NICE WebRTC Redirection

Customer Readiness and Validation Guide

Audience: Network / security engineers, VDI administrators validating readiness
Status: Draft
Last updated: September 24, 2026

For leadership infrastructure actions to adopt NICE WebRTC Redirection, see the Customer Readiness Executive Brief.

1. Purpose

NICE WebRTC Redirection improves real-time voice performance in hosted desktop environments by moving WebRTC media processing from the hosted environment to the user's physical endpoint.

This guide describes the requirements that should be reviewed before deploying NICE WebRTC Redirection, including:

- Supported endpoint operating systems and browsers
- Browser extension deployment
- Endpoint software deployment
- Network and firewall requirements
- NICE STUN connectivity
- NICE CXone WebRTC connectivity
- Audio-device requirements
- Potential conflicts with other WebRTC redirection technologies
- Validation procedures
- Fallback behavior
- Diagnostic information to collect when troubleshooting

The objective is to ensure that the customer's environment is capable of establishing a redirected WebRTC session before production rollout.

---

2. Solution Overview

In a traditional hosted desktop configuration, the CXone application and its WebRTC connection execute within the hosted desktop environment. Audio must then be transported between the hosted desktop and the user's physical endpoint using the desktop platform's audio-remoting technology.

NICE WebRTC Redirection changes this architecture.

The CXone application continues to execute within the hosted browser session, while WebRTC media processing is redirected to software running on the user's physical endpoint.

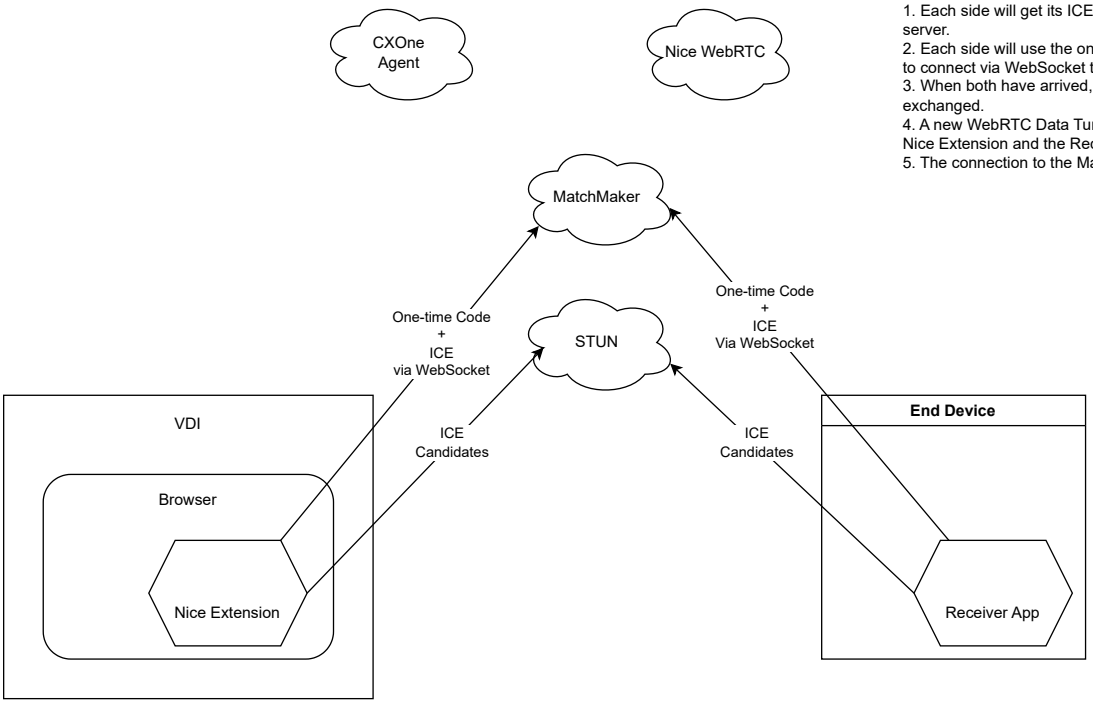
When the **Receiver App** path is active:

1. The NICE browser extension detects the WebRTC session in the hosted browser.
2. The browser extension communicates with the NICE WebRTC Receiver running on the physical endpoint.
3. The browser extension and Receiver are securely associated through the NICE matchmaking service.
4. The WebRTC connection is established from the physical endpoint.
5. The physical endpoint communicates directly with the applicable NICE CXone WebRTC infrastructure.
6. The endpoint's local microphone and audio-output devices are used for the call.

The NICE matchmaking service coordinates the browser and endpoint sessions but is **not part of the voice-media path**.

