Nexthink V6.28

Product Overview

Generated: 4/15/2021 7:47 pm

Table of Contents

Table of Contents

Nexthink End-User IT Analytics

Software components

Nexthink is the main innovator in the field of end user digital experience for security, ITSM, and workplace transformation. Nexthink maps all the IT services, how they are being consumed, and how the IT infrastructure is operating, from the perspective that matters most: the perspective of the end user (employee). Nexthink provides essential visibility and insight into IT operations and security for IT Governance.

Nexthink Architecture

The architecture of Nexthink has been designed to simplify operations, ensure scaling and allow a rapid deployment. The system is composed of six main software components:

- The Collector captures information from all end-user desktops and laptops.
- The Mobile Bridge captures mobile device information from Microsoft Exchange.
- The Engine aggregates Collector and Mobile Bridge information and provides real-time IT analytics.
- The Finder is the rich client application for searching and analyzing data on Engines.
- The Portal aggregates Engine information and provides dashboarding, reporting, and long-term trending analytics.
- The Library is a cloud knowledge database.

Modular product structure

Nexthink offers a modular product structure that can grow with your needs. The product is licensed with respect to the number of monitored physical or virtual devices and, optionally, server users. On top of the basic product (Nexthink Analytics), the following modules can be purchased:

- *Nexthink Act* offers you a way to remotely act on the devices of the end-users for automated or assisted servicing.
- *Nexthink Engage* gives you the means to reach out to the end-users, gather their feedback regarding IT or other subjects, and notify them of relevant issues.
- *Nexthink Enhance* provides additional classification and security-related information to Analytics, including binary threat level and category, as well as web domain reputation, category and hosting country.
- *Nexthink Web & Cloud* grants access to analytics related to intranet and extranet HTTP and HTTPs web requests (now included in the Nexthink Analytics offer for new contracts).
- *Nexthink Integrate* enables the product API and access to continuously improved integration samples, reports, etc.

Nexthink Analytics as well as the modules grant access to investigations, widgets, dashboards, categories, etc. directly from the Nexthink Library, our cloud repository of content.

Operational data sent to Nexthink

Nexthink gathers operational data from customers to offer them additional valuable services:

- *Support telemetry*, for an improved support service.
- *Cloud Intelligence*, for anonymized comparative analysis (including the Digital Experience Score benchmarking).
- *Enhance data*, for risk and compliance management of applications and web browsing. Applies to licensees of the Nexthink Enhance module.

Digital Experience Score

The *Digital Experience Score* measures both the ability to get things done and the satisfaction of your employees with the provided IT environment.

By automatically providing benchmark data to Nexthink, compare the Digital Experience Score of your company to the scores of other companies in the same or in different industries.

Collector

Introduction

The Collector is a light-weight agent based on patented technology. It captures and reports network connections, program executions, web requests, and many other activities and properties from the devices of the end-users on which it runs. It is implemented as a kernel driver and accompanying services, offering remote and automated silent installations with negligible impact on the performance of local desktops, while minimizing network traffic.

CPU usage	Typical memory usage	Network traffic
• Less than 0.015% (on average)	• Kernel 500 KB • User 30-40 MB Temporary memory spikes are possible during campaigns.	• **UDP** (Optional) 0.1 - 0.3 Kbps (on average) • **TCP**

		Depending on · Campaigns · Remote actions · Updates In TCP-only configs · Add documented UDP traffic

The following figure depicts the role of the Collector within the Nexthink solution.

Collector components

The capability of the Collector for gathering user activity data is shared by the kernel driver and the helper service (or daemon) components. By running close to the operating system, the kernel driver detects some kinds of user activities that are only visible at this level.

Click to see the detailed list of components of the Collector.

Features

Multi-Platform

The Collector is available for both Windows and macOS operating systems. The present documentation states the platforms to which each feature applies. Likewise, the data model details the individual pieces of information collected for each platform.

Applies to platforms:

CrashGuard

Since the Windows Collector driver is a kernel-mode component, any error in its internals or its interaction with a misbehaving third-party driver can lead to system instabilities. Even with Nexthink putting as much attention as possible towards delivering bug-free software, the principle of precaution holds. The CrashGuard feature detects every system crash and, by default, it disables the Collector driver itself if the system crashes more than three times in a row after installation.

Applies to platforms:

Kernel traffic interception

Some applications may send and receive data to and from the network using kernel-mode components, actually hiding their network traffic from user-space monitoring applications. Being a kernel driver itself, the Windows Collector is nevertheless able to detect and report such traffic.

Applies to platforms:

Paths aliasing

The Collector identifies commonly used paths (e.g. C:\WINDOWS\, C:\Program Files\) and other special mount locations (removable mount points, network drives) with paths aliases. For example, if the DVD-Rom drive is mounted under D:, the Collector reports an application **setup.exe** being launched from this media as **%RemovableDrive%\setup.exe**.

Reliable connectivity via TCP

When configured to send data through TCP, the Collector relies on the connection-oriented features of the protocol to ensure that the information reaches the Engine.

In addition, when the connection between the Collector and the Engine is lost or not established yet, the Collector is able to buffer up to 15 minutes of data (a maximum of 2500 packets not older than 15 minutes) to send to the Engine once the connection is (re-)established.

Network switching

A change of network interface is transparent to the Collector, except when it invalidates the DNS resolution of the Engine. In the latter case, the process of

adapting to a different network may take a few minutes and the Collector resends the whole context to the Engine.

Event logging

Connection events to the Nexthink Appliance and main errors are written to either the standard Windows event logs or the macOS system log.

On-the-fly configuration

The Collector driver parameters can be changed through the Collector Control Panel extension or the Collector Configuration tool. There is no need to restart the computer for the changes to become effective.

Code signed software

To be able to load and run on Windows devices, the kernel components of the Windows Collector are signed with an official Microsoft certificate. Likewise, user-space components are signed with a valid Nexthink certificate.

In turn, the Mac Collector is signed with the Developer ID certificate of Nexthink and follows Apple notarization process to ensure that it can be installed and run seamlessly on macOS devices.

Related tasks

- Installing the Collector on Windows
- Installing the Collector on macOS

Related references

- Collector configuration tool
- Components of the Collector
- Data model

Finder

Nexthink Finder, built upon powerful visualization techniques, is the search and user interface to render visibility into your IT infrastructure. Analyze IT services and query what you need within seconds. Expand or drill-down the results in a few clicks to reveal swiftly, across the entire network, how many versions of a

particular application are in use and on which workstations, the bandwidth consumed by the application, the servers and domains that the application accesses, the network response times, which users experienced issues, and much more.

Engine

Nexthink Engine is a high-performance analytics software capable of processing millions of endpoint activities in seconds. Events sent in real time by Collectors populate the Engine with activity data, furnishing a rich repository of historical and live IT infrastructure usage data from the end-user perspective. Engine leverages an in-memory database for rapid queries (via the Nexthink Finder) and flexible reporting (via the Nexthink Portal).

Related tasks

- Installing the Appliance

Portal

Nexthink Portal is the reporting tool, collaboration platform and centralized management platform of the Nexthink End-User IT analytics platform. A comprehensive set of dashboards are delivered out-of-the-box but it is possible in a matter of minutes to construct custom dashboards, valuable for anyone in the organization. Personalized metrics are simple to define as drag-and-drop widgets and can be quickly published and shared. Nexthink Portal front-end is a web application running inside a browser.

Related tasks

- Installing the Appliance

Nexthink Library

The Nexthink Library is an online knowledge database that gives you access to content packs with a large set of ready-to-use predefined investigations, templates, dashboards, remote actions, and campaigns that you can directly install and use on your Nexthink setup.

Digital Experience Score

The content of this document is for versions 2.0.0.0 or later of the DEX Score library pack.

Context

The digital workplace is undergoing rapid transformation, driven by industry trends like the anywhere/anytime work styles, the ongoing shift to cloud and the growing number of technologies, service providers, standards and frameworks that, on one side, enable companies to remain competitive and compliant, and on the other make their IT ecosystems more and more complex to manage and subject to changes, issues, and threats. The result of all this is a degraded experience and lost productivity for the employees. For IT departments, it means struggling to contain the costs, witnessing a rise in the expenditures to manage the day-to-day, leaving the smaller portion of the budget for innovation and transformation projects.

Additionally, there is a ?delivery gap?: IT teams assume they are giving their users what they want and need but, when employees are asked about their perception, you hear a very different story. IT departments struggle to truly understand the employees and to identify how to improve their digital experience, relying on SLAs focused on metrics such as service availability and target resolution times that convey very little in terms of how well the services actually perform for the employees. Attempts to complement this with surveys among the employees are often ineffective and with very low response rates.

