-15h30

Guy Marmorat – Cilasoft

Le top 10 des astuces pour répondre aux réglementations et à l'audit IBM i

Given the array of **regulations** that organizations face today, as well as the fact that **auditors** are becoming more demanding, achieving **compliance** is often a complex, difficult process.

This session gives you **practical tips** on how to efficiently report on relevant compliance information from **system and database journals**, how to address the need to audit **"super user activity"** that is part of most regulations, and how to accurately audit all levels of **access** to your **critical data**, including FTP and SQL.

In addition, this session reviews the latest improvements delivered in versions **7.1, 7.2 and 7.3** of IBM i.

By: Guy MARMORAT (CEO, Cilasoft)

Static sources

- User profiles
- System values
- Authorization lists
- Authorities
- Commands
- Shares
- Protocols available
- Function usage
- …

Dynamic sources

- System journal
- Database journals
- Triggers
- Exit points
- QHST, QSYSOPR, QSYSMSG
- Internal application audit trail
- Temporal tables (7.3)
- Joblogs

- Application audit fields in tables
- Generated Columns for Auditing in tables (7.2 & 7.3)

Static and dynamic sources complement each other…

Old method

New method

Special authorities

PRTUSRPRF TYPE(*AUTINFO)

Review *ALLOBJ users

Example from Scott Forstie

```
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)

Default Passwords

ANZDFTPWD

```
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.

Group Profiles

DSPUSRPRF USRPRF(XXX)
TYPE(*GRPMBR)
Interesting fields for DSPUSRPRF OUTPUT(*OUTFILE) :
UPGRPF Group profile
UPGRPI Group profile indicator
UPSUPG Supplemental groups

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

Old method

New method

Limited Capability Users

Interesting fields for DSPUSRPRF
OUTPUT(*OUTFILE) :
UPLTCP Limited capability

```
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

Password attempts

Interesting fields for DSPUSRPRF
OUTPUT(*OUTFILE) :
UPNVSA Verifications not valid

```
SELECT * FROM QSYS2.USER_INFO
 WHERE SIGN_ON_ATTEMPTS_NOT_VALID > 0
```

Not Used within 90 Days

Interesting fields for DSPUSRPRF
OUTPUT(*OUTFILE) :
UPNVSA Verifications not valid
UPPSOD Previous sign-on date:
UPPSOT Previous sign-on time
UPLSTD Last used date

