

The TLS231_INPADOC_LEGAL_EVENT Table

The `TLS231_INPADOC_LEGAL_EVENT` table provides comprehensive details on legal events impacting a patent's life cycle, capturing changes both before and after the grant stage. This dataset is sourced from INPADOC (International Patent Documentation), which consolidates legal event information from patent offices worldwide. Key events may include:

- Request for Examination: Indicates when the applicant formally requests examination.
- Renewal Fee Payments: Records ongoing maintenance fees to keep the patent in force.
- Lapse of Patent: Marks patents that have expired due to non-payment or other reasons.
- Change of Ownership: Captures assignments or transfers of patent rights.
- Withdrawal of Application: Reflects applications withdrawn voluntarily by the applicant.
- National Phase Entries: Notes entries into national phases, especially relevant for EP patents.
- Oppositions and Revocations: Details legal challenges to the patent's validity.

For European (EP) patents, this table includes:

- Most legal events from the EP Bulletin.
- Legal events from national offices related to EP patents in national phases.
- Post-grant events managed by the EPO, such as payments, lapses, and reinstatements.

The INPADOC classification scheme was developed to simplify navigation through the extensive INPADOC database, which contains over 250 million records of legal events related to patents. To address the challenge of accessing specific data efficiently, the EPO introduced a structured classification scheme that groups events based on their nature, making the data easier to retrieve and understand.

A high-level classification, following the WIPO ST.27 standard, organizes events into broad categories. There are currently 21 main categories, identified by single letters from A to Z. For example:

Category	Title	Category	Title
A	Application filing	P	Re-publication of document after modification
B	Application discontinuation	Q	Document publication
C	Application revival	R	Party data change
D	Search and examination	S	Information on licensing and similar transactions
E	Pre-grant review request	T	Administrative procedure adjustment
F	IP right grant	U	Payment
G	Protection beyond IP right term	V	Appeal
H	IP right cessation	W	Other
K	IP right revival	Y	Correction and deletion of event information
L	IP right review request	Z	Classification pending
M	IP right maintenance		

Each category groups related events, such as application filings, grant procedures, licensing, and corrections.

```
In [1]: from epo.tipdata.patstat import PatstatClient
from epo.tipdata.patstat.database.models import (
    TLS201_APPLN,
    TLS231_INPADOC_LEGAL_EVENT,
    TLS803_LEGAL_EVENT_CODE

)
from sqlalchemy import and_, case, func, select, distinct

# Initialise the PATSTAT client
patstat = PatstatClient(env="TEST")

# Access ORM
db = patstat.orm()
```

Key Fields in the TLS231_INPADOC_LEGAL_EVENT Table

EVENT_ID

The `EVENT_ID` serves as a unique identifier (primary key) for each legal event recorded in the INPADOC database, ensuring that each event can be distinctly referenced and tracked. This attribute is vital for maintaining stable references, as it remains unchanged across different editions of PATSTAT. This stability is crucial for consistent data analysis over time. `EVENT_ID` is sourced directly from the INPADOC worldwide legal event database and appears as an XML tag in the source data, allowing researchers to link specific legal events to patents effectively. The stable nature of `EVENT_ID` allows it to function as a persistent identifier, which supports detailed historical and procedural analyses of patent lifecycles. With `EVENT_ID`, analysts can reliably retrieve and track events such as requests for examination, fee payments, lapses, ownership changes, and revocations—essential data points in patent life tracking.

APPLN_ID

Identifier for the patent application, representing the formal request for patent protection. This field is a foreign key that references the `APPLN_ID` in the `TSL201_APPLN` table.

EVENT_SEQ_NR

The `EVENT_SEQ_NR` attribute indicates the chronological order of legal events for a particular patent application, assigning a unique sequence number to each event within the lifecycle of the application. It serves as an essential component for organizing and distinguishing between multiple legal events linked to a single patent application, ensuring clear event tracking. Sourced from the INPADOC database, `EVENT_SEQ_NR` enables analysts to reconstruct the sequence of legal events, such as requests, examinations, and renewals, by maintaining the order in which they occurred. This attribute, when combined with `APPLN_ID`, uniquely identifies each row in the dataset, allowing for detailed tracking and identification of individual events associated with a patent. `EVENT_SEQ_NR` values range from 1 and can go up to around 2,000, reflecting the potentially extensive history of legal activities over a patent's lifecycle, with typical maximums in past editions reaching over 1,000.

EVENT_TYPE

The `EVENT_TYPE` attribute identifies whether a legal event is associated with a national/regional phase of an international or regional patent application.

When `EVENT_TYPE` is set to “REG,” it signifies that the event occurred during the national or regional phase of an international application, such as a European Patent (EP) or Patent Cooperation Treaty (PCT) application, and was processed by a national office. This attribute is crucial for understanding where a specific event took place in the patent application process, particularly for applications transitioning from international to national stages.

`EVENT_TYPE` includes cases like the national phase of EP applications and PCT applications, as well as instances where regional stages exist, such as the Russian phase of a Soviet application or the Hong Kong phase of UK or Chinese applications.

Together with the `EVENT_AUTH` attribute, which specifies the triggering authority, `EVENT_TYPE` allows users to trace the jurisdictional scope of events, providing essential context on whether the event impacts the national or broader regional level.

```
In [3]: event_type_query = (
    db.query(
        TLS231_INPADOC_LEGAL_EVENT.event_type
    )
    .distinct()
    .order_by(TLS231_INPADOC_LEGAL_EVENT.event_type)
)

event_type_res = patstat.df(event_type_query)
event_type_res
```

Out [3]:

	event_type
0	
1	REG

An `EVENT_TYPE` value of REG indicates that the legal event occurred during the national phase of either a regional or international application. This designation is provided by the relevant national office and signifies that the event is linked to the processing of the application within a specific jurisdiction.

The REG code applies broadly and is not exclusive to European Patent (EP) applications. It can be found in various contexts, including:

- National Phase of an EP Application: Legal events occurring when an EP application transitions to its national phase in individual member states.
- National Phase of a PCT Application: Events related to international patent applications under the Patent Cooperation Treaty (PCT) as they enter national jurisdictions.
- RU Phase of Earlier SU Applications: Events associated with Russian phases of applications that originated from the former Soviet Union.
- HK Phase of GB Applications: Events occurring in the Hong Kong phase of applications filed in the United Kingdom.
- HK Phase of CN Applications: Legal events in the Hong Kong phase of Chinese patent applications.

In contrast, events with a blank or empty value for `EVENT_TYPE` indicate that they are non-regional or non-national events. These could include international filing events or events that are not directly linked to any national phase processing, such as publication dates or requests for examination that precede entry into a national phase.

If the `EVENT_TYPE` says REG, it means that the event is happening during the national phase of a patent application. This is like the stage when an international or regional patent application is being processed in a specific country. For example:

- If a European patent application is being handled in a country, that's a REG event.
- The same goes for international applications, which are processed in individual countries after they are filed.

If the `EVENT_TYPE` is empty (meaning it doesn't say REG or anything else), it means the event is not tied to a specific country's patent processes. These could be events like:

The announcement of a new patent application. Requests for examination that happen before the application moves to a country for processing.

EVENT_AUTH

The `EVENT_AUTH` attribute identifies the national office responsible for a specific legal event related to a patent. This indicates the source of the legal event. This means that whenever there's a legal event (like a patent application being processed, or a request for examination), this attribute tells us which country's patent office recorded that event.

