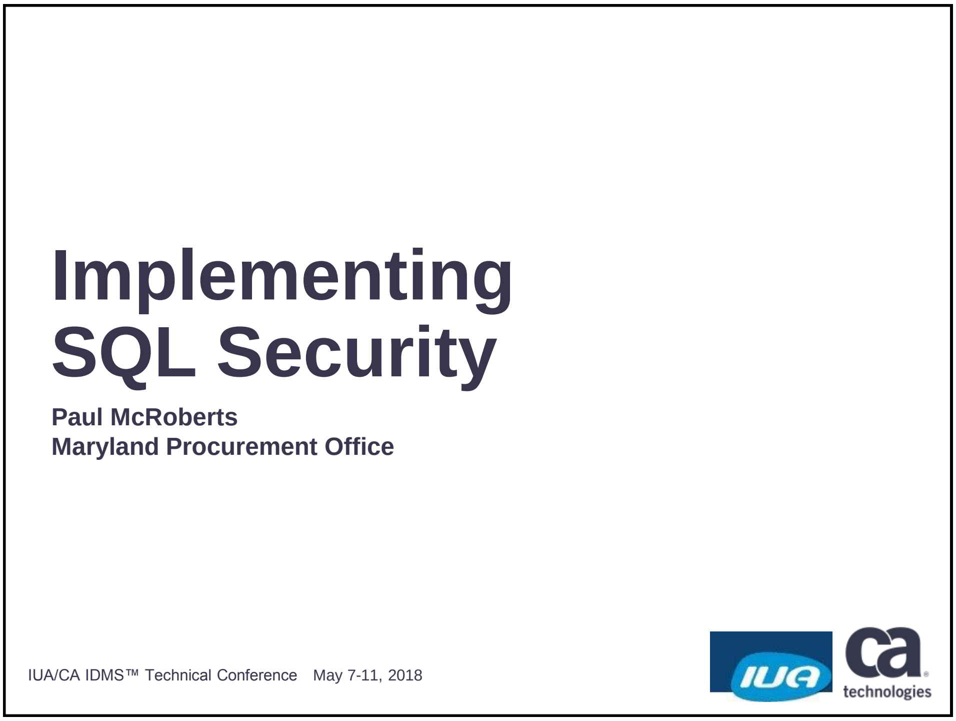
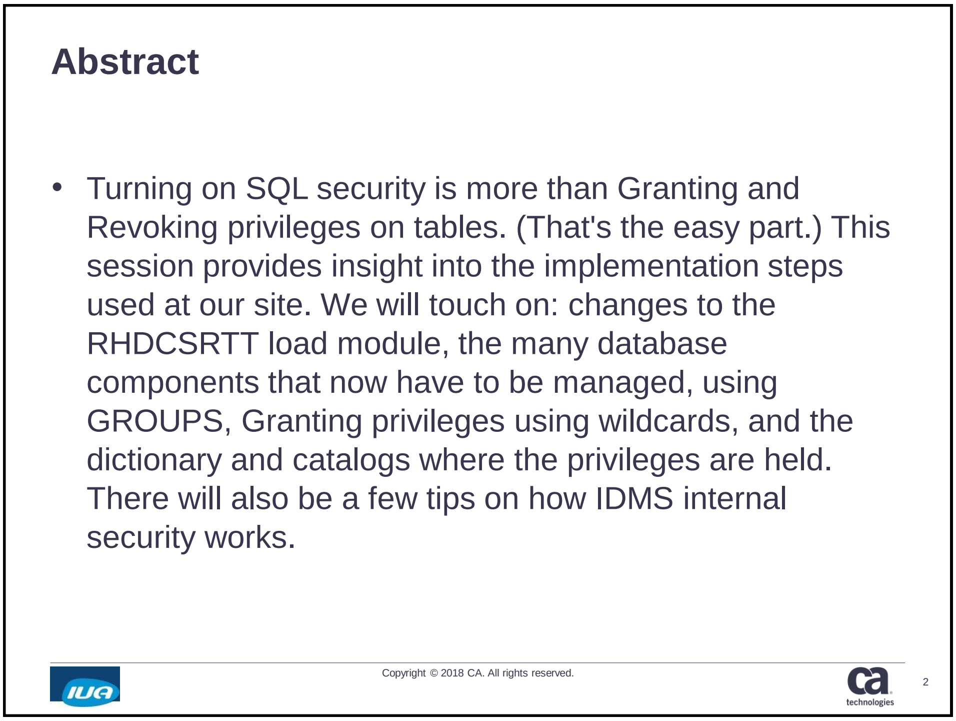


Implementing SQL Security

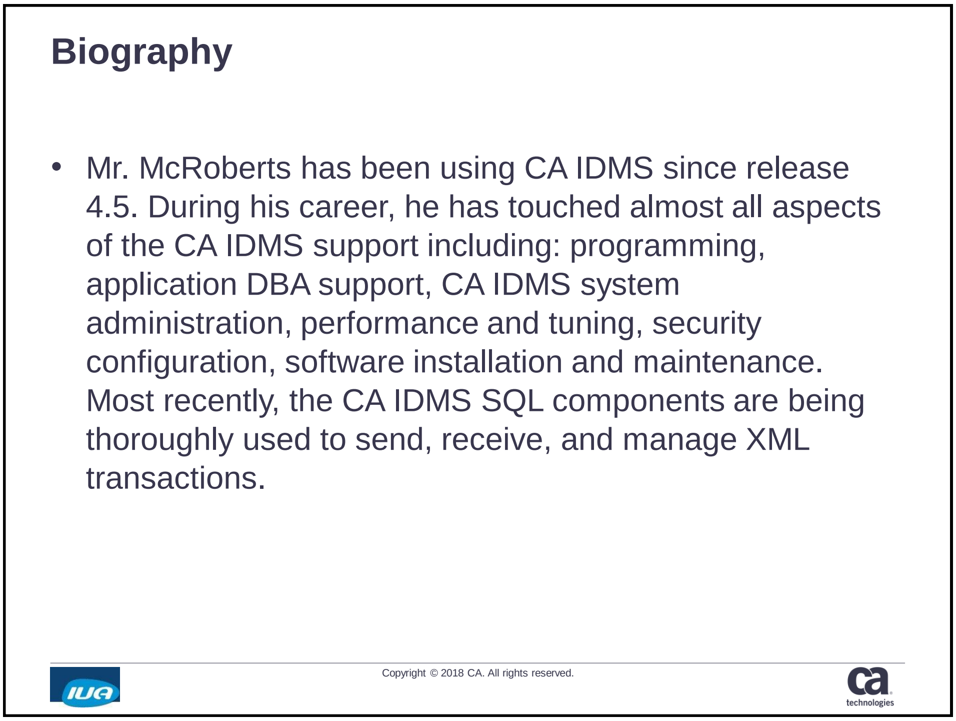
Paul McRoberts
Maryland Procurement Office

IUA/CA IDMS™ Technical Conference May 7-11, 2018



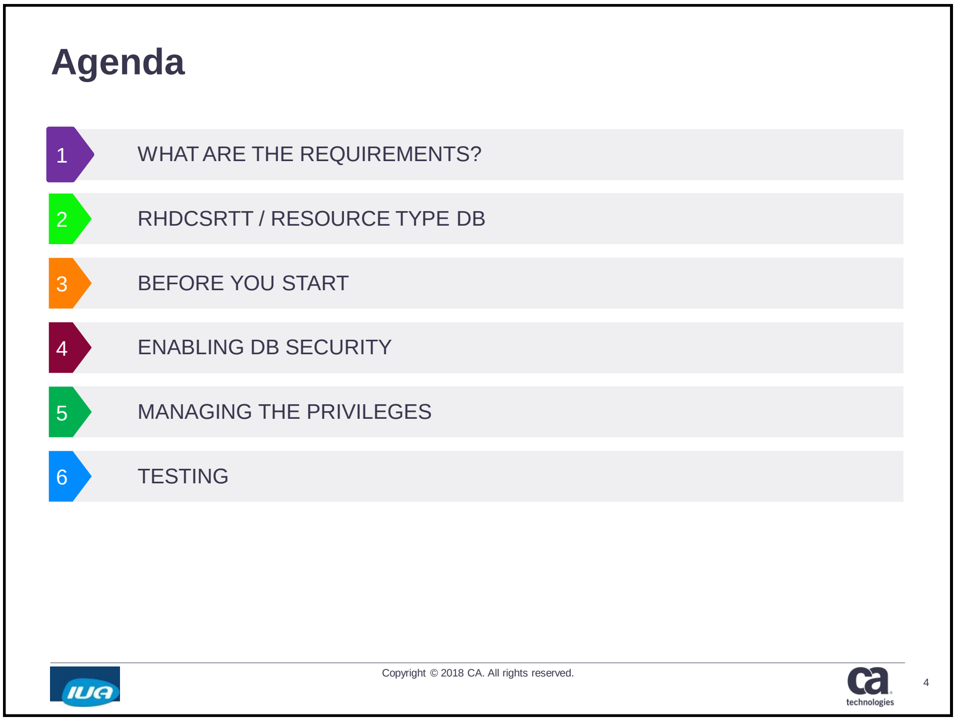
Abstract

- Turning on SQL security is more than Granting and Revoking privileges on tables. (That's the easy part.) This session provides insight into the implementation steps used at our site. We will touch on: changes to the RHDCSRTT load module, the many database components that now have to be managed, using GROUPS, Granting privileges using wildcards, and the dictionary and catalogs where the privileges are held. There will also be a few tips on how IDMS internal security works.



Biography

- Mr. McRoberts has been using CA IDMS since release 4.5. During his career, he has touched almost all aspects of the CA IDMS support including: programming, application DBA support, CA IDMS system administration, performance and tuning, security configuration, software installation and maintenance. Most recently, the CA IDMS SQL components are being thoroughly used to send, receive, and manage XML transactions.