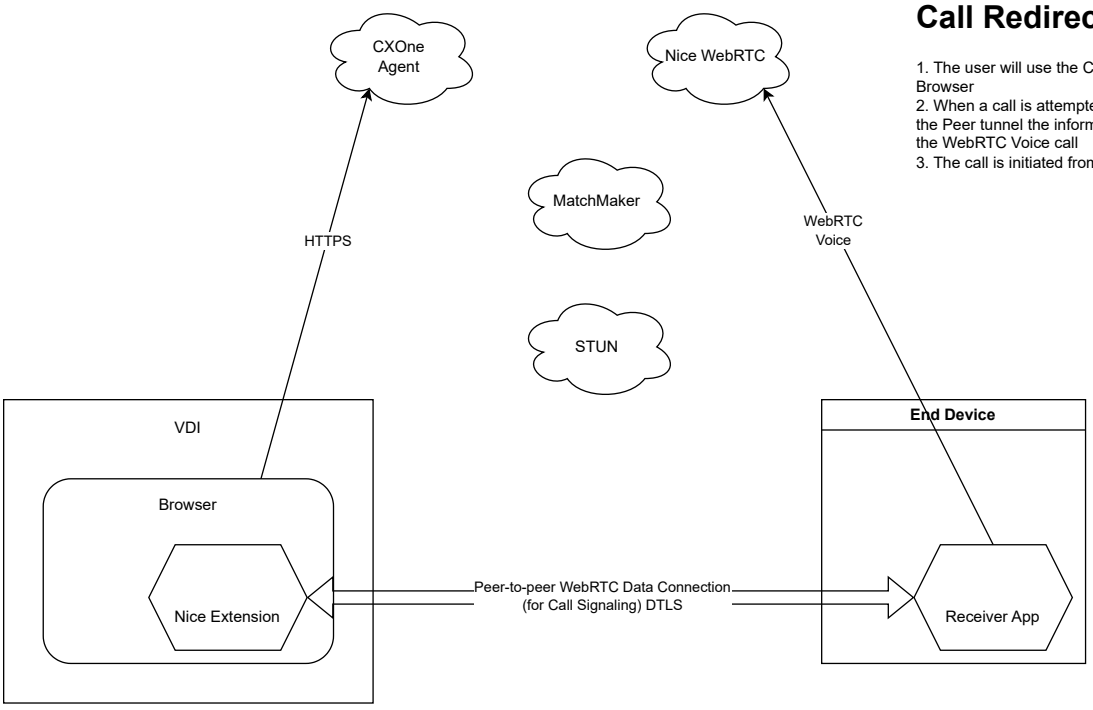
Matchmaking

1. Each side will get its ICE candidates from the STUN server.
2. Each side will use the one-time code (SPAKE 2 encrypted) to connect via WebSocket to the Matchmaker Service
3. When both have arrived, their ICE information is exchanged.
4. A new WebRTC Data Tunnel is established between the Nice Extension and the Receiver App
5. The connection to the Matchmaker service is then dropped



Call Redirection

1. The user will use the CXOne Agent web inside of the VDI Browser
2. When a call is attempted, the Nice Extension instead signals via the Peer tunnel the information that would be required to complete the WebRTC Voice call
3. The call is initiated from the Receiver App



Matchmaking

1. Each side obtains ICE candidates from the STUN service.
2. Each side uses the one-time code to connect over WebSocket to the matchmaking service.
3. When both have arrived, ICE information is exchanged.
4. A WebRTC data tunnel is established between the NICE browser extension and the Receiver.
5. The connection to the matchmaking service is then dropped.

Call redirection

1. The agent uses CXone Agent in the hosted browser.
2. When a call is attempted, the NICE extension signals the information required for the WebRTC voice call over the peer tunnel.
3. The call is initiated from the Receiver on the physical endpoint.

Redirected Media Path

Application and control path:

`CXone Agent → NICE Browser Extension → Secure Matchmaking Service → NICE WebRTC Receiver`

Voice media path:

`Physical Endpoint ⇄ NICE CXone WebRTC Infrastructure`

Voice media does not transit the NICE WebRTC Redirection matchmaking service.

---

3. Graceful Audio Fallback

NICE WebRTC Redirection is designed to fail gracefully.

If a redirected WebRTC session cannot be established, CXone can continue using the normal hosted-desktop audio path:

`CXone → Hosted Desktop Audio → Physical Endpoint`

This allows the agent to continue handling calls even when WebRTC redirection is unavailable.

When fallback occurs:

- The call can continue to operate.
- Audio is transported through the hosted desktop environment.
- The performance and network benefits of endpoint WebRTC redirection are not present.
- The NICE browser extension displays an **unhappy phone** status indicator to show that WebRTC redirection is not active.

> **Important:** A successful CXone call does not, by itself, confirm that NICE WebRTC Redirection is functioning. Validation must confirm that the WebRTC session has actually been redirected to the physical endpoint.

---

4. Supported Environment

The initial NICE WebRTC Redirection release supports the following environments.

Component	Supported
Endpoint Operating System	Microsoft Windows 10/11 x64
Endpoint Operating System	Dell ThinOS 10.x (firmware 2508+)
Endpoint Operating System	ChromeOS (enterprise-managed)
Browser	Google Chrome
Browser	Microsoft Edge
Hosted Desktop Persistence	Persistent or non-persistent

HP ThinPro, IGEL, and macOS endpoints are outside the scope of this product documentation set.

A persistent hosted desktop is **not required**.

For non-persistent environments, the required NICE browser extension and browser policies must be made available each time the user session is created.

4.1 Connection styles by endpoint

Endpoint OS	Citrix UCSDK (in VDI browser)	Omnissa Horizon	Receiver App
Windows	Expected primary path	Expected when Horizon Client + GPO configured	Supported when `MediaProviderPreference` = `MindfireRedirector`
Dell ThinOS 10.x	Via Citrix Workspace App for Linux where available	**Not covered** in this documentation set	Supported via App Builder / WMS
ChromeOS	Use Receiver path (or CWA HTML5 with Receiver)	**Not covered**	MV3 Receiver extension on managed Chromebooks

Policy guides:

- Citrix UCSDK Policy Guide
- Omnissa Horizon Policy Guide
- Receiver App Deployment Guide

4.2 Citrix VDA prerequisite — ProcessWhitelist

Citrix browser-based UCSDK requires the VDA image to allow the **browser executable** on the HDX WebSocket service process allowlist. This is independent of endpoint OS. A missing entry presents as **total redirection failure** (provider badge `N` instead of `C`).

