

Campaign CNP Migration Tool

Product Overview

Updated April 5, 2024

CSPs collectively have expressed a strong desire for better connectivity migration, hereinafter referred to as CNP migration. In the past there was no self-serve migration tool; we are now introducing this feature in an effort to help with business continuity and efficiency.

A new CNP migration tool will be made available for CSPs and CNPs to switch connectivity partners at scale with minimal service disruption.



BETTER MESSAGING

campaign registry

Table of Contents

| | |
|------------------------------------------------------------|-----------|
| Table of Contents..... | 2 |
| Summary..... | 3 |
| CNP Migration Transaction..... | 3 |
| Auto-Acceptance on Prior Campaign Acceptance..... | 6 |
| Integrating with CNP Migration Tools..... | 6 |
| Campaign CNP Migration Examples..... | 7 |
| FAQs..... | 10 |
| Appendix A: New Properties..... | 15 |
| Appendix B: New Webhook Events..... | 16 |
| Appendix C: Updated Existing Webhook Events..... | 19 |
| Appendix D: New and Impacted CSP-API Endpoints..... | 22 |
| Appendix E: New and Impacted DCA-API Endpoints..... | 26 |

Summary

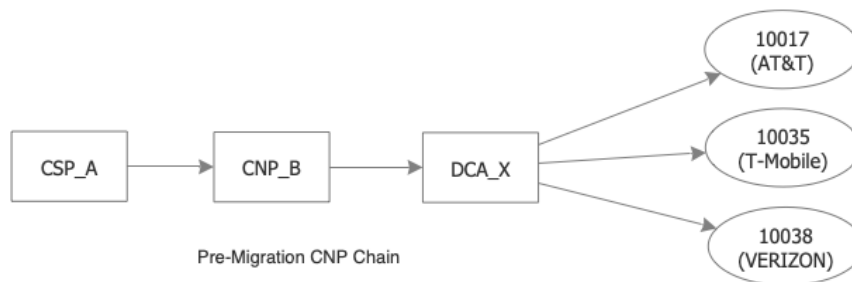
TCR has identified the following migration scenarios to be supported by this new tool:

1. First hop or CSP-initiated migration.
2. Middle hop or CNP-initiated migration.

In order to be migrated the campaign must meet the following criteria:

- At least one DCA has accepted the campaign.
- The campaign is in TCR Status: Active.
- The campaign is not currently Suspended by either an MNO or DCA.

Given an active campaign CNP chain $CSP_A \rightarrow CNP_B \rightarrow DCA_X$:



The CNP migration tool can support the following business cases:

1. CSP_A wants to use a different connectivity partner instead of CNP_B .
 - a. CSP_A can go directly to DCA_X . *E.g.* $CSP_A \rightarrow DCA_X$.
 - b. CSP_A can choose another CNP who is a DCA. *E.g.* $CSP_A \rightarrow DCA_Y$.
 - c. CSP_A can choose another CNP who isn't a DCA. *E.g.* $CSP_A \rightarrow CNP_C$.
2. CNP_B wants to use a different connectivity partner instead of DCA_X .
 - a. Choose another CNP who is a DCA. *E.g.* $CNP_B \rightarrow DCA_Y$.
 - b. While less likely, CNP_B can choose another CNP who isn't a DCA. *E.g.* $CNP_B \rightarrow CNP_C$.

CNP Migration Transaction

A CNP migration transaction captures a sequence of campaign-sharing actions to form a new CNP chain that ultimately replaces the current CNP chain. A CNP migration transaction can be initiated by either an originating CSP or a middle-hop CNP. The campaign must have at least one accepted DCA in order to qualify for migration.



There can only be a single open transaction for a given campaign at any time. Attempts to open a second concurrent CNP migration transaction by anyone in the CNP chain will be rejected by the system.

A CNP migration transaction will remain open until the transaction is completed or is canceled.

Transaction Complete Conditions

A CNP migration transaction is marked as complete when either one of the following conditions has been reached:

1. The campaign is accepted by existing CNP in the pre-migration CNP chain. Examples below under *Existing CNP Match*.
2. All requested MNO Networks have a corresponding DCA acceptance (this rule only applies for campaigns with a CAMPAIGN_DCA_COMPLETE status in the pre-migration CNP chain).
3. The campaign is accepted by the same Primary DCA in the pre-migration CNP chain.
4. The campaign is accepted by a Primary DCA (this rule only applies for campaigns without a CAMPAIGN_DCA_COMPLETE status in the pre-migration CNP chain).

TCR will notify all active participants in the migration with the `CNP_MIGRATION_COMPLETE` event. Former CNPs, who are no longer CNP in the post-migration chain will be notified with the `CNP_MIGRATION_PORT_OUT` event.

Active participants of CNP migration include migration initiators and all recipients of campaign-sharing events.

Existing CNP Match

When a CNP accepts a migration-sharing request, TCR will check if the accepting party is an existing CNP in the pre-migration CNP chain. If the accepting party is sharing to the same upstream, TCR will preserve the continuity of the existing upstream CNP chain to be carried over into the post-migration CNP chain. Business rules for detecting a match are detailed below for CNP and DCA:

Existing CNP Match Rules

- A CNP accepting the campaign sharing request is also a CNP in the pre-migration CNP chain.
- The upstream CNP from the migration campaign sharing request matches that of the campaign sharing request in the pre-migration CNP chain.

Let's illustrate the concept with the following scenario:

- Pre-migration CNP chain: `CSP_A → CNP_B → CNP_C → DCA_X → DCA_Y`



- Provisional CNP chain***: CSP_A → CNP_C → DCA_X
- Post-migration CNP chain: CSP_A → CNP_C → DCA_X → DCA_Y

When CNP_C accepts the CNP migration campaign sharing request from CSP_A, TCR recognizes that CNP_C is already a CNP with the same DCA_X as the upstream CNP. Consequently, the entirety of the upstream CNP chain from CNP_C onward is preserved and carried over into the post-migration CNP chain.