DEX Score

IT leaders need a way to set up governance of the digital employee experience, allowing them to understand the maturity of their organization in that space, set the goals and drive improvements that would no longer be based on trial and error, but on clear evidence derived from organizational data. Such improvements correlate with increased productivity, reduced costs, improved agility, reduced attrition, and reduced time to market. The DEX score is a key indicator that enables them to:

- Understand where, what and how to improve based on factual non-arguable data
- Listen to the voice of employees (VoE) while monitoring hard metrics at the same time

- Benchmark with other organizations - Benchmark internally (e.g. one region against another)
- Set the basis for eXperience Level Agreements (XLAs)

Overview

The *Digital Employee Experience* can be defined as the ability of an employee to get things done by interacting with the IT environment in an enjoyable manner.

By combining user sentiment with hard data retrieved from the endpoints, the Digital Experience Score provides a simple yet actionable way to both monitor the real experience of the employees with their digital workplace and identify opportunities for improvements, which might lead to increased productivity, reduced costs, reduced attrition and reduced time to market. The Digital Experience Score assesses four areas impacting the digital employee experience:

- Device
- Business applications
- Productivity & collaboration tools
- Overall employee satisfaction

The Digital Experience Score implements each one of these areas as a subscore:

The Digital Experience Score is available in the Nexthink Library for both Windows and macOS in two versions: the full version with sentiment data and the version with hard metrics only.

Full version with sentiment data	For Windows and macOS	**Hard metrics only**	For Windows and macOS

More details about the DEX score

To know more about the DEX score, connect to Community to access additional content on this topic.

Related tasks

- Installation and configuration of the DEX score
- Structure of the DEX score
- Computation of the library pack
- How to use the DEX score
- Thresholds used for the hard metrics

Related references

- Digital Experience Score (hard metrics and sentiment data)
- Digital Experience Score (hard metrics only)
- NXQL Data Model

Licensing terms

License agreement

The links provided on this page hold the terms and conditions that govern the use of Nexthink software by customers who purchased a commercial license of Nexthink.

On-premises offering

Customers of Nexthink's on-premises offering are bound by either one the following terms:

- If you purchased your license directly from Nexthink:
  Nexthink On-premise License Agreement
- If you purchased your license from an authorized reseller (excluding Middle East, Turkey, Africa):
  Nexthink On-premise License Agreement (for indirect customers)
- If you purchased your license from an authorized reseller (within Middle East, Turkey, Africa):
  Nexthink On-premise License Agreement (for indirect customers in the META region)

Once the product is installed, find a copy of the licensing terms in any of the provisioned Nexthink Appliances under:

```
/var/nexthink/eula/license.txt
```

Cloud offering

Customers of Nexthink's cloud offering are bound by either one of the following terms:

- If you purchased your license directly from Nexthink:
  Cloud Master Services Agreement
- If you purchased your license from an authorized reseller (excluding Middle East, Turkey, Africa):
  Cloud Terms and Conditions (for indirect customers)
- If you purchased your license from an authorized reseller (within Middle East, Turkey, Africa):
  Cloud Terms and Conditions (for indirect customers in the META region)

Open source software licenses

Nexthink software components make use of third-party software libraries that follow an open source licensing model. These libraries are redistributed in binary form within selected Nexthink components.

Nexthink is grateful to the authors and contributors of all the high quality open source projects that make possible the development of our own product.

Find the full list of open source software libraries used by Nexthink, along with their corresponding licenses, on every deployed Nexthink Appliance under:

```
/var/nexthink/eula/Libraries_licenses.txt
```

Related references

- Nexthink On-premise License Agreement
- Nexthink On-premise License Agreement (for indirect customers)
- Nexthink On-premise License Agreement (for indirect customers in the META region)
- Cloud Master Services Agreement
- Cloud Terms and Conditions (for indirect customers)
- Cloud Terms and Conditions (for indirect customers in the META region)

- Mutual Confidentiality Terms & Conditions November 2020

What's new in V6.28

In a nutshell

UI improvements

New main navigation

A brand new navigation bar on the left of the screen provides more intuitive and consistent navigation. It also gives more space to dashboards so you can better focus on them.

Digital Experience Score

Digital Experience Score for macOS

To provide a complete overview of the digital experience of your employees, the score now supports both Windows and macOS platforms.

Web & Cloud

Support for Web & Cloud on macOS

Web & Cloud is now available for macOS too, so you can monitor all enterprise web services and applications.

All features

UI improvements

New Portal navigation

The Nexthink Portal has now a brand new navigation bar on the left side, providing a more intuitive and consistent user experience.

macOS Collector

Web & Cloud

To enable better monitoring of all the enterprise web services and applications, Web & Cloud is now available for macOS. Nexthink main use cases are now available both macOS and Windows.

Virtualization

Measure and proactively improve the Digital Employee Experience with new dedicated metrics and data: Citrix Round Trip Time, session network latency, client IP and protocol.

Find out more

Digital Employee Experience Score

All DEX scores are fully compatible with macOS. There is now a single package for both macOS and Windows.

A light version of the DEX Score V2 pack is installed by default when the system detects that there is no preexisting DEX Score V2 pack already present. In order to fully benefit from what the DEX score has to offer you, will still need to install the full Nexthink Library pack.

Data-model changes

Data-model changes

New objects for virtualization

Objects represent items recognized by Nexthink.

User

Users of devices (domain, local or system).

Field	Group	Type			
Average Citrix RTT	Session performance	Aggregate			
	Indicates the Citrix RTT averaged across all session performance events. • Citrix ICA RTT represents the screen lag experienced by the user and it measures the time between a user input and the display of the response. • The measure is only available for Citrix ICA				
	NXQL ID:	average_citrix_rtt			
Average session network latency	Session performance	Aggregate			
	Indicates the session network latency averaged across all session performance events. • It measures the performance of the network. • When the value is too high, the user will see some screen lag and a degraded user experience. • The measure is not available for local session.				
	NXQL ID:	average_session_network_latency			
Session duration	Session performance	Aggregate			
	Indicates the duration of the sessions summed across all session performance events.				

	NXQL ID:	session_duration

Device

Devices are Windows, Mac OS or mobile endpoints

Field	Group	Type			
Average Citrix RTT	Session performance	Aggregate			
	Indicates the Citrix RTT averaged across all session performance events. • Citrix ICA RTT represents the screen lag experienced by the user and it measures the time between a user input and the display of the response. • The measure is only available for Citrix ICA				
	NXQL ID:	average_citrix_rtt			
Average session network latency	Session performance	Aggregate			
	Indicates the session network latency averaged across all session performance events. • It measures the performance of the network. • When the value is too high, the user will see some screen lag and a degraded user experience. • The measure is not available for local session.				
	NXQL ID:	average_session_network_latency			
Session duration	Session performance	Aggregate			
	Indicates the duration of the sessions summed across all session performance events.				
	NXQL ID:	session_duration			

Session

Sessions of a user logged on a device.

Field	Group	Type			
Citrix RTT	Session performance	Field			
	Indicates the maximum Citrix RTT which represents the screen lag experienced by the user. • It measures the time between a user input and the display of the response. • It includes the network time. • The measure is only available for Citrix ICA.				
	NXQL ID:	citrix_rtt			
Client IP	Session performance	Field			
	Indicates the IP of the client connecting to the session.				
	NXQL ID:	client_ip			
Device ID	Device	Field			
	Indicates the ID of the device which hosts the session.				
Device name	Device	Field			
	Indicates the name of the device which hosts the session.				
Device SID	Device	Field			
	Indicates the SID of the device which hosts the session.				
Duration	Properties	Field			
	Indicates the duration of the session performance event.				
	NXQL ID:	duration			
End time	Properties	Field			
	Indicates the end time.				
	NXQL ID:	end_time			
ID	Properties	Field			
	Indicates the session event identifier code.				
	NXQL ID:	id			
Session network latency	Session performance	Field			
	Indicates the maximum session network latency. • It measures the performance of the network. • When the value is too high, the user will see some screen lag and a degraded user experience.				

