



# Integrating with DPD for Selfprinter Setups **GeoRouting**

Version 2.0: DPD Belux 08/2021

# Contents

<b>1. Introduction</b>	<b>1</b>
<b>2. General Process description</b>	<b>2</b>
2.1. Subtypes in customer routing file	3
2.2. Input parameter validation	4
2.3. CustomerRoute lookup	4
2.4. CUSTOMER3OSORT	4
2.5. General structure of the file	4
<b>3. Input parameter validation</b>	<b>7</b>
3.1. Version Check	9
3.2. Country validation	11
3.3. PostcodeValidation	16
3.4. Validation of SoCode	18
3.5. Validation of AsCode	20
<b>4. Route Calculation</b>	<b>22</b>
4.1. Calculate Destination	22
4.2. Adding additional information for the destination depot	27
4.3. Resolving the destination Network	31
<b>5. O-Sort calculation</b>	<b>33</b>
5.1. Used Table: CUSTOMERR3OSORT	34
5.2. Technical Background	34
<b>6. Tables only used to create Label</b>	<b>40</b>
6.1. SOSERVICEINFO (by language f.e.: SOSERVICEINFO.EN)	40
6.2. BUSINESSUNIT	40

# Revision

Version	Changes	Author
Version 0.1	Initial version	Sebastian GÖRGEN, DPDgroup
Version 1.1	Bug Fixe on CUSTOMER3OSORT + change column order on DEPOT table	Sebastian GÖRGEN, DPDgroup
Version 1.2	<p>Table ELEMENTS :</p> <ul style="list-style-type: none"> <li>- Elements Descriptions added</li> </ul> <p>Table DEPOT :</p> <ul style="list-style-type: none"> <li>- Change all DEPOT to DEPOTS</li> <li>- Add GPSLONG and SGPSLAT instead of SGPSLOCATION</li> </ul> <p>Table SOCODE :</p> <ul style="list-style-type: none"> <li>- Samples are corrected</li> </ul>	Sebastian GÖRGEN, DPDgroup
Version 2.0	Reworked version DPD Belux (layout and references)	Andy Van Hove, DPD Belux

# 1. Introduction

Dear DPD Customer, Partner, IT Integrator

This document describes the functional and technical layout of the DPDgroup Routing database. It contains the specification of each logical table, how each table will work, how the tables will work together, the layout of the files, the process description and how the database will be distributed. The document describes also the technical implementation of the GeoRouting database. It could be used as a basis for software development.

This document is one of three pillars in the self-development documentation of DPDgroup standardised shipments, the other two being the Data- (GEODATA) and the Label (GEOLABEL) specification components. The specification of the GeoRouting uses the specifications of the Common File Format (GEODATA) and supports the Common Shipment Printing (GEOLABEL).

GeoRouting contains the reference database to the depots, the services and the country zip code allocations within the DPDgroup and also contains rules concerning the management of the GRDB (GeoRoutingDataBase) and restriction to be considered. The GeoRouting is updated frequently to assimilate the operational changes and availabilities in the DPDgroup network.

GeoRouting is the DPDgroup standard and is distributed and understood by each DPD Business Unit throughout the network. A successful integration with the GeoRouting will allow you to easily integrate along the entire network, regardless of the country of shipment origin. The DPD Customer IT department of the country of shipment origin is responsible for validating the composed routing, label and data. In case of a shared integration with multiple DPD Business Units, one BU will take the lead in the validation and support process and coordinate your integration.

This document also contains references to other documentations. To be able to understand the GeoRouting Common Routing Database completely the other project documentations may have to be read as well.

What to expect?

- The GeoRouting explained by flow charts
- Detailed tables with possible values and restrictions
- Guidelines for building the logic and rules for validation
- Examples of the routing creation with reference to label and data
- Appendix explaining terminology and SQL statements

We wish you success and will be at your disposition to handle all technical questions and concerns.

Thank you,

The joint DPDgroup Customer IT departments

# 2. General Process description

Customer file is a simplified routing for customers, which only has routing relevant data in the file and the routing execution is also simplified into a single step. There will be a customer file for each BU to support basic ALLOW checks.

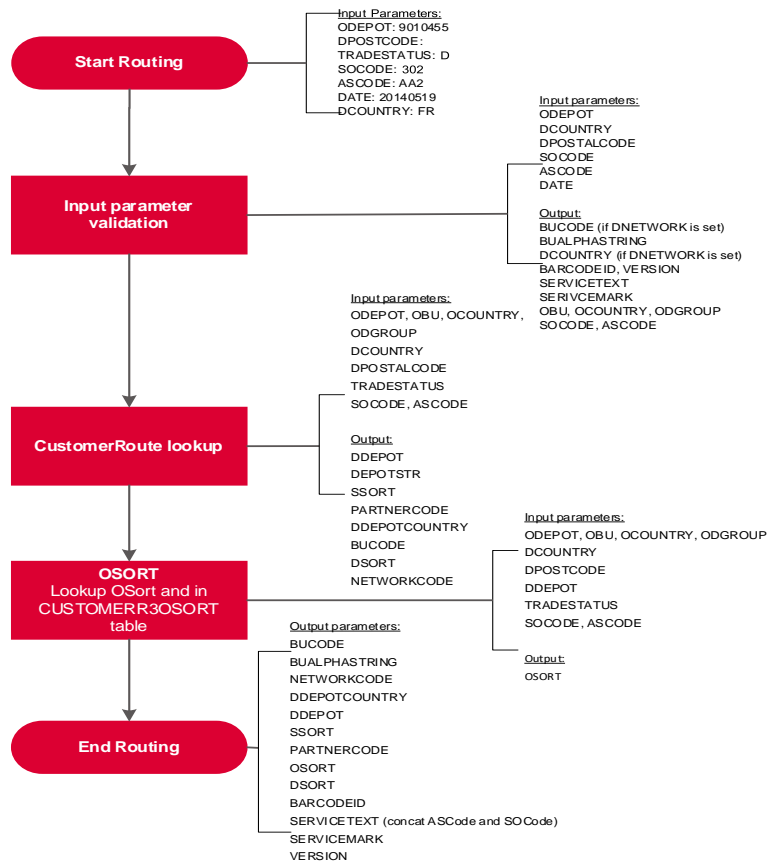
These checks will only contain ALLOW checks based on country and zones and BU.

