| Logo of the European Commission | EUROPEAN COMMISSION  DIRECTORATE-GENERAL  TAXATION AND CUSTOMS UNION  Digital Delivery of Customs and Taxation Policies  Taxation Systems & Digital Governance |
| --- | --- |
|  |  |

**Functional Specifications**

**TIN Algorithms (Public)**

| Date: | 22/11/2024 |
| --- | --- |
| Status: | Sent for acceptance (SfA) |
| Version: | 7.00 |
| Author: | SOFT-DEV |
| Approved by: | DG TAXUD |
| Reference number: | DLV-832-6.4-114-2-2 |

|  |  |
| --- | --- |
| Public: | DG TAXUD external |

|  |  |
| --- | --- |
| Confidentiality: | Publicly available (PA) |

Table of contents

[1 Introduction 3](#_Toc183094329)

[1.1 Objective of this Document 3](#_Toc183094330)

[1.2 Applicative Issues 3](#_Toc183094331)

[1.3 Intended Audience 3](#_Toc183094332)

[1.4 Scope 3](#_Toc183094333)

[1.5 Structure of this Document 3](#_Toc183094334)

[1.6 Document Conventions 3](#_Toc183094335)

[1.7 Reference Documents 3](#_Toc183094336)

[1.8 Applicable Documents 4](#_Toc183094337)

[1.9 Abbreviations and Acronyms 4](#_Toc183094338)

[1.10 Definitions 4](#_Toc183094339)

[1.11 Versioning 4](#_Toc183094340)

[2 MS Specific Algorithms 6](#_Toc183094341)

[2.1 Austria 6](#_Toc183094342)

[2.2 Belgium 7](#_Toc183094343)

[2.3 Bulgaria 9](#_Toc183094344)

[2.4 Croatia 10](#_Toc183094345)

[2.5 Cyprus 11](#_Toc183094346)

[2.6 Czechia 12](#_Toc183094347)

[2.7 Denmark 13](#_Toc183094348)

[2.8 Estonia 15](#_Toc183094349)

[2.9 Finland 17](#_Toc183094350)

[2.10 France 19](#_Toc183094351)

[2.11 Germany 19](#_Toc183094352)

[2.12 Greece 22](#_Toc183094353)

[2.13 Hungary 22](#_Toc183094354)

[2.14 Ireland 23](#_Toc183094355)

[2.15 Italy 23](#_Toc183094356)

[2.16 Latvia 26](#_Toc183094357)

[2.17 Lithuania 26](#_Toc183094358)

[2.18 Luxembourg 28](#_Toc183094359)

[2.19 Malta 31](#_Toc183094360)

[2.20 Netherlands 32](#_Toc183094361)

[2.21 Poland 33](#_Toc183094362)

[2.22 Portugal 35](#_Toc183094363)

[2.23 Romania 36](#_Toc183094364)

[2.24 Slovakia 38](#_Toc183094365)

[2.25 Slovenia 39](#_Toc183094366)

[2.26 Spain 40](#_Toc183094367)

[2.27 Sweden 42](#_Toc183094368)

[2.28 United Kingdom 46](#_Toc183094369)

# Introduction

## Objective of this Document

The aim of this document is to provide the Tax Identification Number (TIN) algorithms used to check the validity of a TIN depending on a Member State (MS).

The Functional Specification (FS) are intentionally presented in a way that is appropriate for any technical solution.

## Applicative Issues

It is essential that the implementation of these algorithms does not allow their content to be deciphered by a process of reverse engineering of any application.

## Intended Audience

This document is intended for:

* The Directorate-General Taxation and Customs Union (DG TAXUD);
* The Members States Administration (MSA).

## Scope

TIN-on-the-Web (ToW) must provide any person that has a Web access with the ability to check the validity of a Tax Identification Number (TIN).

The end-user specifies the TIN. The validation is done by checking the syntax correctness of the TIN, depending on the MS algorithm(s).

The goal of this document is to fully specify the MS algorithms to be used by the TIN validation process of ToW.

## Structure of this Document

Chapter 1 **Introduction:** introduces the purpose and the structure of this document;

Chapter 2 **MS Specific Algorithms:** enumerates and defines all the algorithms of the MSs who are willing to share this information.

## Document Conventions

Reference documents are shown in square brackets [].

## Reference Documents

| **Ref.** | **Title** | **Reference** | **Version** | **Date** |
| --- | --- | --- | --- | --- |
| R01 | ToW - Functional Specification | ToW-FS | 2.30 | 22/11/2024 |
| R02 | VAT Number Construction Rules Functional Description | VIES-VAT Validation Routines-v18.0 | 18.0.01 | 15/05/2017 |

Table 1: Reference Documents

## Applicable Documents

| **Ref.** | **Title** | **Reference** | **Version** | **Date** |
| --- | --- | --- | --- | --- |
| A01 | Framework Contract | TAXUD/2021/CC/162 | N/A | 24/06/2021 |
| A02 | Specific Contract n°20 | TAXUD/2024/DE/122 | N/A | 23/04/2024 |

Table 2: Applicable Documents

## Abbreviations and Acronyms

| **Acronym** | **Meaning** |
| --- | --- |
| ASCII | American Standard Code for Information Interchange |
| DG TAXUD | Directorate-General Taxation and Customs Union |
| EU | European Union |
| FS | Functional Specification |
| MS | Member State |
| MSA | Member State Administration |
| TIN | Tax Identification Number |
| ToW | TIN-on-the-Web |
| TS | Technical Specification |
| VAT | Value Added Tax |
| VIES | VAT Information Exchange System |

Table 3: Abbreviations and Acronyms

## Definitions

| **Definition** | **Meaning** |
| --- | --- |
| Character | Can be a letter, a numeric, “+” or “-“. |
| Letter | Character in the range [A-Z] and [a-z]. |
| Numeric | Character in the range [0-9]. |

Table 4: Definitions

## Versioning

The versioning of this document is based on major and minor numbers in the form *x.y*. These two numbers are synchronized with the two first numbers of the TIN algorithms JAR version. Whenever a functional change is needed in an algorithm, and the new version of the algorithm becomes less restrictive than the previous one, the major number of this document is incremented. While an algorithm needs some clarifications, but without requiring any modification in the implementations, or when an algorithm becomes more restrictive the minor number of this document is incremented.

In addition, a third number zis added at the end in case of documentation issue, which does not change any algorithm except for clarification. This number is free and therefore not synchronized with the TIN algorithms JAR version.

For example, if a country decides to extend the length of all its TIN by prefixing them by one character, while still authorising the previous numbers the algorithm become less restrictive. This means that countries that are still using the previous version of the algorithm will "block" TINs that are valid according to the new rules. In such a case, the major number of this document release number will be incremented.

# MS Specific Algorithms

**Important preliminary remark:** when querying a TIN, the user should not be requested to ask which algorithm is applicable but (s)he must be able to just put the TIN and the check module will validate the TIN against the algorithm(s); if one of the algorithm(s) is correct, then the result must be that the TIN is valid.