*** "Provisional CNP Chain" is the proposed new CNP chain that is waiting to be accepted. Once the "Provisional CNP Chain" is accepted, it becomes the "Post-migration CNP chain".

Existing DCA Match Rules

- DCA must be in the primary DCA role in both the Provisional CNP chain and pre-migration CNP chain.

Let's illustrate the concept with the following scenario:

- Pre-migration CNP chain: CSP_A → CNP_B → DCA_X → DCA_Y
- Provisional CNP chain: CSP_A → CNP_C → DCA_X
- Post-migration CNP chain: CSP_A → CNP_C → DCA_X → DCA_Y

When DCA_X accepts the CNP migration campaign sharing request from CNP_C, we recognize that DCA_X is also in the identical position as the primary DCA in the pre-migration CNP chain. Consequently, the entirety of the upstream CNP chain from DCA_X onward is preserved and carried over into the post-migration CNP chain.

DCA Complete

In a non-CNP migration scenario, TCR notifies the CNPs of a `CAMPAIGN_DCA_COMPLETE` event once the campaign reaches and is accepted by DCA(s) for requested MNO networks. The same principle applies to the CNP migration scenario, but a new webhook event:

`CNP_MIGRATION_COMPLETE` is produced.

Transaction Cancellation

A CNP migration transaction can be canceled in 4 ways:

1. By the migration initiator
2. When the migration transaction fails to complete within the 30 calendar day window
3. Campaign is deactivated or expired
4. Campaign is suspended by MNO or DCA



Once the migration transaction is completed, the transaction cannot be canceled. When the CNP migration transaction is canceled, a `CNP_MIGRATION_CANCEL` webhook event is generated and sent to all participants identified in the Provisional CNP migration chain.

Auto-Acceptance on Prior Campaign Acceptance

This is an optional setting that enables automatic acceptance of campaign sharing requests by the upstream CNP or primary DCA if and only if the upstream accepting party is a participant of the same role in the pre-migration CNP chain. Auto-Acceptance is one setting that can potentially have two effects (auto accept and auto complete). Let's illustrate the concept with the following scenario:

- Pre-migration CNP chain: `CSP_A → CNP_B → CNP_C → CNP_D → DCA_X → DCA_Y`
- Provisional CNP chain: `CSP_A → CNP_C → CNP_D`
 - `CNP_C` has turned Auto-Acceptance feature on, so does not need to select `CNP_D`, it is automatically shared on their behalf.
- Post-migration CNP chain: `CSP_A → CNP_C → CNP_D → DCA_X → DCA_Y`
 - `CNP_C → CNP_D` creates a CNP Match so Auto-Complete kicks in, no further manual sharing to `DCA_X` is required, it is automatically shared on their behalf.

Note in this example, `CNP_C` is a participating CNP in the pre-migration CNP chain. This implies `CNP_C` has already reviewed and accepted the campaign. If `CNP_C` chooses to enable the auto-acceptance on prior acceptance feature, then the system will automatically accept (on `CNP_C`'s behalf) the campaign sharing request from `CSP_A → CNP_C`, and automatically pick `CNP_D` as the upstream (the upstream of `CNP_C` in pre-migration CNP chain).

The auto-acceptance option is disabled by default for both CSP and DCA. This feature can be enabled in the account settings.

Integrating with CNP Migration Tools

CNP migration can be initiated by anyone in the CNP chain including the originating CSP and the middle hop CNP. The stakeholders or participants of the migration can use either the API or portal (once available) to perform tasks based on delegated roles.

While idealizing the CSP and CNP user experience for connectivity migration, we recognize the process of building and staging the CNP chain is nearly identical to the process of building the CNP chain for a new campaign, sharing the following key activities:

- CSP/CNP sharing the campaign with an upstream CNP.
- CNP/DCA accepting or rejecting a campaign shared by downstream CNP.



Given the similarity between the two journeys, we believe our customer is best served if they can continue to follow existing practice for migrating campaign connectivity. In terms of API user experience, CSPs and DCAs will interface with mostly existing API endpoints and webhook events for connectivity migration engagement.

With that said, some API changes may require CSPs and DCAs to update their API integrations to fully support CNP migration. Whether you need to update your API integration depends on your intent to participate in future CNP migration initiated by you or others in the ecosystem. We believe your answers to the following 2 questions can help provide some clarity:

1. Do you expect to initiate a CNP migration for either your campaign or your partner's campaign in the future?
If the answer is Yes, then you will either use the CSP portal (once available) or CSP-API to initiate a CNP migration. If you are exclusively a CSP-API user, then you must bring your API integration up to date. More details are in the CSP-API Endpoints section.
2. Do you expect to participate as a CNP or primary DCA so others can migrate traffic to you from another CNP?
If the answer is Yes, then you will either use the CSP/DCA portal (once available) or CSP/DCA-API to accept a shared campaign associated with a CNP migration. If you are exclusively an API user, then you will need to review your existing API integration to determine if you need to bring your API integration up to date.

In anticipation that some CSPs and DCAs may not be ready to accept CNP migration sharing requests, the CSPs and DCAs can choose to opt out from receiving these campaign sharing requests until they are ready. The optionality can be managed under account settings.