The `EVENT_AUTH` attribute provides the code of the national office that reported the legal event. Each office has a unique two-letter code, as defined by the World Intellectual Property Organization (WIPO) standard ST.3. This coding system helps ensure clarity and consistency across international patent data.

EVENT_CODE

The `EVENT_CODE` attribute is a critical component of the INPADOC database that categorizes legal events related to patents. Each event code provides specific information about the nature of a legal event, which helps in tracking and managing patent rights effectively.

Name: Legal Event Code - This indicates the unique code assigned to a specific legal event.

The `EVENT_CODE` is a string that uniquely identifies a legal event when combined with the country code of the application. For example, if a code reads PGFP, it signifies that the annual renewal fee for a granted European Patent (EP) was paid in a particular country during its national phase. Notably, because certain events, like the payment of renewal fees, occur annually, only the most recent occurrence is recorded to avoid cluttering the database with repetitive entries. For instance, once the ninth annual fee payment for a French national phase is documented, the record for the eighth payment is removed from the table.

The INPADOC database uses thousands of different codes to classify various legal events that can occur during the lifespan of industrial property rights. Each code can vary based on national patent laws, so it's essential to understand which law applies to the event being referenced. For example, an `EVENT_AUTH` of "AT" combined with an `EVENT_CODE` of "ELJ" indicates that a patent has ceased due to the non-payment of the annual renewal fee in Austria.

Three important event codes that relate specifically to EP patents in their national phases include:

- PGFP: Post Grant Fees Paid – signifies that the annual fees were paid.
- PG25: Lapsed in a Contracting State – indicates that the patent has lapsed in a member state, as announced by the national office to the EPO.
- PGRI: Patent Reinstated in Contracting State – denotes that a previously lapsed patent has been reinstated.

```
In [4]: event_code_query = (
    db.query(
        TLS231_INPADOC_LEGAL_EVENT.event_code
    )
    .distinct()
    .order_by(TLS231_INPADOC_LEGAL_EVENT.event_code)
)

event_code_res = patstat.df(event_code_query)
event_code_res
```

Out[4]:

event_code	
0	None
1	0001
2	109
3	110
4	110E
...	...
1270	Z031
1271	Z072
1272	Z131
1273	ZAAA
1274	ZAAB

1275 rows × 1 columns

This table contains all legal event codes which are used in EPO's worldwide legal event database (also called INPADOC database). More information about the combination of EVENT_CODE and EVENT_AUTH can be found in the TLS803_LEGAL_EVENT_CODE table and in the Weekly updates. These files are updated every Thursday. If Thursday is a public holiday, the update takes place on Friday at this link: <https://www.epo.org/en/searching-for-patents/data/coverage/weekly> (<https://www.epo.org/en/searching-for-patents/data/coverage/weekly>).

EVENT_FILING_DATE

The `EVENT_FILING_DATE` attribute represents the date on which a legal event was filed. This attribute is not commonly populated and typically appears in cases related to patent term extensions, re-examination decisions, or limitations. When no filing date is available, it defaults to `9999-12-31`, serving as a placeholder for unspecified dates. This data originates from the INPADOC database, which tracks legal events globally.

```
In [11]: event_filing_date_query = (
    db.query(
        TLS231_INPADOC_LEGAL_EVENT.aplн_id,
        TLS231_INPADOC_LEGAL_EVENT.event_id,
        TLS231_INPADOC_LEGAL_EVENT.event_filing_date
    )
    .filter(TLS231_INPADOC_LEGAL_EVENT.event_filing_date != '9999-12-31')
    .order_by(TLS231_INPADOC_LEGAL_EVENT.aplн_id)
)

event_filing_date_res = patstat.df(event_filing_date_query)
event_filing_date_res
```

Out[11]:

	appln_id	event_id	event_filing_date
0	2708533	117019895	2010-08-09
1	2708533	114006717	2010-08-09
2	15806316	112123820	2010-08-16
3	15844685	71352271	2008-02-18
4	16900309	113247619	2010-12-13
5	47077688	1036062782	2021-10-21
6	49908752	381982313	2010-10-22
7	50798779	444890480	2012-07-16
8	50894247	1000708615	2018-08-02
9	51206398	428751134	2012-01-05
10	51252914	410322241	2010-01-22
11	51252914	367409461	2010-01-22
12	51252914	362084247	2010-01-22
13	51762969	416369308	2011-05-24
14	53925409	451186506	2012-07-16
15	54174210	592172121	2018-09-21
16	54174210	632665377	2018-09-21
17	55686434	364157912	2014-08-05
18	267348730	389951476	2012-02-20
19	340247551	382021442	2015-12-07
20	415014633	506113775	2017-08-31
21	437625355	533323038	2017-07-19
22	502818387	1004118961	2020-11-30
23	539738173	1064053564	2020-07-10

EVENT_PUBLN_DATE

The `EVENT_PUBLN_DATE` attribute represents the date a legal event was published, such as in an official gazette or on an online platform. This date records when the event information became publicly accessible, offering insight into the timing of procedural updates in the patent's legal status.

EVENT_EFFECTIVE_DATE

The `EVENT_EFFECTIVE_DATE` attribute captures the date on which a legal event officially took effect. This date indicates when the event's impact or legal consequence began, making it essential for understanding the practical enforcement timeline of the event. If no date is available, it defaults to `9999-12-31`.

```
In [6]: effective_date_query = (
    db.query(
        TLS231_INPADOC_LEGAL_EVENT.event_id,
        TLS231_INPADOC_LEGAL_EVENT.appln_id,
        TLS231_INPADOC_LEGAL_EVENT.event_effective_date
    )
    .filter(TLS231_INPADOC_LEGAL_EVENT.event_effective_date != '9
999-12-31')
    .order_by(TLS231_INPADOC_LEGAL_EVENT.appln_id)
)

effective_date_res = patstat.df(effective_date_query)
effective_date_res
```

Out[6]:

	event_id	appln_id	event_effective_date
0	309368459	145	2014-09-26
1	98738803	145	2009-09-18
2	263741582	145	2012-08-31
3	103397003	145	2010-04-30
4	309367586	145	2014-09-26
...
324381	1079209741	606036745	2012-11-06
324382	1079215987	606039115	2022-04-02
324383	1079207446	606039173	2023-12-11
324384	1078694611	606078571	2023-09-28
324385	1078694618	606078580	2023-09-26

324386 rows × 3 columns

EVENT_TEXT

The EVENT_TEXT attribute is valuable for capturing context or nuances about legal events that may not be adequately explained through structured data fields.

```
In [8]: event_text_query = (
    db.query(
        TLS231_INPADOC_LEGAL_EVENT.appln_id,
        TLS231_INPADOC_LEGAL_EVENT.event_seq_nr,
        TLS231_INPADOC_LEGAL_EVENT.event_id,
        TLS231_INPADOC_LEGAL_EVENT.event_effective_date,
        TLS231_INPADOC_LEGAL_EVENT.event_type,
        TLS231_INPADOC_LEGAL_EVENT.event_auth,
        TLS231_INPADOC_LEGAL_EVENT.event_code,
        TLS231_INPADOC_LEGAL_EVENT.event_text
    )
    .filter(TLS231_INPADOC_LEGAL_EVENT.event_text.isnot(None)) #  
Exclude None values
    .filter(TLS231_INPADOC_LEGAL_EVENT.event_effective_date != '9  
999-12-31')
    .order_by(TLS231_INPADOC_LEGAL_EVENT.appln_id)
    .limit(10)
)

event_text_res = patstat.df(event_text_query)
event_text_res
```