On each Citrix VDA image:

```
HKLM\SOFTWARE\WOW6432Node\Citrix\WebSocketService
ProcessWhitelist  (REG_MULTI_SZ)
        chrome.exe
        msedge.exe
```

Enter one executable name per line. Restart `CtxHdxWebSocketService` or reboot the VDA after changing the value.

For the full Citrix policy set (Web Studio Teams redirection, BCR exclusion, Workspace App, Chrome/Edge force-install, and production checklist), see the Citrix UCSDK Policy Guide.

---

5. Required Components

NICE WebRTC Redirection consists of two primary customer-side components.

5.1 NICE Browser Extension

The NICE browser extension runs in Chrome or Microsoft Edge within the environment where CXone Agent is executing.

The extension is responsible for:

- Detecting supported CXone WebRTC sessions
- Coordinating redirection with the physical endpoint
- Exchanging the information required to establish WebRTC on the endpoint
- Monitoring the redirection state
- Providing a visual indication when redirection cannot be established

5.2 NICE WebRTC Receiver

The NICE WebRTC Receiver runs on the user's physical endpoint.

The Receiver is responsible for:

- Connecting securely to the NICE matchmaking service
- Associating itself with the corresponding browser session
- Establishing the redirected WebRTC connection
- Performing ICE candidate gathering
- Using the NICE STUN service (`stun.nice-receiver-app.com`) for NAT discovery
- Accessing local microphone and audio-output devices
- Sending and receiving WebRTC media directly with NICE CXone

---

6. Browser Extension Deployment Preparation

The NICE WebRTC Redirection browser extension requires enterprise-managed deployment. Force-install **one** distribution channel per browser profile.

Channel	Hosted-session extension ID	Update URL
Enterprise host	`oaaajlhoicikoedhcdcpaplhjfmnkdlm`	`https://extensions.nice-receiver-app.com/cxone-preview/updates.xml`
Chrome Web Store	`hmfcadoeekoabipnapelhpldlgioeaeb`	`https://clients2.google.com/service/update2/crx`

Block the channel you are not using, and block the public NICE store listing `gcfjbjldfomnopnpdjajjfpldkkdmmoi`.

Customers should therefore have administrative control over Chrome or Edge extension policies in the hosted browser environment.

The customer should be prepared to:

- Allow or force-install a NICE-provided browser extension by the extension ID for the chosen channel.
- Configure browser extension policy through Group Policy, MDM, registry policy, or another supported enterprise browser-management mechanism.
- Allow installation from the NICE enterprise extension host (`extensions.nice-receiver-app.com`) or the Chrome Web Store update service, depending on channel.
- Permit updates to the extension through the chosen deployment mechanism.
- Ensure security policy does not restrict the approved extension package or update location.
- Coordinate changes to an existing NICE browser extension when instructed as part of deployment.

> **Important:** Customers should not manually substitute, unpack, modify, or deploy alternate versions of the NICE browser extension unless specifically instructed.

Organizations that restrict Chrome or Edge extensions exclusively to public extension stores should identify this restriction before deployment. An enterprise-policy exception may be required for the enterprise host channel.

Detailed force-install JSON and managed-policy keys are in the Receiver App Deployment Guide, Citrix UCSDK Policy Guide, and Omnissa Horizon Policy Guide.

---

7. Endpoint Software Deployment

The NICE WebRTC Receiver must be installed on each physical endpoint that will use WebRTC Redirection.

For Windows, ChromeOS, and Dell ThinOS 10 packaging, policy, verification, and production checklists, see the Receiver App Deployment Guide.

NICE does not require a specific endpoint-management product.

Organizations may use their existing software-distribution infrastructure, including technologies such as:

- Group Policy or enterprise software deployment for Windows
- Microsoft Intune or equivalent endpoint-management products
- Dell Wyse Management Suite for ThinOS
- Chrome Enterprise management for ChromeOS
- Manual installation during pilot testing

For ThinOS 10+ environments, Dell Wyse Management Suite supports deployment of application packages to managed ThinOS devices.

Before beginning a production rollout, the customer should confirm:

- The organization has a method to deploy the Receiver to endpoints.
- The deployment team can update the Receiver when new versions are released.
- Endpoint security software permits the Receiver to execute.
- Required network access is available from the physical endpoint.
- Required microphone and audio permissions are available to the Receiver.

---

8. Secure Matchmaking Service

The browser extension and physical endpoint must locate and securely associate with one another before WebRTC redirection can occur.

This coordination is performed through the NICE WebRTC Redirection matchmaking service.

The matchmaking service:

- Uses encrypted HTTPS/WSS communications over TLS.
- Securely associates the hosted browser session with the appropriate physical endpoint.
- Exchanges control information required to establish WebRTC redirection.
- Maintains redirection/session state as required.
- Does **not** relay the WebRTC voice stream.

Both the hosted browser environment and physical endpoint require outbound HTTPS/WSS connectivity to the matchmaking service.

Matchmaking Network Requirement

Source	Destination	Protocol	Port	Purpose
Hosted browser environment	NICE Matchmaking Service	HTTPS/WSS	TCP 443	Registration, pairing, and redirection control
Physical endpoint	NICE Matchmaking Service	HTTPS/WSS	TCP 443	Registration, pairing, and redirection control

Matchmaking endpoint

Item	Value
FQDN	`liaison.nice-receiver-app.com`
URL	`wss://liaison.nice-receiver-app.com/ws`
IP address	`135.237.20.87`
Port	TCP 443

---

9. STUN Requirement

NICE WebRTC Redirection uses the NICE STUN service to assist with WebRTC ICE candidate discovery and NAT traversal.

