



Informatica® Address Verification  
5.8.1

# Developer Guide (On-Premises)

© Copyright Informatica LLC 1993, 2018

This software and documentation contain proprietary information of Informatica LLC and are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright law. Reverse engineering of the software is prohibited. No part of this document may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording or otherwise) without prior consent of Informatica LLC. This Software may be protected by U.S. and/or international Patents and other Patents Pending.

Use, duplication, or disclosure of the Software by the U.S. Government is subject to the restrictions set forth in the applicable software license agreement and as provided in DFARS 227.7202-1(a) and 227.7702-3(a) (1995), DFARS 252.227-7013(1)(ii) (OCT 1988), FAR 12.212(a) (1995), FAR 52.227-19, or FAR 52.227-14 (ALT III), as applicable.

The information in this product or documentation is subject to change without notice. If you find any problems in this product or documentation, please report them to us in writing.

Informatica, Informatica Platform, Informatica Data Services, PowerCenter, PowerCenterRT, PowerCenter Connect, PowerCenter Data Analyzer, PowerExchange, PowerMart, Metadata Manager, Informatica Data Quality, Informatica Data Explorer, Informatica B2B Data Transformation, Informatica B2B Data Exchange Informatica On Demand, Informatica Identity Resolution, Informatica Application Information Lifecycle Management, Informatica Complex Event Processing, Ultra Messaging and Informatica Master Data Management are trademarks or registered trademarks of Informatica LLC in the United States and in jurisdictions throughout the world. All other company and product names may be trade names or trademarks of their respective owners.

Portions of this software and/or documentation are subject to copyright held by third parties, including without limitation: Copyright DataDirect Technologies. All rights reserved. Copyright © Sun Microsystems. All rights reserved. Copyright © RSA Security Inc. All Rights Reserved. Copyright © Ordinal Technology Corp. All rights reserved. Copyright © Aandacht c.v. All rights reserved. Copyright Genivia, Inc. All rights reserved. Copyright Isomorphic Software. All rights reserved. Copyright © Meta Integration Technology, Inc. All rights reserved. Copyright © Intalio. All rights reserved. Copyright © Oracle. All rights reserved. Copyright © Adobe Systems Incorporated. All rights reserved. Copyright © DataArt, Inc. All rights reserved. Copyright © ComponentSource. All rights reserved. Copyright © Microsoft Corporation. All rights reserved. Copyright © Rogue Wave Software, Inc. All rights reserved. Copyright © Teradata Corporation. All rights reserved. Copyright © Yahoo! Inc. All rights reserved. Copyright © Glyph & Cog, LLC. All rights reserved. Copyright © Thinkmap, Inc. All rights reserved. Copyright © Clearpace Software Limited. All rights reserved. Copyright © Information Builders, Inc. All rights reserved. Copyright © OSS Nokalva, Inc. All rights reserved. Copyright Edifecs, Inc. All rights reserved. Copyright Cleo Communications, Inc. All rights reserved. Copyright © International Organization for Standardization 1986. All rights reserved. Copyright © ej-technologies GmbH. All rights reserved. Copyright © Jaspersoft Corporation. All rights reserved. Copyright © International Business Machines Corporation. All rights reserved. Copyright © yWorks GmbH. All rights reserved. Copyright © Lucent Technologies. All rights reserved. Copyright (c) University of Toronto. All rights reserved. Copyright © Daniel Veillard. All rights reserved. Copyright © Unicode, Inc. Copyright IBM Corp. All rights reserved. Copyright © MicroQuill Software Publishing, Inc. All rights reserved. Copyright © PassMark Software Pty Ltd. All rights reserved. Copyright © LogiXML, Inc. All rights reserved. Copyright © 2003-2010 Lorenzi Davide, All rights reserved. Copyright © Red Hat, Inc. All rights reserved. Copyright © The Board of Trustees of the Leland Stanford Junior University. All rights reserved. Copyright © EMC Corporation. All rights reserved. Copyright © Flexera Software. All rights reserved. Copyright © Jinfonet Software. All rights reserved. Copyright © Apple Inc. All rights reserved. Copyright © Telerik Inc. All rights reserved. Copyright © BEA Systems. All rights reserved. Copyright © PDFlib GmbH. All rights reserved. Copyright © Orientation in Objects GmbH. All rights reserved. Copyright © Tanuki Software, Ltd. All rights reserved. Copyright © Ricebridge. All rights reserved. Copyright © Sencha, Inc. All rights reserved. Copyright © Scalable Systems, Inc. All rights reserved. Copyright © jqWidgets. All rights reserved. Copyright © Tableau Software, Inc. All rights reserved. Copyright © MaxMind, Inc. All Rights Reserved. Copyright © TMate Software s.r.o. All rights reserved. Copyright © MapR Technologies Inc. All rights reserved. Copyright © Amazon Corporate LLC. All rights reserved. Copyright © Highsoft. All rights reserved. Copyright © Python Software Foundation. All rights reserved. Copyright © BeOpen.com. All rights reserved. Copyright © CNRI. All rights reserved.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>), and/or other software which is licensed under various versions of the Apache License (the "License"). You may obtain a copy of these Licenses at <http://www.apache.org/licenses/>. Unless required by applicable law or agreed to in writing, software distributed under these Licenses is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the Licenses for the specific language governing permissions and limitations under the Licenses.

This product includes software which was developed by Mozilla (<http://www.mozilla.org/>), software copyright The JBoss Group, LLC, all rights reserved; software copyright © 1999-2006 by Bruno Lowagie and Paulo Soares and other software which is licensed under various versions of the GNU Lesser General Public License Agreement, which may be found at <http://www.gnu.org/licenses/lgpl.html>. The materials are provided free of charge by Informatica, "as-is", without warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose.

The product includes ACE(TM) and TAO(TM) software copyrighted by Douglas C. Schmidt and his research group at Washington University, University of California, Irvine, and Vanderbilt University, Copyright (©) 1993-2006, all rights reserved.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (copyright The OpenSSL Project. All Rights Reserved) and redistribution of this software is subject to terms available at <http://www.openssl.org> and <http://www.openssl.org/source/license.html>.

This product includes Curl software which is Copyright 1996-2013, Daniel Stenberg, <daniel@haxx.se>. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at <http://curl.haxx.se/docs/copyright.html>. Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

The product includes software copyright 2001-2005 (©) MetaStuff, Ltd. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at <http://www.dom4j.org/license.html>.

The product includes software copyright © 2004-2007, The Dojo Foundation. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at <http://dojotoolkit.org/license>.

This product includes ICU software which is copyright International Business Machines Corporation and others. All rights reserved. Permissions and limitations regarding this software are subject to terms available at <http://source.icu-project.org/repos/icu/icu/trunk/license.html>.

This product includes software copyright © 1996-2006 Per Bothner. All rights reserved. Your right to use such materials is set forth in the license which may be found at <http://www.gnu.org/software/kawa/Software-License.html>.

This product includes OSSP UUID software which is Copyright © 2002 Ralf S. Engelschall, Copyright © 2002 The OSSP Project Copyright © 2002 Cable & Wireless Deutschland. Permissions and limitations regarding this software are subject to terms available at <http://www.opensource.org/licenses/mit-license.php>.

This product includes software developed by Boost (<http://www.boost.org/>) or under the Boost software license. Permissions and limitations regarding this software are subject to terms available at [http://www.boost.org/LICENSE\\_1\\_0.txt](http://www.boost.org/LICENSE_1_0.txt).

This product includes software copyright © 1997-2007 University of Cambridge. Permissions and limitations regarding this software are subject to terms available at <http://www.pcre.org/license.txt>.

This product includes software copyright © 2007 The Eclipse Foundation. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at <http://www.eclipse.org/org/documents/epl-v10.php> and at <http://www.eclipse.org/org/documents/edl-v10.php>.

This product includes software licensed under the terms at <http://www.tcl.tk/software/tcltk/license.html>, <http://www.bosrup.com/web/overlib/?License>, <http://www.stlport.org/doc/license.html>, <http://asm.ow2.org/license.html>, <http://www.cryptix.org/LICENSE.TXT>, <http://hsqldb.org/web/hsqLicense.html>, <http://httpunit.sourceforge.net/doc/license.html>, <http://jung.sourceforge.net/license.txt>, [http://www.gzip.org/zlib/zlib\\_license.html](http://www.gzip.org/zlib/zlib_license.html), <http://www.openldap.org/software/release/license.html>, <http://www.libssh2.org>, <http://slf4j.org/license.html>, <http://www.sente.ch/software/OpenSourceLicense.html>, <http://fusesource.com/downloads/license-agreements/fuse-message-broker-v-5-3-license-agreement>, <http://antlr.org/license.html>, <http://aopalliance.sourceforge.net/>, <http://www.bouncycastle.org/licence.html>, <http://www.jgraph.com/jgraphdownload.html>, <http://www.jcraft.com/jsch/LICENSE.txt>, [http://jotm.objectweb.org/bsd\\_license.html](http://jotm.objectweb.org/bsd_license.html), <http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231>, <http://www.slf4j.org/license.html>, <http://nanoxml.sourceforge.net/orig/copyright.html>, <http://www.json.org/license.html>, <http://forge.ow2.org/projects/javaservice/>, <http://www.postgresql.org/about/licence.html>, <http://www.sqlite.org/copyright.html>, <http://www.tcl.tk/software/tcltk/license.html>, <http://www.jaxen.org/faq.html>, <http://www.jdom.org/docs/faq.html>, <http://www.slf4j.org/license.html>, <http://www.iodbc.org/dataspace/iodbc/wiki/IODBC/License>, <http://www.keplerproject.org/md5/license.html>, <http://www.toedter.com/en/jcalendar/license.html>, <http://www.edankert.com/bounce/index.html>, <http://www.net-snmp.org/about/license.html>, <http://www.openmdx.org/#FAQ>, [http://www.php.net/license/3\\_01.txt](http://www.php.net/license/3_01.txt), <http://srp.stanford.edu/license.txt>, <http://www.schneider.com/blowfish.html>, <http://www.jmock.org/license.html>, <http://xsom.java.net>, <http://benalman.com/about/license/>, <https://github.com/CreateJS/EaselJS/blob/master/src/easeljs/display/Bitmap.js>, <http://www.h2database.com/html/license.html#summary>, <http://jsoncpp.sourceforge.net/LICENSE>, <http://jdbc.postgresql.org/license.html>, <http://protobuf.googlecode.com/svn/trunk/src/google/protobuf/descriptor.proto>, <https://github.com/rantav/hector/blob/master/LICENSE>, <http://web.mit.edu/Kerberos/krb5-current/doc/mitK5license.html>, <http://jibx.sourceforge.net/jibx-license.html>, <https://github.com/lyokato/libgeohash/blob/master/LICENSE>, <https://github.com/hjiang/jsonxx/blob/master/LICENSE>, <https://code.google.com/p/lz4/>, <https://github.com/jedisct1/libsodium/blob/master/LICENSE>, <http://one-jar.sourceforge.net/index.php?page=documents&file=license>, <https://github.com/EsotericSoftware/kryo/blob/master/license.txt>, <http://www.scala-lang.org/license.html>, <https://github.com/tinkerpop/blueprints/blob/master/LICENSE.txt>, <http://gee.cs.oswego.edu/dl/classes/EDU/oswego/cs/dl/util/concurrent/intro.html>, <https://aws.amazon.com/ssl/>, <https://github.com/twbs/bootstrap/blob/master/LICENSE>, <https://sourceforge.net/p/xmlunit/code/HEAD/tree/trunk/LICENSE.txt>, <https://github.com/documentcloud/underscore-contrib/blob/master/LICENSE>, and <https://github.com/apache/hbase/blob/master/LICENSE.txt>.

This product includes software licensed under the Academic Free License (<http://www.opensource.org/licenses/afl-3.0.php>), the Common Development and Distribution License (<http://www.opensource.org/licenses/cddl1.php>), the Common Public License (<http://www.opensource.org/licenses/cpl1.0.php>), the Sun Binary Code License Agreement Supplemental License Terms, the BSD License (<http://www.opensource.org/licenses/bsd-license.php>), the new BSD License (<http://opensource.org/licenses/BSD-3-Clause>), the MIT License (<http://www.opensource.org/licenses/mit-license.php>), the Artistic License (<http://www.opensource.org/licenses/artistic-license-1.0>) and the Initial Developer's Public License Version 1.0 (<http://www.firebirdsql.org/en/initial-developer-s-public-license-version-1-0/>).

This product includes software copyright © 2003-2006 Joe Walnes, 2006-2007 XStream Committers. All rights reserved. Permissions and limitations regarding this software are subject to terms available at <http://xstream.codehaus.org/license.html>. This product includes software developed by the Indiana University Extreme! Lab. For further information please visit <http://www.extreme.indiana.edu/>.

This product includes software Copyright (c) 2013 Frank Balluffi and Markus Moeller. All rights reserved. Permissions and limitations regarding this software are subject to terms of the MIT license.

See patents at <https://www.informatica.com/legal/patents.html>.

DISCLAIMER: Informatica LLC provides this documentation "as is" without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of noninfringement, merchantability, or use for a particular purpose. Informatica LLC does not warrant that this software or documentation is error free. The information provided in this software or documentation may include technical inaccuracies or typographical errors. The information in this software and documentation is subject to change at any time without notice.

## NOTICES

This Informatica product (the "Software") includes certain drivers (the "DataDirect Drivers") from DataDirect Technologies, an operating company of Progress Software Corporation ("DataDirect") which are subject to the following terms and conditions:

1. THE DATADIRECT DRIVERS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.
2. IN NO EVENT WILL DATADIRECT OR ITS THIRD PARTY SUPPLIERS BE LIABLE TO THE END-USER CUSTOMER FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR OTHER DAMAGES ARISING OUT OF THE USE OF THE ODBC DRIVERS, WHETHER OR NOT INFORMED OF THE POSSIBILITIES OF DAMAGES IN ADVANCE. THESE LIMITATIONS APPLY TO ALL CAUSES OF ACTION, INCLUDING, WITHOUT LIMITATION, BREACH OF CONTRACT, BREACH OF WARRANTY, NEGLIGENCE, STRICT LIABILITY, MISREPRESENTATION AND OTHER TORTS.

Publication Date: 2018-06-08

# Table of Contents

|  |           |
|--|-----------|
| <b>Preface .....</b>   | <b>10</b> |
| Informatica Resources. ....  | 10        |
| Informatica Network. ....  | 10        |
| Informatica Knowledge Base. ....   | 10        |
| Informatica Documentation. ....  | 11        |
| Informatica Product Availability Matrixes. ....  | 11        |
| Informatica Velocity. ....   | 11        |
| Informatica Marketplace. ....  | 11        |
| Informatica Global Customer Support. ....  | 11        |
| <br><b>Chapter 1: Introduction to Informatica Address Verification (On-Premises)..</b> | <b>12</b> |
| Informatica Address Verification (On-Premises) Overview. ....                          | 12        |
| Address Verification Process. ....   | 13        |
| Character Set Mapping. ....  | 13        |
| Transliteration. ....  | 13        |
| Address Parsing. ....  | 15        |
| Address Validation. ....   | 15        |
| Address Formatting and Standardization. ....   | 16        |
| Address Enrichments. ....  | 16        |
| Schematic Representation of the Address Verification Process. ....                     | 18        |
| Address Objects. ....  | 19        |
| Address Elements and Address Line Elements. ....                                       | 19        |
| Document Type Definition Files. ....   | 20        |
| <br><b>Chapter 2: General Settings.....</b>  | <b>21</b> |
| General Settings Overview. ....  | 21        |
| XML Encoding. ....   | 21        |
| XML Encoding Types. ....   | 22        |
| Unlock Codes and Database Attributes. ....   | 23        |
| Reference Database Order in SetConfig.xml. ....  | 24        |
| Preloading of Reference Address Databases. ....  | 24        |
| Memory Management. ....  | 25        |
| Calculating Memory Requirements. ....  | 26        |
| Configuring General Settings for Informatica Address Verification. ....                | 26        |
| The General Element. ....  | 27        |
| Configuring Unlock Codes and Database Settings. ....                                   | 28        |
| The Unlock Code Element. ....  | 28        |
| The Database Element. ....   | 28        |

|  |               |
|--|---------------|
| <b>Chapter 3: Input Parameters.....</b>                                | <b>30</b>     |
| Input Parameters Overview. . . . .                                     | 30            |
| Input Formatting Options. . . . .                                      | 30            |
| Configuring the Format Settings for Input Addresses. . . . .           | 31            |
| Input Country Settings. . . . .  | 31            |
| Configuring Country Settings for Input Addresses. . . . .              | 32            |
| <br><b>Chapter 4: Process Parameters.....</b>                          | <br><b>33</b> |
| XML Encoding for Parameters, InputData, and Results XML Files. . . . . | 33            |
| Configuring XML Encoding for Parameters, Input, and Result. . . . .    | 34            |
| Process Modes. . . . .   | 34            |
| Address Code LookUp. . . . .   | 34            |
| Batch Mode. . . . .  | 37            |
| Certified Mode. . . . .  | 37            |
| Country Recognition Mode. . . . .                                      | 41            |
| Fast Completion Mode. . . . .  | 41            |
| Interactive Mode. . . . .  | 43            |
| Key Differences Between Fast Completion and Interactive Modes. . . . . | 43            |
| Parse Mode. . . . .  | 44            |
| Specifying the Process Mode. . . . .                                   | 44            |
| Configuring Address Code Lookup. . . . .                               | 45            |
| Configuring Certified-Mode Address Verification. . . . .               | 45            |
| Reference Address Databases for the Certified Mode. . . . .            | 46            |
| Disabling the Certified Mode. . . . .                                  | 47            |
| Certified Mode Process Status Values. . . . .                          | 47            |
| AMAS Status Codes. . . . .   | 47            |
| CASS Status Values. . . . .  | 47            |
| SendRight Status Codes. . . . .  | 48            |
| SERP Status Codes. . . . .   | 48            |
| SNA Status Codes. . . . .  | 48            |
| The MatchingExtendedArchive Attribute. . . . .                         | 48            |
| Extended Archives and Japan Addresses. . . . .                         | 49            |
| Using the Extended Archives to Update a Japan Address. . . . .         | 50            |
| Extended Archives and South Korea Addresses. . . . .                   | 50            |
| Using the Extended Archives to Update a South Korea Address. . . . .   | 52            |
| The MatchingScope and MatchingAlternatives Attributes. . . . .         | 53            |
| Configuring the Verification Scope. . . . .                            | 53            |
| Configuring the MatchingAlternatives Attribute. . . . .                | 54            |
| Optimization Level. . . . .  | 54            |
| Configuring the Optimization Level. . . . .                            | 55            |

|   |           |
|---|-----------|
| <b>Chapter 5: Address Enrichments.....</b>                            | <b>56</b> |
| CAMEO Consumer Segmentation. . . . .                                  | 56        |
| Enabling CAMEO Demographic Profiles. . . . .                          | 58        |
| CAMEO Reference Address Databases. . . . .                            | 58        |
| CAMEO Output Fields. . . . .  | 60        |
| CAMEO Status Values. . . . .  | 60        |
| GeoCoding Types. . . . .  | 61        |
| Supported Countries and Databases. . . . .                            | 62        |
| Configuring GeoCoding. . . . .  | 66        |
| GeoCoding Status Values. . . . .                                      | 66        |
| Country-Specific Address Enrichments. . . . .                         | 67        |
| Reference Address Databases for Country-Specific Enrichments. . . . . | 68        |
| Enrichments for Austria Addresses. . . . .                            | 68        |
| Enabling Address Enrichments for Austria Addresses. . . . .           | 69        |
| Status Codes for Austria Address Enrichments. . . . .                 | 69        |
| Enrichment for Belgium Addresses. . . . .                             | 69        |
| Configuring Address Enrichments for Belgium. . . . .                  | 70        |
| Status Codes for Belgium Address Enrichments. . . . .                 | 70        |
| Enrichments for Brazil Addresses. . . . .                             | 70        |
| Configuring Enrichments for Brazil Addresses. . . . .                 | 71        |
| Status Codes for Brazil Address Enrichments. . . . .                  | 71        |
| Enrichments for France Addresses. . . . .                             | 71        |
| Enabling Address Enrichments for France Addresses. . . . .            | 72        |
| Status Codes for France Address Enrichments. . . . .                  | 72        |
| Enrichments for Germany Addresses. . . . .                            | 72        |
| Enabling Address Enrichments for Germany Addresses. . . . .           | 73        |
| Status Codes for Germany Address Enrichments. . . . .                 | 73        |
| Enrichments for Japan Addresses. . . . .                              | 74        |
| Enabling Address Enrichments for Japan Addresses. . . . .             | 74        |
| Status Codes for Japan Address Enrichments. . . . .                   | 74        |
| Enrichments for South Korea Addresses. . . . .                        | 75        |
| Enabling Address Enrichment for South Korea. . . . .                  | 75        |
| Status Codes for South Korea Address Enrichments. . . . .             | 76        |
| Enrichments for Poland Addresses. . . . .                             | 76        |
| Enabling Address Enrichments for Poland Addresses. . . . .            | 76        |
| Status Codes for Poland Address Enrichments. . . . .                  | 77        |
| Enrichments for Russia Addresses. . . . .                             | 77        |
| Configuring Enrichments for Russia Addresses. . . . .                 | 77        |
| Status Codes for Russia Address Enrichments. . . . .                  | 77        |
| Enrichments for Serbia Addresses. . . . .                             | 78        |
| Enabling Address Enrichments for Serbia Addresses. . . . .            | 78        |

|  |            |
|--|------------|
| Status Codes for Serbia Address Enrichments. . . . .   | 78         |
| Enrichments for South Africa Addresses. . . . .  | 79         |
| Enabling Address Enrichments for South Africa Addresses. . . . .                             | 79         |
| Status Codes for South Africa Address Enrichments. . . . .                                   | 79         |
| Enrichments for Switzerland Addresses. . . . .   | 79         |
| Enabling Address Enrichment for Switzerland Addresses. . . . .                               | 80         |
| Status Codes for Switzerland Address Enrichments. . . . .                                    | 80         |
| Enrichments for the United Kingdom Addresses. . . . .  | 80         |
| Configuring Address Enrichments for the United Kingdom Addresses. . . . .                    | 82         |
| Status Codes for the United Kingdom Enrichments. . . . .                                     | 82         |
| Enrichments for the United States Addresses. . . . .   | 82         |
| Enabling Address Enrichments for the United States Addresses. . . . .                        | 83         |
| Status Codes for the United States Address Enrichments. . . . .                              | 84         |
| <b>Chapter 6: Result Parameters. . . . .</b>   | <b>85</b>  |
| Result Formatting Options. . . . .   | 85         |
| Configuring the Result Formatting Options. . . . .   | 86         |
| Result Country Settings. . . . .   | 87         |
| Configuring Country Settings for the Result Output. . . . .                                  | 87         |
| Configuring the Maximum Number for Results in Interactive and Fast Completion Modes. . . . . | 89         |
| The PreferredScript Attribute. . . . .   | 89         |
| Countries and PreferredScript Options. . . . .   | 90         |
| PreferredScript Examples. . . . .  | 91         |
| The PreferredLanguage Attribute. . . . .   | 93         |
| Multilanguage Support for Belgium Addresses. . . . .   | 94         |
| Multilanguage Support for Canada Addresses. . . . .  | 95         |
| Multilanguage Support for Switzerland Addresses. . . . .                                     | 97         |
| The GlobalPreferredDescriptor Attribute. . . . .   | 98         |
| The ElementAbbreviation Attribute. . . . .   | 98         |
| Configuring the Element Descriptor Format. . . . .   | 99         |
| The AliasLocality and AliasStreet Attributes. . . . .  | 99         |
| Configuring AliasStreet and AliasLocality. . . . .   | 100        |
| The RangesToExpand and FlexibleRangeExpansion Attributes. . . . .                            | 100        |
| Configuring the Range Attributes. . . . .  | 101        |
| The StandardizeInvalidAddresses Attribute. . . . .   | 101        |
| Standardizing Nonvalid Addresses. . . . .  | 101        |
| The DualAddressPriority Attribute. . . . .   | 101        |
| Configuring the Type of Address to Validate. . . . .   | 102        |
| Output Standardization. . . . .  | 102        |
| Configuring Output Standardization at Address Element Level. . . . .                         | 104        |
| <b>Chapter 7: Output Fields. . . . .</b>   | <b>106</b> |
| Output Fields Overview. . . . .  | 106        |

|                                 |     |
|---------------------------------|-----|
| The Result Element. . . . .     | 106 |
| The ResultData Element. . . . . | 107 |
| AddressElements. . . . .        | 107 |
| AddressLines. . . . .           | 110 |
| AddressComplete. . . . .        | 111 |