This simplified routing has a limited functionality; the following features are not supported:

- Csort
- City validation

## Customer file rows routing flow

The simplified routing flow is as follows:



## 2.1. Subtypes in customer routing file

The customer routing files has the following subtype names (most of them are part of the standard georouting as well) :

Name	Description
VERSION	Validity, and version code information.
BUSINESSUNIT	The table provides reference numeric and alphabetic code to identify a Business Unit.
NETWORK	The NETWORK table contains the destination networks of GeoPost. The destination networks are referenced with the user defined range of the ISO country codes (900-999).
POSTALCODE	The POSTALCODE table contains the valid zip codes per country
COUNTRY	The country table is needed to validate country codes, to translate the country code into the country abbreviation which will be printed on the label, to find out the languages which are spoken in the country and if the country has no post code system.
COUNTRY.XX	The country name table will be needed to help the operator or the customer to choose the right country. There will be one table per language to give the routing program the possibility to suit local needs.
DEPOTS	The depots table is needed to validate depot codes, and to get the depot address if it is needed
SOCODE	The SOCode table contains the sorting relevant service codes.
SOSERVICEINFO.XX	The SOServiceinfo_XX table contains the service information in the respective language.
ELEMENTS	The Elements table contains elements which can be associated with an SOCode or an ASCODE.
ELEMENTS.XX	The Elements XX table contains the translation of the element to a specific language.
ASCODE	The table provides reference numeric and alphabetic code to identify a Business Unit.
CUSTOMERR3OSORT	The CUSTOMERR3OSORT is the routing table which selects the origin sort based on the searching parameter fields.
CUSTOMERROUTE	The customer routing table includes all data from NETWORKPLAN, R1DBU, R2ROUTE, R5DSORT tables.

## 2.2. Input parameter validation

In the validation process input parameters are validated and the Postcodes are validated against different pattern rules. You can find the detailed description of this process in Chapter “

Input parameter validation”

### 2.3. CustomerRoute lookup

CustomerRoute lookup process calculates the main information for the label like the Destination Depot, the D-Sort and the S-Sort. You can find the detailed description of this process in Chapter 4 Route Calculation

### 2.4. CUSTOMER3OSORT

The CUSTOMERR3OSORT is the routing table which selects the origin sort based on the searching parameter fields. You can find the detailed description of this process in Chapter 5 O-Sort calculation.

### 2.5. General structure of the file

GeoRDB is an automatically described file with semicolon ( ; ) separated fields. All tokens will be generated in a single file.

The file begins with a subtype definition section, where all used subtypes have to be defined.

Next section is the data content.

Value	Description
#FILE;<Filename>; #VERSION;<NNNNNNNN>; #DEF:GEOROUTING:SUBTYPE1;Field1;Field2;...;FieldN;; #DEF:GEOROUTING:SUBTYPE2;Field1;Field2;...;FieldN;; #DEF:GEOROUTING:SUBTYPE3;Field1;Field2;...;FieldN;; #DEF:GEOROUTING:SUBTYPE4;Field1;Field2;...;FieldN;; #DEF:GEOROUTING:SUBTYPEN;Field1;Field2;...;FieldN;; ...	Subtype definition
NETWORKPLAN; Value1; Value2;Value3; SUBTYPE1;Value1;Value2;...;ValueN; SUBTYPE2;Value1;Value2;...;ValueN; SUBTYPE1;Value1;Value2;...;ValueN; SUBTYPE2;Value1;Value2;...;ValueN; SUBTYPE3;Value1;Value2;...;ValueN;	Value of subtype in the order defined by the dataflow



SUBTYPE4;Value1;Value2;...;ValueN; SUBTYPE5;Value1;Value2;...;ValueN; SUBTYPE1;Value1;Value2;...;ValueN; SUBTYPE2;Value1;Value2;...;ValueN; SUBTYPE3;Value1;Value2;...;ValueN; SUBTYPE4;Value1;Value2;...;ValueN; SUBTYPE5;Value1;Value2;...;ValueN; ...	
#END;	

### 2.5.1. Rules for the file format

- The encoding of the file is UTF-8 without byte order mark (BOM).
- Line ending is in windows-style (\r\n or 0x0d 0x0a).
- Semicolon (;) characters are not allowed in any data field.

## 2.5.2. Version management

For each version of RDB there is a matching RDB file. It is allowed to keep more than one version of RDB but it is suggested to differentiate them by adding alias of version number for each table name except of the version table.

In this case the version table should not follow the logic to extend the table name. The VERSION table will be used then to select the version for routing. More details how to select the proper version can be found in chapter 3.1 Version Check.

Example:

Table: DEPOTS

DEPOTS\_{version number} / DEPOTS\_ 2009070

# 3. Input parameter validation

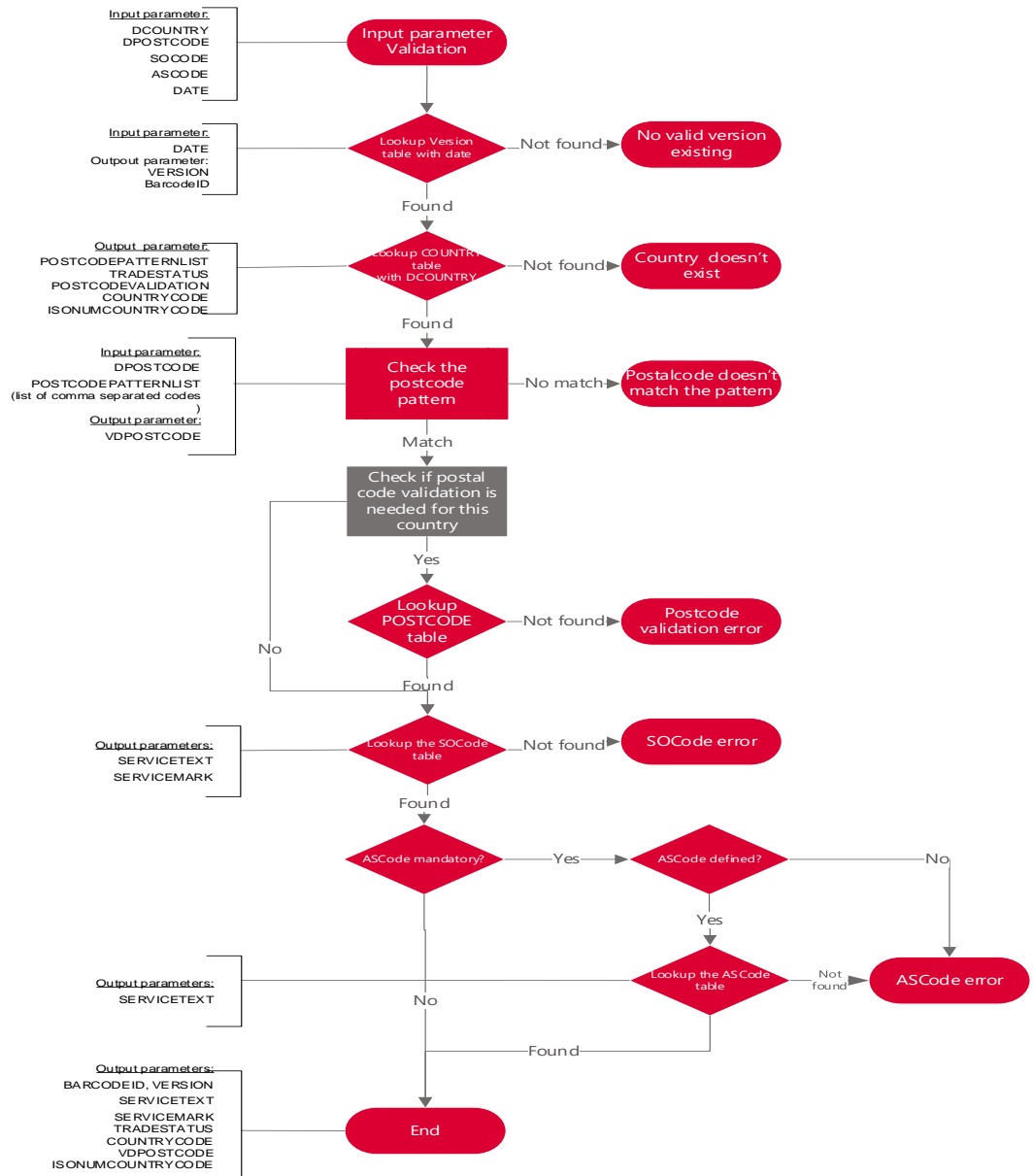
In this chapter, we will describe the tables and the process used for the input parameter validation. The main purpose of this validation is to check all information used for the routing in the beginning of the overall process if they are valid or not. This includes if the product code you are using is existing, the destination country exists and is valid and if DPD is able to deliver in principle to a postcode defined as destination postcode.

