



pears.NetOwnerContactEvents



Generated schema reference. Regenerate this page from the SQL unload; keep hand-maintained business notes in the narrative namespace.

Original SQL

```
CREATE PROCEDURE "pears"."NetOwnerContactEvents"( IN "pWebUserID" CHAR(20),
  IN "pSlice" INTEGER DEFAULT 0,
  IN "pSliceSize" INTEGER DEFAULT 20,
  IN "pStaffID" CHAR(20) DEFAULT NULL,
  IN "pEmploymentID" CHAR(20) DEFAULT NULL,
  IN "pPersonID" CHAR(20) DEFAULT NULL,
  IN "pVacancyID" CHAR(20) DEFAULT NULL,
  IN "pCompanyID" CHAR(20) DEFAULT NULL,
  IN "pAllEvents" CHAR(1) DEFAULT NULL,
  IN "pFilterDescription" CHAR(100) DEFAULT '%',
  IN "pFilterTimeStart" DATE,
  IN "pFilterTimeEnd" DATE,
  IN "pFilterClassType" CHAR(50) DEFAULT '%',
  IN "pFilterPriority" CHAR(1) DEFAULT '%',
  IN "pFilterOutcome" long VARCHAR DEFAULT '%',
  IN "pFilterNotes" long VARCHAR DEFAULT '%',
  IN "pFilterPersonName" CHAR(50) DEFAULT '%',
  IN "pFilterContactName" CHAR(60) DEFAULT '%',
  IN "pFilterStaffName" CHAR(60) DEFAULT '%',
  IN "pFilterPosition" CHAR(60) DEFAULT '%',
  IN "pFilterCompanyName" CHAR(60) DEFAULT '%',
  IN "pOrderBy" CHAR(255) DEFAULT '' )
RESULT(
  "ContactEventID" CHAR(20),
  "Description" CHAR(100),
  "ContactEventTime" "datetime",
  "ClassType" CHAR(50),
  "Priority" CHAR(1),
  "Outcome" long VARCHAR,
  "Notes" long VARCHAR,
  "PersonName" CHAR(50),
  "ContactName" CHAR(60),
  "StaffName" CHAR(60),
  "Position" CHAR(60),
  "CompanyName" CHAR(60) )
```



```
BEGIN
  // IQXNet
  DECLARE "x" INTEGER;
  DECLARE "y" INTEGER;
  SET "x" = "pSliceSize";
  SET "y" = "x"*"pSlice"+1;
  CASE WHEN "pCompanyID" IS NULL AND "pPersonID" IS NULL AND "pEmploymentID"
IS NULL AND "pVacancyID" IS NULL THEN RETURN
  WHEN "pPersonID" IS NOT NULL THEN /* Contact events for a Person record*/
    SELECT top "x" START at "y"
      "ce"."contacteventid",
      "ce"."description",
      "datetime"("ce"."contactdate"+"ce"."contacttime") AS
"ContactEventTime",
      "c"."classdescrip" AS "ClassType",
      "isnull"("ce"."priority",1) AS "Priority",
      "ce"."outcome",
      "ce"."notes" AS "Notes",
      "p"."name" AS "PersonName",
      "cp"."name" AS "ContactName",
      "s"."name" AS "StaffName",
      "v"."position" AS "Position",
      "co"."name" AS "CompanyName"
    FROM "contactevent" AS "ce" KEY JOIN "staff" AS "s"
      , "contactevent" AS "ce" KEY JOIN "contactclass" AS "c"
      , "contactevent" AS "ce" KEY JOIN "person" AS "p"
      , "contactevent" AS "ce" KEY LEFT OUTER JOIN("employment" AS "e" KEY
JOIN("company" AS "co", "person" AS "cp"))
      , "contactevent" AS "ce" KEY LEFT OUTER JOIN("vacancy" AS "v")
      , "contactevent" AS "ce" LEFT OUTER JOIN(SELECT "DivisionID" FROM
"DashboardStaffDivisions"()) AS "d" ON("ce"."DivisionID" IS NULL OR
"ce"."DivisionID" = "d"."DivisionID")
    WHERE "ce"."personid" = "pPersonID"
    AND((SELECT "u"."IQXNetUserID" FROM "IQXNetUser" AS "u" WHERE
"u"."StaffID" = "ce"."StaffID") = "pWebUserID" OR "pAllEvents" = 1)
    ORDER BY "ce"."contactdate" DESC,
      "ce"."contacttime" DESC
  WHEN "pEmploymentID" IS NOT NULL THEN /* Contact events for a Contact
record*/
    SELECT top "x" START at "y"
      "ce"."contacteventid",
      "ce"."description",
      "datetime"("ce"."contactdate"+"ce"."contacttime") AS
"ContactEventTime",
      "c"."classdescrip" AS "ClassType",
