

What Is RDAP (and Why It's More Important Now Than Ever)

Posted on March 11, 2025

Conversations around replacing WHOIS – the main protocol for retrieving information about domain registrants – have been around for decades. Now, WHOIS is being phased out and replaced by the Registration Data Access Protocol (RDAP).

Why is that happening?

WHOIS lacked standardized data formatting and had design flaws such as the lack of authentication or the inability to transmit data securely. It was clear that WHOIS needed to evolve—or be replaced. Different working groups at the Internet Engineering Task Force (IETF) had been quite busy to that end.

Around 2003, one group started working on the Internet Registry Information Service (IRIS)—a protocol aimed at addressing WHOIS' lack of standardized data format by using XML. The standard was [finalized in 2008](#), but, while technically sound, IRIS didn't gain widespread adoption.

IETF recognized that IRIS was perhaps too complex and created a different working group, Web Extensible Internet Registration Data Services (WEIRDS), that led another round of WHOIS modernization efforts. These eventually paved the way for the Requests for Comments (RFCs, IETFs preferred format for working on Internet standards) that define RDAP, which was standardized in 2015.

Fast forward to today, RDAP has become the clear successor to WHOIS. On January 28, 2025, ICANN sent a clear signal that the WHOIS era has ended – they've deprecated [WHOIS port 43](#).

This shift has understandably reignited interest in RDAP. In this post, we will explore this protocol, how it works, and its benefits over WHOIS.

What Is RDAP?

RDAP is a protocol for accessing current domain registration data. It essentially replaces WHOIS and addresses the 43-year-old protocol's shortcomings by providing:

- Structured data in JSON format
- Improved security through HTTPS
- User authentication and role-based access control
- Better support for internationalization

How Does RDAP Work?

RDAP is a REST-based protocol—it operates using HTTP and allows clients to make requests to retrieve registration data in a structured JSON format. We'll take a closer look at how queries and responses work, but before that, we need to tackle a concept that is foundational for RDAP – bootstrapping and bootstrap discovery.

RDAP Bootstrapping and Bootstrap Discovery

Domain data is managed by different organizations (e.g., domain registries). For example, if you want to get the registration details for `example[.]com`, the request will be routed to Verisign, the company in charge of the `.com` gTLD. Queries for `example[.]info` (note: this site doesn't exist) will be sent to Afilias, while InterNIC is responsible for the registration data of `example[.]top` (this domain doesn't exist either; all "example." second level domains are unregistrable and reserved for documentation purposes).

Now, since domain owners buy and register domains at the registrar level, registrars also handle domain registration data. So while registries are the authoritative source for domain registration

data, they may point queries to the registrar handling the specific domain's information.

With thousands of registries managing different TLDs, routing requests to the right organization might be challenging. Also, as an RDAP end-user, how are you supposed to know which server to query? This is where IANA's bootstrap registries come in. IANA maintains bootstrap files to determine which RDAP servers (i.e., operated by a registry or registrar) handle specific registration data.

So when an end-user asks an RDAP client (i.e., a command line utility, lookup tool, or web-based interface) to perform a domain registration lookup, the client fetches the correct RDAP server from <https://data.iana.org/rdap/> and then sends the request to it. This process is called bootstrap discovery.

This differs significantly from how WHOIS worked, as WHOIS clients had to be preconfigured to know the correct server for each TLD.

RDAP Query

Now that the client knows which RDAP server to route the request to, it constructs an RDAP query using HTTP. It's important to note that RDAP enforces encryption, requiring HTTPS for all communication. This ensures that the data transmitted between the client and server is protected from eavesdropping.

The query's URL structure consists of the base URL obtained from the bootstrap registry, the specific resource identifier (e.g., the domain name, IP address, or ASN), and the resource itself.