Agenda

1 WHAT ARE THE REQUIREMENTS?

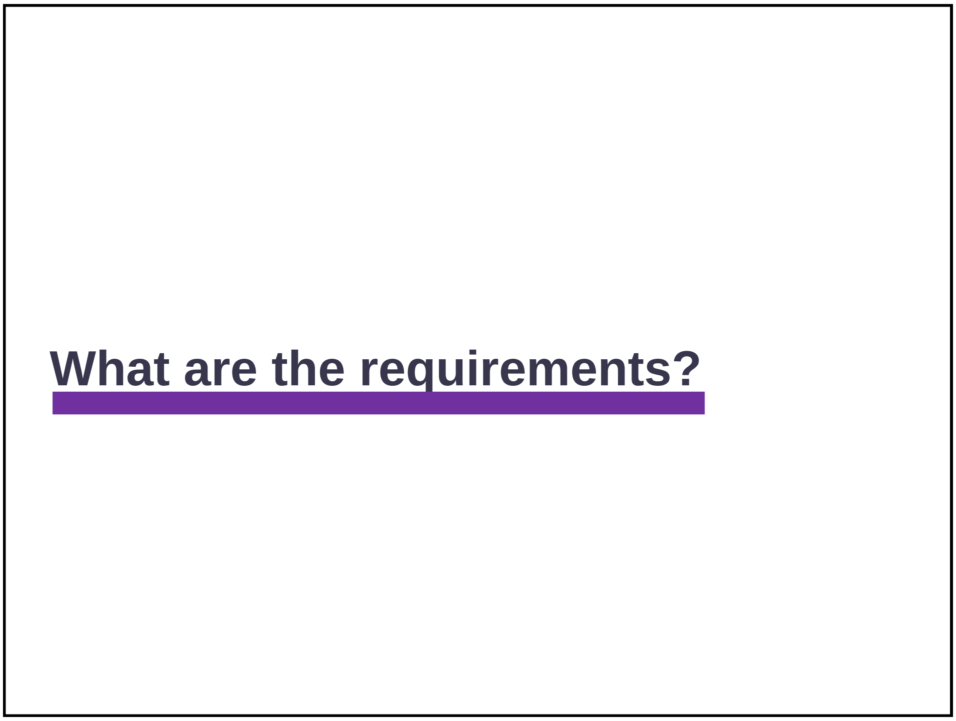
2 RHDCSRTT / RESOURCE TYPE DB

3 BEFORE YOU START

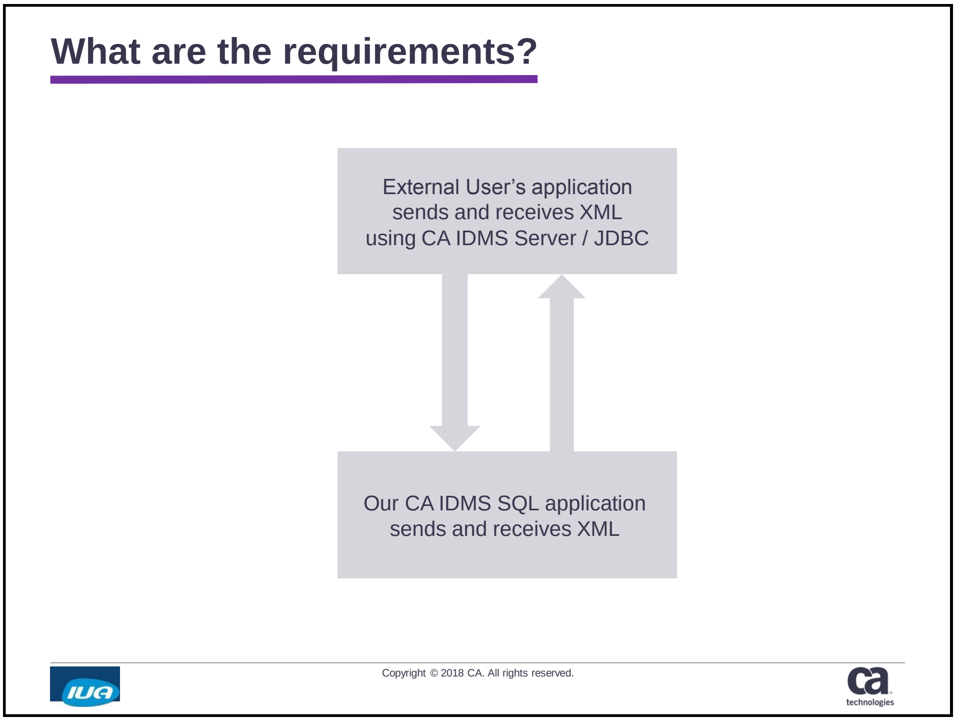
4 ENABLING DB SECURITY

5 MANAGING THE PRIVILEGES

6 TESTING



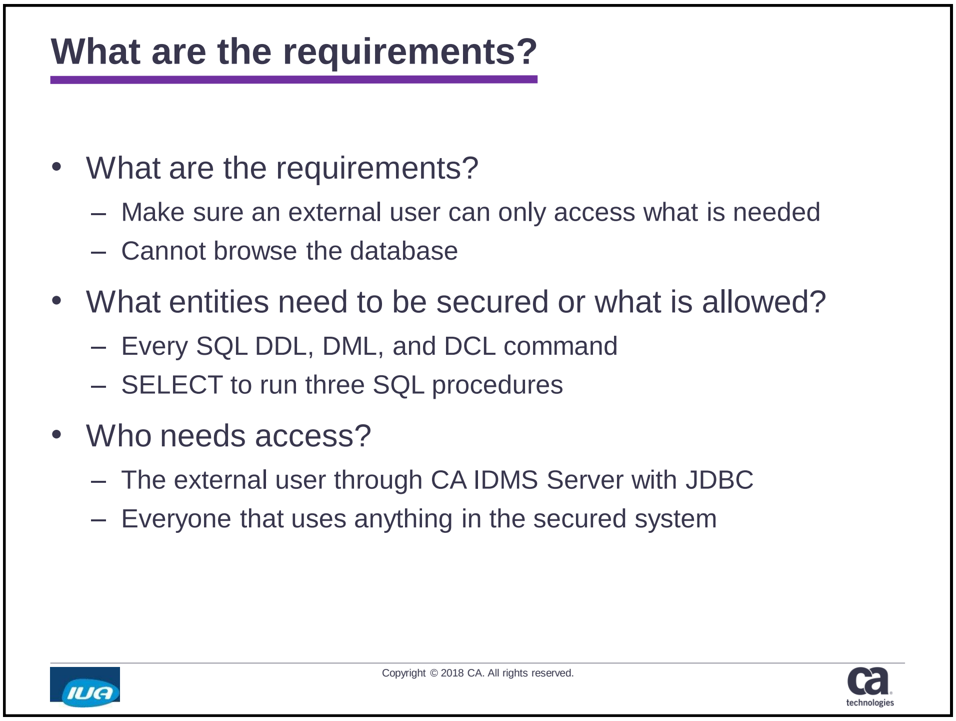
What are the requirements?



What are the requirements?

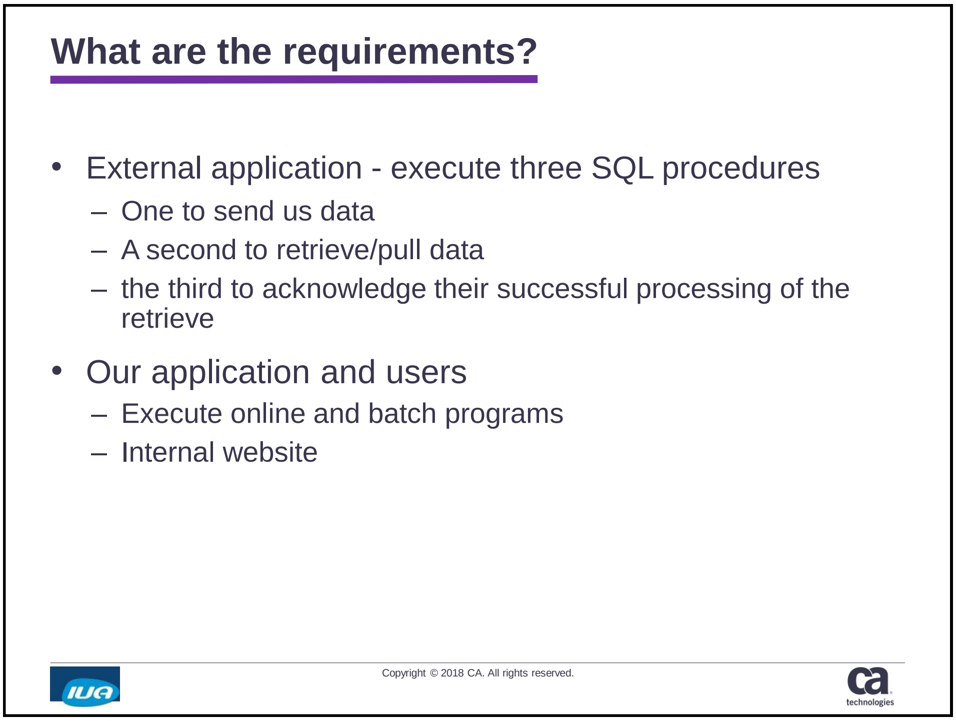
External User's application sends and receives XML using CA IDMS Server / JDBC

Our CA IDMS SQL application sends and receives XML



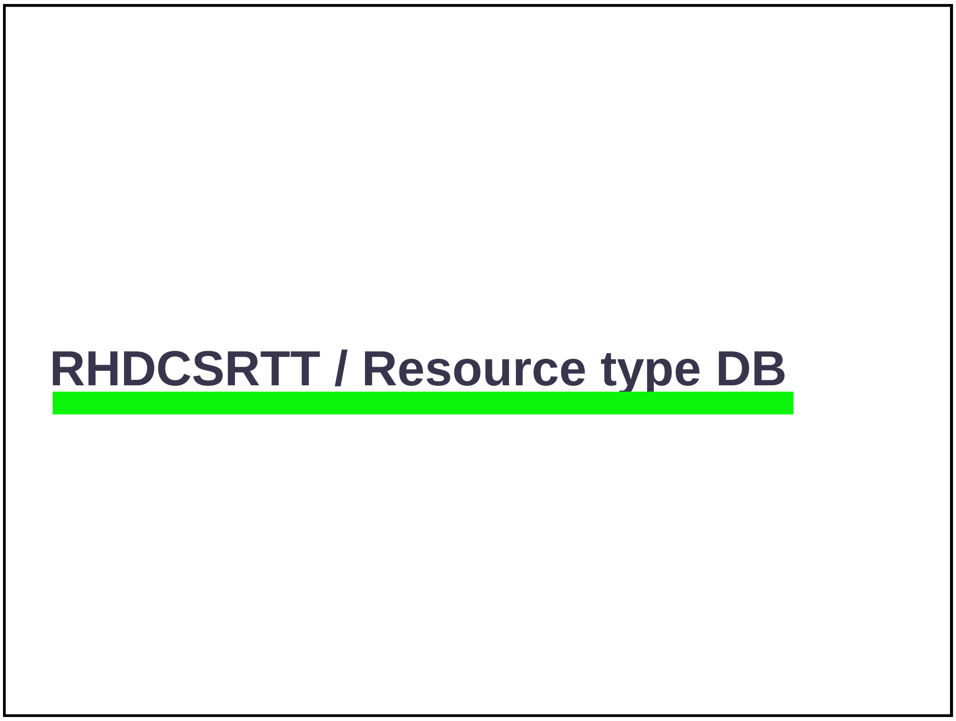
What are the requirements?

- What are the requirements?
  - Make sure an external user can only access what is needed
  - Cannot browse the database
- What entities need to be secured or what is allowed?
  - Every SQL DDL, DML, and DCL command
  - SELECT to run three SQL procedures
- Who needs access?
  - The external user through CA IDMS Server with JDBC
  - Everyone that uses anything in the secured system

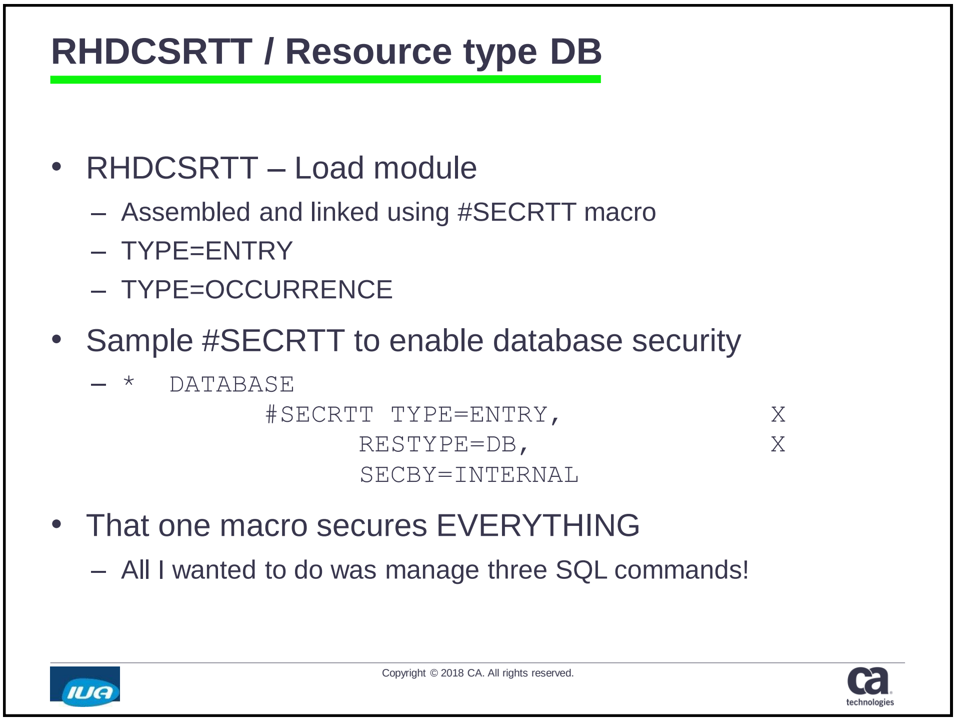


What are the requirements?

- External application - execute three SQL procedures
  - One to send us data
  - A second to retrieve/pull data
  - the third to acknowledge their successful processing of the retrieve
- Our application and users
  - Execute online and batch programs
  - Internal website