```
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;
```

Example from Scott Forstie

Old method

New method

Users with *CMD auditing

Interesting fields for DSPUSRPRF
OUTPUT(*OUTFILE) :
UPAUDL Action auditing value

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

CHGUSRAUD USRPRF(XX) AUDLVL(*CMD)

Objects *USRPRF that can be used by other users

DSPOBJD OBJ(QSYS/*ALL) OBJTYPE(*USRPRF) OUTPUT(*OUTFILE) OUTFILE(QTEMP/USR) OUTMBR(*FIRST *ADD)
DSPOBJAUT OBJ(QSYS/&ODOBNM) OBJTYPE(*USRPRF) OUTPUT(*OUTFILE) OUTFILE(QTEMP/AUT) OUTMBR(*FIRST *ADD)
SELECT oaname, oausr,oaobja FROM aut
where oaname <> oausr and (oausr <>*PUBLIC'
or (oausr = '*PUBLIC' and oaobja <> EXCLUDE'))

api QUSLOBJ or
SELECT * FROM TABLE(QSYS2.OBJECT_STATISTICS('QSYS', '*USRPRF')) as x
then, api QSYRTVUA "Retrieve Users Authorized to an Object"
SELECT oaname, oausr,oaobja FROM aut
where oaname <> oausr and (oausr <> '*PUBLIC'
or (oausr = '*PUBLIC' and oaobja <> '*EXCLUDE'))

IBM User Profiles

Interesting fields for DSPUSRPRF
OUTPUT(*OUTFILE) :
UPCRTBY Created by user

Select * from QSYS2.user_info
where user_creator in ('*IBM','QLPINSTALL')

System values	WRKSYSVAL OUTPUT(*PRINT)	Select * from qsys2.system_value_info
Job Description Users	PRTJOBDAUT LIB(XXX)	Api QWDRJOBD "Retrieve Job Description Information"
Authorization lists	DSPOBJD OBJ(QSYS/*ALL) OBJTYPE(*AUTL) OUTPUT(*OUTFILE) OUTFILE(QTEMP/OBJ) OUTMBR(*FIRST *REPLACE) DSPAUTL AUTL(XXXX) OUTPUT(*OUTFILE) OUTFILE(QTEMP/AUTL) OUTMBR(*FIRST *ADD) SELECT * FROM AUTL DSPAUTLOBJ AUTL(XXXX) OUTPUT(*OUTFILE) OUTFILE(QTEMP/AUTLOBJ) OUTMBR(*FIRST *ADD) SELECT * FROM AUTLOBJ	**DSPAUTL**: SELECT * FROM QSYS2.AUTHORIZATION_LIST_USER_INFO WHERE AUTHORIZATION_NAME = '*PUBLIC'; https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20i%20Technology%20Updates/page/QSYS2.AUTHORIZATION_LIST_USER_INFO%20View **DSPAUTLOBJ**: SELECT * FROM QSYS2.AUTHORIZATION_LIST_INFO WHERE AUTHORIZATION_LIST = 'XXXX' https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20i%20Technology%20Updates/page/QSYS2.AUTHORIZATION_LIST_INFO%20View

no text, nor default value shipped value in the SQL view

Old method

New method