	• The measure is not available for Console session.				
	NXQL ID:	session_network_latency			
Session protocol	Session performance	Field			
	Indicates the protocol used to connect to the session. The possible values are: • Citrix - ICA • RDP • Local session • Multiple: the user connected with different protocols during the timeframe of the event.				
	NXQL ID:	session_protocol			
Start time	Properties	Field			
	Indicates the start time.				
	NXQL ID:	start_time			
User ID	User	Field			
	Indicates the ID of the user connected to the session.				
User name	User	Field			
	Indicates the name of the user connected to the session.				
User SID	User	Field			
	Indicates the SID of the user connected to the session.				

Web & Cloud for macOS

Here are all the new fields available for macOS.

User

Users of devices (domain, local or system)

Field	Group	Type			
Average incoming web bitrate	Availability	Aggregate			
	Average incoming bitrate of all underlying web requests, consolidated over time				
	NXQL ID:	average_incoming_bitrate			

Average outgoing web bitrate	Availability	Aggregate			
	Average outgoing bitrate of all underlying web requests, consolidated over time				
	NXQL ID:	average_outgoing_bitrate			
Average web request duration	Availability	Aggregate			
	Average time between request and last response byte				
	NXQL ID:	average_request_duration			
Average web request size	Traffic	Aggregate			
	Average size of web requests				
	NXQL ID:	average_request_size			
Average web response size	Traffic	Aggregate			
	Average size of web responses				
	NXQL ID:	average_response_size			
Binary paths	Activity	Aggregate			
	List of executed binary paths (max. 50 paths)				
Incoming web traffic	Traffic	Aggregate			
	Total web incoming traffic				
	NXQL ID:	incoming_traffic			
Lowest observed web protocol version	Activity	Aggregate			
	Lowest protocol version observed in web requests (excluding web requests with unknown protocol version)				
	NXQL ID:	lowest_protocol_version			
Number of domains	Inventory	Aggregate			
	Number of domains				
	NXQL ID:	number_of_domains			
Number of web requests	Activity	Aggregate			
	Number of web requests				
	NXQL ID:	number_of_web_requests			
Outgoing web traffic	Traffic	Aggregate			
	Total web outgoing traffic				
	NXQL ID:	outgoing_traffic			
Protocols used in web requests	Activity	Aggregate			
	Protocols used in web requests (HTTP, TLS, HTTP/TLS)				
	NXQL ID:	protocols_used_in_requests			
Successful HTTP requests ratio	Availability	Aggregate			

	Percentage of successful HTTP requests (1xx, 2xx and 3xx)				
	NXQL ID:	successful_http_requests_ratio			
Total web traffic	Traffic	Aggregate			
	Total web traffic (incoming and outgoing)				
	NXQL ID:	total_web_traffic			
Web interaction time	Activity	Aggregate			
	Indicates the time during which at least one executable is doing HTTP or TLS traffic. This is counted with a 5-minute resolution.				
	NXQL ID:	cumulated_web_interaction_duration			

Device

Devices are Windows, Mac OS or mobile endpoints

Field	Group	Type			
Average incoming web bitrate	Availability	Aggregate			
	Average incoming bitrate of all underlying web requests, consolidated over time				
	NXQL ID:	average_incoming_bitrate			
Average outgoing web bitrate	Availability	Aggregate			
	Average outgoing bitrate of all underlying web requests, consolidated over time				
	NXQL ID:	average_outgoing_bitrate			
Average web request duration	Availability	Aggregate			
	Average time between request and last response byte				
	NXQL ID:	average_request_duration			
Average web request size	Traffic	Aggregate			
	Average size of web requests				
	NXQL ID:	average_request_size			
Average web response size	Traffic	Aggregate			
	Average size of web responses				
	NXQL ID:	average_response_size			
Incoming web traffic	Traffic	Aggregate			
	Total web incoming traffic				
	NXQL ID:	incoming_traffic			

Lowest observed web protocol version	Activity	Aggregate			
	Lowest protocol version observed in web requests (excluding web requests with unknown protocol version)				
	NXQL ID:	lowest_protocol_version			
Number of domains	Inventory	Aggregate			
	Number of domains				
	NXQL ID:	number_of_domains			
Number of web requests	Activity	Aggregate			
	Number of web requests				
	NXQL ID:	number_of_web_requests			
Outgoing web traffic	Traffic	Aggregate			
	Total web outgoing traffic				
	NXQL ID:	outgoing_traffic			
Protocols used in web requests	Activity	Aggregate			
	Protocols used in web requests (HTTP, TLS, HTTP/TLS)				
	NXQL ID:	protocols_used_in_requests			
Successful HTTP requests ratio	Availability	Aggregate			
	Percentage of successful HTTP requests (1xx, 2xx and 3xx)				
	NXQL ID:	successful_http_requests_ratio			
Total web traffic	Traffic	Aggregate			
	Total web traffic (incoming and outgoing)				
	NXQL ID:	total_web_traffic			
Web interaction time	Activity	Aggregate			
	Indicates the time during which at least one executable is doing HTTP or TLS traffic. This is counted with a 5-minute resolution.				
	NXQL ID:	cumulated_web_interaction_duration			
Average incoming web bitrate	Availability	Aggregate			
	Average incoming bitrate of all underlying web requests, consolidated over time				
	NXQL ID:	average_incoming_bitrate			
Average outgoing web bitrate	Availability	Aggregate			
	Average outgoing bitrate of all underlying web requests, consolidated over time				
	NXQL ID:	average_outgoing_bitrate			
Average web request duration	Availability	Aggregate			

	Average time between request and last response byte				
	NXQL ID:	average_request_duration			
Average web request size	Traffic	Aggregate			
	Average size of web requests				
	NXQL ID:	average_request_size			
Average web response size	Traffic	Aggregate			
	Average size of web responses				
	NXQL ID:	average_response_size			
Incoming web traffic	Traffic	Aggregate			
	Total web incoming traffic				
	NXQL ID:	incoming_traffic			
Incoming web traffic per device	Traffic	Aggregate			
	Indicates the incoming web traffic divided by the number of devices.				
	NXQL ID:	incoming_web_traffic_per_device			
Lowest observed web protocol version	Activity	Aggregate			
	Lowest protocol version observed in web requests (excluding web requests with unknown protocol version)				
	NXQL ID:	lowest_protocol_version			
Number of domains	Inventory	Aggregate			
	Number of domains				
	NXQL ID:	number_of_domains			
Number of web requests	Activity	Aggregate			
	Number of web requests				
	NXQL ID:	number_of_web_requests			
Outgoing web traffic	Traffic	Aggregate			
	Total web outgoing traffic				
	NXQL ID:	outgoing_traffic			
Outgoing web traffic per device	Traffic	Aggregate			
	Indicates the outgoing web traffic divided by the number of devices.				
	NXQL ID:	outgoing_web_traffic_per_device			
Protocols used in web requests	Activity	Aggregate			
	Protocols used in web requests (HTTP, TLS, HTTP/TLS)				
	NXQL ID:	protocols_used_in_requests			

Successful HTTP requests ratio	Availability	Aggregate			
	Percentage of successful HTTP requests (1xx, 2xx and 3xx)				
	NXQL ID:	successful_http_requests_ratio			
Total web traffic	Traffic	Aggregate			
	Total web traffic (incoming and outgoing)				
	NXQL ID:	total_web_traffic			
Web interaction time	Activity	Aggregate			
	Indicates the time during which at least one executable is doing HTTP or TLS traffic. This is counted with a 5-minute resolution.				
	NXQL ID:	cumulated_web_interaction_duration			

Executable

Executable programs (e.g. 'winword.exe')

Field	Group	Type			
Average incoming web bitrate	Availability	Aggregate			
	Average incoming bitrate of all underlying web requests, consolidated over time				
	NXQL ID:	average_incoming_bitrate			
Average outgoing web bitrate	Availability	Aggregate			
	Average outgoing bitrate of all underlying web requests, consolidated over time				
	NXQL ID:	average_outgoing_bitrate			
Average web request duration	Availability	Aggregate			
	Average time between request and last response byte				
	NXQL ID:	average_request_duration			
Average web request size	Traffic	Aggregate			
	Average size of web requests				
	NXQL ID:	average_request_size			
Average web response size	Traffic	Aggregate			
	Average size of web responses				
	NXQL ID:	average_response_size			
Incoming web traffic	Traffic	Aggregate			
	Total web incoming traffic				
	NXQL ID:	incoming_traffic			