## **Chapter 8: Assessment Codes and Return Codes..... 112**

|  |     |
|--|-----|
| Status Codes, Assessment Codes, and API Return Codes. . . . .                            | 112 |
| Address Verification Process Status Values. . . . .                                      | 113 |
| Element Positions. . . . .   | 115 |
| Element Input Status. . . . .  | 116 |
| Element Relevance Value. . . . .   | 116 |
| Element Result Status. . . . .   | 117 |
| Address Resolution Code. . . . .   | 118 |
| Extended Element Result Status. . . . .  | 118 |
| Result Percentage Values. . . . .  | 119 |
| LanguageISO3 Code. . . . .   | 119 |
| Mailability Score. . . . .   | 120 |
| Address Types. . . . .   | 120 |
| Address Types from the United States Postal Service. . . . .                             | 121 |
| Address Type Indicators for the United States Addresses in Fast Completion Mode. . . . . | 121 |
| Address Type Indicators for Australia Addresses. . . . .                                 | 122 |
| Address Type Indicators for Canada Addresses. . . . .                                    | 122 |
| Address Type Indicators for France Addresses. . . . .                                    | 123 |
| Address Type Indicators for New Zealand Addresses. . . . .                               | 123 |
| Address Type Indicators for Addresses from the Rest of the World. . . . .                | 123 |
| Certified Mode Process Status Values. . . . .  | 124 |
| AMAS Status Codes. . . . .   | 124 |
| CASS Status Values. . . . .  | 124 |
| SendRight Status Codes. . . . .  | 125 |
| SERP Status Codes. . . . .   | 125 |
| SNA Status Codes. . . . .  | 125 |
| API Return Codes. . . . .  | 126 |
| Success. . . . .   | 126 |
| Warnings. . . . .  | 126 |
| Errors. . . . .  | 127 |
| Critical Errors. . . . .   | 129 |
| Very Critical Errors. . . . .  | 129 |

## **Chapter 9: Frequently Asked Questions..... 130**

|  |     |
|--|-----|
| How Do I Manage Unlock Codes?. . . . .                         | 130 |
| How Do I Initialize Informatica Address Verification?. . . . . | 131 |
| How Do I Produce Valid Address Verification XML?. . . . .      | 132 |



|   |     |
|---|-----|
| How Do I Manage Reference Address Databases? . . . . .                                      | 132 |
| How Do I Assign an Address to an AddressObject. . . . .                                     | 133 |
| How Do I Validate an Address? . . . . .   | 134 |
| How Do I Check the Process Mode? . . . . .  | 135 |
| How Do I Retrieve a Suggested Correction? . . . . .   | 135 |
| How Do I Retrieve the Result Status and Additional Information? . . . . .                   | 136 |
| How Do I Use the Quick Address Entry Feature of Informatica Address Verification? . . . . . | 136 |
| How Do I Use Informatica Address Verification with Multiple Processor Cores? . . . . .      | 137 |
| How Do I Use Address Verification for Master Data Management? . . . . .                     | 137 |
| How Do I Use Informatica Address Verification in an eBusiness Environment? . . . . .        | 137 |
| How Do I Specify Processing Parameters and Input and Result Formats? . . . . .              | 138 |
| How Do I Find the Current Post Code for a South Korea Address? . . . . .                    | 140 |
| How Do I Retrieve the Current Informatica Address Verification Settings? . . . . .          | 140 |
| How Do I Determine the Informatica Address Verification Version? . . . . .                  | 140 |

# Preface

The *Informatica Address Verification Developer (On-Premises) Guide* tells you how you can use Informatica Address Verification (On-Premises) to analyze, parse, validate, and enrich postal address data. This guide also describes the result output fields, process status codes, and various assessment codes that Informatica Address Verification (On-Premises) returns.

## Informatica Resources

### Informatica Network

Informatica Network hosts Informatica Global Customer Support, the Informatica Knowledge Base, and other product resources. To access Informatica Network, visit <https://network.informatica.com>.

As a member, you can:

- Access all of your Informatica resources in one place.
- Search the Knowledge Base for product resources, including documentation, FAQs, and best practices.
- View product availability information.
- Review your support cases.
- Find your local Informatica User Group Network and collaborate with your peers.

As a member, you can:

- Access all of your Informatica resources in one place.
- Search the Knowledge Base for product resources, including documentation, FAQs, and best practices.
- View product availability information.
- Find your local Informatica User Group Network and collaborate with your peers.

### Informatica Knowledge Base

Use the Informatica Knowledge Base to search Informatica Network for product resources such as documentation, how-to articles, best practices, and PAMs.

To access the Knowledge Base, visit <https://kb.informatica.com>. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team at [KB\\_Feedback@informatica.com](mailto:KB_Feedback@informatica.com).

## Informatica Documentation

To get the latest documentation for your product, browse the Informatica Knowledge Base at [https://kb.informatica.com/\\_layouts/ProductDocumentation/Page/ProductDocumentSearch.aspx](https://kb.informatica.com/_layouts/ProductDocumentation/Page/ProductDocumentSearch.aspx).

If you have questions, comments, or ideas about this documentation, contact the Informatica Documentation team through email at [infa\\_documentation@informatica.com](mailto:infa_documentation@informatica.com).

## Informatica Product Availability Matrixes

Product Availability Matrixes (PAMs) indicate the versions of operating systems, databases, and other types of data sources and targets that a product release supports. If you are an Informatica Network member, you can access PAMs at

<https://network.informatica.com/community/informatica-network/product-availability-matrices>.

## Informatica Velocity

Informatica Velocity is a collection of tips and best practices developed by Informatica Professional Services. Developed from the real-world experience of hundreds of data management projects, Informatica Velocity represents the collective knowledge of our consultants who have worked with organizations from around the world to plan, develop, deploy, and maintain successful data management solutions.

If you are an Informatica Network member, you can access Informatica Velocity resources at <https://velocity.informatica.com>.

If you have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at [ips@informatica.com](mailto:ips@informatica.com).

## Informatica Marketplace

The Informatica Marketplace is a forum where you can find solutions that augment, extend, or enhance your Informatica implementations. By leveraging any of the hundreds of solutions from Informatica developers and partners, you can improve your productivity and speed up time to implementation on your projects. You can access Informatica Marketplace at <https://marketplace.informatica.com>.

## Informatica Global Customer Support

You can contact a Global Support Center by telephone or through Online Support on Informatica Network.

To find your local Informatica Global Customer Support telephone number, visit the Informatica website at the following link:

<http://www.informatica.com/us/services-and-training/support-services/global-support-centers>.

If you are an Informatica Network member, you can use Online Support at <http://network.informatica.com>.

# CHAPTER 1

## Introduction to Informatica Address Verification (On-Premises)

This chapter includes the following topics:

- [Informatica Address Verification \(On-Premises\) Overview, 12](#)
- [Address Verification Process, 13](#)
- [Address Objects, 19](#)
- [Address Elements and Address Line Elements, 19](#)
- [Document Type Definition Files, 20](#)

## Informatica Address Verification (On-Premises) Overview

You can use Informatica Address Verification (On-Premises) to verify addresses from more than 240 countries and territories. Address Verification uses the most up-to-date and complete postal reference data and ensures quality and accuracy for addresses. You can verify addresses in a quick, efficient, and effective manner.

The address verification workflow includes transliteration, parsing, formatting, validation, and enrichment of addresses. You can use the integrated transliteration capabilities to validate addresses captured in different writing systems and languages. Address Verification also standardizes and formats the address data to meet the compliance requirements that the local postal authorities specify.

Address Verification complies with the following postal certifications:

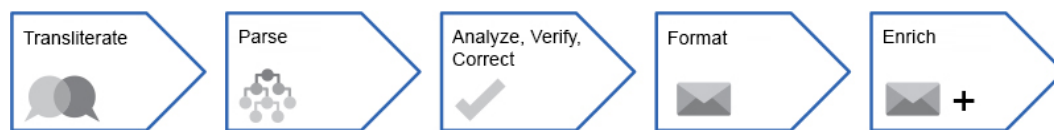
- Address Matching Approval System (AMAS) certification for Australia Post
- Coding Accuracy Support System (CASS) certification for the United States Postal Services
- SendRight certification for New Zealand Post
- Service National de L'Adresse (SNA) certification for La Poste of France
- Software Evaluation and Recognition Program (SERP) certification for Canada Post

Address verification can retrieve address enrichments for valid addresses. Enrichments include geocoordinates, country-specific enrichments such as INSEE codes in France and time-zone codes in the United States, and demographic profiles such as CAMEO.

## Address Verification Process

The address verification process includes multiple steps such as character set mapping, transliteration, parsing, validation, formatting, and enrichment. Based on the parameters you set, Informatica Address Verification completes the verification process and returns the result.

The following image shows the address verification process:



## Character Set Mapping

The character set mapping provides a mapping between characters in source and destination character sets and thus enables conversion between character sets. Informatica Address Verification internally uses Unicode and externally supports multiple character sets including UTF-8, ISO 8859-1, GBK, BIG5, JIS, and EBCDIC.

Character sets use a numeric representation for each of the supported alphabets or characters. Typically, character sets use the same numeric representation for common alphabets or characters. However, some of the language-specific characters have different numeric representations across character sets.

For example, the letter A has the same numeric representation, 65, in both Unicode and Latin character sets. However, the letter Å has different representations in Unicode and Latin character sets. Å is represented by 143 in Unicode and 197 in Latin character sets. Characters that have different numeric representations across character sets fail to appear correctly when you use different character sets to render the data.

To render character sets, Address Verification first converts the input character strings to Unicode. Then it uses the corresponding mapping of the destination character set to render the data with near perfection. If no representation is available for a character in the destination character set, Address Verification maps that character to an underscore character.

## Transliteration

Transliteration converts data between non-Roman characters and Latin characters. Transliteration also helps you replace diacritical and extended characters with plain text equivalents.

Transliteration helps nonnative speakers find an approximate pronunciation of a word based on the pronunciation rules of their own language. Transliterations of ISO character sets use invertible mapping so that the transliteration can be reversed without any information loss. However, for other character sets such as BGN, transliteration is not reversible.

Informatica Address Verification can transliterate to and from the following writing systems:

- Greek (BGN/PCGN 1962, ISO 843 – 1997)
- Cyrillic (BGN/PCGN 1947, ISO 9 – 1995)

**Note:** You can perform Cyrillic transliteration for Belarus, Kazakhstan, Russia, Macedonia, and Ukraine.

- Japanese Katakana, Hiragana, and Kanji
- Chinese Pinyin (Mandarin, Cantonese)

Transliteration goes beyond character set mapping, which is a mapping between different numeric representations of a character. A language such as Japanese with the Katakana, Hiragana, and Kanji characters has no direct representation in the English language. However, each Japanese character has a certain associated sound that can be approximated using phonetic Latin characters.

Multiple transliteration schemes are available for different languages. The following table shows transliteration of characters from different languages:

| Source Character Set | Input | Destination Character Set | Output |
|----------------------|-------|---------------------------|--------|
| Latin                | Ä     | ASCII                     | AE     |
| Latin                | ô     | ASCII                     | g      |
| Japanese             | ↑     | Latin                     | ka     |
| Cyrillic             | Ж     | Latin                     | ZH     |

## Transliteration Limitations

If the source language has fewer syllables and does not have exact matches in other languages, transliteration becomes difficult. Japanese is an example for languages that are difficult to transliterate.

Most languages use a subset of the sounds a person could produce. Different languages use different subsets. If a sound used by one language cannot be represented correctly in a different script, then it needs to be approximated. This approximation might be inaccurate if the sounds in the source and destination languages are significantly different.

The following table shows how transliteration can be inaccurate when the words are transliterated from English to Kana and then back to English:

| Original    | Japanese | Transliterated |
|-------------|----------|----------------|
| Philippines | フィリピン    | Firipin        |
| Düsseldorf  | デュッセルドルフ | Dyusserudorufu |
| Beethoven   | ベートーベン   | Betoben        |

Some Kanji characters in the first name of the contact in Japan addresses are incorrectly transliterated into Arabic numerals instead of Latin alphabets.

The following table shows the Kanji numerals that Informatica Address Verification might incorrectly translate into Arabic numerals instead of Latin alphabets:

| Kanji Numeral | Arabic Equivalent | Latin Transliteration |
|---------------|-------------------|-----------------------|
| 一             | 1                 | ichi                  |
| 二             | 2                 | ni                    |
| 三             | 3                 | san                   |
| 四             | 4                 | yon                   |
| 五             | 5                 | go                    |
| 六             | 6                 | roku                  |
| 七             | 7                 | nana                  |
| 八             | 8                 | hachi                 |
| 九             | 9                 | kyū                   |
| 十             | 10                | jū                    |

## Address Parsing

Informatica Address Verification can identify and categorize address elements and nonaddress elements.

Address Verification can parse fielded and unfielded addresses and return addresses that follow the address structure of the country to which the address belongs. A fielded address contains well-defined address element fields. An unfielded address contains lines of address data that include both address and nonaddress elements. An input address might also contain incorrectly assigned elements or elements that are incorrectly combined with unrelated elements.

Address parsing in Address Verification does not use the reference address databases. You can use the PARSE mode to parse addresses even if you do not have the corresponding reference address databases. When you run address verification in other process modes such as BATCH or CERTIFIED, Address Verification parses the address before the verification process begins. During the verification process, Address Verification might readjust the parsed results based on the reference address data.

Address Verification uses the `FormatDelimiter` attribute setting of the `Input` element in `Parameters.xml` to correctly identify address elements for parsing. If Address Verification fails to identify an element in a fielded address input, Address Verification returns the element in the same field in which it was input.

## Address Validation

You can validate an address against a set of comprehensive reference address databases. Informatica Address Verification validates each element in the input address against the data available in the reference address databases and returns results that include corrected or added information to improve the address quality.

Address Verification returns the best possible matches to an input address and reduces the possibility of incorrect modifications to address elements. If Address Verification cannot fully validate an address, it classifies the address based on the deliverability assessment.

You can choose the address validation process modes based on your requirements.

- If you want to input one or more addresses for validation and receive one correct result for each input address, choose the batch (BATCH) mode.
- To validate addresses based on the guidelines specified by the postal authority of a country, choose the certified (CERTIFIED) mode. The certified mode functions similar to the batch mode except that Address Verification supports the certified mode only for countries that have postal certifications. The certified mode address processing requires additional certified reference address databases.
- To receive a list of address suggestions from which you can manually choose the best match, choose the interactive (INTERACTIVE) mode.
- To receive address suggestions as you enter an address, choose the fast completion (FASTCOMPLETION) mode. In the fast completion mode, Address Verification returns good quality suggestions even for incomplete addresses and truncated address elements.

## Address Formatting and Standardization

After validating an address, Informatica Address Verification formats the address according to country-specific formatting rules and the formatting and standardization attributes you configure.

You can configure the formatting and standardization attributes of the `Result` element and the `AddressElementStandardize` element in `Parameters.xml` to specify the following options:

- Whether to include the country information in the `AddressComplete` output or not.
- The delimiter to separate address elements in the `AddressComplete` output.
- The level of address information in the `AddressComplete` output.
- The maximum number of lines in the `AddressComplete` output.
- The casing of the output.
- The maximum length of strings in each element.
- Whether to use element abbreviations or not.

Address Verification transliterates the result to the script specified for the `PreferredScript` attribute of the `Result` element in `Parameters.xml`.

## Address Enrichments

You can retrieve multiple address enrichments with the validated result output. Address enrichments provide additional information that helps you better understand and use the address data.

Address enrichments require additional databases and unlock codes. Note that you can retrieve address enrichments only for validated addresses in batch and interactive modes.

You can retrieve the following address enrichments for validated addresses:

### **CAMEO Social Profiling**

CAMEO profiles from the Callcredit Information Group provide consumer classification systems that are developed to assess the socioeconomic and geo-demographic profiles of neighborhoods across the world. Informatica Address Verification returns CAMEO profiles as an address enrichment for many countries.

### **Geocoding**

Geocoordinates contain the longitude and latitude of an address and help you find the physical location of an address. Address Verification returns the standard geocoding, arrival point geocoding, or parcel



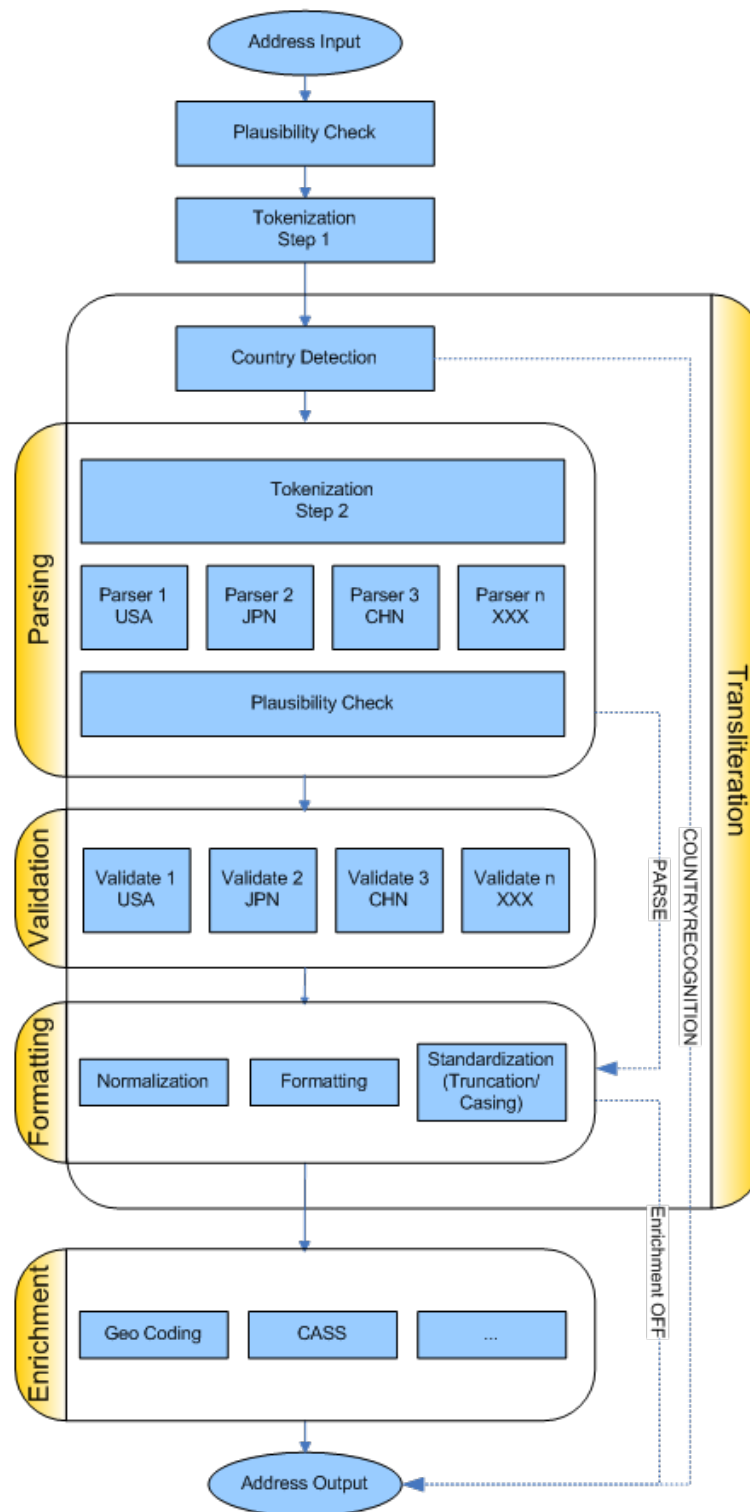
centroid geocoding based on the configuration. For the United Kingdom addresses, you can retrieve rooftop geocoordinates.

#### **Country-Specific Enrichments**

Country-specific enrichments provide information such as unique codes and identifiers that are specific to countries or territories. You can receive country-specific enrichments for select countries. Examples for country-specific enrichments include Postal Address Code for Austria addresses, INSEE codes for France addresses, and time zone codes for the United States addresses.

# Schematic Representation of the Address Verification Process

The following image shows the steps in the address verification process:



# Address Objects

An `AddressObject` is a data structure that stores an input address, validation parameters, result addresses along with any supported enrichment, and the process status and address assessment codes.

An `AddressObject` contains the following information:

- The process, input, result, and address element standardization parameters set in `Parameters.xml`.
- An input address as specified in `InputData.xml`.
- A result as specified in `Result.xml`.
- The last process code that Informatica Address Verification returned.

Address Verification can process multiple address objects simultaneously. In the `SetConfig.xml` file, you can specify the maximum number of address objects that you want Address Verification to handle simultaneously. Note that if you configure a value less than one for the maximum address object count, Address Verification fails to initialize.

To avoid processing overheads and to ensure better performance of Address Verification, reuse the address objects instead of creating and destroying an `AddressObject` every time you want to parse or validate an address. Frequent changes to the processing parameters of an `AddressObject` might slow down the Address Verification performance.

## Address Elements and Address Line Elements

You can enter and receive addresses as address elements and address line elements.

You can enter, or receive in the result, the following address elements:

- Country
- Locality
- Postal Code
- Province
- Street
- Number
- Building
- Subbuilding
- Delivery service
- Organization
- Contact
- Residue

Many of these elements can contain multiple levels of information. For example, the locality element supports up to six levels of information. The contact element supports three levels.

You can also enter or receive address information as address lines. Informatica Address Verification supports the following address line formats:

**RecipientLine**

Contains the recipient information such as contact or organization name. Address Verification supports up to six lines of recipient information.

**DeliveryAddressLine (DAL)**

Contains the delivery information such as street, house number, building, and delivery service information. Address Verification supports up to six lines of delivery address line information.

**CountrySpecificLocalityLine (CSLLN)**

Contains information such as locality, postal code, province, and country details. Address Verification supports up to six lines of country specific locality line information.

**FormattedAddressLine (FAL)**

Contains unfielded data that includes, recipient information, delivery address information, and locality information. Address Verification supports up to 19 lines of formatted address line information.

Address Verification also supports an `AddressComplete` element that contains the complete address as one string. Address Verification uses delimiters to separate address elements within the address complete element.

## Document Type Definition Files

You receive the following document type definition (DTD) files as part of the Informatica Address Verification package:

**SetConfig.dtd**

Defines the structure of the `SetConfig.xml` file that contains the general setting that you pass with the initialization function such as `AD_Initialize()`.

**Parameters.dtd**

Defines the structure of the `Parameters.xml` file that contains the input, process, and result parameters you pass with functions such as `AD_SetParametersXML()` or `AD_Initialize()`.

**InputData.dtd**

Defines the structure of XML input that you set with functions such as `AD_SetInputDataXML()`, and receive for functions such as `AD_GetInputDataXML()`.

**Result.dtd**

Defines the structure of Address Verification response that you receive for functions such as `AD_GetResultXML()`.

**GetConfig.dtd**

Defines the structure of XML response you receive for functions such as `AD_GetConfigSettingsXML()`.