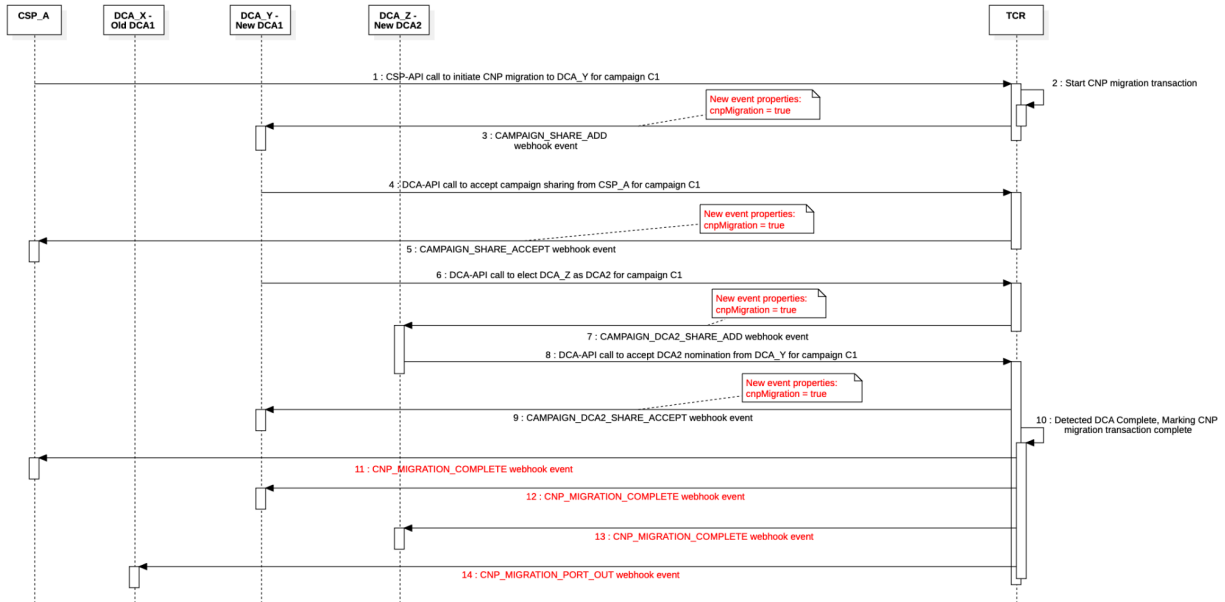
Campaign CNP Migration Examples

Example 1 - CSP Initiated Migration

In this example, the CSP_A initiates a migration to switch the upstream connectivity partner from DCA_X to DCA_Y.

- Pre-migration CNP chain: CSP_A → DCA_X (DCA1)
- Post-migration CNP chain: CSP_A → DCA_Y (DCA1) → DCA_Z (DCA2)

In this particular scenario, DCA_Y relies on DCA2 (DCA_Y) to reach the remaining MNO networks. The CNP migration transaction is completed when DCA_Z accepts the DCA2 role to complete MNO network coverage.



Example 2 - CNP Initiated Migration

In this example, CNP_B initiates the migration to switch upstream connectivity partners from CNP_C to DCA_Y.

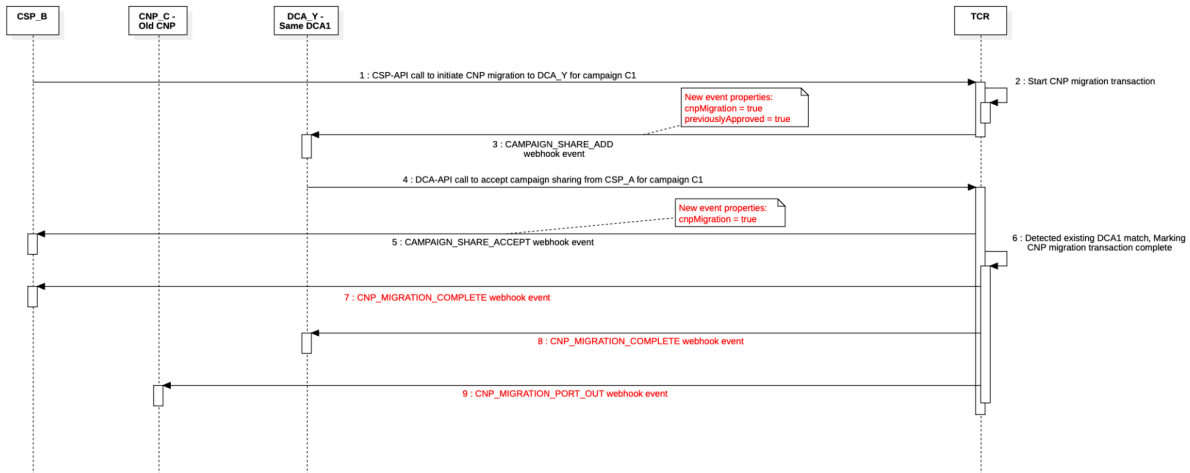
- Pre-migration CNP chain: CSP_A → CNP_B → CNP_C → DCA_Y (DCA1) → DCA_Z (DCA2)
- Post-migration CNP chain: **CSP_A** → CNP_B → DCA_Y (DCA1) → **DCA_Z (DCA2)**

CSP_A (in bolded text in the Post-migration CNP chain) is unaware of the CNP migration because the CNP migration occurs upstream of CSP_A.

DCA_Z (in bolded text in the Post-migration CNP chain) is not a participant in the CNP migration because 1) There is no change to its DCA2 role, and 2) There is no change to its downstream connectivity partner, DCA_Y.

Example 2.1 Auto-Accept on Prior Acceptance Option Disabled

The Auto-Accept on Prior Acceptance option is disabled by default for all CSP and DCA accounts. This feature comes into play for DCA_Y, who is a participant in the pre-migration CNP chain. Under the circumstances, DCA_Y must take action to accept the campaign sharing CNP_B to further the migration process.

