



**Unity Programme**

**UK – Release 4**

**Validate Address Service**

**Technical Design**

**Version 0.4**

Table of Contents

[1 Introduction 4](#_Toc440364206)

[1.1 Talend Platform Overview 4](#_Toc440364207)

[1.2 Objective 5](#_Toc440364208)

[1.3 Distribution List 5](#_Toc440364209)

[1.4 Glossary 5](#_Toc440364210)

[2 Technical Design 7](#_Toc440364211)

[2.1 Requirements Overview 7](#_Toc440364212)

[2.2 High Level Component and Dataflow 7](#_Toc440364213)

[2.3 Design Requirements 8](#_Toc440364214)

[2.3.1 Data Validation and Business Rules 8](#_Toc440364215)

[2.3.2 Fault Codes and Description: 8](#_Toc440364216)

[2.4 Design Decisions 9](#_Toc440364217)

[2.5 Assumptions 9](#_Toc440364218)

[2.6 Risks 9](#_Toc440364219)

[2.7 Issues 10](#_Toc440364220)

[2.8 Interface Details 10](#_Toc440364221)

[2.8.1 Interface Specification Details 10](#_Toc440364222)

[2.8.2 Interface Frequency and Schedule 10](#_Toc440364223)

[2.9 Flowchart 11](#_Toc440364224)

[2.10 Mapping and Transformation Rules 12](#_Toc440364225)

[2.11 Technical Component Details 12](#_Toc440364226)

[3 Non-Functional Design 17](#_Toc440364227)

[3.1 Security and Data Access 17](#_Toc440364228)

[3.2 Performance Considerations 17](#_Toc440364229)

[3.3 Deployment 17](#_Toc440364230)

[3.4 Error Handling and Fault Management 18](#_Toc440364231)

[3.5 Housekeeping Policies 18](#_Toc440364232)

[4 Appendix 19](#_Toc440364233)

[4.1 Validate Address ICD 19](#_Toc440364234)

**Document Maintenance**

***Document Owners***

|  |  |
| --- | --- |
| Functional owner(s) / Architect: Narayana Velaga | |
| Technical Owner(s): Sekhar Reddy Karnat | |
| Current Version: | 0.4 |
| Date Last Updated: | 05/12/2015 |

***Version Control***

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date Updated | Revision Author | Summary of Major Changes Made |
| 0.1 |  | Sekhar | Initial draft |
| 0.2 | 07/05/2015 | Sekhar | Updated the document as per Naveen comments |
| 0.3 | 08/05/2015 | Sekhar | Updated the document as per Jasneet and Narayana(Sec 3.4 & 3.5) Comments |
| 0.4 | 05/12/2015 | Kanishka Gupta | Updated Fault Codes and Description |

**Note:** These documents are strictly for specific Virtusa/HomeServe use only. They shall not be shared with an external party other than HomeServe. These documents should always be kept securely and employees shall use reasonable care protecting these documents from unauthorized use or disclosure to a third party. This category also covers client intellectual property where Virtusa has a non-disclosure agreement with HomeServe.

**Document Approval**

Virtusa Corporation and HomeServe have reviewed this document and hereby agree that the contents herein are accurate. Any changes to this document must be communicated in writing and signed-off by both parties.

|  |  |
| --- | --- |
| Signature | Signature |
| Date: | Date: |
| Name: | Name: |
| Client: HomeServe | Virtusa Corporation |
| Address:  Cable Drive,  Walsall,  West Midlands  WS2 7BN | Address:  Virtusa (UK) Ltd  Level 8  26 Finsbury Square London EC2A 1D  E-mail: [info@virtusa.com](mailto:Info@techprov.com)  URL: [www.virtusa.com](http://www.techprov.com) |

# Introduction

HomeServe is looking to transform its customer experience by moving to customer centric stack through Unity Program. Key Objectives of the program are,

* Transform into a customer-centric business, efficiently offering right product mix via multiple digital channels
* Grow by reducing cycle time to launch new products and integrating new partners
* Engage with customers via the channel they desire and the right product, with a single customer ‘account’

## Talend Platform Overview

The following diagram outlines the overall Talend Platform architecture. This technical design document covers the Validate address service mechanism highlighted in amber



## Objective

This Technical design document outlines the capability of Validate Address Service exposed for validating the address captured in the calling system. Also how the request / response will be handled through Talend ESB.