The configured service is:

`stun:stun.nice-receiver-app.com:19302`

Both the **hosted browser** (NICE extension) and the **physical endpoint** must be able to communicate with this STUN service using UDP. Matchmaking exchanges ICE candidates gathered on both sides.

STUN endpoint

Item	Value
FQDN	`stun.nice-receiver-app.com`
Port	UDP 19302
DNS	CNAME to `stun.l.google.com` (Google public STUN infrastructure)
IP addresses	Resolved A/AAAA records for `stun.nice-receiver-app.com` (change over time; do not hard-code a single IP long-term)

Prefer FQDN-based allowlisting where the firewall or secure web gateway supports it. If IP allowlisting is required, resolve `stun.nice-receiver-app.com` and permit the returned A/AAAA addresses (re-check after DNS TTL expiry).

Source	Destination	Protocol	Destination Port	Purpose
Hosted browser	`stun.nice-receiver-app.com`	STUN/UDP	UDP 19302	NAT discovery and ICE candidate gathering
Physical endpoint	`stun.nice-receiver-app.com`	STUN/UDP	UDP 19302	NAT discovery and ICE candidate gathering

STUN does **not** carry call audio. It allows each side to discover network addressing used when establishing the peer data tunnel and the WebRTC voice path.

9.1 TURN Is Not Used as an Alternate Media Path

NICE WebRTC Redirection does not provide a TURN relay to work around networks that prevent direct WebRTC connectivity.

Customers should therefore configure the endpoint network to permit the required STUN and WebRTC communications.

This is intentional: the redirection solution does not introduce an additional NICE WebRTC Redirection media relay into the call path.

If STUN or the required direct WebRTC connectivity is unavailable, redirection may fail and the session will gracefully fall back to the normal hosted-desktop audio path.

> **Important:** An environment in which CXone calls work normally may still require firewall changes before endpoint WebRTC redirection will function.

---

10. NICE CXone Network Requirements

When WebRTC redirection is active, the physical endpoint establishes the WebRTC connection to NICE CXone.

The endpoint network must therefore allow the applicable NICE CXone Integrated Softphone/WebRTC connectivity.

NICE currently documents the following protocol and port requirements for Integrated Softphone/WebRTC:

Source	Destination	Protocol	Destination Port	Direction
Physical endpoint	NICE CXone	HTTPS	TCP 443	Outbound
Physical endpoint	NICE CXone WebRTC infrastructure	WebRTC media / UDP	UDP 1024–65535	Outbound

The exact NICE FQDNs and IP address ranges depend on the customer's CXone environment.

Customers should use the current **NICE CXone Connectivity Requirements** documentation to identify the requirements applicable to their CXone cluster and Integrated Softphone configuration.

Where NICE provides both FQDNs and IP address ranges, both should be permitted as directed by the NICE connectivity documentation.

This is particularly important for environments using specialized NICE deployments such as FedRAMP, where the required NICE WebRTC infrastructure may differ from the commercial environment.

> The physical endpoint must be able to reach the NICE WebRTC resources appropriate for the customer's own CXone environment.

Customers should not substitute generic NICE IP ranges from another environment.

---

11. Network Requirements Summary

At a minimum, the following communications are required for successful redirection.

Source	Destination (FQDN / IP)	Protocol	Port	Required For
Hosted browser	`liaison.nice-receiver-app.com` / `135.237.20.87`	HTTPS/WSS	TCP 443	Session coordination
Hosted browser	`stun.nice-receiver-app.com` (resolved A/AAAA)	UDP	19302	ICE/NAT discovery
Hosted browser (enterprise channel)	`extensions.nice-receiver-app.com` (Azure Static Web Apps; IPs change)	HTTPS	TCP 443	Extension package / updates
Admin / IT (optional)	`docs.nice-receiver-app.com` (Azure Static Web Apps; IPs change)	HTTPS	TCP 443	Product docs / Receiver downloads
Physical endpoint	`liaison.nice-receiver-app.com` / `135.237.20.87`	HTTPS/WSS	TCP 443	Session coordination
Physical endpoint	`stun.nice-receiver-app.com` (resolved A/AAAA)	UDP	19302	ICE/NAT discovery
Physical endpoint	NICE CXone (tenant connectivity docs)	HTTPS	TCP 443	WebRTC/CXone connectivity
Physical endpoint	NICE CXone WebRTC infrastructure (tenant connectivity docs)	UDP	1024–65535	Real-time voice media

NICE redirection service endpoints (summary)

Service	FQDN	IP address	Port
Matchmaking	`liaison.nice-receiver-app.com`	`135.237.20.87`	TCP 443 (`wss://…/ws`)
STUN	`stun.nice-receiver-app.com`	DNS A/AAAA (via CNAME to `stun.l.google.com`; rotates)	UDP 19302
Documentation / downloads	`docs.nice-receiver-app.com`	Azure Static Web Apps A records (rotate)	TCP 443
Enterprise extension host	`extensions.nice-receiver-app.com`	Azure Static Web Apps A records (rotate)	TCP 443

The hosted desktop environment must continue to meet the normal NICE CXone connectivity requirements for the applications running within that environment.

---

12. Firewalls, Proxies, and Network Security

Enterprise network-security products can affect WebRTC even when normal web browsing functions correctly.