Object Authorities

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

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')

IFS Authorities

RTVDIRINF lists all the directories and IFS objects in output files
api QSYRTVUA gets the authority
DSPAUT does not have any OUTFILE parameter unfortunately.

CRTPF FILE(xxx/IFSDIR) RCDLEN(500)
QSH CMD('ls -lf /directory > /qsys.lib/xxx.lib/ifsdir.file/ifsdir.mbr')
Then, api QSYRTVUA gets the authority

	Old method	New method
User Objects in QSYS	PRTUSROBJ LIB(QSYS)	
File Shares	api QZLSOLST — QSYS.LIB is part of root. Can be protected by autl QPWFSERVER	
Server Authentication Entries	DSPSVRAUTE USRPRF(XXXX) OUTPUT(*PRINT)	SELECT * FROM QSYS2.DRDA_AUTHENTICATION_ENTRY_INFO
Trigger programs	PRTTRGPGM LIB(*ALL)	SELECT * FROM QSYS2.SYSTRIGGERS
Function Usage	DSPFCNUSG OUTPUT(*PRINT)	SELECT * FROM QSYS2.FUNCTION_USAGE SELECT * FROM QSYS2.FUNCTION_INFO
Objects that Adopt Authority	DSPPGMADP USRPRF(XXXX)	
Commands with Prompt Override or Validity Checker Programs	no OUTFILE keyword available	api QCDRCMDI — Prompt override, Validity Checker Program, Allow Limited Users
Check Object Integrity	CHKOBJITG USRPRF(*ALL) OUTFILE(QTEMP/CHKOBJITG))	

5. SECTOOLS versus SQL Services

Old method

GO SECTOOLS

Work with profiles
1. Analyze default passwords

2. Display active profile list
3. Change active profile list
4. Analyze profile activity

5. Display activation schedule
6. Change activation schedule entry

7. Display expiration schedule
8. Change expiration schedule entry

9. Print profile internals

Work with auditing
10. Change security auditing
11. Display security auditing
12. Copy audit journal entries

Reports
20. Submit or schedule security reports to batch

21. Adopting objects
22. Audit journal entries
……

DSPUSRPRF versus user_info:
100% identical
new fields added in both in 7.3:
AUTHORITY_COLLECTION_ACTIVE
AUTCOLREP

New method

Select * from QSYS2.SERVICES_INFO
Full catalog with examples, OS version and PTF Group

Most popular Services :

USER_INFO	List Users & attributes
FUNCTION_INFO	List Functions & attributes
FUNCTION_USAGE	List Function usage IDs