Of the corresponding XML files, you can set values for `Parameters.xml`, `InputData.xml`, and `SetConfig.xml`. You cannot modify `Result.xml` and `GetConfig.xml` files that Address Verification generates.

## CHAPTER 2

# General Settings

This chapter includes the following topics:

- [General Settings Overview, 21](#)
- [XML Encoding, 21](#)
- [Unlock Codes and Database Attributes, 23](#)
- [Preloading of Reference Address Databases, 24](#)
- [Memory Management, 25](#)
- [Configuring General Settings for Informatica Address Verification, 26](#)
- [Configuring Unlock Codes and Database Settings, 28](#)

## General Settings Overview

You can configure the general settings such as XML encoding, memory settings, and database and unlock code settings for Informatica Address Verification in the `SetConfig.xml` file.

## XML Encoding

Character encoding defines unique binary codes for the different characters used in a document. In XML files, XML header tag `<?xml ...?>` contains the encoding information. Informatica Address Verification supports multiple encoding types for input and output.

In the `SetConfig.xml` file, you can specify the XML encoding type for the `GetConfig.xml` file that Address Verification generates.

When you use the XML API and you do not specify an encoding type in `SetConfig.xml`, Address Verification sets the encoding to UTF-8 or UTF-16 based on whether you use 8-bit or 16-bit API functions. For direct API implementations, the default value is ISO-8859-1.

For C-based APIs, you can use either 8-bit or 16-bit functions. A `W` suffix in the function name, such as `AD_SetInputDataElementW()`, identifies a 16-bit function. If you use 16-bit functions, you must set the encoding to a 16-bit encoding. If you use a 16-bit function when the encoding is set to 8-bit, Address Verification returns an encoding error. If you are using Java APIs, you must set the encoding to UTF-16 as Java supports only 16-bit input and output.

## XML Encoding Types

You can configure Informatica Address Verification to use one of the following XML encoding types:

- UTF-8
- ISO-8859-1
- ISO-8859-2
- ISO-8859-3
- ISO-8859-4
- ISO-8859-5
- ISO-8859-6
- ISO-8859-7
- ISO-8859-8
- ISO-8859-9
- ISO-8859-10
- ISO-8859-15
- IBM437
- IBM852
- IBM855
- IBM857
- IBM862
- WINDOWS-1250
- WINDOWS-1251
- WINDOWS-1252
- WINDOWS-1253
- WINDOWS-1254
- WINDOWS-1255
- WINDOWS-1256
- WINDOWS-1257
- US-ASCII
- DBCS\_BIG5
- DBCS\_JIS
- DBCS\_JIS-2004
- DBCS\_GBK
- DBCS\_UHC
- EBCDIC
- UTF-16
- UTF-16BE
- UTF-16LE
- UCS-2
- UCS-2BE

- UCS-2LE

## Unlock Codes and Database Attributes

Informatica Address Verification requires valid unlock codes to access the reference address databases for address validation and enrichments.

Unlock codes are unique 36-character alphanumeric keys that map to specific reference address databases. The single-line address validation has a separate unlock code even though it uses the fast completion reference database. Unlock codes are valid for a specific period of time. Unlock codes have a valid-from date and an expiry date.

You can view the status of the unlock codes in the `GetConfig.xml`. You can retrieve the `GetConfig.xml` file from Address Verification even if you do not have a valid unlock code. If an unlock code that you entered is not yet valid, Address Verification marks it as `NOT_YET_VALID` in `GetConfig.xml`. If the unlock code you entered is past its expiry date, Address Verification marks it as `EXPIRED` in `GetConfig.xml`.

You can get unlock codes from an Address Verification sales representative or you can buy the unlock codes online from the Address Verification Data Quality Center.

You must specify at least one valid unlock code of validation type in the `SetConfig.xml` file before you can process addresses using Address Verification. You can specify multiple unlock code elements in the `SetConfig.xml` file.

To access the reference address databases, you must enter the details of databases that are associated with the unlock codes you specified. You can set the attributes of the `Database` element to specify database details. The database settings include the country to which the database maps, type of database, the path to the location, and the preloading type. You must specify the attributes for at least one `Database` element if you want to use any functionality other than address parsing.

**Note:** The order in which you list the reference database file names in `SetConfig.xml` influences the loading of the database data into memory and can affect the operation of the Address Verification engine. For more information, see ["Reference Database Order in SetConfig.xml" on page 24](#).

If you send process calls to Address Verification without a valid unlock code in `SetConfig.xml`, Address Verification goes into the expired state and returns the following critical error message:

```
AD_SC_CERR_EXPIRED = -1601 - "The engine usage period has expired or is not activated yet"
```

If you receive this error message, you must add a valid unlock code and reinitialize Address Verification before you can start processing addresses.

When there are conflicting database elements and attributes, Address Verification assigns precedence to the first one in the sequence of `Database` elements. If you have a `Database` element with the country attribute set to `ALL`, include that as the last of the `Database` elements you have in `SetConfig.xml`.

## Reference Database Order in SetConfig.xml

At initialization time, Address Verification processes the reference databases in the order in which they appear in `SetConfig.xml`.

When you add database entries to `SetConfig.xml`, consider the following factors:

- Address Verification must have enough memory resources to load the files. The `MaxMemoryUsageMB` attribute specifies the amount of memory that Address Verification can request from the operating system.
- If you add multiple entries for the same database file to `SetConfig.xml`, Address Verification applies the first entry and disregards subsequent entries. Once Address Verification loads a file into memory, it does not load the file again in the current session.

For example, if `SetConfig.xml` contains an entry that specifies a preloading type of `NONE` for a database, and a subsequent entry specifies `FULL` for the same database, Address Verification does not load the database into memory.

- When you add database entries that specify different processing modes, the order of the entries can influence the engine performance. Add reference database entries to `SetConfig.xml` in the following order:
  - Any country-specific database that specifies certified mode.
  - Any country-specific database that specifies batch and interactive modes.
  - Any country-specific database that specifies fast completion mode.
  - Any country-specific database that specifies geocoding mode.
  - Any database entry that does not specify a country but rather uses `ALL` to indicate all countries.

## Preloading of Reference Address Databases

To optimize performance, you can preload reference address databases to the device memory (RAM). You can specify the preloading method and type in the `SetConfig.xml` file.

The `PreloadingMethod` attribute of the `General` element specifies how Informatica Address Verification preloads the reference address databases to the memory. The default method is `MAP`. Address Verification continues to support `LOAD` method for backward compatibility and fallback. However, for new installations you might want to use the `MAP` method, which is faster and efficient than the `LOAD` method.

If you set `PreloadingMethod` to `MAP`, Address Verification uses the file mapping mechanism of the operating system and reads the reference address database only once when you initialize Address Verification. If you set `PreloadingMethod` to `LOAD`, Address Verification uses a memory allocation call and loads the data to the allotted memory block. The `MAP` method requires more virtual memory because of larger alignment requirements of the operating system. However, the `MAP` method has the following advantages over the `LOAD` method:

- In multiprocess conditions, where multiple processes running Address Verification use a common set of `.MD` files, the operating system loads the data into the main memory only once and shares the preloaded reference address databases between separate processes.
- The operating system never writes the reference data contents to the paging file when the memory is low. Note that, in such cases, the data might get dropped from the file system cache. If the data is required at a later point in time, Address Verification reads the data from the system storage.



The `PreloadingType` attribute specifies the preloading type for each of the reference address databases. You can set `PreloadingType` to `FULL`, `PARTIAL`, or `NONE`. The default is `NONE`, which means no preloading of the database.

If you set `PreloadingType` to `FULL`, Address Verification copies the entire reference address database to the memory. The system must have a significant amount of memory to support `FULL` preloading of reference address databases, especially for countries such as the United States and the United Kingdom. `FULL` preloading of reference address databases significantly improves the performance over the other two preloading types, `PARTIAL` and `NONE`.

If you set `PreloadingType` to `PARTIAL`, Address Verification loads only the metadata and indexing structures into the memory. The reference databases remain on the system storage. `PARTIAL` preloading is an alternative when adequate memory is not available to fully load the required databases. `PARTIAL` preloading improves the performance over the preloading type `NONE`.

**Tip:** If you set the `PreloadingType` to `PARTIAL`, use a solid-state drive (SSD) to store the reference address databases as the SSD is much faster than hard-disk drives.

## Memory Management

Informatica Address Verification stores different types of objects, such as address objects, pre-loaded reference address databases, and caches, in its memory. When you make memory allocations for Address Verification, you must consider the different objects that have specific memory requirements.

You can divide the memory requirements of Address Verification into the following blocks:

- General memory block. Allocated for general management functions. Typically, the general memory block size is 7 MB.
- Thread memory block. Allocated for address processing and verification routines. Address Verification creates as many thread memory blocks as the number of simultaneous threads you configure it to process.  
The size of a thread memory block is about 38 MB for 32-bit systems and 48 MB for 64-bit systems.
- Address object memory block. Allocated for storing the address objects defined. Address Verification creates as many address object memory blocks as the number of address objects you configure it to process.  
The size of an address object memory block is about 3.7 MB + (0.24 MB x the value of `MaxResultCount`) in the case of 32-bit systems. For 64-bit systems, the size of an address object memory block is about 4.8 MB + (0.24 MB x the value set for `MaxResultCount`).
- Memory block reserved for caching. Address Verification reserves one cache memory block for each of the verification or processing threads.
- Memory blocks for preloading reference address databases. The value for preloading memory block depends on the number and size of the databases that you want to preload.
- Unallocated memory block.

You can configure the `MaxMemoryUsageMB` parameter to specify the maximum available memory for Address Verification. The values you set for the following parameters control the memory allocation for the memory blocks:

- `MaxThreadCount`. The maximum number of threads that Address Verification can process simultaneously. The value set for this parameter controls the number of thread memory blocks and the total memory allocation for the thread blocks.

- `MaxAddressObjectCount`. The maximum number of address objects that Address Verification can store. You can set a value that is two or three times the value you configured for `MaxThreadCount`. The value you set for `MaxAddressObjectCount` controls the number of address object memory blocks and the total memory allocation for the address objects.
- `CacheSize`. The memory reserved for caching purposes. If you set `CacheSize` to `None`, Address Verification does not allocate any memory for caching. If you set `CacheSize` to `Small`, Address Verification allocates 0.4 MB of cache memory block for each of the threads. If you set `CacheSize` to `Large`, Address Verification allocates 0.75 MB of cache memory block for each of the threads. For example, if `MaxThreadCount` is set to 4 and `CacheSize` to `Small`, Address Verification allocates 1.6 MB for cache memory block.

## Calculating Memory Requirements

You can use the following formula to calculate the maximum memory requirements:

```
The general memory block size + MaxAddressObjectCount x
(size of an address object memory block + MaxResultCount x the size of a result object)
+
MaxThreadCount x (the size of the thread block + the size of the cache memory block)
```

For example:

If the Informatica Address Verification configuration on a 32-bit system includes `MaxThreadCount=4`, `MaxAddressObjectCount=8`, `CacheSize=SMALL`, and `MaxResultCount=20`, you can calculate the dynamic memory requirement as follows:

$7 + 8 \times (3.7 + 20 \times 0.24) + 4 \times (38 + 0.4) = 228.6$  MB where 7 is the general memory block size; 3.7, the size of an address object memory block; 20, the value set for `MaxResultCount`; 0.24, the size of a result object; 4, the number of threads; 38 the size of the thread block; and 0.4, the cache memory block size when `CacheSize` is set to small.

To calculate the total memory requirement, add the total size of the reference address databases that you want to preload to the value you received based on the calculation in the previous paragraph.

If the Address Verification configuration on a 64-bit system includes `MaxThreadCount=6`, `MaxAddressObjectCount=6`, `CacheSize=LARGE`, and `MaxResultCount=100`, you can calculate the dynamic memory requirement as follows:

$7 + 6 \times (4.8 + 100 \times 0.24) + 6 \times (48 + 0.75) = 472.3$  MB where 7 is the general memory block size; 4.8, the size of an address object memory block; 100, the value set for `MaxResultCount`; 0.24, the size of a result object; 6, the number of threads; 48, the size of the thread block; and 0.75, the cache memory block size when `CacheSize` is set to large.

To calculate the total memory requirement, add the total size of the reference address databases that you want to preload to the value you received based on the calculation in the previous paragraph.

## Configuring General Settings for Informatica Address Verification

You can configure the general Informatica Address Verification settings in the `SetConfig.xml` file. The general settings include the XML encoding, the maximum number of address objects and threads, and the maximum memory allocation for Address Verification.

## The General Element

Configure the following attributes of the `General` element:

### **WriteXMLBOM**

Specifies when Informatica Address Verification should write a byte order mark (BOM) in the `GetConfig.xml` file. You can configure one of the following options:

- ALWAYS
- IF\_NECESSARY
- NEVER

The default value is IF\_NECESSARY.

### **WriteXMLEncoding**

Specifies the encoding for the XML files that Address Verification creates. To view the list of encoding types, see [“XML Encoding Types” on page 22](#).

### **CacheSize**

Specifies the size of memory cache to allocate for each of the address verification threads. You can configure one of the following options:

- NONE. Address Verification does not allocate any memory for caching.
- SMALL. Address Verification allocates 0.4 MB of memory for caching.
- LARGE. Address Verification allocates 0.75 MB of memory for caching.

The default value is LARGE.

If you set either SMALL or LARGE for the `CacheSize` attribute, Address Verification allocates one memory cache for each of the address processing threads. For example, if you set `MaxThreadCount` to 4, and `CacheSize` to SMALL, Address Verification allocates  $4 \times 0.4 = 1.6$  MB of memory for caching.

### **MaxThreadCount**

Specifies the maximum number of address processing threads that Address Verification can process simultaneously.

The default value is 1. You can set a minimum value of one, and a maximum value based on the system capabilities. There is no upper cap on the value that you can configure for this attribute. However, consider your system capabilities and number of cores available for Address Verification before you decide on a value for this attribute. For best results, you can consider a value that is one less than the total number of cores on the device.

For example, on a device that has four cores, you can set `MaxThreadCount` to 3.

### **MaxAddressObject**

Specifies the maximum number of address objects that Address Verification can process simultaneously.

You can set a minimum value of 1 and a maximum value based on the system capabilities. The default value is 1. Based on the system capabilities, you can configure a value that is 2 or 3 times the value you set for `MaxThreadCount`. For example, if you set `MaxThreadCount` to 2, the maximum value for `MaxAddressObjectCount` can be 4 or 6.

If you set `MaxAddressObjectCount` to a value less than that of `MaxThreadCount`, Address Verification reduces the value of `MaxThreadCount` to match the value you set for `MaxAddressObjectCount`.

**MaxMemoryUsageMB**

Specifies the maximum amount of memory, in MB, that Address Verification can request from the operating system.

The default value is 512 MB. Consider different memory requirements such as general memory block, thread memory block, caching memory, preloading memory block, and so on when you calculate `MaxMemoryUsageMB`. Address Verification fails to initialize if you set a value lower than the size of memory Address Verification requires.

**MaxResultCount**

Specifies the maximum number of suggestions Address Verification returns while verifying addresses in the Interactive mode. You can configure a value between 1 and 100. The default value is 20.

**PreloadingMethod**

Specifies the preloading method for reference address databases. You can configure one of the following options:

- MAP
- LOAD

MAP is the default value. The LOAD option is available for backward compatibility.

## Configuring Unlock Codes and Database Settings

Before you initialize Informatica Address Verification, you must specify at least one database and its unlock code in the `SetConfig.xml` file. The `UnlockCode` element and all the attributes of the `Database` element are required. The database configuration includes the database name, database type, path to the location where the database is saved, and the preloading type.

### The Unlock Code Element

Specifies an unlock code for accessing the reference address databases. An unlock code is a 36-character long alphanumeric string that you receive from Informatica Address Verification. You can repeat this element to specify multiple unlock codes.

### The Database Element

Configure the following attributes of the `Database` element:

**CountryISO3**

Specifies the ISO3 codes of the countries for which you want to validate addresses. You can specify ALL if you want to validate addresses from all supported countries and territories. If you specify ALL along with ISO3 codes for countries, enter ALL as the last entry in the list. You must have valid unlock codes for the countries that you specify for this attribute. This attribute is required. For example, DEU for Germany and USA for the United States.

**Type**

Specifies the type of reference address databases that you have unlock codes for. You can configure one of the following values:

- BATCH\_INTERACTIVE

- FASTCOMPLETION
- CERTIFIED
- GEOCODING
- GEOCODING\_ARRIVAL\_POINT
- GEOCODING\_PARCEL\_CENTROID
- GEOCODING\_ROOFTOP
- CAMEO
- SUPPLEMENTARY
- ADDRESS\_CODE\_LOOKUP

This attribute is required.

#### **Path**

Specifies the location where you saved the reference address databases.

#### **PreloadingType**

Specifies the preloading type for the database file. You can configure one of the following values:

- FULL
- PARTIAL
- NONE

This attribute is required.

Full preloading of databases requires significant amount of memory as the entire database is preloaded to the system memory. For example, if you want to preload the entire Address Verification reference databases for all supported countries, you require a system memory of about 40 GB.

## CHAPTER 3

# Input Parameters

This chapter includes the following topics:

- [Input Parameters Overview, 30](#)
- [Input Formatting Options, 30](#)
- [Configuring the Format Settings for Input Addresses, 31](#)
- [Input Country Settings, 31](#)
- [Configuring Country Settings for Input Addresses, 32](#)

## Input Parameters Overview

You can configure the attributes of the `Input` element in `Parameters.xml` to specify how Informatica Address Verification interprets input addresses.

Attributes of the `Input` element include the XML settings, formatting options, and country options.

You can configure the `WriteXMLBOM` attribute to specify whether and when Address Verification writes the byte order mark in the XML input.

You can set the `Encoding` attribute to specify the XML encoding to use in the input XML.

## Input Formatting Options

You can configure the formatting attributes of the `Input` element in `Parameters.xml` to specify the formatting of the input address.

You can set the `FormatType` attribute to specify what level of details the input address contains. For example, you can configure the `FormatType` attribute to specify whether the input address contains the contact information or organization information.

You can set the `FormatDelimiter` attribute to specify the element separator to use in an input address. For example, if you set `FormatDelimiter` to `COMMA`, Informatica Address Verification treats information on either side of a comma as separate address elements.

You can configure the `FormatWithCountry` attribute to specify whether the input address contains country information.

# Configuring the Format Settings for Input Addresses

You can configure the format attributes of the `Input` element in `Parameters.xml` to specify input address properties such as the level of details and the element delimiter.

In `Parameters.xml`, specify the following attributes of the `Input` element:

## **FormatType**

Specifies the level of details the input address contains. You can configure one of the following values:

- ALL
- ADDRESS\_ONLY
- WITH\_ORGANIZATION
- WITH\_CONTACT
- WITH\_ORGANIZATION\_CONTACT
- WITH\_ORGANIZATION\_DEPARTMENT

ALL is the default value.

## **FormatDelimiter**

Specifies the element separator. You can configure one of the following options:

- COMMA
- CR
- LF
- CRLF
- PIPE
- SEMICOLON
- SPACE
- TAB

CRLF is the default value. CRLF denotes carriage return and line feed.

## **FormatWithCountry**

Specifies whether the input address contains country information. You can configure one of the following values:

- ON
- OFF

OFF is the default value.

## Input Country Settings

You can configure the country attributes of the `Input` element in `Parameters.xml` to specify country-related information that Informatica Address Verification can use to process the input address.

You can set the `FormatWithCountry` attribute to specify whether the input address contains country information.

You can set the `DefaultCountryISO3` attribute to specify a country name that Address Verification can apply to input addresses that do not contain country information. An element result status value of 4 at positions 19 through 20 indicates that Address Verification recognized the country from the `DefaultCountryISO3` setting. In the country recognition mode, a process status code of **R9** indicates that Address Verification assigned the country based on the `DefaultCountryISO3` setting.

You can set the `ForceCountryISO3` attribute if you want Address Verification to replace any country information in the input address with country information that you specify. If you configure `ForceCountryISO3`, Address Verification validates all input addresses as addresses originating from the country specified for `ForceCountryISO3`. An element result status value of F at positions 19 through 20 indicates that Address Verification recognized the country from the `ForceCountryISO3` setting. In the country recognition mode, a process status code of **RA** indicates that Address Verification assigned the country based on the `ForceCountryISO3` setting.

## Configuring Country Settings for Input Addresses

You can configure the country attributes of the `Input` element in `Parameters.xml` to specify how Informatica Address Verification handles the country information in the input address.

Configure the following attributes of the `Input` element in `Parameters.xml`:

### **FormatWithCountry**

Specifies whether the input address contains country information or not. You can configure one of the following values:

- ON
- OFF

OFF is the default value.

### **DefaultCountryISO3**

Specifies the country to which the input address belongs. Set `DefaultCountryISO3` to the ISO3 code of the country to which the address belongs.

Address Verification uses this setting only in the following instances:

- The input address does not contain country information.
- The input address does not contain a city or locality name from which Address Verification can identify the country.

### **ForceCountryISO3**

Overrides the country information in the input address and assigns the country that you specify for this attribute to the validated output. Set this attribute to the ISO3 code of the country that you want Address Verification to assign to the input address.



## CHAPTER 4

# Process Parameters

This chapter includes the following topics:

- [XML Encoding for Parameters, InputData, and Results XML Files, 33](#)
- [Configuring XML Encoding for Parameters, Input, and Result, 34](#)
- [Process Modes, 34](#)
- [Specifying the Process Mode, 44](#)
- [Configuring Address Code Lookup, 45](#)
- [Configuring Certified-Mode Address Verification, 45](#)
- [Disabling the Certified Mode, 47](#)
- [Certified Mode Process Status Values, 47](#)
- [The MatchingExtendedArchive Attribute, 48](#)
- [The MatchingScope and MatchingAlternatives Attributes, 53](#)
- [Optimization Level, 54](#)

## XML Encoding for Parameters, InputData, and Results XML Files

In `Parameters.xml`, you can specify the XML encoding for `Parameters.xml`, `InputData.xml`, `Result.xml`, and any data that you input into Informatica Address Verification and any result that Address Verification returns.

For C-based APIs, you can use 8-bit or 16-bit functions. A `W` suffix in the function name, such as `AD_SetInputDataElementW()`, identifies a 16-bit function. If you use 16-bit functions, set the encoding to a 16-bit encoding. If you use a 16-bit function when the encoding is set to an 8-bit encoding, Address Verification returns an encoding error. When you use Java APIs, set the encoding to UTF-16 because Java supports only 16-bit input and output.

# Configuring XML Encoding for Parameters, Input, and Result

You can specify the XML encoding to use for parameters, input, and result in the `Parameters.xml` file.

To specify the XML encoding, you can configure the `WriteXMLEncoding` attribute of the following elements in `Parameters.xml`.

- **General.** To specify the XML encoding for the `Parameters.xml` file that Informatica Address Verification generates.
- **Input.** To specify the XML encoding for the `InputData.xml` file that Address Verification generates and any data that you input into Address Verification.
- **Result.** To specify the XML encoding for the `Result.xml` file that Address Verification generates and any result output that Address Verification returns.

For the list of encoding types that you can use in the `WriteXMLEncoding` attribute, see [“XML Encoding Types” on page 22](#).

If you use XML API and do not specify an encoding type, Address Verification sets the encoding to UTF-8 or UTF-16 based on whether you use 8-bit or 16-bit API functions. For direct API, the default is ISO-8859-1.

## Process Modes

The process mode that you select defines the way that Informatica Address Verification processes the input data.

The `AD_Process()` function requires the process mode information to process addresses. You can specify the process mode information in the `Parameters.xml` file. You can choose a process mode that includes address verification or a process mode that does not include address verification.