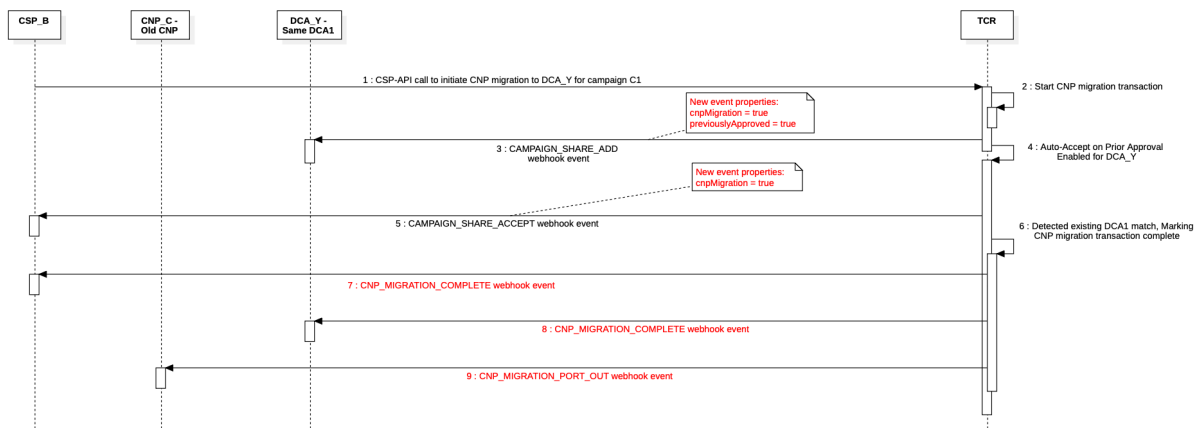


The CAMPAIGN_SHARE_ADD webhook event (line 3 above) sent to DCA_Y includes the previouslyAccepted = true property/value as a reminder that DCA_Y already reviewed and accepted its role for this campaign.

***NOTE:** The flow charts display a “previouslyApproved” attribute which has been replaced with “previouslyAccepted”. Please use “previouslyAccepted” in your implementation.

Example 2.2 Auto-Accept on Prior Acceptance Option Enabled

This example illustrates the effect of Auto-Accept on Prior Acceptance option when enabled for a DCA who has already reviewed and accepted the campaign. Presume DCA_Y opted to enable the Auto-Accept on Prior Acceptance option, then DCA_Y will not need to take an action to accept the campaign sharing from CNP_B.



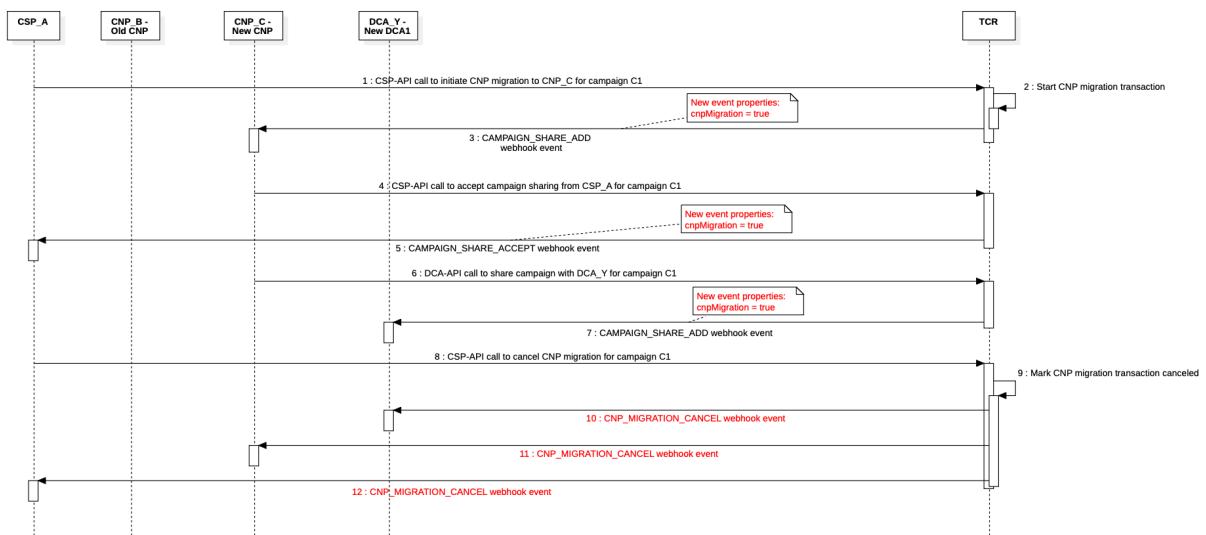
***NOTE:** The flow charts display a “previouslyApproved” attribute which has been replaced with “previouslyAccepted”. Please use “previouslyAccepted” in your implementation.

Example 3 - CNP Migration Cancellation

In this example, CNP_A initiates the migration to switch upstream connectivity partners from CNP_B to CNP_C, but mid-way through the migration CNP_A decides to cancel the migration. As a result of the cancellation, all active participants of the migration are notified of the cancellation event.

- Pre-migration CNP chain: CSP_A → CNP_B → DCA_X
- Provisional migration CNP chain: CSP_A → CNP_C → DCA_Y (pending review)

CSP_A, CNP_C, and DCA_Y are all considered active migration participants.



CNP migration cancellation can also be triggered by the system when a CNP migration takes longer than 30 calendar days.

FAQs

Does this tool support the migration of the campaign from one CSP to another CSP?

No. This tool does not allow you to change ownership of the campaign from one CSP to another CSP. This tool is designed to support changes to your upstream connectivity partner.



BETTER MESSAGING

campaign registry

Does this tool support the migration of a brand from one CSP to another CSP?

No. This tool does not allow you to change ownership of the brand from one CSP to another CSP.

I want to migrate CNP for 100 campaigns, does this tool allow batch migration?

No. Alternatively, you can write a script to invoke the CSP-API endpoint `POST /campaign/{campaignId}/migrateCnp/{upstreamCnpId}` iteratively for the campaigns to be migrated.

When will the CNP migration tool be available in the CSP portal?

To be announced.

Will my traffic be impacted by CNP migration?

There are two sides to a CNP migration: the registry side and the traffic side. TCR will maintain the existing/active CNP chain while building out the new CNP chain. The new CNP chain replaces the existing CNP chain when the CNP migration is completed. TCR responsibility ends once the migration participants are notified of the CNP migration completion event. It is conceivable that traffic may be impacted while the participating CNPs and DCAs provision routes or move TNs for the campaign. It is recommended to contact your upstream CNP to learn more about potential downtime if any.