      "isnull"("ce"."priority",1) AS "Priority",
      "ce"."outcome",
      "ce"."notes" AS "Notes",
```



```
"p"."name" AS "PersonName",
"cp"."name" AS "ContactName",
"s"."name" AS "StaffName",
"v"."position" AS "Position",
"co"."name" AS "CompanyName"
FROM "contactevent" AS "ce" KEY JOIN "staff" AS "s"
    , "contactevent" AS "ce" KEY JOIN "contactclass" AS "c"
    , "contactevent" AS "ce" KEY LEFT OUTER JOIN "person" AS "p"
    , "contactevent" AS "ce" KEY JOIN ("employment" AS "e" KEY
JOIN("company" AS "co", "person" AS "cp"))
    , "contactevent" AS "ce" KEY LEFT OUTER JOIN ("vacancy" AS "v")
    , "contactevent" AS "ce" LEFT OUTER JOIN (SELECT "DivisionID" FROM
"DashboardStaffDivisions"()) AS "d" ON ("ce"."DivisionID" IS NULL OR
"ce"."DivisionID" = "d"."DivisionID")
WHERE "ce"."EmploymentID" = "pEmploymentID"
AND ((SELECT "u"."IQXNetUserID" FROM "IQXNetUser" AS "u" WHERE
"u"."StaffID" = "ce"."StaffID") = "pWebUserID" OR "pAllEvents" = 1)
ORDER BY "ce"."contactdate" DESC,
"ce"."contacttime" DESC
WHEN "pEmploymentID" IS NULL AND "pCompanyID" IS NOT NULL THEN /* Contact
events for a Company record*/
SELECT top "x" START at "y"
    "ce"."contacteventid",
    "ce"."description",
    "datetime"("ce"."contactdate"+"ce"."contacttime") AS
"ContactEventTime",
    "c"."classdescrip" AS "ClassType",
    "isnull"("ce"."priority",1) AS "Priority",
    "ce"."outcome",
    "ce"."notes" AS "Notes",
    "p"."name" AS "PersonName",
    "cp"."name" AS "ContactName",
    "s"."name" AS "StaffName",
    "v"."position" AS "Position",
    "co"."name" AS "CompanyName"
FROM "contactevent" AS "ce" KEY JOIN "staff" AS "s"
    , "contactevent" AS "ce" KEY JOIN "contactclass" AS "c"
    , "contactevent" AS "ce" KEY LEFT OUTER JOIN "person" AS "p"
    , "contactevent" AS "ce" KEY JOIN ("employment" AS "e" KEY
JOIN("company" AS "co", "person" AS "cp"))
    , "contactevent" AS "ce" KEY LEFT OUTER JOIN ("vacancy" AS "v")
    , "contactevent" AS "ce" LEFT OUTER JOIN (SELECT "DivisionID" FROM
"DashboardStaffDivisions"()) AS "d" ON ("ce"."DivisionID" IS NULL OR
"ce"."DivisionID" = "d"."DivisionID")
WHERE "e"."CompanyID" = "pCompanyID"
AND ((SELECT "u"."IQXNetUserID" FROM "IQXNetUser" AS "u" WHERE
"u"."StaffID" = "ce"."StaffID") = "pWebUserID" OR "pAllEvents" = 1)
ORDER BY "ce"."contactdate" DESC,
```



```
"ce"."contacttime" DESC
WHEN "pVacancyID" IS NOT NULL THEN /* Contact events for a Vacancy
record*/
SELECT top "x" START at "y"
    "ce"."contacteventid",
    "ce"."description",
    "datetime"("ce"."contactdate"+"ce"."contacttime") AS
"ContactEventTime",
    "c"."classdescrip" AS "ClassType",
    "isnull"("ce"."priority",1) AS "Priority",
    "ce"."outcome",
    "ce"."notes" AS "Notes",
    "p"."name" AS "PersonName",
    "cp"."name" AS "ContactName",
    "s"."name" AS "StaffName",
    "v"."position" AS "Position",
    "co"."name" AS "CompanyName"
FROM "contactevent" AS "ce" KEY JOIN "staff" AS "s"
    , "contactevent" AS "ce" KEY JOIN "contactclass" AS "c"
    , "contactevent" AS "ce" KEY LEFT OUTER JOIN "person" AS "p"
    , "contactevent" AS "ce" KEY LEFT OUTER JOIN("employment" AS "e" KEY
JOIN("company" AS "co", "person" AS "cp"))
    , "contactevent" AS "ce" KEY JOIN("vacancy" AS "v")
    , "contactevent" AS "ce" LEFT OUTER JOIN(SELECT "DivisionID" FROM
"DashboardStaffDivisions"()) AS "d" ON("ce"."DivisionID" IS NULL OR
"ce"."DivisionID" = "d"."DivisionID")