In the following process modes, Address Verification does not verify addresses:

- Address Code Lookup
- Country Recognition
- Parse

In the following process modes, Address Verification verifies addresses:

- Batch
- Interactive
- Fast Completion
- Certified

## Address Code Lookup

In the address code lookup (`ADDRESSCODELOOKUP`) process mode, you can enter an address code and retrieve the corresponding address. Based on the address code you enter and the availability of

corresponding reference data, Informatica Address Verification returns full or partial address information. The address code lookup process mode requires a separate unlock code.

In the address code lookup mode, you can retrieve addresses from the following countries:

- Germany
- Japan
- South Korea
- Serbia
- South Africa
- United Kingdom

To retrieve addresses in the address code lookup mode, you must install the corresponding address code lookup (AC) database and include the required unlock codes in `SetConfig.xml`. To retrieve address information in the address code lookup mode, you must also specify the address code type in the `InputData.xml` file.

The following table lists the address codes that you can use to retrieve addresses.

| Country | Address Code    | Description  |
|---------|-----------------|--|
| Germany | DEU_AGS         | <p>The Amtliche Gemeindeschlüssel (AGS) is a variable length code that uniquely identifies a locality in Germany. An AGS code might return more than one level of locality information.</p> <p>For example, a DEU_AGS code with a value of 07338018 fetches the following result:</p> <p>Locality: Maxdorf<br/>Province: Rheinland-Pfalz</p> |
| Germany | DEU_LOCALITY_ID | <p>The Locality ID is a variable length code that uniquely identifies a German locality.</p> <p>For example, a DEU_LocalityID code with a value of 68015519 fetches the following result:</p> <p>Locality: Maxdorf<br/>Province: Rheinland-Pfalz</p>   |
| Germany | DEU_STREET_ID   | <p>The Street ID is a variable length code that uniquely identifies a German street address.</p> <p>For example, a DEU_StreetID code with a value of 100560690 fetches the following result:</p> <p>Röntgenstr.<br/>67133<br/>Maxdorf<br/>Germany</p>  |

| Country      | Address Code               | Description  |
|--------------|----------------------------|--|
| Japan        | JPN_CHOUMEI_AZA_CODE       | <p>The Choumei Aza code is an 11-digit code that defines a unique delivery point for Japan addresses.</p> <p>For example, a JPN_CHOUMEI_AZA_CODE of 28201160001 fetches the following result:</p> <pre>〒 670-0081 兵庫県姫路市田寺東1丁目 or 01 Chome Taderahiga-shi Himeji-shi Hyogo-ken 670-0081 Japan</pre> <p>You can use the Choumei Aza code in address code lookup mode to find the current version of an address. For more information, see <a href="#">"Extended Archives and Japan Addresses" on page 49</a>.</p> |
| Japan        | JPN_CHOUMEI_AZA_GAIKU_CODE | <p>The Choumei Aza code is an 11-digit code that defines a unique delivery point for Japan addresses. The Gaiku code is a four-digit code that identifies a city block in Japan. The combined Choumei Aza code and Gaiku code helps you to retrieve more accurate addresses from Address Verification.</p>   |
| South Korea  | KOR_ADDRESS_ID             | <p>The address ID for South Korea addresses is a unique string that Address Verification assigns to a South Korea addresses. You can retrieve the address ID as an enrichment to archived South Korea addresses. You then use the address ID in the address code lookup mode to retrieve a different version of the address.</p> <p>For more information, see <a href="#">"Extended Archives and South Korea Addresses" on page 50</a>.</p>  |
| Serbia       | SRB_PAK                    | <p>The Postal Address Code (PAK) is a six digit code that defines a unique Serbian address to the street level.</p> <p>For example, a SRB_PAK code with a value of 251133 fetches the following result:</p> <pre>Majora Ilica 1 14000 Valjevo Serbia</pre>   |
| South Africa | ZAF_NADID                  | <p>The National Address Database (NAD) ID is a unique numeric ID assigned to a South Africa street address.</p> <p>For example, a ZAF_NADID code with a value of 2170232 fetches the following result:</p> <pre>4 Balmoral Road Vincent East London 5247 South Africa</pre>  |

| Country        | Address Code | Description  |
|----------------|--------------|--|
| United Kingdom | GBR_UDPRN    | <p>The Unique Delivery Point Reference Number (UDPRN) is an eight-character code that uniquely identifies a postal address in the Royal Mail PAF database.</p> <p>For example, a GBR_UDPRN code with a value of 15511432 fetches the following result:</p> <p>Flat 16<br/>Haden Court<br/>Lennox Road<br/>London<br/>N4 3HS<br/>United Kingdom</p> |
| United Kingdom | GBR_UPRN     | <p>Unique Property Reference Number (UPRN) is a numeric code that uniquely identifies a land or property unit in the United Kingdom. For example, a GBR_UPRN code with a value of 151117706 fetches the following result:</p> <p>2 Drumforber Cottage<br/>Laurencekirk<br/>AB30 1RS<br/>United Kingdom</p>   |

## Batch Mode

You can set the process mode to BATCH if you want to verify one or more addresses without any user intervention. In the batch mode, Informatica Address Verification processes and verifies the addresses that you entered and returns the results when the processing is over.

If you want to enter multiple addresses for verification and receive one result for each input address, you can choose the batch mode. If you have multiple addresses to verify, you can create a text file that contains the addresses and provide the file as the input.

In the batch mode, Address Verification does not correct addresses that are ambiguous or difficult to correct without user intervention. However, Address Verification returns parsed results for such addresses.

If the reference address database for the country to which the input address belongs is not available, Address Verification does not verify the address. In such cases, Address Verification processes the address in the PARSE mode and returns the parsed addresses instead of verified and corrected addresses.

## Certified Mode

You can use the certified (CERTIFIED) mode to verify addresses in accordance with the following postal certifications.

- Address Matching Approval System (AMAS) certification for Australia Post. Certified to AMAS Cycle 2016.
- Coding Accuracy Support System (CASS) certification for the United States Postal Services. Certified to Cycle N.
- SendRight certification for New Zealand Post. Certified to SendRight Cycle 2015.
- Service National de L'Adresse (SNA) certification for La Poste of France.  
Address Verification certifies addresses in France to the following levels:
  - CEDEX A. Organization level.

- Hexacle. House-number level.
- Hexaligne 3. Building level.
- Hexaposte. Post code level.
- Hexavia. Street level.
- Software Evaluation and Recognition Program (SERP) certification for Canada Post. Certified to SERP Cycle 2015.

An address that you want to verify in the certified mode must meet the formatting requirements that the postal authority of the country specifies. For example, to verify a Canada address in the certified mode, the input address can include a maximum of two delivery address lines (DALs). To verify a France address in the certified mode, the input address can include a maximum of six formatted address lines (FALs). Of the six FALs in France addresses, three FALs must contain delivery elements other than the post box and locality information.

For certified mode, you must have the certified databases (Cx) installed and the corresponding unlock codes specified in the `SetConfig.xml` file.

The certified mode falls back to the batch mode if the address belongs to a country that Informatica Address Verification does not support for the certified mode.

If you enable the certified mode and do not have the corresponding certified (Cx) reference address database, Address Verification returns an N2 error status.

If you want to disable the certified mode for some of the countries for which the certified mode is available, you can configure `DisableCertifiedModeISO3`. You can set the `DisableCertifiedModeISO3` attribute of the `Process` element to the ISO3 codes of the countries for which you want to disable the certified mode. If you disable the certified mode for some countries, Address Verification uses the batch mode to verify addresses from those countries.

## Certified Mode Verification of Australia Addresses

You can use the certified mode of Informatica Address Verification to verify Australia addresses in accordance with the AMAS guidelines. To verify Australia addresses in the certified mode, you must install the `AUS5C1.MD` and `AUS5C2.MD` databases and include the corresponding unlock codes in `SetConfig.xml`.

The `AUS5C1.MD` and `AUS5C2.MD` databases contain Postal Address File (PAF) data, which includes Australia Post's Delivery Point Identifiers (DPIDs).

## Certified Mode Verification of Canada Addresses

You can use the certified mode of Informatica Address Verification to verify Canada addresses in accordance with the SERP guidelines of Canada Post. To verify Canada addresses in the certified mode, you must install the `CAN5C1.MD` database and include the corresponding unlock code in `SetConfig.xml`. The `CAN5C1.MD` database contains the PoCAD (Point of Call Address Data).

Address Verification is SERP 2015 compliant. SERP 2015 compliance ensures that Address Verification adheres to the following changes to the postal rules and regulations set by Canada Post:

- If the input suite number is outside the only range available for the address in PoCAD, Address Verification marks that address as not valid. When the input postal code maps to a Large Volume Receiver (LVR), Address Verification copies the suite number from the input address to the output address even if the suite number does not match any database entry that contains the correct single suite-civic number combination.

- When the range-based PoCAD has only one address associated with a civic street and if the input address does not match the address in the database, Address Verification marks that address as not valid or noncorrectable.
- When the range-based PoCAD contains a Type 2 record that does not have a route identifier or a delivery mode identifier for a rural address, Address Verification handles that address in the same way it handles Type 1 addresses. However, the following conditions apply to the handling of rural civic addresses:
  - If the input address does not have a match in the range-based PoCAD and the postal code of the input address has a corresponding Type 4 address in the range-based PoCAD, Address Verification marks the address as VQ (Valid but questionable) in the SERP category enrichment.
  - If the input address is a rural address with a street having no civic street number, Address Verification adds the civic street number when a unique correction is possible. If no unique civic street can be added to the input address, Address Verification rejects that address.

Note that Post Office Box numbers from 99900 through 99905 in Canada denote Deliver to Post Office (DTPO) addresses for retail outlet locations. Addresses with Post Office Box numbers from 99900 through 99905 are specifically meant for parcel delivery and should not be used for other mail items such as letters and publications.

For more information about SERP 2015 certification, contact Canada Post.

## Certified Mode Verification of France Addresses

You can use the certified mode of Informatica Address Verification to verify the France addresses in accordance with the La Poste SNA guidelines. The integrators can use the certified mode validation type to develop SNA Application for certification by La Poste.

You do not require any special database files apart from `FRA5BI.md` for certified processing. For more information about the `FRA5BI.md` database, see [Chapter 4, "Process Parameters" on page 33](#).

You can query the `GetConfig.xml` file for the following `EnrichmentSupportInfo` element to check for correct initialization of all the required databases:

```
<EnrichmentSupportInfo CountryISO3="FRA" Type="CERTIFIED">FULL</EnrichmentSupportInfo>
```

For SNA certified processing, you must enter addresses in the six line FormattedAddressLine format, including empty lines wherever a part of the address is missing. The following table lists the values for the six line FormattedAddressLine format:

| Formatted Address Line | Value  |
|------------------------|--|
| Line 1                 | ORGANIZATION IDENTIFICATION or IDENTITY OF THE ADDRESSEE                                       |
| Line 2                 | INDIVIDUAL IDENTIFICATION (Company Contact) or DELIVERY POINT ACCESS INFORMATION (SubBuilding) |
| Line 3                 | DELIVERY POINT LOCATION (Building)   |
| Line 4                 | STREET NUMBER or PLOT and THOROUGHFARE   |
| Line 5                 | DELIVERY SERVICE or THOROUGHFARE COMPLEMENTARY IDENTIFICATION                                  |
| Line 6                 | POSTCODE and LOCALITY or CEDEX POSTCODE and DISTRIBUTION AREA INDICATOR                        |

## Certified Mode Verification of United States Addresses

You can use the certified mode to verify United States addresses in accordance with the CASS guidelines of the United States Postal Service. To verify United States addresses in the certified mode, you must install the United States Cx databases and include the corresponding unlock codes in `SetConfig.xml`.

The Cx databases for the United States are numbered `USA5C1.MD` through `USA5C25.MD`. The Cx databases for the United States contain CASS-related information such as Carrier Route codes, EWS, ZIPMOVE, LACSLink, DPV, DFS2, and SuiteLink.

**Note:** Certified mode verification of United States addresses is available only to customers in the United States.

The following list describes some of the Cx databases:

- The ZIPMOVE database contains information about past ZIP code changes. The ZIPMOVE database is required for CASS-certified address verification.
- The Early Warning System (EWS) database contains information about upcoming changes to ZIP codes. The EWS database is required for CASS-certified address verification.
- The LACSLink database is a secure database of addresses converted in the implementation of the 911 system, which includes changes from rural-style addresses to city-style addresses. The LACSLink database also contains city-style addresses that have been renamed or renumbered.

Effective from CASS Cycle L (2007-2009), LACSLink processing is required for CASS-certified address verification.

- The DPV database has information that helps you check whether a ZIP+4 coded address is in the USPS delivery file as a known address record. You can use the DPV product to confirm known USPS addresses and to identify potential addressing issues that might affect delivery. Note that even when you do not have the certified databases, Informatica Address Verification adds ZIP+4 Codes if `USA5BI.MD` is available.

Effective from CASS Cycle L (2007-2009), DPV processing is required for CASS-certified address verification.

- The SuiteLink database contains suite numbers for business addresses in selected high-rise buildings and targets high-rise addresses with high-volume default mail. The SuiteLink database improves the business addressing information through assignment of suite numbers. The `USA5C18.MD` file contains the latest data from the USPS. If the input address contains firm or company information, Address Verification initiates the SuiteLink verification.

Address Verification also sends records that have input suite data that do not match with the ZIP4 file thorough the SuiteLink process. If Address Verification finds a match for such addresses while verifying against SuiteLink, Address Verification retains the input suite data in the residue component and output in DAL2 as required by the USPS to retain the extraneous data.

Effective from CASS Cycle M (2009-2011), SuiteLink processing is required for CASS-certified address verification.

- The Residential Delivery Indicator (RDI) processing verifies whether the input data maps to a residence address or a business address. The `USA5C22.MD` and `USA5C23.MD` databases contain RDI processing-related information. RDI processing returns either Y or N based on the nature of the address. If Address Verification finds a ZIP9 or ZIP11 match for the input address, Address Verification marks the address as business delivery and returns a value of N. If Address Verification does not find a ZIP9 or ZIP11 for the input address, Address Verification marks the address as residential delivery and returns a value of Y. The RDI processing is useful for parcel shippers.



- The eLOT or the Enhanced Line of Travel database contains information that helps you sort the mail items in approximate carrier-casing sequence. The eLOT information is available in `USA5C24.MD` and `USA5C25.MD` databases. The eLOT information contains an eLOT sequence number field and an ascending or descending code. The eLOT sequence number indicates the first occurrence of delivery made to the add-on range within the carrier route. The ascending or descending code indicates the approximate delivery order within the sequence number.

## Country Recognition Mode

You can use the country recognition process mode to identify the country to which an address belongs. In the country recognition process mode, Informatica Address Verification returns the country information for addresses that do not contain the country information.

Address Verification identifies the country to which an address belongs based on one or more of the following criteria:

- An abbreviation in the input address. For example, the ISO3 code for a country, such as DEU for Germany, or a common abbreviation such as GER for Germany.
- The `DefaultCountryISO3` setting.
- The `ForceCountryISO3` setting.
- Country name in the input address.
- Major town, locality, or province information in the input address.  
Note that this works only if you enter the address in the formatted address line format.
- The script used in the input address.

Note that country recognition works best with addresses that are unambiguous.

## Fast Completion Mode

You can use the fast completion (FASTCOMPLETION) mode to receive address suggestions as you typeahead. You can use the fast completion mode when you want to generate suggestions based on truncated input.

The fast completion mode is particularly useful when you need to quickly enter addresses as you receive address data from the customers. In the fast completion mode, you can receive suggestions to complete partial inputs. Because the fast completion mode is optimized for verifying truncated inputs, the initial letters of an address element input are important in fetching the best results.

In the fast completion mode, you can enter addresses only as fielded input that uses the `AddressElement` items. In the fast completion mode, you cannot use the `AddressLine` input formats such as `FormattedAddressLine` and `DeliveryAddressLine` because Informatica Address Verification does not parse addresses that you enter for verification in the fast completion mode. However, you can use the `AddressComplete` input when you verify addresses from countries that Address Verification supports for single-line address verification.

In the fast completion mode, you can receive up to 100 suggestions based on the value you configured for the `MaxResultCount` parameter. The default value for `MaxResultCount` is 20. You can set the value for `MaxResultCount` in `SetConfig.xml` or in `Parameters.xml`.

**Note:** If you set `MaxResultCount` values in both `SetConfig.xml` and `Parameters.xml`, Address Verification uses the lower of the two values.

## Single Line Address Verification

You can use the single-line address verification to verify addresses entered into the `AddressComplete` element as a single line and receive suggestions to complete the address. Informatica Address Verification supports single line address verification as part of the fast completion mode.

To activate single-line address verification, you need a separate unlock code of type `SINGLE_LINE_VALIDATION`. Contact your sales representative for more information about obtaining the unlock code.

You can verify addresses from the following countries using single line address verification.

- Australia
- Canada
- Germany
- New Zealand
- United Kingdom
- United States

Address Verification identifies address elements in a single-line address input based on their position in the sequence you enter the address elements. So, it is imperative that you follow the order shown in the following table when you enter single-line addresses. When you enter an address in single line, ensure that you do not mix Delivery Address Line (DAL) elements and Country-Specific Location Line (CSLLN) elements.

The following table shows the order of address elements for the supported countries.

| Country        | Order of Address Elements  |
|----------------|--|
| Australia      | Sub-building, House Number, Street, Main Locality, Province, Postal Code                   |
| Canada         | Sub-building, House Number, Street, Delivery Service, Main Locality, Province, Postal Code |
| Germany        | Street, House Number, Postal Code, Locality, Province                                      |
| New Zealand    | Sub-building, House Number, Street, Delivery Service, Locality, Postal Code                |
| United Kingdom | Sub-building, House Number, Street, Main Locality, SubLocality, Postal Code                |
| United States  | Sub-building, House Number, Street, Locality, Province, Postal Code                        |

The typical sequence of address elements is from the specific to the generic. You must enter the elements in the specified sequence even if you leave out some of the elements from the input. However, for optimum results, you enter as many details as possible in the input address.

Even though delimiters are not required in a single-line address input, Address Verification processes a comma or semicolon in the input as an element separator. An input address with delimiters might fetch better suggestions than one without delimiters. Note that Informatica Address Verification Cloud does not support country, organization, building, or contact information in the single-line address input.

If the single-line address input contains only a numeric input, Address Verification considers it as the Postal Code and returns suggestions accordingly. For countries where the house number appears on the left side of the street name or locality, if the single-line address input begins with a number that is followed by a string, Address Verification considers the number as a house number and the following string as the street name or locality. If Address Verification does not find a match for this combination, Address Verification tries to interpret the input as street name without house number or as a combination of postal code and locality.

When there is no perfect match for an input, Address Verification returns multiple suggestions to help you choose the most appropriate result. Address Verification returns multiple suggestions based on the value you configure for `MaxResultCount` in `Parameters.xml` and `SetConfig.xml` files.

Address Verification returns an N6 process status code if single line address verification is not available for the country to which the input address belongs. Address Verification returns an N7 process status code if you have not provided the unlock key for single line address verification in `SetConfig.xml`.

## Interactive Mode

You can use the interactive (INTERACTIVE) mode to manually resolve ambiguities and select the best possible corrections. In the interactive mode, Informatica Address Verification returns multiple address suggestions from which you can choose the most appropriate result.

For each address that you enter for verification in the interactive mode, you can receive a maximum of 100 suggestions from Address Verification. You can configure the `MaxResultCount` parameter in `SetConfig.xml` and `Parameters.xml` to specify the maximum number of suggestions that you want to receive for each verification request. The default value for `MaxResultCount` is 20. If you set values for `MaxResultCount` in both `SetConfig.xml` and `Parameters.xml`, Address Verification uses the lower of the two values.

The interactive mode is particularly useful when you work with addresses that are ambiguous. You can view possible corrections for the address you entered and choose the correct address to ensure high-quality address data in your master database or CRM systems.

If the reference address database for the country to which the address belongs is not available, Address Verification uses the PARSE mode. In such cases, Address Verification returns the parsed address instead of suggestions or corrections.

## Key Differences Between Fast Completion and Interactive Modes

The fast completion and interactive modes differ in the following aspects:

### Database

Fast completion mode uses the Fast Completion (FC) reference address databases.

Interactive mode uses the Batch and Interactive (BI) reference address databases.

### Partial Input and Typeahead Support

Fast completion mode is optimized to work with truncated input data, and supports typeahead functionality. You receive suggestions as you enter an address. In the fast completion mode Informatica Address Verification returns quality results for partial address input. However, in the interactive mode, Address Verification requires almost complete address to validate.

### Post Code Lookup

In the fast completion mode, you can search for address suggestions by entering minimum information such as post codes. For many countries, you can enter a post code and receive a list of addresses that belong to that post code. In the interactive mode, Address Verification requires almost complete address input to generate any suggestion.

### Address Input Format

In the interactive mode, you can enter fielded, partially-fielded, or unfielded addresses. In the fast completion mode, Address Verification supports only fielded addresses.

## Enrichments

In the interactive mode, you can configure Address Verification to return enrichments such as country-specific enrichments or CAMEO codes. Address Verification does not support enrichments in the fast completion mode.

## Parse Mode

You can use the parse (PARSE) mode to separate out address information from fielded or unfielded address inputs. In the parse mode, Informatica Address Verification does not validate the address.

Address parsing in Address Verification functions independent of the reference address databases. You can use the PARSE mode to parse addresses even if you do not have the corresponding reference address databases. Address Verification does not use the reference address databases to process an address in the parse mode. So, the address element assignment in the parse mode might vary with the address element assignment in a validated output.

In the parse mode, Address Verification separates address input into tokens that can be used to process the address further. You can use the parse mode if the input is not correctly formatted or you need to segregate address items in an input address. Note that the parse mode does not work if Address Verification does not recognize the country to which the input address belongs.

## Specifying the Process Mode

You can set the `Mode` attribute of the `Process` element in `Parameters.xml` to specify the process mode of Informatica Address Verification. By default, the `Mode` attribute is set to `BATCH`.

You must install the corresponding reference address databases and specify the unlock codes in the `SetConfig.xml` file. For example, if you want to validate the U.S. addresses in batch mode, you must install the batch and interactive (BI) reference address database for the U.S. addresses and include its unlock code in `SetConfig.xml`. Similarly, to validate the U.S. addresses in the certified mode, you must install the certified reference address databases (Cx) for the U.S., and include the unlock codes in `SetConfig.xml`.

- In `Parameters.xml`, set the `Mode` attribute of the `Process` element to one of the following values based on your requirement.
  - ADDRESSCODELOOKUP
  - BATCH
  - CERTIFIED
  - COUNTRYRECOGNITION
  - FASTCOMPLETION
  - INTERACTIVE
  - PARSE

# Configuring Address Code Lookup

You can use the ADDRESSCODELOOKUP mode to retrieve address information from Informatica Address Verification by entering address codes. To use the address code lookup mode, you must install the address code lookup (AC) databases.

1. In `Parameters.xml`, set the `Mode` attribute of the `Process` element to ADDRESSCODELOOKUP.
2. In `InputData.xml`, set the `Type` attribute of the `AddressCode` element to one of the following values to specify the address code type for which you want to retrieve addresses:
  - DEU\_AGS
  - DEU\_LOCALITY\_ID
  - DEU\_STREET\_ID
  - GBR\_UDPRN
  - GBR\_UPRN
  - JPN\_CHOUMEI\_AZA\_CODE
  - JPN\_CHOUMEI\_AZA\_GAIKU\_CODE
  - KOR\_ADDRESS\_ID
  - SRB\_PAK
  - ZAF\_NADID

# Configuring Certified-Mode Address Verification

You can configure Informatica Address Verification to verify addresses in certified mode that complies with the postal regulations of a country. Address Verification supports certified-mode address verification for Australia, Canada, France, New Zealand, and the United States. Before you enable the certified mode, you must install the certified (Cx) reference address databases and include the unlock codes in the `SetConfig.xml` file.