Out[8]:

	appln_id	event_seq_nr	event_id	event_effective_date	event_type	event_auth	event_
0	287	33	521126450	2018-01-30	REG	PT	
1	32613	16	106205663	2009-12-10	REG	PT	
2	33215	24	401914668	2016-04-22	REG	PT	
3	39806	14	108073231	2010-04-30	REG	PT	
4	65324	78	113968720	2010-12-02	REG	PT	M
5	65324	24	106200385	2009-01-28	REG	PT	
6	92511	25	228903473	2013-04-18	REG	PT	
7	108742	9	345100762	2011-01-21	REG	GB	
8	124159	11	106205697	2009-04-21	REG	PT	
9	173121	13	146915390	2011-11-24	REG	PT	

```
In [10]: inpadoc_events_query = (
    db.query(
        TLS231_INPADOC_LEGAL_EVENT.event_id,
        TLS231_INPADOC_LEGAL_EVENT.appln_id,
        TLS231_INPADOC_LEGAL_EVENT.event_seq_nr,
        TLS231_INPADOC_LEGAL_EVENT.event_type,
        TLS231_INPADOC_LEGAL_EVENT.event_auth,
        TLS231_INPADOC_LEGAL_EVENT.event_code,
        TLS231_INPADOC_LEGAL_EVENT.event_filing_date,
        TLS231_INPADOC_LEGAL_EVENT.event_publn_date,
        TLS231_INPADOC_LEGAL_EVENT.event_effective_date,
        TLS231_INPADOC_LEGAL_EVENT.event_text
    )
    .order_by(TLS231_INPADOC_LEGAL_EVENT.appln_id)
)

inpadoc_events_res = patstat.df(inpadoc_events_query)
inpadoc_events_res
```

Out[10]:

	event_id	appln_id	event_seq_nr	event_type	event_auth	event_code	even
0	78558748	145	10		EP	AK	
1	79119139	145	14		EP	REF	
2	89177521	145	18		EP	PG25	
3	96544106	145	31		EP	PG25	
4	97254339	145	32		EP	PG25	
...
1904447	1079449639	606045580		2		US	FEPP
1904448	1079449638	606045580		1		US	FEPP
1904449	1079285519	606045580		3		US	STCF
1904450	1078694611	606078571		1		JP	A521
1904451	1078694618	606078580		1		JP	A521

1904452 rows × 10 columns

Reference to patent documents (application or publication)

When a legal event is recorded, it may include references to associated patent documents. This could pertain to either a patent application or a patent publication, but the database does not specify which. These references typically occur in the context of PCT (Patent Cooperation Treaty) applications or EP (European Patent) patents that are republished by regional or national offices. References to documents are given if PCT or EP patents are re-published by a regional / national office:

- PCT applications entering the regional / national phase are typically assigned a new regional / national number
- EP patents get a new national number by some offices (DE, AT, ES, EE, SK and GR).

When a legal event occurs related to a patent, it may involve references to specific patent documents. These documents could be either a patent application (a request to get a patent) or a patent publication (an official document that describes the details of a granted patent).

These references are essential because they help track the history of a patent and any legal changes that happen to it. For example, when a Patent Cooperation Treaty (PCT) application moves into a national phase (which means it is being processed by a specific country), it gets a new identification number for that country. Similarly, when a European Patent (EP) is published in a specific country, it might also receive a new number.

There are different ways these documents can be linked to legal events, and this is done through specific pieces of information called attributes. Depending on the situation, certain attributes will always have information, some may have it, and others will remain empty.

- Document Authority (REF_DOC_AUTH): This tells us which organisation issued the document.
- Document Number (REF_DOC_NR): This provides the identification number of the document.
- Document Type (REF_DOC_KIND): This describes what kind of document it is (like an application or a publication).
- Document Date (REF_DOC_DATE): This shows when the document was published. This can also be included or not, depending on the situation.
- Additional Information (REF_DOC_TEXT): This is a place for any extra details about the document. This is often included but not always.

If a legal event references a document, it will always include the document authority and document number. The document type and date may or may not be included, and there won't be any additional information. In a second case, the document authority and document number are not included, and the document type is provided. The document date might be included, but there will not be any additional information. In a third case, the authority and document number are not applicable, and the document date is not included. However, there will be additional information available.

Attribute	Case 1	Case 2	Case 3
-----------	--------	--------	--------

REF_DOC_AUTH	Yes	No	No
REF_DOC_NR	Yes	No	No
REF_DOC_KIND	Maybe	Yes	No
REF_DOC_DATE	Maybe	Maybe	No
REF_DOC_TEXT	No	No	Yes

REF_DOC_AUTH

The `REF_DOC_AUTH` attribute indicates the publication authority of a referenced document, providing a two-character code that complies with WIPO ST.3, though it does not specify whether the referenced document is an application or a publication.

REF_DOC_NR

The `REF_DOC_NR` attribute represents the serial number of a referenced document, which can include letters and leading zeros

REF_DOC_KIND

The `REF_DOC_KIND` attribute indicates the kind code of the referenced document, which does not clarify whether the document is an application or a publication, and can contain up to 2 ASCII characters.

REF_DOC_DATE

The `REF_DOC_DATE` attribute represents the date of the referenced document, which could be either the application or publication date; however, it does not specify whether the document is an application or a publication. The date is formatted as a date and defaults to `9999-12-31` when not provided.

REF_DOC_TEXT

The `REF_DOC_TEXT` attribute serves as a free text reference to a document, providing additional information about the referenced document. It does not specify whether the document is an application or a publication and can contain up to 1,000 characters. The default value for this attribute is empty when no text is provided.

```
In [9]: ref_doc_query = (
    db.query(
        TLS231_INPADOC_LEGAL_EVENT.ref_doc_auth,
        TLS231_INPADOC_LEGAL_EVENT.ref_doc_nr,
        TLS231_INPADOC_LEGAL_EVENT.ref_doc_kind,
        TLS231_INPADOC_LEGAL_EVENT.ref_doc_date,
        TLS231_INPADOC_LEGAL_EVENT.ref_doc_text,
    )
    .filter(TLS231_INPADOC_LEGAL_EVENT.ref_doc_date != '9999-12-31')
    .limit(10)
)

ref_doc_res = patstat.df(ref_doc_query)
ref_doc_res
```

Out[9]:

	ref_doc_auth	ref_doc_nr	ref_doc_kind	ref_doc_date	ref_doc_text
0	AU	2018357827	A	2018-05-14	None
1	AU	2021286394	A	2021-04-16	None
2		None	A1	2009-11-02	None
3		None	A1	2009-01-01	None
4		None	A1	2009-12-04	None
5		None	A1	2009-04-16	None
6		None	A1	2010-03-02	None
7		None	A1	2009-05-08	None
8	AU	2020390429	A	2020-10-13	None
9	AU	2017294578	A	2017-07-06	None

Party

PARTY_TYPE

The `PARTY_TYPE` attribute specifies the role of a party associated with a legal event. It categorizes the party as either the owner (`OWN`), inventor (`INV`), representative (`REP`), opponent (`OPP`), or licensee (`LIC`), with "`OTH`" used for other roles. This attribute helps identify the type of party involved in various aspects of a patent's lifecycle, such as ownership transfers, opposition proceedings, or licensing agreements. Knowing the specific party type can reveal valuable information about who is responsible for managing, challenging, or licensing the patent rights.

