PSQuery Security Queries

PT 8.60
Randall Groncki

Introduction

Users have access to records in PSQuery through the Query Access Manager. Records are assigned to Query security trees. However, finding the bigger picture of what records a user has access to through many roles and permission lists is somewhat daunting. Just as difficult, questions about which roles and permission lists have access to a specific record can be a trip down the multiple rabbit holes.

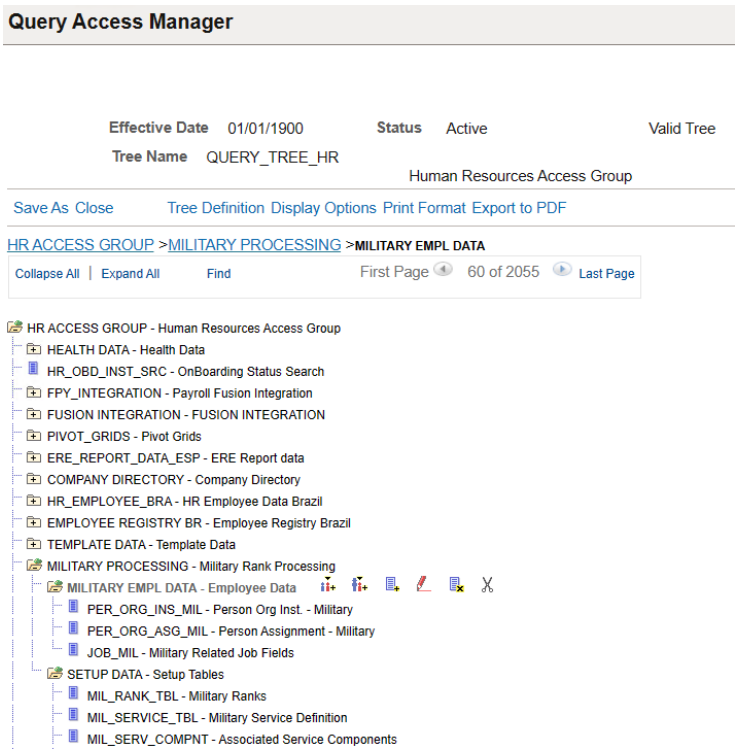
I've created a few PSQiueries and SQL statements to help clarify the current relationships between the Records, Roles and Users. Hopefully these can help you adjust security in your implementation so your PSQuery users have correct record access.

PSQuery Record Security Refresher

Query administrators assign records to one or more Query Security trees to allow and control access to the record. This is not row level security for that record, but just whether a user has access to a record at all.

Using the Query Access Manager, administrators assign records to one or more Query Security Trees. The records can appear on multiple trees and also appear on the same tree multiple times on different branches.