Here're a few of working examples of RDAP queries:

```
https://rdap.verisign.com/com/v1/domain/example.com
```

```
https://rdap.publicinterestregistry.org/rdap/domain/example.org
```

```
http://rdap.apnic.net/ip/2001:dc0:2001:11::194
```

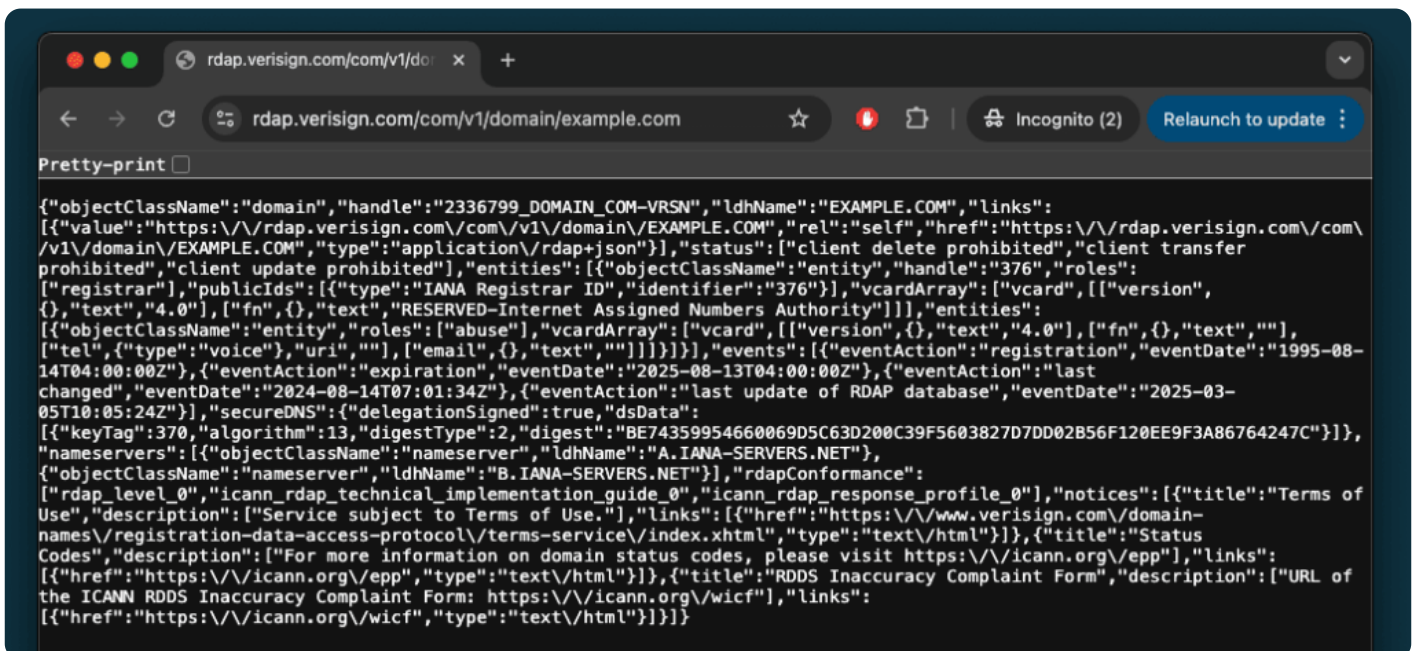
However, as you can see, to form a query, you need to know the address of the RDAP server,

which is different for every TLD. Curiously, unlike WHOIS server names, RDAP server names are not standardized – so there’s no way to guess them. That’s inconvenient – and that’s exactly what the [lookup tools](#) solve for you by performing the bootstrap discovery

RDAP Response

When the RDAP server receives the query, it retrieves the requested registration data and formats the responses as a JSON object.

