

REFINITIV

DATASCOPE SELECT & TICK HISTORY

BEST PRACTICES & FAIR USAGE POLICY

DOCUMENT VERSION 2.2

Date of Issue: April 2021

© [Refinitiv](#) 2021. All Rights Reserved. Reproduction or redistribution of Refinitiv content, including by framing or similar means, is prohibited without the prior written consent of Refinitiv. Refinitiv and the Refinitiv logo are trademarks of Refinitiv and its affiliated companies.

Contents

About This Document	5
Best Practices.....	5
Fair Usage	5
Prevention Rather than Cure	5
1 Best Practices & Fair Usage Policies Common to All Interfaces .	6
Specify Only the Dates, Fields, and Instruments You Need	6
Querying Multiple Instruments Together Is Faster Than One at a Time	7
Use a URL Based on Domain Name, Not on IP Address	7
Retrieve Reports When Completed, Not in the Sequence Submitted	7
Get Tick History Files Faster by Downloading Them Directly From Amazon7	
Download Reports Before They Become Unavailable	8
Use Extraction Notes to Troubleshoot Reports	8
Understand Extraction File Management.....	8
Understand User Account Administration	8
Avoid Open-Ended Searches	8
Java Support	9
2 REST API Best Practices & Fair Usage Policies	10
Reuse Authentication Tokens for Faster Authentication	10
Set a Realistic Polling Interval	10
Do Not Resubmit After a Timeout.....	10
Retrying REST API Error Responses.....	11
Check Content-Encoding Header.....	11
Avoid On-demand Extraction Requests for Repeated Tasks.....	11
Make Non-Time-Critical Requests Outside of Market Close Times	11
Avoid Repeated Price Requests Outside of Market Close Times.....	12
Disperse Requests Instead of Submitting Large Request Sets at Once	12
Perform Asynchronous Requests to Avoid 408 Timeout Errors	12
Do Not Use the SDK to Host Applications in Azure.....	12

3	Limits	13
	Limits Common to All Interfaces (GUI, REST API, SOAP API, FTP)	13
	Limits Specific to the REST API	16
4	Structuring Extraction Request – Quick Reference	18
5	Getting Help	20
	Refinitiv Developer Community	20
	Service and Support.....	20
	Feedback	20
	Your Personal Information	20

About This Document

This guide assists users and developers in understanding the best practices and fair usage policies for Refinitiv DataScope Select and Tick History.

Best Practices

This guide describes the recognized techniques for using DataScope Select and Tick History more quickly and effectively while reducing opportunities for error. It also describes the limits you will want to keep in mind when issuing report requests. Failure to adhere to the limits and tolerances described in this guide will result in placing an unnecessary burden on the DataScope Select API and may result in the user account being disabled.

Fair Usage

Refinitiv is committed to ensuring that all of our DataScope and Tick History services are reliable and of value to all of our customers. As data becomes more and more of a valuable resource, the company is finding that many of the traditional transaction processing use cases that were envisaged for the DataScope solutions are changing rapidly to support data discovery and quantitative analysis.

These latter use cases often make huge demands on the system resources and a very small number can disproportionately affect a large number of clients. In order to ensure that all clients can continue to rely on our services, we need to deem certain processes as excessive which may then lead to specific action in order to return the service to the benefit of all.

Prevention Rather than Cure

Refinitiv is passionate about its clients maximising the benefit they receive both from Refinitiv data and Refinitiv solutions.

Refinitiv will work with clients on respective use cases to help determine which product is best suited to fulfil the client's requirement and how best to access that solution.

Refinitiv offers its clients many ways to license data and solutions ranging from individual licenses, enterprise agreements and most recently the Refinitiv Access programme.

One of the benefits of the Refinitiv Access programme is that it allows our clients to investigate where else Refinitiv data may apply under a single license agreement without the usual constraints. The lack of constraints however can also inadvertently lead to the wrong solution being used in the wrong way so Refinitiv would ask that any new use cases that are envisaged are worked through collaboratively with the Refinitiv Team to ensure the optimum outcome is reached both for the individual client concerned and the broader user community.

All new use cases should be discussed with the client Account Manager and an appropriate Solutions Consultant. If new use cases are not discussed with the Refinitiv team and inappropriate behaviour in data processing over the platform is observed, it may result in the client account being disabled.