In `Parameters.xml`, configure the following attributes of the `Process` element:

1. To enable certified-mode address verification, set the `Mode` attribute to CERTIFIED.
2. To enable the certified mode verification of the United States addresses, set the `EnrichmentCASS` attribute to ON .  
The default value is OFF.
3. To enable the certified mode verification of Canada addresses, set the `EnrichmentSERP` attribute to ON .  
The default value is OFF.
4. To enable the certified mode verification of France addresses, set the `EnrichmentSNA` attribute to ON .  
The default value is OFF.
5. To enable the certified mode verification of Australia addresses, set the `EnrichmentAMAS` attribute to ON .  
The default value is OFF.
6. To enable the certified mode verification of New Zealand addresses, set the `EnrichmentSendRight` attribute to ON .  
The default value is OFF.

For best results in the certified mode, set following attributes to the default values:

- The `PreferredLanguage` attribute of the `Result` element. The default value is `DATABASE`.
- The `MatchingAlternatives` attribute of the `Process` element. The default value is `ALL`.
- The `MatchingScope` attribute of the `Process` element. The default value is `ALL`.
- The `OptimizationLevel` attribute of the `Process` element. The default value is `STANDARD`.
- The `GlobalMaxLength` attribute of the `Result` element.
- The `GlobalCasing` attribute of the `Result` element. The default value is `NATIVE`.
- Attributes of the `AddressElementStandardize` element.

## Reference Address Databases for the Certified Mode

Informatica Address Verification supports the following reference address databases for the certified mode address verification:

### Australia

`AUS5C1.MD`

`AUS5C2.MD`

`AUS5CICF.MD`

### Canada

`CAN5C1.MD`

### France

Address Verification does not provide a separate certified mode reference address database for France. Use the batch and interactive mode database `FRA5BI.MD` to process France addresses in the certified mode.

### New Zealand

`NZL5C1.MD`

### United States

`USA5C1.MD` through `USA5C25.MD`.

# Disabling the Certified Mode

You can disable the certified (CERTIFIED) mode for one or more of the countries for which the mode is available. If you disable the certified mode for a country, Informatica Address Verification uses the BATCH mode to verify addresses from that country.

- In `Parameters.xml`, set the `DisableCertifiedModeISO3` attribute of the `Process` element to the ISO3 codes of the countries for which you want to disable the certified mode.

**Note:** Address Verification returns an N2 status message when all of the following conditions are true:

You set the process mode to CERTIFIED.

You do not have the Cx databases for one or more countries for which the certified mode verification is available.

You did not disable the certified mode for countries for which you do not have the Cx databases.

You enter an address from a country for which the certified mode is available but you do not have the Cx database.

## Certified Mode Process Status Values

When you validate addresses in the certified mode, Informatica Address Verification returns process status values that are specific for the certified mode you selected. The certified mode status values help you understand and analyze the results you receive in the certified mode.

### AMAS Status Codes

When you validate Australia addresses in the certified mode, Informatica Address Verification returns the following status codes along with the address output.

#### **EAM0**

AMAS output is not available for the input address.

#### **EAM1**

AMAS output for the input address is provided. The address is corrected or validated and DPID for the address is provided.

#### **EMA2**

AMAS output is not provided. The input address could not be corrected or validated, and so no DPID is provided.

### CASS Status Values

When you verify United States addresses in the certified mode, Informatica Address Verification returns the following status values to help you interpret and analyze the verification results.

#### **ECA0**

CASS output is not available for the input address.

#### **ECA1**

CASS output is not complete for the input address.

**ECA2**

This code is reserved for future use.

**ECA5**

CASS output provided for the input address.

## SendRight Status Codes

When you validate New Zealand addresses in the certified mode, Informatica Address Verification returns the following status values along with the address output.

**ESR0**

SendRight output is not available for the input address.

**ESR1**

SendRight output is provided for the input address.

## SERP Status Codes

When you validate Canada addresses in the certified mode, Informatica Address Verification returns the following status codes to help you interpret and analyze the SERP results.

**ESE0**

SERP output is not available for the input address.

**ESE1**

SERP output is provided.

If you set the process mode to certified and the SERP enrichment to on, Address Verification provides two enrichments: Category and Excluded Flag.

## SNA Status Codes

When you validate France addresses in the certified mode, Informatica Address Verification returns the following status values along with the address output.

**ESN0**

SNA output not available for the input address.

**ESN1**

SNA output for the input address is provided.

## The MatchingExtendedArchive Attribute

You can configure Informatica Address Verification to obtain the current version of an older address in Japan and South Korea.

Use the MatchingExtendedArchive attribute of the Process element to configure Address Verification to verify older addresses. To verify a non-current address, set the MatchingExtendedArchive attribute of the Process element to ON.



For Japan addresses, configure the `MatchingExtendedArchive` attribute to obtain the new Choumei Aza code for an older address. You can then enter the new Choumei Aza code in address code lookup mode to retrieve the current version of the address. Address Verification returns the address and the new Choumei Aza code that Japan Post issued when the older address was updated. To receive Choumei Aza codes in the address output, you must enable Japan address enrichments and you must install the Japan archive address and enrichment databases.

**Note:** A Japan address can have more than one older version. Because Address Verification returns the address and new Choumei Aza code that Japan Post issued when the older address was updated, the output data might not reflect the current address. You can use the new Choumei Aza code that Address Verification returned to retrieve the current Choumei Aza code and address. For information on how to retrieve the Choumei Aza code for an address that has multiple older versions, see [“Using the Extended Archives to Update a Japan Address” on page 50](#).

For South Korea addresses, configure the `MatchingExtendedArchive` attribute to validate older, land lot-based addresses and older, six-digit post codes. You retrieve a corresponding address ID value as an enrichment. You can then enter the address ID in address code lookup mode to retrieve the current, street-based version of the address and the current, five-digit post code. To retrieve the address ID, you must enable South Korea address enrichments and you must install the South Korea enrichment database. For information on how to retrieve the current, street-based version of a South Korea address, see [“Extended Archives and South Korea Addresses” on page 50](#).

If you set `MatchingExtendedArchive` to OFF, Address Verification considers only the current, street-based address and the current post code as valid. To evaluate the addresses or post codes in the older formats, set `MatchingExtendedArchive` to ON.

If you set `MatchingExtendedArchive` to ON, Address Verification returns an Extended Element Result status of F for a lot-based address. If you set the `MatchingExtendedArchive` attribute to OFF, Address Verification rejects or corrects outdated lot-based addresses.

## Extended Archives and Japan Addresses

A Japan address might have a current version and one or more older versions. You can use Address Verification to obtain the current version of an older address. Use the Choumei Aza codes to obtain the current version.

The Japan archive and enrichment databases can contain multiple versions of an address. When you verify an older Japan address, Address Verification returns the next version of the address and the next version of the Choumei Aza code in the database. However, if Japan Post updated the address more than once, the address and the new Choumei Aza code that Address Verification returns might not reflect the current address data. Enter the new Choumei Aza code that Address Verification returns in address code lookup to retrieve any later version of the address and the corresponding Choumei Aza code. The output address from Address Verification is the current address when the Choumei Aza code input value and the new Choumei Aza code output value are identical.

To obtain the current address information, set the `MatchingExtendedArchive` attribute of the `Process` element to ON.

The following table shows a Japan address with two older versions and the Choumei Aza codes for each version:

| Address Version | Address                 | Choumei Aza Code | New Choumei Aza Code |
|-----------------|-------------------------|------------------|----------------------|
| Oldest          | 〒 587-0002 大阪府南河内郡美原町黒山 | 27385006000      | 27201334000          |
| Intermediate    | 〒 5870002 大阪府堺市美原町黒山    | 27201334000      | 27147007000          |
| Current         | 〒 587-0002 大阪府堺市美原区黒山   | 27147007000      | 27147007000          |

The address is the current address when the Choumei Aza codes match.

## Using the Extended Archives to Update a Japan Address

To update an older Japan address, first validate the address in batch or interactive mode. Address validation returns a Choumei Aza code for the address. Enter the Choumei Aza code in address code lookup to retrieve the newer address. You might need to perform some steps more than once.

Before you update the address, set the `MatchingExtendedArchive` attribute of the `Process` element to ON. To obtain the current address and the current Choumei Aza code for an older address, perform the following steps:

1. Enter a valid older address to Address Verification in batch or interactive mode.  
Address Verification returns a validated address and a new Choumei Aza code.
2. Configure Address Verification in address code lookup mode, and enter the Choumei Aza code that you received in the output from the previous step.  
Address Verification returns a newer version of the address and the Choumei Aza code for the newer address.
3. Compare the input Choumei Aza code and the output Choumei Aza code from the operation in step 2.
  - If the input and output codes match, the output address and the output Choumei Aza code represent the current version of the address.
  - If the input and output codes do not match, repeat step 2.

When you repeat step 2, use the most recent Choumei Aza code that Address Verification returns. You might need to repeat the steps more than once.

## Extended Archives and South Korea Addresses

A current South Korea address uses street-based information to identify the address location. An older South Korea address uses lot-based information to identify the address location. You can use Address Verification to obtain the current version of an older address.

Additionally, current post codes in South Korea contain five-digits. Older post codes in South Korea contain six-digits. You can use Address Verification to obtain the current version of an older post code.

To obtain the current address or post code, you must first verify the address in batch or interactive mode. Address Verification returns an address ID value that you can use in address code lookup mode to find the current address data. If you enter a lot-based address or a six-digit post code as the input, set the `MatchingExtendedArchive` attribute of the `Process` element to ON.

**Note:** The address ID value (ADDRESS\_ID) is a proprietary value that Address Verification uses to retrieve different versions of the address data. The address ID value does not add any information to the address outside of its use in address code lookup mode.

### Combinations of Old and New Information in South Korea Addresses

A South Korea address can combine current and non-current post codes with street-based and lot-based address information. A lot-based address might have an older or newer post code, and a street-based address might have an older or newer post code. You can use Address Verification to obtain different combinations of the address and the post code. The steps that you take depend on the combination of address data values in the input address and the versions of the address data that you require.

The following table shows the different types of address that you can verify:

| Address Type                       | Address Information         |
|------------------------------------|-----------------------------|
| Lot-based,<br>old post code        | 경기도 의왕시 삼동 559<br>437-848   |
| Lot-based,<br>current post code    | 경기도 의왕시 삼동 559<br>16100     |
| Street-based,<br>old post code     | 경기도 의왕시 까치골1길 36<br>437-848 |
| Street-based,<br>current post code | 경기도 의왕시 까치골1길 36<br>16100   |

### Final Characters in the Address ID Value

When you verify an address in batch or interactive mode, Address Verification returns an address ID with the final character A, B, C, or D. The final character identifies the types of information that the address contains.

The final character identifies the following combinations:

- A: Current post code, current street-based address
- B: Current post code, lot-based address
- C: Old post code, current street-based address
- D: Old post code, lot-based address

To obtain a different combination of address data, replace the final character in the address ID value with the character that represents the combination that you require. Then enter the modified address ID in address code lookup mode.

### Building Data and Sub-Building Data

Address Verification does not verify building or sub-building data in a South Korea address. If you enter an address that contains building or sub-building data in batch or interactive mode, Address Verification copies the data to the output address.

Address Verification does not return building or sub-building data for a South Korea address in address code lookup mode. If you enter an address ID code for a South Korea address that contains building or sub-building data, you must add the relevant information to the output address. The building or sub-building data that you add to the output address must match the output address format.

## Using the Extended Archives to Update a South Korea Address

To update the information for a South Korea address, first verify the address in batch or interactive mode. The verified output address includes an address ID value. Enter the address ID value in address code lookup mode to obtain the address information that you need.

Before you update lot-based information or a six-digit post code, set the `MatchingExtendedArchive` attribute of the `Process` element to ON.

To retrieve the street address data or the five-digit post code for an address, perform the following steps:

1. Enter a South Korea address in batch or interactive mode. You can enter a street address or a lot-based address.

Address Verification returns the validated address data and an address ID value.

**Note:** Step 1 does not update a lot-based address to a street address or a six-digit post code to a five-digit post code.

2. Review the address ID value in the output from step 1.

The final character of the address ID value indicates the type of information in the output address. Replace the final character in the ID value with the character that represents the address data that you need.

Use one of the following characters:

- A. Returns the current post code and the current street-based address.
- B. Returns the current post code and the lot-based address.
- C. Returns the old post code and the current street-based address.
- D. Returns the old post code and the lot-based address.

3. Configure Address Verification in address code lookup mode.
4. Enter the address ID value that you modified.

Address Verification returns the address information that corresponds to the address ID value.

5. If the address that you entered in step 1 contains building or sub-building information, copy the information from the step 1 address to the final output address.

### Example: Old Address to Current Address

The following address contains lot-based information and a non-current, six-digit post code:

```
247 Dangjeong-dong  
Gunpo-si, Gyeonggi-do 435-713  
Korea
```

Address Verification returns the following address ID value for the address:

```
9899900BE6A56708D9D663275D
```

To obtain the current, street-based information and the current post code for the address, change the D in the address ID value to A. By default, Address Verification returns the address in the Hangul script.

The `PreferredScript` attribute determines the script for the output address. To return the address in the Latin script, set the `PreferredScript` attribute to ASCII. Find the `PreferredScript` attribute in the `Result` element of the `Parameters.xml` file.

The following table shows the output address in the Hangul and Latin scripts when you change the address ID character to A:

| Output Address in the Hangul Script | Output Address in the Latin Script           |
|-------------------------------------|--|
| 경기도 군포시 농심로 35<br>15842             | 35 Nongsim-ro<br>Gunpo-si, Gyeonggi-do 15842 |

## The MatchingScope and MatchingAlternatives Attributes

You can configure the level of address matching that you want Informatica Address Verification to perform while it verifies an address. The `MatchingScope` attribute defines the level of granularity to which Address Verification verifies an address. The `MatchingAlternatives` attribute specifies whether to consider alternative names while verifying an address.

You can set the `MatchingScope` attribute of the `Process` element in `Parameters.xml` to specify the address elements that you want Address Verification to verify. You can set the matching scope to locality, street, or delivery-point level. The default value of `MatchingScope` is `ALL` so that Address Verification verifies all possible elements including locality, street, and delivery-point elements in the input address. You receive the best results if you set the `MatchingScope` attribute to `ALL` as Address Verification verifies all possible elements. If you do not want Address Verification to consider some of the address elements for verification, you can set `MatchingScope` to a value other than `ALL`.

If the reference address database does not contain the specified level of information for an address, Address Verification verifies only those elements that the database contains.

You can configure Address Verification to verify address elements based on synonyms or alternative names in the reference address database. Synonyms of an address element include alternative names of a place or a street, such as vanity names in United States addresses. The archival name of an address element refers to an outdated name of a place or street.

To match input address elements with synonyms or archival names in the reference address database, set the `MatchingAlternatives` attribute to `SYNONYM_ONLY`, `ARCHIVE_ONLY`, or `ALL`. The default value of `MatchingAlternatives` is `ALL`. You can set the `MatchingAlternatives` attribute to `NONE` if you do not want Address Verification to consider synonyms or archival names when it verifies an address.

### Configuring the Verification Scope

You can specify the address elements that you want Informatica Address Verification to verify in an input address. Address Verification verifies only the elements that meet the specified scope.

- In `Parameters.xml`, set the `MatchingScope` attribute of the `Process` element to one of the following values:
  - **LOCALITY\_LEVEL.** To ignore information other than the Locality and Postal Code information while verifying the address.
  - **STREET\_LEVEL.** To verify up to the street information including the street name along with the `LOCALITY_LEVEL` elements. If you set `MatchingScope` to `STREET_LEVEL`, Address Verification ignores information such as house number and subbuilding.

- **DELIVERYPOINT\_LEVEL.** To verify up to the delivery point information including house number and post box number and ignore further elements such as building, subbuilding, and organization.
- **ALL.** To verify all elements, including street, locality, and delivery point information, that Address Verification can verify.

## Configuring the MatchingAlternatives Attribute

To specify whether Informatica Address Verification should match input address elements with synonyms or archived names in the reference address database, configure the `MatchingAlternatives` attribute of the `Process` element in `Parameters.xml`.

- In `Parameters.xml`, set one of the following values for the `MatchingAlternatives` attribute of the `Process` element.
  - ALL
  - ARCHIVE\_ONLY
  - NONE
  - SYNONYM\_ONLY

The default value is ALL.

## Optimization Level

You can configure the `OptimizationLevel` attribute of the `Process` element in `Parameters.xml` to choose between processing speed and quality of verified output. You can set `OptimizationLevel` to one of the following values:

### NARROW

Informatica Address Verification honors input assignment strictly, with the exception of separation of House Number from Street information.

### STANDARD

Address Verification makes the following element separations.

- Province from Locality information
- Postal Code from Locality information
- House Number from Street information
- SubBuilding from Street information
- DeliveryService from Street information
- SubBuilding from Building information
- Locality from PostalCode information

### WIDE

Address Verification does element separation in the same way it does element separation when you set `OptimizationLevel` to STANDARD. Additionally, Address Verification verifies six parsing candidates to increase the scope of verification and takes additional reference data entries into account for matching.

Consider the following factors when you set the optimization level:

- If you set the optimization level to **WIDE**, Address Verification might return addresses with extensive corrections. Extensive corrections to an address might adversely affect the result quality.
- If you set the optimization level to **NARROW**, Address Verification requires the minimum processing resources. If you set the optimization level to **WIDE**, Address Verification requires the maximum processing resources.
- The `OptimizationLevel` setting does not have any impact if the required reference data is not available for the country to which the input address belongs.

## Configuring the Optimization Level

You can specify the level of optimization that you want Informatica Address Verification to apply while parsing address elements for verification. The wider the level of optimization, the more matches and results Address Verification returns.

- In `Parameters.xml`, set the `OptimizationLevel` attribute of the `Process` element to one of the following values.
  - **NARROW.** To strictly follow the input assignment of address elements. The only change is to separate house number from the street information.
  - **STANDARD.** To parse and independently verify address elements including postal code, locality, province, subbuilding, and house number information. This is the default value for the `OptimizationLevel` attribute.
  - **WIDE.** To parse and independently verify all possible elements.

## CHAPTER 5

# Address Enrichments

This chapter includes the following topics:

- [CAMEO Consumer Segmentation, 56](#)
- [GeoCoding Types, 61](#)
- [Country-Specific Address Enrichments, 67](#)
- [Enrichments for Austria Addresses, 68](#)
- [Enrichment for Belgium Addresses, 69](#)
- [Enrichments for Brazil Addresses, 70](#)
- [Enrichments for France Addresses, 71](#)
- [Enrichments for Germany Addresses, 72](#)
- [Enrichments for Japan Addresses, 74](#)
- [Enrichments for South Korea Addresses, 75](#)
- [Enrichments for Poland Addresses, 76](#)
- [Enrichments for Russia Addresses, 77](#)
- [Enrichments for Serbia Addresses, 78](#)
- [Enrichments for South Africa Addresses, 79](#)
- [Enrichments for Switzerland Addresses, 79](#)
- [Enrichments for the United Kingdom Addresses, 80](#)
- [Enrichments for the United States Addresses, 82](#)

## CAMEO Consumer Segmentation

CAMEO profiles from the Callcredit Information Group provide consumer classification systems that indicate the socio-economic and geo-demographic profiles of neighborhoods across the world. You can retrieve CAMEO profiles as an address enrichment for many countries.

To receive CAMEO information in the validated addresses, you must install the CAMEO reference address databases and include the corresponding unlock codes in `SetConfig.xml`.

**Note:** Informatica Address Verification adds CAMEO information only to validated addresses. To enable CAMEO enrichment, you must install the Batch and Interactive reference address databases along with the CAMEO databases. You cannot retrieve enrichments in the fast completion mode.

The CAMEO profiles contain the latest socio-demographic and lifestyle data at microcell level based on parameters such as age, education, income, and general interests.



The following list shows some of the use cases for CAMEO profiles.

- Enhance and segment consumer databases
- Improve your understanding about customers and responders
- Find more prospects by finding look-alikes
- Perform area and location analysis
- Understand market potential
- Perform advanced statistical analysis and modeling

You can configure Address Verification to include CAMEO information in the validated output for the following countries:

- Australia
- Austria
- Belgium
- Brazil
- Canada
- Czech Republic
- Denmark
- Estonia
- Finland
- France
- Germany
- Hong Kong
- Hungary
- Indonesia
- Ireland
- Italy
- Japan
- Mexico
- Netherlands
- New Zealand
- Norway
- Poland
- Portugal
- Romania
- Russian Federation
- Singapore
- Slovakia
- South Africa
- Spain
- Sweden
- Switzerland

- United Kingdom
- United States

## Enabling CAMEO Demographic Profiles

You can configure Informatica Address Verification to include CAMEO demographic profile information as an enrichment to addresses. Before you enable CAMEO enrichment, you must install CAMEO databases and include their unlock codes in the `SetConfig.xml` file.

- In `Parameters.xml`, set the `EnrichmentCAMEO` attribute of the `Process` element to `ON`.  
The default value of this attribute is `OFF`.

## CAMEO Reference Address Databases

Install one or more of the following reference address databases to retrieve CAMEO data as an address enrichment:

### Australia

`AUS5CA.MD`

### Austria

`AUT5CA.MD`

### Belgium

`BEL5CA.MD`

### Brazil

`BRA5CA.MD`

### Canada

`CAN5CA.MD`

### Czech Republic

`CZE5CA.MD`

### Denmark

`DNK5CA.MD`

### Estonia

`EST5CA.MD`

### Finland

`FIN5CA.MD`

### France

`FRA5CA.MD`

### Germany

`DEU5CA.MD`

### Hong Kong

`HKG5CA.MD`

**Hungary**

HUN5CA.MD

**Indonesia**

IDN5CA.MD

**Ireland**

IRL5CA.MD

**Italy**

ITA5CA.MD

**Japan**

JPN5CA.MD

**Mexico**

MEX5CA.MD

**Netherlands**

NLD5CA.MD

**New Zealand**

NZL5CA.MD

**Norway**

NOR5CA.MD

**Poland**

POL5CA.MD

**Portugal**

PRT5CA.MD

**Romania**

ROU5CA.MD

**Russian Federation**

RUS5CA.MD

**Singapore**

SGP5CA.MD

**Slovakia**

SVK5CA.MD

**South Africa**

ZAF5CA.MD

**Spain**

ESP5CA.MD

**Sweden**

SWE5CA.MD

**Switzerland**

CHE5CA.MD

**United Kingdom**

GBR5CA.MD

**United States**

USA5CA.MD

## CAMEO Output Fields

When you validate an address with CAMEO enrichment enabled, you receive the following information in the validated output.

**CATEGORY**

A code and description for the age and affluence of the address at the country level

**GROUP**

A code and description for the neighborhood of the address at the country level.

**INTERNATIONAL**

A code and description of the age and affluence at an International level

**MVID**

A match key that you can use to link your CAMEO encoded addresses to CAMEO Analysis, a Callcredit Information Group product. CAMEO Analysis is a separate product that you can license directly from Callcredit Information Group.

## CAMEO Status Values

If you enable CAMEO enrichment, Informatica Address Verification returns the following status codes in the validated output to help you interpret or analyze the CAMEO information.

**ECON**

The validated address does not contain any CAMEO code because the CAMEO database is not available for the selected country

**EC0I**

The validated address does not contain any CAMEO code because no CAMEO lookup was done for the address because Address Verification could not correct the address and returned an Ix process status.

**EC00**

The validated address does not contain any CAMEO code because Address Verification could not find a CAMEO code for the input address.

**EC01**

The validated address contains CAMEO code.

# GeoCoding Types

You can configure Informatica Address Verification to add geocoordinates as an enrichment to a verified address. Geocoordinates indicate the longitude and latitude of an address. Address Verification follows the World Geodetic System standards (WGS 84) for geocoding.