GROUP_PROFILE_ENTRIES	List group profiles & associated users
DRDA_AUTHENTICATION_ENTRY_INFO	List Server Authentication Entries (DRDA)
SYSTEM_VALUE_INFO	List System Values
OBJECT_STATISTICS	List objects & attributes
DISPLAY_JOURNAL	Display Journal entries
QCMDEXC	Run a Command
TCPIP_INFO	Retrieve TCP/IP information
NETSTAT_INFO	List the current connections
WLM_SET_CLIENT_INFO	Set values to client resgisters
DATABASE_MONITOR_INFO	List the current monitors

OBJECT_PRIVILEGES	List object authorities
AUTHORIZATION_LIST_INFO	List objects secured by an auth. list
AUTHORIZATION_LIST_USER_INFO	List auth. lists and their authorities.

https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20i%20Technology%20Updates/page/DB2%20for%20i%20-%20Services

Static sources

- User profiles
- System values
- Authorization list
- Authorities
- Commands
- Shares
- Protocols available
- Function usage
- …

Advantages
✓Reliable
✓infalsifiable
✓not selective
✓close to the system

Drawbacks
✓Impact on CPU?
✓Noise?
✓Readable?

INDEPENDENT FROM THE APPLICATION

Dynamic sources

The winner is « Journals »!

- System journal
- Database journals
- Triggers
- Exit points
- QHST, QSYSOPR, QSYSMSG
- Internal application audit trail
- Temporal tables (7.3)
- Joblogs

- Application audit fields in tables
- Auditing Columns in tables (7.3)

Static and dynamic sources complement each other…

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	***JOBUSRPGM still the default value - not allowed for QAUDJRN *RMTADR and *PGMLIB are useful;**
CRTJRN, CHGJRN **Manage receivers . . . . . MNGRCV**	*SYSTEM *USER	**Detachment based on the journal receiver threshold and at each IPL**
CRTJRN, CHGJRN **Delete receivers . . . . . DLTRCV**	*NO *YES	**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.**

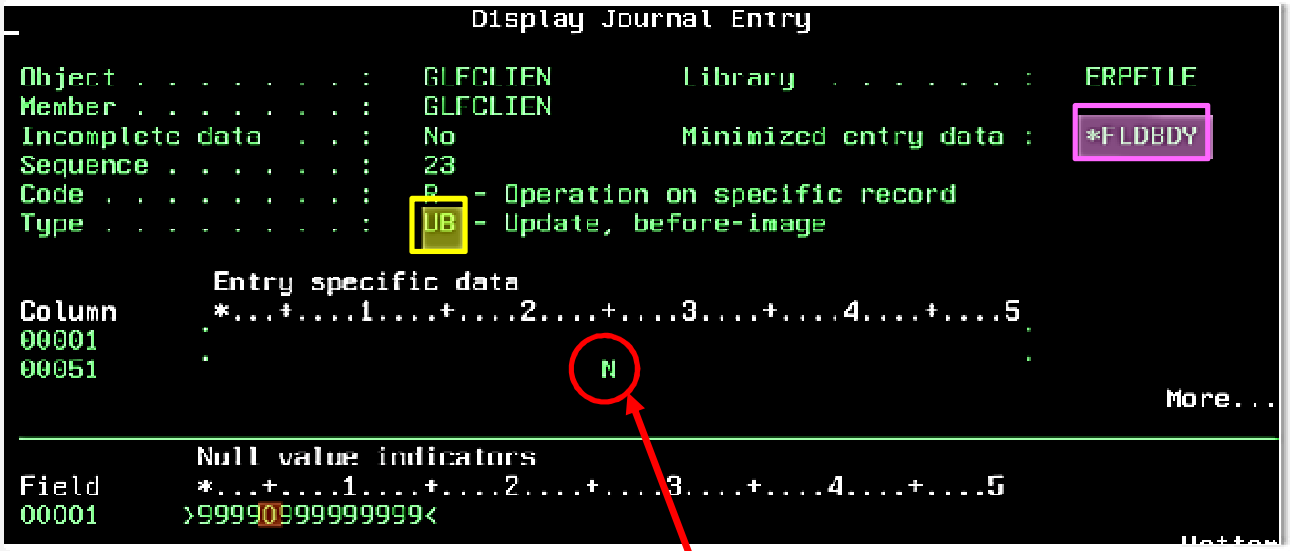
Command & Keyword

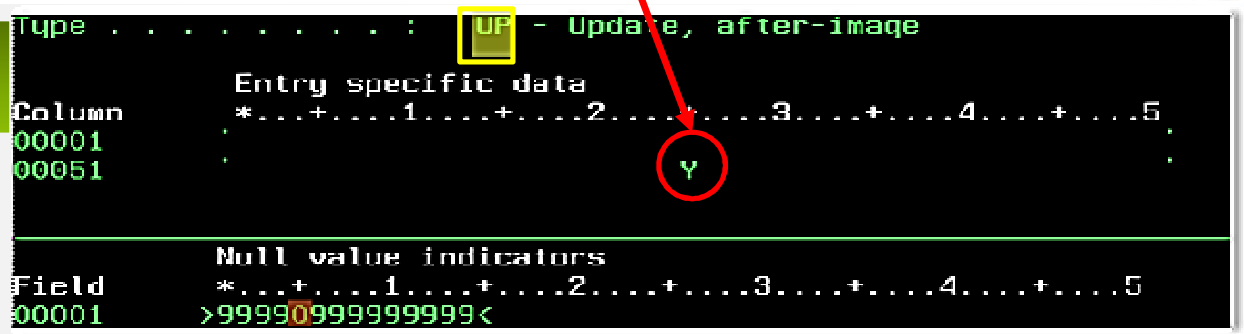
CRTJRN, CHGJRN
Minimize entry specific data . .
MINENTDTA

Possible Values

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...

Goal

How to find journal codes and journal entry types?

Statistics by entry type

Answer

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
```