WHERE "e"."VacancyID" = "pVacancyID"
AND((SELECT "u"."IQXNetUserID" FROM "IQXNetUser" AS "u" WHERE
"u"."StaffID" = "ce"."StaffID") = "pWebUserID" OR "pAllEvents" = 1)
ORDER BY "ce"."contactdate" DESC,
    "ce"."contacttime" DESC
END CASE
END /* TEST
call NetTestSetup('');
delete from contactevent where contacteventid in ('Test1','Test2','Test3');
delete from vacancy where vacancyid = 'TEST2';
delete from employment where employmentid = 'TEST2';

update staff set name='TestStaff' where staffid='TEST';
insert into person (personid,name,keyname,forenames,surname) on existing
update defaults off values ('TEST2','test2','TEST2','test2','test2');
insert into person (personid,name,keyname,forenames,surname) on existing
update defaults off values ('TEST3','test3','TEST3','test3','test3');
insert into employment (employmentid,companyid,personid) on existing update
defaults off values ('TEST','TEST','TEST');
insert into employment (employmentid,companyid,personid) on existing update
defaults off values ('TEST2','TEST','TEST2');
insert into vacancy
```



```
(vacancyid,departmentid,employmentid,tempdeskid,position) on existing update
defaults off values ('TEST','~~','TEST','TEST','test');
insert into vacancy
(vacancyid,departmentid,employmentid,tempdeskid,position) on existing update
defaults off values ('TEST2','~~','TEST','TEST','test2');
insert into contactevent
(contacteventid,staffid,personid,vacancyid,employmentid,contactdate,contacttime,description,outcome,classcode,who,notes,whenentered,whoentered,priority)
on existing update values
('TEST1','TEST','TEST','TEST','TEST',now(*),now(*),'TestEntry','TestEntryOutcome','~~','TEST','TestEntryNotes',now(*),'TEST.OWNER','1');
insert into contactevent
(contacteventid,staffid,personid,vacancyid,employmentid,contactdate,contacttime,description,outcome,classcode,who,notes,whenentered,whoentered,priority)
on existing update values
('TEST2','TEST','TEST','TEST','TEST2',now(*),dateadd(mi,-10,now(*)), 'TestEntry2','TestEntryOutcome2','~~','TEST','TestEntryNotes2',now(*),'TEST.OWNER','5');
insert into contactevent
(contacteventid,staffid,personid,vacancyid,employmentid,contactdate,contacttime,description,outcome,classcode,who,notes,whenentered,whoentered,priority)
on existing update values
('TEST3','TEST','TEST3','TEST2','TEST2',now(*),dateadd(mi,-10,now(*)), 'TestEntry3','TestEntryOutcome3','~~','TEST','TestEntryNotes3',now(*),'TEST.OWNER','5');

select * from
NetOwnerContactEvents('TEST.OWNER',0,50,null,null,'test',null,null,null,'testentry',today(*),today(*),'Test','%','TestEntryOutcome','TestEntryNotes','test','test','TestStaff','test','test','description asc');
expect
ContactEventID=TEST1,ClassType=Test*,StaffName=TestStaff,ContactName=test,Description=TestEntry;
delete from contactevent where contacteventid in ('Test1','Test2','Test3');
delete from vacancy where vacancyid = 'TEST2';
delete from employment where employmentid = 'TEST2';

*/
/* DOC
2016-07-25 PC Provide master list for Owner diary
2016-08-11 PC pCompanyID parameter and pAllEvents parameter added
2016-08-22 GJ pSlice and pSliceSize added
2016-09-26 PC change sort order to ignore priority, and order on contact date first, with the most recent at the top.
2017-04-24 ET Changed filter to include events on the start and end dates.
2018-09-14 JW Performance improvement for pulse

*/
```