You can enable geocoding at the global level and specify the type of geocoding that you want to receive in the output. To enable geocoding, install one or more geocoding databases.

You can configure Address Verification to return the following types of geocoordinates in the verified output:

## **Arrival Point GeoCoding**

Arrival point geocoordinates map to a point in the center of the street segment in front of a house or a building. Arrival point is the default geocoding type.

To retrieve arrival point geocoding, install the Arrival Point database (GCAP) for the country in question. If the arrival point geocoordinates do not exist, Address Verification returns the standard geocoordinates. To fall back to the standard geocoordinates, you must install the standard geocoding (GC) database. Otherwise, Address Verification returns the EGC0 (no geocode available) status code for the address.

## **Parcel Centroid GeoCoding**

Parcel centroid geocoordinates map to a point at the geographic center of the parcel of land that the address represents. To retrieve parcel centroid geocoding, install the Parcel Centroid database (GCCP) for the country in question. If the parcel centroid geocoordinates do not exist, Address Verification returns the status code EGC0 to indicate that geocoordinates are unavailable.

## **Rooftop GeoCoding**

Rooftop geocoordinates map to the center of the roof of the primary building on a parcel of land. Rooftop geocoordinates pinpoint an address with the highest level of precision. You can retrieve rooftop geocoordinates for verified United Kingdom addresses.

To retrieve rooftop geocoordinates, install the Rooftop geocoding database (GCRT) for the country in question. If rooftop geocoordinates are not available, Address Verification returns the status code EGC0.

**Note:** When you specify rooftop geocoding, Address Verification returns an EGC0 status code for any address outside the United Kingdom. If the input data contains addresses from countries other than the United Kingdom, create a separate address object for the United Kingdom addresses and specify rooftop geocoding for the address object.

## **Standard GeoCoding**

Standard geocoordinates are interpolated geocoordinates. Standard geocoding interpolates a point close to an address based on the geocoordinates of nearby addresses. For example, if the geocoordinates for the first and last houses on a street are available, Address Verification uses the known geocoordinates to interpolate the geocoordinates for a house that is situated in between. To retrieve standard geocoordinates, install the GC database for the country in question. The geocoding status codes EGC4 through EGC8 indicate the level of accuracy for the interpolated geocoordinates.

If you enable geocoding and you do not specify a geocoding type, Address Verification returns one of the following geocoding types:

- Arrival point geocoordinates for countries that support arrival point geocoding.
- Standard geocoordinates for countries that do not support arrival point geocoding.

## Supported Countries and Databases

The following table lists the countries for which you can retrieve geocodes, the geocoding databases that you can install, and the type of geocodes that you receive:

| Country            | Database Filename | Geocoding Type          |
|--------------------|-------------------|-------------------------|
| Andorra            | AND5GC.MD         | Standard (Interpolated) |
| Argentina          | ARG5GC.MD         | Standard (Interpolated) |
| Australia          | AUS5GCAP.MD       | Arrival Point           |
| Australia          | AUS5GC.MD         | Standard (Interpolated) |
| Australia          | AUS5GCPC.MD       | Parcel Centroid         |
| Austria            | AUT5GCAP.MD       | Arrival Point           |
| Austria            | AUT5GC.MD         | Standard (Interpolated) |
| Austria            | AUT5GCPC.MD       | Parcel Centroid         |
| Belgium            | BEL5GC.MD         | Standard (Interpolated) |
| Brazil             | BRA5GC.MD         | Standard (Interpolated) |
| Bulgaria           | BGR5GC.MD         | Standard (Interpolated) |
| Canada             | CAN5GCAP.MD       | Arrival Point           |
| Canada             | CAN5GC.MD         | Standard (Interpolated) |
| Canada             | CAN5GCPC.MD       | Parcel Centroid         |
| Cayman Islands     | CYM5GC.MD         | Standard (Interpolated) |
| Columbia           | COL5GC.MD         | Standard (Interpolated) |
| Costa Rica         | CRI5GC.MD         | Standard (Interpolated) |
| Croatia            | HRV5GCAP.MD       | Arrival Point           |
| Croatia            | HRV5GC.MD         | Standard (Interpolated) |
| Croatia            | HRV5GCPC.MD       | Parcel Centroid         |
| Czech Republic     | CZE5GC.MD         | Standard (Interpolated) |
| Denmark            | DNK5GCAP.MD       | Arrival Point           |
| Denmark            | DNK5GC.MD         | Standard (Interpolated) |
| Denmark            | DNK5GCPC.MD       | Parcel Centroid         |
| Dominican Republic | DOM5GC.MD         | Standard (Interpolated) |

| Country     | Database Filename | Geocoding Type          |
|-------------|-------------------|-------------------------|
| El Salvador | SLV5GC.MD         | Standard (Interpolated) |
| Estonia     | EST5GCAP.MD       | Arrival Point           |
| Estonia     | EST5GC.MD         | Standard (Interpolated) |
| Estonia     | EST5GCPC.MD       | Parcel Centroid         |
| Finland     | FIN5GCAP.MD       | Arrival Point           |
| Finland     | FIN5GC.MD         | Standard (Interpolated) |
| Finland     | FIN5GCPC.MD       | Parcel Centroid         |
| France      | FRA5GCAP.MD       | Arrival Point           |
| France      | FRA5GC.MD         | Standard (Interpolated) |
| France      | FRA5GCPC.MD       | Parcel Centroid         |
| Germany     | DEU5GCAP.MD       | Arrival Point           |
| Germany     | DEU5GC.MD         | Standard (Interpolated) |
| Germany     | DEU5GCPC.MD       | Parcel Centroid         |
| Gibraltar   | GIB5GC.MD         | Standard (Interpolated) |
| Greece      | GRC5GC.MD         | Standard (Interpolated) |
| Holy See    | VAT5GC.MD         | Standard (Interpolated) |
| Honduras    | HND5GC.MD         | Standard (Interpolated) |
| Hong Kong   | HKG5GC.MD         | Standard (Interpolated) |
| Hungary     | HUN5GCAP.MD       | Arrival Point           |
| Hungary     | HUN5GC.MD         | Standard (Interpolated) |
| Hungary     | HUN5GCPC.MD       | Parcel Centroid         |
| Iceland     | ISL5GC.MD         | Standard (Interpolated) |
| India       | IND5GC.MD         | Standard (Interpolated) |
| Ireland     | IRL5GC.MD         | Standard (Interpolated) |
| Italy       | ITA5GCAP.MD       | Arrival Point           |
| Italy       | ITA5GC.MD         | Standard (Interpolated) |
| Italy       | ITA5GCPC.MD       | Parcel Centroid         |

| Country       | Database Filename | Geocoding Type          |
|---------------|-------------------|-------------------------|
| Jamaica       | JAM5GC.MD         | Standard (Interpolated) |
| Latvia        | LVA5GCAP.MD       | Arrival Point           |
| Latvia        | LVA5GC.MD         | Standard (Interpolated) |
| Latvia        | LVA5GCPC.MD       | Parcel Centroid         |
| Liechtenstein | LIE5GCAP.MD       | Arrival Point           |
| Liechtenstein | LIE5GC.MD         | Standard (Interpolated) |
| Liechtenstein | LIE5GCPC.MD       | Parcel Centroid         |
| Lithuania     | LTU5GCAP.MD       | Arrival Point           |
| Lithuania     | LTU5GC.MD         | Standard (Interpolated) |
| Lithuania     | LTU5GCPC.MD       | Parcel Centroid         |
| Luxembourg    | LUX5GCAP.MD       | Arrival Point           |
| Luxembourg    | LUX5GC.MD         | Standard (Interpolated) |
| Luxembourg    | LUX5GCPC.MD       | Parcel Centroid         |
| Malta         | MLT5GC.MD         | Standard (Interpolated) |
| Malaysia      | MYS5GC.MD         | Standard (Interpolated) |
| Mexico        | MEX5GCAP.MD       | Arrival Point           |
| Mexico        | MEX5GC.MD         | Standard (Interpolated) |
| Monaco        | MCO5GCAP.MD       | Arrival Point           |
| Monaco        | MCO5GC.MD         | Standard (Interpolated) |
| Monaco        | MCO5GCPC.MD       | Parcel Centroid         |
| Netherlands   | NLD5GCAP.MD       | Arrival Point           |
| Netherlands   | NLD5GC.MD         | Standard (Interpolated) |
| Netherlands   | NLD5GCPC.MD       | Parcel Centroid         |
| New Zealand   | NZL5GC.MD         | Standard (Interpolated) |
| Norway        | NOR5GCAP.MD       | Arrival point           |
| Norway        | NOR5GC.MD         | Standard (Interpolated) |
| Norway        | NOR5GCPC.MD       | Parcel Centroid         |



| Country                                      | Database Filename | Geocoding Type          |
|--|-------------------|-------------------------|
| Panama                                       | PAN5GC.MD         | Standard (Interpolated) |
| Peru   | PER5GC.MD         | Standard (Interpolated) |
| Poland                                       | POL5GCAP.MD       | Arrival Point           |
| Poland                                       | POL5GC.MD         | Standard (Interpolated) |
| Poland                                       | POL5GCPC.MD       | Parcel Centroid         |
| Portugal                                     | PRT5GC.MD         | Standard (Interpolated) |
| San Marino                                   | SMR5GC.MD         | Standard (Interpolated) |
| Singapore                                    | SGP5GC.MD         | Standard (Interpolated) |
| Slovakia                                     | SVK5GCAP.MD       | Arrival Point           |
| Slovakia                                     | SVK5GC.MD         | Standard (Interpolated) |
| Slovenia                                     | SVN5GCAP.MD       | Arrival Point           |
| Slovenia                                     | SVN5GC.MD         | Standard (Interpolated) |
| Slovenia                                     | SVN5GCPC.MD       | Parcel Centroid         |
| South Africa                                 | ZAF5GC.MD         | Standard (Interpolated) |
| South Georgia and the South Sandwich Islands | SGS5GC.MD         | Standard (Interpolated) |
| Spain  | ESP5GC.MD         | Standard (Interpolated) |
| Sweden                                       | SWE5GCAP.MD       | Arrival Point           |
| Sweden                                       | SWE5GC.MD         | Standard (Interpolated) |
| Sweden                                       | SWE5GCPC.MD       | Parcel Centroid         |
| Switzerland                                  | CHE5GCAP.MD       | Arrival Point           |
| Switzerland                                  | CHE5GC.MD         | Standard (Interpolated) |
| Switzerland                                  | CHE5GCPC.MD       | Parcel Centroid         |
| Turkey                                       | TUR5GC.MD         | Standard (Interpolated) |
| United Kingdom                               | GBR5GCRT.MD       | Rooftop                 |
| United Kingdom                               | GBR5GC.MD         | Standard (Interpolated) |
| United States                                | USA5GCAP.MD       | Arrival Point           |
| United States                                | USA5GC.MD         | Standard (Interpolated) |

| Country       | Database Filename | Geocoding Type          |
|---------------|-------------------|-------------------------|
| United States | USA5GPCPC.MD      | Parcel Centroid         |
| Venezuela     | VEN5GC.MD         | Standard (Interpolated) |

## Configuring GeoCoding

You can configure Informatica Address Verification to return geocoding information in the result output. For certain countries, you can also specify the type of geocoding to include in the output. Before you enable geocoding, you must install the geocoding reference address databases and specify the corresponding unlock codes in the `SetConfig.xml` file.

1. In `Parameters.xml`, set the `EnrichmentGeocoding` attribute of the `Process` element to `ON`.  
The default value for `EnrichmentGeocoding` is `OFF`.
2. To specify the type of geocoding, set the `EnrichmentGeocodingType` attribute of the `Process` element to one of the following values:
  - `ARRIVAL_POINT`. To enable arrival point geocoding. This is the default value for the `EnrichmentGeocodingType` attribute.
  - `PARCEL_CENTROID`. To enable the parcel centroid geocoding.
  - `ROOFTOP`. To enable the rooftop geocoding.
  - `NONE`. To enable the standard geocoding.

## GeoCoding Status Values

Informatica Address Verification returns the following GeoCoding status values that help you understand the status of a GeoCoding request.

### **EGCN**

Address Verification cannot find the geocoding database.

### **EGCU**

The geocoding database is not unlocked.

### **EGCC**

The geocoding database is corrupt.

### **EGC0**

Address Verification could not append geocoordinates to the input address because no geocoordinates are available for the address.

### **EGC4**

Geocoordinates are only partially accurate to the postal code level. For example, 795xx.

### **EGC5**

Geocoordinates are accurate to the postal code level.

### **EGC6**

Geocoordinates are accurate to the locality level.

**EGC7**

Geocoordinates are accurate to the street level.

**EGC8**

Geocoordinates are accurate to the house number level. (Estimated location of the parcel of land with street-side offset.)

**EGC9**

High-precision arrival point geocoordinates. (Measured entryway to the parcel of land.)

**EGCA**

High-precision parcel centroid geocoordinates. (Measured center of the parcel of land.)

**EGCB**

High-precision rooftop geocoordinates.

## Country-Specific Address Enrichments

You can retrieve country-specific address enrichments as enrichments to validated addresses. The country-specific enrichments provide additional information about addresses and improve the address quality and usability.

You can configure Informatica Address Verification to include address enrichments for addresses from the following countries.

- Austria
- Belgium
- Brazil
- France
- Germany
- Japan
- South Korea
- Poland
- Russia
- Serbia
- South Africa
- Switzerland
- United Kingdom
- United States

To include country-specific enrichments in the validated address output, you must install the batch and interactive reference address database and the supplementary address databases for the country and include the corresponding unlock codes in `SetConfig.xml`.

## Reference Address Databases for Country-Specific Enrichments

Informatica Address Verification supports the following reference address databases for country-specific enrichments:

### **Austria**

AUT5E1.MD

### **Belgium**

BEL5E1.MD

### **Brazil**

BRA5E1.MD

### **France**

FRA5E1.MD

### **Germany**

DEU5E1.MD

### **Japan**

JPN5E1.MD

### **South Korea**

KOR5E1.MD

### **Poland**

POL5E1.MD

### **Russia**

RUS5E1.MD

### **Serbia**

SRB5E1.MD

### **South Africa**

ZAF5E1.MD

### **Switzerland**

CHE5E1.MD

### **United Kingdom**

GBR5E1.MD

### **United States**

USA5E1.MD

## Enrichments for Austria Addresses

You can configure Informatica Address Verification to include the Postal Address Code (PAC) in the validated output for Austria addresses. To enable address enrichments for Austria, you must install the latest

supplementary address database for Austria and include the corresponding unlock code in the `SetConfig.xml` file.

For example, when you validate the following Austria address with address enrichment enabled, Address Verification returns the PAC value of 105176447 in the validated output.

```
<InputData>
  <AddressElements>
    <Country Item="1" Type="NAME">AUT</Country>
    <Locality Item="1" Type="COMPLETE">Perchtoldsdorf</Locality>
    <PostalCode Item="1" Type="UNFORMATTED">2380</PostalCode>
    <Province Item="1" Type="COUNTRY_STANDARD">Niederösterreich</Province>
    <Street Item="1" Type="COMPLETE">Plättenstraße</Street>
    <Number Item="1" Type="COMPLETE">7</Number>
  </AddressElements>
</InputData>
```

## Enabling Address Enrichments for Austria Addresses

You can configure Informatica Address Verification to include address enrichments such as Postal Address Code in validated Austria addresses. Before you enable address enrichments for Austria, install the relevant databases and include the corresponding unlock codes in the `SetConfig.xml` file.

- In `Parameters.xml`, set the `EnrichmentSupplementaryAT` attribute of the `Process` element to `ON`.  
The default value of this attribute is `OFF`.

## Status Codes for Austria Address Enrichments

When you validate Austria addresses with the country-specific enrichments enabled, Informatica Address Verification returns the following status codes along with the validated output.

### **EAT0**

Country-specific enrichment is not available for the input address.

### **EAT1**

The output address contains country-specific enrichments.

### **EATC**

The country-specific database is corrupt.

### **EATN**

Address Verification could not find the country-specific database.

### **EATU**

The country-specific database has not been unlocked.

## Enrichment for Belgium Addresses

You can configure Informatica Address Verification to add the National Institute of Statistics (StatBel) or NIS codes as an enrichment to validated Belgium addresses.

Before you enable the enrichment for Belgium addresses, you must license and install the relevant databases and include the corresponding unlock codes in `SetConfig.xml`.

NIS codes are five-digit codes that uniquely identify geographic areas in Belgium. If you enable enrichment for Belgium, Address Verification returns a nine-digit code that contains the five-digit NIS code and a four-digit Neighborhood ID.

For example, Address Verification returns `21004A001` as an enrichment to the following address:

```
Rue au Beurre 1  
1000 Bruxelles  
BEL
```

In this example, `21004` is the NIS code and `A001` is the Neighborhood ID.

## Configuring Address Enrichments for Belgium

You can configure Informatica Address Verification to include NIS codes in the validated address for Belgium.

- In `Parameters.xml`, set the `EnrichmentSupplementaryBE` attribute of the `Process` element to `ON`.  
The default value of this attribute is `OFF`.

## Status Codes for Belgium Address Enrichments

When you validate Belgium addresses with the country-specific enrichments enabled, Informatica Address Verification returns the following status codes along with the validated address.

### **EBE0**

Country-specific enrichment is not available for the input address.

### **EBE1**

The validated address contains country-specific enrichments.

### **EBEC**

The country-specific database is corrupt.

### **EBEN**

Address Verification could not find the country-specific database.

### **EBEU**

The country-specific database has not been unlocked.

## Enrichments for Brazil Addresses

You can configure Informatica Address Verification to include the Brazilian Institute of Geography and Statistics (IBGE) code as enrichment to validated Brazil addresses. To enable the address enrichment, you must license and install the supplementary address database for Brazil, and include the corresponding unlock code in the `SetConfig.xml` file.

The IBGE code is a seven-digit numeric code that identifies cities and states in Brazil. The IBGE code is useful for e-commerce operations as you can use this code for taxation and audit purposes.

### Example: IBGE Code in Address Output

When you validate the following address with the address enrichment enabled, Address Verification returns an IBGE code of `IBGE_CODE: 2606101` as an enrichment to the validated output.

```
Rua da Matriz 9  
Centro  
Glória do Goitá-pe  
55620-000  
Brazil
```

## Configuring Enrichments for Brazil Addresses

You can configure Informatica Address Verification to include the IBGE code in validated output for Brazil addresses. Before you enable address enrichments for Brazil, license and install the relevant reference address databases and include the corresponding unlock codes in the `SetConfig.xml` file.

- In `Parameters.xml`, set the `EnrichmentSupplementaryBR` attribute of the `Process` element to `ON`.  
By default, this attribute is set to `OFF`.

## Status Codes for Brazil Address Enrichments

When you validate Brazil addresses with the country-specific enrichments enabled, Informatica Address Verification returns the following status codes along with the validated output.

### EBR0

Country-specific enrichment is not available for the input address.

### EBR1

The validated address contains country-specific enrichment.

### EBRC

The country-specific database is corrupt.

### EBRN

Address Verification could not find the country-specific database.

### EBRU

The country-specific database has not been unlocked.

## Enrichments for France Addresses

You can configure Informatica Address Verification to include the INSEE code and the INSEE-9 code in the validated output for France addresses. To enable address enrichments for France, you must install the latest supplementary database for France and include the corresponding unlock code in the `SetConfig.xml`.

The INSEE code is a numeric indexing code that the French National Institute for Statistics and Economic Studies (INSEE) use to identify entities such as French communes and departments. INSEE codes are particularly helpful in uniquely identifying French communes that share the same name, spelling, and pronunciation. Of a five-digit INSEE code for a commune, the first two digits represent the department and the last three denote the commune. INSEE codes are also used as National Identification Numbers for French citizens.

The INSEE-9 code is also known as the IRIS code. IRIS stands for aggregated units for statistical information in French, and represents a demographic group that contains a maximum of 2000 people. France is composed of around 16,100 IRIS units including 650 units in overseas departments.

For example, you receive INSEE code 47001 and INSEE 9 code 470010115 when you validate the following France address:

```
<InputData>
  <AddressElements>
    <Country Item="1" Type="NAME">FRA</Country>
    <Locality Item="1" Type="COMPLETE">AGEN</Locality>
    <PostalCode Item="1" Type="UNFORMATTED">47000</PostalCode>
    <Street Item="1" Type="COMPLETE">RUE DU PUIITS DU SAUMON</Street>
    <Number Item="1" Type="COMPLETE">6</Number>
  </AddressElements>
</InputData>
```

## Enabling Address Enrichments for France Addresses

You can configure Informatica Address Verification to include enrichments such as INSEE code and INSEE 9 code in the validated output for France addresses. Before you enable address enrichments for France, install the relevant reference address databases and include the corresponding unlock codes in the `SetConfig.xml` file.

- In `Parameters.xml`, set the `EnrichmentSupplementaryFR` attribute of the `Process` element to `ON`.

The default value for this attribute is `OFF`.

## Status Codes for France Address Enrichments

When you validate France addresses with the country-specific enrichments enabled, Informatica Address Verification returns the following status codes along with the validated output.

### **EFR0**

Country-specific enrichment is not available for the input address.

### **EFR1**

The validated address contains country-specific enrichments. Note that this does not indicate that all the enrichments supported for that country are provided in the output.

### **EFRC**

The country-specific database is corrupt.

### **EFRN**

Address Verification could not find the country-specific database.

### **EFRU**

The country-specific database has not been unlocked.

## Enrichments for Germany Addresses

You can configure Informatica Address Verification to include multiple address enrichments in validated Germany addresses. To enable address enrichments for Germany, you must license and install the latest



supplementary address database for Germany and included the corresponding unlock codes in `SetConfig.xml`.

You can configure Address Verification to include the following enrichments in validated Germany addresses.

#### **AGS**

The Amtliche Gemeindeschlüssel (AGS) is a variable-length code that uniquely identifies a locality in Germany.

#### **LOCALITY\_ID**

The Locality ID is a variable length code that uniquely identifies a locality in Germany.

#### **STREET\_ID**

The Street ID is a variable length code that uniquely identifies a street address in Germany.

#### **STREET\_CODE**

The Street Code is a three-digit code that identifies a street in Germany. Positions 6,7, and 8 of the Frachtleitcode or Freight code form the street code. A street code value of 994 indicates that the address points to a packstation.

For example, when you validate

```
Röntgenstr. 9  
67133 Maxdorf  
Germany
```

Address Verification returns the following additional information in the validated output:

```
DEU_AGS: 07338018  
DEU_LOCALITY_ID: 68015519  
DEU_STREET_ID: 100560690  
DEU_STREET_CODE: 057
```

## Enabling Address Enrichments for Germany Addresses

You can configure Address Verification to include address enrichments such as `DEU_AGS`, `DEU_LOCALITY_ID`, and `DEU_STREET_ID` in validated output of Germany addresses. Before you enable address enrichment for Germany addresses, install the relevant reference address databases and include the corresponding unlock codes in the `SetConfig.xml` file.

- To enable address enrichments for Germany addresses, set the `EnrichmentSupplementaryDE` attribute of the `Process` element in `Parameters.xml` to `ON`.

The default value of this attribute is `OFF`.

## Status Codes for Germany Address Enrichments

When you validate Germany addresses with the country-specific enrichments enabled, Informatica Address Verification returns the following status codes along with the validated output.

#### **EDE0**

Country-specific enrichment is not available for the input address.

#### **EDE1**

The result contains country-specific enrichments. Note that this status code does not indicate that the result contains all the enrichments supported for the country.

#### **EDEC**

The country-specific database is corrupt.

#### EDEN

Address Verification could not find the country-specific database.

#### EDEU

The country-specific database has not been unlocked.

## Enrichments for Japan Addresses

You can configure Informatica Address Verification to include enrichments in the validated Japan addresses. To enable address enrichments for Japan, you must license and install the relevant address databases for Japan and include the corresponding unlock code in `SetConfig.xml`.