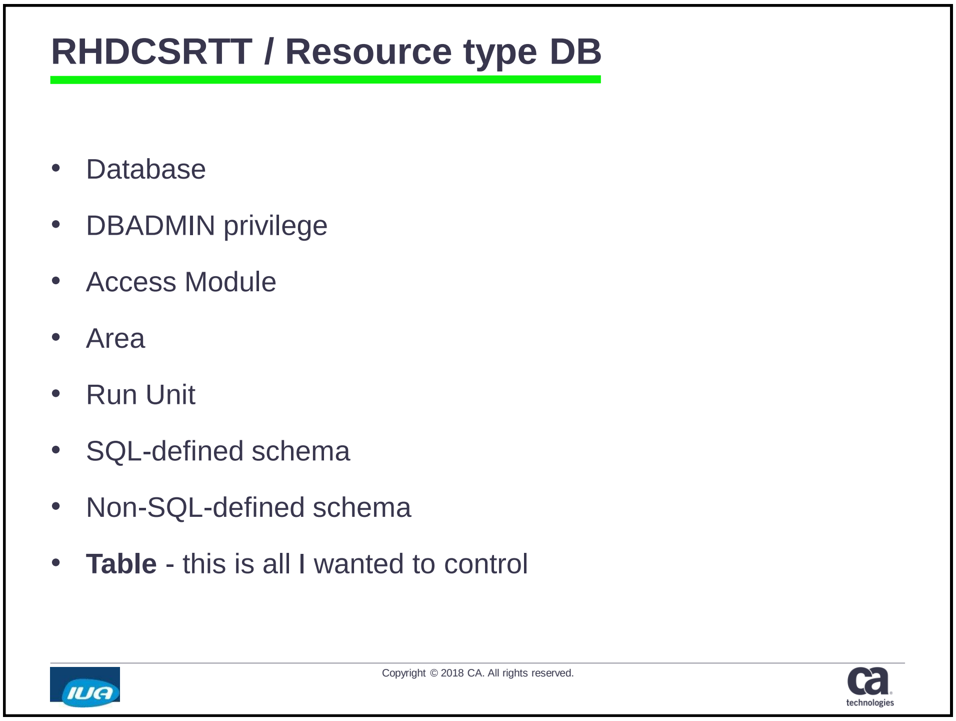
RHDCSRTT / Resource type DB



RHDCSRTT / Resource type DB

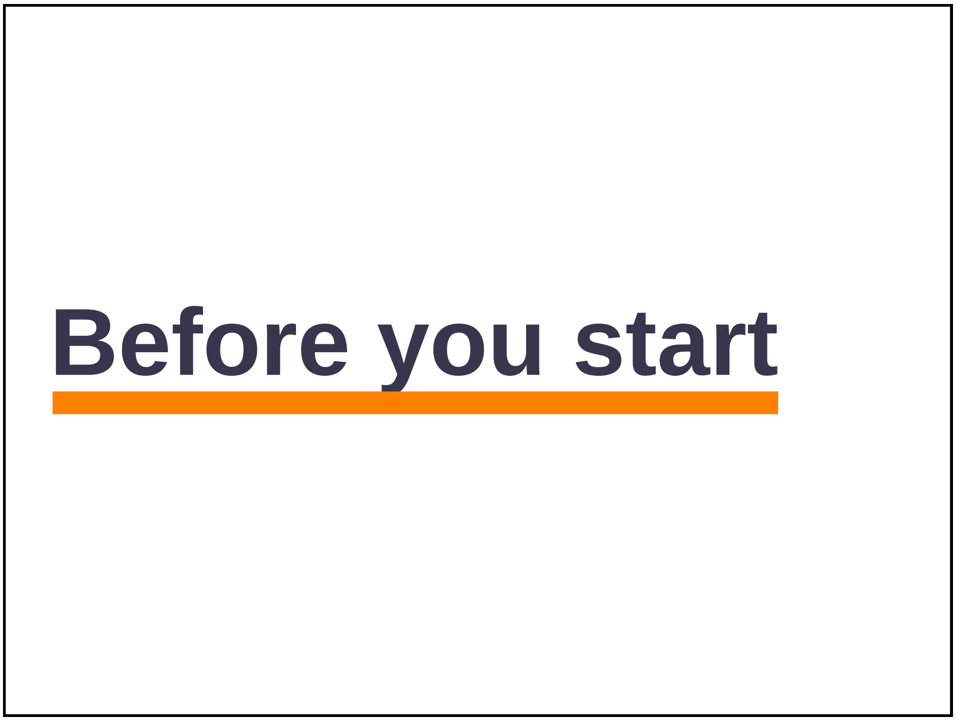
- RHDCSRTT – Load module
  - Assembled and linked using #SECRTT macro
  - TYPE=ENTRY
  - TYPE=OCCURRENCE
- Sample #SECRTT to enable database security

  -
    ```
    *   DATABASE
                 #SECRTT TYPE=ENTRY,                 X
                       RESTYPE=DB,                   X
                       SECBY=INTERNAL
    ```
- That one macro secures EVERYTHING
  - All I wanted to do was manage three SQL commands!

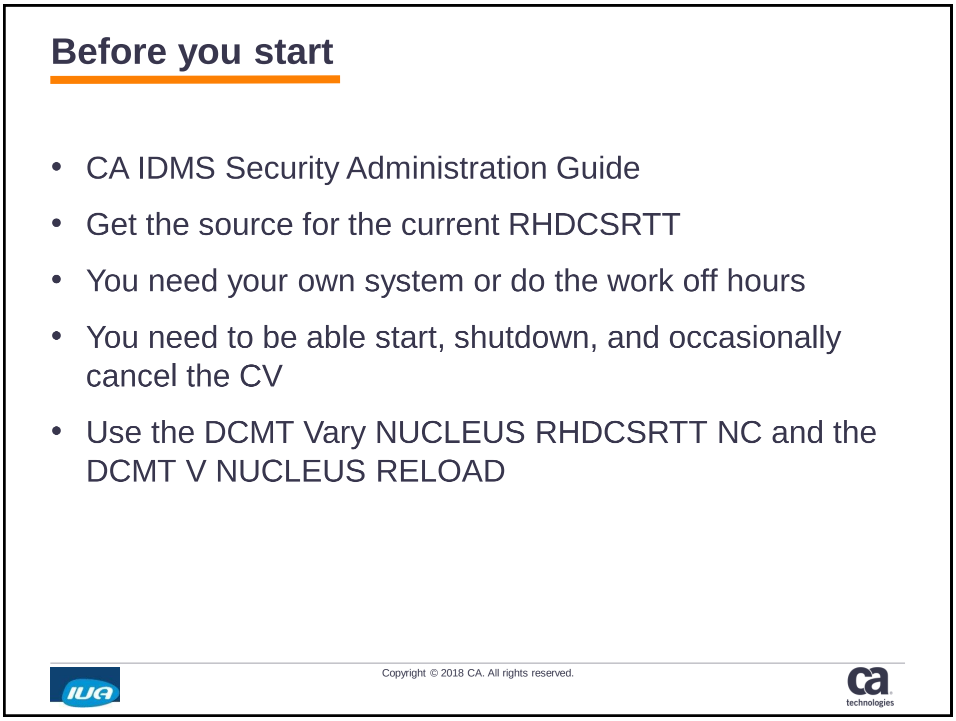


RHDCSRTT / Resource type DB

- Database
- DBADMIN privilege
- Access Module
- Area
- Run Unit
- SQL-defined schema
- Non-SQL-defined schema
- **Table** - this is all I wanted to control

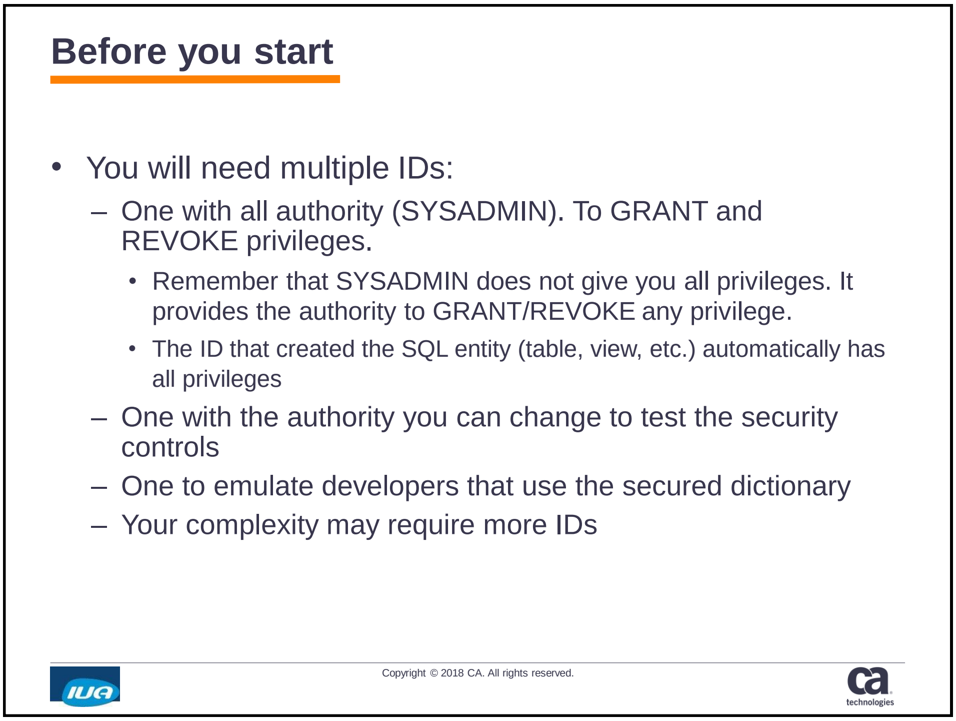


Before you start



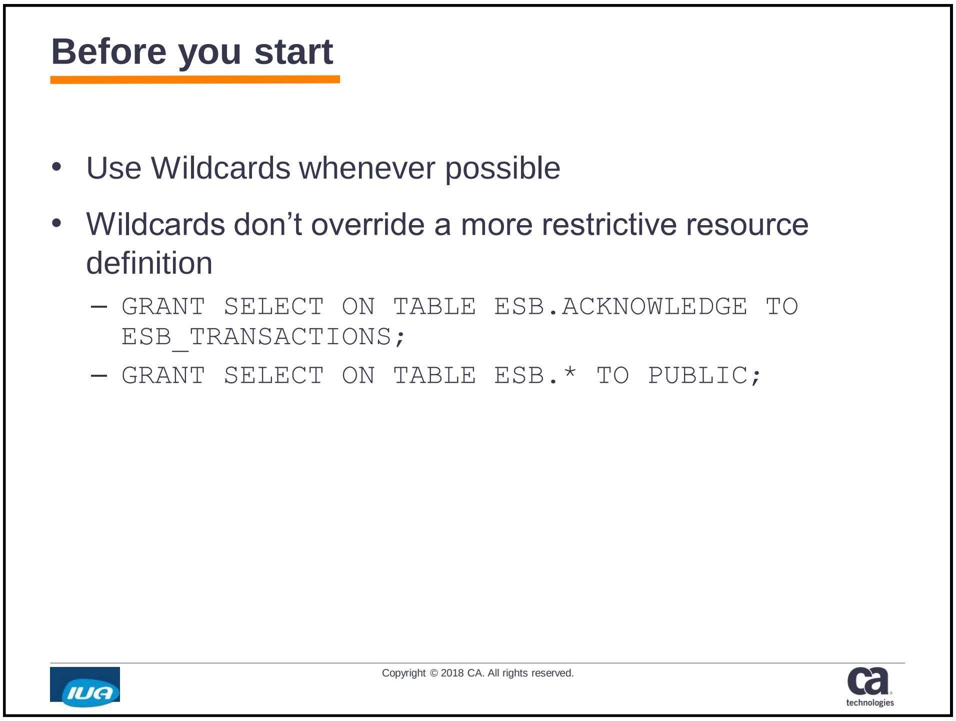
Before you start

- CA IDMS Security Administration Guide
- Get the source for the current RHDCSRTT
- You need your own system or do the work off hours
- You need to be able start, shutdown, and occasionally cancel the CV
- Use the DCMT Vary NUCLEUS RHDCSRTT NC and the DCMT V NUCLEUS RELOAD



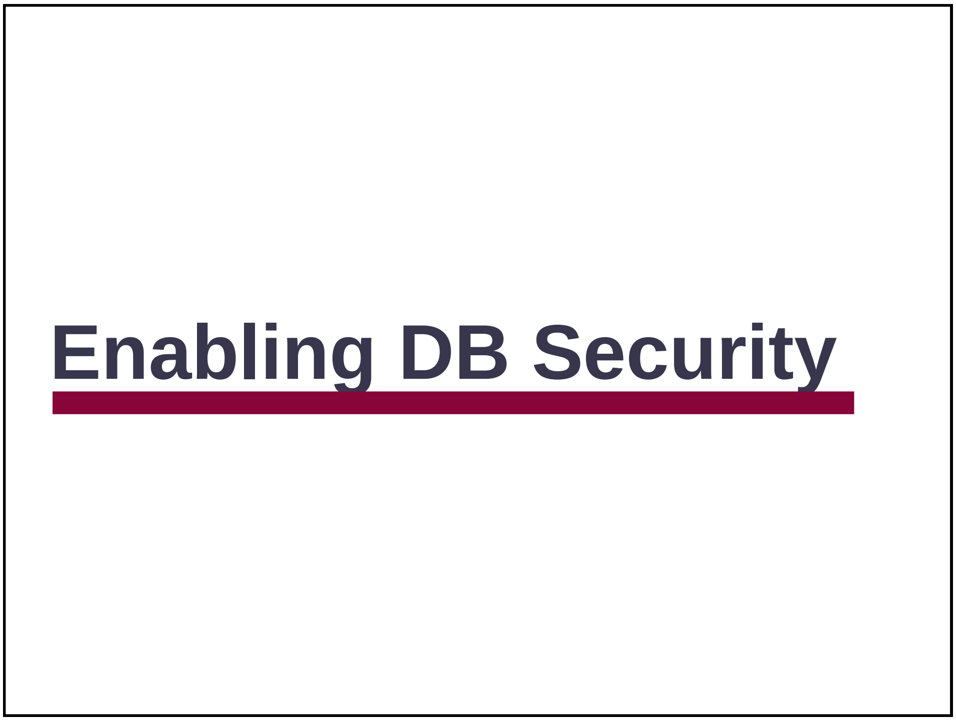
Before you start

- You will need multiple IDs:
  - One with all authority (SYSADMIN). To GRANT and REVOKE privileges.
    - Remember that SYSADMIN does not give you all privileges. It provides the authority to GRANT/REVOKE any privilege.
    - The ID that created the SQL entity (table, view, etc.) automatically has all privileges
  - One with the authority you can change to test the security controls
  - One to emulate developers that use the secured dictionary
  - Your complexity may require more IDs

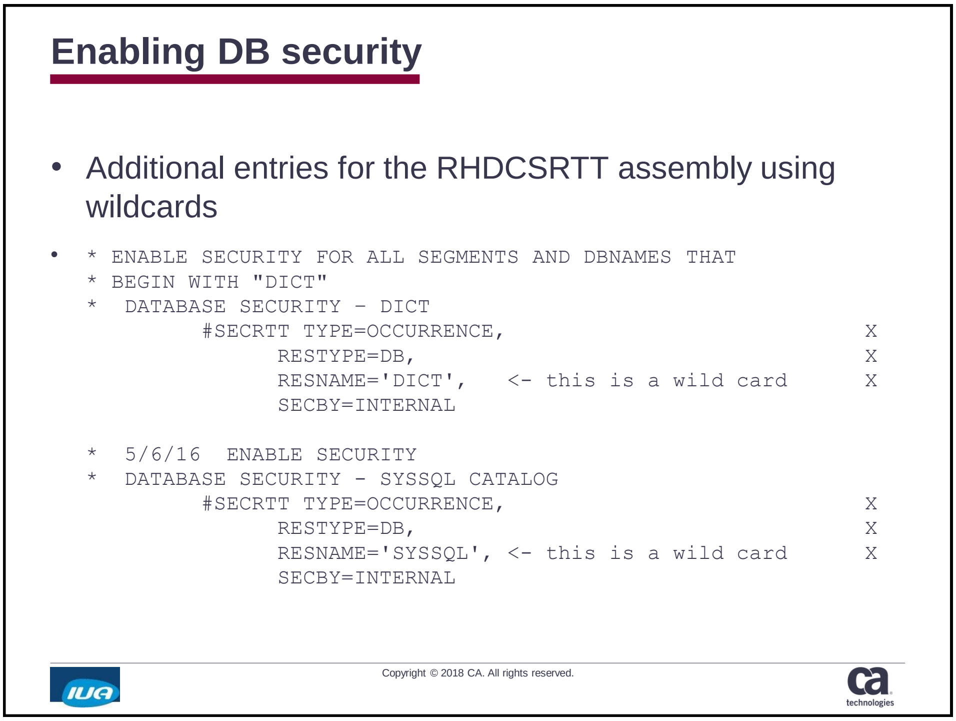


Before you start

- Use Wildcards whenever possible
- Wildcards don’t override a more restrictive resource definition
  - `GRANT SELECT ON TABLE ESB.ACKNOWLEDGE TO ESB_TRANSACTIONS;`
  - `GRANT SELECT ON TABLE ESB.* TO PUBLIC;`



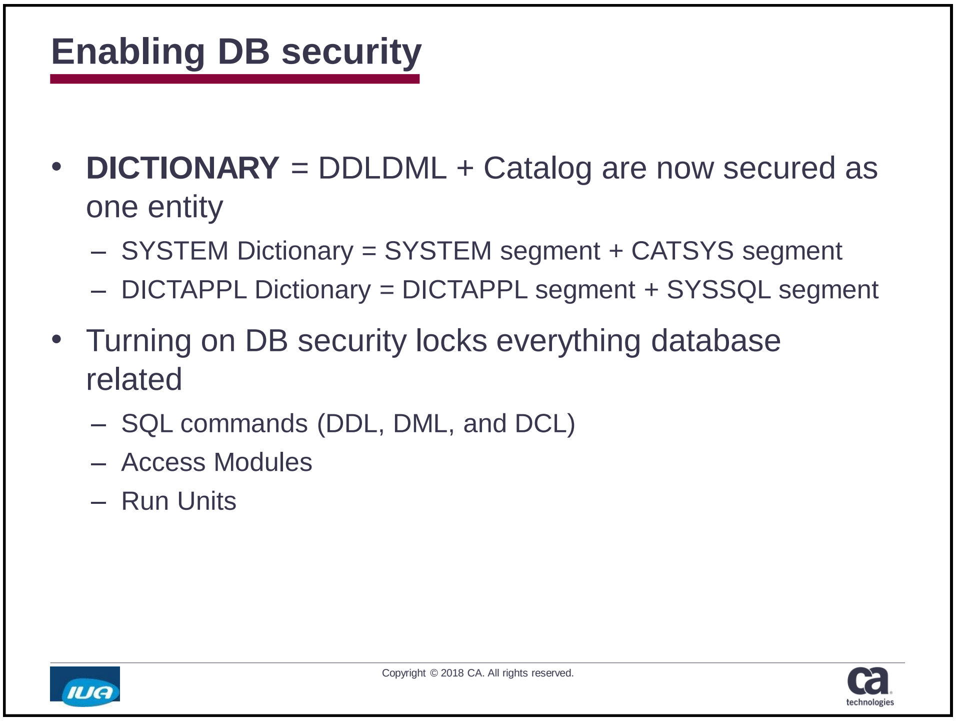
Enabling DB Security



Enabling DB security

- Additional entries for the RHDCSRTT assembly using wildcards

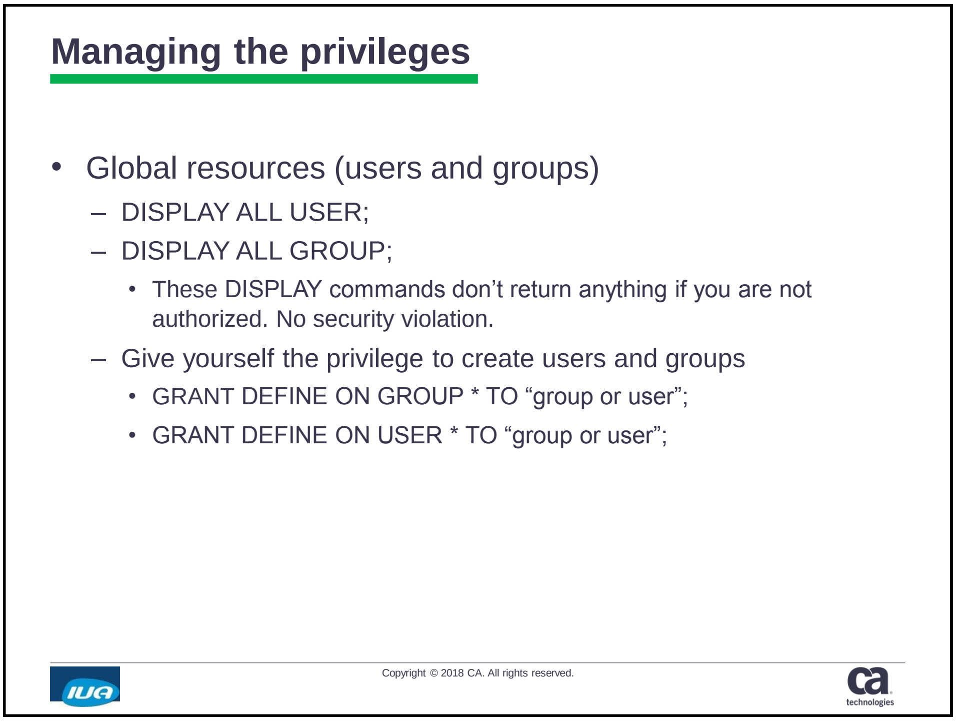
```
* ENABLE SECURITY FOR ALL SEGMENTS AND DBNAMES THAT
* BEGIN WITH "DICT"
*  DATABASE SECURITY - DICT
         #SECRTT TYPE=OCCURRENCE,                            X
               RESTYPE=DB,                                   X
               RESNAME='DICT',   <- this is a wild card      X
               SECBY=INTERNAL