GO

```
COMMENT TO PRESERVE FORMAT ON PROCEDURE "pears"."NetOwnerContactEvents" IS
{CREATE PROCEDURE pears."NetOwnerContactEvents"(IN pWebUserID CHAR(20),
    IN pSlice INTEGER DEFAULT 0,
    IN pSliceSize INTEGER DEFAULT 20,
    IN pStaffID CHAR(20) DEFAULT NULL,
    IN pEmploymentID CHAR(20) DEFAULT NULL,
    IN pPersonID CHAR(20) DEFAULT NULL,
    IN pVacancyID CHAR(20) DEFAULT NULL,
    IN pCompanyID CHAR(20) DEFAULT NULL,
    IN pAllEvents CHAR(1) DEFAULT NULL,
    IN pFilterDescription CHAR(100) DEFAULT '%',
    IN pFilterTimeStart DATE,
    IN pFilterTimeEnd DATE,
    IN pFilterClassType CHAR(50) DEFAULT '%',
    IN pFilterPriority CHAR(1) DEFAULT '%',
    IN pFilterOutcome long VARCHAR DEFAULT '%',
    IN pFilterNotes long VARCHAR DEFAULT '%',
    IN pFilterPersonName CHAR(50) DEFAULT '%',
    IN pFilterContactName CHAR(60) DEFAULT '%',
    IN pFilterStaffName CHAR(60) DEFAULT '%',
    IN pFilterPosition CHAR(60) DEFAULT '%',
    IN pFilterCompanyName CHAR(60) DEFAULT '%',
    IN pOrderBy CHAR(255) DEFAULT ''
)

RESULT(ContactEventID CHAR(20),
    Description CHAR(100),
    ContactEventTime datetime,
    ClassType CHAR(50),
    Priority CHAR(1),
    Outcome long VARCHAR,
    Notes long VARCHAR,
    PersonName CHAR(50),
    ContactName CHAR(60),
    StaffName CHAR(60),
    POSITION CHAR(60),
    CompanyName CHAR(60)
)

BEGIN
// IQXNet
    DECLARE x INTEGER;
    DECLARE y INTEGER;
    SET x=pSliceSize;
    SET y=x*pSlice+1;
CASE WHEN pCompanyID IS NULL AND pPersonID IS NULL AND pEmploymentID IS NULL
```



```
AND pVacancyID IS NULL THEN RETURN
```

```
WHEN pPersonID IS NOT NULL THEN /* Contact events for a Person record*/
```

```
SELECT top x START at y
```

```
ce.contacteventid,
```

```
ce.description,
```

```
datetime(ce.contactdate + ce.contacttime) AS ContactEventTime,
```

```
c.classdescrip AS ClassType,
```

```
isnull(ce.priority,1) AS Priority,
```

```
ce.outcome,
```

```
ce.notes AS "Notes",
```

```
p.name AS PersonName,
```

```
cp.name AS ContactName,
```

```
s.name AS StaffName,
```

```
v.position AS POSITION,
```

```
co.name AS CompanyName
```

```
FROM
```

```
contactevent ce KEY JOIN staff s,
```

```
contactevent ce KEY JOIN contactclass c,
```

```
contactevent ce KEY JOIN person p,
```

```
contactevent ce KEY LEFT OUTER JOIN (employment e KEY JOIN
```

```
(company co, person cp)),
```

```
contactevent ce KEY LEFT OUTER JOIN (vacancy v),
```

```
contactevent ce LEFT JOIN (SELECT DivisionID FROM
```

```
DashboardStaffDivisions()) AS d ON (ce.DivisionID IS NULL OR ce.DivisionID =  
d.DivisionID)
```

```
WHERE
```

```
ce.personid = pPersonID AND
```

```
((SELECT u.IQXNetUserID FROM IQXNetUser u WHERE u.StaffID =
```

```
ce.StaffID) = pWebUserID OR pAllEvents=1 )
```

```
ORDER BY
```

```
ce.contactdate DESC,
```

```
ce.contacttime DESC
```

```
WHEN pEmploymentID IS NOT NULL THEN /* Contact events for a Contact  
record*/
```

```
SELECT top x START at y
```

```
ce.contacteventid,
```

```
ce.description,
```

```
datetime(ce.contactdate + ce.contacttime) AS ContactEventTime,
```

```
c.classdescrip AS ClassType,
```

```
isnull(ce.priority,1) AS Priority,