Beside of this most important step, the input parameter validation includes also all tables and checks you need to perform to get the data for the label, that are included in the tables directly.

## Process Overview

In the validation process, we do the following main checks:

- Check if Country exists in the country table and check if the postcode format is matching our expectations (postcode pattern matching)
  - For countries using Postal Code Systems, validate if the Postal Code provided is in the list of valid postal codes (Lookup POSTALCODE table)
- Validate Product of the Parcel
  - Check Input Parameter SoCode in SOCODE Table and extract the SERVICETEXT used on the label
    - For SoCodes requiring an AsCode check if Input Parameter AsCode exists and validate it against AsCode Table and extract the SERVICETEXT used on the label
- Check if Routingdatabase is up-to-date
  - Check validity in VERSION table and lookup for the Barcode-ID (first digit in Code128 28 Barcode



### 3.1. Version Check

This step validates if the Routingdatabase you're using is up-to-date. The date for the validation is the date when you create the shipment.

#### 3.1.1.Used Table:VERSION

A version is valid if your date is between or equal the ValidFrom and ValidTo dates. You have to select the highest version number to ensure, that you get the latest routing result. We will use this version number also to create and select results in all other tables.

Fields used to select the proper values are marked in grey; fields used to retrieve the results are marked in red.

Field	Length	Opt.	Comment
VersionNumber	8	N	Internal Version number of the database. Printed on the Label
ValidFrom	8	N	Starting date for the validity of the routing database. Example: 20200901
ValidTo	8	N	Ending date for the validity of the routing database. Example: 20200930
BarcodeID	1	N	BarcodeID is the starting character of the 28 digit barcode.

If you don't receive a match validation has failed, stop Routing Process with ROUTING OUTDATED Error

## 3.1.2. Technical Background

### 3.1.2.1. Creating VERSION Table

```
CREATE TABLE
version
(
    versionnumber varchar(8) NOT NULL,
    validfrom integer NOT NULL,
    validto integer NOT NULL,
    barcodeid varchar(1) NOT NULL,
PRIMARY KEY(versionnumber)
);
```

### 3.1.2.2. Loading VERSION Table

Each field of VERSION subtype should be inserted as a new row

### 3.1.2.3. Search Query

```
SELECT
    version.barcodeid,
    version.validto
    version.versionnumber

FROM
    version

WHERE
    '$shipment_date' > version.validfrom

ORDER BY
    version.validfrom desc

LIMIT
    1;
```

If you don't receive a match validation has failed, stop Routing Process with ROUTING OUTDATED Error

## 3.2. Country validation

We use the country table in the validation process to answer the following questions.

- Does the Delivery Country exist in COUNTRY Table?
- Is the Delivery Postalcode matching the Postcode Pattern for the Country?
- Does the Delivery Country require checking the Delivery Postcode via the POSTALCODE Table?

### 3.2.1.Used Table: COUNTRY

Fields used as input are marked in dark grey; fields used as output are marked in red.

Field	Length	Format	Comment
CountryCode	2	A	ISO 3166 country code – alpha 2 Used as the Country identification in all following steps
ISONumCountryCode	3	N	ISO 3166 country code – numeric
ISOAlpha3CountryCode	3	A	ISO 3166 country code – 3-digit alpha
DestinationLanguages	128	A	Comma separated list of destination languages based on ISO 639-1
FlagPostCodeNo	1	N	Does this country uses Postcodes 1=no, 0=yes
PostCodePattern	128	A	See PostCode Pattern validation
FlagNeedState	1	N	Does the country requires to assign a state in the country
PostCodeValidation	2	A	See PostCodeValidation
CityValidation	2	A	Mode to validate the City table – not used in Customer Specification
FreeTradeArea	2	A	Free Trade Area (FTA) of the country – customs clearance needed if destination country is not the same. If no FTA defined, customs clearance needed for sure
DummyCountryCode	1	A	Is this country a dummy country?

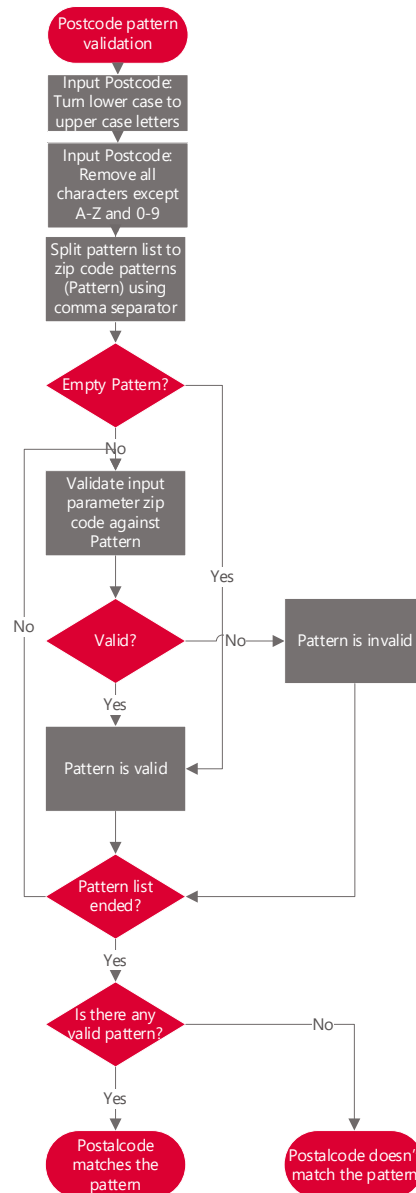
### 3.2.2. Postcode Pattern validation

The Postcode Pattern validation has the two following purposes:

- Remove unnecessary information from the Postcode
- Validate if the Postcode matches the format of the Postcode of the country

Postcode Pattern validation process is only used if the “FlagPostCodeNo” is set to “0” in the country table for this Destination Country. For Countries where the “FlagPostCodeNo” the postcode will be handles as an empty one and should be discarded for all other processes including the generation of the Shipment Data.