#### CHOUMEI\_AZA\_CODE

An 11-digit code that uniquely identifies a delivery point in Japan.

If you specify the CHOUMEI\_AZA\_CODE enrichment, Address Verification returns the old Choumei Aza code in the validated output. To receive the Choumei Aza code for an earlier version of the address, you must also enable the MatchingExtendedArchive attribute of the Process element in `Parameters.xml` along with the address enrichment settings.

#### NEW\_CHOUMEI\_AZA\_CODE

The Choumei Aza code that Japan Post issues when it updates an address. The new Choumei Aza code is also an 11-digit code that uniquely identifies a delivery point in Japan.

Address Verification returns the new Choumei Aza code that Japan Post specifies when it updates an address. However, Japan Post might update an address more than once. To ensure that you have the current address data and the current Choumei Aza code for an address, verify the new Choumei Aza code in the address code lookup mode. For more information, see [“Using the Extended Archives to Update a Japan Address” on page 50](#).

#### GAIKU\_CODE

A four-digit code that uniquely identifies a city block in Japan. You can use a combination of the new Choumei Aza code and the Gaiku code in the AddressCodeLookup function to identify a unique address in Japan.

## Enabling Address Enrichments for Japan Addresses

You can configure Informatica Address Verification to include address enrichments such as the Choumei Aza code and the Gaiku code in Japan addresses. Before you enable address enrichments for Japan, install the relevant reference address databases and include the corresponding unlock codes in the `SetConfig.xml` file.

- In `Parameters.xml`, set the EnrichmentSupplementaryJP attribute of the Process element to ON.

The default value of this attribute is OFF.

## Status Codes for Japan Address Enrichments

When you validate Japan addresses with the country-specific enrichments enabled, Informatica Address Verification returns the following status codes along with the validated output.

#### EJPO

Country-specific enrichment is not available for the input address.

**EJP1**

The result contains country-specific enrichments. Note that this status code does not indicate that the result contains all available enrichments for the country.

**EJPC**

The country-specific database is corrupt.

**EJPN**

Address Verification could not find the country-specific database.

**EJPU**

The country-specific database has not been unlocked.

## Enrichments for South Korea Addresses

You can configure Address Verification to return different types of information for a South Korea address. When you verify a South Korea address, Address Verification returns an address ID value that you can use in address code lookup mode to obtain the address information that you need.

The address ID value is composed of eight digits followed by a single letter. The eight digits uniquely identify the address. The final character indicates the combination of new and old information that the address contains.

Enter the address ID value in address code lookup mode to update the address information. When you verify the address in address code lookup mode, you can return the address in the older, land lot-based format or the current, street-based format. You can also return the six-digit post code or the current five-digit post code for the address.

When you enter the address ID value in address code lookup mode, you must update the final character to specify the types of information that you need. For more information, see [“Extended Archives and South Korea Addresses” on page 50](#).

## Enabling Address Enrichment for South Korea

You can configure Address Verification to return different types of information for a South Korea address. A South Korea address can store information in the current, street-based format or the older, land lot-based format. A South Korea address can also contain post code information in the current, six-digit post code or the older, five-digit format. You can use the address ID value to retrieve each type of information from the reference data.

To enable the address ID enrichment for South Korea, install the supplementary reference address database for South Korea and set the `EnrichmentSupplementaryKR` attribute of the `Process` element in `Parameters.xml` to `ON`.

To verify addresses that use the older, land lot-based address format or the older, six-digit post code, set the `MatchingExtendedArchive` attribute of the `Process` element in `Parameters.xml` to `ON`.

For more information on verifying the different types of South Korea address information, see [“Extended Archives and South Korea Addresses” on page 50](#).

## Status Codes for South Korea Address Enrichments

When you validate South Korea addresses with the country-specific enrichments enabled, Informatica Address Verification returns the following status codes along with the validated output.

### **EKR0**

Country-specific enrichment is not available for the input address.

### **EKR1**

The result contains country-specific enrichments. Note that this status code does not indicate that the result contains all the enrichments that are available for the country.

### **EKRC**

The country-specific database is corrupt.

### **EKRN**

Address Verification could not find the country-specific database.

### **EKRU**

The country-specific database has not been unlocked.

## Enrichments for Poland Addresses

You can configure Informatica Address Verification to include the Gmina code, Locality TerytIDs, and Street TerytID as enrichments for validated Poland addresses. To enable enrichments for Poland addresses, you must install the supplementary database for Poland and include the corresponding unlock code in the `SetConfig.xml` file.

Official Register of the Territorial Division of the Country (TERYT) is the agency responsible for identifiers and names of territories, localities, roads, and buildings in Poland. Gmina is the Polish equivalent of communes or municipalities. TERYT assigns and manages Gmina code and TerytIDs.

For example, when you validate

```
<InputData>
  <AddressElements>
    <Country Item="1" Type="NAME">POL</Country>
    <Locality Item="1" Type="COMPLETE">Wrocław</Locality>
    <PostalCode Item="1" Type="UNFORMATTED">50510</PostalCode>
    <Province Item="1" Type="COUNTRY_STANDARD">dolnośląskie</Province>
    <Street Item="1" Type="COMPLETE">ul. Laskowa</Street>
    <Number Item="1" Type="COMPLETE">1</Number>
  </AddressElements>
</InputData>
```

Address Verification returns the following enrichment values:

```
GINA_CODE: 2183
LOCALITY_TERYT_ID: 0986544
STREET_TERYT_ID: 10666
```

## Enabling Address Enrichments for Poland Addresses

You can configure Informatica Address Verification to include address enrichments such as `GINA_CODE`, `LOCALITY_TERYT_ID`, and `STREET_TERYT_ID` in validated address output for Poland addresses. Before you

enable address enrichments for Poland addresses, install the relevant databases and include the corresponding unlock codes in the `SetConfig.xml` file.

- In `Parameters.xml`, set the `EnrichmentSupplementaryPL` attribute of the `Process` element to `ON`.  
By default, this attribute is set to `OFF`.

## Status Codes for Poland Address Enrichments

When you validate Poland addresses with the country-specific enrichments enabled, Informatica Address Verification returns the following status codes along with the validated output.

### **EPL0**

Country-specific enrichment is not available for the input address.

### **EPL1**

The result contains country-specific enrichments. Note that this status code does not indicate that the result contains all the enrichments that are available for the country.

### **EPLC**

The country-specific database is corrupt.

### **EPLN**

Address Verification could not find the country-specific database.

### **EPLU**

The country-specific database has not been unlocked.

## Enrichments for Russia Addresses

You can configure Informatica Address Verification to include the Federal Information Addressing System (FIAS) ID in the validated output for Russia addresses.

The FIAS ID is an alphanumeric string.

## Configuring Enrichments for Russia Addresses

You can configure Informatica Address Verification to include the FIAS ID in the validated output for Russia addresses. To enable address enrichments for Russia, you must license and install the relevant reference address databases and include the corresponding unlock codes in `SetConfig.xml`.

- In `Parameters.xml`, set the `EnrichmentSupplementaryRU` attribute of the `Process` parameter to `ON`.  
The default value of this attribute is `OFF`.

## Status Codes for Russia Address Enrichments

When you validate Russia addresses with the country-specific enrichments enabled, Informatica Address Verification returns the following status codes along with the validated output.

### **ERU0**

Country-specific enrichment is not available for the input address.

**ERU1**

The result contains country-specific enrichment.

**ERUC**

The country-specific database is corrupt.

**ERUN**

Informatica Address Verification could not find the country-specific database.

**ERUU**

The country-specific database has not been unlocked.

## Enrichments for Serbia Addresses

You can configure Informatica Address Verification to include the Postal Address Code (PAK) as an enrichment to the validated output for Serbia addresses.

The PAK is a six-digit code that maps to the street level. Including the PAK in an address ensures correct and prompt delivery to recipients in Serbia. You do not need the PAK for items that you address to a P.O. Box, *poste restante*, or to a military address.

## Enabling Address Enrichments for Serbia Addresses

You can configure Informatica Address Verification to include Postal Address Code (PAK) in validated output for Serbia addresses. Before you enable address enrichments for Serbia addresses, license and install the relevant databases and include the corresponding unlock codes in `SetConfig.xml`.

- In `Parameters.xml`, set the `EnrichmentSupplementaryRS` attribute of the `Process` element to `ON`.

The default value of this attribute is `OFF`.

## Status Codes for Serbia Address Enrichments

When you validate Serbia addresses with the country-specific enrichments enabled, Informatica Address Verification returns the following status codes along with the validated output.

**ERS0**

Country-specific enrichment is not available for the input address.

**ERS1**

The result contains the country-specific enrichment.

**ERSC**

The country-specific database is corrupt.

**ERSN**

Informatica Address Verification could not find the country-specific database.

**ERSU**

The country-specific database has not been unlocked.

# Enrichments for South Africa Addresses

You can configure Informatica Address Verification to include the National Address Database (NAD) ID in the validated output for South Africa addresses. The NAD ID is a unique numeric ID that is assigned to street addresses in South Africa. Before you enable enrichment, you must license and install the relevant reference address database and include the corresponding unlock codes in `SetConfig.xml`.

For example, Address Verification returns the NAD ID value of 2170232 in the output when you validate the following address:

```
4 Balmoral Road  
Vincent  
East London  
5247  
South Africa
```

## Enabling Address Enrichments for South Africa Addresses

You can configure Informatica Address Verification to include address enrichments such as the NAD ID in validated output of South Africa addresses. Before you enable address enrichments for South Africa addresses, license and install the relevant databases and include the corresponding unlock codes in `SetConfig.xml`.

- In `Parameters.xml`, set the `EnrichmentSupplementaryZA` attribute of the `Process` element to `ON`.  
The default value of this attribute is `OFF`.

## Status Codes for South Africa Address Enrichments

When you validate South Africa addresses with the country-specific enrichments enabled, Informatica Address Verification returns the following status codes along with the validated output.

### **EZA0**

Country-specific enrichment is not available for the input address.

### **EZA1**

The result contains the country-specific enrichment.

### **EZAC**

The country-specific database is corrupt.

### **EZAN**

Address Verification could not find the country-specific database.

### **EZAU**

The country-specific database has not been unlocked.

# Enrichments for Switzerland Addresses

You can configure Informatica Address Verification to include the additional postal code characters in the validated output for Switzerland addresses. To enable address enrichments for Switzerland, you must license

and install the relevant reference address databases for Switzerland and include the corresponding unlock codes in `SetConfig.xml`.

Address Verification returns the additional postal code characters in an enrichment field called `POCO_EXT`.

For example, when you validate the following Switzerland address with the address enrichment enabled, Address Verification returns a `POCO_EXT` value of 05.

```
Hohlen 1  
3800 Sundlaenen  
Switzerland
```

## Enabling Address Enrichment for Switzerland Addresses

You can configure Informatica Address Verification to include address the `POCO_EXT` code as enrichment to validated Switzerland addresses. Before you enable address enrichments for Switzerland, license and install the relevant databases and include the corresponding unlock codes in `SetConfig.xml`.

- In `Parameters.xml`, set the `EnrichmentSupplementaryCH` attribute of the `Process` element to `ON`.  
The default value of this attribute is `OFF`.

## Status Codes for Switzerland Address Enrichments

When you validate Switzerland addresses with the country-specific enrichments enabled, Informatica Address Verification returns the following status codes along with the validated output.

### **ECH0**

Country-specific enrichment is not available for the input address.

### **ECH1**

The result contains the country-specific enrichment.

### **ECHC**

The country-specific database is corrupt.

### **ECHN**

Address Verification could not find the country-specific database.

### **ECHU**

The country-specific database has not been unlocked.

## Enrichments for the United Kingdom Addresses

You can configure Informatica Address Verification to include address enrichments in the validated United Kingdom addresses. To enable address enrichments for the United Kingdom addresses, you must license and install the relevant reference address databases and include the corresponding unlock codes in `SetConfig.xml`.

### **ADDRESS\_KEY**

An eight-digit numeric code that maps to an address in the Postcode Address File (PAF) from the Royal Mail. An Address Key in conjunction with Organization Key and the Post Code Type uniquely identifies an address.



**DELIVERY\_POINT\_SUFFIX**

A two-character suffix that the Royal Mail assigns to a mailbox in a United Kingdom post code area. The first character in a delivery point suffix is a number and the second character is a letter. A combination of a post code and the delivery point suffix identifies a mailbox.

**DELIVERY\_POINT\_TYPE**

A single-character code that indicates whether the address points to a residence (R), a small organization (O), or a large organization (L).

**ORGANISATION\_KEY**

A unique 8-digit numeric code that Royal Mail assigns to small organizations.

**UDPRN**

An eight-character code that uniquely identifies each postal address in the Royal Mail PAF database. The Unique Delivery Point Reference Number (UDPRN) remains uniquely tied to the physical delivery point regardless of changes in the address.

**UPRN**

A numeric code that uniquely identifies a land or property unit in the United Kingdom. The Unique Property Reference Number (UPRN) is a code that the United Kingdom government assigns and can contain a maximum of 12 digits.

**Examples**

The following examples show enrichment values for different delivery point type addresses.

| Delivery Point Type | Input Address  | Enrichment Values  |
|---------------------|--|--|
| Residence           | FLAT 17<br>GROVE HOUSE<br>WAVERLEY GROVE<br>LONDON<br>N3 3PU<br>UNITED KINGDOM | DELIVERY_POINT_SUFFIXES: 1H<br>DELIVERY_POINT_TYPE: R<br>UDPRN: 15498195<br>ADDRESS_KEY: 18161676<br>ORGANIZATION_KEY: 00000000<br>UPRN: 200123099   |
| Large Organization  | PO BOX 43078<br>LONDON<br>NW1 1SF<br>UNITED KINGDOM                            | DELIVERY_POINT_SUFFIXES: 1A<br>DELIVERY_POINT_TYPE: L<br>UDPRN: 17635833<br>ADDRESS_KEY: 02356470<br>ORGANIZATION_KEY: 00000000<br>UPRN: 10015054387 |
| Small Organization  | 17A THE GROVE<br>LONDON<br>N3 1QN<br>UNITED KINGDOM                            | DELIVERY_POINT_SUFFIXES: 1H<br>DELIVERY_POINT_TYPE: O<br>UDPRN: 15491057<br>ADDRESS_KEY: 28470295<br>ORGANIZATION_KEY: 01464593<br>UPRN: 200210632   |

## Configuring Address Enrichments for the United Kingdom Addresses

You can configure Address Verification to add enrichments to valid United Kingdom addresses. Address Verification can add address keys, delivery point suffixes, Unique Property Reference Numbers (UPRN), and Unique Delivery Point Reference Numbers (UDPRN) to the addresses. Before you enable the address enrichments, license and install the relevant reference address databases and include the corresponding unlock codes in `SetConfig.xml`.

- In `Parameters.xml`, set the `EnrichmentSupplementaryGB` attribute of the `Process` element to `ON`.  
The default value of the attribute is `OFF`.

## Status Codes for the United Kingdom Enrichments

When you validate the United Kingdom addresses with country-specific enrichments enabled, Informatica Address Verification returns the following status codes along with the validated addresses.

### **EGB0**

Country-specific enrichments not available for the input address.

### **EGB1**

The result contains country-specific enrichments. Note that this status code does not indicate that the result contains all available enrichments.

### **EGBC**

Country-specific enrichment database is corrupt.

### **EGBN**

Address Verification could not find the country-specific enrichment database.

### **EGBU**

The country-specific enrichment database has not been unlocked.

## Enrichments for the United States Addresses

You can configure Informatica Address Verification to include multiple address enrichments when you validate the United States addresses. Before you enable address enrichments for the United States addresses, license and install the relevant databases and include the corresponding unlock codes in `SetConfig.xml`.

To enable address enrichments for the United States addresses, set the `EnrichmentSupplementaryUS` attribute of the `Process` element in `Parameters.xml` to `ON`.

### **COUNTY\_FIPS\_CODE**

A three-digit number that identifies a county in the United States. The United States Federal Information Processing Standard (FIPS) maintains a set of codes that identify states, counties, and other territorial possessions. The two-digit state code identifies each state. The three-digit county code identifies a county within a state. The five digits of the state and county codes together uniquely identify a county.

**STATE\_FIPS\_CODE**

A two-digit number that identifies a state in the United States. The United States FIPS controls the numeric and alphabetic codes that identify states and other territories of the United States.

**MSA\_ID**

The Metropolitan Statistical Area identification number (MSAID) is a four-digit code that identifies a United States urban area that has a population of 50,000 or more.

**CBSA\_ID**

The Core-Based Statistical Area (CBSA) identification number is a five-digit number that identifies an urban area with a population greater than 10,000. A CBSA can be a Metropolitan Statistical Area or Micropolitan Statistical Area. A Metropolitan Statistical Area has over 50,000 inhabitants. A Micropolitan Statistical Area has between 10,000 and 50,000 inhabitants.

**FINANCE\_NUMBER**

The finance number is a six-digit code assigned to the United States post offices and other postal facilities for collection of cost and statistical data. The first two digits of the finance number identify the state. The final four digits identify the USPS post office or postal facility.

**RECORD\_TYPE**

A single-character code that describes the type of mailbox or delivery. For example, the code can indicate if the address is in a high-rise building (value H) or a post office box (value P).

**CMSA\_ID**

The Consolidated Metropolitan Statistical Area (CMSA) identification number is four-digit code. A PMSA becomes a CMSA if local opinion favors the designation.

**TIME\_ZONE\_CODE**

A one to three character-long numeric value that indicates the time difference of the address locality with GMT. For example, the TIME\_ZONE\_CODE for Eastern Standard Time is -5.

**TIME\_ZONE\_NAME**

A three-character long code that identifies the time zone to which the address belongs. For example, EST for Eastern Standard Time localities.

**CENSUS\_TRACT\_NO**

A six-digit number that identified the Census Tract, statistical subdivision of a county.

**CENSUS\_BLOCK\_NO**

A four-digit number that identifies a Census Block, which is the smallest entity for which the Census bureau collects census information.

**CENSUS\_BLOCK\_GROUP**

A 12-digit code that identifies a Census Block Group. A Census Block Group is a group of Census blocks sharing the same first digit, which is the last digit in the 12-digit Census Block Group code.

## Enabling Address Enrichments for the United States Addresses

You can configure Informatica Address Verification to include multiple address enrichments in the validated United States addresses. Before you enable address enrichments for the United States addresses, license and install the relevant databases and include the corresponding unlock codes in `SetConfig.xml`.

- In `Parameters.xml`, set the `EnrichmentSupplementaryUS` attribute of the `Process` element to ON.

The default value of this attribute is OFF.

## Status Codes for the United States Address Enrichments

When you validate the United States addresses with the address enrichments enabled, Informatica Address Verification returns the following status codes along with the address output.

### **EUS0**

Country-specific enrichment is not available for the input address.

### **EUS1**

The result contains country-specific enrichments. Note that this status code does not indicate that the result contains all available enrichments.

### **EUSC**

The country-specific enrichment database is corrupt.

### **EUSN**

Address Verification could not find the country-specific enrichment database.

### **EUSU**

The country-specific enrichment database has not been unlocked.

## CHAPTER 6

# Result Parameters

This chapter includes the following topics:

- [Result Formatting Options, 85](#)
- [Configuring the Result Formatting Options, 86](#)
- [Result Country Settings, 87](#)
- [Configuring Country Settings for the Result Output, 87](#)
- [Configuring the Maximum Number for Results in Interactive and Fast Completion Modes, 89](#)
- [The PreferredScript Attribute, 89](#)
- [The PreferredLanguage Attribute, 93](#)
- [The GlobalPreferredDescriptor Attribute, 98](#)
- [The ElementAbbreviation Attribute, 98](#)
- [Configuring the Element Descriptor Format, 99](#)
- [The AliasLocality and AliasStreet Attributes, 99](#)
- [The RangesToExpand and FlexibleRangeExpansion Attributes, 100](#)
- [The StandardizeInvalidAddresses Attribute, 101](#)
- [The DualAddressPriority Attribute, 101](#)
- [Output Standardization, 102](#)

## Result Formatting Options

You can configure the formatting-related attributes of the `Result` element in `Parameters.xml` to specify the formatting of the validated output. The formatting-related attributes of the `Result` element include `FormatType`, `FormatDelimiter`, `FormatMaxLine`, `GlobalCasing`, and `GlobalMaxLength`.

You can configure the `FormatType` attribute to specify the address elements that you want Informatica Address Verification to include in `FormattedAddressLine` and `AddressComplete` output.

You can configure the `FormatDelimiter` attribute to specify the delimiter that you want Address Verification to use in the output address.

You can configure the `FormatMaxLine` attribute to specify the maximum number of lines that you want Address Verification to include the output. You can specify a number from one through 19. The default is 19.

You can configure the `GlobalCasing` attribute to specify the casing that you want Address Verification to apply for the output. Note that the casing setting at the address element level takes precedence over the

global setting. The global setting takes effect only when the address element-level `Casing` attribute is set to `GLOBAL`.

You can configure the `GlobalMaxLength` attribute to specify the maximum number of characters that you want in the output string. The address element-level `MaxLength` attribute setting takes precedence over the global setting. The global setting takes effect only when the address element-level `MaxLength` attribute is set to `GLOBAL` or to a value of zero, 0. The default value for the `GlobalMaxLength` attribute is 1024.

## Configuring the Result Formatting Options

You can configure the formatting-related attributes of the `Result` element in `Parameters.xml` to specify the formatting to apply on the verified output. The formatting-related attributes that you can configure include `FormatType`, `FormatDelimiter`, `FormatMaxLines`, `GlobalCasing`, and `GlobalMaxLength`.

Set the following attributes of the `Result` element in `Parameters.xml`.

### **FormatType**

Specifies the elements that you want to include in results. Set one of the following values:

- `ALL`
- `ADDRESS_ONLY`
- `WITH_ORGANIZATION`
- `WITH_CONTACT`
- `WITH_ORGANIZATION_CONTACT`
- `WITH_ORGANIZATION_DEPARTMENT`

The default value is `ALL`.

### **FormatDelimiter**

Specifies the element descriptor for the address complete output. Set one of the following values:

- `CRLF`
- `LF`
- `CR`
- `SEMICOLON`
- `COMMA`
- `TAB`
- `PIPE`
- `SPACE`

The default value is `CRLF` which denotes carriage return and line feed.

**Note:** If you set `FormatDelimiter` to `COMMA` or `SPACE`, you cannot use the result output for further verification using Informatica Address Verification.

### **FormatMaxLines**

Specifies the maximum number of address lines you want Address Verification to include in a result. Set a value between 1 and 19 for the attribute.

The default value is 19.

### GlobalCasing

Specifies the casing of the address output. Set one of the following values.

- LOWER
- MIXED
- NATIVE
- NOCHANGE
- UPPER

The default value is NATIVE. The casing setting at the address-element level takes precedence over the global setting. The global setting takes effect only when the address element-level `Casing` attribute is set to GLOBAL.

### GlobalMaxLength

Specifies the maximum length of output strings. Set the `GlobalMaxLength` attribute to the maximum number of characters you want in the output string.

The default value for `GlobalMaxLength` is 1024. The address element-level `MaxLength` attribute setting takes precedence over the global setting. The global setting takes effect only when the address element-level `MaxLength` attribute is set to GLOBAL.

## Result Country Settings

You can configure the country information-related attributes of the `Result` element in `Parameters.xml` to specify the handling of country information in the Informatica Address Verification result output.

The country information-related attributes of the `Result` element include `CountryType`, `CountryOfOriginISO3`, and `FormatWithCountry`.

If you want Address Verification to include the country information in formatted address line (FAL) output, you can set the `FormatWithCountry` attribute to ON. The default value of `FormatWithCountry` is OFF.