## Austria

|  |  |  |
| --- | --- | --- |
| **Structure: TIN Format** | [C1, C2,"-", C3, C4, C5, “/”, C6, C7, C8, C9] | Where C1 to C9 are characters and C1 and C2 are separated from the rest by a hyphen and C3 to C5 are separated from C6 to C9 by a slash.  Note: the hyphen and the slash are not mandatory in all cases (e.g. for IT processing, hyphen and slash should be omitted). |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C7, C8, C9 | A numeric. |
| **Structure: Special Characters** | If any, special characters (dash, slash or other signs) should be skipped. | |

|  |  |  |
| --- | --- | --- |
| **Syntax: Check Digit** | C9 | 1. Multiply the values of each position by the corresponding weight:   C1 1  C2 2  C3 1  C4 2  C5 1  C6 2  C7 1  C8 2   1. If the product of a doubling operation is > 9, sum the digits of the product; 2. Add up the results of the above multiplications; 3. The result of the sum of the digits is subtracted from 100 and the unit digit of this operation is the check digit. |
| **Syntax: Sample** | 931736581 | 1. 9 \* 1 = 9, 3 \* 2 = 6, 1 \* 1 = 1, 7 \* 2 = 14, 3 \* 1 = 3, 6 \* 2 = 12, 5 \* 1 = 5, 8 \* 2 = 16; 2. 1 + 4 = 5, 1 + 2 = 3, 1 + 6 = 7; 3. 9 + 6 + 1 + 5 + 3 + 3 + 5 + 7 = 39; 4. 100 – 39 = 61, 1 is the check digit. |

Table 5: MS Specific Algorithms - Austria 2

## Belgium

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11] | | Where C1 to C11 are characters. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11 | | A numeric. |
| **Structure: Special Characters** | If any, special characters (dash, slash or other signs) should be skipped. | | |
| **Structure: Rules** | C1, C2 | Two last digits of a year. | |
| C3, C4 | A month (in the range 00...12, 00 is acceptable for person not born in Belgium and with an uncertain date of birth). | |
| C5, C6 | A day of month (in the range 00...31 depending on month and year, 00 is acceptable for person not born in Belgium and with an uncertain date of birth). | |

|  |  |  |
| --- | --- | --- |
| **Syntax: Check Digit** | C10, C11 | 1. Get the remainder of the division by 97 of the number composed by C1, C2, C3, C4, C5, C6, C7, C8 and C9; 2. 97 - remainder of the previous division is the check number. |
| **Syntax: Sample** | 00012511119 (person born 25/01/1900) | 1. 125111 MOD 97 = 78; 2. Check digit = 97 - 78 = 19. |

Table 6: MS Specific Algorithms - Belgium 1

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11] | | Where C1 to C11 are characters. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11 | | A numeric. |
| **Structure: Special Characters** | If any, special characters (dash, slash or other signs) should be skipped. | | |
| **Structure: Rules** | C1, C2 | Two last digits of a year. | |
| C3, C4 | A month (in the range 00...12, 00 is acceptable for person not born in Belgium and with an uncertain date of birth). | |
| C5, C6 | A day of month (in the range 00...31 depending on month and year, 00 is acceptable for person not born in Belgium and with an uncertain date of birth). | |

|  |  |  |
| --- | --- | --- |
| **Syntax: Check Digit** | C10, C11 | 1. Get the remainder of the division by 97 of the number composed by number 2 and C1, C2, C3, C4, C5, C6, C7, C8 and C9; 2. 97 - remainder of the previous division is the check number. |
| **Syntax: Sample** | 00012511148(person born 25/01/2000) | 1. 2000125111 MOD 97 = 49; 2. Check digit = 97 - 49 = 48. |

Table 7: MS Specific Algorithms - Belgium 2

## Bulgaria

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, C7, C8, C9, C10] | | Where C1 to C10 are characters. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C7, C8, C9, C10 | | A numeric. |
| **Structure: Rules** | C1, C2 | Two last digits of a year. | |
| C3, C4 | A month (in the range 1...12, 21...32, 41...52):   * Range 21...32: add 20 if the date of birth is < 01/01/1900; * Range 41...52: add 40 if the date of birth is > 31/12/1999). | |
| C5, C6 | Day of month (in the range 1...31). | |

|  |  |  |
| --- | --- | --- |
| **Syntax: Check Digit** | C10 | 1. Multiply the values of each position by the corresponding weight:   C1 2  C2 4  C3 8  C4 5  C5 10  C6 9  C7 7  C8 3  C9 6   1. Add up the results of the above multiplications; 2. Get modulo 11 of the result of the previous addition; 3. Check digit = remainder if remainder < 10 Check digit = 0 if remainder = 10. |
| **Syntax: Sample** | 7501010010 | 1. 7 \* 2 = 14, 5 \* 4 = 20, 0 \* 8 = 0, 1 \* 5 = 5, 0 \* 10 = 0, 1 \* 9 = 9, 0 \* 7 = 0, 0 \* 3 = 0, 1 \* 6 = 6; 2. 14 + 20 + 0 + 5 + 0 + 9 + 0 + 0 + 6 = 54; 3. 54 MOD 11 = 10; 4. Check digit = 0. |

Table 8: MS Specific Algorithms – Bulgaria

## Croatia

|  |  |  |
| --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11] | Where C1 to C11 are characters. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11 | A numeric. |
| **Structure: Rules** | C1, C2, C3, C4, C5, C6, C7, C8, C9, C10 | Random number. |
| C11 | Check digit by the international standard ISO 7064 (MOD 11, 10). |

|  |  |  |
| --- | --- | --- |
| **Syntax: Check Digit** | C11 | 1. C1 is summed with 10; 2. The sum integer is divided by 10, and the rest is kept; if that number is 0 it gets replaced by number 10 (this latter number is called subtotal); 3. The obtained subtotal is multiplied by 2; 4. The obtained number is divided by 11, and the rest is kept; this number mathematically cannot be 0 because the result of the previous step is always an even number; 5. The next digit is summed with the rest from the previous step; 6. Steps 2, 3, 4 and 5 are repeated until all digits are expanded; 7. If the rest of the final step is equal to 1, the check digit is 0. Otherwise, the check digit is a difference between 11 and the rest in the last step. |
| **Syntax: Sample** | 94577403194 | C1 to C10 are 9457740319   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Step** | **Digit** | **Sum** | **Subtotal** | **Multiplication** | **Rest** | | 1 | 9 | 9+10=19 | 9 | 9\*2=18 | 7 | | 2 | 4 | 4+7=11 | 1 | 1\*2=2 | 2 | | 3 | 5 | 5+2=7 | 7 | 7\*2=14 | 3 | | 4 | 7 | 7+3=10 | 10 | 10\*2=20 | 9 | | 5 | 7 | 7+9=16 | 6 | 6\*2=12 | 1 | | 6 | 4 | 4+1=5 | 5 | 5\*2=10 | 10 | | 7 | 0 | 0+10=10 | 10 | 10\*2=20 | 9 | | 8 | 3 | 3+9=12 | 2 | 2\*2=4 | 4 | | 9 | 1 | 1+4=5 | 5 | 5\*2=10 | 10 | | 10 | 9 | 9+10=19 | 9 | 9\*2=18 | 7 |   11-7=4, the check digit is 4. The TIN is correct. |

Table 9: MS Specific Algorithms – Croatia

## Cyprus

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, C7, C8, C9] | | Where C1 to C9 are characters. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C7, C8 | | A numeric. |
| C9 | | A capital letter. |
| **Structure: Rules** | C1 | 0, 6 or 9 for individuals. | |

|  |  |  |
| --- | --- | --- |
| **Syntax: Check Digit** | C9 | 1. Add up the numbers in the even positions; 2. Consider all the numbers at the odd positions of the field and for each number find the corresponding value from the table below, and add them up:   0 1  1 0  2 5  3 7  4 9  5 13  6 15  7 17  8 19  9 21   1. Add the two sums obtained; 2. Get modulo 26 of the result of the previous addition; 3. Remainder + 65 gives the American Standard Code for Information Interchange (ASCII) code of a character (A to Z) which is the check character. |
| **Syntax: Sample** | 00123123T | 1. 0 + 2 + 1 + 3 = 6; 2. 1 + 0 + 7+ 5 = 13; 3. 6 + 13 = 19; 4. 19 MOD 26 = 19; 5. 19 + 65 = 84 = T. |
| 60123123H | 1. 0 + 2 + 1 + 3 = 6; 2. 15 + 0 + 7+ 5 = 27; 3. 6 + 27 = 33; 4. 33 MOD 26 = 7; 5. 7+ 65 = 72 = H. |
| 99652156X | 1. 9 + 5 + 1 + 6 = 21; 2. 21 + 15 + 5 + 13 = 54; 3. 21 + 54 = 75; 4. 75 MOD 26 = 23; 5. 23 + 65 = 88 = X. |