#### 3.2.2.1. Postcode Pattern validation flow





### 3.2.2.2. Insight how to read the Postcode Pattern

A pattern may consist of a number of characters where each character in the pattern corresponds to a character class, i.e. a number of permitted characters in the post code. In addition to the screening for certain characters a character class may define if a character has to be removed from the post code to be able to fit it into the barcode. The following character classes are defined:

Character class	Permitted characters	Remove character	
A	Alphabetic characters	A-Z	No
B		Yes	
N	Numerical characters	0-9	No
O		Yes	
?	Alphanumeric characters	A-Z 0-9	No
X		Yes	

A pattern may consist of any number of character classes.

### 3.2.2.3. Samples for Postcode Pattern Validation

To verify a post code, the first pattern from the corresponding pattern list will be compared with the post code. If the length of the pattern and the length of the post code are not the same, the next pattern in the list will be compared with the post code. If the lengths are the same, each character in the post code will be compared with the corresponding character class. If the post code matches the pattern, the post code is considered to be verified and the process can be aborted. If the post code does not match the pattern, the next pattern from the pattern list will be compared with the post code.

If the entire pattern list has been compared with the post code and no match has been found, the post code is considered to be not verified and has to be rejected.

Example:

Pattern list: |NNNN,NNNNN,ANNAA,ANNNA,ANNNA,ANNNA,ANNNA|

Input Post code: AL 22-ZW

Post Code used in Pattern validation : AL22ZW

Comparison pattern 1:

AL22ZW

NNNN

Length mismatch!

Comparison pattern 2:

AL22ZW

NNNNN

Length mismatch!

Comparison pattern 3:

AL22ZW

ANNAA

Length mismatch!

Comparison pattern 4:

AL22ZW

ANNNA

Mismatch in the second character class from the left!

Comparison pattern 5:

AL22ZW

AANNAA

The post code matches the pattern.

### 3.2.2.4. How to handle removing of characters

The Postcode Pattern also contains characters, which are removed during the Pattern validation. Character classes B, O and X define which parts of the Postcodes will be removed. This guarantees that the postcode fits in the Barcode.

Example:

Pattern: NNNNNNNOO

Post code input: 123456789

Post code after pattern Validation: 1234567

## 3.2.3. Technical Background

### 3.2.3.1. Creating COUNTRY Table

```
CREATE TABLE
Country_ '$versionnumber'
(
countrycode varchar(2) NOT NULL ,
        isonumcountrycode integer (3) NOT NULL ,
        isoalpha3countrycode varchar(3) NOT NULL ,
        destinationlanguages varchar(128),
        flagpostcodeno integer NOT NULL ,
        postcodepattern varchar(128),
        flagneedstate integer NOT NULL ,
        postcodevalidation varchar(2) NOT NULL ,
        cityvalidation varchar(2) NOT NULL ,
        freetradearea varchar(2),
        dummycountrycode varchar(1),
PRIMARY KEY(CountryCode)
);
```

### 3.2.3.2. Loading COUNTRY Table

1. Before loading the table data has to be deleted.
2. Each field of COUNTRY subtype should be inserted as a new row.

### 3.2.3.3. Search Query

```
SELECT
        country_ '$versionnumber'.countrycode,
country_ '$versionnumber'.postcodepattern,
        country_ '$versionnumber'.postcodevalidation,

FROM
        country_ '$versionnumber'

WHERE
        country_ '$versionnumber'.isonumcountrycode=' $destination_country'
or
        country_ '$versionnumber'.countrycode=' $destination_country';
```

If you don't receive a match validation has failed, stop Routing Process with Country Error

### 3.3. PostcodeValidation

The field describes if Postcode validation is needed. Possible options:

NO - Skip Postcode Validation

VI - Skip Postcode Validation

VB - Perform Postcode Validation: Only continue with the Routing Process if Postcode in Postcode table exists.

#### 3.3.1.Used Table: POSTALCODE

The POSTALCODE Table contains the ranges of Postcodes existing in a Country. Fields used as input are marked in grey; fields used as output are marked in red.

Field	Length	Format	Comment
CountryCode	2	A	ISO 3166 country code – alpha 2
BeginPostCode	9	A/N	Start of postal code range
EndPostCode	9	A/N	End of the postal code range
PostCodeType	1	A	<ul style="list-style-type: none"><li>· D – Dummy zip code: Not a valid zip code, used for administrative reasons.</li><li>· P – Post box postcode: Postcode belongs to postcode delivery location</li><li>· C – CEDEX: special postcode assigned to recipients with large volumes of mail.</li></ul> Empty – it is official destination postcode

#### 3.3.2.POSTALCODE Lookup Process principles

- Only use ranges, that have the same length as the validated Post Code from Postcode Validation
- To check if a postcode is in a range postcodes are defined ascending the following logic 0-9 and then A-Z

### 3.3.3. Technical Background

#### 3.3.3.1. Creating POSTALCODE Table

```
CREATE TABLE
postalcode_ '$versionnumber'
(
    countrycode varchar(2) NOT NULL ,
    beginpostcode varchar(9) NOT NULL ,
    endpostcode varchar(9),
    postcodetype varchar(1),
PRIMARY KEY(countrycode,beginpostcode)
);
```

#### 3.3.3.2. Loading POSTALCODE Table

1. Before loading the table data has to be deleted.
2. Each field of POSTALCODE subtype should be inserted as a new row.

### 3.3.3.3. Search Query

```
SELECT
    postcode_ '$versionnumber'.beginpostcode
FROM
    postcode_ '$versionnumber'
WHERE
    postcode_ '$versionnumber'.countrycode = $Destination_Country
    AND '$Destination_Postcode' between postcode.beginpostcode and
    postcode_ '$versionnumber'.endpostcode
    AND len(postcode_ '$versionnumber'.beginpostcode) = len($Destination_Postcode);
-- The length function (len) is not part of the SQL Standard. Please check the documentation of you Database
which function you have to use.
```

If you don't receive a match validation has failed, stop Routing Process with Postcode Error

## 3.4. Validation of SoCode

This step validates the SoCode part of the Service you want to use for routing and checks if the Service requires an AsCode that has to be validated as well

### 3.4.1. Used Table: SOCODE

The SOCODE Table contains a list of all existing SoCodes.