1 Best Practices & Fair Usage Policies Common to All Interfaces

This chapter describes Refinitiv DataScope Select and Tick History best practices and fair usage policies that apply to all interfaces (GUI, API, and FTP):

- [Specify Only the Dates, Fields, and Instruments You Need](#)
- [Querying Multiple Instruments Together Is Faster Than One at a Time](#)
- [Use a URL Based on Domain Name, Not on IP Address](#)
- [Retrieve Reports When Completed, Not in the Sequence Submitted](#)
- [Get Tick History Files Faster by Downloading Them Directly From Amazon](#)
- [Download Reports Before They Become Unavailable](#)
- [Use Extraction Notes to Troubleshoot Reports](#)
- [Understand Extraction File Management](#)
- [Understand User Account Administration](#)
- [Avoid Open-ended Searches](#)
- [Java Support](#)

Specify Only the Dates, Fields, and Instruments You Need

Report performance can be very sensitive to the length of the date range on which you are reporting, to the number of fields you are reporting on (even if some are empty), and to the number of instruments. Specify only the date range, fields, and instruments that you need.

Fields With Static Values

Avoid requesting fields with static values (such as Asset Type and Instrument ID Type) in requests that return many rows of data, because those static fields will be unnecessarily retrieved each time. Instead, request reference and other static data in a separate extraction request. Even empty fields require extraction time.

RIC Chains

Only request a RIC chain if you need most or all of the RICs in the chain. DataScope Select and Tick History fully expand a chain to resolve it to all of its constituent RICs, so if you need only a few RICs, you can improve performance by requesting them individually.

Secondary Identifiers

DataScope Select

Best practice is to build your instrument list using RICs. If RICs cannot be used, use another identifier and a specified Price Source in import or chosen from search results. If only a secondary identifier can be provided, ensure the import preferences are set properly for desired outcomes. If all RICs for a secondary identifier are desired, import using "*" as the Price Source or choose RIC as the preferred identifier in Search, and then select all identifiers returned in your search results to add to your instrument list.

Tick History

In many cases, one CUSIP, ISIN, or SEDOL maps to multiple RICs. The Time and Sales, Market Depth, and Intraday Summaries report types return data for all of a non-RIC identifier's corresponding RICs. If you do not want data for all of the RICs, map the non-RIC identifier to the set of RICs that you need, and report on only that set.

Note that by default, a request for Elektron Timeseries, Terms and Conditions, or Corporate Actions delivers data for the RIC corresponding to the specified CUSIP, ISIN, or SEDOL. If you also specify an exchange, it delivers data for the RIC corresponding to the specified CUSIP, ISIN, or SEDOL on that exchange. If you want results for several exchanges, you need one entry per exchange.

Querying Multiple Instruments Together Is Faster Than One at a Time

Submitting a single job that reports on multiple instruments is faster than submitting several jobs that each report on one instrument.

Frequent, repeated tasks monopolize system resources and impede performance in both extraction processing and in accessing the GUI platform. Instead, submit fewer requests with multiple instruments, up to the maximum number of instruments allowed for the report type. Please see [Structuring Extraction Request – Quick Reference](#) for instrument limits. Combine requests of the same report template type that are to be made within five minutes of each other into a single request.

Should Refinitiv identify a user that persistently submits extraction requests for a single instrument where a batch request is more appropriate, the user will be contacted by the Refinitiv team to help the user design the correct extraction request.

Use a URL Based on Domain Name, Not on IP Address

When accessing DataScope Select, use a domain name URL (for example, <https://hosted.datascope.reuters.com/DataScope/Home>), not an IP address (which comprises only digits and periods). Using a URL based on domain name insulates you from changes to hosting infrastructure, and ensures that in the event of a failover to another hosting site you do not need to change scripts or browser bookmarks.

Retrieve Reports When Completed, Not in the Sequence Submitted

Reports of the same type (for example, End Of Day Pricing reports) run sequentially, and complete in the order in which you submitted them. Reports of different types run in parallel (independently of each other) and complete when they are done.

Via the REST API, you can poll report jobs to determine when each one is done, which enables you to retrieve each one as soon as possible. Via the GUI, you can check the **Completed Extractions** tab to determine when each report is available.