PARTY_SEQ_NR

`PARTY_SEQ_NR` is the sequence number assigned to a party involved in a legal event. This attribute ranges from 0 to 50, but currently, it is limited to values of either 0 or 1. The number indicates the order of parties associated with a particular event, where "`0`" typically represents the primary or only party.

PARTY_NEW

The `PARTY_NEW` attribute captures the name of a party associated with a legal event and may include address details, separated by commas or semicolons. This information is sourced from the INPADOC database and is formatted as free text up to 1,000 characters.

PARTY_OLD

The `PARTY_OLD` attribute provides unstructured text, listing former owners and other related information, potentially including multiple names and addresses. It can contain up to 1,000 characters and is sourced from the INPADOC database.

`PARTY_NEW` and `PARTY_OLD` record changes in ownership or representation, which are important for tracking patent transactions over time. `PARTY_NEW` shows current party details (such as a new owner or representative), while `PARTY_OLD` provides details on former parties. These attributes are especially valuable for identifying ownership history, ensuring the accuracy of ownership records, and providing transparency around the management of intellectual property rights, such as in cases of mergers, acquisitions, or licensing transitions.

```
In [10]: party_changes_query = (
    db.query(
        TLS231_INPADOC_LEGAL_EVENT.appln_id,
        func.count(TLS231_INPADOC_LEGAL_EVENT.party_new).label("new_party_count"),
        func.count(TLS231_INPADOC_LEGAL_EVENT.party_old).label("old_party_count")
    )
    .filter(
        TLS231_INPADOC_LEGAL_EVENT.party_new.isnot(None) | TLS231_INPADOC_LEGAL_EVENT.party_old.isnot(None)
    )
    .group_by(TLS231_INPADOC_LEGAL_EVENT.appln_id)
    .having(func.count(TLS231_INPADOC_LEGAL_EVENT.party_new) > 0)
    .limit(10)
)

party_changes_res = patstat.df(party_changes_query)
party_changes_res
```

Out[10]:

	appln_id	new_party_count	old_party_count
0	565139625	2	2
1	450329911	1	1
2	456636983	4	4
3	456634850	1	1
4	471586649	4	4
5	474398349	1	1
6	474679440	18	18
7	477522111	4	4
8	479263108	4	4
9	479263542	1	1

This query identifies patent applications with party changes by counting the occurrences of new and old parties linked to each application ID. By retrieving both `new_party_count` and `old_party_count`, it offers insight into the turnover of ownership or other party-related changes associated with each patent application.

```
In [13]: detailed_party_changes_with_date_query = (
    db.query(
        TLS231_INPADOC_LEGAL_EVENT.appln_id,
        TLS231_INPADOC_LEGAL_EVENT.event_id,
        TLS231_INPADOC_LEGAL_EVENT.event_effective_date,
        TLS231_INPADOC_LEGAL_EVENT.event_type,
        TLS231_INPADOC_LEGAL_EVENT.event_auth,
        TLS231_INPADOC_LEGAL_EVENT.event_code,
        TLS231_INPADOC_LEGAL_EVENT.party_new,
        TLS231_INPADOC_LEGAL_EVENT.party_old
    )
    .filter(
        TLS231_INPADOC_LEGAL_EVENT.party_new.isnot(None),
        TLS231_INPADOC_LEGAL_EVENT.party_old.isnot(None),
        TLS231_INPADOC_LEGAL_EVENT.event_effective_date != '9999-
12-31'
    )
    .order_by(TLS231_INPADOC_LEGAL_EVENT.appln_id, TLS231_INPADOC_
LEGAL_EVENT.event_effective_date)
    .limit(10)
)

detailed_party_changes_with_date_res = patstat.df(detailed_party_
changes_with_date_query)
detailed_party_changes_with_date_res
```

Out[13]:

	appln_id	event_id	event_effective_date	event_type	event_auth	event_code	
0	145	309368459	2014-09-26	REG	DE	R081	MONIT S'
1	31022	584741296	2019-04-11	REG	BE	PD	GEF
2	31022	631937815	2020-07-30	REG	NL	PD	HECI
3	31022	1000347358	2021-11-22	REG	BE	HC	GEF
4	31022	696805458	2021-11-26	REG	NL	HC	INGENI
5	32613	388302516	2014-03-17	REG	DE	R081	DEUT
6	32613	286624650	2014-03-17	REG	DE	R081	BAYE PRO
7	32613	426035736	2016-08-10	REG	NL	HC	DEUT
8	33215	1036905786	2022-12-20	REG	NL	PD	GLC
9	50723	336967147	2012-11-22	REG	DE	R081	XIA I C

This query provides a breakdown by event for each application, allowing you to compare `PARTY_NEW` and `PARTY_OLD` directly. By examining these values, you can confirm if party information changes across events for the same application or if the parties are simply re-listed in each event without actual changes.

This approach can reveal whether each `appln_id` genuinely experiences shifts in party data over time or if events consistently log the same party information.

SPC (Supplementary Protection Certificate)

SPC_NR

The `SPC_NR` attribute represents the application and/or publication number assigned to a Supplementary Protection Certificate (SPC). The Supplementary Protection Certificate (SPC), which is a special type of patent that provides extra protection for certain pharmaceutical products. This number is essential for uniquely identifying an SPC within the legal framework of patent protection, allowing for easy reference and retrieval of information regarding specific SPCs.

The domain for this attribute allows for up to 40 ASCII characters, accommodating various formats of numbers that may include alphanumeric characters, depending on the issuing authority. As there is no default value assigned, the presence of an `SPC_NR` indicates that a valid SPC has been filed or published.

This attribute is sourced from the INPADOC database, which documents worldwide legal events pertaining to patents, including SPCs. The `SPC_NR` is vital for stakeholders in the pharmaceutical and biotechnology industries, as it aids in tracking the status and legal standing of SPCs, which can significantly impact the commercial exploitation of patented products and the management of intellectual property rights.

SPC_FILING_DATE

The `SPC_FILING_DATE` attribute indicates the filing date of a Supplementary Protection Certificate (SPC), which is an important legal document that extends the protection of certain pharmaceutical and plant protection products beyond the standard patent term. This date is crucial as it marks the official submission of the SPC application to the relevant authority, initiating the process for obtaining additional market exclusivity for a product.

The domain of this attribute is a date format, and it is essential for tracking the timeline of SPC applications. The default value of '`9999-12-31`' serves as a placeholder, indicating that a valid filing date has not been established.

This information is sourced from the INPADOC database, which provides comprehensive coverage of worldwide legal events related to patents. Understanding the `SPC_FILING_DATE` is vital for patent holders and stakeholders in the pharmaceutical industry, as it affects the strategic planning for product launches and market entry, and it influences the overall lifecycle management of a product's intellectual property rights.

SPC_PATENT_EXPIRY_DATE

The `SPC_PATENT_EXPIRY_DATE` attribute indicates the expiry date of the original patent on which a Supplementary Protection Certificate (SPC) is based. This date is crucial for determining the duration of additional protection granted under the SPC regime, which extends the patent rights for a maximum of five years beyond the original patent's expiry.

The domain for this attribute is a date format, and it carries a default value of '`9999-12-31`', which typically signifies that the expiry date is either not applicable or not yet determined. When a valid expiry date is present, it provides stakeholders with important information regarding the lifespan of the underlying patent, which is essential for strategic planning in patent portfolio management and market access for pharmaceutical and biotechnology products.