Table 10: MS Specific Algorithms - Cyprus

## Czechia

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, ["/"], C7, C8, C9] | | Where C1 to C9 are characters.  YYMMDD999 - 1 block of 9 digits issued for people born till 31.12.1953. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C7, C8, C9 | | A numeric. |
| **Structure: Special Characters** | If any, special characters (slash) between C6 and C7 should be skipped; it is purely optional. | | |
| **Structure: Rules** | C1, C2 | Two last digits of a year. | |
| C3, C4 | Month (in the range 1...12 for male) or month + 50 (in the range 51…62 for female). | |
| C5, C6 | Day of month (in the range 1...31 depending on month and year). | |

|  |  |
| --- | --- |
| **Syntax: Check Digit** | Not publicly available |
| **Syntax: Sample** | Not publicly available |

Table 11: MS Specific Algorithms – Czechia 1

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, ["/"], C7, C8, C9, C10] | | Where C1 to C10 are characters.  YYMMDD999C - 1 block of 10 digits issued for people born after 1.1.1954. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C7, C8, C9, C10 | | A numeric. |
| **Structure: Special Characters** | If any, special characters (slash) should be skipped. | | |
| **Structure: Rules** | C1, C2 | Two last digits of a year: Must be within the range:   * 00 - last two digits of current year for people born in 2000 and later; * 54 - 99 for people born between 1954 and 1999. | |
| C3, C4 | Month (in the range 1...12 only for male) or month + 20 (in the range 21…32 only for male) or month + 50 (in the range 51...62 only for female) or month + 70 (in the range 71…82 only for female). | |
| C5, C6 | Day of month (in the range 1...31 depending on month and year). | |

|  |  |
| --- | --- |
| **Syntax: Check Digit** | Not publicly available |
| **Syntax: Sample** | Not publicly available |

Table 12: MS Specific Algorithms – Czechia 2

## Denmark

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, “-“, C7, C8, C9, C10] | | Where C1 to C10 are characters.  C1 to C6 are separated from C7 to C10 by a hyphen. This hyphen is optional and should be skipped in the validation. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C7, C8, C9, C10 | | A numeric. |
| **Structure: Rules** | C1, C2 | Day of month (in the range 1...31 depending on month and year). | |
| C3, C4 | Month (in the range 1...12). | |
| C5, C6 | Two last digits of a year. | |

|  |  |  |
| --- | --- | --- |
| **Syntax: Check Digit** | C10 | 1. Multiply the values of each position by the corresponding weight:   C1 4  C2 3  C3 2  C4 7  C5 6  C6 5  C7 4  C8 3  C9 2   1. Add up the results of the above multiplications; 2. Get modulo 11 of the result of the previous addition. The remainder must not be 1; 3. Check digit = 11 – remainder, or check digit = 0 if the result of the modulo operation of the third step is 0. |
| **Syntax: Additional Rules on Check Digit** | Several ranges of figures are not possible and should be reported as erroneous if a TIN is checked in those ranges: | |
| **Syntax: Sample** | 010111-1113 | 1. 0 \* 4 = 0, 1 \* 3 = 3 ,0 \* 2 = 0, 1 \* 7 = 7, 1 \* 6 = 6, 1 \* 5 = 5, 1 \* 4 = 4, 1 \* 3 = 3, 1 \* 2 = 2; 2. 0 + 3 + 0 + 7 + 6 + 5 + 4 + 3 + 2 = 30; 3. 30 MOD 11 = 8; 4. Check digit = 11 - 8 = 3. |

Table 13: MS Specific Algorithms - Denmark

## Estonia

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11] | | Where C1 to C11 are characters. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11 | | A numeric. |
| **Structure: Rules** | C1 | In the range 1…6. | |
| C2, C3 | Two last digits of a year. | |
| C4, C5 | Month (in the range 1...12). | |
| C6, C7 | Day of month (in the range 1...31 depending on month and year). | |
| C8, C9, C10 | In the range 001...710. | |

|  |  |  |
| --- | --- | --- |
| **Syntax: Check Digit** | C11 | 1. Multiply the values of each position by the corresponding weight:   C1 1  C2 2  C3 3  C4 4  C5 5  C6 6  C7 7  C8 8  C9 9  C10 1   1. Add up the results of the above multiplications; 2. Get modulo 11 of the result of the previous addition; 3. Check digit:    1. If remainder is less than 10, the remainder is the check digit;    2. If remainder is 10, use the following table instead of the previous one:   C1 3  C2 4  C3 5  C4 6  C5 7  C6 8  C7 9  C8 1  C9 2  C10 3   * If remainder is less than 10, the remainder is the check digit; * If remainder is 10, the check digit is 0. |
| **Syntax: Sample** | 37102250382 | 1. 3 \* 1 = 3, 7 \* 2 = 14, 1 \* 3 = 3, 0 \* 4 = 0, 2 \* 5 = 10, 2 \* 6 = 12, 5 \* 7 = 35, 0 \* 8 = 0, 3 \* 9 = 27, 8 \* 1 = 8; 2. 3 + 14 + 3 + 0 + 10 + 12 + 35 + 0 + 27 + 8 = 112; 3. Check digit: 112 MOD 11 = 2. |
| 32708101201 | 2. 3 + 4 + 21 + 0 + 40 + 6 + 0 + 8 + 18 + 0 = 100; 3. Check digit: 100 MOD 11 = 1. |
| 46304280206 | 2. 4 + 12 + 9 + 0 + 20 + 12 + 56 + 0 + 18 + 0 = 131; 3. 131 MOD 11 = 10; 5. 12 + 24 + 15 + 0 + 28 + 16 + 72 + 0 + 4 + 0 = 171; 6. Check digit: 171 MOD 11 = 6. |

Table 14: MS Specific Algorithms - Estonia

## Finland

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11] | | Where C1 to C11 are characters. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C8, C9, C10 | | A numeric. |
| C7 | | A letter between A and F or between U and Y or a “+” or “-“ character (see below). |
| C11 | | A numeric or a letter (see below). |
| **Structure: Rules** | C1, C2 | Day of month (in the range 1...31 depending on month and year). | |
| C3, C4 | Month (in the range 1...12). | |
| C5, C6 | Two last digits of a year. | |
| C7 | + , -, a letter between A and F, or a letter between U and Y. Note:   * "+" (plus) means year of birth between 1800 and 1899; * "-" (minus) or U or V or W or X or Y means year of birth between 1900 and 1999; * A or B or C or D or E or F means year of birth in 2000 and above.   The structure check should absolutely ensure that one of the 3 characters is included in the TIN. | |

|  |  |  |
| --- | --- | --- |
| **Syntax: Check Digit** | C11 | 1. Concatenate C1, C2, C3, C4, C5, C6, C8, C9, C10 (warning: C7 is not part of the check digit); 2. Calculate the modulo 31 of the abovementioned number; 3. The result of calculating modulo 31 will give as a result a number, which will provide the check mark through the following table:   0 0  1 1  2 2  3 3  4 4  5 5  6 6  7 7  8 8  9 9  10 A  11 B  12 C  13 D  14 E  15 F  16 H  17 J  18 K  19 L  20 M  21 N  22 P  23 R  24 S  25 T  26 U  27 V  28 W  29 X  30 Y |
| **Syntax: Sample** | 131052-308T | 1. 131052308; 2. 131052308 MOD 31 = 25; 3. Check character = T. |