Incoming web traffic per device	Traffic	Aggregate			
	Indicates the incoming web traffic divided by the number of devices.				
	NXQL ID:	incoming_web_traffic_per_device			
Lowest observed web protocol version	Activity	Aggregate			
	Lowest protocol version observed in web requests (excluding web requests with unknown protocol version)				
	NXQL ID:	lowest_protocol_version			
Number of domains	Inventory	Aggregate			
	Number of domains				
	NXQL ID:	number_of_domains			
Number of web requests	Activity	Aggregate			
	Number of web requests				
	NXQL ID:	number_of_web_requests			
Outgoing web traffic	Traffic	Aggregate			
	Total web outgoing traffic				
	NXQL ID:	outgoing_traffic			
Outgoing web traffic per device	Traffic	Aggregate			
	Indicates the outgoing web traffic divided by the number of devices.				
	NXQL ID:	outgoing_web_traffic_per_device			
Protocols used in web requests	Activity	Aggregate			
	Protocols used in web requests (HTTP, TLS, HTTP/TLS)				
	NXQL ID:	protocols_used_in_requests			
Successful HTTP requests ratio	Availability	Aggregate			
	Percentage of successful HTTP requests (1xx, 2xx and 3xx)				
	NXQL ID:	successful_http_requests_ratio			
Total web traffic	Traffic	Aggregate			
	Total web traffic (incoming and outgoing)				
	NXQL ID:	total_web_traffic			
Web interaction time	Activity	Aggregate			
	Indicates the time during which at least one executable is doing HTTP or TLS traffic. This is counted with a 5-minute resolution.				
	NXQL ID:	cumulated_web_interaction_duration			

Binary

Executable binary files (e.g. 'winword.exe - 10.0.6843')

Field	Group	Type			
Average incoming web bitrate	Availability	Aggregate			
	Average incoming bitrate of all underlying web requests, consolidated over time				
	NXQL ID:	average_incoming_bitrate			
Average outgoing web bitrate	Availability	Aggregate			
	Average outgoing bitrate of all underlying web requests, consolidated over time				
	NXQL ID:	average_outgoing_bitrate			
Average web request duration	Availability	Aggregate			
	Average time between request and last response byte				
	NXQL ID:	average_request_duration			
Average web request size	Traffic	Aggregate			
	Average size of web requests				
	NXQL ID:	average_request_size			
Average web response size	Traffic	Aggregate			
	Average size of web responses				
	NXQL ID:	average_response_size			
Incoming web traffic	Traffic	Aggregate			
	Total web incoming traffic				
	NXQL ID:	incoming_traffic			
Incoming web traffic per device	Traffic	Aggregate			
	Indicates the incoming web traffic divided by the number of devices.				
	NXQL ID:	incoming_web_traffic_per_device			
Lowest observed web protocol version	Activity	Aggregate			
	Lowest protocol version observed in web requests (excluding web requests with unknown protocol version)				
	NXQL ID:	lowest_protocol_version			
Number of domains	Inventory	Aggregate			
	Number of domains				

	NXQL ID:	number_of_domains			
Number of web requests	Activity	Aggregate			
	Number of web requests				
	NXQL ID:	number_of_web_requests			
Outgoing web traffic	Traffic	Aggregate			
	Total web outgoing traffic				
	NXQL ID:	outgoing_traffic			
Outgoing web traffic per device	Traffic	Aggregate			
	Indicates the outgoing web traffic divided by the number of devices.				
	NXQL ID:	outgoing_web_traffic_per_device			
Protocols used in web requests	Activity	Aggregate			
	Protocols used in web requests (HTTP, TLS, HTTP/TLS)				
	NXQL ID:	protocols_used_in_requests			
Successful HTTP requests ratio	Availability	Aggregate			
	Percentage of successful HTTP requests (1xx, 2xx and 3xx)				
	NXQL ID:	successful_http_requests_ratio			
Total web traffic	Traffic	Aggregate			
	Total web traffic (incoming and outgoing)				
	NXQL ID:	total_web_traffic			
Web interaction time	Activity	Aggregate			
	Indicates the time during which at least one executable is doing HTTP or TLS traffic. This is counted with a 5-minute resolution.				
	NXQL ID:	cumulated_web_interaction_duration			

Port

Connection ports (TCP or UDP)

Field	Group	Type			
Average incoming web bitrate	Availability	Aggregate			
	Average incoming bitrate of all underlying web requests, consolidated over time				
	NXQL ID:	average_incoming_bitrate			
Average outgoing web bitrate	Availability	Aggregate			

	Average outgoing bitrate of all underlying web requests, consolidated over time				
	NXQL ID:	average_outgoing_bitrate			
Average web request duration	Availability	Aggregate			
	Average time between request and last response byte				
	NXQL ID:	average_request_duration			
Average web request size	Traffic	Aggregate			
	Average size of web requests				
	NXQL ID:	average_request_size			
Average web response size	Traffic	Aggregate			
	Average size of web responses				
	NXQL ID:	average_response_size			
Incoming web traffic	Traffic	Aggregate			
	Total web incoming traffic				
	NXQL ID:	incoming_traffic			
Incoming web traffic per device	Traffic	Aggregate			
	Indicates the incoming web traffic divided by the number of devices.				
	NXQL ID:	incoming_web_traffic_per_device			
Lowest observed web protocol version	Activity	Aggregate			
	Lowest protocol version observed in web requests (excluding web requests with unknown protocol version)				
	NXQL ID:	lowest_protocol_version			
Number of domains	Inventory	Aggregate			
	Number of domains				
	NXQL ID:	number_of_domains			
Number of web requests	Activity	Aggregate			
	Number of web requests				
	NXQL ID:	number_of_web_requests			
Outgoing web traffic	Traffic	Aggregate			
	Total web outgoing traffic				
	NXQL ID:	outgoing_traffic			
Outgoing web traffic per device	Traffic	Aggregate			
	Indicates the outgoing web traffic divided by the number of devices.				
	NXQL ID:	outgoing_web_traffic_per_device			

Protocols used in web requests	Activity	Aggregate			
	Protocols used in web requests (HTTP, TLS, HTTP/TLS)				
	NXQL ID:	protocols_used_in_requests			
Successful HTTP requests ratio	Availability	Aggregate			
	Percentage of successful HTTP requests (1xx, 2xx and 3xx)				
	NXQL ID:	successful_http_requests_ratio			
Total web traffic	Traffic	Aggregate			
	Total web traffic (incoming and outgoing)				
	NXQL ID:	total_web_traffic			
Web interaction time	Activity	Aggregate			
	Indicates the time during which at least one executable is doing HTTP or TLS traffic. This is counted with a 5-minute resolution.				
	NXQL ID:	cumulated_web_interaction_duration			

Destination

Devices receiving connections

Field	Group	Type			
Average incoming web bitrate	Availability	Aggregate			
	Average incoming bitrate of all underlying web requests, consolidated over time				
	NXQL ID:	average_incoming_bitrate			
Average outgoing web bitrate	Availability	Aggregate			
	Average outgoing bitrate of all underlying web requests, consolidated over time				
	NXQL ID:	average_outgoing_bitrate			
Average web request duration	Availability	Aggregate			
	Average time between request and last response byte				
	NXQL ID:	average_request_duration			
Average web request size	Traffic	Aggregate			
	Average size of web requests				
	NXQL ID:	average_request_size			
Average web response size	Traffic	Aggregate			
	Average size of web responses				

	NXQL ID:	average_response_size			
Incoming web traffic	Traffic	Aggregate			
	Total web incoming traffic				
	NXQL ID:	incoming_traffic			
Incoming web traffic per device	Traffic	Aggregate			
	Indicates the incoming web traffic divided by the number of devices.				
	NXQL ID:	incoming_web_traffic_per_device			
Lowest observed web protocol version	Activity	Aggregate			
	Lowest protocol version observed in web requests (excluding web requests with unknown protocol version)				
	NXQL ID:	lowest_protocol_version			
Number of domains	Inventory	Aggregate			
	Number of domains				
	NXQL ID:	number_of_domains			
Number of web requests	Activity	Aggregate			
	Number of web requests				
	NXQL ID:	number_of_web_requests			
Outgoing web traffic	Traffic	Aggregate			
	Total web outgoing traffic				
	NXQL ID:	outgoing_traffic			
Outgoing web traffic per device	Traffic	Aggregate			
	Indicates the outgoing web traffic divided by the number of devices.				
	NXQL ID:	outgoing_web_traffic_per_device			
Protocols used in web requests	Activity	Aggregate			
	Protocols used in web requests (HTTP, TLS, HTTP/TLS)				
	NXQL ID:	protocols_used_in_requests			
Successful HTTP requests ratio	Availability	Aggregate			
	Percentage of successful HTTP requests (1xx, 2xx and 3xx)				
	NXQL ID:	successful_http_requests_ratio			
Total web traffic	Traffic	Aggregate			
	Total web traffic (incoming and outgoing)				
	NXQL ID:	total_web_traffic			
Web interaction time	Activity	Aggregate			