Fields used as input are marked in grey; fields used as output are marked in red.

Field	Length	Format	Comment
SOCODE	3	N	Sorting code. Used in the 28-Digit Code 128 Barcode and in the Shipment Data
ServiceText	18	A/N	Service Text used in the routing part of the Label
ServiceMark	1	A	Mark to print on the Label for the service in the routing part of the label
ElementCodes	128	A/N	Elements used for the Code – not used in Customer Specification
ASCodeNeeded	1	A	If set to "Y" AsCode is required for this Service

## 3.4.2. Technical Background

### 3.4.2.1. Creating SOCODE Table

```
CREATE TABLE
socode_ '$versionnumber'
(
    socode integer (3) NOT NULL ,
    servicetext varchar(18) NOT NULL ,
    servicemark varchar(1) ,
    soelementcodes varchar(128) NOT NULL ,
    ascodeneeded varchar(1)
);
```

### 3.4.2.2. Loading SOCODE Table

Before loading the table data has to be deleted.

Each field of SOCODE subtype should be inserted as a new row

### 3.4.2.3. Search Query

```
SELECT
    socode_ '$versionnumber'.servicetext,
    socode_ '$versionnumber'.servicemark,
    socode_ '$versionnumber'.ascodeneeded
FROM
    socode_ '$versionnumber'
WHERE
    '$Input_socode'=socode_ '$versionnumber'.socode;
```

If you don't receive a match validation has failed, stop Routing Process with SOCODE Error

## 3.5. Validation of AsCode

This step validates the AsCode part of the Service that you might need for routing. Only needed if the SoCode requires an AsCode (see chapter 2.3.1).

The Validation is a simple lookup if the AsCode you want to use for routing exists in the ASCODE Table and selecting the ServiceText for using it on the Label.

### 3.5.1.Used Table: ASCODE

The SOCODE Table contains a list of all existing SoCodes.

Fields used as input are marked in dark grey; fields used as output are marked in red.

Field	Length	Format	Comment
ASCode	6	A/N	Additional service code. Used in the routing and in the Shipment Data
ServiceText	18	A/N	Service Text used in the routing part of the Label
ElementCodes	128	A/N	Elements used for the Code – not used in Customer Specification

### 3.5.2. Technical Background

#### 3.5.2.1. Creating ASCODE Table

```
CREATE TABLE
ascode_ '$versionnumber'
(
    ASCode varchar(6) NOT NULL ,
    ServiceText varchar(18) NOT NULL ,
    ElementCodes varchar(128) NOT NULL,
PRIMARY KEY(ASCode)
);
```

#### 3.5.2.2. Loading ASCODE Table

1. Before loading the table data has to be deleted.
2. Each field of ASCODE subtype should be inserted as a new row



### 3.5.2.3. Search Query

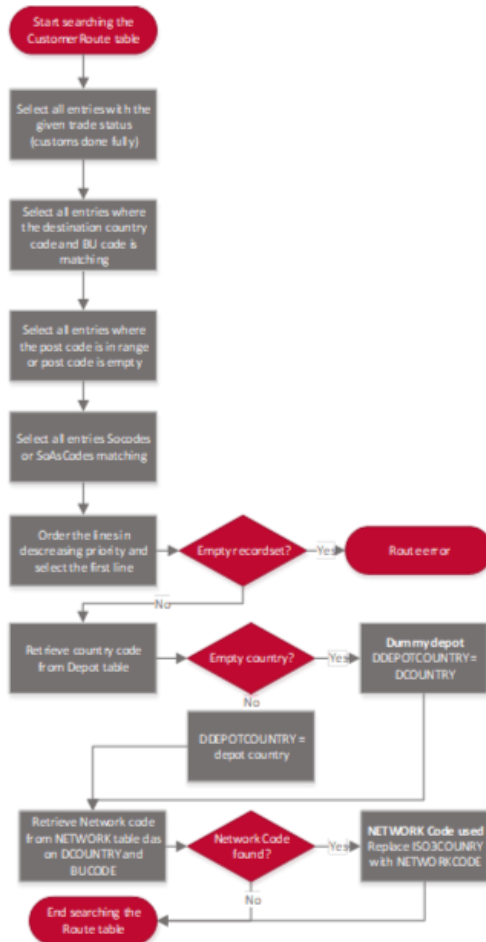
```
SELECT
    ascode_ '$versionnumber'.servicetext
FROM
    ascode_ '$versionnumber'
WHERE
    '$input_ascode'=ascode_ '$versionnumber'.ascode;;
```

If you don't receive a match validation has failed, stop Routing Process with ASCODE Error

# 4. Route Calculation

In this chapter, we will describe the process to calculate the main information for the DPD label and the shipment data. This contains the Destination Depot, the S-Sort and the D-Sort.

The routing calculation is a single lookup to the table CUSTOMERROUTE. The calculated Destination Depot has to be rechecked by the DEPOT table. For creating the proper barcode on the label this step also includes the calculation of the network code.



## 4.1. Calculate Destination

The route calculation retrieves the destination information for the shipment, which has to be printed on the shipment label and put into the preadvice data sent to DPD.

### 4.1.1.Used Table: CUSTOMERROUTE

Fields used as input are marked in grey; fields used as output are marked in grey.

Field	Length	Format	Comment
BuCode	3	N	Code of the destination business unit
DestinationCountryCode	2	A	Destination Country
TradeStatus	1	A	If the parcel is custom cleared (F - Custom cleared, C - Not cleared, empty - Default)
BeginPostCode	9	A/N	Starting digits for the postal code
EndPostCode	9	A/N	Ending number or letter for the postal code range
Socodes	2048	Text	List of Socodes definition
SoasCodes	2048	Text	List of Soascodes definition
Priority	5	N	Priority of the line to be used in the routing algorithm
DDepot	7	N	Destination depot (f.e.: 0010450)
SSort	5	A/N	Sortation code used in the routing string on the label
PartnerCode	3	A/N	Partner code in case of hand over
Dsort	5	A/N	D-Sort used in the D-Sort field on the label and in the data

### 4.1.2.Handling the Service

The Service is defined based on only the SoCode or the combination of So- and AsCode depending of the service you're using.

In the Validation process, you have checked if the SoCode you are using requires an AsCode already.

The services in the CUSTOMERROUTE table are defined in two separate columns. In the Socode column you will find a comma-separated list of the matching services only defined by SoCode, in the SoAsCodes column you will find a comma-separated list of the matching Service based on SoCode and AsCode.

SoCodes entries are defined in the format O\$\$SOCODE.

SoAsCodes entries are defined in the format O\$\$SOCODEA\$ASCODE.