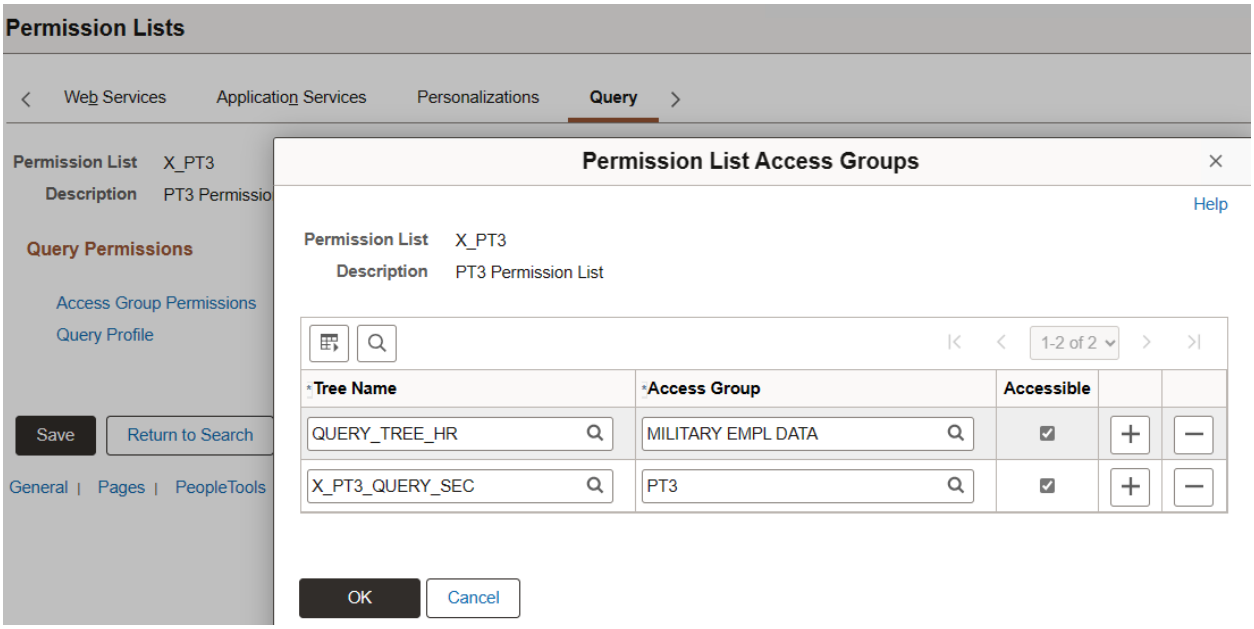
Navigation: Menu > PeopleTools > Security > Query Security > Query Access Manager.



Administrators then use permission lists to grant access to these records by granting access to a node on one or more Query Security Trees. Access can be granted at the root node which allows access to every record on that tree or they can grant access to a branch of that tree. Granting access to a tree branch enables records on every record and child branch of that tree.

Navigation: Menu > PeopleTools > Security > Permissions and Roles > Permission Lists (Query Tab)

Click on the “Access Group Permissions” link



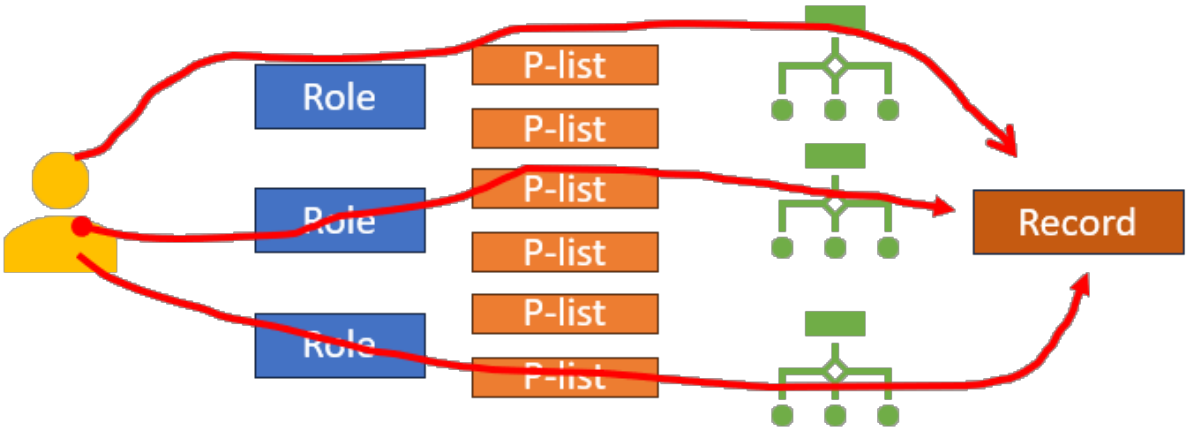
Unchecking the Accessible box denies access to the records in that tree branch to this permission list.

The permission list is assigned to a role and any user that has that role has access to these records in PSQuery.

Where it gets confusing…

A role can have many permission lists and users have multiple roles. On the other hand, records are assigned to trees and can appear on the same tree multiple times.