```

```
ce.outcome,
```

```
ce.notes AS "Notes",
```

```
p.name AS PersonName,
```

```
cp.name AS ContactName,
```

```
s.name AS StaffName,
```

```
v.position AS POSITION,
```



```
co.name AS CompanyName
FROM
  contactevent ce KEY JOIN staff s,
  contactevent ce KEY JOIN contactclass c,
  contactevent ce KEY LEFT OUTER JOIN person p,
  contactevent ce KEY JOIN (employment e KEY JOIN (company co,
person cp)),
  contactevent ce KEY LEFT OUTER JOIN (vacancy v),
  contactevent ce LEFT JOIN (SELECT DivisionID FROM
DashboardStaffDivisions()) AS d ON (ce.DivisionID IS NULL OR ce.DivisionID =
d.DivisionID)
WHERE
  ce.EmploymentID = pEmploymentID AND
  ((SELECT u.IQXNetUserID FROM IQXNetUser u WHERE u.StaffID =
ce.StaffID) = pWebUserID OR pAllEvents=1 )
ORDER BY
  ce.contactdate DESC,
  ce.contacttime DESC

WHEN pEmploymentID IS NULL AND pCompanyID IS NOT NULL THEN /* Contact
events for a Company record*/
SELECT top x START at y
  ce.contacteventid,
  ce.description,
  datetime(ce.contactdate + ce.contacttime) AS ContactEventTime,
  c.classdescrip AS ClassType,
  isnull(ce.priority,1) AS Priority,
  ce.outcome,
  ce.notes AS "Notes",
  p.name AS PersonName,
  cp.name AS ContactName,
  s.name AS StaffName,
  v.position AS POSITION,
  co.name AS CompanyName
FROM
  contactevent ce KEY JOIN staff s,
  contactevent ce KEY JOIN contactclass c,
  contactevent ce KEY LEFT OUTER JOIN person p,
  contactevent ce KEY JOIN (employment e KEY JOIN (company co,
person cp)),
  contactevent ce KEY LEFT OUTER JOIN (vacancy v),
  contactevent ce LEFT JOIN (SELECT DivisionID FROM
DashboardStaffDivisions()) AS d ON (ce.DivisionID IS NULL OR ce.DivisionID =
d.DivisionID)
WHERE
  e.CompanyID = pCompanyID AND
  ((SELECT u.IQXNetUserID FROM IQXNetUser u WHERE u.StaffID =
ce.StaffID) = pWebUserID OR pAllEvents=1 )
```




```
ORDER BY
    ce.contactdate DESC,
    ce.contacttime DESC

WHEN pVacancyID IS NOT NULL THEN /* Contact events for a Vacancy
record*/
SELECT top x START at y
    ce.contacteventid,
    ce.description,
    datetime(ce.contactdate + ce.contacttime) AS ContactEventTime,
    c.classdescrip AS ClassType,
    isnull(ce.priority,1) AS Priority,
    ce.outcome,
    ce.notes AS "Notes",
    p.name AS PersonName,
    cp.name AS ContactName,
    s.name AS StaffName,
    v.position AS POSITION,
    co.name AS CompanyName
FROM
    contactevent ce KEY JOIN staff s,
    contactevent ce KEY JOIN contactclass c,
    contactevent ce KEY LEFT OUTER JOIN person p,
    contactevent ce KEY LEFT OUTER JOIN (employment e KEY JOIN
(company co, person cp)),
    contactevent ce KEY JOIN (vacancy v),
    contactevent ce LEFT JOIN (SELECT DivisionID FROM
DashboardStaffDivisions()) AS d ON (ce.DivisionID IS NULL OR ce.DivisionID =
d.DivisionID)
WHERE
    e.VacancyID = pVacancyID AND
    ((SELECT u.IQXNetUserID FROM IQXNetUser u WHERE u.StaffID =
ce.StaffID) = pWebUserID OR pAllEvents=1 )
ORDER BY
    ce.contactdate DESC,
    ce.contacttime DESC

END CASE;

END

/* TEST
call NetTestSetup('');
delete from contactevent where contacteventid in ('Test1','Test2','Test3');
delete from vacancy where vacancyid = 'TEST2';