Table 15: MS Specific Algorithms - Finland

## France

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13] | | Where C1 to C13 are characters. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C8, C9, C10, C11, C12, C13 | | A numeric. |
| **Structure: Rules** | C1 | Must be 0, 1, 2 or 3. | |

|  |  |  |
| --- | --- | --- |
| **Syntax: Check Digit** | C11, C12, C13 | 1. Concatenate C1, C2, C3, C4, C5, C6, C7, C8, C9, C10; 2. Get modulo 511 of the result of the previous result; 3. Check digit = remainder if remainder < 100, C11 = 0 (if remainder < 10, C11 = 0 and C12 = 0). |
| **Syntax: Sample** | 30 23 217 600 053 | 1. 3023217600; 2. 3023217600 mod 511 = 53; 3. Check digit = 053. |

Table 16: MS Specific Algorithms - France

## Germany

|  |  |  |
| --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13] | Where C1 to C13 are characters. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13 | A numeric. |
| **Structure: Special Characters** | If any, slash should be skipped. | |

|  |  |  |
| --- | --- | --- |
| **Syntax: Check Digit** | C5 | 1. Must always be 0. |
| **Syntax: Sample** | 5133081508159 | 1. 13 characters 2. Only digits 3. C5 is a 0 |

Table 17: MS Specific Algorithms - Germany 1

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11] | | Where C1 to C11 are characters. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11 | | A numeric. |
| **Structure: Special Characters** | If any, slash should be skipped. | | |
| **Structure: Rules** | C1 | Must never be 0. | |
| C1, C2, C3, C4, C5, C6, C7, C8, C9, C10 | One and only one mandatory duplicate or triple value:   * One of the first ten digits is used twice (the recurrent digits do not have to be located at subsequent positions but they can be); * One of the first ten digits is used tree times (only two recurrent digits are allowed to be one after another). | |

|  |  |  |
| --- | --- | --- |
| **Syntax: Check Digit** | C11 | 1. Initialize the variable X to 10. 2. Take C1 + X modulo 10. If result is 0, result is 10; 3. Multiply the result by 2; 4. Take modulo 11 of the result. Update the value of variable X with the result of this operation; 5. Take C2 + X modulo 10. If result is 0, result is 10; 6. Multiply the result by 2; 7. Take modulo 11 of the result. Update the value of variable X with the result of this operation; 8. Apply steps 5, 6 and 7 in an analogue way for digits C3 to C10. Consider that last value called Y; 9. 11 - Y = check digit. If check digit = 10, replace it by 0. |
| **Syntax: Sample** | 86095742719 | 1. (8 + 10) MOD 10 = 8; 2. 8 \* 2 = 16; 3. 16 MOD 11 = 5; 4. (6 + 5) MOD 10 = 1; 5. 1 \* 2 = 2; 6. 2 MOD 11 = 2; 7. (0 + 2) MOD 10 = 2; 8. 2 \* 2 = 4; 9. 4 MOD 11 = 4; 10. (9 + 4) MOD 10 = 3; 11. 3 \* 2 = 6; 12. 6 MOD 11 = 6; 13. (5 + 6) MOD 10 = 1; 14. 1 \* 2 = 2; 15. 2 MOD 11 = 2; 16. (7 + 2) MOD 10 = 9; 17. 9 \* 2 = 18; 18. 18 MOD 11 = 7; 19. (4 + 7) MOD 10 = 1; 20. 1 \* 2 = 2; 21. 2 MOD 11 = 2; 22. (2 + 2) MOD 10 = 4; 23. 4 \* 2 = 8; 24. 8 MOD 11 = 8; 25. (7 + 8) MOD 10 = 5; 26. 5 \* 2 = 10; 27. 10 MOD 11 = 10; 28. (1 + 10) MOD 10 = 1; 29. 2 \* 1 = 2; 30. 2 MOD 11 = 2; 31. Check digit = 11 - 2 = 9. |
| 65929970489 | 1. (6 + 10) MOD 10 = 6; 2. 6 \* 2 = 12; 3. 12 MOD 11 = 1; 4. (5 + 1) MOD 10 = 6; 5. 6 \* 2 = 12; 6. 12 MOD 11 = 1; 7. (9 + 1) MOD 10 = 0 replaced by 10; 8. 10 \* 2 = 20; 9. 20 MOD 11 = 9; 10. (2 + 9) MOD 10 = 1; 11. 1 \* 2 = 2; 12. 2 MOD 11 = 2; 13. (9 + 2) MOD 10 = 1; 14. 1 \* 2 = 2; 15. 2 MOD 11 = 2; 16. (9 + 2) MOD 10 = 1; 17. 1 \* 2 = 2; 18. 2 MOD 11 = 2; 19. (7 + 2) MOD 10 = 9; 20. 9 \* 2 = 18; 21. 18 MOD 11 = 7; 22. (0 + 7) MOD 10 = 7; 23. 7 \* 2 = 14; 24. 14 MOD 11 = 3; 25. (4 + 3) MOD 10 = 7; 26. 7 \* 2 = 14; 27. 14 MOD 11 = 3; 28. (8 + 3) MOD 10 = 1; 29. 2 \* 1 = 2; 30. 2 MOD 11 = 2; 31. Check digit = 11 - 2 = 9. |

Table 18: MS Specific Algorithms - Germany 2

## Greece

|  |  |  |
| --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, C7, C8, C9] | Where C1 to C9 are characters. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C7, C8, C9 | A numeric. |

|  |  |
| --- | --- |
| **Syntax: Check Digit** | Not publicly available |
| **Syntax: Sample** | Not publicly available |

Table 19: MS Specific Algorithms - Greece

## Hungary

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, C7, C8, C9, C10] | | Where C1 to C10 are characters. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C7, C8, C9, C10 | | A numeric. |
| **Structure: Rules** | C1 | 8. | |

|  |  |  |
| --- | --- | --- |
| **Syntax: Check Digit** | C10 | 1. Multiply the values of each position by the corresponding weight:   C1 1  C2 2  C3 3  C4 4  C5 5  C6 6  C7 7  C8 8  C9 9   1. Add up the results of the above multiplications; 2. Get modulo 11 of the result of the previous addition; 3. Check digit = remainder. |
| **Syntax: Sample** | 8071592153 | 1. 8 \* 1 = 8, 0 \* 2 = 0, 7 \* 3 = 21, 1 \* 4 = 4, 5 \* 5 = 25, 9 \* 6 = 54, 2 \* 7 = 14, 1 \* 8 = 8, 5 \* 9 = 45; 2. 8 + 0 + 21 + 4 + 25 + 54 + 14 + 8 + 45 = 179; 3. 179 MOD 11 = 3; 4. Check digit = 3. |