Can a campaign be migrated if currently under suspension by one or more MNOs?

No. CNP migration requests will be declined by TCR until suspension(s) are lifted.

Can a campaign be deactivated or expired if the campaign is the subject of a CNP migration?

Yes. `CNP_MIGRATION_CANCEL` will be generated for the participants of the CNP migration.



BETTER MESSAGING

campaign registry

There is a lot to digest with all of the API changes. What is the impact on me as a CSP if I have no immediate plan to change my upstream connectivity partner?

The new API endpoints for initiating and cancellation CNP migrations do not apply to you if you have no plan to migrate upstream connectivity partners. There should be no behavior change on other API endpoints that would impact your existing API integration. The new webhook event `CNP_MIGRATION_COMPLETE` may be materially important as it signifies the completion of a CNP migration.

If you are an upstream connectivity partner for others and have a desire to do so for CNP migration, then you should review your API and webhook integrations for a CNP migration scenario. If your system is not ready to accept CNP migration (as an upstream CNP), then you can exclude yourself from participating in CNP migration.

How long does it take to complete a CNP migration for one campaign?

The key factor is the number of new participants introduced in the new CNP chain. Every new active participant must review and accept their role as a CNP or DCA. More participants means potentially more waiting time.

Conversely, the turnaround time can be short in a scenario where the migration involves bypassing a middle hop while maintaining the same remaining CNPs/DCAs.

I initiated a CNP migration, how do I track the progress?

If you are the originating CSP, you can issue an API call to `GET /campaign/{campaignId}/sharing` or use the `GET /partnerCampaign/{campaignId}/sharing` if you are a CNP. These API calls will tell you if your upstream CNP has accepted or rejected the shared campaign. Your visibility is limited to the adjacent upstream hop. You may also receive progress updates by webhook events including `CAMPAIGN_SHARE_ACCEPT`, `CAMPAIGN_SHARE_DELETE`, and `CNP_MIGRATION_COMPLETE` throughout the migration journey.

Will I be notified of a `CAMPAIGN_DCA_COMPLETE` event from a CNP migration that is deemed DCA complete?

When TCR detects a DCA complete condition from a CNP migration, the CNP migration transaction is marked as complete; hence, triggering the `CNP_MIGRATION_COMPLETE` event to be generated and sent to all active CNP migration participants. **If pre-migration, the**



BETTER MESSAGING

campaign registry

campaign was not `CAMPAIGN_DCA_COMPLETE`, then a new `CAMPAIGN_DCA_COMPLETE` event will be sent out. If pre-migration, the campaign was already in a `CAMPAIGN_DCA_COMPLETE` state, then a new `CAMPAIGN_DCA_COMPLETE` event will **not** be sent out

I am a DCA, do I need to update my DCA-API or Webhook Integration?

A DCA can participate in a CNP migration as either a DCA1 or DCA2 role to the CNP chain. There are two new properties: `cnpMigration` and `previouslyAccepted` introduced in the `CAMPAIGN_SHARE_ADD` and `CAMPAIGN_DCA2_SHARE_ADD` webhook events to identify the migration context. We urge you to review the migration [examples](#) to assess impact to your current workflow for handling campaigns shared by your downstream customers.

As a DCA1, can I use this tool to initiate CNP migration?

If you are a DCA1 (aka primary DCA), you should use the DCA-API `PUT /campaign/{campaignId}/dca/{dcaId}` to change DCA2.

Will TCR maintain historical records of CNP changes resulting from CNP migrations?

Yes, TCR will maintain audit trails of CNP changes. We expect to make necessary enhancements to API and Portal allowing this data to be consumed by key stakeholders of the ecosystem.

Can the Auto-Accept on Prior Acceptance option under my account be set by downstream CNP ID?

No. Auto-Accept on Prior Acceptance option is a global setting. When enabled, it will require no action to accept a campaign sharing for a CNP migration campaign that you have previously reviewed and accepted for your role in the CNP chain. If disabled, you will need to take an action to either accept or reject a CNP migration campaign shared with you.

What will happen to the existing CNP to DCA migration endpoint?

`DELETE /campaignElectedDCA/campaign/{campaignId}`

This endpoint was our first available migration feature. The endpoint will be deprecated and removed at some point in 2024. Exact date will be announced on a biweekly Release Notes communication.



If a CNP is not integrated with the tools, and is shared a migration campaign by a downstream partner, what is the experience for both parties?

Assuming CNP_A opts out of migration participation, any sharing request (in a migration context) with upstream CNP_A will be rejected by the system with an error code indicating unsupported migration partner.

Can a campaign be migrated if it is not currently DCA Complete?

Yes, as long as the campaign is accepted by one DCA then it can be migrated.

Will the new CNP/DCA have access to previous suspension history before or after accepting migrating campaigns?

New CNP/DCAs will not have access to previous suspension history before or after accepting migrating campaigns. TCR will introduce a new feature after the release of the migration feature which will allow more visibility into the campaign's suspension/unsuspension history.

Can I complete a provisional campaign sharing request to a Secondary DCA if the same Secondary DCA currently has the same pre-migration campaign sitting in PENDING sharing status?

A provisional share to the same secondary DCA will not be successful if the partner has the pre-migration campaign in a PENDING sharing status. Below is an example of what will not be allowed:

- Pre-migration CNP chain: CSP_A → CNP_B → DCA_X → DCA_Y (PENDING review)
- Provisional migration CNP chain: CSP_A → CNP_C → DCA_Z → DCA_Y (PENDING review)
 - In the provisional chain, DCA_Z has shared the campaign to DCA_Y but DCA_Y has not accepted or declined the campaign in the pre-migration chain. In the provisional chain, DCA_Z will receive an error when trying to share to DCA_Y.

Appendix A: New Properties

Two new properties will be introduced: **cnpMigration** and **previouslyAccepted**. The new properties are mentioned throughout the document where applicable.