*  5/6/16  ENABLE SECURITY
*  DATABASE SECURITY - SYSSQL CATALOG
         #SECRTT TYPE=OCCURRENCE,                            X
               RESTYPE=DB,                                   X
               RESNAME='SYSSQL', <- this is a wild card      X
               SECBY=INTERNAL
```



Enabling DB security

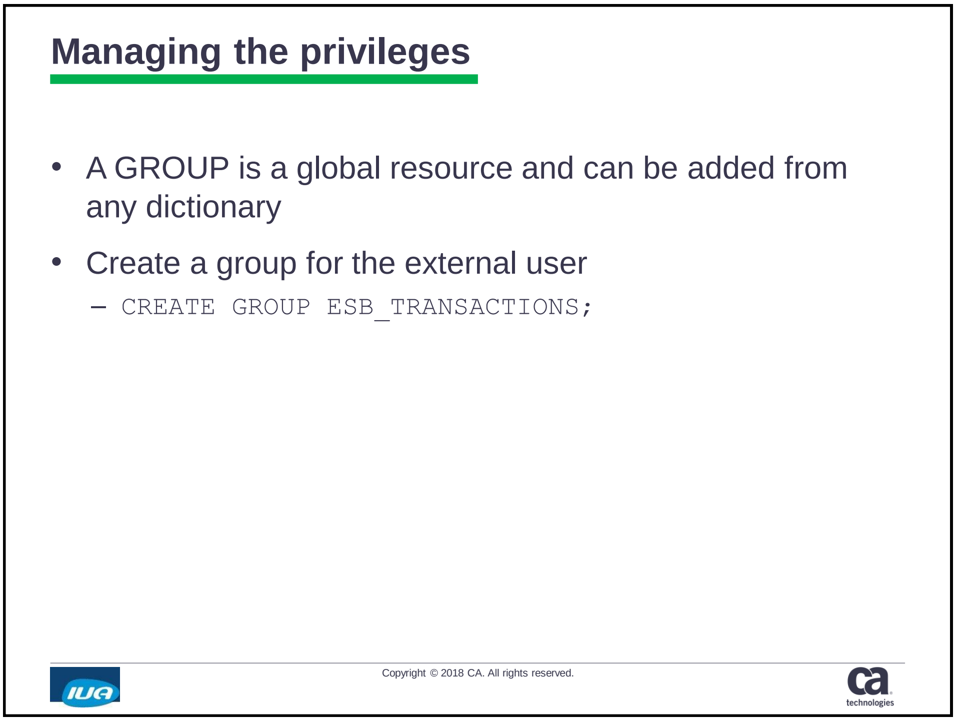
- **DICTIONARY** = DDLDML + Catalog are now secured as one entity
  - SYSTEM Dictionary = SYSTEM segment + CATSYS segment
  - DICTAPPL Dictionary = DICTAPPL segment + SYSSQL segment
- Turning on DB security locks everything database related
  - SQL commands (DDL, DML, and DCL)
  - Access Modules
  - Run Units

Managing the privileges



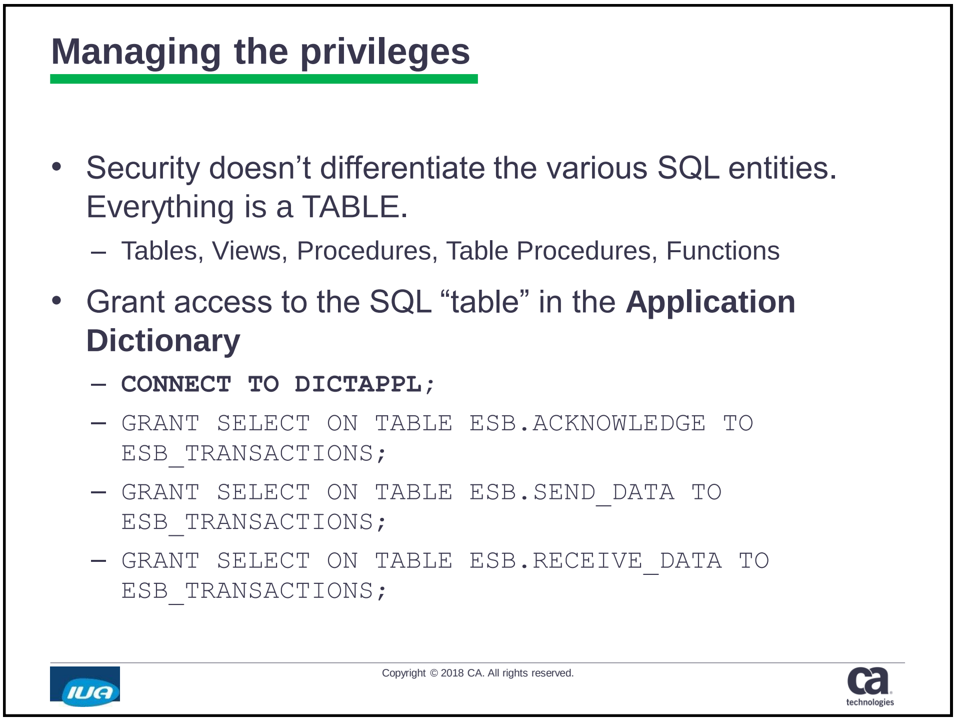
Managing the privileges

- Global resources (users and groups)
  - DISPLAY ALL USER;
  - DISPLAY ALL GROUP;
    - These DISPLAY commands don’t return anything if you are not authorized. No security violation.
  - Give yourself the privilege to create users and groups
    - GRANT DEFINE ON GROUP * TO “group or user”;
    - GRANT DEFINE ON USER * TO “group or user”;



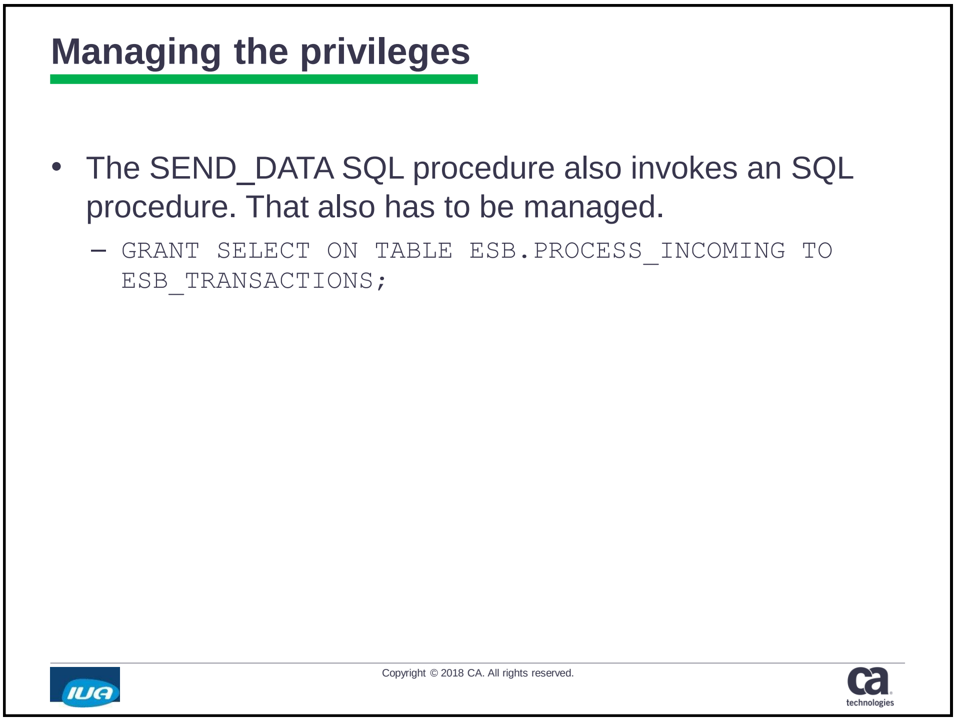
Managing the privileges

- A GROUP is a global resource and can be added from any dictionary
- Create a group for the external user
  - `CREATE GROUP ESB_TRANSACTIONS;`



Managing the privileges

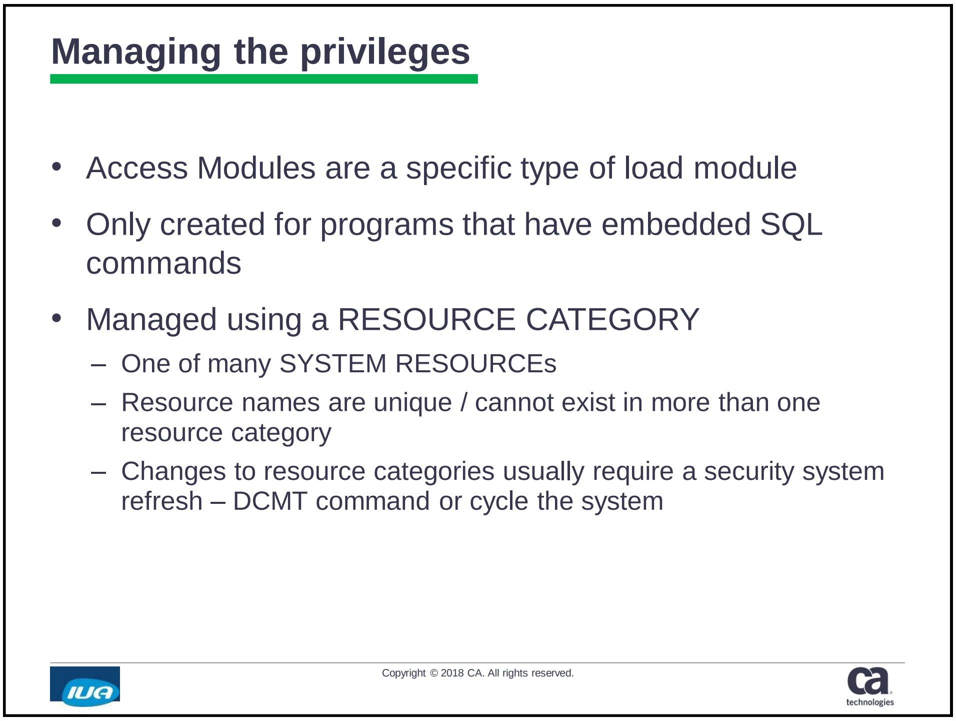
- Security doesn’t differentiate the various SQL entities. Everything is a TABLE.
  - Tables, Views, Procedures, Table Procedures, Functions
- Grant access to the SQL “table” in the **Application Dictionary**
  - **`CONNECT TO DICTAPPL;`**
  - `GRANT SELECT ON TABLE ESB.ACKNOWLEDGE TO ESB_TRANSACTIONS;`
  - `GRANT SELECT ON TABLE ESB.SEND_DATA TO ESB_TRANSACTIONS;`
  - `GRANT SELECT ON TABLE ESB.RECEIVE_DATA TO ESB_TRANSACTIONS;`



Managing the privileges

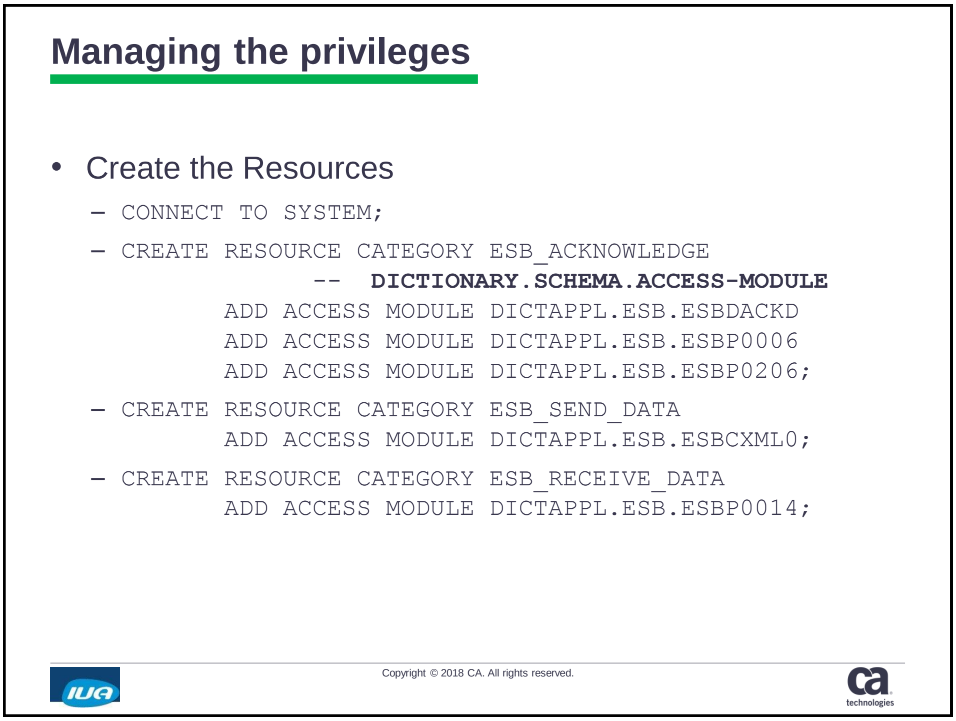
- The SEND_DATA SQL procedure also invokes an SQL procedure. That also has to be managed.