Table 20: MS Specific Algorithms - Hungary

## Ireland

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, C7, C8, C9] | | Where C1 to C9 are characters. C9 is optional. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C7 | | A numeric. |
| C8 | | A letter in the range [A-W]. |
| **Structure: Rule** | C9 | A letter in the range [A-I], or “W”. | |

|  |  |  |
| --- | --- | --- |
| **Syntax: Check Digit** | 1. In reverse order, each digit is multiplied by a weight started at 2:   C7 2  C6 3  C5 4  C4 5  C3 6  C2 7  C1 8  LetterToNumber(C9) 9   1. LetterToNumber(C9) is based on the following mapping: “A”=1, “B”=2, “C”=3, “D”=4, “E”=5, “F”=6, “G”=7, “H”=8, “I”=9 A “W” or absence of character in position 9 is allocated a numeric value of 0. 2. Add up each result; 3. The remainder of the modulo 23 indicates the character position on the alphabet according to the following mapping: 0=”W”, 1=”A”, 2=”B”, 3=”C”… 22=”V” | |
| **Syntax: Samples** | 1234567T | 1. 0 \* 9 + 1 \* 8 + 2 \* 7 + 3 \* 6 + 4 \* 5 + 5 \* 4 + 6 \* 3 + 7 \* 2 = 112; 2. 112 MOD 23 = 20 = T. |
| 1234567TW | 1. 0 \* 9 + 1 \* 8 + 2 \* 7 + 3 \* 6 + 4 \* 5 + 5 \* 4 + 6 \* 3 + 7 \* 2 = 112; 2. 112 MOD 23 = 20 = T. |
| 1234577W | 1. 0 \* 9 + 1 \* 8 + 2 \* 7 + 3 \* 6 + 4 \* 5 + 5 \* 4 + 7 \* 3 + 7 \* 2 = 115; 2. 112 MOD 23 = 0 = W. |
| 1234577WW | 1. 0 \* 9 + 1 \* 8 + 2 \* 7 + 3 \* 6 + 4 \* 5 + 5 \* 4 + 7 \* 3 + 7 \* 2 = 115; 2. 112 MOD 23 = 0 = W. |
| 1234577IA | 1. 1 \* 9 + 1 \* 8 + 2 \* 7 + 3 \* 6 + 4 \* 5 + 5 \* 4 + 7 \* 3 + 7 \* 2 = 124; 2. 124 MOD 23 = 9 = I |

Table 21: MS Specific Algorithms - Ireland

## Italy

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13, C14, C15, C16] | | Where C1 to C16 are characters. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C9, C12, C16 | | A letter (uppercase or lowercase). |
| C7, C8, C10, C11, C13, C14, C15 | | A numeric. |
| **Structure: Rules** | C7, C8 | Two last digits of a year. | |
| C9 | A letter representing a month; the letter can only take the values:   * January: A * February: B * March: C * April: D * May: E * June: H * July: L * August: M * September: P * October: R * November: S * December: T. | |
| C10, C11 | Day of month (in the range 1...31 for men) or day of month + 40 (in the range 41...71 for women). | |
| C12, C13, C14, C15 | A code (1 letter + 3 numeric) representing the place of birth, being a city (when born in Italy) or a country (when born out of Italy), from the following list of values: | |
| C7, C8, C10, C11, C13, C14, C15 | When two or more individuals have the same fifteen alphanumeric character, one or more of the seven numerical characters are replaced, starting from the right, according to correspondence shown in the following table: | |

|  |  |  |
| --- | --- | --- |
| **Syntax: Check Digit** | 1. Each of the first fifteen characters, depending on its relevant position (even or odd), is converted into a numeric value, according to correspondence shown in the tables below:      1. The numerical values thus determined are added together and their sum is divided by 26. The check character (C16) is obtained by converting the remainder of the division in the corresponding alphabetic character according to the table below: | |
| **Syntax: Sample** | DMLPRY77D15H501F |  |

Table 22: MS Specific Algorithms - Italy

## Latvia

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11] | | Where C1 to C11 are characters. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11 | | A numeric. |
| **Structure: Rules** | C1, C2 | Day of month (in the range 1...31 depending on month and year. | |
| C3, C4 | Month (in the range 00...12). | |
| C5, C6 | Two last digits of a year. | |
| C7 | Digit indicating the century:   * 0: person born between 1800 and 1899; * 1: person born between 1900 and 1999; * 2: person born from 2000 or above. | |
| **Structure: Sample** | 01011012345. | | |

|  |  |
| --- | --- |
| **Syntax: Check Digit** | Not publicly available |
| **Syntax: Sample** | Not publicly available |

Table 23: MS Specific Algorithms - Latvia 1

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11] | | Where C1 to C11 are characters. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11 | | A numeric. |
| **Structure: Rules** | C1 | Digit “3” | |
| C2 | Digit “2” | |
| C3, C4, C5, C6, C7, C8, C9, C10 | Digit from 0 to 9, generated automatically by information system. | |
| **Structure: Sample** | 32579461005 | | |

|  |  |
| --- | --- |
| **Syntax: Check Digit** | Not publicly available |
| **Syntax: Sample** | Not publicly available |

Table 24: MS Specific Algorithms - Latvia 2[[1]](#footnote-2)

## Lithuania

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11] | | Where C1 to C11 are characters. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11 | | A numeric. |
| **Structure: Rules** | C1 | Can only take the value 1 to 6. | |
| C2, C3 | Two last digits of a year. | |
| C4, C5 | Month (in the range 1...12). | |
| C6, C7 | Day of month (in the range 1...31 depending on month and year). | |
| **Structure: Sample** | 10101010005. | | |
| **yntax: Check Digit** | C11 | 1. Multiply the values of each position by the corresponding weight:   C1 1  C2 2  C3 3  C4 4  C5 5  C6 6  C7 7  C8 8  C9 9  C10 1   1. Add up the results of the above multiplications; 2. Get modulo 11 of the result of the previous addition; 3. C11 = remainder if remainder is not 10; 4. If remainder is 10, calculate a new check digit with over corresponding weight:   C1 3  C2 4  C3 5  C4 6  C5 7  C6 8  C7 9  C8 1  C9 2  C10 3   1. Add up the results of the above multiplications; 2. Get modulo 11 of the result of the previous addition; 3. C11 = remainder if remainder is not 10; if remainder is 10, C11 = 0. | |
| **Syntax: Sample** | 33309240064 | 1. 3\*1=3; 3\*2=6; 3\*3=9; 0\*4=0; 9\*5=45; 2\*6=12; 4\*7=28; 0\*8=0; 0\*9=0; 6\*1=6 2. 3 + 6 + 9 + 0 + 45 + 12 + 28 + 0 + 0 + 6 = 109 3. 109 MOD 11 = 10 4. 3\*3=9; 3\*4=12; 3\*5=15; 0\*6=0; 9\*7=63; 2\*8=16; 4\*9=36; 0\*1=0; 0\*2=0; 6\*3=18 5. 9 + 12 + 15 + 0 + 63 + 16 + 36 + 0 + 0 +18 = 169 6. 169 MOD 11 = 4 (i.e. C11) | |