There is a many to many relationship between the user and the records with multiple roles, permission lists and trees granting access to a record, sometimes multiple times.



The same record can be granted to a user multiple times through different roles, permission lists and trees.

A better view of the situation.

Functional Users

Download the PeopleTools project from GitHub or PeopleToolsTechTips.com and have your developers load it to a demo or development environment. This project contains a new view and five prompted queries to show the relationships between PSQuery Records and tress, users, permission lists and roles.

Query Name	Description
X_PT3_QUERY_SEC_RECORDS_PL	What PList can see Query Rec
X_PT3_QUERY_SEC_PL_RECORDS	What records can PL see
X_PT3_QUERY_SEC_USER_RECORDS	What recs can a user see
X_PT3_QUERY_SEC_RECORDS_ROLES	What roles can see Query Rec
X_PT3_QUERY_SEC_RECORD_USERS	What users can see a Query Rec

Here is the records you will need access in order to use these Queries:

- X_PT3_QRYSEC_VW
- PSROLEUSER
- PSROLECLASS

Here is a sample result of X_PT3_QUERY_SEC_PL_RECORDS query looking at the custom X_PT3 permission list showing what trees, branches, records are granted to this permission list and the Tree Path of that record.

Permission List = X_PT3

View All | Rerun Query | Download to Excel | Download to XML

First 1-6 of 6 Last

Row	Permission List	Record	Rec Descr	Tree	Access Grp	Access	Path
1	X_PT3	PER_ORG_INS_MIL	Person Org Inst. - Military	QUERY_TREE_HR	MILITARY EMPL DATA	Y	MILITARY EMPL DATA -> PER_ORG_INS_MIL
2	X_PT3	JOB_MIL	Military Related Job Fields	QUERY_TREE_HR	MILITARY EMPL DATA	Y	MILITARY EMPL DATA -> JOB_MIL
3	X_PT3	PER_ORG_ASG_MIL	Person Assignment - Military	QUERY_TREE_HR	MILITARY EMPL DATA	Y	MILITARY EMPL DATA -> PER_ORG_ASG_MIL
4	X_PT3	PSROLEUSER	Role User	X_PT3_QUERY_SEC	PT3	Y	PT3 -> PSROLEUSER
5	X_PT3	PSROLECLASS	Role Classes	X_PT3_QUERY_SEC	PT3	Y	PT3 -> PSROLECLASS
6	X_PT3	X_PT3_QRYSEC_VW	PT3 Query Security View	X_PT3_QUERY_SEC	PT3	Y	PT3 -> X_PT3_QRYSEC_VW

Technical users and Developers

These three tables show you the relationship between a permission list and a record.

Record	Description
PS_SCRTY_ACC_GRP	This is the child record in the permission list that connects a permission list to a specific tree and node
PSTREENODE	The detailed definition of the trees. This is the actual records and branches. Each row of this table defines the record name or the branch name and it's immediate parent node
PSTREEDEFN	Top level tree definition. This record will provide the correct effective date and whether the tree is active.

The work is done on the PSTREENODE table by using a revers SYS_CONNECT_BY_PATH in the SQL to show the path from the target record on the tree up through the hierarchy to the root node.

The core is a view the connects the path of the record up through the parent tree nodes using the Oracle SYS_CONNECT_BY_PATH SQL function. This may not work in DB2 or other PeopleSoft supported databases – you may have to adjust your code if you are not using Oracle DB.

```
SELECT CONNECT_BY_ROOT P.TREE_NODE AS RECNAME,
   P.TREE_NAME,
   P.EFFDT,
   P.TREE_NODE_TYPE ,
   P.SETID ,
   REVERSE(LTRIM(SYS_CONNECT_BY_PATH (REVERSE(P.TREE_NODE),' >- '),' >- ')) AS TREE_PATH ,
   P.TREE_NODE    AS ACCESS_PATH
  FROM PSTREENODE P
   CONNECT BY PRIOR P.PARENT_NODE_NUM = P.TREE_NODE_NUM
  AND PRIOR P.TREE_NAME             = P.TREE_NAME
  AND PRIOR P.EFFDT                 = P.EFFDT
  AND PRIOR P.SETCNTRLVALUE         = P.SETCNTRLVALUE
```

Tying this view in with the other records along with PSROLECLASS (Permission lists in a role) AND PSROLEUSER (Users granted the role) provides a view of all the paths a user has access to a record.