  -
    ```
    GRANT SELECT ON TABLE ESB.PROCESS_INCOMING TO
    ESB_TRANSACTIONS;
    ```



Managing the privileges

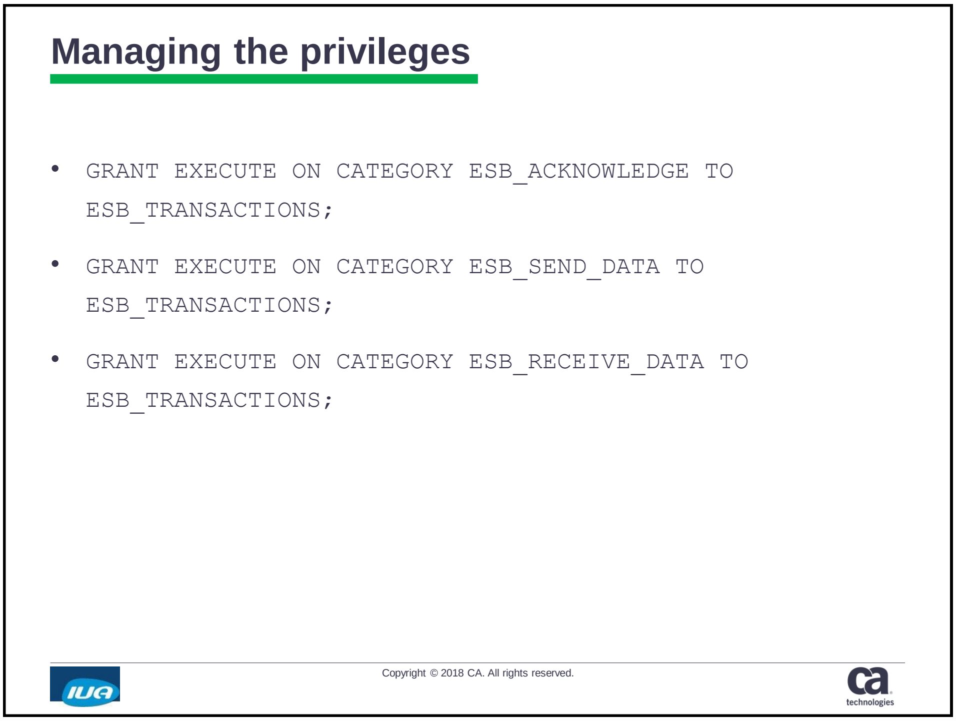
- Access Modules are a specific type of load module
- Only created for programs that have embedded SQL commands
- Managed using a RESOURCE CATEGORY
  - One of many SYSTEM RESOURCEs
  - Resource names are unique / cannot exist in more than one resource category
  - Changes to resource categories usually require a security system refresh – DCMT command or cycle the system



Managing the privileges

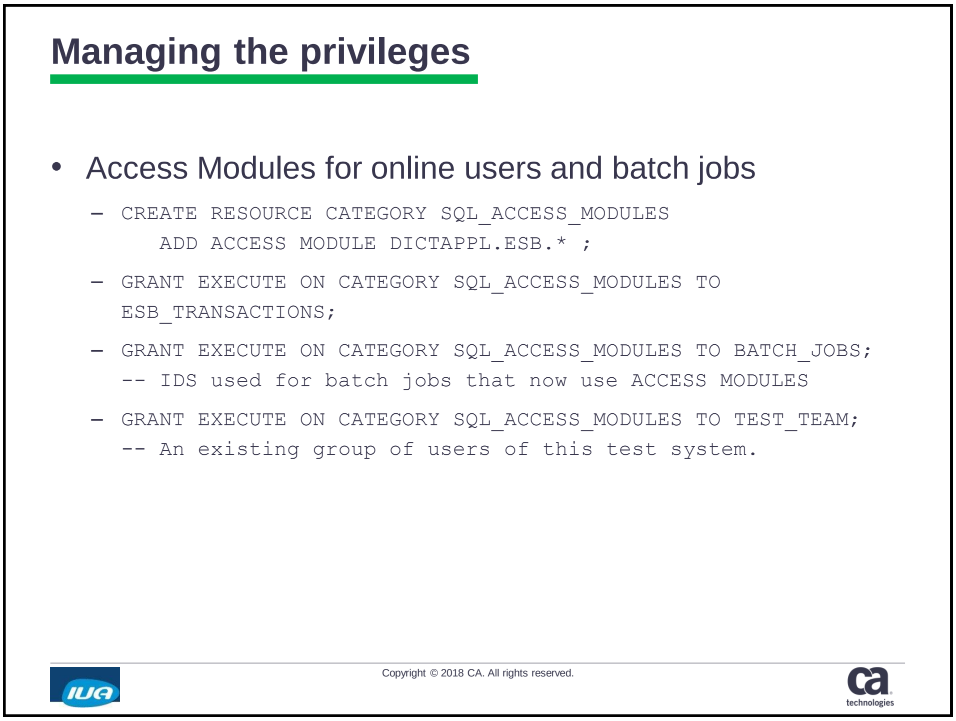
- Create the Resources

```
– CONNECT TO SYSTEM;
– CREATE RESOURCE CATEGORY ESB_ACKNOWLEDGE
                --   DICTIONARY.SCHEMA.ACCESS-MODULE
         ADD ACCESS MODULE DICTAPPL.ESB.ESBDACKD
         ADD ACCESS MODULE DICTAPPL.ESB.ESBP0006
         ADD ACCESS MODULE DICTAPPL.ESB.ESBP0206;
– CREATE RESOURCE CATEGORY ESB_SEND_DATA
         ADD ACCESS MODULE DICTAPPL.ESB.ESBCXML0;
– CREATE RESOURCE CATEGORY ESB_RECEIVE_DATA
         ADD ACCESS MODULE DICTAPPL.ESB.ESBP0014;
```