Customers should review:

- Perimeter firewalls
- Endpoint firewalls
- Secure web gateways
- VPN products
- SASE products
- Explicit or transparent proxies
- TLS inspection
- DNS filtering
- Endpoint network-isolation policies
- EDR or application-control software
- UDP filtering and timeout policies

12.1 HTTPS Does Not Validate WebRTC

Successfully opening CXone in a browser only proves that basic web connectivity is available.

It does **not** demonstrate that:

- `stun.nice-receiver-app.com` is reachable over UDP 19302.
- ICE candidate gathering can complete successfully.
- The endpoint can reach NICE WebRTC media resources.
- UDP 1024–65535 is available to the applicable NICE infrastructure.
- A usable ICE candidate pair can be established.

These items must be validated separately.

12.2 TLS Inspection

The NICE matchmaking connection uses TLS-secured HTTPS/WSS.

TLS inspection or proxy technologies must not alter or block these connections in a way that prevents:

- Certificate validation
- HTTPS communication
- WebSocket upgrades
- Long-lived secure WebSocket sessions

If matchmaking connectivity fails in an environment using TLS inspection, the matchmaking FQDN should be tested without inspection as part of troubleshooting.

---

13. Competing WebRTC Redirection Technologies

Only one WebRTC redirection or optimization technology should control the CXone WebRTC session.

Other browser extensions, browser policies, multimedia-redirection technologies, or endpoint products that intercept WebRTC APIs may interfere with NICE WebRTC Redirection.

Before deployment, customers should identify any existing technology that:

- Intercepts `RTCPeerConnection`
- Modifies WebRTC behavior
- Redirects browser multimedia
- Redirects WebRTC media
- Replaces or virtualizes WebRTC APIs
- Provides browser-based voice optimization
- Alters media-device APIs

Any conflicting WebRTC redirection or optimization technology affecting the CXone application should be disabled or removed as directed during deployment.

> Multiple WebRTC interception mechanisms should not be enabled simultaneously for the CXone application.

---

14. Audio Device Requirements

When WebRTC Redirection is active, voice media and audio devices are handled by the physical endpoint.

Before validation:

- Connect the headset or other intended audio device to the physical endpoint.
- Verify that the physical endpoint recognizes the microphone.
- Verify that the physical endpoint recognizes the desired audio-output device.
- Confirm that operating-system privacy controls permit microphone access.
- Confirm that endpoint security software does not prevent the Receiver from accessing audio devices.
- Verify that another application has not exclusively locked the required audio device.

When troubleshooting a redirected call, audio-device troubleshooting should primarily be performed on the **physical endpoint**, not within the hosted desktop.

---

15. Pre-Deployment Readiness Checklist

Before scheduling validation, confirm the following:

- [ ] Endpoint operating system is supported.
- [ ] Chrome or Edge is available in the hosted environment.
- [ ] Administrators can deploy enterprise browser-extension policy.
- [ ] Administrators can permit a NICE-provided extension installation/update source if required.
- [ ] NICE WebRTC Receiver can be installed on the physical endpoint.
- [ ] TCP 443 access to the NICE matchmaking service is permitted from the hosted environment.
- [ ] TCP 443 access to the NICE matchmaking service is permitted from the endpoint.
- [ ] UDP 19302 access to `stun.nice-receiver-app.com` is permitted from the hosted browser and from the endpoint.
- [ ] Applicable NICE CXone FQDNs and IP ranges have been identified.
- [ ] NICE Integrated Softphone/WebRTC TCP 443 connectivity is permitted from the endpoint.
- [ ] NICE Integrated Softphone/WebRTC UDP 1024–65535 connectivity is permitted from the endpoint.
- [ ] Proxy and TLS-inspection requirements have been reviewed.
- [ ] Other WebRTC/media-redirection technologies have been identified.
- [ ] Conflicting WebRTC redirection technologies have been disabled where required.
- [ ] Endpoint microphone access has been verified.
- [ ] Endpoint audio output has been verified.
- [ ] A representative user and physical endpoint are available for validation.
- [ ] The customer can place a test CXone call.

---

16. Validation Procedure

Validation should be performed in layers.

The objective is to determine exactly where redirection succeeds or fails rather than relying solely on whether a call can be completed.

16.1 Step 1 — Verify Browser Extension Installation

In Chrome:

1. Open `chrome://extensions`.
2. Locate the NICE WebRTC browser extension.
3. Verify that the extension is enabled.
4. Verify the expected version.
5. Confirm that Chrome does not report an extension-policy or installation error.

In Microsoft Edge:

1. Open `edge://extensions`.
2. Locate the NICE WebRTC browser extension.
3. Verify that the extension is enabled.
4. Verify the expected version.
5. Confirm that Edge does not report an extension-policy or installation error.

If the extension is enterprise-managed, browser policy can also be reviewed using:

- `chrome://policy`
- `edge://policy`

Confirm that the expected extension policy has been successfully applied.

Expected result: NICE browser extension is installed and enabled.

---

17. Step 2 — Verify Receiver Health

Start or inspect the NICE WebRTC Receiver on the physical endpoint.

Verify the status information made available by the Receiver, including where applicable:

- Receiver version
- Receiver running state
- Operating-system information
- Network status
- Matchmaking service connectivity
- Pairing/session state
- Audio-device availability
- Most recent error or warning

Expected result: Receiver is running and reports that it is available for redirection.

If the Receiver cannot connect to the matchmaking service, test:

1. DNS resolution for the matchmaking FQDN.
2. HTTPS connectivity to the matchmaking service.
3. TCP 443 from the endpoint.
4. Proxy configuration.
5. TLS inspection.
6. Endpoint firewall rules.
7. EDR/application-control logs.