Table 25: MS Specific Algorithms - Lithuania

## Luxembourg

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13] | | Where C1 to C13 are characters. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C8, C9, C10, C11, C12, C13 | | A numeric. |
| **Structure: Rules** | C1, C2, C3, C4 | Four digits of a year. | |
| C5, C6 | Month (in the range 01...12). | |
| C7, C8 | Day (in the range 01...31). | |

|  |  |  |
| --- | --- | --- |
| **Syntax: Check Digit** | C12 | 1. Multiply the values of each position by the corresponding weight:   C1 2  C2 1  C3 2  C4 1  C5 2  C6 1  C7 2  C8 1  C9 2  C10 1  C11 2  C12 1   1. If the product of a doubling operation is > 9, sum the digits of the product; 2. Add up the results of the above multiplications; 3. Get modulo 10 of the result of the previous addition; 4. If remainder = 0, C12 is valid. Otherwise the TIN is not valid. |
| C13 | 1. Create an array *n* containing the individual C1 to C11 and C13 of the TIN (where ni = the value of the corresponding C), taken from right to left: Description: C:\Users\crespero\Documents\TIN on the web\Luxembourg\Verhoeff-4bis.png 2. Initialize the checksum c to 0; 3. For each index i of the array *n*, starting at 0, replace c by d(c,p(i mod 8, ni)), according to the following tables:  Description: C:\Users\crespero\Documents\TIN on the web\Luxembourg\Verhoeff-3.png 4. Check digit c if c = 0, C13 is valid. Otherwise, the TIN is not valid. |
|  | C12, C13 | The TIN is only valid if the check digits of C12 and C13 are valid. |
| **Syntax: Sample** | 1893120105732 | Verification of C12:   1. 1 \* 2 = 2, 8 \* 1 = 8, 9 \* 2 = 18, 3 \* 1 = 3, 1 \* 2 = 2, 2 \* 1 = 2, 0 \* 2 = 0, 1 \* 1 = 1, 0 \* 2 = 0, 5 \* 1 = 5, 7 \* 2 = 14, 3 \* 1 = 3; 2. 1 + 8 = 9, 1 + 4 = 5; 3. 2 + 8 + 9 + 3 + 2 + 2 + 0 + 1 + 0 + 5 + 5 + 3 = 40; 4. 40 mod 10 = 0; 5. Check digit = remainder = 0, C12 is valid. |
| 1893120105732 | Verification of C13:   1. Description: C:\Users\crespero\Documents\TIN on the web\Luxembourg\Verhoeff-5.png 2. c = 0; 3. Description: C:\Users\crespero\Documents\TIN on the web\Luxembourg\Verhoeff-6.png 4. Check digit = 0, C13 is valid. |
|  | 1893120105732 | C12 and C13 are valid, the TIN is valid. |

Table 26: MS Specific Algorithms - Luxembourg

## Malta

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, C7, C8] | | Where C1 to C8 are characters. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C7 | | A numeric. |
| C8 | | A letter. |
| **Structure: Rule** | C8 | M, G, A, P, L, H, B or Z. | |
| C1 - C7 when C8 is A or P | C1 - C7 is included between 1 (0000001) and 9999999. | |
| C1 - C7 when C8 is M, G, L, H, B or Z | C1 - C5 is included between 0 (00000) and 32000 and C6 - C7 are included between 00 and 99 (but with the strict exclusion of 0000000). | |
| **Structure: Special Characters** | From an IT perspective, there should always be 8 characters. If the first 4 digits are omitted in the query, the result should anyway be reported in 8 characters by including leading zeros. E.g. when "199Z" or "34581M" is queried, the results should be shown respectively as "0000199Z" and "0034581M". | | |
| **Structure: Sample** | 1234567A. | | |

|  |  |
| --- | --- |
| **Syntax: Check Digit** | Not publicly available |
| **Syntax: Sample** | Not publicly available |

Table 27: MS Specific Algorithms - Malta 1

|  |  |  |
| --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, C7, C8, C9] | Where C1 to C9 are characters. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C7, C8, C9 | A numeric. |
| **Structure: Rule** | C1-C2 | C1-C2 can only take the values 11, 22, 33, 44, 55, 66, 77 or 88 |

|  |  |
| --- | --- |
| **Syntax: Check Digit** | Not publicly available |
| **Syntax: Sample** | Not publicly available |

Table 28: MS Specific Algorithms - Malta 2

## Netherlands

|  |  |  |
| --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, C7, C8, C9] | Where C1 to C9 are characters. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C7, C8, C9 | A numeric. |

|  |  |  |
| --- | --- | --- |
| **Syntax: Rule** | C9 | 1. Multiply the values of each position by the corresponding weight:   C1 9  C2 8  C3 7  C4 6  C5 5  C6 4  C7 3  C8 2   1. Add up the results of the above multiplications; 2. Get modulo 11 of the result of the previous addition; 3. Check digit = remainder (if remainder = 10, the TIN is not valid). |
| **Syntax: Sample** | 174559434 | 1. 1 \* 9 = 9, 7 \* 8 = 56, 4 \* 7 = 28, 5 \* 6 = 30, 5 \* 5 = 25, 9 \* 4 = 36, 4 \* 3 = 12, 3 \* 2 = 6; 2. 9 + 56 + 28 + 30 + 25 + 36 + 12 + 6 = 202; 3. 202 MOD 11 = 4; 4. Check digit = 4. |

Table 29: MS Specific Algorithms - Netherlands

## Poland

This algorithm is the same as the one used for this MS in VIES-VATValidationRoutines [R02].

|  |  |  |
| --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, C7, C8, C9, C10] | Where C1 to C10 are characters. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C7, C8, C9, C10 | A numeric. |

|  |  |  |
| --- | --- | --- |
| **Syntax: Rule** | C10 | 1. Multiply the values of each position by the corresponding weight:   C1 6  C2 5  C3 7  C4 2  C5 3  C6 4  C7 5  C8 6  C9 7   1. Add up the results of the above multiplications; 2. Get modulo 11 of the result of the previous addition; 3. Check digit = remainder (if remainder = 10, the TIN is not valid). |
| **Syntax: Sample** | 2234567895 | 1. 2 \* 6 = 12, 2 \* 5 = 10, 3 \* 7 = 21, 4 \* 2 = 8,5 \* 3 = 15,6 \* 4 = 24, 7 \* 5 = 35, 8 \* 6 = 48, 9 \* 7 = 63; 2. 12 + 10 + 21 + 8 + 15 + 24 + 35 + 48 + 63 = 236; 3. 236 MOD 11 = 5; 4. Check digit = 5. |

Table 30: MS Specific Algorithms - Poland 1

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure: Format** | [C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11] | | Where C1 to C11 are characters. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11 | | A numeric. |
| **Structure: Rule** | C1 - C2 | Stand for the last numbers of the year of birth. | |
| C3 - C4 | Stand for the month of birth. However for birthdates between 1900 and 1999 no change to C3 - C4 is made, for other birthdates:   * Between 1800 and 1899: month field is increased by 80; * Between 2000 and 2099: month field is increased by 20; * Between 2100 and 2199: month field is increased by 40; * Between 2200 and 2299: month field is increased by 60.   The adopted method of coding the month of birth allows distinguishing at least five centuries.  Therefore, the following months in the various centuries will have the following numbers: | |
| C5 - C6 | Stand for the day of the birth. | |
| C7 - C9 | Stand for an ordinal number. | |
| C10 | Denotes sex (digits 0, 2, 4, 6, 8 for females and digits 1, 3, 5, 7, 9 for males). | |

|  |  |  |
| --- | --- | --- |
| **Syntax: Rule** | C11 | 1. Multiply the values of each position by the corresponding weight:   C1 1  C2 3  C3 7  C4 9  C5 1  C6 3  C7 7  C8 9  C9 1  C10 3   1. Add up the last digits of the above multiplications results; 2. Value of the last digit of the sum deduct from 10. If this last digit is 0, 0 (not 10); 3. The result of subtraction is the check digit. |
| **Syntax: Sample** | 02070803628 | 1. A person born on the 08/07/1902, sex - female; 2. 0 \* 1 = 0, 2 \* 3 = 6, 0 \* 7 = 0, 7 \* 9 = 63, 0 \* 1 = 0, 8 \* 3 = 24, 0 \* 7 = 0, 3 \* 9 = 27, 6 \* 1 = 6, 2 \* 3 = 6; 3. 0 + 6 + 0 + 3 + 0 + 4 + 0 + 7 + 6 + 6 = 32; 4. 10 - 2 = 8; 5. Check digit = 8. |