Get Tick History Files Faster by Downloading Them Directly From Amazon

- You can download files faster by retrieving them directly from the Amazon Web Services (AWS) cloud in which they are hosted.
To do this via the GUI, select the user preference **Enable Direct Download From S3** in the **Tick History File Delivery** section of the **Preferences** screen.
- For information about doing this via the REST API, see the [Tick History REST API User Guide](#).

Download Reports Before They Become Unavailable

Extractions older than 45 calendar days are automatically deleted and cannot be recovered. Please download your extraction files upon receipt. **DataScope Select should not be used as an archiving repository**. Any file manipulation, such as changes to the file names and formats, should be performed post-download, on your own computer.

On-demand extractions expire after 7 days.

Use Extraction Notes to Troubleshoot Reports

All extraction requests return notes, such as information about quotas, embargoes, and permissions. These notes can be helpful to you when you troubleshoot problems. It's a good idea to examine these notes on a regular basis and learn what kinds of information are available. Always provide the corresponding notes when raising an extraction query.

Understand Extraction File Management

You can keep the files associated with your most recent 1,000 extractions.

If there are files from more than 1,000 extractions, the files from the oldest extractions that are more than 24 hours old are deleted, beginning with the oldest. The files from the most recent 1,000 extractions are always retained, and the files that are less than a day old are always retained.

The deletion process is initiated daily at UTC+00:00. This process applies to extraction-related files. Files that do not appear to belong to an extraction—that is, files that do not have a corresponding notes file—are deleted once they are more than 24 hours old.

Understand User Account Administration

User accounts that have been inactive for 9 months or longer that are not under contract will be automatically deleted, as well as any associated files tied to the account. Users under contract can re-activate their accounts by logging in to DataScope Select.

Deactivation of accounts will also result in a reset of preference settings and data files. No extraction files or preference settings will exist for re-activated users.

Avoid Open-Ended Searches

DataScope Select and Tick History support simple, ad-hoc searches and are not intended to be used as a discovery tool. Prefixed wildcards used in search queries will be ignored, as this practice is not an efficient use of the tool.

If users continue to submit open-ended searches, they will be contacted by their Account Manager, a Solution Consultant or a Customer Success Manager to see how best this can be resolved. Once a suitable alternative solution has been determined, Refinitiv would expect the user to reflect the appropriate changes.

If the user fails to administer the agreed upon changes, this may result in the user account being disabled.

Java Support

The DataScope Select user interface (GUI) and REST API support the same releases of Java Platform Standard Edition (usually referred to as Java SE) that are supported by Oracle's Premier Support program.

It is important that you use only supported versions of Java with DataScope Select. DataScope Select and Tick History are tested and validated only with supported Java versions, and you may get non-standard behavior with other Java versions.

2 REST API Best Practices & Fair Usage Policies

This chapter describes Refinitiv DataScope Select's best practices and fair usage policies for the REST API interface. The DataScope Select REST API supports both Tick History use as well as Pricing and Reference Data usage. Refinitiv constantly monitors and reviews the API usage by its customers to ensure optimum system performance. In exceptional circumstances, Refinitiv may have to adjust these limits and tolerances should it deem such action necessary to preserve system stability and performance. The Practices described below cover both use cases:

- [Reuse Authentication Tokens for Faster Authentication](#)
- [Set a Realistic Polling Interval](#)
- [Do Not Resubmit After a Timeout](#)
- [Retrying REST API Error Responses](#)
- [Check Content-Encoding Header](#)
- [Avoid On-demand Extraction Requests for Repeated Tasks](#)
- [Make Non-Time-Critical Requests Outside of Market Close Times](#)
- [Avoid Repeated Price Requests Outside of Market Close Times](#)
- [Disperse Requests Instead of Submitting Large Request Sets at Once](#)
- [Perform Asynchronous Requests to Avoid 408 Timeout Errors](#)
- [Do Not Use the SDK to Host Applications in Azure](#)

Reuse Authentication Tokens for Faster Authentication

Creating and reusing an authentication token is the fastest and most efficient way of handling authentication. It is superior to providing a user ID and password each time you submit a request, and to creating a new authentication token each time you submit a request.

Set a Realistic Polling Interval

A large report that will take a long time to execute does not need a short polling interval. The more frequently you poll, the more system resources are consumed, so set the interval to an appropriate period relative to the report. The DataScope Select API currently has a limit of 200 polls in any 60 second period but a recommended poll rate of 30 polls in any 60 second period. Refinitiv would ask its users to poll as close to the recommended interval as possible.