---

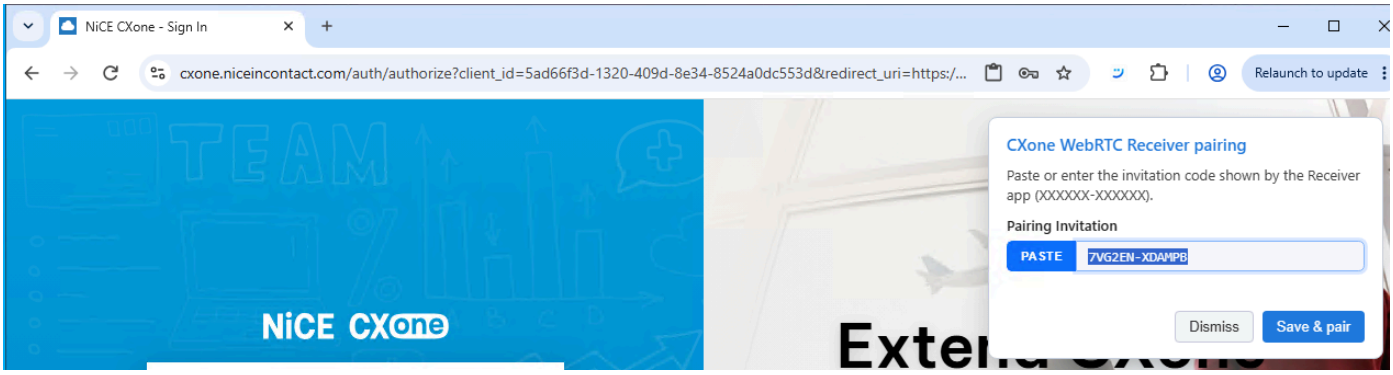
18. Step 3 — Verify Matchmaking Connectivity

Both the browser extension and Receiver must connect to the secure matchmaking service.

Where diagnostic status is available, confirm:

- Browser extension connected
- Receiver connected
- Browser/endpoint association established
- No matchmaking authentication or connection errors are present

Expected result: Browser environment and endpoint are successfully associated.



A failure at this stage is a **control-path problem**, not a NICE WebRTC media problem.

Review:

- TCP 443
- DNS
- HTTPS/WSS access
- TLS inspection
- Proxy behavior
- Session/pairing information
- Browser-extension logs
- Receiver logs

---

19. Step 4 — Validate STUN Connectivity

STUN connectivity should be tested from the **physical endpoint** and, where practical, confirmed as reachable from the **hosted browser** network as well. Both sides gather ICE candidates during matchmaking.

The endpoint must be capable of sending UDP traffic to:

`stun.nice-receiver-app.com:19302`

Where a Receiver network diagnostic is available, run the STUN or WebRTC readiness test.

Useful diagnostic results include:

- DNS resolution succeeded
- STUN request sent
- STUN response received
- ICE gathering started
- Host candidate discovered
- Server-reflexive candidate discovered
- ICE gathering completed

Expected result: A STUN response is received and usable ICE candidates can be generated.

If STUN fails:

1. Verify DNS resolution for `stun.nice-receiver-app.com`.
2. Verify outbound UDP destination port 19302.
3. Review endpoint firewall rules.
4. Review network firewall rules.
5. Review VPN/SASE security policy.
6. Confirm UDP traffic is not being forced through an incompatible proxy.
7. Repeat the test from the same network the production agents will use.

A successful TCP/HTTPS test is **not** a substitute for this test.

If STUN remains unavailable, NICE WebRTC Redirection may be unable to establish a direct media path and calls can fall back to hosted-desktop audio.

---

20. Step 5 — Validate NICE WebRTC Connectivity

Review the customer's NICE CXone Connectivity Requirements.

Confirm the endpoint has access to the Integrated Softphone/WebRTC resources specified for the customer's CXone environment.

Verify:

- Required FQDNs resolve.
- Required NICE IP ranges are permitted.
- TCP 443 is allowed.
- UDP 1024–65535 is allowed to the applicable NICE WebRTC infrastructure.

Where both FQDNs and IP addresses are provided by NICE, configure both as required by the NICE documentation.

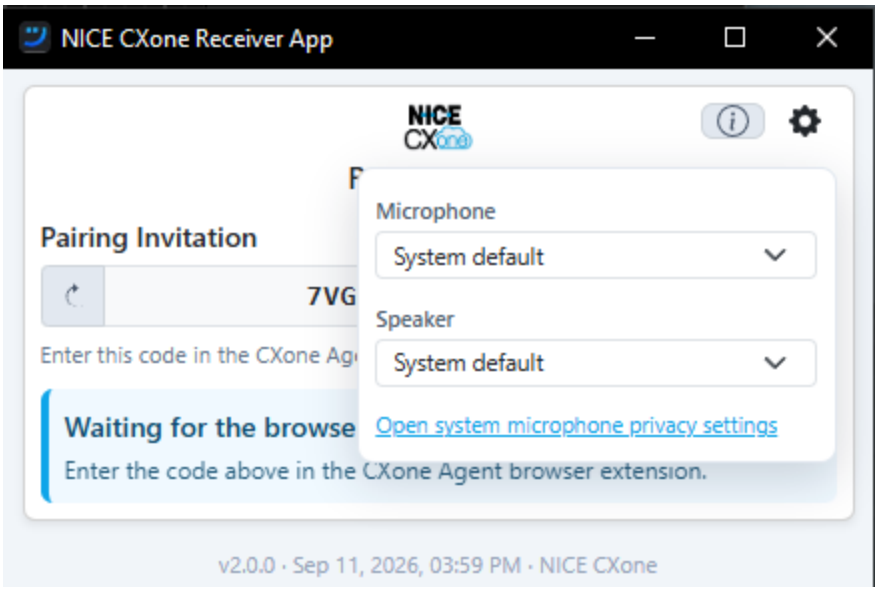
Expected result: The physical endpoint can reach the NICE WebRTC environment required by the customer's CXone configuration.