Managing the privileges

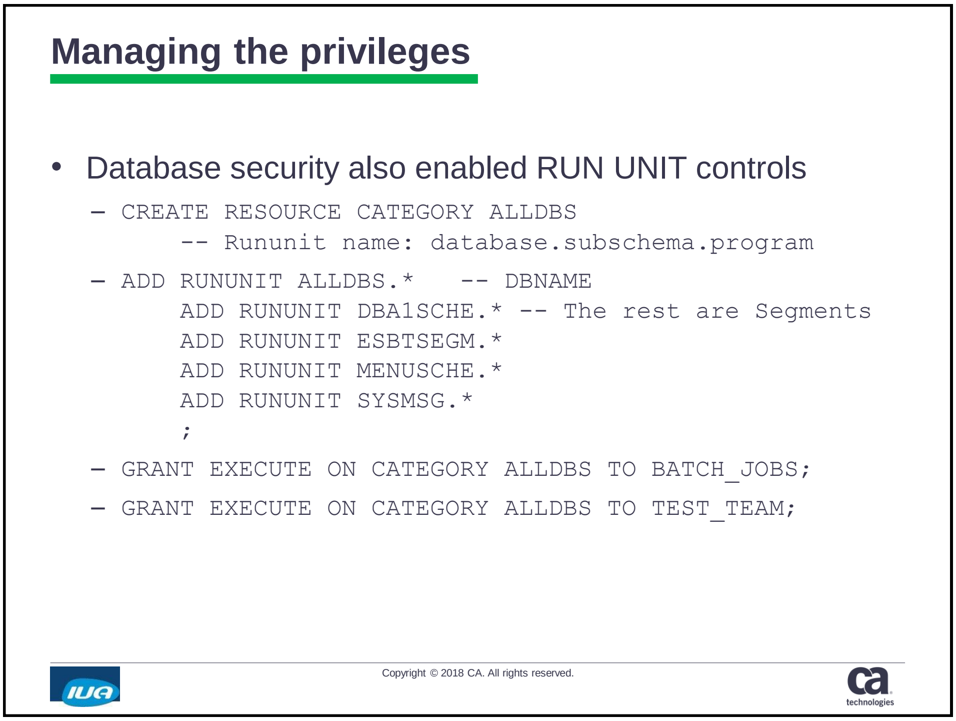
```
• GRANT EXECUTE ON CATEGORY ESB_ACKNOWLEDGE TO
  ESB_TRANSACTIONS;

• GRANT EXECUTE ON CATEGORY ESB_SEND_DATA TO
  ESB_TRANSACTIONS;

• GRANT EXECUTE ON CATEGORY ESB_RECEIVE_DATA TO
  ESB_TRANSACTIONS;
```



Managing the privileges

- Access Modules for online users and batch jobs

```
– CREATE RESOURCE CATEGORY SQL_ACCESS_MODULES
     ADD ACCESS MODULE DICTAPPL.ESB.* ;
– GRANT EXECUTE ON CATEGORY SQL_ACCESS_MODULES TO
  ESB_TRANSACTIONS;
– GRANT EXECUTE ON CATEGORY SQL_ACCESS_MODULES TO BATCH_JOBS;
  -- IDS used for batch jobs that now use ACCESS MODULES
– GRANT EXECUTE ON CATEGORY SQL_ACCESS_MODULES TO TEST_TEAM;
  -- An existing group of users of this test system.
```

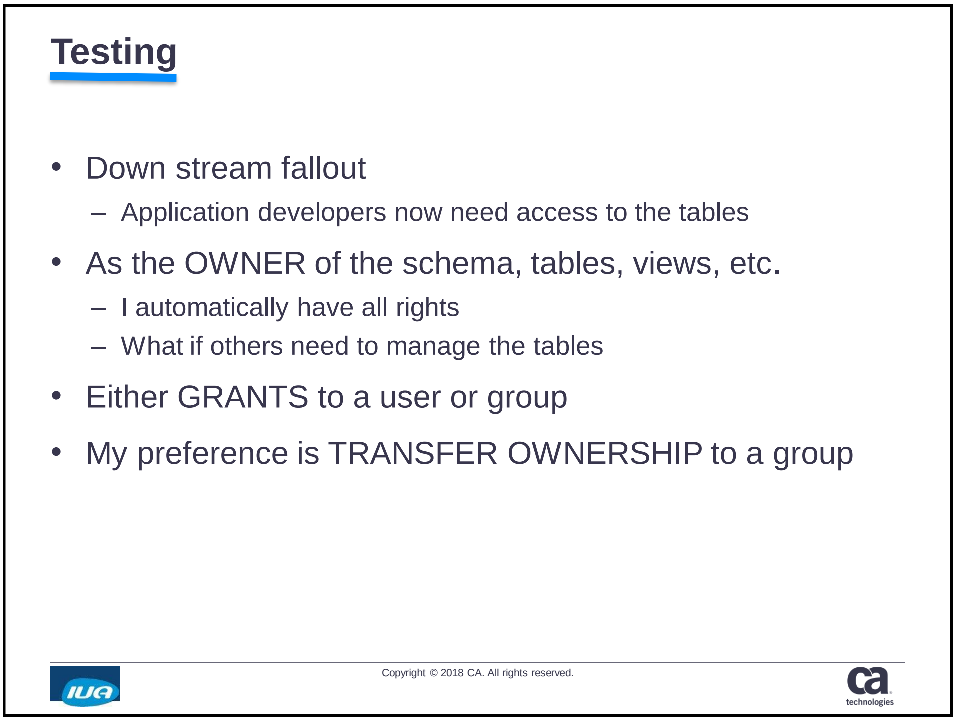


Managing the privileges

- Database security also enabled RUN UNIT controls

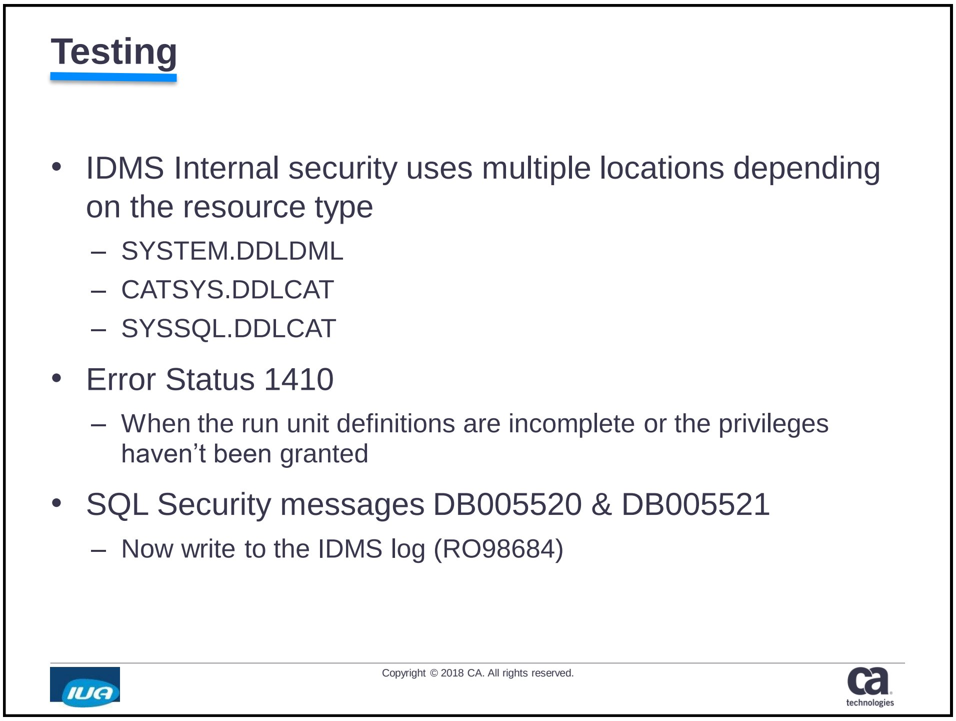
```
– CREATE RESOURCE CATEGORY ALLDBS
      -- Rununit name: database.subschema.program
– ADD RUNUNIT ALLDBS.*   -- DBNAME
      ADD RUNUNIT DBA1SCHE.* -- The rest are Segments
      ADD RUNUNIT ESBTSEGM.*
      ADD RUNUNIT MENUSCHE.*
      ADD RUNUNIT SYSMSG.*
      ;
– GRANT EXECUTE ON CATEGORY ALLDBS TO BATCH_JOBS;
– GRANT EXECUTE ON CATEGORY ALLDBS TO TEST_TEAM;
```

Testing



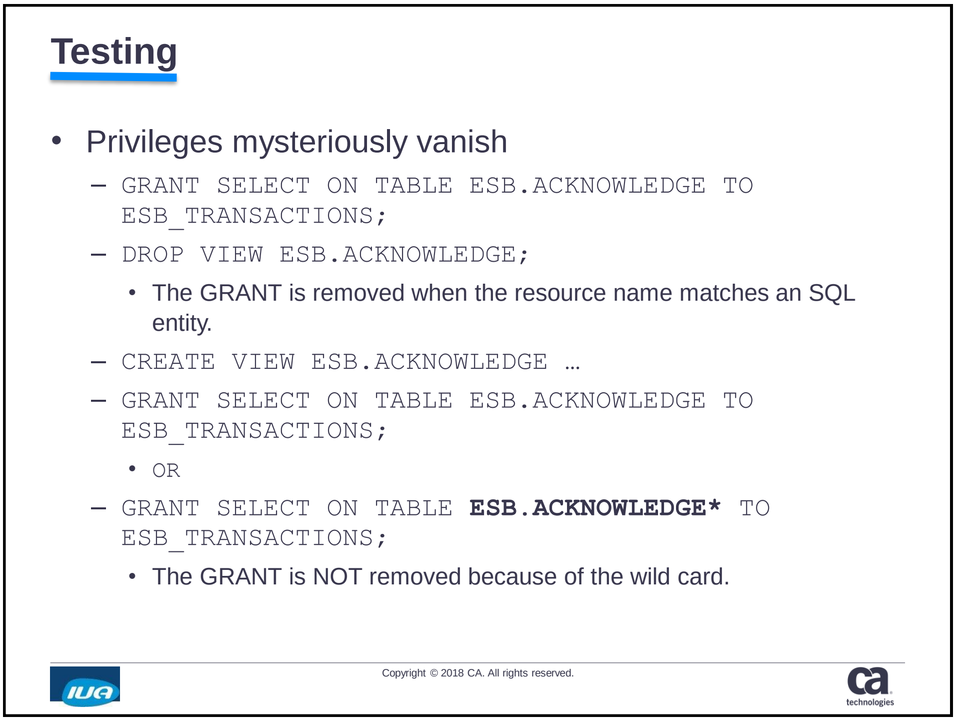
Testing

- Down stream fallout
  - Application developers now need access to the tables
- As the OWNER of the schema, tables, views, etc.
  - I automatically have all rights
  - What if others need to manage the tables
- Either GRANTS to a user or group
- My preference is TRANSFER OWNERSHIP to a group



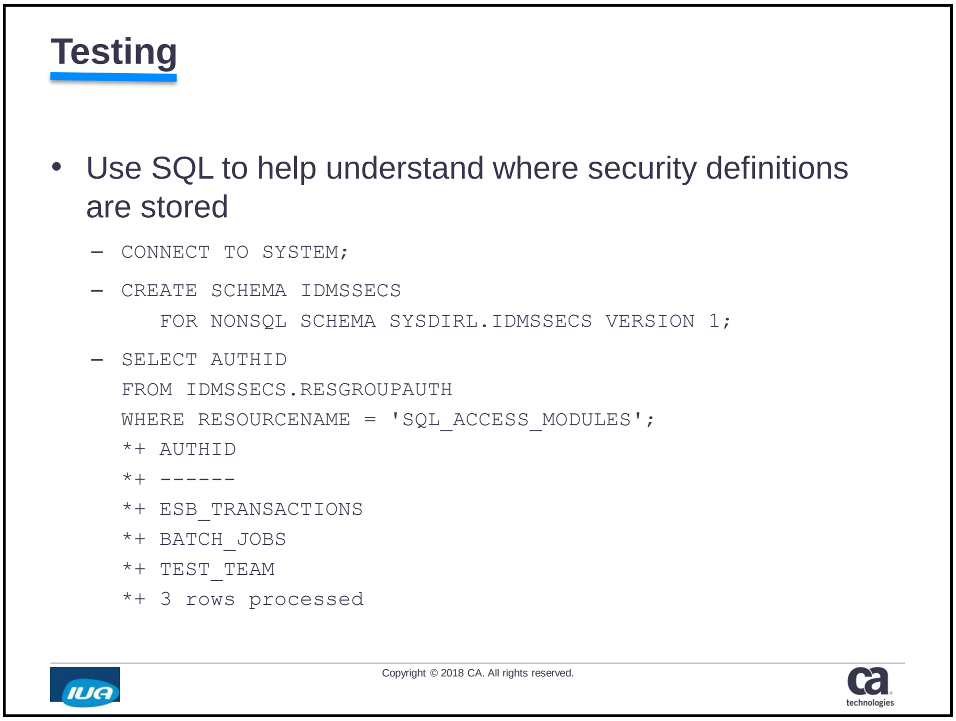
Testing

- IDMS Internal security uses multiple locations depending on the resource type
  - SYSTEM.DDLDML
  - CATSYS.DDLCAT
  - SYSSQL.DDLCAT
- Error Status 1410
  - When the run unit definitions are incomplete or the privileges haven’t been granted
- SQL Security messages DB005520 & DB005521
  - Now write to the IDMS log (RO98684)



Testing

- Privileges mysteriously vanish
  - GRANT SELECT ON TABLE ESB.ACKNOWLEDGE TO ESB_TRANSACTIONS;
  - DROP VIEW ESB.ACKNOWLEDGE;
    - The GRANT is removed when the resource name matches an SQL entity.
  - CREATE VIEW ESB.ACKNOWLEDGE …
  - GRANT SELECT ON TABLE ESB.ACKNOWLEDGE TO ESB_TRANSACTIONS;
    - OR
  - GRANT SELECT ON TABLE **ESB.ACKNOWLEDGE*** TO ESB_TRANSACTIONS;
    - The GRANT is NOT removed because of the wild card.