What Query Records can a User see?

```
WITH TREE_NODE AS
                    (SELECT CONNECT_BY_ROOT P.TREE_NODE AS RECNAME,
                     P.TREE_NAME,
                     P.EFFDT,
                     P.TREE_NODE_TYPE ,
                     P.SETID ,
                     REVERSE(LTRIM(SYS_CONNECT_BY_PATH (REVERSE(P.TREE_NODE),' >- '),' >- ')) AS
TREE_PATH ,
                     P.TREE_NODE    AS ACCESS_PATH
                    FROM PSTREENODE P
                     CONNECT BY PRIOR P.PARENT_NODE_NUM = P.TREE_NODE_NUM
                    AND PRIOR P.TREE_NAME             = P.TREE_NAME
                    AND PRIOR P.EFFDT                 = P.EFFDT
                    AND PRIOR P.SETCNTRLVALUE         = P.SETCNTRLVALUE   )
SELECT DISTINCT RU.ROLEUSER,
 B.RECNAME ,
 RD.RECDESCR,
 ru.rolename,
 C.CLASSID,
 C.TREE_NAME,
 a.effdt,
 C.ACCESS_GROUP,
 C.ACCESSIBLE,
 B.TREE_PATH
FROM PS_SCRTY_ACC_GRP C ,
 TREE_NODE B ,
 PSTREEDEFN A ,
 PSRECDEFN RD,
 PSROLECLASS RC,
 PSROLEUSER RU
WHERE C.TREE_NAME   = B.TREE_NAME
AND C.ACCESS_GROUP  = B.ACCESS_PATH
AND C.TREE_NAME     = A.TREE_NAME
AND B.SETID         = ' '
AND A.SETID         = B.SETID
AND A.TREE_STRCT_ID = 'ACCESS_GROUP'
AND A.EFF_STATUS    = 'A'
AND A.EFFDT         =   (SELECT MAX(D.EFFDT)  FROM PSTREEDEFN D
                 WHERE D.SETID   = ' '
                 AND D.TREE_NAME = A.TREE_NAME
                 AND D.EFFDT    <= sysdate   )
AND B.TREE_NODE_TYPE = 'G'
AND RD.RECNAME       = B.RECNAME
AND RC.CLASSID = C.CLASSID
AND RU.ROLENAME = RC.ROLENAME
AND RU.ROLEUSER      = 'PT3';
```