	Indicates the time during which at least one executable is doing HTTP or TLS traffic. This is counted with a 5-minute resolution.	
	NXQL ID:	cumulated_web_interaction_duration

Domain

Domain names

Field	Group	Type			
Average incoming web bitrate	Availability	Aggregate			
	Average incoming bitrate of all underlying web requests, consolidated over time				
	NXQL ID:	average_incoming_bitrate			
Average outgoing web bitrate	Availability	Aggregate			
	Average outgoing bitrate of all underlying web requests, consolidated over time				
	NXQL ID:	average_outgoing_bitrate			
Average web request duration	Availability	Aggregate			
	Average time between request and last response byte				
	NXQL ID:	average_request_duration			
Average web request size	Traffic	Aggregate			
	Average size of web requests				
	NXQL ID:	average_request_size			
Average web response size	Traffic	Aggregate			
	Average size of web responses				
	NXQL ID:	average_response_size			
Domain category	Properties	Field			
	Indicates the category of the domain: • '-' : not yet tagged or internal domain				
	NXQL ID:	domain_category			
First seen	Properties	Field			
	The first time the domain has been seen				
	NXQL ID:	first_seen			
Hosting country	Properties	Field			

	Indicates in which country the domain is hosted: • '-' : not yet tagged, internal domain or not known by Nexthink Library				
	NXQL ID:	hosting_country			
Hostname	Properties	Field			
	The hostname of the fully qualified domain name				
	NXQL ID:	hostname			
Incoming web traffic	Traffic	Aggregate			
	Total web incoming traffic				
	NXQL ID:	incoming_traffic			
Incoming web traffic per device	Traffic	Aggregate			
	Indicates the incoming web traffic divided by the number of devices.				
	NXQL ID:	incoming_web_traffic_per_device			
Internal domain	Properties	Field			
	Indicates whether the domain is considered internal: • yes: the domain is not reported to Nexthink Library and subdomains are not compressed using the '*' pattern • no: the domain is reported to the Nexthink Library (if the license includes the Security module); complex subdomains are compressed using the '*' pattern				
	NXQL ID:	internal_domain			
Last seen	Properties	Field			
	The last time the domain has been seen				
	NXQL ID:	last_seen			
Lowest observed web protocol version	Activity	Aggregate			
	Lowest protocol version observed in web requests (excluding web requests with unknown protocol version)				
	NXQL ID:	lowest_protocol_version			
Name	Properties	Field			
	The fully qualified domain name				
	NXQL ID:	name			
Number of web requests	Activity	Aggregate			

	Number of web requests				
	NXQL ID:	number_of_web_requests			
Outgoing web traffic	Traffic	Aggregate			
	Total web outgoing traffic				
	NXQL ID:	outgoing_traffic			
Outgoing web traffic per device	Traffic	Aggregate			
	Indicates the outgoing web traffic divided by the number of devices.				
	NXQL ID:	outgoing_web_traffic_per_device			
Protocols used in web requests	Activity	Aggregate			
	Protocols used in web requests (HTTP, TLS, HTTP/TLS)				
	NXQL ID:	protocols_used_in_requests			
Reputation	Properties	Field			
	Indicates the reputation of the domain: • '-': internal domain or not yet tagged • 'trustworthy': clean domain which has not been connected to any security risks • 'low risk': benign domain which rarely delivers dangerous content • 'moderate risk': generally benign domain which has exhibited potentially risky behavior • 'high risk': potentially malicious domain which delivers dangerous content				
	NXQL ID:	threat_level			
Storage policy	Properties	Field			
	Event storage policy for the domain (web request or none)				
	NXQL ID:	storage			
Successful HTTP requests ratio	Availability	Aggregate			
	Percentage of successful HTTP requests (1xx, 2xx and 3xx)				
	NXQL ID:	successful_http_requests_ratio			
Total web traffic	Traffic	Aggregate			
	Total web traffic (incoming and outgoing)				
	NXQL ID:	total_web_traffic			
UID	Properties	Field			

	Indicates the universally unique identifier (based on domain name).				
Web interaction time	Activity	Aggregate			
	Indicates the time during which at least one executable is doing HTTP or TLS traffic. This is counted with a 5-minute resolution.				
	NXQL ID:	cumulated_web_interaction_duration			

Web request

HTTP or TLS requests

Field	Group	Type			
Application name	Application	Field			
	Name of the application which made the web request				
Binary paths	Application	Field			
	Paths of the binary which made the web request				
Binary version	Application	Field			
	Version of the binary which made the web request				
Cardinality	Properties	Field			
	Number of underlying web requests, consolidated over time				
	NXQL ID:	cardinality			
Connections duration	Properties	Field			
	The time between start of the first connection and end of the last underlying connection				
	NXQL ID:	connections_duration			
Device ID	Device	Field			
	Unique identifier code of the web request source				
Device name	Device	Field			
	Indicates the name of the device: • For Windows: NetBios Name • For Mac OS: computer name used on the network				

	• For Mobile: composed by mailbox name and device friendly name				
Device SID	Device	Field			
	Windows security identifier of the web request source				
Domain name	Domain	Field			
	Name of the web request destination domain				
End time	Properties	Field			
	Web request end time, corresponding to the moment when the last underlying TCP connection was closed				
	NXQL ID:	end_time			
Executable name	Application	Field			
	Name of the executable which made the web request				
HTTP status	Properties	Field			
	HTTP response status code				
	NXQL ID:	http_status			
ID	Properties	Field			
	Unique request identifier code				
	NXQL ID:	id			
Incoming web traffic	Traffic	Field			
	Incoming web traffic of all underlying web requests, consolidated over time				
	NXQL ID:	incoming_traffic			
Network response time	Availability	Field			
	Average TCP connection establishment time of all underlying connections, consolidated over time				
	NXQL ID:	network_response_time			
Outgoing web traffic	Properties	Field			
	Outgoing web traffic of all underlying web requests, consolidated over time				
	NXQL ID:	outgoing_traffic			
Port number	Port	Field			
	Port number of the web request				

Protocol	Properties	Field			
	Web request protocol (HTTP, TLS)				
	NXQL ID:	protocol			
Protocol version	Properties	Field			
	Web request protocol version				
	NXQL ID:	protocol_version			
Service related	Properties	Field			
	Indicates whether the web request is related to a configured service: • yes: these requests are always visible by all users • no: depending on the privacy settings, requests not related to a service might not be visible by everyone				
	NXQL ID:	service_related			
Signature ID	Properties	Field			
	ID of the related web request signature, i.e. a user executing a certain process on a particular device which emits requests to a specific domain				
	NXQL ID:	signature_id			
Start time	Properties	Field			
	Web request start time				
	NXQL ID:	start_time			
URL path	Properties	Field			
	Indicates the expression used to match the web request against web-based services with URL path: `'-': the web request did not match against any service with URL path`				
User ID	User	Field			
	Unique identifier code of the user who made the web request				
User name	User	Field			
	Name of the user who made the web request				
User SID	User	Field			

	Indicates the Windows security identifier for the user who made the web request. • For Mac 0S: the value is 'S-0-0' if the user is not in Active Directory				
Web request duration	Properties	Field			
	Average time between request and last response byte of all underlying requests, consolidated over time				
	NXQL ID:	web_request_duration			

What's new in V6.27

All features

In a nutshell

Digital Experience Score

Digital Experience Score Library Pack V2

To help measure digital employee experience and proactively improve it, the new version of the Digital Experience Score pack is enhanced with more accurate data that can be customized for the needs of each organization.

Find out more

Data collection

Application startup duration

To provide more information on which applications are affecting digital employee experience, the Collector now measures the time from process start until a window is displayed.

Find out more

Disk type

Solid-state drives (*SSDs*) offer a speed advantage over the more traditional mechanical hard disk drives (*HDDs*), usually at a higher cost per unit of storage, and provide an overall better experience. This higher read speed makes SSDs ideal for use as system disk to boot up your devices and load your most used applications faster.

With Nexthink V6.27, know whether your devices are equipped with either an SSD or an HDD as system disk.

Find out more

Boot duration and fast startup

Fast startup is a feature first introduced in Windows 8 and continued in Windows 10 that reduces the time that a device needs to boot up. Because a fast startup sequence does not follow the same steps as a full boot, Nexthink versions previous to V6.27 were not considering fast startups as boot events.