Table 31: MS Specific Algorithms - Poland 2

## Portugal

|  |  |  |
| --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, C7, C8, C9] | Where C1 to C9 are characters. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C7, C8, C9 | A numeric. |

|  |  |  |
| --- | --- | --- |
| **Syntax: Rule** | C9 | 1. Multiply the values of each position by the corresponding weight:   C1 9  C2 8  C3 7  C4 6  C5 5  C6 4  C7 3  C8 2   1. Add up the results of the above multiplications; 2. Get modulo 11 of the result of the previous addition; 3. Check digit = 11 - remainder:  * If check digit < = 9 then check digit is OK (11 – remainder); * If check digit = 10 then check digit is 0 (zero); * If check digit = 11 then check digit is 0 (zero). |
| **Syntax: Sample** | 299999998 | 1. 2 \* 9 = 18, 9 \* 8 = 72, 9 \* 7 = 63, 9 \* 6 = 54, 9 \* 5 = 45, 9 \* 4 = 36, 9 \* 3 = 27, 9 \* 2 = 18; 2. 18 + 72 + 63 + 54 + 45 + 36 + 27 + 18 = 333; 3. MOD(333;11) = 3; 4. Check digit = 11 - 3 = 8. |

Table 32: MS Specific Algorithms - Portugal

## Romania

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13] | | Where C1 to C13 are characters. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13 | | A numeric. |
| **Structure: Rules** | C1 | In the range 1...9:   * 1 is for the male person born between 1900 and 1999; * 2 is for the female person born between 1900 and 1999; * 3 is for the male person born between 1800 and 1899; * 4 is for the female person born between 1800 and 1899; * 5 is for the male person born between 2000 and 2099; * 6 is for the female person born between 2000 and 2099; * 7 is for the foreign male obtaining temporary residence in Romania; * 8 is the foreign female obtaining temporary residence in Romania; * 9 is for the foreign citizens. | |
| C2, C3 | Two last digits of a year (in the range 00…99). | |
| C4, C5 | Month (in the range 1...12). | |
| C6, C7 | Day of month (in the range 1...31 depending on month and year). | |
| C8, C9 | County or district code (can only take the values 01 to 47 as well as 51 and 52). | |
| **Structure: Sample** | 8001011234567. | | |

|  |  |
| --- | --- |
| **Syntax: Check Digit** | Not publicly available |
| **Syntax: Sample** | Not publicly available |

Table 33: MS Specific Algorithms – Romania 1

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13] | | Where C1 to C13 are characters. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13 | | A numeric. |
| **Structure: Rules** | C1 | 9. | |
| C2, C3, C4 | 0. | |
| C5, C6, C7, C8, C9, C10, C11, C12 | In the range 0...9. | |
| **Structure: Sample** | 900012345678. | | |

|  |  |
| --- | --- |
| **Syntax: Check Digit** | Not publicly available |
| **Syntax: Sample** | Not publicly available |

Table 34: MS Specific Algorithms - Romania 2

## Slovakia

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure: TIN Format** | [C1 C2 C3 C4 C5 C6 C7 C8 C9 C10] | | Where C1 to C10 are digits. |
| **Structure: Range** | C1…C10 | | A numeric |
| **Structure: Special Characters** | If there is slash between C6 and C7, it should be skipped. | | |
| **Structure: Rules** | C1, C2 | In the range 0...9 | |
| C3, C4 | In the range 01...12 or in range 51…62 | |
| C5, C6 | In the range 01...31 | |
| C7 – C10 | In the range 0...9 | |

|  |  |  |
| --- | --- | --- |
| **Syntax: Check Digit** | C10 | If the number consisting of the digits C1, C2 is lower than 54, C10 might not be present. |
| **Syntax: Sample** | 7711167420 | C1…C10 numeric |
| 281203054 | C1…C9 numeric, C10 is not present because C1, C2 is lower than 54 |

Table 35: MS Specific Algorithms – Slovakia 1

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, C7, C8, C9, C10] | | Where C1 to C10 are characters. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C7, C8, C9, C10 | | A numeric. |
| **Structure: Special Characters** | No. | | |
| **Structure: Rules** | C1 to C10 | In the range 0...9 | |

|  |  |
| --- | --- |
| **Syntax: Check Digit** | No syntax check: only validation of structure. |

Table 36: MS Specific Algorithms – Slovakia 2

## Slovenia

This algorithm is the same as the one used for this MS in VIES-VATValidationRoutines [R02].

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, C7, C8] | | Where C1 to C8 are characters. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C7, C8 | | A numeric. |
| **Structure: Rules** | C1, C2, C3, C4, C5, C6, C7 | Must be between 1000000 and 9999999 included. | |

|  |  |  |
| --- | --- | --- |
| **Syntax: Check Digit** | C8 | 1. Multiply the values of each position by the corresponding weight:   C1 8  C2 7  C3 6  C4 5  C5 4  C6 3  C7 2   1. Add up the results of the above multiplications; 2. Get modulo 11 of the result of the previous addition; 3. Check digit = 11 - remainder. If result = 10, Check digit = 0. |
| **Syntax: TIN Format** | 15012557 | 1. 1 \* 8 = 8, 5 \* 7 = 35, 0 \* 6 = 0, 1 \* 5 = 5, 2 \* 4 = 8, 5 \* 3 = 15, 5 \* 2 = 10; 2. 8 + 35 + 0 + 5 + 8 + 15 + 10 = 81; 3. 81 MOD 11 = 4; 4. Check digit = 11 - 4 = 7. |

Table 37: MS Specific Algorithms - Slovenia

## Spain

|  |  |  |
| --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, C7, C8, C9] | Where C1 to C9 are characters. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C7, C8 | A numeric. |
| C9 | A letter. |
| **Structure: Special Characters** | From an IT perspective, there should always be 9 characters. If digits are omitted in the query, the result should anyway be reported in 9 characters by including leading zeros. E.g. 54237A should be understood as 00054237A and should not be blocking the check. | |
| **Structure: Sample** | 12345678A | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Syntax: Check Digit** | C9 | 1. Take the remainder of modulo 23 of the 8 first characters;  2. Add 1 to the remainder of operation 1;  3. The check letter corresponds to this figure in the table below:   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | | T | R | W | A | G | M | Y | F | P | D | X | B | N | J | Z | S | Q | V | H | L | C | K | E | |
| **Syntax: Sample** | (000)54237A | 1. (000)54237 MOD 23 = 3;  2. 3 + 1 = 4;  3. Check character = A. |

Table 38: MS Specific Algorithms - Spain 1

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure: TIN Format** | [C1,C2, C3, C4, C5, C6, C7, C8, C9] | | Where C1 to C9 are characters. |
| **Structure: Range** | C1, C9 | | A letter. |
| C2, C3, C4, C5, C6, C7, C8 | | A numeric. |
| **Structure: Rule** | C1 | Must be X, Y, Z, K, L or M. | |
| **Structure: Sample** | X1234567A. | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Syntax: Check Digit** | C9 | 0. Replace the leading letter by the corresponding digit and concatenate the result with the other characters:   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | X | Y | Z | K | L | M | | 0 | 1 | 2 | 0 | 0 | 0 |     1. Follow the Syntax Check from algorithm 1 above. |
| **Syntax: Sample** | X1234567L | 0. Replace X by 0 è 01234567;  1. 01234567 MOD 23 = 19;  2. 19 + 1 = 20;  3. Check character = L. |
| Z1234567R | 0. Replace Z by 2 è 21234567;  1. 21234567 MOD 23 = 1;  2. 1 + 1 = 2;  3. Check character = R. |
| M2812345C | 0. Replace M by 0 è 02812345;  1. 02812345 MOD 23 = 20;  2. 20 + 1 = 21;  3. Check character = C. |