Sourced from the INPADOC database, which documents worldwide legal events, the `SPC_PATENT_EXPIRY_DATE` is vital for monitoring patent expiration timelines, allowing companies to prepare for the transition to generic competition or to strategize for the continuation of exclusivity in the marketplace.

This attribute should not be confused with attribute `SPC_EXTENSION_DATE`, which is typically after the `SPC_PATENT_EXPIRY_DATE`.

SPC_EXTENSION_DATE

The `SPC_EXTENSION_DATE` attribute represents the last date of validity for a granted Supplementary Protection Certificate (SPC). This date is essential for understanding the full duration of additional protection conferred by the SPC, which can extend the exclusivity of a pharmaceutical or biotechnology product beyond the original patent expiration.

The domain for this attribute is formatted as a date, and it defaults to '`9999-12-31`', indicating that the extension date may not yet be determined or applicable. When a valid extension date is provided, it offers critical information to stakeholders regarding the termination of the SPC's validity, allowing for informed decision-making related to market strategies, product lifecycle management, and potential entry of generic competitors.

Sourced from the INPADOC database, which compiles legal event data from various jurisdictions, the `SPC_EXTENSION_DATE` is a key factor for companies in the pharmaceutical sector as they navigate the complexities of patent law and regulatory frameworks while planning for product launches and exclusivity timelines.

SPC_TEXT

The `SPC_TEXT` attribute provides additional information related to a Supplementary Protection Certificate (SPC) in an unstructured text format. This field can include a variety of details such as product names, product registrations, relevant dates, and other pertinent information that does not fit into predefined categories.

The domain for this attribute allows for up to 1,000 characters, offering sufficient space for comprehensive descriptions. Since the default value is empty, this attribute may not always contain information, but when populated, it can provide valuable insights and context regarding the specific SPC, enhancing the understanding of the product's regulatory status and history.

Sourced from the INPADOC database, the `SPC_TEXT` attribute is particularly useful for stakeholders in the pharmaceutical and biotechnology industries, as it consolidates relevant details that aid in compliance, marketing, and strategic planning efforts related to the products covered by the SPC. By analyzing this free-form text, users can gain deeper insights into the implications of the SPC and the products it protects.

```
In [12]: from datetime import datetime

today_date = datetime.today().date()
# Query to get currently valid SPCs with their filing dates and e
xpiry dates, excluding '9999-12-31'
active_spc_query = (
    db.query(
        TLS231_INPADOC_LEGAL_EVENT.appln_id,
        TLS231_INPADOC_LEGAL_EVENT.spc_nr,
        TLS231_INPADOC_LEGAL_EVENT.spc_filing_date,
        TLS231_INPADOC_LEGAL_EVENT.spc_patent_expiry_date,
        TLS231_INPADOC_LEGAL_EVENT.spc_extension_date
    )
    .filter(
        TLS231_INPADOC_LEGAL_EVENT.spc_nr.isnot(None),
        TLS231_INPADOC_LEGAL_EVENT.spc_filing_date != '9999-12-3
1',
        TLS231_INPADOC_LEGAL_EVENT.spc_patent_expiry_date != '999
9-12-31',
        TLS231_INPADOC_LEGAL_EVENT.spc_extension_date != '9999-12
-31',
        TLS231_INPADOC_LEGAL_EVENT.spc_patent_expiry_date >= toda
y_date, # Check if the patent expiry date is today or later
        TLS231_INPADOC_LEGAL_EVENT.spc_extension_date >= today_da
te # Check if the extension date is today or later
    )
    .order_by(TLS231_INPADOC_LEGAL_EVENT.appln_id)
)

active_spc_res = patstat.df(active_spc_query)
active_spc_res
```

Out [12]:

The query identifies SPCs that are currently active by checking that they have valid filing dates and that neither the original patent expiry date nor any extension has passed. This result is useful for analyzing SPCs that still confer protection as of today.

Designated states and extension states

DESIGNATED_STATES

The DESIGNATED_STATES attribute represents a list of two-letter country codes that indicate the states designated in a patent application. This list is essential for identifying the jurisdictions in which the patent rights are sought or recognized, providing clarity on the geographical scope of protection.

The domain for this attribute allows for up to 1,000 ASCII characters and consists of an alphabetically ordered list of two-character country codes, as defined by the WIPO Standard ST.3. Each code in the list is separated by a comma (","). For example, valid entries may look like "FR" for France, "AT,DE" for Austria and Germany, or "DE,FR,GB,NL" for Germany, France, Great Britain, and the Netherlands.

The default value for this attribute is empty, indicating that no designated states have been specified. When populated, the DESIGNATED_STATES attribute serves as a critical reference for stakeholders, including patent holders and legal practitioners, as it clarifies the extent of the patent's territorial coverage.

This information is sourced from the INPADOC database, which documents worldwide legal events related to patents. Understanding the DESIGNATED_STATES is vital for patent strategy, as it influences decisions regarding market entry, enforcement of patent rights, and compliance with the legal requirements of different jurisdictions. By analyzing this attribute, users can assess the international reach of a patent and navigate the complexities of patent law across various countries.

EXTENSION_STATES

The EXTENSION_STATES attribute signifies a list of country codes representing extension states where patent protection can be extended beyond the member states of a regional patent office. This feature is particularly relevant for applicants who wish to expand the geographical scope of their patent rights to non-member states through mechanisms provided by regional offices, such as the European Patent Office (EPO).

The domain for this attribute allows for up to 30 ASCII characters, formatted as an alphabetically ordered list of two-character country codes, in accordance with the WIPO Standard ST.3. Each country code in the list is separated by a comma (","). For instance, valid entries could include "ME" for Montenegro or "BA,ME" for Bosnia and Herzegovina and Montenegro.

The default value for the `EXTENSION_STATES` attribute is empty, indicating that no extension states have been specified. When populated, this attribute is crucial for applicants and legal professionals, as it clarifies the additional jurisdictions where patent rights may be enforced, thereby enhancing strategic planning for patent filings and market entry.

This information is drawn from the broader context of patent law and practice, specifically focusing on the capabilities of regional patent offices to facilitate the extension of patent protections. Understanding the `EXTENSION_STATES` attribute is essential for stakeholders in the intellectual property field, as it provides insights into the opportunities available for extending patent coverage, ensuring compliance with international patent law, and optimizing the management of patent portfolios across different jurisdictions.

```
In [13]: grouped_designated_and_extension_query = (
    db.query(
        TLS231_INPADOC_LEGAL_EVENT.appln_id,
        TLS231_INPADOC_LEGAL_EVENT.event_id,
        TLS231_INPADOC_LEGAL_EVENT.designated_states,
        TLS231_INPADOC_LEGAL_EVENT.extension_states
    )
    .order_by(TLS231_INPADOC_LEGAL_EVENT.appln_id)
    .filter(TLS231_INPADOC_LEGAL_EVENT.designated_states != None)
)

grouped_designated_and_extension_res = patstat.df(grouped_designated_and_extension_query)
grouped_designated_and_extension_res
```

Out [13]:

	<code>appln_id</code>	<code>event_id</code>	<code>designated_states</code>	<code>extensi...</code>
0	145	78558748	AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HU,I...	
1	145	69979341	AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HU,I...	
2	145	78427049	AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HU,I...	
3	146	68196024	AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HU,I...	
4	146	77877703	AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HU,I...	
...
41025	600215002	1073902693	AL,AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,H...	
41026	600328169	1072264590	AL,AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,H...	
41027	600479748	1072264583	AL,AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,H...	
41028	600479749	1072264588	AL,AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,H...	
41029	603309930	1076915402	AL,AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,H...	