What Query Records can a Permission List See?

```
WITH TREE_NODE AS
                (SELECT CONNECT_BY_ROOT P.TREE_NODE AS RECNAME,
                  P.TREE_NAME,
                  P.EFFDT,
                  P.TREE_NODE_TYPE ,
                  P.SETID ,
                  REVERSE(LTRIM(SYS_CONNECT_BY_PATH (REVERSE(P.TREE_NODE),' >- '),' >- ')) AS
               TREE_PATH ,
                  P.TREE_NODE    AS ACCESS_PATH
                FROM PSTREENODE P
                  CONNECT BY PRIOR P.PARENT_NODE_NUM = P.TREE_NODE_NUM
                AND PRIOR P.TREE_NAME             = P.TREE_NAME
                AND PRIOR P.EFFDT                 = P.EFFDT
                AND PRIOR P.SETCNTRLVALUE         = P.SETCNTRLVALUE   )
SELECT DISTINCT B.RECNAME ,
  RD.RECDESCR,
  C.CLASSID,
  C.TREE_NAME,
  a.effdt,
  C.ACCESS_GROUP,
  C.ACCESSIBLE,
  B.TREE_PATH
FROM PS_SCRTY_ACC_GRP C ,
  TREE_NODE B ,
  PSTREEDEFN A ,
  PSRECDEFN RD
WHERE C.TREE_NAME   = B.TREE_NAME
AND C.ACCESS_GROUP  = B.ACCESS_PATH
AND C.TREE_NAME     = A.TREE_NAME
AND B.SETID         = ' '
AND A.SETID         = B.SETID
AND A.TREE_STRCT_ID = 'ACCESS_GROUP'
AND A.EFF_STATUS    = 'A'
AND A.EFFDT         =   (SELECT MAX(D.EFFDT)  FROM PSTREEDEFN D
                  WHERE D.SETID   = ' '
                  AND D.TREE_NAME = A.TREE_NAME
                  AND D.EFFDT    <= sysdate   )
AND B.TREE_NODE_TYPE = 'G'
AND RD.RECNAME       = B.RECNAME
AND C.CLASSID        = 'X_PT3'
```

What users can see a Query Record?

```
WITH TREE_NODE AS
                (SELECT CONNECT_BY_ROOT P.TREE_NODE AS RECNAME,
                  P.TREE_NAME,
                  P.EFFDT,
                  P.TREE_NODE_TYPE ,
                  P.SETID ,
                  REVERSE(LTRIM(SYS_CONNECT_BY_PATH (REVERSE(P.TREE_NODE),' >- '),' >- ')) AS
               TREE_PATH ,
                  P.TREE_NODE    AS ACCESS_PATH
                FROM PSTREENODE P
                  CONNECT BY PRIOR P.PARENT_NODE_NUM = P.TREE_NODE_NUM
                AND PRIOR P.TREE_NAME             = P.TREE_NAME
                AND PRIOR P.EFFDT                 = P.EFFDT
                AND PRIOR P.SETCNTRLVALUE         = P.SETCNTRLVALUE   )
SELECT DISTINCT RU.ROLEUSER,
  B.RECNAME ,
  RD.RECDESCR,
 -- C.CLASSID,
  C.TREE_NAME,
  a.effdt,
  C.ACCESS_GROUP,
  C.ACCESSIBLE,
  B.TREE_PATH
FROM PS_SCRTY_ACC_GRP C ,
  TREE_NODE B ,
  PSTREEDEFN A ,
  PSRECDEFN RD,
  PSROLECLASS RC,
  PSROLEUSER RU
WHERE C.TREE_NAME   = B.TREE_NAME
AND C.ACCESS_GROUP  = B.ACCESS_PATH
AND C.TREE_NAME     = A.TREE_NAME
AND B.SETID         = ' '
AND A.SETID         = B.SETID
AND A.TREE_STRCT_ID = 'ACCESS_GROUP'
AND A.EFF_STATUS    = 'A'
AND A.EFFDT         =   (SELECT MAX(D.EFFDT)  FROM PSTREEDEFN D
                  WHERE D.SETID   = ' '
                  AND D.TREE_NAME = A.TREE_NAME
                  AND D.EFFDT    <= sysdate   )
AND B.TREE_NODE_TYPE = 'G'
AND RD.RECNAME       = B.RECNAME
AND RC.CLASSID = C.CLASSID
AND RU.ROLENAME = RC.ROLENAME
AND B.RECNAME       = 'MU_CHANGE_TBL';
```