This document mainly aimed to support the construction phase and it covers the following details,

1. High level data integration flow
2. Design Decisions and Assumptions
3. Mapping specifications
4. Talend Components
5. Non-functional designs

## Distribution List

Approval List

|  |  |  |
| --- | --- | --- |
| **Name** | **Organization** | **Role** |
| Mark Buck | HomeServe | Solution Architect |
|  |  |  |
|  |  |  |
|  |  |  |

Discussion / Review List

In addition to those on the approval list above comments will be sought from the following people/groups.

|  |  |  |
| --- | --- | --- |
| **Name** | **Organization** | **Role** |
| Jasneet Singh | Virtusa | Lead Data Architect |
| Naveen Mula | Virtusa | Lead Data Architect |
| Satya T | Virtusa | Offshore Architect |
|  |  |  |

Governance

|  |  |  |
| --- | --- | --- |
| **Name** | **Applicable?** | **Approval Details** |
| Technical Design Authority | Y |  |
| Business Design Authority | N | NA |
|  |  |  |
|  |  |  |

## Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| MDM | Master Data Management |
| SOAP | Simple Object Access Protocol |
| XML | eXtended Markup Language |
| WSDL | Web Services Description Language |
| SAM | Service Activity Monitoring |
| TAC | Talend Administration Console |
| ESB | Enterprise Service Bus |
| DQ | Data Quality |

# Technical Design

## Requirements Overview

As part of Unity Programme, Release 4 will provide a capability to validate address captured in the calling system against Capscan by using Validate Address Service.

Below are the derived technical requirements for Validate Address Service

* Create Validate Address service in Talend ESB to accept validation request and handle response from Capscan
* Create SOAP wrapper in Talend ESB to expose Capscan MatchCode API to the calling systems for basic address validation process
* Global schema to be used for request and response for Address Validation service
* Build SOAP Fault response for exceptions

## High Level Component and Dataflow

The following diagram explains the high level component and dataflow of source system and Capscan integration.



**Validate Address Service Criteria:**

Source system can use the following search criteria/options to check the UK address with Capscan.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Search Fields | Option1 | Option2 | Option3 | Option4 |
| Flatnumber | Y | Y | N | N |
| Housenumber | Y | N | Y | N |
| Country | Y | Y | Y | Y |
| Postcode | Y | Y | Y | Y |

There are more attributes present in request that may be available from source systems, but the business rules will be implemented as per the search options.

## Design Requirements

## Data Validation and Business Rules

The following business rules will be applied coming from source system.

|  |  |  |  |
| --- | --- | --- | --- |
| Ref. | Data Element | Business Rule Overview | Error Code |
| Capscan-BR01 | Postcode | Cleansing Rules – Postcode should be converted to UPPER and validate the post code format with UK format when country code Is GB(Reuse the existing postcode regex) | NA |
| Capscan-BR02 | Mandatory check | Check the request for below mandatory fields for   * PostCode * Country | Ref 2.3.2 |

## Fault Codes and Description:

|  |  |  |  |
| --- | --- | --- | --- |
| Ref. | Validation | Error Code | Error Description |
| 1 | Invalid PostCode | 103 | Invalid Postcode Format |
| 2 | Technical Errors | 105 | Technical Server/Job/Sevice error |
| 3 | SSID | 106 | Invalid Source System Identifier |
| 4 | Mandatory Check | 107 | XSD Mandatory fields are missing |
| 5 | Connection | 120 | Capscan Server is down or not reachable |

## Design Decisions

| Ref. | Design Decision |
| --- | --- |
| D01 | Source system will use Validate Address Service for Address validation against Capscan Matchcode API |
| D02 | Source system will send request in XML as per globalised schema for address validation |
| D03 | Talend job will validate xsd and mandatory fields check before sending request to Capscan |
| D04 | Validate Address real-time services will be built and deployed using Talend ESB (version 5.6.1) module |
| D05 | Validate Address function in Talend ESB will accept validation request and handle results(multiple or zero) from Capscan |
| D06 | Integration pattern between Source System and Capscan will be via existing HomeServe ESB infrastructure [Source system <-> Talend ESB <-> Capscan] |
| D07 | Incoming SOAP request messages will be authenticated by https – username and password mechanism (Syncope) |
| D08 | Validate address service will connect to Capscan by using intermediate java libraries. |

## Assumptions

| Ref. | Design Decision |
| --- | --- |
| A01 | The data exchange between Source System and Capscan should be based on ICD format agreed. Refer the appendix section 4.1 |

**Note: Please refer to ICD-INT-18 for more assumptions Section 2.1**

## Risks

|  |  |  |  |
| --- | --- | --- | --- |
| Risk No. | Description | Comments | Owner |
| R01 | Service Interruption between source system and Capscan. | In case of integration service failure between source system and Capscan (e.g. service is down, network related issue etc.). | Infrastructure |
| R02 | Address not found | We could have invalid address being captured in Pega. This may need manual intervention to correct the address | Calling Application |

## Issues

| Sno. | Description | Resolution | Owner | Status |
| --- | --- | --- | --- | --- |
|  | None |  |  |  |

## Interface Details

## Interface Specification Details

Source System will consume the Validate Address real time service (24X7) hosted within HomeServe network.