Table 39: MS Specific Algorithms - Spain 2

## Sweden

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, “-“, C7, C8, C9, C10] | | Where C1 to C10 are characters. C1 to C6 must be separated from C7 to C10 by a hyphen. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C7, C8, C9, C10 | | A numeric. |
| **Structure: Special Characters** | Hyphen is mandatory. The hyphen has an importance and will not be skipped for the purpose of TIN validation. | | |
| **Structure: Rules** | C1, C2 | Two last digits of a year. | |
| C3, C4 | Month (in the range 1...12). | |
| C5, C6 | Day of month (in the range 1...31 depending on month and year). | |

|  |  |  |
| --- | --- | --- |
| **Syntax: Check Digit** | C10 | 1. Multiply the values of each position by the corresponding weight:   C1 2  C2 1  C3 2  C4 1  C5 2  C6 1  C7 2  C8 1  C9 2   1. Add up the results of the above multiplications. NB: 12 is regarded as 1 + 2; 2. The unit digit in the sum of the digits is subtracted from 10 and the result is the check digit. If the resulting number is 10, the check digit is 0. |
| **Syntax: Sample** | 640823-3234 | 1. 6 \* 2 = 12, 4 \* 1 = 4, 0 \* 2 = 0, 8 \* 1 = 8, 2 \* 2 = 4, 3 \* 1 = 3, 3 \* 2 = 6, 2 \* 1 = 2, 3 \* 2 = 6; 2. 1 + 2 + 4 + 0 + 8 + 4 + 3 + 6 + 2 + 6 = 36; 3. Check digit = 10 - 6 = 4. |

Table 40: MS Specific Algorithms - Sweden 1

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, “-“, C7, C8, C9, C10] | | Where C1 to C10 are characters. C1 to C6 must be separated from C7 to C10 by a hyphen. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C7, C8, C9, C10 | | A numeric. |
| **Structure: Special Characters** | Hyphen is mandatory. The hyphen has an importance and will not be skipped for the purpose of TIN validation. | | |
| **Structure: Rules** | C1, C2 | Two last digits of a year. | |
|  | C3, C4 | Month (in the range 1...12). | |
| C5, C6 | Day of month + 60 (in the range 61...91 depending on month and year). | |

|  |  |  |
| --- | --- | --- |
| **Syntax: Check Digit** | C10 | 1. Multiply the values of each position by the corresponding weight:   C1 2  C2 1  C3 2  C4 1  C5 2  C6 1  C7 2  C8 1  C9 2   1. Add up the results of the above multiplications. NB: 12 is regarded as 1 + 2; 2. The unit digit in the sum of the digits is subtracted from 10 and the result is the check digit. If the resulting number is 10, the check digit is 0. |
| **Syntax: Sample** | 640883-3231 | 1. 6 \* 2 = 12, 4 \* 1 = 4, 0 \* 2 = 0, 8 \* 1 = 8, 8 \* 2 = 16, 3 \* 1 = 3, 3 \* 2 = 6, 2 \* 1 = 2, 3 \* 2 = 6; 2. 1 + 2 + 4 + 0 + 8 + 1 + 6 + 3 + 6 + 2 + 6 = 39; 3. Check digit = 10 - 9 = 1. |

Table 41: MS Specific Algorithms - Sweden 2

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12] | | Where C1 to C12 are charactersand written as a single block. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12 | | A numeric. |
| **Structure: Special Characters** |  | | |
| **Structure: Rules** | C1, C2 | Can only be 18 or 19 or 20 (i.e. century of birth). | |
| C1, C2, C3, C4 | Four digits of a year. | |
| C5, C6 | Month (in the range 01...12). | |
| C7, C8 | Day of month (in the range 01...31 depending on month and year). | |

|  |  |  |
| --- | --- | --- |
| **Syntax: Check Digit** | C12 | 1. Multiply the values of each position from C3 to C11 by the corresponding weight:   C3 2  C4 1  C5 2  C6 1  C7 2  C8 1  C9 2  C10 1  C11 2   1. Add up the results of the above multiplications. NB: 12 is regarded as 1 + 2; 2. The unit digit in the sum of the digits is subtracted from 10 and the result is the check digit. If the resulting number is 10, the check digit is 0. |
| **Syntax: Sample** | 196408233234 | 1. 6 \* 2 = 12, 4 \* 1 = 4, 0 \* 2 = 0, 8 \* 1 = 8, 2 \* 2 = 4, 3 \* 1 = 3, 3 \* 2 = 6, 2 \* 1 = 2, 3 \* 2 = 6; 2. 1 + 2 + 4 + 0 + 8 + 4 + 3 + 6 + 2 + 6 = 36; 3. Check digit = 10 - 6 = 4. |

Table 42: MS Specific Algorithms - Sweden 3

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12] | | Where C1 to C12 are characters and written as a single block. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12 | | A numeric. |
| **Structure: Special Characters** |  | | |
| **Structure: Rules** | C1, C2 | Can only be 18 or 19 or 20 (i.e. century of birth). | |
|  | C1, C2, C3, C4 | Four digits of a year. | |
| C5, C6 | Month (in the range 01...12). | |
| C7, C8 | Day of month + 60 (in the range 61...91 depending on month and year). | |

|  |  |  |
| --- | --- | --- |
| **Syntax: Check Digit** | C12 | 1. Multiply the values of each position by the corresponding weight:   C3 2  C4 1  C5 2  C6 1  C7 2  C8 1  C9 2  C10 1  C11 2   1. Add up the results of the above multiplications. NB: 12 is regarded as 1 + 2; 2. The unit digit in the sum of the digits is subtracted from 10 and the result is the check digit. If the resulting number is 10, the check digit is 0. |
| **Syntax: Sample** | 196408833231 | 1. 6 \* 2 = 12, 4 \* 1 = 4, 0 \* 2 = 0, 8 \* 1 = 8, 8 \* 2 = 16, 3 \* 1 = 3, 3 \* 2 = 6, 2 \* 1 = 2, 3 \* 2 = 6; 2. 1 + 2 + 4 + 0 + 8 + 1 + 6 + 3 + 6 + 2 + 6 = 39; 3. Check digit = 10 - 9 = 1. |

Table 43: MS Specific Algorithms - Sweden 4

## United Kingdom

|  |  |  |
| --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, C7, C8, C9, C10] | Where C1 to C10 are characters. |
| **Structure: Range** | C1, C2, C3, C4, C5, C6, C7, C8, C9, C10 | A numeric. |
| **Structure: Sample** | 1234567890. | |

|  |  |
| --- | --- |
| **Syntax: Check Digit** | No syntax check: only validation of structure. |

Table 44: MS Specific Algorithms - United Kingdom 1

|  |  |  |  |
| --- | --- | --- | --- |
| **Structure: TIN Format** | [C1, C2, C3, C4, C5, C6, C7, C8, C9] | | Where C1 to C9 are characters. |
| **Structure: Range** | C1, C2 | | A letter. |
| C3, C4, C5, C6, C7, C8 | | A numeric. |
| C9 | | A letter or whitespace (meaning optional in this last case). |
| **Structure: Rules** | C1 | Must not be D, F, I, Q, U, V. | |
| C2 | Must not be D, F, I, O, Q, U, V. | |
| C1, C2 | Combination of C1 and C2 must not be GB, NK, TN or ZZ. | |
| C9 | Must be A, B, C, D or a whitespace. | |
| **Structure: Sample** | AA123456A. | | |

|  |  |
| --- | --- |
| **Syntax: Check Digit** | No syntax check: only validation of structure. |

Table 45: MS Specific Algorithms - United Kingdom 2

1. This algorithm is applicable as of 01/07/2017. [↑](#footnote-ref-2)