To better measure the actual number of system boots and the average boot duration of devices, Nexthink V6.27 regards fast startups as genuine boot events. Individual boot events in Nexthink indicate now whether they correspond to a full boot or a fast startup.

In addition, the accuracy of the boot duration has been improved to better reflect the employee experience. In previous versions of Nexthink, the start of the kernel and the launching of the `LogonUI.exe` process were respectively taken as the start and stop events of the boot sequence. The boot duration is now measured from the instant an employee presses the power button to the moment the sign-in screen shows up.

Find out more

Act

Support for Act on MacOS

To allow seamless deployment, Bash is now used as supported scripting language for MacOS.

Find out more

APIs and Integrations

Audit trail API

Conveniently retrieve the audit logs from Nexthink Appliances to automate the inspection of security-relevant activity of Nexthink users.

Find out more

NXT protocol additions

Launch the Finder via a URL (thanks to the `nxt://` protocol) to edit a particular campaign or remote action.

Find out more

Data-model changes

Application startup duration

New fields have been added to support the new application startup duration data.

Field	Group	Type			
Average application startup duration	Activity	Aggregate			
	Indicates the average time between the start of the process and the time a window is displayed (not taking into account the splash screen)				
	NXQL ID:	average_process_start_time			

Field	Group	Type			
Startup duration	Properties	Field			
	Indicates the time between the start of the process and the time a window is displayed (not taking into account the splash screen). The value is averaged over all underlying executions.				
	NXQL ID:	startup_duration			

Boot improvements

Boot-related fields have had "system" removed from their names and new fields related to boots have been added to reflect the improvements in recording system boots.

Field	Group	Type			
Average boot duration	Startup	Aggregate			
	Indicates the average (full or fast startup) boot duration.				

	NXQL ID:	average_boot_duration
Fast startup boot duration baseline	Startup	Field
	Indicated the fast startup boot duration averaged over the fast startups. In the calculation, recent boots weigh more than older boots (exponentially weighted moving average).	
	NXQL ID:	average_fast_startup_duration
Full boot duration baseline	Startup	Field
	Indicated the full boot duration averaged over the last full boots. In the calculation, recent full boots weigh more than older full boots (exponentially weighted moving average).	
	NXQL ID:	average_boot_duration
Last boot time	Startup	Field
	Indicates the time of the last (full or fast startup) boot.	
	NXQL ID:	last_system_boot
Last boot time duration	Startup	Field
	Indicates the duration of the last (full or fast startup) boot.	
	NXQL ID:	last_boot_duration
Number of boots	Startup	Aggregate
	Indicates the number of (full or fast startup) boot.	
	NXQL ID:	number_of_boots
Number of days since last full boot	Startup	Field
	Indicates the number of days since the last full boot.	
	NXQL ID:	number_of_days_since_last_boot

Field	Group	Type
Boot type	Properties	Field
	Indicates the boot type. The possible values are: • full boot • fast startup	

	NXQL ID:	boot_type

Disk drive improvements

In line with the improved reporting around disk type and health, several new fields have been added. Several disk-related fields have also been made available for macOS.

Field	Group	Type			
System disk type	Hardware	Field			
	Indicates the type of the disk from which the device is booting. • HDD • SSD "-" : data not available				
	NXQL ID:	boot_disk_type			
Total non-system drive capacity	Local drives	Field			
	Total capacity of all non-system drives				
	NXQL ID:	total_nonsystem_drive_capacity			
Total non-system drive free space	Local drives	Field			
	Total free space on all non-system drives				
	NXQL ID:	total_nonsystem_drive_free_space			
Total non-system drive usage	Local drives	Field			
	Total use percentage of all non-system drives				
	NXQL ID:	total_nonsystem_drive_usage			

Application not responding aggregation on macOS

Following on from changes in V6.25, aggregations over device errors are now available for macOS.

Field	Group	Type			
Application not responding event ratio	Errors	Aggregate			
	Indicates the number of application not responding events per 100 executions.				
	NXQL ID:	application_not_responding_event_ratio			

What's new in V6.26

All features

Apollo Design

Apollo design is now available for on-premise customers

To ensure visual consistency, the new Apollo design is now available for on-premise customers.

Customized login screen is available.

Modules and widgets have a new look and feel:

When migrating from V6.25 or earlier, scroll bars may appear in preexisting densely-packed dashboards because of the additional spacing between widgets of the new design.

Software metering dashboard has new design and simplified interaction.

Device locator has a new design as well.

Engage

Campaign auto dashboards

The following content applies exclusively to the Nexthink Cloud offering.

To gain more visibility on running campaigns, a new dashboard is automatically created when a campaign is published. The dashboard provides an overview of each campaign as well as its results.

Campaign dashboards will be available by the end of May.

Act

Act API v2

The new version of the Act API enables targeting particular Engines when triggering the execution of a remote action.

Find out more

Proxy support

To reach the Nexthink Cloud from a corporate network, both Finder and Collector support the connection through a proxy server.

Finder proxy support

The Finder supports connection through a proxy server by using the proxy settings specified in Windows. When connecting to the Nexthink Cloud, the Finder communicates both with the Portal and with the Engines through a single port (TCP 443).

Find out more

Mac Collector proxy support

Specify the custom proxy settings for the Mac Collector either manually or by providing a *proxy auto-configuration (PAC)* file during installation.

Find out more

Windows Collector improved proxy support

The Windows Collector now supports Integrated Windows Authentication when connecting through a proxy.

In addition to the system settings, you can now configure the proxy settings for the Collector either manually or by providing a dedicated PAC file expressly for the Windows Collector.

Specify the proxy settings during installation via the Nexthink Collector Installer or after installation with the Collector Configuration Tool (Nxtcfg).

Find out more

Windows Virtual Desktop

The Collector supports the installation on *Windows Virtual Desktop (WVD)*, Microsoft's VDI for Azure. Check out the especial conditions that apply when installing the Collector on WVD, in particular for multi-session Windows 10.

Find out more

More data from Active Directory

The following content applies exclusively to the Nexthink Cloud offering.

User objects hold four additional fields of data coming from Active Directory:

- Location
- Locality name
- Country code
- Organizational unit name.

These fields will be populated by the next version of the Data Enricher; therefore, although present in all Engines, the fields will hold meaningful data in Nexthink Cloud only.

Find out more

Data-model changes

Data from Active Directory

The following content applies exclusively to the Nexthink Cloud offering.

User objects hold four additional fields of data coming from Active Directory:

- Location
- Locality name
- Country code

- Organizational unit name.

These fields will be populated by the next version of the Data Enricher; therefore, although present in all Engines, the fields will hold meaningful data in Nexthink Cloud only.

Field	Group	Type			
Country code	Properties	Field			
	Country/Region, represented as a 2-character code based on ISO-3166, as listed in Active Directory.				
	NXQL ID:	country			
Locality name	Properties	Field			
	The user's locality as city or town, as listed in Active Directory.				
	NXQL ID:	locality			
Location	Properties	Field			
	The user's location as listed in Active Directory.				
	NXQL ID:	location			
Organizational unit name	Properties	Field			
	The name of the organizational unit, as listed in Active Directory.				
	NXQL ID:	org_unit			

What's new in V6.25

All features

Nexthink Cloud

The following content applies exclusively to the Nexthink Cloud offering.

Apollo Design

To modernize user experience, Cloud customers have a refreshed Portal interface that introduces the visual components of the in-house developed Apollo Design system.

Reduce eye strain in low ambient light conditions thanks to the new **Dark** mode.

Login screen, modules and widgets have a new look look and feel.

Collector proxy support

To facilitate communication to the cloud, the Collector supports HTTP and SOCKS5 proxy.

Identity and proxy support

Identity Brokers provides simplified onboarding and improved user experience.

Data Enricher

The Data Enricher complements the information about users and destinations that Collectors send to your Nexthink Cloud instance.

Find out more

Engage

Do Not Disturb period (Windows)

To avoid interrupting employees too frequently, configure the new *Do Not Disturb* period in the Portal. Within this period, an employee who just answered a campaign does not receive any other campaign notification. The default Do Not Disturb period is 6 hours.

For matters that cannot wait, override the Do Not Disturb period with urgent campaigns. When creating a new campaign in the Finder, set the urgency of the campaign to be either **urgent** or **non-urgent**. Urgent campaigns are displayed as soon as they are published, regardless of the Do Not Disturb period. For their part, non-urgent campaigns respect the Do Not Disturb period.

Find out more

GetSID API

The *GetSID API* lets you retrieve the Security Identifier (SID) of end users.

