

# Package ‘vacols’

October 6, 2016

**Title** VACOLS

**Version** 0.0.1

**Description** Scripts for the analysis of the Department of Veterans Affairs'  
VACOLS database

**Depends** R (>= 3.3.1)

**License** Public Domain/CC0

**Encoding** UTF-8

**LazyData** true

**Imports** ROracle,  
magrittr,  
dplyr,  
tidyr

**RoxygenNote** 5.0.1

## R topics documented:

event_abeyance . . . . .	2
event_activation . . . . .	2
event_all . . . . .	3
event_assignment . . . . .	4
event_cavc . . . . .	4
event_certification . . . . .	5
event_decision . . . . .	6
event_docket . . . . .	6
event_endStates . . . . .	7
event_form9 . . . . .	8
event_getDateCols . . . . .	8
event_getPriorLocs . . . . .	9
event_hearing . . . . .	9
event_nod . . . . .	10
event_omo . . . . .	11
event_qr . . . . .	11
event_soc . . . . .	12
event_ssoc . . . . .	13
event_transcript . . . . .	13
event_vacolsCreation . . . . .	14
event_vso . . . . .	15
vacolsConnect . . . . .	15

**Index****17**


---

event_abeyance	<i>Retrieve a log of Abeyance (TO_ABEYANCE and FROM_ABEYANCE) events matching the specified criteria.</i>
----------------	-----------------------------------------------------------------------------------------------------------

---

**Description**

Retrieve a log of Abeyance (TO\_ABEYANCE and FROM\_ABEYANCE) events matching the specified criteria.

**Usage**

```
event_abeyance(con, where)
```

**Arguments**

con	OraConnection to VACOLS created by vacolsConnect.
where	(optional) SQL statement to filter results.

**Value**

A dataframe of events, with the variables BFCORLID, APPEAL\_DATE, BFKEY, EVENT\_TYPE and DATE.

**Methodology**

Uses the LOCDOUT and LOCDIN dates, respectively, from events where the location is "24" or "39". The STAFF table is joined when querying the PRIORLOC table, and individuals (those where STITLE is not null) are replaced with the parent location (STITLE). Merged appeals and dummy data are excluded.

**Examples**

```
event_abeyance(con, where = "BFDNOD > date '2009-01-01' and BFDNOD < date '2009-12-31'")
```

---

event_activation	<i>Retrieve a log of Activated at BVA (ACTIVATION) events matching the specified criteria.</i>
------------------	------------------------------------------------------------------------------------------------

---

**Description**

Retrieve a log of Activated at BVA (ACTIVATION) events matching the specified criteria.

**Usage**

```
event_activation(con, where)
```

**Arguments**

con	OraConnection to VACOLS created by vacolsConnect.
where	(optional) SQL statement to filter results.

**Value**

A dataframe of events, with the variables BFCORLID, APPEAL\_DATE, BFKEY, EVENT\_TYPE and DATE.

**Methodology**

Uses the LOCDOUT date from events in the PRIORLOC table where the location is "01". The STAFF table is joined when querying the PRIORLOC table, and individuals (those where STITLE is not null) are replaced with the parent location (STITLE). Merged appeals and dummy data are excluded.

**Examples**

```
event_activation(con, where = "BFDNOD > date '2009-01-01' and BFDNOD < date '2009-12-31'")
```

---

event\_all

*Retrieve a log of all events matching the specified criteria. Consult individual event functions for methodology details.*

---

**Description**

Includes: Notice of Disagreement (NOD), Statement of Case (SOC), Form 9 (FORM9), Supplemental Statement of Case (SSOC), Certification to BVA (CERTIFICATION), Activated at BVA (ACTIVATION), Hearing Held (HEARING), Transcript Received (TRANSCRIPT), Service Organization Review (TO\_VSO and FROM\_VSO), Assignment (ASSIGNMENT), Abeyance (TO\_ABEYANCE and FROM\_ABEYANCE), Outside Medical Opinion (TO\_OMO and FROM\_OMO), Decision (DECISION), Quality Review (QR), Withdrawn (WITHDRAWN), Dismissed (DISMISSED), Remand (REMAND), AOJ Grant (AOJ\_GRANT), Final Dispatch (DISPATCH), Vacated (VACATED), CAVC Decision (CAVC)

Does not include: VACOLS Creation (VACOLS), Assigned to Docket (DOCKET)

**Usage**

```
event_all(con, where)
```

**Arguments**

con	OraConnection to VACOLS created by vacolsConnect.
where	(optional) SQL statement to filter results.