Should Refinitiv identify a user that persistently polls in excess of 60 times in any 60 second period, the user will be contacted by the Refinitiv team to reduce the number of polls to be more in line with the recommended level. Should a user persist in breaching the 60 polls, their account may be disabled.

Do Not Resubmit After a Timeout

If your procedure that polls a report job times out on your system, do not resubmit the report job. It is only your polling procedure that has timed out: the original report job is still queued to execute, or is still executing, on the DataScope Select platform. Resubmitting the report job will not get you the report faster. (In fact, resubmitting it might have the opposite effect, because you will now have an additional instance of the job running.) If your polling procedure frequently times out, consider increasing your timeout period.

Retrying REST API Error Responses

When making a request to the REST API, do not retry when the response has a status code of 501 or less. These errors are not recoverable.

For errors with a status code of 502 or greater, the retry logic should not retry more than three times per minute and no more than 10 times. Less is preferable so a downed server is not flooded with unserviceable requests.

Note that the C# Toolkit will automatically retry 502 or greater responses three times. It retries after one second. If still failing, the C# Toolkit will try again after five seconds. If still failing, the C# Toolkit will try one last time after 30 seconds. If this last call fails, an exception will be raised in the code.

Check Content-Encoding Header

File compression requests may be intermittently denied for various reasons with increased likelihood during periods of heavy use.

Use of the DataScope Select API SDK tool kit provides protection against compression request denials by system services. We recommend that clients not using the SDK tool kit check for the Content-Encoding parameter in the response header and do not attempt to unzip the results unless it is present.

Alternately, clients can use an unzip product that recognizes the compression status and only attempts to unzip if compressed. Always protect against compression request denials by one of these methods as a best practice, as compression requests cannot be 100% guaranteed.

Avoid On-demand Extraction Requests for Repeated Tasks

Scheduling determines when a report request will be processed (data extracted from the data sources). On-demand reports are acted upon immediately, while scheduled reports run by calendar day, week or day of the month at an hour of a day in a reoccurring manner.

Frequently repeated tasks, such as requesting the same data, at the same time, every day, should be performed using a stored and scheduled report and not an on-demand report. Using an on-demand report for this type of workflow may result in possible delays.

Should Refinitiv identify a user that persistently issues an on-demand report where a scheduled report is more appropriate, the user will be contacted by the Refinitiv team to help the user design the correct scheduling.

Make Non-Time-Critical Requests Outside of Market Close Times

Non-time-critical requests, including searches and extractions of terms and conditions and reference data, as well as historical pricing, should be done outside of regional market close times.

Should Refinitiv identify that a user is persistently searching for or extracting terms and conditions or reference data inside market open hours, the Refinitiv team will contact the user to best determine how to schedule such a query outside of market hours if that is possible.

If, after consultation with the Refinitiv team on how to rectify the situation and where the circumstances are such that the query running outside of market hours does not adversely affect the client operations, the client ignores the advice and continues to search or extract such data within market open hours, they run the risk of the account being disabled.

Avoid Repeated Price Requests Outside of Market Close Times

Once a market has closed, users should avoid repetitively searching for or extracting the same data such as End of Day prices. Once the market has closed, these prices will no longer change so clients should avoid these repetitive redundant queries. Refinitiv will look to contact any user that repeatedly makes requests for redundant pricing data after the market has closed and advise the user of such activity and request that such requests be terminated.

Should a user subsequently not terminate such redundant requests, they run the risk that the account may be disabled.

Disperse Requests Instead of Submitting Large Request Sets at Once

Disperse your requests instead of submitting a large set of requests at once. Submitting large sets of requests simultaneously can impede system performance. Please disperse your requests within the defined limits and only submit the requests that you need.

Perform Asynchronous Requests to Avoid 408 Timeout Errors

A 408 Request Timeout error occurs when a synchronous request to the server takes longer to perform than the server is configured to wait. To avoid this error, perform asynchronous requests, particularly for long-running requests for historical data in Price History, Historical Reference Data and Elektron Timeseries reports.

Requests are synchronous by default, with a 30 second timeout. A request can be set to asynchronous by including the following header: *Prefer:respond-async*, with a default wait time of 30 seconds. For instructions, please see the [Async](#) topic in the DataScope Select REST API Help.