Find out more

macOS Collector

Report device errors on macOS

The Mac Collector now reports macOS kernel panic errors (as system crashes) and better detects all kinds of hard resets.

Collector Assignment

Improvements to the rule-based assignment of Collector to Engines and entities include:

- The use of the new **Collector string tag** in the assignment rules.
- Assignment rules for entities only (and not for Engines).
- The dynamic reassignment of Collectors when assignment rules change.
- The assignment simulation takes into account whether rule-based assignment is actually enabled in Collectors.

Find out more

Support for Windows 10 version 1909

Nexthink supports the latest update of Windows 10, namely version 1909. The Windows Collector has been extensively tested on this new version of the popular operating system and can be confidently installed on any device that runs Windows 10 version 1909.

Deprecated features

Print monitoring dropped

Since version V6.18, print monitoring has been disabled by default in Nexthink. Starting from V6.25, print monitoring is officially deprecated; which means that the feature can still be enabled, but it is no longer supported and it can be completely removed from a future version of Nexthink.

Mobile unsupported in Nexthink Cloud offering

The Nexthink Cloud offering supports both Windows and Mac OS platforms, but not Mobile. Therefore, no data related to mobile devices are available through managed appliances that belong to the Nexthink Cloud offering.

Discontinued support for macOS Sierra

Although the Mac Collector V6.25 can still run on macOS 10.12 Sierra, official support for this platform is discontinued, as Apple no longer supports this version of the popular operating system. Future versions of the Mac Collector are not guaranteed to run on macOS 10.12 Sierra.

Data-model changes

Device errors on MacOS

The Mac Collector now reports macOS kernel panic errors (as system crashes) and hard resets.

Field	Group	Type			
Number of application not responding events	Errors	Aggregate			
	Number of application not responding events				
	NXQL ID:	number_of_application_not_responding_events			
Number of system crashes	Errors	Aggregate			
	Indicates the number of system crashes.				

New Collector String Tag

Improvements to the rule-based assignment of Collector to Engines and entities include the use of the new Collector string tag in the assignment rules.

Find out more

Field	Group	Type			
Collector string tag	Nexthink Collector	Field			
	Indicates the Collector string tag				
	NXQL ID:	cltr_string_tag			

What's new in V6.24

New features

Engage

Updated appearance of Engage notifications

To improve the readability of questions and answers, campaign notifications have a refreshed look and feel on the Windows operating system, approaching the philosophy of Engage on macOS.

Starting from Windows Collector V6.24, answers to campaign questions are arranged vertically and the width of notifications is kept uniform, so that end users experience a gentler transition between questions.

Find out more

Collector connectivity and assignment

Collector communicates via TCP port 443 by default

To simplify network management, the Collector now sends all data through TCP port 443 by default. Sending data through a single well-known port reduces the chances of firewalls blocking Collector traffic. Communication through TCP port 443 cannot rely on the default set of certificates generated during Appliance federation though. To enable the communication of Collector data through TCP port 443, replace the default digital certificates in the Appliance by your own custom certificates or reload them in case of migration. When migrating, Appliances keep their previous configuration to communicate with your installed base of Collectors; therefore you need to make the changes manually. Contact Nexthink Customer Success Services in case of doubt.

Nevertheless, it is still possible to configure Collectors to either send all data through a custom TCP port (above TCP 1024) or send activity data through UDP, as in previous versions of Nexthink; although the latter is no longer recommended.

The Cloud offering of Nexthink requires a custom TCP port for the Collector (default 8443), as TCP port 443 is not supported yet.

Find out more

Local IP address to assign Collectors

The Collector now reports the local IP address of the device; that is, the IP address of the device in the local network. The local IP address provides an alternative to the conventional IP address of the device, which is obtained from the source IP address of Collector packets and thus subject to change in transit if network address translation takes place; that is, if the device and the Engine lie on different networks. This is always the case, for example, when Collectors report to an instance of Nexthink in the Cloud.

The new local IP address can be used in place of or in combination with the conventional IP address of the device to write the rules for assigning Collectors to Engines.

Find out more

Assigning Collectors to Appliances with multiple names

When entering the network parameters of the Nexthink Appliances, administrators can specify more than one fully qualified domain name (DNS name) or IP address per Appliance. Remember that only the first of the external DNS names specified is used for rule-based Collector assignment.

Find out more

Finder improvements

Score tabs navigation

Both the user and the device views in the Finder display up to ten score tabs. On some screens, not all tabs fit simultaneously.

Now the Finder includes a navigation tool that let you scroll through the score tabs when they do not fit on the screen.

Find out more

Cross-Engine search without auto-complete

To enable search across Engines in setups where the connectivity between Finder and Portal is limited, specify a new intermediate level of Cross-Engine features that includes the search but not the auto-complete feature, which might be too costly in terms of network resources.

Find out more

APIs and integrations

Two new APIs let you retrieve information about services and connected Engines programmatically from the Portal Appliance and NXQL now fully supports JSON output.

Services API

Programmatically retrieve data about the health and performance of the IT services monitored by Nexthink as perceived by the end users.

Find out more

List Engines API

Programmatically retrieve the list of Engines that are linked to the Portal, including their connection status.

Find out more

NXQL fully supports JSON output

Responses to NXQL queries can come in different formats. The available output formats are CSV, HTML, XML and JSON. The JSON output format was offered as a technical preview until now. Starting from V6.24, JSON format is fully supported as output format for NXQL queries.

Find out more

Login and access

Protection of local accounts

To protect local accounts against brute force attacks, a local account is temporarily blocked after five failed login attempts.

Find out more

Portal error pages

The default server pages that indicate an error in the connection of the web browser to the Portal have been replaced by custom error pages that keep the look and feel of the Portal.

Find out more

Report application not responding events in macOS

The Mac Collector now reports *application not responding* events. Note that the semantics of a non-responding application in macOS are different from Windows.

Find out more

Helping Support diagnose your issues

After contacting Nexthink Support, you are usually requested to download a diagnostics script to your Appliances. The results of executing this Support script are an invaluable resource for the Support team to pinpoint the cause of any issue on your Appliances. However, the script had to be manually downloaded and executed from the CLI of the impacted Appliances.

Starting from V6.24, the Support script is included in all Nexthink Appliances by default and you can run it comfortably from the Web Console.

Find out more

Data-model changes

Last local IP address

In line with improvements around local IP address reporting, the last local IP address has been added to the data model for devices.

Field	Group	Type			
Last local IP address	Network	Field			
	Indicates the local IP address of the device. This field requires a collector version newer than 6.23 and connected through TCP.				
	NXQL ID:	last_local_ip_address			

What's new in V6.23

New features

Engage

Trigger campaign API

To engage with the end user through third-party tools (for instance, to automatically launch a satisfaction survey from a ticketing system after a ticket is closed), trigger campaigns programmatically thanks to the new Engage API.

Find out more

Increased number of custom fields available for campaigns

The number of custom fields dedicated to Engage campaigns has been increased from 150 to 500 custom fields. In practice, this means that you can run more campaigns simultaneously.

Find out more

Portal APIs

Get data from the Portal programmatically thanks to two new APIs.

Count metrics API

Retrieve the details of count metrics to know about the objects that took part in the count and their attributes.

Find out more

Software Metering API

Get statistics about actual program usage to optimize the purchase of software licenses.

Find out more

Digital Experience Score on Mac devices

Assess the Digital Employee Experience of all employees that use Mac devices and take action to continuously improve it.

Find out more

Data collection

Collector support for recent OS updates

The Collector supports the latest versions of the popular operating systems on which it runs:

- Windows 10, version 1903
- macOS 10.15 Catalina

Find out more

Collector configuration available in Finder

Find out the configuration options of every deployed Collector comfortably from the Finder without having to connect to each device individually. Leverage investigations to look for unusual or unwanted Collector setting.

The Collector requires an active TCP connection with the Engine to send its configuration information.

Find out more

User Management

Password of local accounts

For centralized user management and improved security, Nexthink recommends provisioning individual accounts from corporate identity management solutions (SAML-compliant or Active Directory). Local accounts are still useful for creating accounts to call Nexthink APIs.

As a first step to make local accounts more secure in Nexthink, there is a new requirement on the minimum password length of local accounts. By default, new passwords must be eight characters long (no impact on existing passwords).

This minimum length is configurable.

Find out more

Data-model changes

Additional reported Collector fields

The following fields concerning the Collector have been added. Note that **Collector assignment license UID** is not new, but is now accessible via NXQL.