Old method is to use DSPJRN OUTFILE(..) and query the resulting file

Alternative Answer

Website:

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


```
select JOURNAL_ENTRY_TYPE , count(*)
from table(Display_Journal('QSYS','QAUDJRN', Journal_Codes =>
'T'))
as x
group by journal_entry_type
order by journal_entry_type
```

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

Goal	Answer
Secure journals and receivers	Correct Authorities on journals & receivers
Saving journal receivers	QjoRtvJrnReceiverInformation and check field Status

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

Status. The status of the journal receiver. The status can be one of the following:

1 The journal receiver is currently attached to the journal.

2 The journal receiver is online. The journal receiver has not been saved, and it has been detached from the journal.

3 The journal receiver was saved after it was detached. The journal receiver storage was not freed when it was saved.

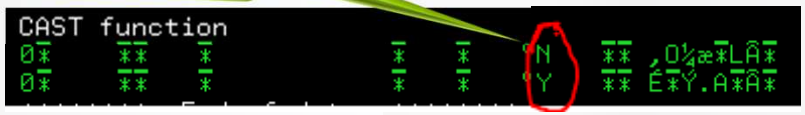
4 The journal receiver was saved after it was detached. The journal receiver storage was freed when it was saved.

5 The journal receiver status is partial for one of the following reasons:

Command & Keyword	Possible Values	
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 !**

Goal / Concern	Answer	More Information
How to interpret the program & program library in a journal entry?	Last value in the call stack which is not QSYS	example: DFU = QDZTD00001, but … update within STRSQL could be QCMD
Details of journal entries	Old method is to use DSPJRN OUTFILE(..) and analyze the resulting file	Select cast(entry_data as char(200)) from table(Display_Journal('IJRNDTA','ERPJRN', starting_receiver_name => '*CURCHAIN')) as x where object like '% GLFCLIEN %' and journal_code = 'R'

Example of the Entry_Data field when *FLDBDY (only the value of the field changed is recorded)



Example of the Entry_Data field when *NONE (the entire record is recorded)

7. Typical reports on Database Journals

- Changes to application access files (users, roles, menus, options, general parameters)
- Changes to business critical data (client, item, pricing, discount, credit limit, salary, bank account, …)
- Changes to data subject to regulations (personal identity information, credit card, medical data, customer address, …)
- Changes to data via programs that are not part of the application (DFU, SQL, 3rd party products, …)
- Changes to data outside of normal business hours

8. System Audit Journal: Pollution?

Goal / Concern

Are ZC entry types still used for replication?

Answer

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
- …..

More Information

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

8. System Audit Journal: Pollution?

Goal / Concern	Answer	More Information
ZR entry types used for what?	Also common to see some ZR entries while there was no real reason for that. In this case: there is no risk to change the audit value back to *CHANGE	Use command CHGAUD for IFS objects
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

8. System Audit Journal: Command auditing

Goal / Concern

How to record full commands?

How to see only the commands that have been typed on the command line?

Answer

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)

CPYAUDJRNE ENTTYP(CD)
select cdtstp,cduspf, cdjob, cdcmds
from qtemp/qauditcd where cdetyp <> 'X' and cdclp in ('B','N')

Select entry_timestamp as Timestamp,
substr(current_user,1,10) as Current_User,job_name,
substr(cast(entry_data as char(250)),31,200) as Command
From table(Display_Journal('QSYS','QAUDJRN',
starting_receiver_name => '*CURCHAIN')) as x
where journal_entry_type = 'CD' and
cast(entry_data as char(1)) <>'X' and
substr(cast(entry_data as char(30)),30,1) in ('B','N')

More Information

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

From the Security Reference Guide, layout for CD:

185	253	639	Where run	Char(1)	Where the CL command was run.
				Y	From a compiled OPM CL program or an ILE CL Program
				R	From a REXX procedure
				E	The command string was passed as a parameter to one of the Command Analyzer APIs: QCMDEXC, QCAPCMD, or QCAEXEC
				B	In a batch job but not for any of the reason listed under Y, R, or E. Typical case would be that the CL command was run using STRDBRDR or SBMDBJOB command or was specified on the CMD parameter of the SBMJOB command.
				N	Interactively from a command line or by choosing a menu option that runs a CL command

New values introduced in 7.2 & 7.1 + PTF SI44865

8. Typical reports on the System Audit journal

Changes to System values	SV	
Changes to Network Attributes	NA	
Changes to Auditing Values	AD	
Changes to Authorities/Ownership	CA/OW	
Command auditing	CD	(user or command audited)
Authorization failures	AF	
Adopted authority	AP	
Sign on violations	PW	
Changes to Exit Points	GR	Action = ZC & Function registration = A,D,R
Changes to Function Usage	GR	Action = ZC & Function registration = F
Change attempts to Function Usage	GR	Action = ZC / Field1 = * USAGEFAILURE (useful for RCAC)
Changes to JOBDs	JD	
Jobs	JB	
Changes to objects	CO, DO, OM, OR,	
Changes to RCAC	AX	
Changes to user profiles	CP	(deletions not included)
Actions to spooled files	SP	

Goal / Concern	Answer
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?	New QAUDLVL and QAUDLVL2 values: *NETSECURE, *NETTELSVR, and *NETUDP. The QAUDLVL and QAUDLVL2 value *NETCMN now only writes security audit journal entries for a subset of the *NETSCK functions. It does not write security audit journal entries for accepts and connects. 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.