If a synchronous call times out, the request will fail completely; no data will be returned. Asynchronous calls do not time out. If the request cannot be completed inside the wait time, then a 202 Accepted message is returned with a location URL that allows later retrieval of the requested data. You can keep retrying this URL until the request completes as indicated with a 200 Success message.

Do Not Use the SDK to Host Applications in Azure

We do not recommend using the DataScope Select SDK for applications hosted in Azure. Azure is not tested with the DataScope Select SDK and, therefore, not directly supported. The SDK uses the regular.NET Framework class HttpClient with keep-alive enabled. The Azure Load Balancer may reuse existing connection after some period of inactivity (240 seconds by default) and ignores the keep-alive settings of an open connection. As a result, SDK clients who host their applications in Azure may experience intermittent disconnects.

3 Limits

Some Refinitiv DataScope Select and Tick History limits apply universally to all interfaces (GUI, REST API, SOAP API, and FTP), and some [apply only to the REST API](#).

Limits Common to All Interfaces (GUI, REST API, SOAP API, FTP)

This section describes limits that do not distinguish between requests submitted via the GUI, the REST API, the SOAP API, and FTP. When DataScope Select or Tick History evaluates one of these limits, it sums the requests submitted from all of these interfaces.

- [Prioritizing Extractions to Ensure Resource Availability for All Users](#)
- [Extraction Limits: Concurrent Requests per Report Template Type](#)
- [Extraction Limits: RICs per Report Template Type](#)
- [Extraction Limits: RIC-days in Concurrent Tick History Requests](#)

Prioritizing Extractions to Ensure Resource Availability for All Users

DataScope Select and Tick History ensure that one user who submits a large number of requests all at once against a given report template type cannot monopolize that template's resources at the expense of all other users. If other users submit requests against that template shortly after the first user, a queuing fairness algorithm ensures that the template executes requests for a mix of users, but still weights the mix of jobs to reflect the earlier submission time of the first user's requests.

Extraction Limits: Concurrent Requests per Report Template Type

Limits are enforced on the number of concurrent extraction requests a user can make and on the number of extraction requests for that user that can be processed concurrently at any one time.

In order to support consistent performance and optimize response times for the most users, DataScope Select and Tick History apply execution limits and queuing on a per-report template basis.

You can submit a maximum of 50 concurrent requests per report template for all report types except Corporate Actions and Terms and Conditions (Maximum Concurrent Requests per Report Template = 500) and Intraday Pricing (Maximum Concurrent Requests per Report Template = 1500).

When you reach your limit for the number of extraction requests for a report template type, any additional extraction request that you submit against that template type will fail, with the extraction notes explaining why it failed. If you submitted the request via the REST or SOAP API, the over-the-limit request will return HTTP status 429 with a description of the problem. Once any request ends, the tally of submitted requests is reduced by one, enabling you to successfully submit an additional request.

The number of concurrent extractions that can be processed on a per-report and per-user basis is two for all report templates except Intraday Pricing, Historical Reference, Elektron Timeseries report templates. Intraday Pricing supports concurrent extraction processing of up to 50 reports. Concurrent extraction processing is not supported for Historical Reference and Elektron Timeseries report templates. Any remaining user extraction requests remain in a **Queued** status. Once the maximum concurrent extractions complete, the next set of requests will be processed, and so on, until all requests are completed.

When does a request begin counting against my limit?

The precise time may vary by a minute or two, but for practical purposes:

- Assume that a request that you submit on-demand or that is scheduled to run immediately will count against your limit as soon as it is submitted.
- Assume that a request that is scheduled to run at a specified time will count against your limit at that time.
- Assume that a request that is triggered to run by a triggering condition will count against your limit at the time it is triggered.

How can I track how many requests I have currently running?

- The precise time that a new request is evaluated against the limit for that report template type may be affected by internal system latencies, and those latencies may cause a brief discrepancy between a request's reported status and its actual status. This discrepancy can result in the DataScope Select platform reporting a newly submitted request as queued and then as executing, and then a few moments later as having been rejected before executing.
- To keep a consistent tally of how many requests you have executing, we recommend that you consider a request to count against your tally as soon as you submit it, and that you do not consider it to be removed from that tally until you receive a status that the request has completed, failed, been rejected, or has been canceled.
- In this way you can keep your own tally of how many requests you have running, and your tally will never exhibit a brief discrepancy that leads you to mistakenly submit more requests than you should.