For example, the response for the first query from the three above looks like this:

A screenshot of a web browser window showing a RDAP response in JSON format. The browser's address bar displays "rdap.verisign.com/com/v1/domain/example.com". The page content is a large block of JSON text, which is partially obscured by a "Pretty-print" button. The visible JSON includes fields like "objectClassName", "handle", "ldhName", "links", "status", "entities", "publicIds", "events", "secureDNS", "nameservers", and "rdapConformance".

```
{
  "objectClassName": "domain",
  "handle": "2336799_DOMAIN_COM-VRSN",
  "ldhName": "EXAMPLE.COM",
  "links": [
    {
      "value": "https://rdap.verisign.com/com/v1/domain/EXAMPLE.COM",
      "rel": "self",
      "href": "https://rdap.verisign.com/com/v1/domain/EXAMPLE.COM",
      "type": "application/rdap+json"
    }
  ],
  "status": [
    "client delete prohibited",
    "client transfer prohibited",
    "client update prohibited"
  ],
  "entities": [
    {
      "objectClassName": "entity",
      "handle": "376",
      "roles": [
        "registrar"
      ],
      "publicIds": [
        {
          "type": "IANA Registrar ID",
          "identifier": "376"
        }
      ],
      "vcardArray": [
        {
          "version": "4.0",
          "fn": "RESERVED-Internet Assigned Numbers Authority"
        }
      ],
      "events": [
        {
          "eventAction": "registration",
          "eventDate": "1995-08-14T04:00:00Z"
        },
        {
          "eventAction": "expiration",
          "eventDate": "2025-08-13T04:00:00Z"
        },
        {
          "eventAction": "last changed",
          "eventDate": "2024-08-14T07:01:34Z"
        },
        {
          "eventAction": "last update of RDAP database",
          "eventDate": "2025-03-05T10:05:24Z"
        }
      ],
      "secureDNS": {
        "delegationSigned": true,
        "dsData": [
          {
            "keyTag": 370,
            "algorithm": 13,
            "digestType": 2,
            "digest": "BE74359954660069D5C63D200C39F5603827D7DD02B56F120EE9F3A86764247C"
          }
        ]
      },
      "nameservers": [
        {
          "objectClassName": "nameserver",
          "ldhName": "A.IANA-SERVERS.NET"
        },
        {
          "objectClassName": "nameserver",
          "ldhName": "B.IANA-SERVERS.NET"
        }
      ],
      "rdapConformance": [
        "rdap_level_0",
        "icann_rdap_technical_implementation_guide_0",
        "icann_rdap_response_profile_0"
      ],
      "notices": [
        {
          "title": "Terms of Use",
          "description": "Service subject to Terms of Use.",
          "links": [
            {
              "href": "https://www.verisign.com/domain-names/registration-data-access-protocol/terms-service/index.xhtml",
              "type": "text/html"
            }
          ]
        },
        {
          "title": "Status Codes",
          "description": "For more information on domain status codes, please visit https://icann.org/epp",
          "links": [
            {
              "href": "https://icann.org/epp",
              "type": "text/html"
            }
          ]
        },
        {
          "title": "RDDS Inaccuracy Complaint Form",
          "description": "URL of the ICANN RDDS Inaccuracy Complaint Form: https://icann.org/wicf",
          "links": [
            {
              "href": "https://icann.org/wicf",
              "type": "text/html"
            }
          ]
        }
      ]
    }
  ]
}
```

It’s perfectly machine-readable but not very human-readable. To improve readability, the client parses the response and presents it in a more understandable format.

For automation purposes, it can leave it as is or reformat into something else. WhoisXML API's lookup tool, for one, can present domain registration data in either JSON or XML format.

```
▼<WhoisRecord>
  <domainName>example.com</domainName>
  <parseCode>16392</parseCode>
  ▼<audit>
    <createdDate>2025-02-22 09:17:09 UTC</createdDate>
    <updatedDate>2025-02-22 09:17:09 UTC</updatedDate>
  </audit>
  <registrarName>RESERVED-Internet Assigned Numbers Authority</registrarName>
  <registrarIANAID>376</registrarIANAID>
  <dataError>RESERVED_DOMAIN_NAME</dataError>
  ▼<registryData>
    <createdDate>1995-08-14T04:00:00Z</createdDate>
    <updatedDate>2025-02-22T09:16:56Z</updatedDate>
    <expiresDate>2025-08-13T04:00:00Z</expiresDate>
    <domainName>example.com</domainName>
    ▼<nameServers>
      <rawText>A.IANA-SERVERS.NET B.IANA-SERVERS.NET </rawText>
      ▼<hostNames>
        <Address>A.IANA-SERVERS.NET</Address>
        <Address>B.IANA-SERVERS.NET</Address>
      </hostNames>
      <ips/>
    </nameServers>
    <status>clientDelete Prohibited clientTransfer Prohibited clientUpdate Prohibited</status>
```

RDAP clients also cache the IANA bootstrap registries so they won't have to query them every single time they receive a look up request. This process significantly speeds up RDAP queries and takes the load off IANA's servers.