---

21. Step 6 — Verify Audio Devices

On the physical endpoint:

1. Confirm the desired headset is connected.
2. Verify the microphone appears in the operating system.
3. Verify the playback device appears in the operating system.
4. Confirm microphone permissions.
5. If available, use the Receiver's microphone test or level indicator.
6. If available, use the Receiver's speaker/audio-output test.



Expected result: Required input and output devices are available to the Receiver.

---

22. Step 7 — Open CXone Agent

Open CXone Agent using the normal customer workflow.

Where extension diagnostic information is available, verify that the extension reports:

- CXone application detected

- WebRTC redirection support active
- Receiver available
- Endpoint associated
- Ready for a WebRTC session

Expected result: The browser extension is ready to redirect a CXone WebRTC session.

---

23. Step 8 — Place a Test Call

Place a test call using the customer's normal CXone workflow.

Do **not** use call completion alone as the success criterion.

Observe the NICE WebRTC Redirection browser-extension indicator.

Successful Redirection

When WebRTC is successfully redirected:

- The Receiver should show an active WebRTC session where status information is available.
- The browser extension should indicate successful redirection.
- Audio should use the physical endpoint's devices.
- WebRTC media should originate from the endpoint.

Fallback

If redirection cannot be established:

- The call may still succeed.
- Audio will use the normal hosted-desktop audio path.
- The browser extension will display the **unhappy phone** indicator.

The unhappy-phone indicator should therefore be treated as an indication that the call is operating in fallback mode rather than proof of a general CXone calling failure.

---

24. Step 9 — Review WebRTC Diagnostics

Where available, collect WebRTC session information from the Receiver.

Useful diagnostic fields include:

Connection State

- Peer connection state
- ICE gathering state
- ICE connection state
- Signaling state

ICE Information

- Number of candidates
- Candidate types
- Local candidate
- Remote candidate
- Selected candidate pair
- Transport protocol
- Local address
- Remote address

Media Information

- Audio track state
- Packets sent
- Packets received
- Bytes sent
- Bytes received
- Round-trip time
- Jitter
- Packet loss
- Codec
- Bitrate

A healthy redirected session should show an established ICE connection and active media transport between the physical endpoint and the NICE WebRTC infrastructure.

---

25. Understanding Validation Results

The environment can generally be placed into one of three states:

State	CXone Call	Endpoint Redirection	Meaning
Redirected	Working	Active	Desired operating state
Fallback	Working	Inactive	Call functional, redirection requires investigation

State	CXone Call	Endpoint Redirection	Meaning
Call Failure	Not working	N/A or unknown	General CXone/network/audio troubleshooting required

Redirected

No unhappy-phone indication is present and endpoint WebRTC diagnostics confirm an active redirected session.

Fallback

The CXone call works, but the browser extension displays the unhappy-phone indicator.

Investigate:

- Receiver availability
- Matchmaking connectivity
- NICE STUN connectivity
- ICE candidate gathering
- NICE WebRTC UDP connectivity
- Firewall restrictions
- Competing redirection technologies

Call Failure

The call itself does not function.

This may indicate a general CXone, Integrated Softphone, network, authentication, browser, or audio problem independent of WebRTC Redirection.

---

26. Diagnostic Information

When requesting assistance with a redirection issue, collect as much of the following information as possible.

Environment

- Date and time of test
- Endpoint operating system and version

- Browser and browser version
- NICE browser-extension version
- NICE WebRTC Receiver version
- Whether the endpoint is connected through VPN
- Whether SASE/security filtering is in use
- Whether TLS inspection is enabled
- Whether the test occurred on the production agent network

Redirection Status

- Browser-extension status
- Whether the unhappy-phone icon appeared
- Receiver status
- Matchmaking connection status
- Pairing/session status
- Last reported error
- Session or correlation identifier, if available

Network Diagnostics

- Matchmaking DNS result
- Matchmaking HTTPS/WSS status
- STUN DNS result (`stun.nice-receiver-app.com`)
- STUN response status
- ICE gathering state
- ICE connection state
- Candidate types
- Selected candidate pair
- Protocol used
- NICE WebRTC destination
- Packet loss
- Jitter
- Round-trip time

Audio

- Selected microphone
- Selected output device
- Microphone permission state
- Whether microphone input is detected locally
- Whether audio output works locally

Logs

Where available, collect:

- Browser-extension diagnostic logs
- Receiver logs
- Browser policy information
- Network/security product logs
- WebRTC session statistics

When both browser and Receiver logs contain a common session or correlation identifier, include that identifier with the support request.

---

27. Diagnostic Interpretation

Common failure patterns can often identify the affected layer.

Observation	Likely Area
Extension not installed	Browser deployment/policy
Extension installed but Receiver unavailable	Matchmaking, Receiver, or pairing
Both components connected but STUN fails	Endpoint firewall/network UDP restrictions
STUN succeeds but ICE fails	NICE media access, NAT, firewall, or WebRTC path
ICE succeeds but no microphone	Endpoint device/privacy permissions
Call works with unhappy phone	Redirection failed; hosted-audio fallback active
Call works and redirection status is healthy	Normal redirected operation

---

28. Recommended Pilot Validation

Before broad deployment, NICE recommends validating WebRTC Redirection using representative production conditions.