41030 rows × 4 columns

In [17]: `import pandas as pd`

```
# Query to get designated states per application ID and event ID
designated_states_comparison_query = (
    db.query(
        TLS231_INPADOC_LEGAL_EVENT.appln_id,
        TLS231_INPADOC_LEGAL_EVENT.event_id,
        TLS231_INPADOC_LEGAL_EVENT.designated_states
    )
    .filter(TLS231_INPADOC_LEGAL_EVENT.designated_states.isnot(None))
    .order_by(TLS231_INPADOC_LEGAL_EVENT.appln_id, TLS231_INPADOC_LEGAL_EVENT.event_id) # Order by application ID and event ID
)

# Execute the query and retrieve results
designated_states_comparison_res = patstat.df(designated_states_comparison_query)

# Group by application ID and aggregate designated states into a list
grouped_designated_states = (
    designated_states_comparison_res
    .groupby('appln_id')['designated_states']
    .agg(lambda x: list(set(x))) # Use set to avoid duplicates
    .reset_index()
)

# Initialize a list to store the consistency results
consistency_results = []

# Iterate over the grouped DataFrame to check consistency
for index, row in grouped_designated_states.iterrows():
    appln_id = row['appln_id']
    states_list = row['designated_states']

    # Check if all designated states are the same
    is_consistent = len(set(tuple(sorted(states))) for states in states_list)) == 1

    consistency_results.append({
        'appln_id': appln_id,
        'consistent': is_consistent,
        'designated_states': states_list
    })

consistency_df = pd.DataFrame(consistency_results)
consistency_df
```

Out[17]:

	appln_id	consistent	designated_states
0	145	True	[AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HU,...
1	146	True	[AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HU,...
2	186	True	[AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HR,...
3	287	False	[AT,BE,BG,CH,CY,LI, AT,BE,BG,CH,CY,CZ,DE,DK,EE...
4	620	True	[AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HU,...
...
20380	600215002	True	[AL,AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,...
20381	600328169	True	[AL,AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,...
20382	600479748	True	[AL,AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,...
20383	600479749	True	[AL,AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,...
20384	603309930	True	[AL,AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,...

20385 rows × 3 columns

This code retrieves the list of designated states for each application ID and event ID from the `TLS231_INPADOC_LEGAL_EVENT` table, filtering for non-null values. It then checks if the designated states are consistent across events for each application by comparing the sets of states, and stores the consistency check results in a DataFrame.

```
In [3]: # Query to get designated states for a specific application ID (287) across all its events
designated_states_specific_app_query = (
    db.query(
        TLS231_INPADOC_LEGAL_EVENT.appln_id,
        TLS231_INPADOC_LEGAL_EVENT.event_id,
        TLS231_INPADOC_LEGAL_EVENT.event_effective_date,
        TLS231_INPADOC_LEGAL_EVENT.event_type,
        TLS231_INPADOC_LEGAL_EVENT.event_auth,
        TLS231_INPADOC_LEGAL_EVENT.event_code,
        TLS231_INPADOC_LEGAL_EVENT.event_text,
        TLS231_INPADOC_LEGAL_EVENT.designated_states,
        TLS803_LEGAL_EVENT_CODE.event_category_code,
        TLS803_LEGAL_EVENT_CODE.event_category_title
    )
    .join(
        TLS803_LEGAL_EVENT_CODE,
        (TLS231_INPADOC_LEGAL_EVENT.event_auth == TLS803_LEGAL_EVENT_CODE.event_auth) &
        (TLS231_INPADOC_LEGAL_EVENT.event_code == TLS803_LEGAL_EVENT_CODE.event_code)
    )
    .filter(TLS231_INPADOC_LEGAL_EVENT.appln_id == 287) # Filter for application ID 287
    .filter(TLS231_INPADOC_LEGAL_EVENT.designated_states.isnot(None)) # Ensure designated states are not None
    .order_by(TLS231_INPADOC_LEGAL_EVENT.event_id) # Order by event ID for clarity
)

designated_states_specific_app_res = patstat.df(designated_states_specific_app_query)
designated_states_specific_app_res
```

Out[3]:

	appln_id	event_id	event_effective_date	event_type	event_auth	event_code	event_te
0	287	78552005	9999-12-31		EP	AK	No
1	287	310756131	9999-12-31		EP	AK	No
2	287	354349761	9999-12-31		EP	RBV	No
3	287	358384208	9999-12-31		EP	AKX	No
4	287	361439941	9999-12-31		EP	RBV	No
5	287	496340656	9999-12-31		EP	AK	No

Fee payments

FEE_COUNTRY

The attribute `FEE_COUNTRY` specifies the country or territory where the annual renewal fee for a patent has been paid. This field is particularly useful for tracking the distribution of fee payments across various national offices, as it shows which country is responsible for processing the renewal of a patent in its jurisdiction. The value follows WIPO Standard ST.3, using two-character ASCII codes, like "DE" for Germany or "FR" for France.

`FEE_COUNTRY` is only populated when the event code is "PGFP" (Post Grant Fee Paid for an EP patent) or "VSFP" (Annual Fee Paid to Validation State). For European patents, renewal fees are not paid directly to the EPO but to the national office of each EPO member state where the patent is valid. This attribute helps provide insights into the territorial scope of EP patents and the fees required to maintain their validity in specific countries.

FEE_PAYMENT_DATE

The `FEE_PAYMENT_DATE` attribute records the date on which the annual renewal fee was paid for a European (EP) patent. This date is essential for tracking the timeliness of renewal payments, which help to maintain the validity of a patent within designated countries. Represented in date format, it captures when the payment was completed and is used to verify compliance with national or regional renewal requirements.

The default value for this attribute is "9999-12-31," which indicates an unspecified or placeholder date when no actual payment date is recorded. This attribute is sourced from the INPADOC database and reflects fee-payment events documented by national patent offices, providing important temporal insights into patent maintenance activities across various jurisdictions.

FEE_RENEWAL_YEAR

The `FEE_RENEWAL_YEAR` attribute indicates the specific year of the renewal cycle for which the annual renewal fee was paid for a European (EP) patent. Represented as a number from 1 to 25, it reflects the year of maintenance, with common entries up to 20, while additional years are occasionally present for cases like supplementary protection certificates (SPCs).

Unlike the `FEE_PAYMENT_DATE`, which marks the actual payment date, `FEE_RENEWAL_YEAR` only records the applicable year in the patent's lifecycle for which the fee was paid. For example, if a renewal fee was paid on April 1, 2020, for the seventh year of renewal, the value of `FEE_RENEWAL_YEAR` would be "7" rather than "2020."

Notably, some national patent offices do not require payments for certain years of maintenance. For instance, the Netherlands, Switzerland, Great Britain, Austria, San Marino, and Italy have exemptions from renewal payments for specific early years, which may still generate renewal events (PGFP) if needed for procedural reasons, like translation filing. The INPADOC database provides this information, capturing only the most recent payment record per patent.