## Interface Frequency and Schedule

| Interface Name | Frequency |
| --- | --- |
| Validate Address Service | Immediate (real-time)  On demand |

## Flowchart

The following flowchart covers the detailed data and component flow for Validate Address service implementation.

**Flow Steps:** Below are the job design flow steps.

* Request will come from source system in xml format.
* Request should contain user authentication in header if authentication failed then send an error.
* Validate the request and send an error if not valid with error code and description.
* Load the Context variables and check the connection with Capscan if failed to connect capscan send an error. This connection will be open always and it is not required to close the connection for every response.
* Check the xsd/wsdl validity.
* Check the mandatory fields and post code format in the request.
* Send results to Capscan and lookup the results with Capscan.
* Generate the response in xml format from capscan and send back to source system with results.
* Service can have functional errors and technical errors. Design releated errors called functional errors and all remaining errors are technical errors.

## Mapping and Transformation Rules

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Item | Mapping Document | Version |
| 1 | Source system to Capscan |  | 0.1 |

## Technical Component Details

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Functional Component | Description | Key Development Guidelines |
| 1 | Talend ESB – Https Authentication | This component will be used to authenticate the incoming message using https – username and password. | Use Talend ESB out-of-box security option to enable the https authentication. |
| 2 | Talend ESB –  Connection Pooling | This component is used to open a connection to Capscan server. | tFixedFlowInput: generates as many lines and columns as you want using the context variables.  tJavaFlex: Should be use to write the java custom code to define the Capscan connection  tLibraryLoad: Should be used to define the capscan jar file. |
| 3 | Talend ESB -  WSDL/XSD Definitions | The Xml schemas and WSDL definitions will be maintained within server for each service exposed from MDM using Talend ESB | Use Talend ESB perspective to create and maintain the XSD and WSDLs**.**  Use Talend ESB Runtime options to validate WSDL |
| 4 | Talend DI - Real-time Request Listener | Listens the Source Request SOAP from Source system and send the request to downstream | tESBProviderRequest: used to capture SOAP request from Source system and parse the request related attributes send it to downstream |
| 5 | Talend DI - Data Validation | Validate the data from the Source System Request which is XML | tMap: Validates the request attribute based on the validation rules define. |
| 6 | Talend DI - Data Mapping/Transformations | In this component we apply Transformation Logic to load and validate the data | tFixedFlowInput: generates as many lines and columns as you want using the context variables.  tConvertType: allows specific conversions at runtime from one Talend java type to another  tXMLMap: transforms and routes data from single or multiple sources to single or multiple destinations.  tHashOutput: This component loads data to the cache memory to offer high-speed access, facilitating transactions involving a large amount of data.  tMap: transforms and routes data from single or multiple sources to single or multiple destinations.  tJava: makes it possible to extend the functionalities of a Talend Job through using Java commands.  tFlowToItereate:  iterates on the input data and generates global variables  tJavaFlex: Should be use to write the java custom code to define the Capscan connection and result sets |
| 7 | Talend DI - Business Rules | This Component checks the business Rules by applying various validation checks (regular expression) | Rules should be applied on the attributes to validate the request attributes.  Eg: Validate/check the postcode attribute using matches (<pattern>) |
| 8 | Talend DI / MDM - Response Generator | Generates the Response on success/failure and send the messages to Ensura | tESBProviderResponse: Should be used to generate response in xml format with error code and error description.  tESBProviderFault: Should be used for generating error messages with code and description |
| 9 | Talend DI - Functional Error handling | Functional errors are generally included in the every response to intimate the users about the functional error and send with proper error code and error message. | tESBProviderResponse: should be used to generate response in xml format with error code and error description. |
| 10 | Talend DI - Technical Error Handling | Errors are generally occurs when the component failed to execute like Capscan connection issues or any component failed inside a job | tESBProviderFault: used for generating error messages with code and description |

**Capscan MatchCode API Details:**

Below are the Capscan Match Code API libraries used to connect capscan and cleanse the address. These jar files will be used for connecting Capscan to process the address data validation in real time.