What happens if I have more extraction requests than the concurrent request limit?

- Do not send more requests than are allowed for the specified report type. If you send more requests than are allowed, those requests will fail. To avoid this, combine the requests into a single request that adheres to the maximum number of instruments allowed for that report template type. Batching requests into a single request is a more efficient use of system resources.

Extraction Limits: RICs per Report Template Type

The maximum number of instruments that you can extract in a single request is determined by the report template type that you are using for the extraction. This limit is the same regardless of whether you are extracting the instruments using the REST API or the GUI.

Note that your input list of instruments can include different kinds of instrument identifiers (such as simple RICs, RIC chains, CUSIPs, ISINs, and SEDOLs). DataScope Select then expands and resolves this input list into a list of simple RICs:

- A RIC chain is expanded into its constituent simple RICs.
- Non-RIC instrument identifiers (such as CUSIPs, ISINs, or SEDOLs) are resolved to simple RICs. In some cases, the mapping between other instrument codes and RICs is not 1:1.

The fully expanded and resolved list of RICs may contain more instrument identifiers than the original input list from which it was derived. For some report types, you are allowed more instrument identifiers in the fully resolved list than in the input list. For other report types, the limits on the fully resolved list and on the input list are the same.

For more information about the limits for the input list and for the fully resolved list, for each report type, please see [Structuring Extraction Request – Quick Reference](#) for instrument limits.

Extraction Limits: RIC-days in Concurrent Tick History Requests

Each user can extract up to 5,000,000 RIC-days in requests that are running at any one time. For each user, Tick History maintains a tally of the total RIC-days being extracted by the user's currently processing requests.

- If a user submits a request that increases his or her tally beyond that limit, Tick History rejects that new request.
- When one of the user's currently running requests ends (because it completes, fails, or is canceled), the request's RIC-days are subtracted from the user's "RIC-days" tally. Those reductions free up resources for the user to submit additional requests.
- The limit is evaluated separately for each user, that is, for each Tick History login account. The number of requests that one user is running does not affect the tallies of other users.

What is a RIC-day?

A "RIC-day" is a unit of measure in which data is extracted for one day for one RIC.

- A request's total number of days is the number of days between the query start date and query end date, inclusive.
- A request's total number of RICs is the sum of all fully resolved RICs being extracted.
 - A RIC chain is counted as the number of RICs in the fully resolved chain.
 - If the request uses instrument types other than RIC (such as SEDOL or CUSIP), the instruments of those other types are resolved to RICs, and it is the RICs that are counted against the limit. (In some cases, the mapping between RICs and other instrument codes is not 1:1.)
- A request's total number of RIC-days is its total number of days multiplied by its total number of RICs.

For example, if a request reports on the period from July 8 to July 23, it covers 16 days. If it reports on three RICs, then $3 \text{ RICs} * 16 \text{ days} = 48 \text{ RIC-days}$.

To which report types does this limit apply?

This limit applies only to RIC-days in requests against these report types:

- Tick History Time and Sales
- Tick History Market Depth
- Tick History Intraday Summaries
- Tick History Raw

To which interfaces does this limit apply?

The limit does not distinguish between requests submitted via the GUI and via the API. The RIC-days in requests that are submitted via each interface are summed together.

When do a request's RIC-days begin counting against my limit?

The precise time may vary by a minute or two, but for practical purposes:

- Assume that when you submit a request on-demand or schedule it to run immediately, its RIC-days will count against your limit as soon as the request is submitted.
- Assume that when a request is scheduled to run at a specified time, its RIC-days will count against your limit at that time.

When are the number of RIC-days evaluated?

When you submit a request, Tick History evaluates how many RIC-days the request would extract if it ran, and decides whether the request would cause the user to exceed his or her RIC-day limit given his or her RIC-day tally at the time of submission. If Tick History adds the submitted request's RIC-days to the user's current RIC-day tally and finds that it would increase the tally beyond the limit of 5,000,000 RIC days, Tick History rejects the submitted request before it has a chance to run.

Limits Specific to the REST API

Authentication Token Limits

An authentication token remains valid for 24 hours. If a token has expired, you simply create a new one. You can reuse a token as often as you like within that 24-hour period. You can create as many tokens as you want, as often as you want.

Endpoint Frequency Limits

The REST API employs endpoint-based rate management to ensure consistent performance for all DataScope Select and Tick History users.