Field	Group	Type			
Collector assignment	Nexthink Collector	Field			
	Indicates whether Collector assignment service is enabled or disabled • disabled: indicates that the Collector feature is disabled • enabled: indicates that the Collector feature is enabled "-" : data not available				
	NXQL ID:	cltr_ca_status			
Collector assignment license UID	Nexthink Collector	Field			
	Indicates the Collector assignment license UID				
	NXQL ID:	cltr_ca_license_uid			
Collector CrashGuard count	Nexthink Collector	Field			
	Indicates the number of consecutive hard resets or system crashes of the device				
	NXQL ID:	cltr_crash_guard_count			
Collector CrashGuard limit	Nexthink Collector	Field			
	Indicates the Collector CrashGuard limit				
	NXQL ID:	cltr_crash_guard_limit			
		Field			

Collector CrashGuard protection interval	Nexthink Collector				
	Indicates the CrashGuard monitoring interval in minutes				
	NXQL ID:	cltr_crash_guard_protection_interval			
Collector CrashGuard reactivation interval	Nexthink Collector	Field			
	Indicates the Collector CrashGuard reactivation interval in hours				
	NXQL ID:	cltr_crash_guard_react_interval			
Data transport protocol	Nexthink Collector	Field			
	Specifies if the Collector data is sent over TCP or UDP • UDP: the Collector data traffic is sent over UDP • TCP: the Collector data traffic is sent over TCP "-" : data not available				
	NXQL ID:	cltr_data_channel_protocol			
Engage	Nexthink Collector	Field			
	Indicates whether Engage is enabled or disabled • enabled: indicates that the status of Engage service in Collector is enabled • enabled except on server OS: indicates that the status of Engage service in Collector is enabled on all devices except on servers • disabled: indicates that the status of Engage service in Collector is disabled "-" : data not available				
	NXQL ID:	cltr_engage_service_status			
IP protocol DNS resolution	Nexthink Collector	Field			
	Indicates the DNS resolution preference for Collector in terms of IP protocol version on the device • IPv4: prefer IPv4				

	• IPv6: prefer IPv6 "-" : data not available				
	NXQL ID:	cltr_dns_res_preference			
Message maximum segment size	Nexthink Collector	Field			
	Indicates the maximum segment size of packets sent by Collector				
	NXQL ID:	cltr_max_segment_size			
Monitoring of unresponsive applications	Nexthink Collector	Field			
	Indicates whether the Collector is monitoring for unresponsive applications on the device • disabled: indicates that the Collector feature is disabled • enabled: indicates that the Collector feature is enabled "-" : data not available				
	NXQL ID:	cltr_freezes_monitoring			
Packages and updates scan interval	Nexthink Collector	Field			
	Indicates the interval, in hours, after which the Collector checks for newly installed packages and updates				
	NXQL ID:	cltr_installs_scan_interval			
Print monitoring	Nexthink Collector	Field			
	Indicates whether the Collector printing monitoring is enabled or disabled • disabled: indicates that the Collector feature is disabled • enabled: indicates that the Collector feature is enabled "-" : data not available				
	NXQL ID:	collector_print_monitoring_status			

Script execution policy	Nexthink Collector	Field			
	Indicates the Powershell script execution policy • unrestricted: indicates that Act service in Collector can execute any kind of scripts • signed, trusted: indicates that Act service in Collector can only execute scripts signed by a trusted authority • signed, trusted or nexthink: indicates that Act service in Collector can only execute scripts signed by a trusted authority or by Nexthink • disabled: indicates that Act service in Collector cannot execute scripts "-" : data not available				
	NXQL ID:	cltr_ra_execution_policy			
SMB print monitoring	Nexthink Collector	Field			
	Indicates whether SMB printing monitoring is enabled or disabled • disabled: indicates that the Collector feature is disabled • enabled: indicates that the Collector feature is enabled "-" : data not available				
	NXQL ID:	cltr_smb_print_mon_status			
VDI/Kiosk support	Nexthink Collector	Field			
	Indicates whether the Collector reports user logon events and user interactions in virtualized and embedded (kiosk mode) environments • disabled: indicates that the Collector feature is disabled				

	• enabled: indicates that the Collector feature is enabled "-" : data not available				
	NXQL ID:	cltr_custom_shells			
Visibility from Add or Remove Programs	Nexthink Collector	Field			
	Indicates whether Collector is hidden in the "Add or Remove Programs" • invisible: indicates that the Collector application is not shown in the "add or remove programs" list • visible: indicates that the Collector application is shown in the "add or remove programs" list "-" : data not available				
	NXQL ID:	cltr_is_visible			
Web & Cloud monitoring	Nexthink Collector	Field			
	Indicates whether Web & Cloud monitoring is enabled or disabled • disabled: indicates that the Collector feature is disabled • enabled: indicates that the Collector feature is enabled "-" : data not available				
	NXQL ID:	cltr_web_mon_status			

User UID now accessible via NXQL

Field	Group	Type			
UID	Properties	Field			
	Indicates the universally unique identifier (based on user SID).				
	NXQL ID:	user_uid			

What's new in V6.22

New features

Engage

Support more languages

To increase the chances of addressing end-users in their mother tongue, the user interface of campaign notifications now supports additional languages. Remember to update all Collectors to ensure compatibility with the new languages.

Find out more

Increased limit on published campaigns

Create and publish more campaigns to better engage with the end users. Starting from V6.22, the limit on the number of campaigns that you can publish is linked to the type of campaign:

- 15 one-off or recurring campaigns.
- 15 continuous satisfaction measurement campaigns.
- 100 manual campaigns.
- 1000 campaigns embedded in remote actions.

Note however that the limit on the number of available custom fields may actually be more restrictive than the limit on the number of campaigns.

Find out more

User management

Support for single sign-on with Azure AD

To broaden the choice of single sign-on solutions, Azure AD is now officially supported in addition to AD FS to simplify the login experience of Nexthink users.

Instructions are given on how to configure Azure AD to act as a SAML identity provider for the Nexthink Portal.

Find out more

Just-In-Time provisioning of user accounts with SAML

Avoid manually adding users to your system -- a process that can be both tedious and error prone -- by leveraging SAML to provide relevant user information on user logon.

Find out more

Hardening of local accounts

To enforce end users to log in to Nexthink via a corporate account and thus improve security, disable local accounts for interactive users.

Find out more

Data Collection

Logging the connection status of the Collector

To improve the visibility of the status of the connection between the Collector and the Nexthink Appliance, the changes in the connection status, as well as related errors, are now logged.

- Windows Collectors log messages to the Windows Event Log.
- Mac Collectors log messages to the system log.

Find out more

Reduce Collector unloading by CrashGuard

When the Collector driver has issues with other kernel drivers in a device, the *CrashGuard* mechanism prevents the device from repeatedly crashing time and again by stopping the loading of the Collector driver on startup. The

CrashgGuard protection is triggered when the Collector detects that the device is reset a few consecutive times within a particular time interval after being started.

To reduce the number of false positives, the default values that trigger the CrashGuard mechanism have been updated. Starting from V6.22, the Collector must detect five hard resets, each one within the first four hours after the device has started, to trigger the CrashGuard protection.

Find out more

Miscellaneous

Copy and paste dashboards and widgets

Reusing Portal content has been greatly simplified by introducing the ability to copy and paste individual widgets and full dashboards.

Create dashboards faster by reusing widgets from other dashboards or by starting from an existent dashboard, and not from scratch.

Find out more

Additional audit logs

The Nexthink Appliance V6.22 registers additional messages in its log to let you find out the relevant activities that took place in your setup: successful or failed logons, changes in the configuration, component start and stop, etc.

Find out more

Data-model changes

Collector assignment license UID field added

In order to improve license management by ensuring devices can be clearly, reliably and uniquely identified across Engines, the field **Collector assignment license UID** has been added to Device objects.

Field	Group	Type			
Collector assignment license UID	Nexthink Collector	Field			
	Indicates the Collector assignment license UID				

User UID field added to NXQL data model

To make the unique identifier of users available through NXQL.

Name	Type				Properties
user_uid	md5				
	Indicates the universally unique identifier				

Security upgrade

Nexthink strongly recommends that customers upgrade all Windows Collectors to version V6.22.2.10, released on September 5, 2019, and downloadable from here.

This new version addresses a couple of currently known security vulnerabilities.

Related references

- Windows Collector V6.22.X (Release Notes)
- Nexthink V6.22 Release (Product Downloads)
- Security bulletin (Knowledge base)