Exceptions as of 2018 are:

- Netherlands have no fee for OY3.
- Switzerland has no fee for OY3.
- Great Britain has no fee for OY3 and 4.
- Austria has no fee for OY 3, 4 and 5.
- San Marino has no fee in year 3, but since they require a translation to be filed, there are PGFP's for OY3 if needed in the year of due payment
- Italy has no fee in years 3 or 4, but since they require a translation to be filed, there are PGFP's for OY3 & 4 if needed in the year of due payment

FEE_TEXT

The `FEE_TEXT` attribute provides additional, free-form information regarding an annual renewal fee payment for a European (EP) patent. This text can include specific remarks

```
In [19]: fee_attributes_query = (
    db.query(
        TLS231_INPADOC_LEGAL_EVENT.appln_id,
        TLS231_INPADOC_LEGAL_EVENT.event_id,
        TLS231_INPADOC_LEGAL_EVENT.fee_country,
        TLS231_INPADOC_LEGAL_EVENT.fee_payment_date,
        TLS231_INPADOC_LEGAL_EVENT.fee_renewal_year,
        TLS231_INPADOC_LEGAL_EVENT.fee_text
    )
    .filter(TLS231_INPADOC_LEGAL_EVENT.fee_country.isnot(None),
            TLS231_INPADOC_LEGAL_EVENT.fee_payment_date != '9999-12-31')
    .order_by(TLS231_INPADOC_LEGAL_EVENT.appln_id) # Order by application ID
)

fee_attributes_res = patstat.df(fee_attributes_query)

fee_attributes_res
```

Out[19]:

	appln_id	event_id	fee_country	fee_payment_date	fee_renewal_year	fee_text
0	145	99302616	AT	2009-08-20	3	None
1	145	693531927	DE	2021-07-19	15	None
2	146	1071483673	DE	2023-08-22	17	None
3	146	1061225358	IT	2017-08-16	17	None
4	146	1067922640	ES	2023-09-18	17	None
...
53169	562100667	1077073288	FR	2023-12-26	3	None
53170	562187360	1077073171	DE	2023-12-12	3	None
53171	563240382	1077080184	FR	2023-11-22	3	None
53172	563240382	1077080182	DE	2023-11-21	3	None
53173	568220999	1078292740	FI	2023-12-19	3	None

53174 rows × 6 columns

Lapses

LAPSE_COUNTRY

The `LAPSE_COUNTRY` attribute indicates the country or territory where a granted European patent application has lapsed. Represented by two-character country codes as defined by WIPO Standard ST.3, this attribute identifies the specific office that recorded the lapse. The field is only populated when the `EVENT_CODE` equals PG25, VS25, or PG2D, signifying a lapse event. The value is empty for all other event codes. This attribute is sourced from the INPADOC legal event database, particularly within the notification section of lapse-related legal events.

LAPSE_DATE

The `LAPSE_DATE` attribute represents the effective date on which a patent's legal protection in a particular country or region ceased. This date is critical as it marks when the patent holder's rights expired, allowing others in that jurisdiction to use the invention without infringing the patent. The effective lapse date is based on either missed renewal fees or the expiration of the statutory patent term, depending on the jurisdiction's requirements.

In the data, the `LAPSE_DATE` is only filled when the `EVENT_CODE` is either "PG25" (for post-grant lapse of an EP patent in designated states) or "VS25" (for a lapse in validation states). If no lapse occurs, or if the patent is still active, this attribute defaults to a placeholder date, typically 9999-12-31.

LAPSE_TEXT

The `LAPSE_TEXT` attribute provides additional context and information regarding the lapse of a patent. This text is stored as free-form text and can include various details related to the circumstances surrounding the lapse, such as reasons for the lapse, notifications issued to the patent holder, or any pertinent legal commentary.

This attribute is relevant for patent stakeholders seeking a deeper understanding of the factors contributing to a patent's lapse. The `LAPSE_TEXT` field is populated only when the `EVENT_CODE` corresponds to "PG25" (indicating a post-grant lapse of an EP patent in designated states) or "VS25" (indicating a lapse in validation states). If no lapse has occurred or the information is not applicable, this field remains empty, indicating that there is no additional commentary or detail available.

```
In [20]: lapse_attributes_query = (
    db.query(
        TLS231_INPADOC_LEGAL_EVENT.appln_id,
        TLS231_INPADOC_LEGAL_EVENT.event_id,
        TLS231_INPADOC_LEGAL_EVENT.lapse_country,
        TLS231_INPADOC_LEGAL_EVENT.lapse_date,
        TLS231_INPADOC_LEGAL_EVENT.lapse_text
    )
    .filter(
        TLS231_INPADOC_LEGAL_EVENT.lapse_country.isnot(None),
        TLS231_INPADOC_LEGAL_EVENT.lapse_date != '9999-12-31'
    )
    .order_by(TLS231_INPADOC_LEGAL_EVENT.appln_id, TLS231_INPADOC_LEGAL_EVENT.event_id)
)

lapse_attributes_res = patstat.df(lapse_attributes_query)
lapse_attributes_res
```

Out [20]:

	appln_id	event_id	lapse_country	lapse_date	lapse_text
0	145	85583233	LT	2008-12-17	LAPSE BECAUSE OF FAILURE TO SUBMIT A TRANSLATI...
1	145	89177519	FI	2008-12-17	LAPSE BECAUSE OF FAILURE TO SUBMIT A TRANSLATI...
2	145	89177520	LV	2008-12-17	LAPSE BECAUSE OF FAILURE TO SUBMIT A TRANSLATI...
3	145	89177521	NL	2008-12-17	LAPSE BECAUSE OF FAILURE TO SUBMIT A TRANSLATI...
4	145	89177522	PL	2008-12-17	LAPSE BECAUSE OF FAILURE TO SUBMIT A TRANSLATI...
...
261945	570596329	1078297589	IS	2023-10-07	LAPSE BECAUSE OF FAILURE TO SUBMIT A TRANSLATI...
261946	570596329	1078297592	PT	2023-10-09	LAPSE BECAUSE OF FAILURE TO SUBMIT A TRANSLATI...
261947	570596329	1078297595	RO	2023-06-07	LAPSE BECAUSE OF FAILURE TO SUBMIT A TRANSLATI...
261948	570596329	1078297598	SK	2023-06-07	LAPSE BECAUSE OF FAILURE TO SUBMIT A TRANSLATI...
261949	570596329	1078297600	SM	2023-06-07	LAPSE BECAUSE OF FAILURE TO SUBMIT A TRANSLATI...

261950 rows × 5 columns

Reinstatements

Reinstatement in the context of patents is the process by which a lapsed or abandoned patent application or granted patent is restored to its active status. This can happen if the patent holder missed a deadline or failed to pay a required fee, causing the patent to lapse. However, in certain situations, the patent office allows the patent holder to reinstate the patent if they provide a valid reason, like an unintentional oversight, and complete any necessary formalities or payments.

Once reinstated, the patent or application regains its original protection status, meaning it's as if it was never lapsed, and the owner can continue to enforce their patent rights.

REINSTATE_COUNTRY

The `REINSTATE_COUNTRY` attribute indicates the country or territory where a patent application has been reinstated, representing the office responsible for the reinstatement. It includes the application authority's country code or "WO" for PCT (Patent Cooperation Treaty) applications. This field is populated only when the `EVENT_CODE` is "PGRI" (indicating a reinstatement event), following the standard two-character country codes defined by WIPO Standard ST.3.

REINSTATE_DATE

The `REINSTATE_DATE` attribute represents the effective date when a patent application was reinstated. This field records the reinstatement date only if the event code is "PGRI" (indicating a reinstatement event). The date is formatted as a standard date, with a default value of "9999-12-31" if not applicable.