In the technical background we describe the two different search queries you can use to select the proper results based on the Service.

### 4.1.3. Technical Background

You have to create and use for each version of the Routingdatabase you get from DPD this table with the appended Version number. The general process is described in chapter 2.. [update with reference when merging]. The Samples already including the versionnumber as a variable added with an added underscore in front.

#### 4.1.3.1. Creating CUSTOMERROUTE Table

```
CREATE TABLE
Customerroute_.$versionnumber
(
bucode integer (3) NOT NULL ,
        destinationcountrycode varchar (2) NOT NULL ,
        tradestatus varchar(1) ,
        beginpostcode varchar(9),
        endpostcode varchar(9) ,
        socodes varchar(2048),
        soascodes varchar(2048),
        priority integer (5) NOT NULL,
        ddepot integer (7) NOT NULL,
        ssort varchar (5) ,
        partnercode (3),
        dsort varchar (5),
);
```

#### 4.1.3.2. Loading CUSTOMERROUTE Table

1. Before loading the table data has to be deleted.
2. Each field of CUSTOMERROUTE subtype should be inserted as a new row.

#### 4.1.3.3. Search Query – Service is SOCODE

```
SELECT
    customerroute_ '$versionnumber'.bucode,
customerroute_ '$versionnumber'.ddepot,
customerroute_ '$versionnumber'.ssort,
    customerroute_ '$versionnumber'.dsort,

FROM
    Customerroute_ '$versionnumber'

WHERE

-- Select correct destination country
    Customerroute_ '$versionnumber'.destinationcountcode=' $destination_country'

-- Select correct range of postcodes
AND (
    customerroute_ '$versionnumber'.beginpostcode=" OR
    (
'$Destination_Postcode' between customerroute_ '$versionnumber'.beginpostcode and
customerroute_ '$versionnumber'.endpostcode
AND len(customerroute.beginpostcode) = len($Destination_Postcode)
-- The length function (len) is not part of the SQL Standard. Please check the documentation of you Database
which function you have to use.
    )
)

-- Select the matching Services
AND (customerroute_ '$versionnumber'.socodes =" OR customerroute_ '$versionnumber'.socodes like
' % $deliverysocode %')

ORDER BY
    CUSTOMERRROUTE_ '$VERSIONNUMBER'.priority DESC

LIMIT
    1
```

If you don't receive a match validation has failed, stop Routing Process with Routing Error

#### 4.1.3.4. Search Query – Service is SOCODE + ASCODE

```
SELECT
    customerroute_ '$versionnumber'.ddepot,
customerroute_ '$versionnumber'.ssort,
    customerroute_ '$versionnumber'.dsort,

FROM
    Customerroute_ '$versionnumber'

WHERE

-- Select correct destination country
    Customerroute_ $versionnumber.destinationcountrcode=' $destination_country'

-- Select correct range of postcodes
AND (
    customerroute_ '$versionnumber'.beginpostcode=" OR
    (
'$Destination_Postcode' between customerroute_ 'versionnumber'.beginpostcode and
customerroute_ 'versionnumber'.endpostcode
AND len(customerroute.beginpostcode) = len($Destination_Postcode)
-- The length function (len) is not part of the SQL Standard. Please check the documentation of you Database
which function you have to use.
    )
    )

-- Select the matching Services
AND (customerroute_ '$versionnumber'.soascodes =" OR customerroute_ '$versionnumber'.soascodes like
' % $deliverysoascode % ' )

ORDER BY
    CUSTOMERRROUTE_ '$VERSIONNUMBER'.priority DESC

LIMIT
    1
```

If you don't receive a match validation has failed, stop Routing Process with Routing Error

## 4.2. Adding additional information for the destination depot

To be able to create the Routing String on the label you have to extract some information from the DEPOTS table.

### 4.2.1. Used Table: DEPOTS

Fields used as input are marked in dark grey; fields used as output are marked in red.

Field	Len.	Format	Comment
GeoPostDepotNumber	7	A/N	Depot number (f.e.: 0010450)
DepotStr	4	A/N	4-digit code printed on the Shipment label
IATALikeCode	3	A/N	Compatibility reasons only
GroupID	64	A	DPD internal group ID
CompanyName	35	A/N	Depot Name
CompanyName2	35	A/N	Depot Name
Street	35	A/N	Depot address information
PropNum	8	A/N	House number
Address2	35	A/N	Depot address information
Address3	35	A/N	Depot address information
Floor	35	A/N	Depot address information
Building	35	A/N	
CountryCode	2	A	Location of the Depot. Used on the Shipment label
State	2	A/N	State where Depot is located
PostCode	9	A/N	Operational Postcode of the Depot. Used in Routing!
CityName	60	A/N	Depot address information
Phone	30	A/N	Depot contact information
Fax	30	A/N	Depot contact information
Mail	50	A/N	Depot contact information
WEB	35	A/N	Depot contact information

BUCode	3	N	Business Unit the depot is assigned to
SGPSLAT	16	A/N	Geo. Latitude in WGS84 standard
SGPSLONG	16	A/N	Geo. Longitude coordinate Number (15.11) => Separator is "." (dot) in Geodata . Use of WGS84 standard
DummyDepot	1	A	The value should be set to Y if the depot does not represent a physical depot but a logical one (like a partner or something else). Otherwise the value is N.
Addresspostcode	9	A/N	Official Postcode of Depot. Used on the label!

### 4.2.2. Technical Background

You have to create and use for each version of the Routingdatabase you get from DPD this table with the appended Version number. The general process is described in chapter 2.. [update with reference when merging]. The Samples already including the versionnumber as a variable added with an added underscore in front.



#### 4.2.2.1. Creating DEPOTS Table

CREATE TABLE

depots

(

```
geopostdepotnumber varchar(7) NOT NULL ,
    depotstr varchar(4) NOT NULL ,
    ialikecode varchar(3),
    groupid varchar(64),
    companyname varchar(35) NOT NULL ,
    companyname2 varchar(35),
    street varchar(35) ,
    propnum varchar(8),
    address2 varchar(35),
    address3 varchar(35),
    floor varchar(35),
    building varchar(35),
    countrycode varchar(2),
    state varchar(2),
    postcode varchar(9),
    cityname varchar(60),
    phone varchar(30),
    fax varchar(30),
    mail varchar(50),
    web varchar(35),
    bucode integer (3)NOT NULL ,
    sgpslat varchar(16),
    sgpslon varchar(16),
    dummydepot varchar(1) NOT NULL
    addresspostcode varchar (9)
    PRIMARY KEY (geopostdepotnumber)
```

);

#### 4.2.2.2. Loading algorithm

1. Before loading the table data has to be deleted.
2. Each field of DEPOTS subtype should be inserted into version as a new row.