**Value**

A dataframe of events, with the variables BFCORLID, APPEAL\_DATE, BFKEY, EVENT\_TYPE and DATE.

**Examples**

```
event_all(con, where = "BFDNOD > date '2009-01-01' and BFDNOD < date '2009-12-31'")
```

---

event_assignment	<i>Retrieve a log of Assignment (ASSIGNMENT) events matching the specified criteria.</i>
------------------	------------------------------------------------------------------------------------------

---

### Description

Retrieve a log of Assignment (ASSIGNMENT) events matching the specified criteria.

### Usage

```
event_assignment(con, where)
```

### Arguments

con	OraConnection to VACOLS created by vacolsConnect.
where	(optional) SQL statement to filter results.

### Value

A dataframe of events, with the variables BFCORLID, APPEAL\_DATE, BFKEY, EVENT\_TYPE and DATE.

### Methodology

Uses the LOCDIN date from the first of the events in the PRIORLOC table where its location is the decision team (D[1-5]) and the subsequent event is also owned by the decision team. This is intended to capture the handoff from judge to attorney. The STAFF table is joined when querying the PRIORLOC table, and individuals (those where STITLE is not null) are replaced with the parent location (STITLE). Merged appeals and dummy data are excluded.

### Examples

```
event_assignment(con, where = "BFDNOD > date '2009-01-01' and BFDNOD < date '2009-12-31'")
```

---

event_cavc	<i>Retrieve a log of CAVC Decision (CAVC) events matching the specified criteria.</i>
------------	---------------------------------------------------------------------------------------

---

### Description

Retrieve a log of CAVC Decision (CAVC) events matching the specified criteria.

### Usage

```
event_cavc(con, where)
```

### Arguments

con	OraConnection to VACOLS created by vacolsConnect.
where	(optional) SQL statement to filter results.

**Value**

A dataframe of events, with the variables BFCORLID, APPEAL\_DATE, BFKEY, EVENT\_TYPE and DATE.

**Methodology**

Uses the date from the CVDDEC column, uniquing on the BFKEY column. Merged appeals and dummy data are excluded.

**Examples**

```
event_cavc(con, where = "BFDNOD > date '2009-01-01' and BFDNOD < date '2009-12-31'")
```

---

event_certification	<i>Retrieve a log of Certification to BVA (CERTIFICATION) events matching the specified criteria.</i>
---------------------	-------------------------------------------------------------------------------------------------------

---

**Description**

Retrieve a log of Certification to BVA (CERTIFICATION) events matching the specified criteria.

**Usage**

```
event_certification(con, where)
```

**Arguments**

con	OraConnection to VACOLS created by vacolsConnect.
where	(optional) SQL statement to filter results.

**Value**

A dataframe of events, with the variables BFCORLID, APPEAL\_DATE, BFKEY, EVENT\_TYPE and DATE.

**Methodology**

Uses the date from the BF41STAT column, limited to appeals where the Type Action (BFAC) indicates that it is the original action (1). Merged appeals and dummy data are also excluded.

**Examples**

```
event_certification(con, where = "BFDNOD > date '2009-01-01' and BFDNOD < date '2009-12-31'")
```

---

event_decision	<i>Retrieve a log of Decision (DECISION) events matching the specified criteria.</i>
----------------	--------------------------------------------------------------------------------------

---

**Description**

Retrieve a log of Decision (DECISION) events matching the specified criteria.

