CR046

Address Link Removal (Automated)

**Contents**

[1 Introduction 3](#_Toc6310525)

[2 High Level Design 4](#_Toc6310526)

[2.1.1 Receive Dispute API 4](#_Toc6310527)

[2.1.2 Triage Process (numero Workflow) 6](#_Toc6310528)

[2.1.3 Testing 6](#_Toc6310529)

[3 Impact 7](#_Toc6310530)

[3.1 Delivery Schedule 7](#_Toc6310531)

[3.2 Resource 7](#_Toc6310532)

[3.3 Infrastructure 7](#_Toc6310533)

[3.4 Licensing 7](#_Toc6310534)

[3.5 Service Costs 7](#_Toc6310535)

**Amendment History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date Issued** | **Author** | **Reason for Change / Description** |
| V1.0 | 17/04/2019 | Kevin Wing | Initial version |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

The purpose of this change request is to automate the removal of Address Links for Address Link Disputes where possible. This will be achieved by running a series of SQL Stored Procedures within the Receive Dispute API of the Optimus Solution.

The SQL Stored Procedures provided will contain the required logic to perform the removal of the Address Link, and provide an output upon execution allowing numero to make subsequent decisions. The current process for dealing with Address Link Disputes will be adopted where we are unable to automate the removal of a link.

We will be unable to progress with this Change Request until the SQL Stored Procedures and any associated schema changes have been applied to the SQL test environment used by the Optimus Solution (pllwindvsql012).

There is also dependency on the flexibility of test data for this change to be completed successfully.

# High Level Design

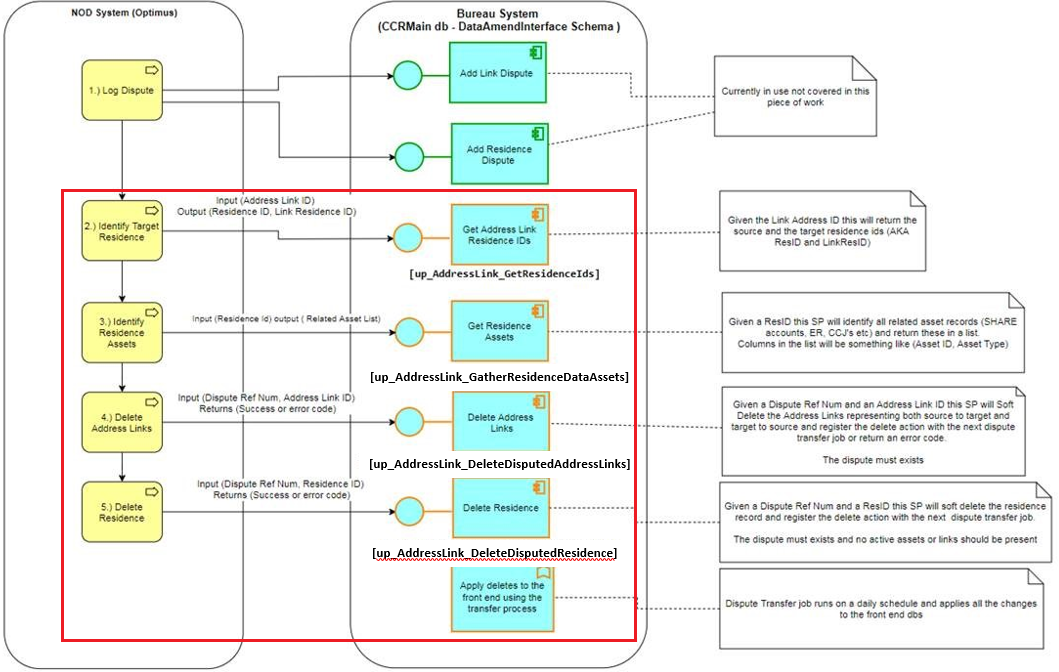
### Receive Dispute API

The initial changes will be triggered at the point at which we receive an Address Link Dispute, specifically, the Receive Dispute API that processes XML data received for each dispute. This step occurs before the Triage workflow process within the numero application.

The Receive Dispute API contains the hourly grouping rules performed before creating a Dispute. It also contains the logic for obtaining the relevant Sources (Suppliers) for each dispute, which in our case will be based on the Address Link ID. These steps will be unchanged in relation to this change request.

Once we have completed the steps above we will execute the supplied SQL Stored procedures sequentially within the Receive Dispute API. Typically, the execution of each stored procedure will provide an output to determine if we can execute the next.

We have been provided with a flow diagram below, which indicates at a high level each SQL Stored Procedure required to complete the removal of an Address Link. It is important to note that these procedures along with any associated schema changes will need to be present within our test environment (pllwindvsql012). The target database is CCRMain.