#### 4.2.2.3. Search Query

SELECT

depots\_ '\$versionnumber'.depotstr,

depots\_ '\$versionnumber'.countrycode

FROM

depots\_ '\$versionnumber'

WHERE

depots\_ '\$versionnumber'.geopostdepotnumber='\$ddepot';

## 4.3. Resolving the destination Network

For operational reasons the ISO3-digit country code on the label could be replaced by the 3 digit Network code. This happens only by exception and is handled via the NETWORK table.

### 4.3.1. Used Table: NETWORK

Fields used as input are marked in grey; fields used as output are marked in red.

Field	Length	Format	Comment
NetworkCode	3	A/N	Replaces ISO3-digit country code on the label
BUCODE	3	N	
CountryCode	2	A	Country code where the network is operating
NetworkName	64	A/N	Name of the network

### 4.3.2. Technical Background

You have to create and use for each version of the Routingdatabase you get from DPD this table with the appended Version number. The general process is described in chapter 2.. [update with reference when merging]. The Samples already including the versionnumber as a variable added with an added underscore in front.

#### 4.3.2.1. Creating NETWORK Table

```
CREATE TABLE
network
(
networkcode varchar(3) NOT NULL ,
    bucode integer (3) NOT NULL ,
    countrycode varchar(2) NOT NULL ,
    networkname varchar(64) NOT NULL,
PRIMARY KEY(NetworkCode)
);
```

#### 4.3.2.2. Loading algorithm

1. Before loading the table data has to be deleted.
2. Each field of DEPOTS subtype should be inserted into version as a new row.

#### 4.3.2.3. Search Query

SELECT

network\_ '\$versionnumber'.networkcode,

FROM

network\_ '\$versionnumber'

WHERE

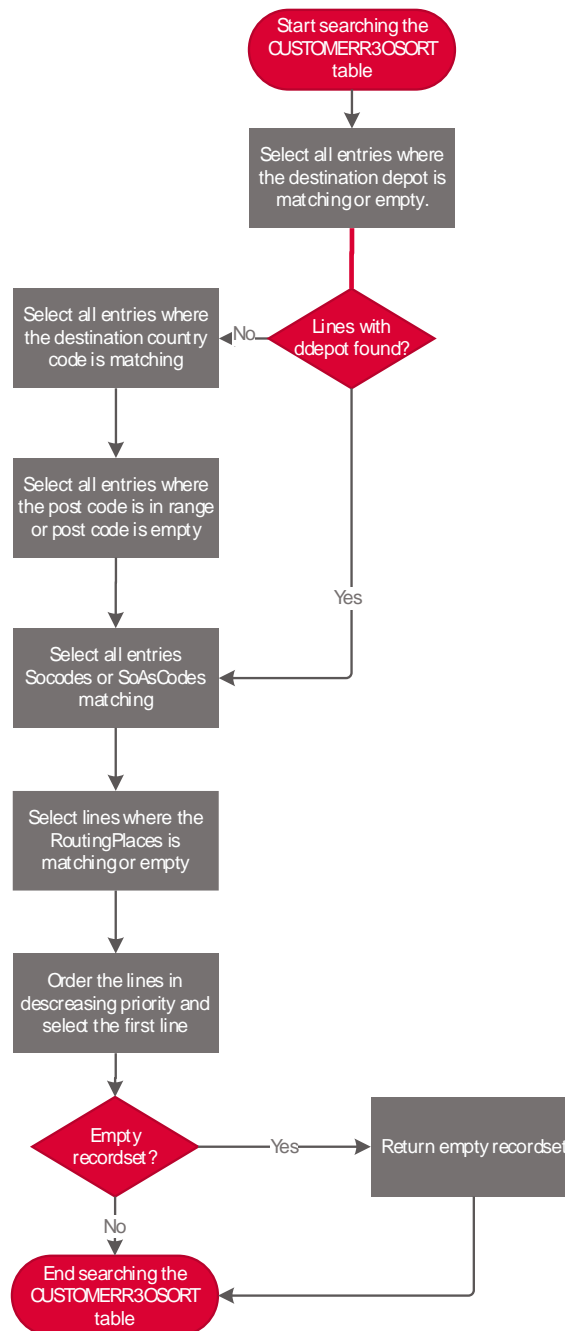
network\_ \$versionnumber.countrycode=' \$destination\_country'

AND

Network\_ \$versionnumber.bucode=\$dbu;

# 5. O-Sort calculation

The following chapter describes the CUSTOMERR3OSORT calculation process. CUSTOMERR3OSORT is the routing table which selects the origin sort based on the searching parameter fields and is mandatory for correct routing of parcels.



## 5.1. Used Table: CUSTOMERR3OSORT

Fields used as input are marked grey; fields used as output are marked in grey.

Field	Lenght	Format	Comment
RoutingPlaces	2048	TEXT	Exception routes from the default routing
DestinationCountryCode	2	A	Destination country code of the parcel
DDepot	7	N	Destination depot (ex.: 0010450) result of Routing Calculation step
BeginPostCode	9	A/N	Starting digits for the postal code
EndPostCode	9	A/N	Ending number or letter for the postal code range
Socodes	2048	Text	List of Socodes definition
SoasCodes	2048	Text	List of Soascodes definition
Priority	3	N	Priority of the line to be used in the routing algorithm
OSort	5	A/N	Origin sortation code

### 5.1.1. RoutingPlaces

The field defines a rule for locations to calculate the proper OSORT DPDgroup locations. The rule consist of a comma separated list of elements, the possibilities are:

Routing place type	Model	Syntax	Description	Example
Country code	single	Caa	a = single country code	CFR
Depot code	single	Dnnnnnnn	n = single depot code	D0010455
Business unit	single	Bnnn	n = BUCODE	B002

If the field is empty this record will be used for all routing places which do not exist in other records.

## 5.2. Technical Background

You have to create and use for each version of the Routingdatabase you get from DPD this table with the appended Version number. The general process is described in chapter 2.. [update with reference when merging]. The Samples already including the versionnumber as a variable added with an added underscore in front.

### 5.2.1. Creating CUSTOMERR3OSORT Table

```
CREATE TABLE
CUSTOMERR3OSORT_ $versionnumber
(
routingplace varchar (2048) ,
        destinationcountrycode integer (2),
        ddepot integer (7),
        beginpostcode varchar(9),
        endpostcode varchar(9) ,
        socodes varchar(2048),
        soascodes varchar(2048),
        priority integer (5) NOT NULL,
        osort varchar (5) ,

);
```

### 5.2.2. Loading CUSTOMERR3OSORT Table

1. Before loading the table data has to be deleted.
2. Each field of CUSTOMERR3OSORT subtype should be inserted as a new row.