**Usage**

```
event_decision(con, where)
```

**Arguments**

con	OraConnection to VACOLS created by vacolsConnect.
where	(optional) SQL statement to filter results.

**Value**

A dataframe of events, with the variables BFCORLID, APPEAL\_DATE, BFKEY, EVENT\_TYPE and DATE.

**Methodology**

Looks for a series of events in the following set of locations: 1. the decision team (D[1-5]), 2. the administration team (A.+ISUP|OPR), and 3. Central Dispatch ("30"). Uses the LOCDIN date from the first event. The STAFF table is joined when querying the PRIORLOC table, and individuals (those where STITLE is not null) are replaced with the parent location (STITLE). Merged appeals and dummy data are excluded.

**Examples**

```
event_decision(con, where = "BFDNOD > date '2009-01-01' and BFDNOD < date '2009-12-31'")
```

---

event_docket	<i>Retrieve a log of Assigned to Docket (DOCKET) events matching the specified criteria.</i>
--------------	----------------------------------------------------------------------------------------------

---

**Description**

Retrieve a log of Assigned to Docket (DOCKET) events matching the specified criteria.

**Usage**

```
event_docket(con, where)
```

**Arguments**

con	OraConnection to VACOLS created by vacolsConnect.
where	(optional) SQL statement to filter results.

**Value**

A dataframe of events, with the variables BFCORLID, APPEAL\_DATE, BFKEY, EVENT\_TYPE and DATE.

**Methodology**

Uses the date from the TIADTIME column on the FOLDER table, limited to appeals where the Type Action (BFAC) indicates that it is the original action (1). Merged appeals and dummy data are also excluded.

**Examples**

```
event_docket(con, where = "BFDNOD > date '2009-01-01' and BFDNOD < date '2009-12-31'")
```

---

event_endStates	<i>Retrieve a log of end state events matching the specified criteria.</i>
-----------------	----------------------------------------------------------------------------

---

**Description**

Includes: Withdrawn (WITHDRAWN), Dismissed (DISMISSED), Vacated (VACATED), Remand (REMAND), AOJ Grant (AOJ\_GRANT), Final Dispatch (DISPATCH)

**Usage**

```
event_endStates(con, where)
```

**Arguments**

con	OraConnection to VACOLS created by vacolsConnect.
where	(optional) SQL statement to filter results.

**Value**

A dataframe of events, with the variables BFCORLID, APPEAL\_DATE, BFKEY, EVENT\_TYPE and DATE.

**Methodology**

Uses the date from the BFDDEC column, classified according to the BFDC column. Consult EventEndStateClassifier in constants.R for the mapping used. Merged appeals and dummy data are excluded.

**Examples**

```
event_endStates(con, where = "BFDNOD > date '2009-01-01' and BFDNOD < date '2009-12-31'")
```

---

event_form9	<i>Retrieve a log of Form 9 (FORM9) events matching the specified criteria.</i>
-------------	---------------------------------------------------------------------------------

---

### Description

Retrieve a log of Form 9 (FORM9) events matching the specified criteria.

### Usage

```
event_form9(con, where)
```

### Arguments

con	OraConnection to VACOLS created by vacolsConnect.
where	(optional) SQL statement to filter results.

### Value

A dataframe of events, with the variables BFCORLID, APPEAL\_DATE, BFKEY, EVENT\_TYPE and DATE.

### Methodology

Uses the date from the BFD19 column, limited to appeals where the Type Action (BFAC) indicates that it is the original action (1). Merged appeals and dummy data are also excluded.

### Examples

```
event_form9(con, where = "BFDNOD > date '2009-01-01' and BFDNOD < date '2009-12-31'")
```

---

event_getDateCols	<i>Retrieve a log of events that appear in VACOLS as date columns.</i>
-------------------	------------------------------------------------------------------------

---

### Description

Retrieve a log of events that appear in VACOLS as date columns.

### Usage

```
event_getDateCols(con, cols, labs, join, where)
```

### Arguments

con	OraConnection to VACOLS created by vacolsConnect.
cols	Vector of column names to include in results.
labs	(optional) Vector of labels to be used for the EVENT_TYPE variable.
join	(optional) SQL statement to join a table other than BRIEFF.
where	(optional) SQL statement to filter results.