Why Bootstrapping Makes RDAP Robust

As a relatively new concept (at least in the context of domain registration data), bootstrapping offers several advantages.

- **Decentralization:** IANA maintains the bootstrap registries, but the actual registration data is distributed among many different RDAP servers. These servers are operated by various organizations. If one server goes down, it does not bring the whole system down.

- **Scalability and automation:** The bootstrap registry system can easily accommodate Internet growth. All it takes is adding new entries for new domain names, IP address blocks, and autonomous system numbers (ASNs). And because the bootstrap registries are machine-readable, the entire discovery process can be fully automated.
- **Security:** Using HTTPS to download the bootstrap registries helps prevent tampering and ensures that clients get accurate information.
- **Ease of maintenance:** If something changes about a certain RDAP server, these changes only need to be reflected in IANA's bootstrap file – and all clients will get the changes from it. This is much more manageable than having to manually reconfigure every client (as is the case with WHOIS).

WHOIS Is Presumably Dead, Long Live RDAP

The transition to RDAP has been happening for several years. ICANN has required gTLD registries and registrars to implement RDAP since 2019. But it wasn't until January 28, 2025 when the ICANN officially sunsetted the requirement for gTLD registries and registrars to maintain WHOIS services.

Even though ccTLD registries have more flexibility, as they follow regional policies that differ from the global gTLD requirements, the deprecation of WHOIS port 43 on the gTLD still means that RDAP has become the authoritative source for gTLD registration data. As a result, access to WHOIS via port 43 was also expected to be closed.

However, our internal testing shows that port 43 remains open for queries, even for gTLD data in many cases. We expect the universal shutdown of this port to be gradual.

What Are the Differences Between RDAP and WHOIS?

At first glance, RDAP might seem like just another way to look up domain registration data, much like the familiar WHOIS protocol. While this is true to some extent, RDAP introduces significant architectural and data handling improvements beyond simply swapping one protocol for another.

So, how is RDAP different from WHOIS? Here's a quick rundown.

	WHOIS	RDAP
Protocol	Text-based	REST-based
Response format	Text	JSON
Standardized queries and responses	No	Yes
User authentication	No	Yes
International support	No	Yes
Built-in security	No	Yes (forced HTTPS)

Let's discuss each difference in some more detail.

Protocol

WHOIS uses a proprietary text-based protocol traditionally accessed over TCP on port 43. It requires specialized client software and is less integrated with standard web technologies.

On the other hand, RDAP relies on the RESTful approach, which, among other things, means that it uses HTTP/HTTPS, the same protocol web browsers use. The use of the RESTful approach makes it easier for developers to work with RDAP, as they can use simple GET requests to retrieve data instead of opening a connection on port 43 and manually handling text-based responses.

Response Format

WHOIS returns data as plain text, while RDAP uses JSON, a standardized and machine-readable format.