What Roles can a see a Query Record?

```
WITH TREE_NODE AS
                (SELECT CONNECT_BY_ROOT P.TREE_NODE AS RECNAME,
                  P.TREE_NAME,
                  P.EFFDT,
                  P.TREE_NODE_TYPE ,
                  P.SETID ,
                  REVERSE(LTRIM(SYS_CONNECT_BY_PATH (REVERSE(P.TREE_NODE),' >- '),' >- ')) AS
               TREE_PATH ,
                  P.TREE_NODE    AS ACCESS_PATH
                 FROM PSTREENODE P
                  CONNECT BY PRIOR P.PARENT_NODE_NUM = P.TREE_NODE_NUM
                 AND PRIOR P.TREE_NAME             = P.TREE_NAME
                 AND PRIOR P.EFFDT                 = P.EFFDT
                 AND PRIOR P.SETCNTRLVALUE         = P.SETCNTRLVALUE   )
SELECT DISTINCT B.RECNAME ,
  RD.RECDESCR,
  C.CLASSID,
  C.TREE_NAME,
  a.effdt,
  C.ACCESS_GROUP,
  C.ACCESSIBLE,
  B.TREE_PATH
FROM PS_SCRTY_ACC_GRP C ,
  TREE_NODE B ,
  PSTREEDEFN A ,
  PSRECDEFN RD
WHERE C.TREE_NAME   = B.TREE_NAME
AND C.ACCESS_GROUP  = B.ACCESS_PATH
AND C.TREE_NAME     = A.TREE_NAME
AND B.SETID         = ' '
AND A.SETID         = B.SETID
AND A.TREE_STRCT_ID = 'ACCESS_GROUP'
AND A.EFF_STATUS    = 'A'
AND A.EFFDT         =   (SELECT MAX(D.EFFDT)  FROM PSTREEDEFN D
                 WHERE D.SETID   = ' '
                 AND D.TREE_NAME = A.TREE_NAME
                 AND D.EFFDT    <= sysdate   )
AND B.TREE_NODE_TYPE = 'G'
AND RD.RECNAME       = B.RECNAME
AND B.RECNAME        = 'MU_CHANGE_TBL';
```

What Permission Lists can see a Query Record?

```
WITH TREE_NODE AS
                (SELECT CONNECT_BY_ROOT P.TREE_NODE AS RECNAME,
                  P.TREE_NAME,
                  P.EFFDT,
                  P.TREE_NODE_TYPE ,
                  P.SETID ,
                  REVERSE(LTRIM(SYS_CONNECT_BY_PATH (REVERSE(P.TREE_NODE),' >- '),' >- ')) AS
               TREE_PATH ,
                  P.TREE_NODE    AS ACCESS_PATH
                FROM PSTREENODE P
                 CONNECT BY PRIOR P.PARENT_NODE_NUM = P.TREE_NODE_NUM
                AND PRIOR P.TREE_NAME             = P.TREE_NAME
                AND PRIOR P.EFFDT                 = P.EFFDT
                AND PRIOR P.SETCNTRLVALUE         = P.SETCNTRLVALUE   )
SELECT DISTINCT B.RECNAME ,
  RD.RECDESCR,
  C.CLASSID,
  C.TREE_NAME,
  a.effdt,
  C.ACCESS_GROUP,
  C.ACCESSIBLE,
  B.TREE_PATH
FROM PS_SCRTY_ACC_GRP C ,
  TREE_NODE B ,
  PSTREEDEFN A ,
  PSRECDEFN RD
WHERE C.TREE_NAME   = B.TREE_NAME
AND C.ACCESS_GROUP  = B.ACCESS_PATH
AND C.TREE_NAME     = A.TREE_NAME
AND B.SETID         = ' '
AND A.SETID         = B.SETID
AND A.TREE_STRCT_ID = 'ACCESS_GROUP'
AND A.EFF_STATUS    = 'A'
AND A.EFFDT         =   (SELECT MAX(D.EFFDT)  FROM PSTREEDEFN D
                WHERE D.SETID   = ' '
                AND D.TREE_NAME = A.TREE_NAME
                AND D.EFFDT    <= sysdate   )
AND B.TREE_NODE_TYPE = 'G'
AND RD.RECNAME       = B.RECNAME
AND B.RECNAME        = 'MU_CHANGE_TBL';
```