### 5.2.3. Search Query – Step 1 Check for DDEPOT Service is SOCODE

```
SELECT
    CUSTOMERR3OSORT_ '$versionnumber'.osort,

FROM
    CUSTOMERR3OSORT_ '$versionnumber'

WHERE
-- select correct routing place
    (
CUSTOMERR3OSORT_ '$versionnumber'.routingplace like '%D$odepot%' or
CUSTOMERR3OSORT_ '$versionnumber'.routingplace like '%B$bucode%' or
CUSTOMERR3OSORT_ '$versionnumber'.routingplace like '%C$origin_country%' or
CUSTOMERR3OSORT_ '$versionnumber'.routingplace =''

-- Select correct destination depot
        CUSTOMERR3OSORT_ '$versionnumber'.ddepot='$ddepot'

-- Select the matching Services
AND (CUSTOMERR3OSORT_ '$versionnumber'.socode ='' OR CUSTOMERR3OSORT
_ '$versionnumber'.socodes like '%$deliverysocode%')

ORDER BY
    CUSTOMERR3OSORT_ '$VERSIONNUMBER'.priority DESC

LIMIT
    1
```



## 5.2.4. Search Query – Step 1 Check for DDEPOT Service is SOCODE and ASCODE

```
SELECT
    CUSTOMERR3OSORT_ '$versionnumber'.osort,

FROM
    CUSTOMERR3OSORT_ '$versionnumber'

WHERE
-- select correct routing place
    (
CUSTOMERR3OSORT_ '$versionnumber'.routingplace like '%D$odepot%' or
CUSTOMERR3OSORT_ '$versionnumber'.routingplace like '%B$bucode%' or
CUSTOMERR3OSORT_ '$versionnumber'.routingplace like '%C$origin_country%' or
CUSTOMERR3OSORT_ '$versionnumber'.routingplace =''

-- Select correct destination depot
        CUSTOMERR3OSORT_ '$versionnumber'.ddepot='$ddepot'

-- Select the matching Services
AND (CUSTOMERR3OSORT_ '$versionnumber'.soascodes ='' OR CUSTOMERR3OSORT
_ '$versionnumber'.soascodes like '%$deliverysoascode%')

ORDER BY
    R3osort_ '$VERSIONNUMBER'.priority DESC

LIMIT 1;
```

## 5.2.5. Search Query - Step 2 Check for Destination Country and Postcode Service is SOCODE

```
SELECT
    CUSTOMERR3OSORT_ '$versionnumber'.osort,

FROM
    CUSTOMERR3OSORT_ '$versionnumber'

WHERE
-- select correct routing place
    (
CUSTOMERR3OSORT_ '$versionnumber'.routingplace like '%D$odepot%' or
CUSTOMERR3OSORT_ '$versionnumber'.routingplace like '%B$bucode%' or
CUSTOMERR3OSORT_ '$versionnumber'.routingplace like '%C$origin_country%' or
CUSTOMERR3OSORT_ '$versionnumber'.routingplace =''
-- Select correct destination country
        CUSTOMERR3OSORT_ '$versionnumber'.destinationcountrcode='$destination_country'

-- Select correct range of postcodes
AND (
        CUSTOMERR3OSORT_ '$versionnumber'.beginpostcode='' OR
        (
'$Destination_Postcode' between CUSTOMERR3OSORT_ '$versionnumber'.beginpostcode and
CUSTOMERR3OSORT_ '$versionnumber'.endpostcode
AND len(CUSTOMERR3OSORT_ '$versionnumber'.beginpostcode) = len($Destination_Postcode)
-- The length function (len) is not part of the SQL Standard. Please check the documentation of you Database
which function you have to use.
        )
        )

-- Select the matching Services
AND (CUSTOMERR3OSORT_ '$versionnumber'.socodes ='' OR CUSTOMERR3OSORT
_ '$versionnumber'.socodes like '%$deliverysocode%')

ORDER BY
    CUSTOMERR3OSORT_ '$VERSIONNUMBER'.priority DESC

LIMIT
    1
```

## 5.2.6. Search Query – Step 2 Check for Destination Country and Postcode Service is SOCODE and ASCODE

```
SELECT
    CUSTOMERR3OSORT_ '$versionnumber'.osort,

FROM
    CUSTOMERR3OSORT_ '$versionnumber'

WHERE
-- select correct routing place
    (
CUSTOMERR3OSORT_ '$versionnumber'.routingplace like '%D$odepot%' or
CUSTOMERR3OSORT_ '$versionnumber'.routingplace like '%B$buocode%' or
CUSTOMERR3OSORT_ '$versionnumber'.routingplace like '%C$origin_country%' or
CUSTOMERR3OSORT_ '$versionnumber'.routingplace =''
-- Select correct destination country
        CUSTOMERR3OSORT_ '$versionnumber'.destinationcountrcode='$destination_country'

-- Select correct range of postcodes
AND (
        CUSTOMERR3OSORT_ '$versionnumber'.beginpostcode='' OR
        (
'$Destination_Postcode' between CUSTOMERR3OSORT_ '$versionnumber'.beginpostcode and
CUSTOMERR3OSORT_ '$versionnumber'.endpostcode
AND len(CUSTOMERR3OSORT_ '$versionnumber'.beginpostcode) = len('$Destination_Postcode')
-- The length function (len) is not part of the SQL Standard. Please check the documentation of you Database
which function you have to use.
        )
        )

-- Select the matching Services
AND (CUSTOMERR3OSORT_ '$versionnumber'.soascodes ='' OR CUSTOMERR3OSORT
_ '$versionnumber'.soascodes like '%$deliverysoascode%')

ORDER BY
    CUSTOMERR3OSORT_ '$VERSIONNUMBER'.priority DESC

LIMIT
    1
```

# 6. Tables only used to create Label

## 6.1. SOSERVICEINFO (by language f.e.: SOSERVICEINFO.EN)

The SOServiceinfo\_XX table contains additional Service Text, to use on the label in the respective language.

Field	Length	Format	Comment
SOCODE	3	N	
SOFieldInfo	32	A/N	Short description of the SOCODE in respective language. Could be printed on the label or used in the BUs application.

## 6.2. BUSINESSUNIT

This table contains the list of all available Business Units. You need to use it to create the Destination BU information on the Label.

Fields

BUAlphaStr	3	A	BU String printed on the label
BUCode	3	N	Unique identification of Business Unit
BUName	35	A/N	
BUDefaultCountry	2	A/N	Assigned country for a Business Unit
TypeofBU	6	A/N	Defines the type of the Business Unit – not used on your side
BUFinancialGroup	35	A/N	Defines grouping of the Business Unit – not used on your side
BUGeographGroup	35	A/N	Defines grouping of the Business Unit – not used on your side
BUTechnicalgroup	35	A/N	Defines grouping of the Business Unit – not used on your side
DPDContract	3	A/N	Defines grouping of the Business Unit – not used on your side

### DPDgroup

26 rue Guynemer - Lemnys – Bât. D  
92130 Issy-les-Moulineaux - France