The frequency of each REST API user's requests is measured against each endpoint. If your requests against a particular endpoint over a defined period of time (usually several seconds or minutes) exceed the per-user limit for that endpoint, each of your requests that is over the limit within that period of time will return HTTP status code 429 (Too Many Requests) with a description of the problem, and fail. You should wait several seconds and then resubmit the request and reduce the frequency with which you submit those requests.

Most users will rarely (if ever) reach these limits because the limit for each endpoint represents a very heavy level of use.

Each frequency limit depends primarily on the number of requests that you submit against that endpoint within the defined period of time. But how that limit is applied will depend also on secondary request processing factors that may be specific to your situation. The limits shown below are approximate because they are influenced by secondary factors. They are an absolute upper limit, but may vary by a small amount; we provide them to you to give you a sense of the range within which you will reach the limit. For example, a limit of 1000 might indicate a range of approximately 850 to 1,000.

The table below shows frequency limits defined by category of endpoint with maximum request limits per time period. Both recommended and current maximum request limits are shown. (Recommended limits are provided for some endpoint categories; the others are forthcoming.) The Fair Usage Requests Level denotes the point at which Refinitiv regards Fair Usage has been surpassed and any persistent infringement beyond these levels may result in the account being disabled. The first recourse, however, will be for the Refinitiv team to contact the user that is processing beyond the Fair Usage Limits to work with them to lower these to nearer to the Recommended Levels.

You are encouraged to code to the current fair usage request limits. Doing so ensures optimal performance and reliability of the DataScope Select platform for all users.

Note that the limits below are current as of **12 February 2021**. They are likely to be reduced further.

Category	Recommended Maximum Requests	Current Fair Usage Requests	Current Maximum Requests	Time Period (seconds)	Endpoints
All Requests	500	1,750	2,000	60	All endpoints in the REST API.

Category	Recommended Maximum Requests	Current Fair Usage Requests	Current Maximum Requests	Time Period (seconds)	Endpoints
EvaluationConnect (formerly AskTRPS)		300	400	300	This comprises the endpoints in the EvaluationConnect folder in the REST API Reference tree .
Authentication		20	30	300	This comprises the endpoints in the Authentication folder in the REST API Reference tree .
Extractions, All	500	750	1,000	60	All extraction-related endpoints.
Extractions, On Demand	100	250	400	60	This comprises ExtractRaw, ExtractWithNotes, Extract, ExtractNow, and ExtractMultipleNow.
Extractions, Standard (includes VBD)		750	900	60	This comprises the endpoints in the StandardExtractions folder in the REST API Reference tree .
Miscellaneous (called Category 6 in 429 error messages)		150	250	60	This comprises ExtractedFiles and GetPackageDeliveriesByDateRange.
On Demand Result Polling	30	60	200	60	All on-demand request polling, including ExtractResult, ExtractWithNotesResult, ExtractRawResult, and ExtractNowResult.
Quota		200	400	300	This comprises the endpoints in the Quota folder in the REST API Reference tree .
Users		250	400	300	This comprises the endpoints in the Users folder in the REST API Reference tree .
Search	30	100	1,500	60	This comprises the endpoints in the Users folder in the REST API Reference tree .

4 Structuring Extraction Request – Quick Reference

This chapter summarizes the best practices for structuring an extraction request for quick reference. Limits are identified in the table below. Note that exceeding these limits will result in processing delays:

- Run fewer requests with more records.
- Stay within maximum limits for extraction submissions and extraction processing requests.
- Request non-time critical data (Terms and Conditions, Reference Data, Historical Data) outside of peak market hours.
- Run asynchronous requests to avoid time out errors for long-running requests.
- When running extractions of the same report type with maximum records, only run one at time. Wait for the extraction to complete before running the next one. An exception is for Intraday Pricing reports, which can run concurrently.