We also discovered some differences in the query results during our internal testing. Here's a screenshot of the registration details of swantonartscouncil[.]com retrieved through RDAP.

```
▼<WhoisRecord>
  <createdDate>1997-09-15T07:00:00+0000</createdDate>
  <updatedAt>2024-08-02T02:17:33+0000</updatedAt>
  <expiresDate>2028-09-13T07:00:00+0000</expiresDate>
  ▼<registrant>
    <organization>Google LLC</organization>
    <state>CA</state>
    <country>UNITED STATES</country>
    <countryCode>US</countryCode>
    <rawText>Registrant Organization: Google LLC Registrant State/Province: CA Registrant Country: US Registrant
    Email: Select Request Email Form at https://domains.markmonitor.com/whois/google.com</rawText>
  </registrant>
  ▼<administrativeContact>
    <organization>Google LLC</organization>
    <state>CA</state>
    <country>UNITED STATES</country>
    <countryCode>US</countryCode>
    <rawText>Admin Organization: Google LLC Admin State/Province: CA Admin Country: US Admin Email: Select Request
    Email Form at https://domains.markmonitor.com/whois/google.com</rawText>
  </administrativeContact>
  ▼<technicalContact>
    <organization>Google LLC</organization>
    <state>CA</state>
    <country>UNITED STATES</country>
    <countryCode>US</countryCode>
    <rawText>Tech Organization: Google LLC Tech State/Province: CA Tech Country: US Tech Email: Select Request
    Email Form at https://domains.markmonitor.com/whois/google.com</rawText>
  </technicalContact>
  <domainName>google.com</domainName>
  ▼<nameServers>
    <rawText>ns1.google.com ns2.google.com ns3.google.com ns4.google.com </rawText>
    ▼<hostNames>
      <Address>ns1.google.com</Address>
      <Address>ns2.google.com</Address>
      <Address>ns3.google.com</Address>
      <Address>ns4.google.com</Address>
    </hostNames>
    <ips/>
  </nameServers>
  <status>clientUpdateProhibited clientTransferProhibited clientDeleteProhibited serverUpdateProhibited
  serverTransferProhibited serverDeleteProhibited</status>
```

Data for a domain received over RDAP

This differs from the same domain's registration data obtained through WHOIS. Take a look at the registrant name and email address fields (details blurred for privacy).



```
▼ <WhoisRecord>
  <createdDate>2015-02-27T01:09:41Z</createdDate>
  <updatedDate>2025-02-11T08:06:36Z</updatedDate>
  <expiresDate>2026-02-27T01:09:41Z</expiresDate>
  ▼ <registrant>
    <name>██████ Ethan</name>
    <street1>██████ RD</street1>
    <city>SWANTON</city>
    <state>VT</state>
    <postalCode>05488-8048</postalCode>
    <country>UNITED STATES</country>
    <countryCode>US</countryCode>
    <email>██████@GMAIL.COM</email>
    <telephone>18027527533</telephone>
    <rawText>Registrant Name: ██████ Ethan Registrant Street: ██████ RD Registrant City: SWANTON Registrant
    State/Province: VT Registrant Postal Code: 05488-8048 Registrant Country: US Registrant Phone: ██████7533
    Registrant Email: ██████@GMAIL.COM</rawText>
  </registrant>
  ▼ <administrativeContact>
    <name>██████ RUPP</name>
    <organization>██████ RUPP</organization>
    <street1>██████ RD</street1>
    <city>SWANTON</city>
    <state>VT</state>
    <postalCode>05488-8048</postalCode>
    <country>UNITED STATES</country>
    <countryCode>US</countryCode>
    <email>██████@gmail.com</email>
    <telephone>18027527533</telephone>
    <rawText>Admin Name: ██████ RUPP Admin Organization: ETHAN RUPP Admin Street: ██████ RD Admin City: SWANTON
    Admin State/Province: VT Admin Postal Code: 05488-8048 Admin Country: US Admin Phone: ██████7533 Admin
    Email: ██████@gmail.com</rawText>
  </administrativeContact>
  ▼ <technicalContact>
    <name>INC, BLUEHOST</name>
    <organization>BLUEHOST.COM</organization>
    <street1>5335 GATE PKWY.</street1>
    <city>JACKSONVILLE</city>
    <state>FL</state>
    <postalCode>32256</postalCode>
    <country>UNITED STATES</country>
    <countryCode>US</countryCode>
    <email>whois@bluehost.com</email>
    <telephone>17136595940</telephone>
    <fax>18017651992</fax>
```

Data for a domain received over WHOIS

As you can see, the WHOIS response contains some registrant details that were not included in the RDAP response.