The Process / SQL Stored Procedures are as follows:

1. **Identify Target Resource** – up\_AddressLink\_GetResidenceIds

* Inputs: Address Link ID
* Outputs: Residence ID, Link Residence ID

1. **Identify Residence Assets** – up\_AddressLink\_GatherResidenceDataAssets

* Inputs: Residence ID
* Outputs: Related Asset ID

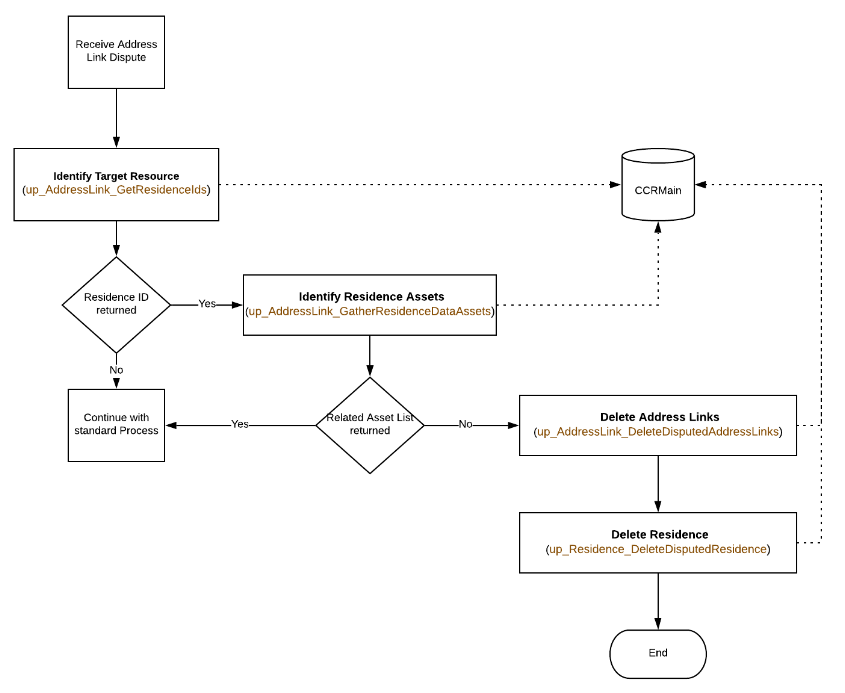
1. **Delete Address Links** – up\_AddressLink\_DeleteDisputedAddressLinks

* Inputs: Dispute Reference Number, Address Link ID
* Outputs: Success / Failure flag

1. **Delete Residence** – up\_Residence\_DeleteDisputedResidence

* Inputs: Dispute Reference Number, Residence ID
* Outputs: Success / Failure Flag

The diagram below illustrates the Stored Procedure execution process.



On completion of the process above (Delete Residence) we will mark the Dispute Case to indicate an Address Link Removal automation success or failure. This will be done within our stash repository against the DisputeCase stash type.

### Triage Process (numero Workflow)

Once we reach the Triage process within the numero workflow we can make decisions based on the flag value we created in the previous step.

#### Address Link Removal - Successful

Where we have successfully automated the removal of an address link we will perform the following steps:

* Suppress the creation of any Supplier Requests relevant to the Address Link Dispute
* Suppress the creation of any Supplier Notifications relevant to the Address Link Dispute
* Suppress the creation of any SLA Notifications relevant to the Address Link Dispute
* Update the Dispute Case status to “Complete Successful”
* Spawn a “Dispute Correspondence” work item. The following entities/values will be required:
  + Dispute Outcome for Corres: Accept
  + Dispute Case Actioning Outcome: Address Link Removed (Auto)

#### Address Link Removal - Unsuccessful

Where we have not been able to automate the removal of the Address Links, we will continue with the current process as it is today.

### Numero Configuration / CMS

We will provision a new paragraph with our CMS which will be used within the automated outbound correspondence letter, once the address link has been removed and the Dispute Case has been automatically closed. We will also create additional rules to facilitate the spawned Outbound correspondence work item. This will ensure the correct CMS paragraph is inserted into the letter.

### Testing

**Development**

Providing there is sufficient data (Address Link IDs) in the relevant CCRMain tables, we can use them within our test dispute cases. This may present an issue looking up the Credit Report for the Consumer when we are unable to match the Address Link ID to one on the Credit Report. However, we can work around this to satisfy testing in our development environment.

**System Test**

When the changes are promoted to the System Test environment we will require a level of flexibility on the test data available within CCRMain. This is because the testers will be raising Disputes via Noddle and Ginger. We need to ensure that Address Link Disputes raised from these sources have a matching Address Link ID against a Residence ID within the CCRMain tables.

If we are unable to achieve this, we will not be able to complete end to end testing in System Test.

# Impact

## Delivery Schedule

This is a major change which is required as part of Phase 1 Go-Live.

## Resource

|  |  |
| --- | --- |
| Description of work | Effort (days) |
| Detailed Design  Build – Connect API  Build - Stash  Build – Numero Build  Build – Numero CMS  Internal Testing  Project Management | 1  6  0.5  0.5  0.5  3  1 |
| **Total Effort** | **14** |

## Infrastructure

There are no impacts on the proposed solution architecture because of this change.

## Licensing

There are no licensing implications related to this change.

## Service Costs

There are no service costs related to this change.