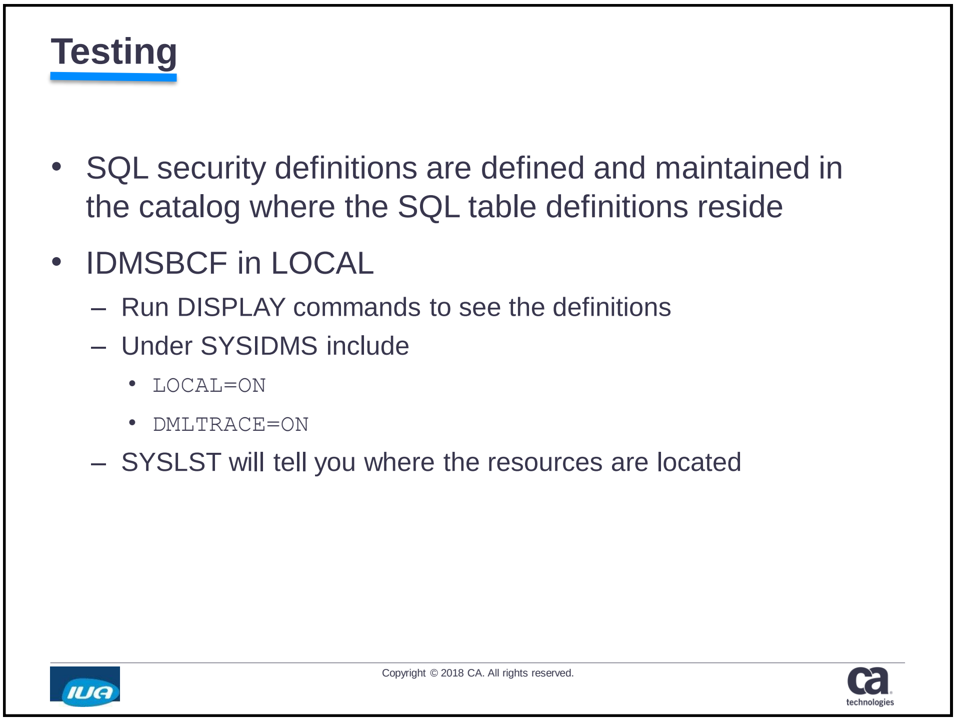
Testing

- Use SQL to help understand where security definitions are stored

  -
    ```
    CONNECT TO SYSTEM;
    ```

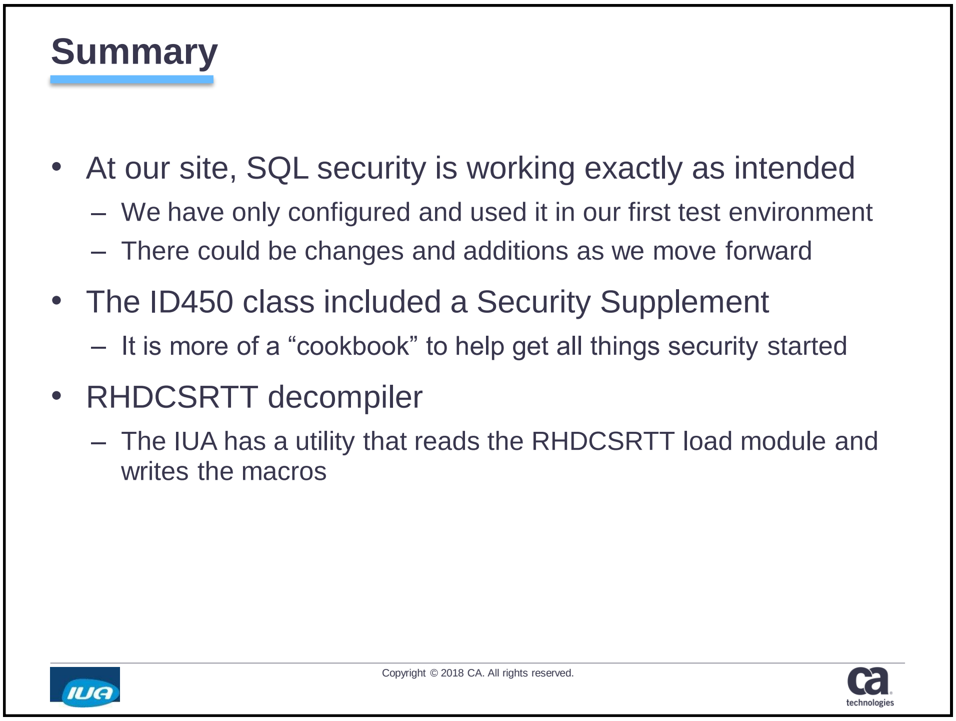
  -
    ```
    CREATE SCHEMA IDMSSECS
       FOR NONSQL SCHEMA SYSDIRL.IDMSSECS VERSION 1;
    ```

  -
    ```
    SELECT AUTHID
    FROM IDMSSECS.RESGROUPAUTH
    WHERE RESOURCENAME = 'SQL_ACCESS_MODULES';
    *+ AUTHID
    *+ ------
    *+ ESB_TRANSACTIONS
    *+ BATCH_JOBS
    *+ TEST_TEAM
    *+ 3 rows processed
    ```



Testing

- SQL security definitions are defined and maintained in the catalog where the SQL table definitions reside
- IDMSBCF in LOCAL
  - Run DISPLAY commands to see the definitions
  - Under SYSIDMS include
    - `LOCAL=ON`
    - `DMLTRACE=ON`
  - SYSLST will tell you where the resources are located



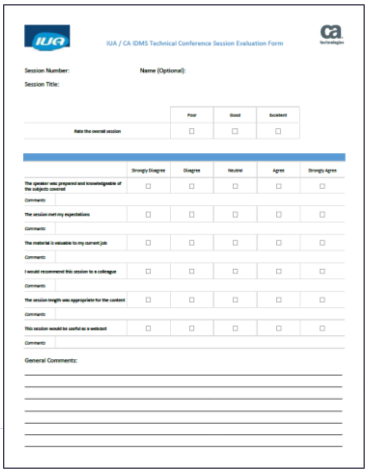
Summary

- At our site, SQL security is working exactly as intended
  - We have only configured and used it in our first test environment
  - There could be changes and additions as we move forward
- The ID450 class included a Security Supplement
  - It is more of a “cookbook” to help get all things security started
- RHDCSRTT decompiler
  - The IUA has a utility that reads the RHDCSRTT load module and writes the macros

Questions & Answers

Please Complete a Session Evaluation Form

- The number for this session is **A08**
- After completing your session evaluation form, place it in the envelope at the front of the room



IUA / CA IDMS Technical Conference Session Evaluation Form

Session Number: Name (Optional):

Session Title:

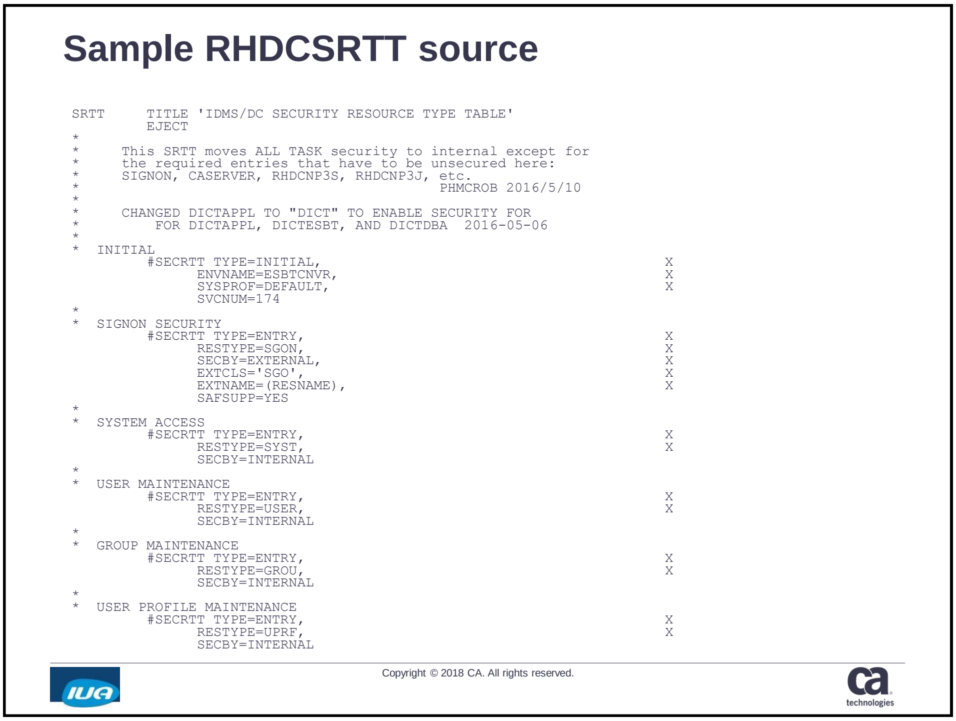
	Poor	Good	Excellent
Rate the overall session	□	□	□

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The speaker was prepared and knowledgeable of the subjects covered	□	□	□	□	□
Comments					
The session met my expectations	□	□	□	□	□
Comments					
The material is valuable to my current job	□	□	□	□	□
Comments					
I would recommend this session to a colleague	□	□	□	□	□
Comments					
The session length was appropriate for the content	□	□	□	□	□
Comments					
This session would be useful as a webcast	□	□	□	□	□
Comments					

General Comments:

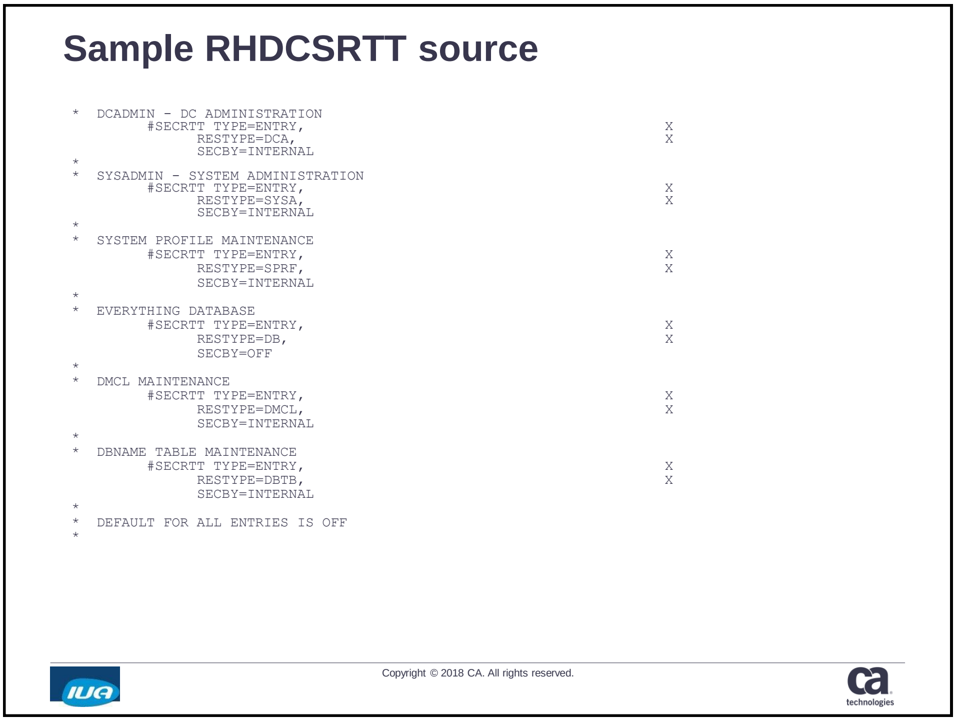
Appendix

Sample RHDCSRTT source



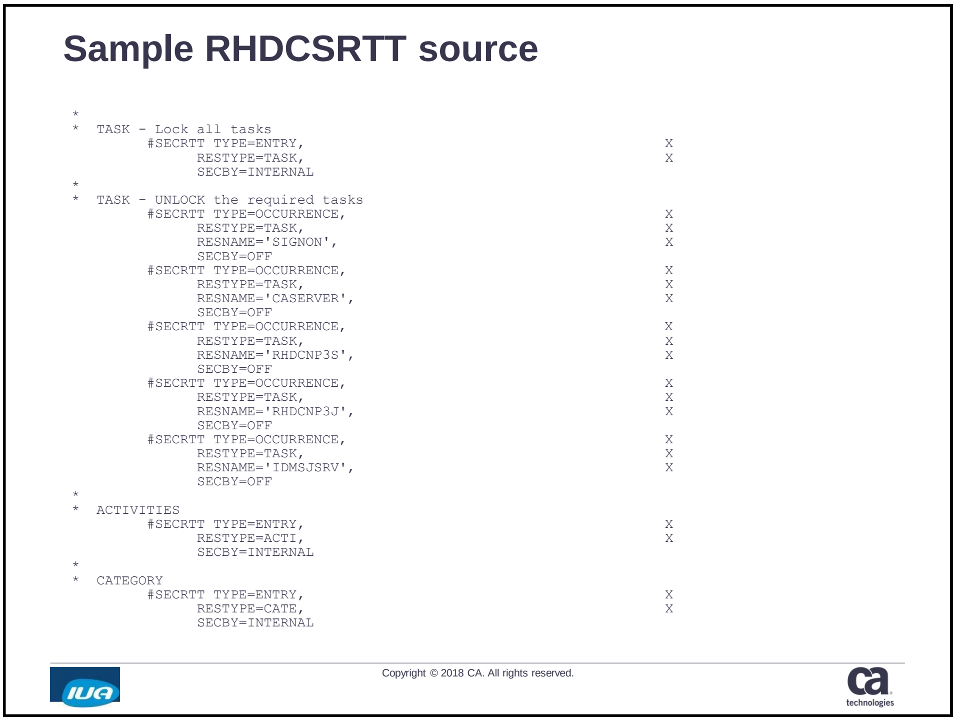
Sample RHDCSRTT source