The pilot should include:

- A supported endpoint
- The customer's normal endpoint network
- The customer's normal hosted browser environment
- Production-equivalent firewall and security policies

- A representative agent
- The headset model expected in production
- The customer's actual CXone environment
- At least one inbound or outbound test call

Where agents commonly use multiple network configurations, such as office, home, VPN, or branch networks, each materially different network path should be validated.

---

29. Production Rollout Considerations

After successful pilot validation:

1. Define the endpoint Receiver deployment method.
2. Define the enterprise browser-extension deployment policy.
3. Validate software deployment with a small pilot group.
4. Confirm network requirements across all affected endpoint networks.
5. Document the expected browser-extension status indicators.
6. Provide support staff with the validation and diagnostic procedure in this guide.
7. Expand deployment in controlled groups.
8. Monitor redirection success and fallback conditions during rollout.

Because fallback allows calls to continue, organizations should not rely exclusively on call-failure reports to identify redirection problems.

The browser-extension redirection indicator and available Receiver diagnostics should be included in production support procedures.

---

30. Customer Readiness Summary

A customer is ready to validate NICE WebRTC Redirection when:

- A supported endpoint operating system is available.
- Chrome or Edge is available in the hosted browser environment.
- The organization can deploy the NICE browser extension through enterprise policy.
- The NICE WebRTC Receiver can be deployed to the endpoint.
- Both components can securely reach the NICE matchmaking service over TCP 443.
- The hosted browser and the physical endpoint can reach `stun.nice-receiver-app.com` over UDP 19302.
- The physical endpoint meets the NICE Integrated Softphone/WebRTC network requirements.
- No competing WebRTC redirection technology is controlling the CXone WebRTC session.
- Local microphone and output devices are accessible.

- The organization can distinguish successful endpoint redirection from hosted-audio fallback.

Following these prerequisites before rollout significantly simplifies validation and makes it possible to isolate browser, endpoint, network, WebRTC, and audio issues independently.

---

31. Q&A

Business leaders

How do we know redirection is actually working?
A successful CXone call alone is not proof. Validation must show the hosted session is associated with the endpoint and that WebRTC media originates on the physical endpoint (not hosted-desktop audio fallback). Use the layered validation procedure in this guide.

What is the business risk if firewall changes slip?
Calls may fall back to normal hosted-desktop audio. Agents can still take calls, but latency and quality benefits of redirection will not appear until STUN, matchmaking, and CXone media paths are open from endpoint networks.

Is matchmaking in the call path for compliance or recording?
No. Matchmaking does not carry voice media. Screen recording of the agent UI remains on the hosted desktop for Citrix UCSDK and Horizon paths; voice devices and RTP are on the endpoint when redirection is active.

IT administrators

Why can CXone work while redirection fails?
Hosted-desktop Softphone/WebRTC may still succeed when endpoint STUN, matchmaking, or media UDP is blocked. Treat that as fallback, not a pass for redirection readiness.

Do we need both FQDNs and IPs on the firewall?
Publish both where you have them. Matchmaking has a stable IP (`135.237.20.87`). STUN and the enterprise extension host resolve to rotating addresses — prefer FQDN-based allowlisting and re-resolve if you must use IPs.

Which STUN host do we allow?
`stun.nice-receiver-app.com` on UDP 19302, from both the hosted browser and the physical endpoint.

Can TLS inspection stay enabled on matchmaking?
Only if it does not break certificate validation, WebSocket upgrade, or long-lived WSS. If pairing fails under inspection, exclude `liaison.nice-receiver-app.com` and retest.

VDI administrators

Does validation differ for Citrix or Horizon vs the Receiver App?
Network and STUN checks still apply for ICE/media. Connection-style proof differs: Receiver shows pairing Connected; Citrix expects toolbar `C` and `HdxRtcEngine` on the endpoint; Horizon expects toolbar `H`. See the Citrix and Omnissa policy guides for path-specific checks.

Can we validate from a lab VLAN only?
Pilot on a lab segment is fine for tooling, but production readiness requires the same firewall, proxy, VPN/SASE, and DNS policies agents use in production.

What if another multimedia redirection product is already enabled?
Only one WebRTC redirection or optimization technology may own the CXone session. Disable competing interception before validating NICE redirection.

Network administrators

What are the concrete NICE redirection endpoints?

Service	FQDN	IP	Port
Matchmaking	`liaison.nice-receiver-app.com`	`135.237.20.87`	TCP 443
STUN	`stun.nice-receiver-app.com`	Resolved A/AAAA (rotate)	UDP 19302
Documentation / downloads	`docs.nice-receiver-app.com`	Azure SWA A records (rotate)	TCP 443
Enterprise extension host	`extensions.nice-receiver-app.com`	Azure SWA A records (rotate)	TCP 443

CXone media FQDNs and IP ranges come from this tenant's NICE CXone Connectivity Requirements.

Is there a TURN relay if UDP is blocked?
No. Direct WebRTC UDP (and STUN) must be allowed; otherwise expect fallback to hosted-desktop audio.

Which direction is traffic?
Outbound from hosted browser and physical endpoint to matchmaking and STUN; outbound from the physical endpoint to CXone HTTPS and WebRTC UDP. Voice does not hairpin through the VDI host when redirection is active.

UDP 1024–65535 to CXone looks broad — can we narrow it?
Use the destination FQDNs/IP ranges in your tenant's NICE connectivity documentation. The port range is the Integrated Softphone/WebRTC media requirement; do not invent a smaller range without NICE guidance for your cluster.