**Value**

A dataframe of events, with the variables BFCORLID, APPEAL\_DATE, BFKEY, EVENT\_TYPE and DATE.

**Examples**

```
cols <- c("BFNOD", "BFDSOC", "BFD19")
labs <- c("NOD", "SOC", "FORM9")
event_getDateCols(con, cols, labs, where = "BFDNOD > date '2009-01-01' and BFDNOD < date '2009-12-31'")

event_getDateCols(con, c("HEARING_DATE"), c("hearing"), join = "HEARSCHED on FOLDER_NR = BFKEY", where = "HEA
```

---

event_getPriorLocs	<i>Retrieve a log of events from the VACOLS PRIORLOC table.</i>
--------------------	-----------------------------------------------------------------

---

**Description**

Retrieve a log of events from the VACOLS PRIORLOC table.

**Usage**

```
event_getPriorLocs(con, where)
```

**Arguments**

con	OraConnection to VACOLS created by vacolsConnect.
where	(optional) SQL statement to filter results.

**Value**

A dataframe of events, with the variables BFCORLID, APPEAL\_DATE, BFKEY, LOCDOUT, LOCDIN, LOCSTTO, LOCSTOUT, LOCSTRCV, LOC and LOC\_NAME.#'

**Examples**

```
event_getPriorLocs(con, "LOCSTTO = '48' and LOCSTOUT > date '2015-01-01' and LOCSTOUT < date '2015-12-31'")
```

---

event_hearing	<i>Retrieve a log of Hearing Held (HEARING) events matching the specified criteria.</i>
---------------	-----------------------------------------------------------------------------------------

---

**Description**

Retrieve a log of Hearing Held (HEARING) events matching the specified criteria.

**Usage**

```
event_hearing(con, where)
```

**Arguments**

con                      OraConnection to VACOLS created by vacolsConnect.  
 where                    (optional) SQL statement to filter results.

**Value**

A dataframe of events, with the variables BFCORLID, APPEAL\_DATE, BFKEY, EVENT\_TYPE and DATE.

**Methodology**

Uses the date from the HEARING\_DATE column on the FOLDER table, limited to appeals where the hearing has been held (HEARING\_DISP = 'H'), and excluding RO formal hearings (HEARING\_TYPE <> 'F'). Merged appeals and dummy data are also excluded.

**Examples**

```
event_hearing(con, where = "BFDNOD > date '2009-01-01' and BFDNOD < date '2009-12-31'")
```

---

event_nod	<i>Retrieve a log of Notice of Disagreement (NOD) dates matching the specified criteria.</i>
-----------	----------------------------------------------------------------------------------------------

---

**Description**

Retrieve a log of Notice of Disagreement (NOD) dates matching the specified criteria.

**Usage**

```
event_nod(con, where)
```

**Arguments**

con                      OraConnection to VACOLS created by vacolsConnect.  
 where                    (optional) SQL statement to filter results.

**Value**

A dataframe of events, with the variables BFCORLID, APPEAL\_DATE, BFKEY, EVENT\_TYPE and DATE.

**Methodology**

Uses the date from the BFDNOD column, limited to appeals where the Type Action (BFAC) indicates that it is the original action (1). Merged appeals and dummy data are also excluded.

**Examples**

```
event_nod(con, where = "BFDNOD > date '2009-01-01' and BFDNOD < date '2009-12-31'")
```

---

event_omo	<i>Retrieve a log of Outside Medical Opinion (TO_OMO and FROM_OMO) events matching the specified criteria.</i>
-----------	----------------------------------------------------------------------------------------------------------------

---

**Description**

Retrieve a log of Outside Medical Opinion (TO\_OMO and FROM\_OMO) events matching the specified criteria.

**Usage**

```
event_omo(con, where)
```

**Arguments**

con	OraConnection to VACOLS created by vacolsConnect.
where	(optional) SQL statement to filter results.