* Capscan.jar
* Capscan-common-lang.jar
* AddressValidation.jar – To be build using the above libraries

Methods:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Method Name | Parameter Type | Parameter Name | Data Type | Expected Value(Dev) |
| McConnection(String, String, String, int) | Input | PoolName | String | TALEND\_PAF |
|  | Input | Hostname | String | Test: prd2capscd4.hgb.hs.int |
|  | Input | appName | String | “ValidateAddressService” |
|  | Input | Connection  Mode | Int | 1(CONNORIENTED) |
| McDisconnect() | Input | NA | NA | NA |
| search(int, Vector, Vector, Vector, int) | Input | Matcher | Int | SINGLE SEARCH = 0 |
|  | Input | getlist | Vector | List of select columns |
|  | Input | Data | Vector | List of input fields |
|  | Input | TIMEOUT | int | 15 |

**AddressValidation.jar:** In this jar we can define all the capscan input and output parameters along with search type.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Method Name | Parameter Type | Parameter Name | Data Type | Expected Value |
| createConnection(String, String, String, int) | Input | PoolName | String | TALEND\_PAF |
|  | Input | Hostname | String | Test: prd2capscd4.hgb.hs.int |
|  | input | appName | String | “contest” |
|  | input | Connection  Mode | Int | 1(CONNORIENTED) |
| search(Vector, MatchCodeConnection) | Input | Data | Vector | List of input fields |
|  | Input | Obj\_conn | MactchCodeConnection Object | Connection object from MatchCode Class |
| search(Vector, boolean, int) | input | Data | Vector | List of input fields |
|  | Input | combinedListFormat | boolean | false |
|  | Input | Matcher | Int | CROSS\_MATCH = 0 |
| buildDtos(String[], boolean) | Input | Resultset | String[] | Result from search method |
|  | Input | combinedListFormat | boolean | False |
| buildAddressDtos(results, combinedListFormat) | Input | Resultset | String[] | Result from search method |
|  | Input | combinedListFormat | boolean | False |
| processAmbiguity(results, combinedListFormat) | Input | Resultset | String[] | Result from search method |
|  | Input | combinedListFormat | boolean | False |
| AssembledAddressDto(String[] results) | Input | Resultset | String[] | Result from search method |

# Non-Functional Design

## Security and Data Access

The Real-time service “**Validate Address”** will be hosted within HomeServe network. The protocol will be SOAP Over HTTPS and requests and response end points will be secured with **Username & Password** as per the HomeServe security policies. All required SSL certificates should be configured in all environments (TAC) to support HTTPS

The following steps should be taken to enable the http username and password authentication for each service,

1. SOAP request from source system header will have username and password as per the security policy
2. Source System will be responsible for the storing of the username/password pair required to access the service, this information should be stored securely and safely
3. Source system will have credentials created in syncope and these credentials will be unique for each source system and will be passed as part of SOAP header.

## Performance Considerations

The validate address service will align with following performance requirements.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Service | Number of Request per Hour | Actual Response Time | Expected Response Time |
| 1 | ValidateAddress | 100 | 2190 ms | 3 sec |
| 2 | ValidateAddress | 500 | 6237 ms | 10 sec |
| 3 | ValidateAddress | 1000 | 10833 ms | 25 Sec |
| 4 | ValidateAddress | 5000 | 48209 ms | 100 Sec |

The following key performance guidelines need to be considered while implementing the service to meet the above performance requirements. Also refer the Talend Development Guidelines and Best practices for more details.

## Deployment

Validate Address service will be deployed and monitored using Talend ESB module. The service artifacts will be available in Nexus repository and using Talend Administrator Centre (TAC), these artifacts will be deployed in Talend ESB runtime.

## Error Handling and Fault Management

To assist client systems with exception handling, all functional/business logic related errors will be part of SOAP service response with an appropriate Error code and Error Description will be sent as per the Error Catalogue.

## Housekeeping Policies

Talend services actual request and response messages will be logged in Talend Administrator Console (ESB Job Conductor) using Service Activity Monitoring (SAM) feature. The housekeeping policy will be aligned with Talend ESB policy defined for HomeServe.

# Appendix

## Validate Address ICD

<https://serveusa.sharepoint.com/sites/HomeServe/GlobalResources/GlobalProjects/Ensura/Programme%20Documentation/INT18_MDM_ValidateAddressServices_ICD.docx?d=wce342ae0f93344e5a0d30b709a0856bf>