The new **cnpMigration** property is intended to distinguish campaign-sharing events affiliated with CNP sharings. Possible values are: true or false

The new **previouslyAccepted** property in combination with the **cnpMigration** property is designed to remind the upstream CNP or DCA that they've previously reviewed and accepted the campaign sharing. Possible values are: true or false



Appendix B: New Webhook Events

There are three new webhook events: `CNP_MIGRATION_COMPLETE`, `CNP_MIGRATION_CANCEL`, and `CNP_MIGRATION_PORT_OUT`.

| Event | Description |
|-------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>CNP_MIGRATION_COMPLETE</code> | <p>Marks the successful completion of a CNP migration transaction. All active participants of the migration will receive this event when the transaction completes.</p> <p>Given the following example where <code>CNP_B</code> initiates the CNP migration:</p> <ul style="list-style-type: none">• Pre-migration CNP chain: <code>CSP_A</code> → <code>CNP_B</code> → <code>CNP_C</code> → <code>DCA_X</code>• Post-migration CNP chain: <code>CSP_A</code> → <code>CNP_B</code> → <code>DCA_Y</code> <p><code>CNP_MIGRATION_COMPLETE</code> webhook event will be sent to <code>CNP_B</code> and <code>DCA_Y</code> as result of the CNP migration. Note that <code>CSP_A</code> will not receive the event because it is not an active participant of the migration.</p> <p>Sample Webhook</p> <pre>{ "brandName": "Marq", "campaignId": "CAMPXXX", "brandReferenceId": null, "brandId": "BRANXXX", "description": "CNP migration on campaign CAMPXXX is completed", "mock": false, "eventType": "CNP_MIGRATION_COMPLETE", "campaignReferenceId": null }</pre> |



| | |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CNP_MIGRATION_CANCEL | <p>Generated when the CNP migration is terminated prematurely. A CNP migration cancellation can be triggered by the campaign initiator or by the system due to expiration. This event is shared with all active participants of the migration.</p> <p>Given the following example where CSP_A initiates the CNP migration:</p> <ul style="list-style-type: none">• Pre-migration CNP chain: CSP_A → CNP_B → CNP_C → DCA_X• Provisional CNP chain: CSP_A → CNP_D → DCA_Z <p>While the campaign is pending DCA_Z acceptance, CSP_A decides to cancel the migration. As a result of this action, CNP_MIGRATION_CANCEL webhook event will be sent to active participants of the migration, including CSP_A, CNP_D and DCA_Z.</p> <p>Sample Webhook</p> <pre>{ "brandName": "911 Inc", "campaignId": "CAMPXXX", "brandReferenceId": null, "brandId": "BRANXXX", "description": "CNP migration on campaign CAMPXXX is cancelled, Reason: cancel migration", "mock": false, "eventType": "CNP_MIGRATION_CANCEL", "campaignReferenceId": null }</pre> |
| CNP_MIGRATION_PORT_OUT | <p>Notifies those CNPs and DCAs removed from the CNP chain as result of a CNP migration. Given the following example:</p> |



- Pre-migration CNP chain: CSP_A → CNP_B → DCA_X → DCA_Y
- Post-migration CNP chain: CSP_A → DCA_Z

CNP_MIGRATION_PORT_OUT webhook event will be sent to CNP_B, DCA_X and DCA_Y because they are no longer part of the post-migration CNP chain.

Sample Webhook

```
{  
  "campaignId": "CAMPXXX",  
  "cnpld": "SCSPXXX",  
  "description": "SCSPXXX revoked as a CNP for campaign  
CAMPXXX, Reason: CNP migration",  
  "mock": false,  
  "eventType": "CNP_MIGRATION_PORT_OUT"  
}
```

Appendix C: Updated Existing Webhook Events

Two of the new properties (**cnpMigration** and **previouslyAccepted**) will be introduced to the below existing webhook events for CSPs, CNPs and DCAs.

New **cnpMigration** Property

The new **cnpMigration** property is intended to distinguish campaign-sharing events affiliated with CNP sharings. The **cnpMigration** property can appear in the following existing webhook events:

- CAMPAIGN_SHARE_ADD
- CAMPAIGN_SHARE_DELETE
- CAMPAIGN_SHARE_ACCEPT
- CAMPAIGN_DCA2_SHARE_ADD
- CAMPAIGN_DCA2_SHARE_DELETE
- CAMPAIGN_DCA2_SHARE_ACCEPT

Below is a CNP migration scenario where the **CAMPAIGN_DCA2_SHARE_ADD** event is produced with the **cnpMigration** property:

- Pre-migration CNP chain: CSP_A → DCA_X
- Provisional CNP chain: CSP_A → DCA_Y → DCA_Z

When DCA_Y shares the campaign with DCA_Z (DCA 2 role), DCA_Z will be notified with the **CAMPAIGN_DCA2_SHARE_ADD** event with the **cnpMigration** property

New **previouslyAccepted** Property

The **previouslyAccepted** property in combination with the **cnpMigration** property is designed to remind the upstream CNP or DCA that they've previously reviewed and approved the campaign sharing. The **previouslyAccepted** property may appear in the following campaign-sharing events:

- CAMPAIGN_SHARE_ADD
- CAMPAIGN_DCA2_SHARE_ADD

Below is a CNP migration scenario where the **CAMPAIGN_SHARE_ADD** event would include the **previouslyAccepted** property:

- Pre-migration CNP chain: CSP_A → CNP_B → CNP_C → DCA_X
- Provisional CNP chain: CSP_A → CNP_C



When CSP_A shares the campaign with CNP_C during CNP migration, TCR recognizes CNP_C is already a CNP for the campaign, hence inserting the `previouslyAccepted = true` property/value as a reminder that the campaign was previously approved by CNP_C.