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

10. Auditing Columns in 7.3

Goal / Concern

7.3: Auditing Columns

Answer

```
ALTER TABLE erpfile.glfcliaud
ADD COLUMN audit_type_change CHAR (1)
GENERATED ALWAYS AS (DATA CHANGE OPERATION)
ADD COLUMN audit_user VARCHAR(128)
GENERATED ALWAYS AS (SESSION_USER)
ADD COLUMN audit_client_IP VARCHAR(128)
GENERATED ALWAYS AS (SYSIBM.CLIENT_IPADDR)
ADD COLUMN audit_job_name VARCHAR(28)
GENERATED ALWAYS AS (QSYS2.JOB_NAME)          ;
```

More Information

Very useful, easy to implement, compatible with journal

Limitations: last context recorded only. Delete not covered

Goal / Concern

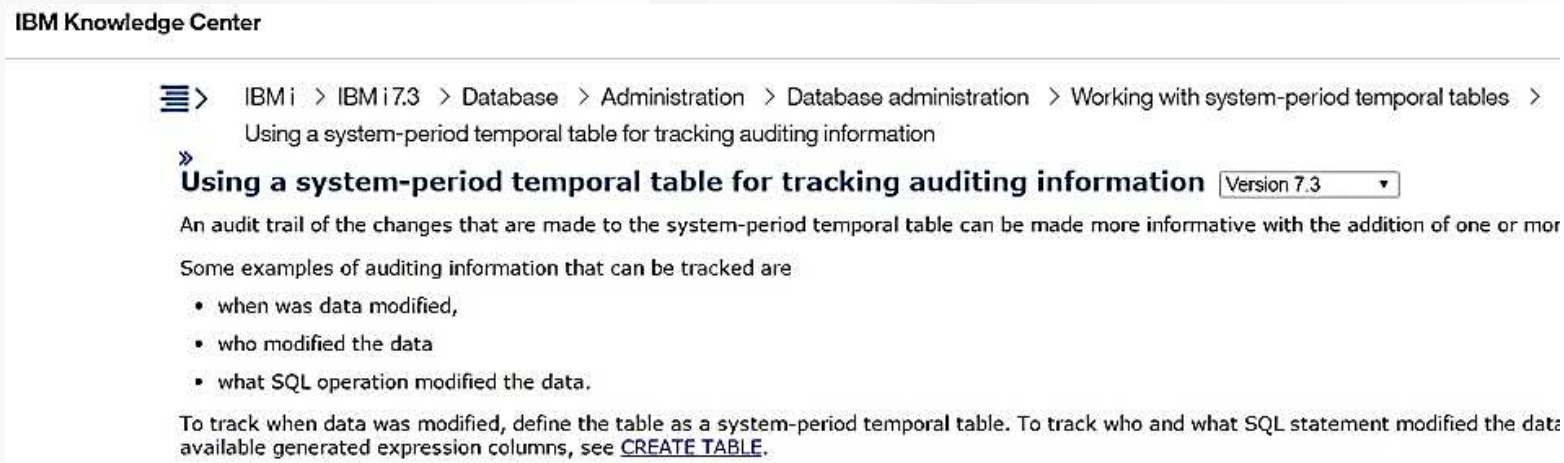
7.3: Temporal Tables

Answer

- 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

More Information

http://www.ibm.com/support/knowledgecenter/ssw_ibm_i_73/rzahf/rzahftmprladdextrarow.htm



IBM Knowledge Center

IBM i > IBM i 7.3 > Database > Administration > Database administration > Working with system-period temporal tables > Using a system-period temporal table for tracking auditing information

Using a system-period temporal table for tracking auditing information Version 7.3

An audit trail of the changes that are made to the system-period temporal table can be made more informative with the addition of one or mor

Some examples of auditing information that can be tracked are

- when was data modified,
- who modified the data
- what SQL operation modified the data.

To track when data was modified, define the table as a system-period temporal table. To track who and what SQL statement modified the data available generated expression columns, see CREATE TABLE.

- **Tracking at object level - Who is opening this file?**
- **Tracking at record level - Who is reading this record?**

At the object level :

- 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

At the record level :

- 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)

Ressources

- Security reference guides
- Redbooks
- developerWorks
- Wikis

Research on the web

Jeff Uehling + security

Scott Forstie + DB2

Thank you to those who have reviewed and approved my presentation!