```
SRTT     TITLE 'IDMS/DC SECURITY RESOURCE TYPE TABLE'
         EJECT
*
*     This SRTT moves ALL TASK security to internal except for
*     the required entries that have to be unsecured here:
*     SIGNON, CASERVER, RHDCNP3S, RHDCNP3J, etc.
*                                           PHMCROB 2016/5/10
*
*     CHANGED DICTAPPL TO "DICT" TO ENABLE SECURITY FOR
*         FOR DICTAPPL, DICTESBT, AND DICTDBA  2016-05-06
*
*  INITIAL
         #SECRTT TYPE=INITIAL,                                         X
               ENVNAME=ESBTCNVR,                                       X
               SYSPROF=DEFAULT,                                        X
               SVCNUM=174
*
*  SIGNON SECURITY
         #SECRTT TYPE=ENTRY,                                           X
               RESTYPE=SGON,                                           X
               SECBY=EXTERNAL,                                         X
               EXTCLS='SGO',                                           X
               EXTNAME=(RESNAME),                                      X
               SAFSUPP=YES
*
*  SYSTEM ACCESS
         #SECRTT TYPE=ENTRY,                                           X
               RESTYPE=SYST,                                           X
               SECBY=INTERNAL
*
*  USER MAINTENANCE
         #SECRTT TYPE=ENTRY,                                           X
               RESTYPE=USER,                                           X
               SECBY=INTERNAL
*
*  GROUP MAINTENANCE
         #SECRTT TYPE=ENTRY,                                           X
               RESTYPE=GROU,                                           X
               SECBY=INTERNAL
*
*  USER PROFILE MAINTENANCE
         #SECRTT TYPE=ENTRY,                                           X
               RESTYPE=UPRF,                                           X
               SECBY=INTERNAL
```



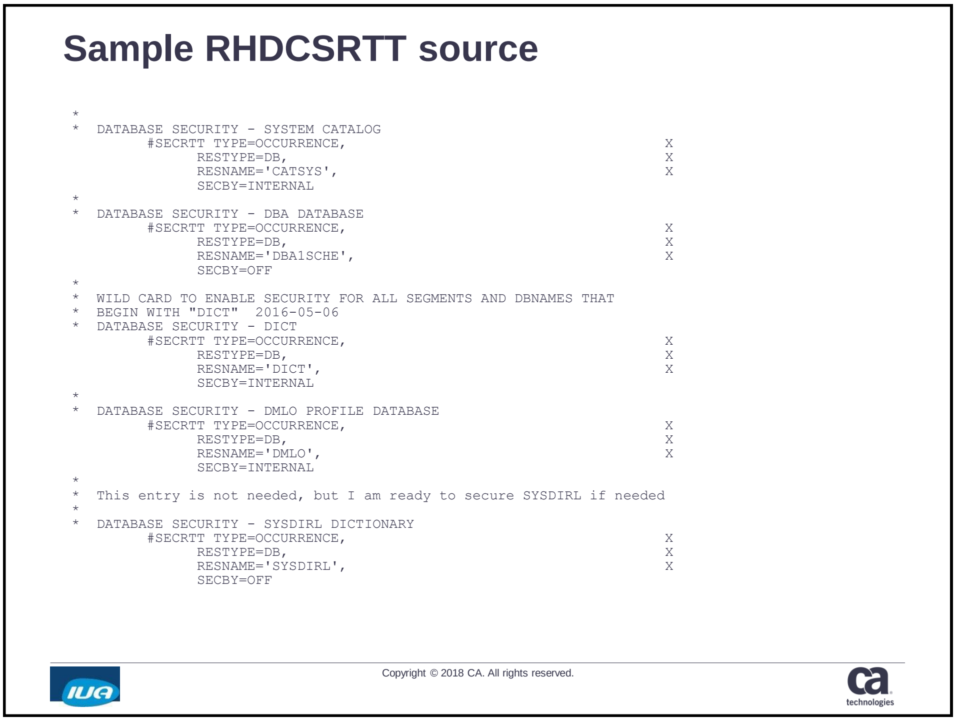
Sample RHDCSRTT source

```
*  DCADMIN - DC ADMINISTRATION
         #SECRTT TYPE=ENTRY,                                           X
               RESTYPE=DCA,                                            X
               SECBY=INTERNAL
*
*  SYSADMIN - SYSTEM ADMINISTRATION
         #SECRTT TYPE=ENTRY,                                           X
               RESTYPE=SYSA,                                           X
               SECBY=INTERNAL
*
*  SYSTEM PROFILE MAINTENANCE
         #SECRTT TYPE=ENTRY,                                           X
               RESTYPE=SPRF,                                           X
               SECBY=INTERNAL
*
*  EVERYTHING DATABASE
         #SECRTT TYPE=ENTRY,                                           X
               RESTYPE=DB,                                             X
               SECBY=OFF
*
*  DMCL MAINTENANCE
         #SECRTT TYPE=ENTRY,                                           X
               RESTYPE=DMCL,                                           X
               SECBY=INTERNAL
*
*  DBNAME TABLE MAINTENANCE
         #SECRTT TYPE=ENTRY,                                           X
               RESTYPE=DBTB,                                           X
               SECBY=INTERNAL
*
*  DEFAULT FOR ALL ENTRIES IS OFF
*
```



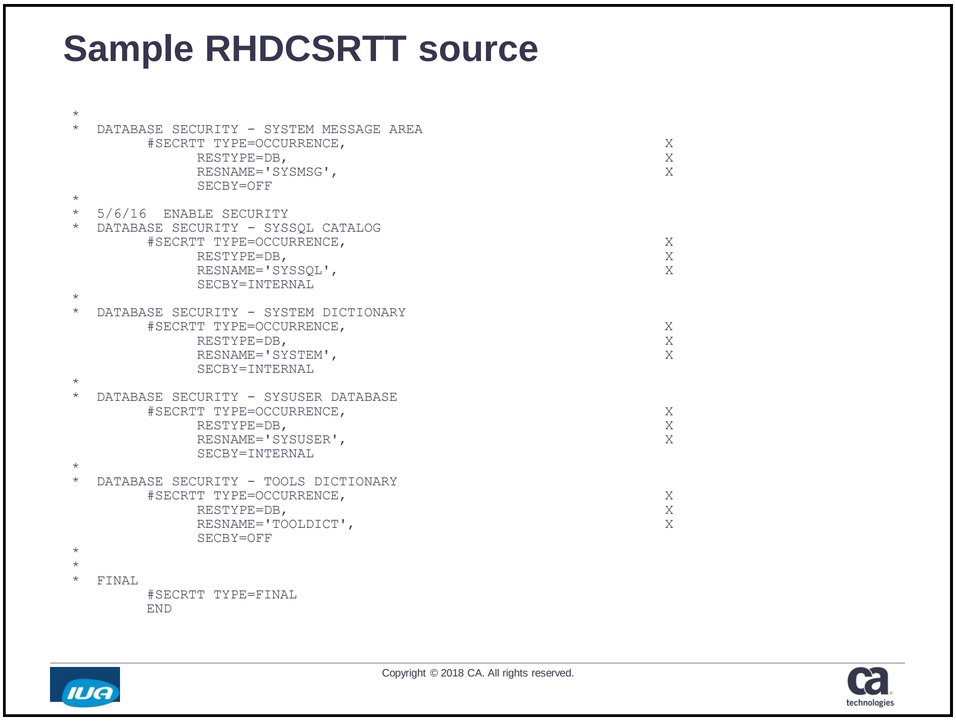
Sample RHDCSRTT source

```
*
*  TASK - Lock all tasks
         #SECRTT TYPE=ENTRY,                                           X
               RESTYPE=TASK,                                           X
               SECBY=INTERNAL
*
*  TASK - UNLOCK the required tasks
         #SECRTT TYPE=OCCURRENCE,                                      X
               RESTYPE=TASK,                                           X
               RESNAME='SIGNON',                                       X
               SECBY=OFF
         #SECRTT TYPE=OCCURRENCE,                                      X
               RESTYPE=TASK,                                           X
               RESNAME='CASERVER',                                     X
               SECBY=OFF
         #SECRTT TYPE=OCCURRENCE,                                      X
               RESTYPE=TASK,                                           X
               RESNAME='RHDCNP3S',                                     X
               SECBY=OFF
         #SECRTT TYPE=OCCURRENCE,                                      X
               RESTYPE=TASK,                                           X
               RESNAME='RHDCNP3J',                                     X
               SECBY=OFF
         #SECRTT TYPE=OCCURRENCE,                                      X
               RESTYPE=TASK,                                           X
               RESNAME='IDMSJSRV',                                     X
               SECBY=OFF
*
*  ACTIVITIES
         #SECRTT TYPE=ENTRY,                                           X
               RESTYPE=ACTI,                                           X
               SECBY=INTERNAL
*
*  CATEGORY
         #SECRTT TYPE=ENTRY,                                           X
               RESTYPE=CATE,                                           X
               SECBY=INTERNAL
```



Sample RHDCSRTT source

```
*
*  DATABASE SECURITY - SYSTEM CATALOG
         #SECRTT TYPE=OCCURRENCE,                                      X
               RESTYPE=DB,                                             X
               RESNAME='CATSYS',                                       X
               SECBY=INTERNAL
*
*  DATABASE SECURITY - DBA DATABASE
         #SECRTT TYPE=OCCURRENCE,                                      X
               RESTYPE=DB,                                             X
               RESNAME='DBA1SCHE',                                     X
               SECBY=OFF
*
*  WILD CARD TO ENABLE SECURITY FOR ALL SEGMENTS AND DBNAMES THAT
*  BEGIN WITH "DICT"  2016-05-06
*  DATABASE SECURITY - DICT
         #SECRTT TYPE=OCCURRENCE,                                      X
               RESTYPE=DB,                                             X
               RESNAME='DICT',                                         X
               SECBY=INTERNAL
*
*  DATABASE SECURITY - DMLO PROFILE DATABASE
         #SECRTT TYPE=OCCURRENCE,                                      X
               RESTYPE=DB,                                             X
               RESNAME='DMLO',                                         X
               SECBY=INTERNAL
*
*  This entry is not needed, but I am ready to secure SYSDIRL if needed
*
*  DATABASE SECURITY - SYSDIRL DICTIONARY
         #SECRTT TYPE=OCCURRENCE,                                      X
               RESTYPE=DB,                                             X
               RESNAME='SYSDIRL',                                      X
               SECBY=OFF
```



Sample RHDCSRTT source

```
*
*  DATABASE SECURITY - SYSTEM MESSAGE AREA
         #SECRTT TYPE=OCCURRENCE,                                      X
               RESTYPE=DB,                                             X
               RESNAME='SYSMSG',                                       X
               SECBY=OFF
*
*  5/6/16  ENABLE SECURITY
*  DATABASE SECURITY - SYSSQL CATALOG
         #SECRTT TYPE=OCCURRENCE,                                      X
               RESTYPE=DB,                                             X
               RESNAME='SYSSQL',                                       X
               SECBY=INTERNAL
*
*  DATABASE SECURITY - SYSTEM DICTIONARY
         #SECRTT TYPE=OCCURRENCE,                                      X
               RESTYPE=DB,                                             X
               RESNAME='SYSTEM',                                       X
               SECBY=INTERNAL
*
*  DATABASE SECURITY - SYSUSER DATABASE
         #SECRTT TYPE=OCCURRENCE,                                      X
               RESTYPE=DB,                                             X
               RESNAME='SYSUSER',                                      X
               SECBY=INTERNAL
*
*  DATABASE SECURITY - TOOLS DICTIONARY
         #SECRTT TYPE=OCCURRENCE,                                      X
               RESTYPE=DB,                                             X
               RESNAME='TOOLDICT',                                     X
               SECBY=OFF
*
*
*  FINAL
         #SECRTT TYPE=FINAL
         END
```