Report Template	BROWSER/REST API/FTP		
	Record Processing Limit Before/After File Code & Change Expansion	Concurrent Extraction Request Limit	Concurrent Extraction Processing Limit
Bond Schedules	150,00 Before & After Expansion	50	2
Commodities Corrections History	50,000 Before & After Expansion	50	2
Composite	60,000 Before Expansion; 150,000 After Expansion	50	2
Corporate Actions - Standard Events	50,000 Before Expansion; 150,000 After Expansion	500	2
Corporate Actions - IPO Events	50,000 Before Expansion; 150,000 After Expansion	500	2
Corporate Actions - ISO 15022 Events	50,000 Before Expansion; 150,000 After Expansion	500	2
Elektron Timeseries	30,000 Before & After Expansion	50	1
EOD Pricing	75,000 Before Expansion; 150,000 After Expansion	50	2
Estimates Actuals	3,000 Before & After Expansion	50	2
Estimates Company Footnotes	3,000 Before & After Expansion	50	2
Estimate Detail	3,000 Before & After Expansion	50	2
Estimate Footnotes	3,000 Before & After Expansion	50	2
Estimate Summary	3,000 Before & After Expansion	50	2
Fixed Income Analytics	50,000 Before & After Expansion	50	2
Fund Allocation	600 Before & After Expansion	50	2
Historical Reference	30,000 Before & After Expansion	50	1
Intraday Pricing	50,000 Before & After Expansion	1,500	50

Report Template	BROWSER/REST API/FTP		
	Record Processing Limit Before/After File Code & Change Expansion	Concurrent Extraction Request Limit	Concurrent Extraction Processing Limit
Legal Entity Audit	10000 Before & After Expansion	50	2
Legal Entity Detail	50,000 Before & After Expansion	50	2
Legal Entity Hierarchy	2000 Before & After Expansion	50	2
MBS Factor History	20000 Before & After Expansion	50	2
News Items	0-7 Days - 50,000 8-31 Days - 25,000 32-366 Days - 10,000 367+ Days - 5,000 RIC-less extractions - 400,000	50	2
Ownership	500 Before & After Expansion	50	2
Premium EOD Pricing	50,000 Before & After Expansion	50	2
Premium Pricing	50,000 Before & After Expansion	50	2
Price History	30,000 Before & After Expansion	50	2
Ratings	50,000 Instruments Before; 150,000 After Expansion 1,000 Entities Before & After Expansion	50	2
Single Price History	50,000 Before & After Expansion	50	2
StarMine	50,000 Before & After Expansion	50	2
Symbol Cross Reference	50,000 Before & 250,000 After Expansion	50	2
Technical Indicators	50,000 Before & After Expansion	50	2
Terms and Conditions	50,000 Instruments Before; 250,000 After Expansion 1,000 Entities Before & After Expansion	500	2
Tick History Intraday Summaries	30,000 Before & After Expansion	50	2
Tick History Market Depth	30,000 Before & After Expansion	50	2
Tick History Time and Sales	30,000 Before & After Expansion	50	2
Tranche Factor History	3,000 Before & After Expansion	50	2

5 Getting Help

Refinitiv Developer Community

The [Refinitiv Developer Community](#) provides API support for DataScope Select and Tick History, and offers self-service learning through documentation and tutorials for performing authentication and setting up a development environment using Python and Postman.

The portal provides tools, documentation, sample code, learning materials and community Q & A forums to help you work effectively to build new, extended or integrated products using our open and controlled APIs.

Service and Support

The Refinitiv Statement of Service is available on [MyRefinitiv](#). MyRefinitiv is the Refinitiv portal that provides a single access point for timesaving support services, along with billing, user management, and information. For support using DataScope Select, please raise a query by accessing [Help & Support](#) at [MyRefinitiv](#). You are encouraged to [subscribe](#) to the following support channels to keep informed of changes to products and data, and to be notified of any service issues or changes:

- **Change Notifications**

- **Product** change notifications detail new, enhanced, or changed functionality, which may require your action, in products that you use.
- **Content** change notifications alert you to upcoming changes to real-time and historical data across all asset classes that are relevant to you.
- **RIC** change notifications inform you of planned changes to Refinitiv Identification Codes (formerly Reuters Instrument Codes).

- **Service Alerts**

You can subscribe to alerts about planned maintenance and unplanned service issues affecting your products and services and be notified via SMS or email.

Feedback

We invite your comments, corrections, and suggestions about this document: access the [Feedback](#) option under [Help & Support](#) at [MyRefinitiv](#). Your feedback helps us continue to improve our user assistance.

Your Personal Information

Refinitiv is committed to the responsible handling and protection of personal information. We invite you to review our [Privacy Statement](#), which describes how we collect, use, disclose, transfer, and store personal information when needed to provide our services and for our operational and business purposes.