Thank you to members of my team who participated in creating this presentation!


Products
QJRN/400, CONTROLER, EAM, DVM, and CENTRAL
interfaced with SIEM Solutions
QJRN/400
EAM
DVM
CENTRAL
POST FILE
Audit & Security on IBM i
Auditing, traceability, privacy data, data
protection and
help companie
*Detect
*Comply
Sarbanes-Oxle
Card Industry
Basel II, HIPAA
cilasoft
Expert en Sécurité sur IBM i
Copyright 2017 © Cilasoft

Qui sommes-nous ?

+ de 27 ans d'expérience sur IBM i (AS/400, iSeries)

Cilasoft est éditeur de Logiciels d'Audit, de Compliance et de Sécurité spécialisés sur IBM i

Cilasoft est certifié :

- IBM Advanced Business Partner
  - Ready For PureSystems
    - Ready For Security Intelligence

Ses solutions sont reconnues comme leaders sur la plateforme IBM i et sont référencées dans le

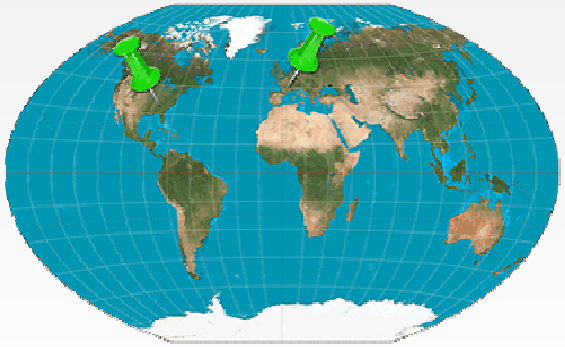
IBM Global Solutions Directory

Et

IBM i Solution Editions

Qui sommes-nous ?

Mise en Conformité:
SOX, GDPR, PCI-DSS, Bâle II, Solvency, Etc.



Editeur à l'International

La Suite Compliance Cilasoft :

1. **QJRN/400**
2. **CONTROLER**
3. **DATABASE VIEW MONITOR (DVM)**
4. **ELEVATED AUTHORITY MANAGER (EAM)**
5. **CENTRAL**
6. **POST FILE**
7. **JOB LOG Explorer** (outil gratuit)

- **Siège Social : Annecy, France**
- **Filiale : Atlanta, USA**
- **Réseaux de Partenaires à travers le Monde**
- **Clients : + de 300 dans 70 pays**
  + 1500 partitions installées

La Suite Compliance

QJRN/400

AUDIT & COMPLIANCE

- Rapports et Alertes sur les évènements Bases de données et Système
- System Examiner (sources statiques)
- Interface SIEM

EAM

ELEVATION DE DROITS

- Attribution de droits élevés temporairement et selon besoins
- Activité auditée et loguée

CENTRAL

- Consolidation & déploiement de données DB2
- Lancement de commandes simultanément sur machines distantes
- Préconfiguré pour Cilasoft

POST FILE

- Rapports et Alertes
- MSGQ, Menus, profiles

CONTROLER

CONTRÔLE D'ACCES GLOBAL

- ODBC, JDBC, OLE DB
- FTP, DDM, DRDA, NetServer
- Jobs, Sockets
- File open, SQL engine
- Commandes

DVM

AUDIT DES ACCES EN LECTURE

- Audit des Accès en lecture sur vos données sensibles au niveau enregistrement

JOB LOG EXPLORER

- Aide à l'analyse des joblogs grâce à ses filtres puissants
- Multi-critères, mult-langues
- Joblogs sous forme de fichier texte en local ou importation depuis la machine directement

Université IBM i 2017

17 et 18 mai – IBM Client Center de Bois-Colombes

S9 –

“ **Le top 10 des astuces pour répondre aux réglementations et à l'audit IBM i** ”

Mercredi 17 mai – 14h50-15h30

Guy Marmorat – Cilasoft

contact@cilasoft.com | www.cilasoft.com