

VACOLS

February 22, 2017

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 date from the TIDRECV column on the FOLDER table. 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	<i>Retrieve a log of all events matching the specified criteria. Consult individual event functions for methodology details.</i>
-----------	--

Description

Includes: Notice of Disagreement (NOD), VACOLS Creation (VACOLS), Substitution (SUBSTITUTION), Assigned to Docket (DOCKET), Statement of Case (SOC), Form 9 (FORM9), Supplemental Statement of Case (SSOC), Certification to BVA (CERTIFICATION), Activated at BVA (ACTIVATION), Hearing Held (HEARING), Hearing Not Held (HEARING_EXCEPTION), Transcription (TO_TRANSCRIPT and FROM_TRANSCRIPT), Translation (TRANSLATION_REQ, TO_TRANSLATION and FROM_TRANSLATION), 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 Drafted (DRAFT_DECISION), Decision Signed (SIGNED_DECISION), Decision Issued (DECISION), Quality Review (QR), Outcoding (OUTCODING), Withdrawn (WITHDRAWN), Dismissed (DISMISSED), Remand (REMAND), Remand Return (REMAND_RETURN), AOJ Grant (AOJ_GRANT), Final Dispatch (DISPATCH), Vacated (VACATED), CAVC Decision (CAVC)

Usage

```
event_all(con, join, 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 date from the DEASSIGN column on the DECASS table. 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, join, 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 Signed Decision (SIGNED_DECISION) events matching the specified criteria.</i>
----------------	--

Description

Retrieve a log of Signed Decision (SIGNED_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. one or more of the administration team (A.+ISUPIOPR), 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_drafted	<i>Retrieve a log of Decision Drafted (DRAFTED) events matching the specified criteria.</i>
---------------	---

Description

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

Usage

```
event_drafted(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 DERECEIVE column on the DECASS table. Merged appeals and dummy data are excluded.

Examples

```
event_drafted(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, join, 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, join, 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) and Hearing Not Held (HEARING_EXCEPTION) events matching the specified criteria.</i>
---------------	--

Description

Retrieve a log of Hearing Held (HEARING) and Hearing Not Held (HEARING_EXCEPTION) events matching the specified criteria.

Usage

```
event_hearing(con, join, 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, excluding formal hearings (HEARING_DISP = "F"), and RO formal hearings (HEARING_TYPE <> "F"). Hearings where the HEARING_DISP = "H" and marked as held; others as not held. If a CLSDATE on an unheld hearing is provided, that is used instead of HEARING_DATE. 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_outcoding	<i>Retrieve a log of OUTCODING (OUTCODING) events matching the specified criteria.</i>
-----------------	--

Description

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

Usage

```
event_outcoding(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 TIOCTIME column on the FOLDER table. Merged appeals and dummy data are excluded.

Examples

```
event_outcoding(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" (Quality Review). 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_remreturn	<i>Retrieve a log of Remand Return (REMAND_RETURN) events matching the specified criteria.</i>
-----------------	--

Description

Retrieve a log of Remand Return (REMAND_RETURN) events matching the specified criteria.

Usage

```
event_remreturn(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 where the location is "96" (Remand Return). If checked out to 96 multiple times, the last time is used. 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_remreturn(con, where = "BFDNOD > date '2009-01-01' and BFDNOD < date '2009-12-31'")
```

event_review	<i>Retrieve a log of Case Review (CASE_REVIEW) events matching the specified criteria.</i>
--------------	--

Description

Retrieve a log of Case Review (CASE_REVIEW) events matching the specified criteria.

Usage

```
event_review(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 TIDKTIME column on the FOLDER table. Merged appeals and dummy data are excluded.

Examples

```
event_review(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, join, 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_substitution	<i>Retrieve a log of Substitution (SUBSTITUTION) events matching the specified criteria.</i>
--------------------	--

Description

Retrieve a log of Substitution (SUBSTITUTION) events matching the specified criteria.

Usage

```
event_substitution(con, join, 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 for appeals where BFSUB is "S". Merged appeals and dummy data are excluded.

Examples

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

event_transcript	<i>Retrieve a log of Transcription (TO_TRANSCRIPT and FROM_TRANSCRIPT) events matching the specified criteria.</i>
------------------	--

Description

Retrieve a log of Transcription (TO_TRANSCRIPT and FROM_TRANSCRIPT) events matching the specified criteria.

Usage

event_transcript(con, join, 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 CONSENT and CONRET columns 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_translation	<i>Retrieve a log of Translation (TRANSLATION_REQ, TO_TRANSLATION and FROM_TRANSLATION) events matching the specified criteria.</i>
-------------------	---

Description

Retrieve a log of Translation (TRANSLATION_REQ, TO_TRANSLATION and FROM_TRANSLATION) events matching the specified criteria.

Usage

```
event_translation(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

REQ uses the LOCDOUT date from events in the PRIORLOC table where the location is "14". TO and FROM use the LOCDOUT and LOCDIN dates, respectively, of events in the PRIORLOC table where the location is "18". 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_translation(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, join, 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, [1](#)
event_activation, [2](#)
event_all, [2](#)
event_assignment, [3](#)
event_cavc, [4](#)
event_certification, [4](#)
event_decision, [5](#)
event_docket, [6](#)
event_drafted, [6](#)
event_endStates, [7](#)
event_form9, [8](#)
event_getDateCols, [8](#)
event_getPriorLocs, [9](#)
event_hearing, [9](#)
event_nod, [10](#)
event_omo, [11](#)
event_outcoding, [11](#)
event_qr, [12](#)
event_remreturn, [13](#)
event_review, [13](#)
event_soc, [14](#)
event_ssoc, [15](#)
event_substitution, [15](#)
event_transcript, [16](#)
event_translation, [17](#)
event_vacolsCreation, [17](#)
event_vso, [18](#)

vacolsConnect, [19](#)