**Value**

A dataframe of events, with the variables BFCORLID, APPEAL\_DATE, BFKEY, EVENT\_TYPE and DATE.

**Methodology**

Uses the LOCDOUT and LOCDIN dates, respectively, from events where the location is "92" (Outside BVA) that are preceded by an event where the location is "20" (Opinion Request). The STAFF table is joined when querying the PRIORLOC table, and individuals (those where STITLE is not null) are replaced with the parent location (STITLE). Merged appeals and dummy data are excluded.

**Examples**

```
event_omo(con, where = "BFDNOD > date '2009-01-01' and BFDNOD < date '2009-12-31'")
```

---

event_qr	<i>Retrieve a log of Quality Review (QR) events matching the specified criteria.</i>
----------	--------------------------------------------------------------------------------------

---

**Description**

Retrieve a log of Quality Review (QR) events matching the specified criteria.

**Usage**

```
event_qr(con, where)
```

**Arguments**

con	OraConnection to VACOLS created by vacolsConnect.
where	(optional) SQL statement to filter results.

**Value**

A dataframe of events, with the variables BFCORLID, APPEAL\_DATE, BFKEY, EVENT\_TYPE and DATE.

**Methodology**

Uses the LOCDIN date from events where the location is "48" (Outside BVA). The STAFF table is joined when querying the PRIORLOC table, and individuals (those where STITLE is not null) are replaced with the parent location (STITLE). Merged appeals and dummy data are excluded.

**Examples**

```
event_qr(con, where = "BFDNOD > date '2009-01-01' and BFDNOD < date '2009-12-31'")
```

---

event_soc	<i>Retrieve a log of Statement of Case (SOC) events matching the specified criteria.</i>
-----------	------------------------------------------------------------------------------------------

---

**Description**

Retrieve a log of Statement of Case (SOC) events matching the specified criteria.

**Usage**

```
event_soc(con, where)
```

**Arguments**

- con                   OraConnection to VACOLS created by vacolsConnect.
- where                (optional) SQL statement to filter results.

**Value**

A dataframe of events, with the variables BFCORLID, APPEAL\_DATE, BFKEY, EVENT\_TYPE and DATE.

**Methodology**

Uses the date from the BFDSOC column, limited to appeals where the Type Action (BFAC) indicates that it is the original action (1). Merged appeals and dummy data are also excluded.

**Examples**

```
event_soc(con, where = "BFDNOD > date '2009-01-01' and BFDNOD < date '2009-12-31'")
```

---

event_ssoc	<i>Retrieve a log of Supplemental Statement of Case (SSOC) events matching the specified criteria.</i>
------------	--------------------------------------------------------------------------------------------------------

---

**Description**

Retrieve a log of Supplemental Statement of Case (SSOC) events matching the specified criteria.

**Usage**

```
event_ssoc(con, where)
```

**Arguments**

con	OraConnection to VACOLS created by vacolsConnect.
where	(optional) SQL statement to filter results.

**Value**

A dataframe of events, with the variables BFCORLID, APPEAL\_DATE, BFKEY, EVENT\_TYPE and DATE.

**Methodology**

Uses the dates from the BFSSOC1, BFSSOC2, BFSSOC3, BFSSOC4, and BFSSOC5 columns from any appeal, uniqued by the BFCORLID. Merged appeals and dummy data are excluded.

**Examples**

```
event_ssoc(con, where = "BFDNOD > date '2009-01-01' and BFDNOD < date '2009-12-31'")
```

---

event_transcript	<i>Retrieve a log of Transcript Received (TRANSCRIPT) events matching the specified criteria.</i>
------------------	---------------------------------------------------------------------------------------------------

---

**Description**

Retrieve a log of Transcript Received (TRANSCRIPT) events matching the specified criteria.

**Usage**

```
event_transcript(con, where)
```

**Arguments**

con	OraConnection to VACOLS created by vacolsConnect.
where	(optional) SQL statement to filter results.