Sample webhooks for each event:

CAMPAIGN_SHARE_ADD

```
{  
  "actorId": "SCSPXXX",  
  "previouslyAccepted": false,  
  "campaignId": "CAMPXXX",  
  "cnpId": "SCNPXXX",  
  "cnpMigration": true,  
  "description": "SCSPXXX shared with SCNPXXX campaign CAMPXXX",  
  "mock": false,  
  "eventType": "CAMPAIGN_SHARE_ADD"  
}
```

CAMPAIGN_SHARE_DELETE

```
{  
  "campaignId": "CAMPXXX",  
  "cnpId": "SCNPXXX",  
  "cnpMigration": true,  
  "description": "Upstream CNP declined sharing request for campaign CAMPXXX.  
Explanation: ",  
  "mock": false,  
  "eventType": "CAMPAIGN_SHARE_DELETE"  
}
```

CAMPAIGN_SHARE_ACCEPT

```
{  
  "campaignId": "CAMPXXX",  
  "cnpId": "SCNPXXX",  
  "cnpMigration": true,  
  "description": "DCAX accepted SCNPXXX's sharing request for campaign CAMPXXX",  
  "mock": false,  
  "eventType": "CAMPAIGN_SHARE_ACCEPT"  
}
```

CAMPAIGN_DCA2_SHARE_ADD

```
{  
  "cspId": "SCSPXXX",
```



```
"dcaId": "DCAY",  
"brandName": "Brand X",  
"previouslyAccepted": true,  
"campaignId": "CAMPXXX",  
"dcaName": "DCA Y",  
"brandId": "BRANXXX",  
"cnpMigration": true,  
"description": "DCAX shared with DCAY campaign CAMPXXX",  
"eventType": "CAMPAIGN_DCA2_SHARE_ADD",  
"cspName": "CSP X"  
}
```

CAMPAIGN_DCA2_SHARE_DELETE

```
{  
  "cspId": "SCSPXXX",  
  "dcaId": "DCAY",  
  "brandName": "Brand X",  
  "actorId": "DCAX",  
  "campaignId": "CAMPXXX",  
  "dcaName": "DCA Y",  
  "brandId": "BRANXXX",  
  "cnpMigration": true,  
  "description": "DCAX declined DCAY's sharing request for campaign CAMPXXX. Explanation:  
",  
  "eventType": "CAMPAIGN_DCA2_SHARE_DELETE",  
  "cspName": "CSP X"  
}
```

CAMPAIGN_DCA2_SHARE_ACCEPT

```
{  
  "cspId": "SCSPXXX",  
  "dcaId": "DCAY",  
  "brandName": "Brand X",  
  "campaignId": "CAMPXXX",  
  "dcaName": "DCA Y",  
  "brandId": "BRANXXX",  
  "cnpMigration": true,  
  "description": "DCAX accepted DCAY's sharing request for campaign CAMPXXX",  
  "eventType": "CAMPAIGN_DCA2_SHARE_ACCEPT",  
  "cspName": "CSP X"  
}
```



Appendix D: New and Impacted CSP-API Endpoints

Below are all of the new and existing CSP-API endpoints used by CSP/CNP participants in a CNP migration.

New CSP-API Endpoints

There will be 3 new API endpoints for initiating a CNP migration. This API endpoint will be added under a new group: CNP Migration in the Swagger doc.

Initiate a CNP Migration

POST /campaign/{campaignId}/migrateCnp/{upstreamCnpId}

NOTE: Required integration for a CSP or CNP who wants to initiate a CNP migration via CSP-API.

Request:

campaignId

upstreamCnpId

Response:

204 - Successful.

501 - Invalid input parameter.

502 - Campaign record not found.

518 - Campaign expired.

590 - TCR internal system error. Please contact TCR support.

592 - Operation declined. CNP migration cannot be initiated due to data constraints.

This endpoint is used to initiate a CNP migration transaction. For the **upstreamCnpId** parameter, the caller must choose a CNP that is different from the current upstream CNP. The chosen upstream CNP will be notified via webhook event `CAMPAIGN_SHARE_ADD` with an additional event property/value `cnpMigration = TRUE`.

Example 1: CSP1 can use this endpoint to migrate from CNP2 to CNP3.

- Pre-migration CNP chain: CSP1 → CNP2
- Post-migration CNP chain: CSP1 → CNP3

Example 2: CNP2 can use this endpoint to migrate from CNP3 to CNP4.

- Pre-migration CNP chain: CSP1 → CNP2 → CNP3
- Post-migration CNP chain: CSP1 → CNP2 → CNP4



Fetch CNP Migration Transaction Status

GET /campaign/{campaignId}/migrateCnp

NOTE: Optional integration for a CSP or CNP who wants to query the current status of a CNP migration.

Request:

campaignId

Response:

Sample

```
{
  "campaignId": "CAMPXXX",
  "status": "ACTIVE",
  "cancellationReason": "Change of mind",
  "expirationDate": "2024-04-04T05:43:45.286Z",
  "createDate": "2024-04-04T05:43:45.286Z"
}
```

This endpoint is used to query the status of a CNP migration transaction. All active participants of a migration are allowed access to this endpoint. This endpoint will return the following information:

- Migration creates a timestamp.
- Migration status. 3 possible statuses: Open, Complete, or Canceled.
- Migration final timestamp. Null value until migration is in complete or canceled status.
- Migration final explanation. Populated with a cancellation reason which is supplied by the initiator.

Only the party who initiated the CNP migration is allowed to cancel the migration. A migration transaction that is already completed cannot be canceled.

This API endpoint when invoked will generate the `CNP_MIGRATION_CANCEL` webhook event, which is sent to all active participants of the migration.

Example 1: CNP2, migration initiator decides to cancel the CNP migration while the migration transaction is open.

- Provisional migration CNP chain: CNP2 → CNP3 → CNP5 (pending)
- Participants CNP2, CNP3, and CNP5 will be notified of the `CNP_MIGRATION_CANCEL` event.



Cancel an Open CNP Migration Transaction

`DELETE /campaign/{campaignId}/migrateCnp?explanation={explanation}`

NOTE: Optional integration for a CSP or CNP who wants to cancel a CNP migration via CSP-API.

Request:

`campaignId`

`explanation` - Optional

Response:

204 - Successful.