Standardized Queries and Responses

Due to the plain text format and the lack of strict standardization, the structure and content of WHOIS output could vary significantly between providers. With RDAP, clients send a uniform

query structure and registrars and registries are required to respond using the standardized JSON format with standardized fields. This makes it much easier to process the responses, as their structure is always the same when using RDAP.

User Authentication

RDAP allows for the possibility of tiered access control, although it's not a mandatory part of the protocol. Individual RDAP server operators can implement their own access control mechanisms to restrict which data is returned to different clients. WHOIS does not offer this feature—all relevant domain data is displayed for everyone to see.

International support

WHOIS has limitations in handling different languages and character sets as it was primarily designed for ASCII characters. This meant that domain names or registration information containing characters from other languages (like Cyrillic, Arabic, or Chinese) often couldn't be accurately represented or displayed.

RDAP, on the other hand, was designed to fully support Unicode—an international standard for encoding characters from virtually all writing systems. The protocol makes it easier for users around the world to access and manage domain name registration data in their native languages.

Built-In Security

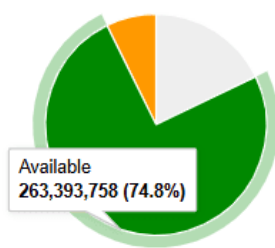
WHOIS lacks built-in encryption. And since data is transmitted in plain text, it is vulnerable to interception and modification.

RDAP connections, on the other hand, are forced over HTTPS. Communication between the client and RDAP server is encrypted, which prevents unauthorized parties from intercepting or modifying domain data.

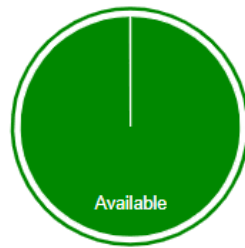
RDAP Rollout

While RDAP is undoubtedly the future of registration data access, the reality as of 20 February 2025 is a bit more nuanced. RDAP deployment is only at 74.8% across all TLDs.

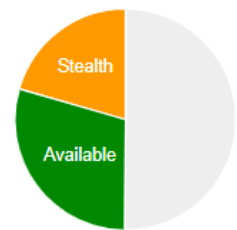
RDAP deployment by TLD type (based on approximate # domains)