**Value**

A dataframe of events, with the variables BFCORLID, APPEAL\_DATE, BFKEY, EVENT\_TYPE and DATE.

**Methodology**

Uses the date from the CONRET column on the FOLDER table, limited to appeals where the hearing has been held (HEARING\_DISP = 'H'), and excluding RO formal hearings (HEARING\_TYPE <> 'F'). Merged appeals and dummy data are also excluded.

**Examples**

```
event_transcript(con, where = "BFDNOD > date '2009-01-01' and BFDNOD < date '2009-12-31'")
```

---

event_vacolsCreation	<i>Retrieve a log of VACOLS Creation (VACOLS) events matching the specified criteria.</i>
----------------------	-------------------------------------------------------------------------------------------

---

**Description**

Retrieve a log of VACOLS Creation (VACOLS) events matching the specified criteria.

**Usage**

```
event_vacolsCreation(con, where)
```

**Arguments**

- con                    OraConnection to VACOLS created by vacolsConnect.
- where                (optional) SQL statement to filter results.

**Value**

A dataframe of events, with the variables BFCORLID, APPEAL\_DATE, BFKEY, EVENT\_TYPE and DATE.

**Methodology**

Uses the minimum LOCDOUT timestamp from the PRIORLOC table. Merged appeals and dummy data are excluded.

**Examples**

```
event_vacolsCreation(con, where = "BFDNOD > date '2009-01-01' and BFDNOD < date '2009-12-31'")
```

---

event_vso	<i>Retrieve a log of Service Organization Review (TO_VSO and FROM_VSO) events matching the specified criteria.</i>
-----------	--------------------------------------------------------------------------------------------------------------------

---

### Description

Retrieve a log of Service Organization Review (TO\_VSO and FROM\_VSO) events matching the specified criteria.

### Usage

```
event_vso(con, where)
```

### Arguments

con	OraConnection to VACOLS created by vacolsConnect.
where	(optional) SQL statement to filter results.

### Value

A dataframe of events, with the variables BFCORLID, APPEAL\_DATE, BFKEY, EVENT\_TYPE and DATE.

### Methodology

Uses the LOCDIN date from events in the PRIORLOC table where the location is not "55" and the subsequent event's location is "55" for TO\_VSO, and the LOCDOUT date from events where the location is "55" and the subsequent location is not "55" for FROM\_VSO. The STAFF table is joined when querying the PRIORLOC table, and individuals (those where STITLE is not null) are replaced with the parent location (STITLE). Merged appeals and dummy data are excluded.

### Examples

```
event_vso(con, where = "BFDNOD > date '2009-01-01' and BFDNOD < date '2009-12-31'")
```

---

vacolsConnect	<i>Open a connection to VACOLS. Looks for the environment variables vacols_db_password or vacols_dev_password, depending on the environment specified by dev.</i>
---------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------

---

### Description

Open a connection to VACOLS. Looks for the environment variables vacols\_db\_password or vacols\_dev\_password, depending on the environment specified by dev.

### Usage

```
vacolsConnect(dev = FALSE, password = NULL)
```

**Arguments**

dev	If FALSE (the default), uses the VACOLS production environment. If TRUE, uses the development environment.
password	Override the environment variables.

**Value**

An `OraConnection` object.



# Index

event\_abeyance, [2](#)  
event\_activation, [2](#)  
event\_all, [3](#)  
event\_assignment, [4](#)  
event\_cavc, [4](#)  
event\_certification, [5](#)  
event\_decision, [6](#)  
event\_docket, [6](#)  
event\_endStates, [7](#)  
event\_form9, [8](#)  
event\_getDateCols, [8](#)  
event\_getPriorLocs, [9](#)  
event\_hearing, [9](#)  
event\_nod, [10](#)  
event\_omo, [11](#)  
event\_qr, [11](#)  
event\_soc, [12](#)  
event\_ssoc, [13](#)  
event\_transcript, [13](#)  
event\_vacolsCreation, [14](#)  
event\_vso, [15](#)  
  
vacolsConnect, [15](#)