500 - Authentication error. Not a migration owner.

501 - Invalid input parameter.

502 - Record not found.

590 - TCR internal system error. Please contact TCR support.

This endpoint is used to cancel an open CNP migration transaction. Only the party who initiated the CNP migration is allowed to cancel the migration. A migration transaction that is already completed cannot be canceled.

This API endpoint when invoked will generate the `CNP_MIGRATION_CANCEL` webhook event, which is sent to all active participants of the migration.

Example 1: CNP2, migration initiator decides to cancel the CNP migration while the migration transaction is open.

- Provisional migration CNP chain: CNP2 → CNP3 → CNP5 (*pending*)
- Participants CNP2, CNP3, and CNP5 will be notified of the `CNP_MIGRATION_CANCEL` event.

Existing CNP Migration Relevant CSP-API Endpoints

In this section, we will review all the API endpoints used by stakeholders to query campaign-sharing details.

`GET /campaign/{campaignId}/sharing`

`GET /partnerCampaign/{campaignId}/sharing`

The updates are intended to be non-code-breaking. All newly introduced API parameters are optional and do not alter the API endpoint's behavior if new API parameter(s) are not supplied in the API invocation.



BETTER MESSAGING

campaign registry

These 2 API endpoints are updated to recognize new optional `cnpMigration` query parameters with default value = false. The caller should set `cnpMigration` to true when participating in a connectivity migration journey.

These API endpoints are used to query campaign-sharing details. With the introduction of CNP migration, there is a distinction between the pre-migration CNP chain and the Provisional CNP migration chain. While the default behavior is unchanged the API endpoint will return campaign-sharing details for the Provisional CNP campaign if the `cnpMigration` query parameter is true.

PUT /campaign/{campaignId}/sharing/{upstreamCnpId}

This API endpoint is used by CSP for sharing a campaign with an upstream CNP. It is NOT to be used by the CSP for any CNP migration journey. If the intent is to initiate a CNP migration, then the CSP should use the new API endpoint `POST`

`/campaign/{campaignId}/migrateCnp/{upstreamCnpId}` for this purpose.

DELETE /partnerCampaign/{campaignId}/sharing

A nominated upstream CNP can use this API endpoint to decline a campaign sharing stemming from a CNP migration process. In doing so, the downstream CNP will be notified via webhook event `CAMPAIGN_SHARE_DELETE` with an additional event property/value `migration = TRUE`.

PUT /partnerCampaign/{campaignId}/sharing/{upstreamCnpId}

A nominated upstream CNP can use this API endpoint to accept and share a campaign to further the CNP migration process. The system will produce `CAMPAIGN_SHARE_ACCEPT` to notify downstream CNP of campaign sharing acceptance with a new event property/value `cnpMigration = TRUE`. If the system detects an existing CNP match, then the CNP migration transaction is marked complete, and `CNP_MIGRATION_COMPLETE` and `CNP_MIGRATION_PORT_OUT` events are produced. Otherwise, the CNP migration transaction remains open, and the upstream CNP is notified of the `CAMPAIGN_SHARE_ADD` event marked `cnpMigration = TRUE`.

Appendix E: New and Impacted DCA-API Endpoints

In this section, we will identify all the DCA-API endpoints potentially used by DCA participating in CNP migration.

New DCA-API Endpoints

There are no new DCA-API endpoints.

Existing CNP Migration Relevant DCA-API Endpoints

In this section, we will review all the API endpoints used by stakeholders to query campaign-sharing details.

GET /dcaCampaign/{campaignId}/sharing

GET /partnerCampaign/{campaignId}/sharing

The updates are intended to be non-code-breaking. All newly introduced API parameters are optional and do not alter the API endpoint's behavior when these API parameter(s) are not supplied to the API invocation.

These 2 API endpoints are updated to recognize new optional **cnpMigration** query parameters with default value = false. The caller should set **cnpMigration** to true when participating in a connectivity migration journey.

These API endpoints are used to query campaign-sharing details. With the introduction of CNP migration, there is a distinction between the pre-migration CNP chain and the Provisional CNP migration chain. While the default behavior is unchanged the API endpoint will return campaign-sharing details for the Provisional CNP campaign if the **cnpMigration** query parameter is true.

PUT /partnerCampaign/{campaignId}/sharing

A nominated DCA can use this API endpoint to accept its role as the primary DCA. The system will produce **CAMPAIGN_SHARE_ACCEPT** to notify downstream CNP of campaign sharing acceptance with a new event property/value **cnpMigration = TRUE**. If the system detects an existing primary DCA match or if the DCA complete condition is met, then the CNP migration transaction is marked complete, and **CNP_MIGRATION_COMPLETE** and **CNP_MIGRATION_PORT_OUT** events are produced. Otherwise, the CNP migration transaction remains open.

DELETE /partnerCampaign/{campaignId}/sharing



BETTER MESSAGING

campaign registry

The nominated primary DCA can use this API endpoint to decline campaign sharing. In doing so, the downstream CNP will be notified via webhook event `CAMPAIGN_SHARE_DELETE` with an additional event property/value `cnpMigration = TRUE`.

PUT /dcaCampaign/{campaignId}/sharing

This API endpoint is used by DCA to accept its role as a secondary DCA. The system will produce `CAMPAIGN_DCA2_SHARE_ACCEPT` to notify the primary DCA of campaign sharing acceptance with a new event property/value `cnpMigration = TRUE`. If the DCA complete condition is met, then the CNP migration transaction is marked complete, and `CNP_MIGRATION_COMPLETE` and `CNP_MIGRATION_PORT_OUT` events are produced. Otherwise, the CNP migration transaction remains open.

DELETE /dcaCampaign/{campaignId}/sharing

The nominated secondary DCA can use this API endpoint to decline campaign sharing. In doing so, the downstream CNP will be notified via webhook event `CAMPAIGN_SHARE_DELETE` with an additional event property/value `cnpMigration = TRUE`.