All TLDs



Generic TLDs

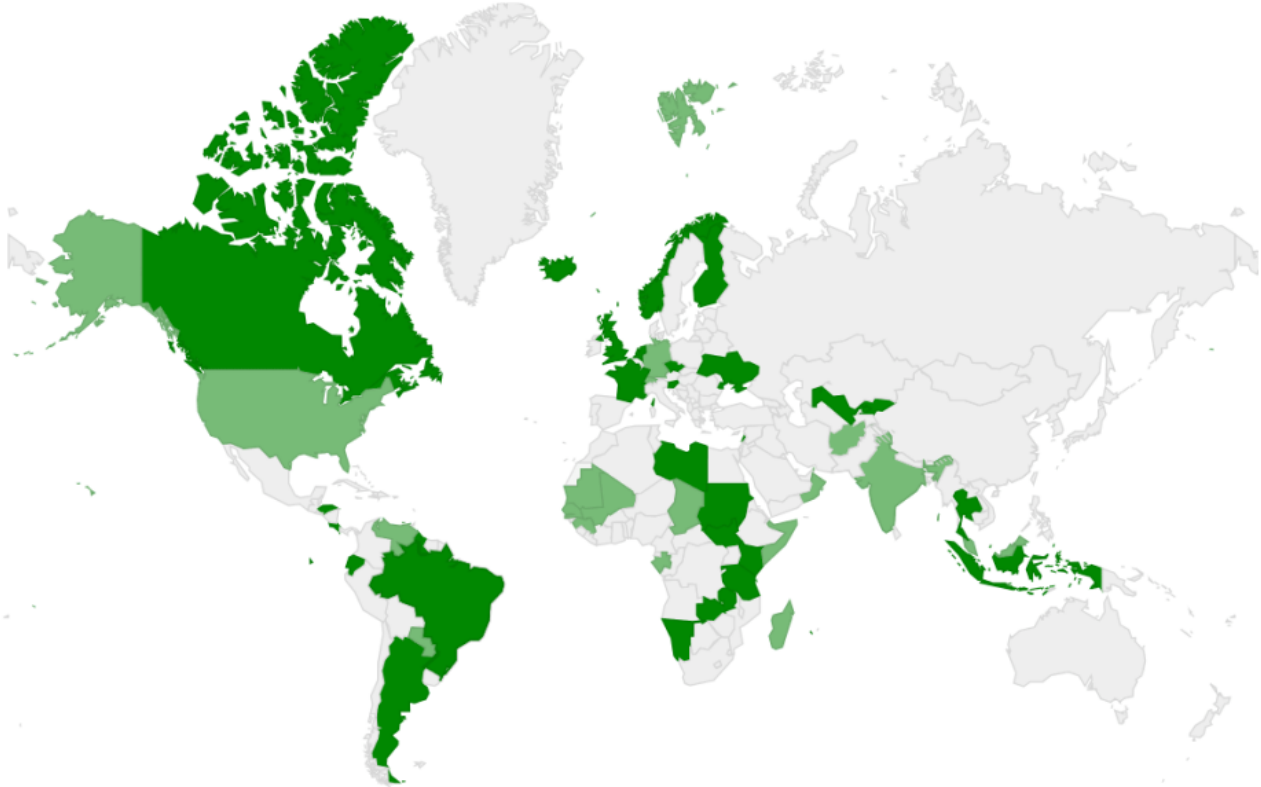


Country-code TLDs

Source: <https://deployment.rdap.org/>

Zooming in on ccTLDs, the figure is lower—only 29.4% of ccTLDs have deployed RDAP. This is related to the scope of ICANN's authority. ICANN's mandates, including the transition to RDAP, apply only to gTLDs, whereas ccTLDs operate under their own management.

World map showing ccTLD RDAP deployments



Source: <https://deployment.rdap.org/>

How to Use RDAP

RDAP is a RESTful protocol, so it's only natural that you could use it on API and lookups. In fact, our WHOIS API is RDAP-ready. It gives you the option to retrieve registration data through RDAP or WHOIS. We added a special "rdap" parameter that accepts boolean values in the following way:

- 1 - retrieve results in RDAP
- 0 - retrieve results in WHOIS

We also implemented an automatic fallback mechanism to RDAP in case WHOIS data cannot be retrieved. You can find all the details in the [WHOIS API documentation](#).

If you want to try it yourself, [register](#) to get free API credits and API key and try running the

following query in the terminal, replacing API_KEY with your actual API key:

```
curl --location 'https://www.whoisxmlapi.com/whoisserver/WhoisService' \  
  --header 'Content-Type: application/json' \  
  --data '{  
    "domainName": "example.com",  
    "apiKey": "API_KEY",  
    "outputFormat": "JSON",  
    "rdap": 1  
  }'
```

It's worth noting that, like WHOIS, RDAP only provides current registration data. Neither of them is designed to be historical archives, so if you're looking for historical data, you'll need a [WHOIS History API](#). As maintainers of a WHOIS/RDAP client, we record the past domain information for our users. Rest assured that we will continue updating our historical WHOIS repository with data received over RDAP.

What Does Transition to RDAP Mean for You

We've detailed how RDAP works and how it differs from WHOIS. But the underlying question is: How does this transition actually affect organizations?

Domain registrars and registries felt the initial impact of the transition the most. It was them who had to invest in new systems and processes to implement RDAP services. For the average domain registrant, however, the day-to-day experience likely won't change dramatically. You'll still register domains through your chosen registrar, and you might not even notice the underlying protocol shift.

However, the transition will have some noticeable impact on cybersecurity professionals and other domain data users. While domain ownership data will become more reliable and standardized, users may have reduced access to certain information due to the enhanced privacy that RDAP's tiered access could potentially introduce. The actual level of privacy depends on how registrars implement controls and what local regulations mandate.

Access to non-public domain registration data may need to be routed through the [Registration Data Request Service \(RDRS\)](#), where requesters are required to provide [detailed justification](#) for their need to access the data.