delete from employment where employmentid = 'TEST2';

update staff set name='TestStaff' where staffid='TEST';
insert into person (personid,name,keyname,forenames,surname) on existing
```



```
update defaults off values ('TEST2','test2','TEST2','test2','test2');
insert into person (personid,name,keyname,forenames,surname) on existing
update defaults off values ('TEST3','test3','TEST3','test3','test3');
insert into employment (employmentid,companyid,personid) on existing update
defaults off values ('TEST','TEST','TEST');
insert into employment (employmentid,companyid,personid) on existing update
defaults off values ('TEST2','TEST','TEST2');
insert into vacancy
(vacancyid,departmentid,employmentid,tempdeskid,position) on existing update
defaults off values ('TEST','~~','TEST','TEST','test');
insert into vacancy
(vacancyid,departmentid,employmentid,tempdeskid,position) on existing update
defaults off values ('TEST2','~~','TEST','TEST','test2');
insert into contactevent
(contacteventid,staffid,personid,vacancyid,employmentid,contactdate,contactt
ime,description,outcome,classcode,who,notes,whenentered,whoentered,priority)
on existing update values
('TEST1','TEST','TEST','TEST','TEST',now(*),now(*),'TestEntry','TestEntryOut
come','~~','TEST','TestEntryNotes',now(*),'TEST.OWNER','1');
insert into contactevent
(contacteventid,staffid,personid,vacancyid,employmentid,contactdate,contactt
ime,description,outcome,classcode,who,notes,whenentered,whoentered,priority)
on existing update values
('TEST2','TEST','TEST','TEST','TEST2',now(*),dateadd(mi,-10,now(*)), 'TestEnt
ry2','TestEntryOutcome2','~~','TEST','TestEntryNotes2',now(*),'TEST.OWNER','
5');
insert into contactevent
(contacteventid,staffid,personid,vacancyid,employmentid,contactdate,contactt
ime,description,outcome,classcode,who,notes,whenentered,whoentered,priority)
on existing update values
('TEST3','TEST','TEST3','TEST2','TEST2',now(*),dateadd(mi,-10,now(*)), 'TestE
ntry3','TestEntryOutcome3','~~','TEST','TestEntryNotes3',now(*),'TEST.OWNER'
,'5');

select * from
NetOwnerContactEvents('TEST.OWNER',0,50,null,null,'test',null,null,null,'tes
tentry',today(*),today(*),'Test','%','TestEntryOutcome','TestEntryNotes','te
st','test','TestStaff','test','test','description asc');
expect
ContactEventID=TEST1,ClassType=Test*,StaffName=TestStaff,ContactName=test,De
scription=TestEntry;
delete from contactevent where contacteventid in ('Test1','Test2','Test3');
delete from vacancy where vacancyid = 'TEST2';
delete from employment where employmentid = 'TEST2';

*/

/* DOC
```



2016-07-25 PC Provide master list for Owner diary
2016-08-11 PC pCompanyID parameter and pAllEvents parameter added
2016-08-22 GJ pSlice and pSliceSize added
2016-09-26 PC change sort order to ignore priority, and order on contact date first, with the most recent at the top.
2017-04-24 ET Changed filter to include events on the start and end dates.
2018-09-14 JW Performance improvement for pulse

*/
}

From:

<https://iqxusers.co.uk/iqxhelp/> - **iqx**

Permanent link:

https://iqxusers.co.uk/iqxhelp/doku.php?id=database:procedures:pears_netownercontactevents

Last update: **2026/08/07 19:24**