REINSTATE_TEXT

The `REINSTATE_TEXT` attribute provides supplementary information in free-text format about the reinstatement of a patent application. It is filled only when the event code is "PGRI," signifying a reinstatement event. This field can hold up to 1,000 characters, but as of the latest update, it has not yet been populated in any records.

```
In [21]: reinstatement_query = (
    db.query(
        TLS231_INPADOC_LEGAL_EVENT.appln_id,
        TLS231_INPADOC_LEGAL_EVENT.event_id,
        TLS231_INPADOC_LEGAL_EVENT.event_code,
        TLS231_INPADOC_LEGAL_EVENT.reinstate_country,
        TLS231_INPADOC_LEGAL_EVENT.reinstate_date,
        TLS231_INPADOC_LEGAL_EVENT.reinstate_text
    )
    .filter(TLS231_INPADOC_LEGAL_EVENT.event_code == "PGRI")
    .order_by(TLS231_INPADOC_LEGAL_EVENT.appln_id)
)

reinstatement_results = patstat.df(reinstatement_query)
reinstatement_results
```

Out[21]:

	appln_id	event_id	event_code	reinstate_country	reinstate_date	reinstate_text
0	40753	617465846	PGRI	ES	2020-03-18	None
1	136521	134638297	PGRI	IT	2011-05-01	None
2	136521	127357720	PGRI	IT	2011-05-01	None
3	228637	541160091	PGRI	IT	2017-06-16	None
4	228637	485167937	PGRI	IT	2017-06-16	None
...
106	504824143	1068333342	PGRI	CH	2023-08-26	None
107	504824143	1068333351	PGRI	LI	2023-08-26	None
108	505177418	1040005836	PGRI	LT	2022-12-23	None
109	506617443	1068366107	PGRI	LI	2023-08-22	None
110	506617443	1068366095	PGRI	CH	2023-08-22	None

111 rows × 6 columns

Patent classification

CLASS_SCHEME

The CLASS_SCHEME attribute indicates the scheme of the classification used in a patent event, specifically detailing whether the patent is classified under the International Patent Classification (IPC) system. It has a limited domain, with "IPC" as the primary value and an empty string as the default when classification data is not available or applicable. This attribute helps clarify which classification system is applied to the patent document in question, supporting standardized referencing and analysis across patents.

CLASS_SYMBOL

The CLASS_SYMBOL attribute refers to the corrected classification symbol of a patent, typically presented in the WIPO ST.8 format. This symbol is used to classify patents according to the International Patent Classification (IPC) system. It can be up to 50 ASCII characters long and may vary in structure; some symbols follow the structured WIPO ST.8 format, while others may be unstructured. The default value for this attribute is empty, indicating that the classification symbol may not be provided in some cases. This attribute plays a crucial role in identifying the technical area of the invention within the patent system.

```
In [4]: classification_query = (
    db.query(
        TLS231_INPADOC_LEGAL_EVENT.appln_id,
        TLS231_INPADOC_LEGAL_EVENT.event_id,
        TLS231_INPADOC_LEGAL_EVENT.event_code,
        TLS231_INPADOC_LEGAL_EVENT.event_effective_date,
        TLS231_INPADOC_LEGAL_EVENT.class_scheme,
        TLS231_INPADOC_LEGAL_EVENT.class_symbol
    )
    .filter(
        TLS231_INPADOC_LEGAL_EVENT.class_symbol != None,
        TLS231_INPADOC_LEGAL_EVENT.event_effective_date != '9999-12-31'
    )
    .order_by(TLS231_INPADOC_LEGAL_EVENT.appln_id, TLS231_INPADOC_LEGAL_EVENT.event_id)
)

classification_results = patstat.df(classification_query)
classification_results
```

Out [4]:

	appln_id	event_id	event_code	event_effective_date	class_scheme	class_sym
0	11660859	336818599	R079	2011-12-09	IPC	F16D00651800
1	12946211	227236910	R079	2013-03-20	IPC	G01N00290700
2	12962841	286099046	R079	2014-04-02	IPC	F16J00154600
3	12962841	319840549	R079	2014-04-02	IPC	F16J00155400

4	12962841	319842510	R079	2014-04-02	IPC	F16J0015540()
5	13484302	351938249	R079	2015-04-22	IPC	H02S0040380()
6	13643318	374327810	R079	2003-01-08	IPC	F03D0003040()
7	13727869	373418977	R079	1999-06-15	IPC	F03B0003040()
8	14993321	250001248	R079	2011-11-07	IPC	G01M0099000()
9	14998454	298990489	R079	2014-06-30	IPC	F03D0011000()
10	15030848	274096549	R079	2013-12-11	IPC	E05B0081240()
11	15031587	228657545	R079	2013-03-21	IPC	F16D0059020()
12	15035142	281734549	R079	2014-02-05	IPC	H02S0040300()
13	15063151	281735574	R079	2014-02-03	IPC	H02S0050000()
14	15081701	306575778	R079	2014-08-29	IPC	A01M0029160()
15	15092091	235232605	R079	2013-04-26	IPC	H02P0029000()
16	15124173	249660614	R079	2011-12-01	IPC	A01M0029160()
17	15138397	299707682	R079	2014-07-04	IPC	E05B0085020()
18	15739397	316707267	R079	2014-10-30	IPC	F01D0005160()
19	57289076	282747086	R079	2014-02-12	IPC	F16H0057033()
20	274170528	180837030	R079	2012-07-26	IPC	F16L0037140()
21	274170528	303244255	R079	2012-08-27	IPC	F16L0037084()
22	274879479	282773254	R079	2014-02-06	IPC	H02S0040300()
23	319379076	198044116	R079	2012-09-28	IPC	A01M0029160()
24	320410121	312423848	R079	2011-11-07	IPC	F16H0003660()
25	325205410	275731173	R079	2013-12-13	IPC	E05B0085020()
26	330421815	219838011	R079	2011-12-21	IPC	F03D0011040()
27	331358505	219843552	R079	2011-12-09	IPC	F16D0065180()
28	335934205	250051115	R079	2011-12-09	IPC	F16D0065180()
29	336609934	216416128	R079	2013-01-17	IPC	H02K0021240()
30	339191728	281741756	R079	2014-02-05	IPC	H02S0050100()
31	342101864	314628892	R079	2014-10-24	IPC	F16P0003120()
32	353627993	281738197	R079	2014-01-30	IPC	H02S0040360()
33	364028575	282772837	R079	2014-02-07	IPC	H02S0050100()
34	365536955	274108083	R079	2013-12-11	IPC	B23K0026067()
35	376508081	275739153	R079	2013-12-13	IPC	B23K0026342()
36	378210971	248453648	R079	2013-07-16	IPC	B63H0009000()
37	378211830	282772943	R079	2014-02-06	IPC	H02S0050100()

38	379470397	282769765	R079	2014-02-06	IPC	H02S00403000
39	405070606	282764440	R079	2014-02-10	IPC	H02S00101200
40	405070606	282774179	R079	2012-06-25	IPC	H02N00060000
41	406804874	281740098	R079	2014-02-05	IPC	H02S00302000
42	407147664	333931806	R079	2015-01-28	IPC	F16B00110000
43	407147664	333934519	R079	2012-08-27	IPC	F16L00050000
44	409464819	295284012	R079	2014-06-18	IPC	H02K00210200
45	420727754	333924861	R079	2015-01-30	IPC	H02S00200000