You can configure the `CountryType` attribute to specify how you want Address Verification to include the country information in the result output. Based on the value you set for `CountryType`, Address Verification returns the ISO3 code, ISO2 code, or name of the country in one of the supported languages. The default is `NAME_EN`, which denotes the name of the country in English.

If the ISO3 code you set for `CountryOfOriginISO3` points to same the country that the address belongs to, Address Verification does not include the country information in the FAL output.

For example, if you have set `FormatWithCountry` to ON, and `CountryOfOriginISO3` to USA, the FAL output includes the country information only for addresses from countries other than the United States.

## Configuring Country Settings for the Result Output

You can configure the `FormatwithCountry`, `CountryType`, and `CountryofOrigin` attributes of the `Result` element in `Parameters.xml` to specify the handling of country-related information in the result output.

Configure the following attributes of the `Result` element in `Parameters.xml`:

**FormatWithCountry**

Specifies whether or not to include the country name in the output. Set this attribute to ON or OFF.

The default value is OFF.

**CountryType**

Specifies the format in which you want to provide the country information in the output. Set this attribute to one of the following values:

- ABBREVIATION
- ISO2
- ISO3
- ISO\_NUMBER
- NAME\_CN
- NAME\_DA
- NAME\_DE
- NAME\_EN
- NAME\_ES
- NAME\_FI
- NAME\_FR
- NAME\_GR
- NAME\_HU
- NAME\_IT
- NAME\_JP
- NAME\_KR
- NAME\_NL
- NAME\_PL
- NAME\_PT
- NAME\_RU
- NAME\_SA
- NAME\_SE

The default value is Name\_EN.

**CountryOfOriginISO3**

Specifies the country to which an address belongs. Set this attribute to the ISO3 code of the country to which the address belongs.

If the address belongs to the same country that you specified for `CountryOfOriginISO3`, Informatica Address Verification does not include the country name in the result.



# Configuring the Maximum Number for Results in Interactive and Fast Completion Modes

You can specify the maximum number of address suggestions that you receive in the interactive and fast completion modes. Use the `MaxResultCount` attribute to set the maximum number of address suggestions. You can configure a `MaxResultCount` attribute in the `Global` element in `SetConfig.xml` or in the `Result` element in `Parameters.xml`.

The default value is 20. If you set `MaxResultCount` in both `SetConfig.xml` and `Parameters.xml`, Informatica Address Verification uses the lower of the two values.

- In `SetConfig.xml`, set a value from 1 through 100 for the `MaxResultCount` attribute of the `Global` element.
- In `Parameters.xml`, set a value from 1 through 100 for the `MaxResultCount` attribute of the `Result` element.

## The PreferredScript Attribute

The `PreferredScript` attribute defines the policy that Informatica Address Verification uses to select a character script for an output address. Informatica Address Verification can return addresses from different countries in different scripts. To specify the preferred script set for an output address, set the `PreferredScript` attribute of the `Result` element in `Parameters.xml`. For most countries, Informatica Address Verification returns addresses in Latin-1 or ASCII regardless of the script that you specify.

You can set one of the following values for the `PreferredScript` attribute:

### **DATABASE**

Returns the address in the script in which the reference address database stores the address. DATABASE is the default value.

### **POSTAL\_ADMIN\_PREF**

Returns the address in the script that the postal service local to the address prefers.

### **POSTAL\_ADMIN\_ALT**

Returns the address in the script that the postal service local to the address approves as an alternative script.

### **ASCII\_SIMPLIFIED**

Returns the address in the ASCII simplified script.

### **ASCII\_EXTENDED**

Returns the address in the ASCII-extended script. The script supports the expansion of special characters, for example Õ to OE.

### **LATIN**

Returns the address in the Latin script.

### **LATIN\_1**

Returns the address in the Latin-1 script.

## LATIN\_ALT

Returns the address in the Latin alternative script that is used in transliteration.

## PRESERVE\_INPUT

Returns the address in the same script as the input address. You can set the value for addresses in Belarus, China, Greece, Japan, Kazakhstan, Macedonia, Russia, and Ukraine.

If the input address contains elements in more than one script, Address Verification overrides the PRESERVE\_INPUT setting and returns the address in the default script in the reference address database. For example, if you specify PRESERVE\_INPUT and you enter a Japan address that contains both Kanji and Latin elements, Address Verification returns all address elements in Kanji. Kanji is the default script for the Japan reference address database.

**Note:** Setting a PreferredScript value does not guarantee that the output address will contain characters exclusively in the script that you specify. If the preferred script that you specify does not support an input character, the engine copies the input character to the output. This can result in the output address containing data in more than one script. If the input contains address elements that are not in the corresponding database, Address Verification attempts to transliterate the input according to the PreferredScript setting.

## Countries and PreferredScript Options

The following table lists the preferred scripts that Address Verification can specify for different countries:

| Country    | DATABASE            | POSTAL_ADMIN_PREF   | POSTAL_ADMIN_ALT    | LATIN   | LATIN_ALT  | LATIN_1  |
|------------|---------------------|---------------------|---------------------|---|--|--|
| <b>BLR</b> | Belarusian Cyrillic | Belarusian Cyrillic | Belarusian Cyrillic | ASCII, transliterated to the Belarusian database standard | ASCII, transliterated to the Belarusian BGN standard | ASCII, transliterated to the Belarusian BGN standard |
| <b>BGR</b> | Bulgarian Cyrillic  | Bulgarian Cyrillic  | Bulgarian Cyrillic  | ASCII, transliterated to the Bulgarian BGN standard       | ASCII, transliterated to the Bulgarian BGN standard  | ASCII, transliterated to the Bulgarian BGN standard  |
| <b>CHN</b> | Hanzi               | Hanzi               | Hanzi               | Latin, Mandarin transliteration                           | Latin, Cantonese transliteration                     | Latin-1  |
| <b>CZE</b> | Latin-2             | Latin-2             | Latin-2             | ASCII   | Latin-2  | Latin-1  |
| <b>GRC</b> | Greek               | Greek               | Greek               | Latin-1, transliterated to the ISO standard               | ASCII, transliterated to the BGN standard            | ASCII  |
| <b>HKG</b> | ASCII               | ASCII               | ASCII               | ASCII   | ASCII  | ASCII  |
| <b>HUN</b> | Latin-2             | Latin-2             | Latin-2             | ASCII   | Latin-2  | Latin-1  |
| <b>ISR</b> | ASCII               | ASCII               | ASCII               | ASCII   | ASCII  | ASCII  |

| Country    | DATABASE | POSTAL_ADMIN_PREF | POSTAL_ADMIN_ALT | LATIN                                       | LATIN_ALT  | LATIN_1  |
|------------|----------|-------------------|------------------|---|--|--|
| <b>JPN</b> | Kanji    | Kanji             | Kana             | Latin-7                                     | Latin-7  | ASCII  |
| <b>KAZ</b> | Cyrillic | Cyrillic          | Cyrillic         | Latin-2, transliterated to the ISO standard | ASCII, transliterated to the BGN standard            | ASCII, transliterated to the BGN standard            |
| <b>KOR</b> | Hangul   | Hangul            | Hangul           | ASCII                                       | ASCII  | ASCII  |
| <b>LVA</b> | Latin-7  | Latin-7           | Latin-7          | ASCII                                       | Latin-7  | ASCII  |
| <b>MDA</b> | Latin-2  | Latin-2           | Latin-2          | ASCII                                       | Latin-2  | Latin-1  |
| <b>MKD</b> | Cyrillic | Cyrillic          | Cyrillic         | ASCII                                       | ASCII, transliterated to the Macedonian BGN standard | ASCII, transliterated to the Macedonian BGN standard |
| <b>POL</b> | Latin-2  | Latin-2           | Latin-2          | ASCII                                       | Latin-2  | Latin-1  |
| <b>ROU</b> | Latin-3  | Latin-3           | Latin-3          | ASCII                                       | Latin-3  | Latin-1  |
| <b>RUS</b> | Cyrillic | Cyrillic          | Cyrillic         | Latin-2, transliterated to the ISO standard | ASCII, transliterated to the BGN standard            | ASCII, transliterated to the BGN standard            |
| <b>SVK</b> | Latin-2  | Latin-2           | Latin-2          | ASCII                                       | Latin-2  | Latin-1  |
| <b>TWN</b> | Hanzi    | Hanzi             | Hanzi            | Latin, Mandarin transliteration             | Latin, Cantonese transliteration                     | Latin-1  |
| <b>UKR</b> | Cyrillic | Cyrillic          | Cyrillic         | ASCII                                       | ASCII, transliterated to the Ukrainian BGN standard  | ASCII, transliterated to the Ukrainian BGN standard  |

**Note:** Address Verification can return addresses in ASCII-simplified and ASCII-extended characters for each country.

## PreferredScript Examples

The following examples show how the `PreferredScript` attribute functions in different scenarios.

### Japan address input in Kanji script

The following Japan address is in Kanji script. The `PreferredScript` attribute is set to `PRESERVE_INPUT`.

```
<InputData>
  <AddressElements>
    <Country Item="1" Type="NAME">JAPAN</Country>
    <Locality Item="1" Type="COMPLETE">大阪市</Locality>
    <Locality Item="2" Type="COMPLETE">都島区</Locality>
```

```

        <Locality Item="3" Type="COMPLETE">内代町</Locality>
        <PostalCode Item="1" Type="UNFORMATTED">〒 534-0013</PostalCode>
        <Province Item="1" Type="COUNTRY_STANDARD">大阪府</Province>
        <Street Item="1" Type="COMPLETE">2 丁目</Street>
    </AddressElements>
</InputData>

```

The output is in Kanji script:

```

<FormattedAddressLine Line="1">JAPAN 〒 534-0013 大阪府大阪市都島区内代町 2 丁目</FormattedAddressLine>

```

## Japan address input in mixed script

The following input address contains both Latin and Kanji scripts.

```

<InputData>
  <AddressElements>
    <Country Item="1" Type="NAME">JAPAN</Country>
    <Locality Item="1" Type="COMPLETE">ŌSAKA-SHI</Locality>
    <Locality Item="2" Type="COMPLETE">MIYAKOJIMA-KU</Locality>
    <Locality Item="3" Type="COMPLETE">UCHINDAI-CHŌ</Locality>
    <PostalCode Item="1" Type="UNFORMATTED">534-0013</PostalCode>
    <Province Item="1" Type="COUNTRY_STANDARD">ŌSAKA-FU</Province>
    <Street Item="1" Type="COMPLETE">2 CHŌME</Street>
    <Organization Item="1" Type="COMPLETE">才</Organization>
  </AddressElements>
</InputData>

```

The output in this case, where the input contains both Latin and Kanji scripts, is in Kanji script , which is the default script of the Japanese address database.

```

<FormattedAddressLine Line="1">〒 534-0013 大阪府大阪市都島区内代町 2 丁目才</FormattedAddressLine>

```

## Russia address input in Latin script

The following input address contains Latin script.

```

<InputData>
  <AddressElements>
    <Country Item="1" Type="NAME">RUSSIAN FEDERATION</Country>
    <Locality Item="1" Type="COMPLETE">Majma</Locality>
    <PostalCode Item="1" Type="UNFORMATTED">649100</PostalCode>
    <Province Item="1" Type="COUNTRY_STANDARD">Altaj</Province>
    <Street Item="1" Type="COMPLETE">ul. Celinnaâ</Street>
    <Number Item="1" Type="COMPLETE">1</Number>
  </AddressElements>
</InputData>

```

The output in this case is in Latin script because of the PRESERVE\_INPUT setting.

```

<FormattedAddressLine Line="1">ul. Celinnaâ 1</FormattedAddressLine>
<FormattedAddressLine Line="2">Majma</FormattedAddressLine>
<FormattedAddressLine Line="3">Altaj</FormattedAddressLine>
<FormattedAddressLine Line="4">649100</FormattedAddressLine>

```

## Russia address in mixed input that contains Cyrillic and Latin scripts:

The following input address contains both Cyrillic and Latin scripts.

```

<InputData>
  <AddressElements>
    <Country Item="1" Type="NAME">RUSSIAN FEDERATION</Country>
    <Locality Item="1" Type="COMPLETE">Majma</Locality>
    <PostalCode Item="1" Type="UNFORMATTED">649100</PostalCode>
    <Province Item="1" Type="COUNTRY_STANDARD">Altaj</Province>
    <Street Item="1" Type="COMPLETE">ul. Celinnaâ</Street>
    <Number Item="1" Type="COMPLETE">1</Number>
    <Organization Item="1" Type="COMPLETE">й</Organization>
  </AddressElements>
</InputData>

```

If you set `PreferredScript` to `PRESERVE_INPUT`, the output contains only Cyrillic because that is the default script of the Russia reference address database.

```
<FormattedAddressLine Line="1">Й</FormattedAddressLine>
<FormattedAddressLine Line="2">ул. Целинная 1</FormattedAddressLine>
<FormattedAddressLine Line="3">Майма</FormattedAddressLine>
<FormattedAddressLine Line="4">Алтай</FormattedAddressLine>
<FormattedAddressLine Line="5">649100</FormattedAddressLine>
```

## The PreferredLanguage Attribute

Informatica Address Verification can return address elements in multiple languages for addresses in Belgium, Canada, China, and Switzerland. You can configure the `Parameters.xml` file to set the language preferences. The reference address database might contain multiple language options for a single address element or for a complete address.

A reference address database that contains multiple language options includes the following languages:

- The main spoken language of the region to which each address belongs. This is the default language of the address in the reference address database.

A reference database can recognize more than one default language in a country. For example, in the Switzerland reference address database, the default language for addresses from Zurich is German, from Geneva is French, and from Ticino is Italian.

- Any other language that Address Verification supports for an address. For example, the Belgium reference address database contains address elements in Flemish, French, and German.

To define a preferred language policy, set one of the following values for the `PreferredLanguage` attribute of the `Result` element in `Parameters.xml`:

### **DATABASE**

Returns address elements in the default language for the address in the reference address database.

`DATABASE` is the default value for the `PreferredLanguage` attribute.

### **ENGLISH**

Returns province and locality information in English if the reference address database contains the English names for the locality and province. For example, Address Verification can replace München with Munich and Bayern with Bavaria in Germany addresses.

### **ALTERNATIVE\_1, ALTERNATIVE\_2, or ALTERNATIVE\_3**

Returns address elements in one of the alternative languages in the reference address database. The alternative languages depend on the country to which the address belongs.

### **PRESERVE\_INPUT**

Returns address elements in the same language as the input address. Address Verification preserves the language if the reference address database contains the address in the input language.

If the input address contains multiple languages, Address Verification returns the address in the default database language.

## Rules and Guidelines for the PreferredLanguage Setting

Consider the following rules and guidelines when you configure a preferred language:

- If you set the `PreferredLanguage` attribute to a language that is not available in the reference address database, Address Verification returns the address in the default database language. In such cases, Address Verification returns an extended element result status (EERS) value of **E**. The value indicates that the output uses the default language because the reference database does not contain the information in the language that you specified.
- The language or the format in which Address Verification returns the country name depends on the value you configure for the `CountryType` attribute of the `Result` element.
- If the input address contains an address element that the reference database does not contain, Address Verification copies the element to the result output. For such elements, Address Verification returns the information in the same language as the address input. An element input status (EIS) value of **1** indicates that the reference address database does not contain the element. An element result status (ERS) value of **1** indicates that Address Verification copied the information from the input because the reference database does not contain the information.
- If you use Informatica Address Verification Cloud, you can set a `PreferredLanguage` value under the `ValidationParameters` in the SOAP request.
- If the database contains some but not all address elements in the language you specified, Address Verification returns the address partially in the language you specified and partially in the default language.

## Multilanguage Support for Belgium Addresses

You can configure Address Verification to return the province, locality, and street information in valid Belgium addresses in English, Flemish, French, or German.

To set the preferred language, configure the `PreferredLanguage` attribute of the `Result` element in `Parameters.xml`.

The following table describes the `PreferredLanguage` values that you can set for Belgium address data:

| Option         | Description   |
|----------------|---|
| ENGLISH        | Returns the province, locality, and street information in English if the reference address database contains the information in English.<br>Returns the other address elements in the main language of the region to which the address belongs.   |
| DATABASE       | Default value. Returns addresses in the main language of the region to which the address belongs. The address might be Flemish, French, or German based on the region.  |
| PRESERVE_INPUT | Returns addresses in the language that the input address uses.<br>Address Verification preserves the language if the reference address database contains the address in the input language. If the input address contains more than one language, Address Verification returns the output in the database language. |
| ALTERNATIVE_1  | Returns the province, locality, and street information in Flemish.  |
| ALTERNATIVE_2  | Returns the province, locality, and street information in French.   |
| ALTERNATIVE_3  | Returns the province, locality, and street information in German.   |

## Example: PreferredLanguage Settings and Results for Belgium

You enter the following Belgian address for verification:

Rue Royale 4

Bruxelles 1000

Belgium

The following table shows the validated results and the EERS codes for different PreferredLanguage settings when you enter the address:

| DATABASE  | ENGLISH  | ALTERNATIVE_1 (Flemish)  | ALTERNATIVE_2 (French)  | ALTERNATIVE_3 (German)   | PRESERVE_INPUT  |
|---|--|--|---|--|---|
| <b>Output address</b><br>Rue Royale 4<br>1000<br>Bruxelles<br>Belgium | <b>Output address</b><br>Rue Royale 4<br>1000<br>Brussels<br>Belgium | <b>Output address</b><br>Koningsstraat 4<br>1000<br>Brussel<br>Belgium | <b>Output address</b><br>Rue Royale 4<br>1000<br>Bruxelles<br>Belgium | <b>Output address</b><br>Rue Royale 4<br>1000 Brüssel<br>Belgium | <b>Output address</b><br>Rue Royale 4<br>Bruxelles<br>1000<br>Belgium |
| <b>EERS:</b><br>00505050000<br>000000000                              | <b>EERS:</b><br>005000E00000<br>00000000                             | <b>EERS:</b><br>005000000000<br>000000                                 | <b>EERS:</b><br>005000000000<br>00000000                              | <b>EERS:</b><br>005000E000000<br>0000000                         | <b>EERS:</b><br>00500000000000<br>000000                              |

The table indicates that the input address and the results for DATABASE, ALTERNATIVE\_2, and PRESERVE\_INPUT are in French. The result for ALTERNATIVE\_1 is Flemish. The table indicates that the street information is available only in Flemish and French. An EERS value of 5 at position seven (Street) for the DATABASE setting indicates that the database contains alternatives: Flemish in this case. For ENGLISH and for ALTERNATIVE\_3, the EERS value for position seven is E. An EERS value of E indicates that the output is in the default database language, French, because the information is not available in the specified language.

## Multilanguage Support for Canada Addresses

You can configure Address Verification to return valid Canada addresses in English or in French.

The Canada reference address data uses the English language for addresses from all provinces except Quebec. The reference data uses the French language for Quebec addresses.

To set the preferred language, configure the PreferredLanguage attribute of the Result element in Parameters.xml.

The following table describes the `PreferredLanguage` values that you can set for Canada address data:

| Option        | Description  |
|---------------|--|
| DATABASE      | Default value. Returns addresses in English for all provinces except Quebec. Returns Quebec addresses in French.   |
| ALTERNATIVE_1 | Returns all addresses in English.  |
| ALTERNATIVE_2 | Returns Quebec addresses in French.<br>In provinces other than Quebec, returns the street descriptors, directional information, and province names in French. Returns other address elements in English. |

### Example: PreferredLanguage Settings and Results for Canada Addresses

The following table shows two Canada addresses and the validated results for different preferred language settings:

| Input Address   | DATABASE  | ALTERNATIVE_1<br>(English)   | ALTERNATIVE_2<br>(French)  | PRESERVE_INPUT  |
|---|---|--|--|---|
| Bankers Hall<br>West<br>Tower<br><br>888 - 3rd St<br>SW<br><br>10th<br>Floor<br><br>Calgary, AB<br>T2P<br>5C5<br><br>Canada | Bankers Hall<br>West<br>Tower<br><br>888 - 3rd St<br>SW<br><br>10th<br>Floor<br><br>Calgary, AB<br>T2P<br>5C5<br><br>Canada | Bankers Hall<br>West<br>Tower<br><br>888 - 3rd St<br>SW<br><br>10th<br>Floor<br><br>Calgary, AB T2P<br>5C5<br><br>Canada | Bankers Hall<br>West<br>Tower<br><br>888 Rue 3 So<br>10th<br>Floor<br><br>Calgary, AB T2P<br>5C5<br><br>Canada | Bankers Hall<br>West<br>Tower<br><br>888 - 3rd St<br>SW<br><br>10th<br>Floor<br><br>Calgary, AB<br>T2P<br>5C5<br><br>Canada |
| 615 Av<br>Monique<br><br>Québec QC G1B<br>2A8<br>Canada   | 615 Av<br>Monique<br><br>Québec QC G1B<br>2A8<br>Canada   | 615 Monique<br>Ave<br><br>Québec QC G1B<br>2A8<br>Canada   | 615 Av Monique<br>Québec QC G1B<br>2A8<br>Canada   | 615 Av<br>Monique<br>Québec QC G1B<br>2A8<br>Canada   |

For the first example in the table, the input address and the results for DATABASE, ALTERNATIVE\_1, and PRESERVE\_INPUT are in English. The result for the ALTERNATIVE\_2 setting is in French.

For the second example, which is an address from Quebec, the input address and the results for DATABASE, ALTERNATIVE\_2, and PRESERVE\_INPUT are in French. The result for the ALTERNATIVE\_1 setting is in English.



## Multilanguage Support for Switzerland Addresses

You can configure Address Verification to return locality information and province (Canton) information in valid Switzerland addresses in French, German, or Italian.

To set the preferred language, configure the `PreferredLanguage` attribute of the `Result` element in `Parameters.xml`.

The following table describes the `PreferredLanguage` values that you can set for Switzerland address data:

| Option         | Description   |
|----------------|---|
| ENGLISH        | Returns the locality and province information in English if the reference address database contains the information in English.<br>Returns the other address elements in the main language of the region to which the address belongs.<br>Address Verification returns the locality information in English for select localities such as Geneva and Zurich. |
| DATABASE       | Default value. Returns addresses in the main language of the region to which the address belongs.<br>For example, Address Verification returns a Zurich address in German and a Geneva address in French.   |
| PRESERVE_INPUT | Returns addresses in the language that the input address uses.<br>Address Verification preserves the language if the reference address database contains the address in the input language. If the input address contains more than one language, Address Verification returns the output in the database language.   |
| ALTERNATIVE_1  | Returns the province and locality information in German.  |
| ALTERNATIVE_2  | Returns the province and locality information in French.  |
| ALTERNATIVE_3  | Returns the province and locality information in Italian.   |

**Note:** Address Verification also returns street information for addresses in Biel/Bienne in the alternative language that you configure.

### Example: PreferredLanguage Settings and Results for Switzerland Addresses

You enter the following Switzerland address for verification:

Beundenweg 36

2503 Biel/Bienne

The following table shows the validated results for different `PreferredLanguage` settings when you enter the address:

| DATABASE                              | ENGLISH                               | ALTERNATIVE_1 (German)                | ALTERNATIVE_2 (French)                  | ALTERNATIVE_3 (Italian)      | PRESERVE_INPUT                        |
|---------------------------------------|---------------------------------------|---------------------------------------|---|------------------------------|---------------------------------------|
| Beundenweg 36<br>2503 Biel/<br>Bienne | Beundenweg 36<br>2503 Biel/<br>Bienne | Beundenweg 36<br>2503 Biel/<br>Bienne | Chemin des<br>Oeuches 36<br>2503 Bienne | Beundenweg 36<br>2503 Bienna | Beundenweg 36<br>2503 Biel/<br>Bienne |

The table indicates that the results for DATABASE, ENGLISH, ALTERNATIVE\_1, and PRESERVE\_INPUT are in German. The result for ALTERNATIVE\_2 is in French.

In the ALTERNATIVE\_3 result, you receive the locality information in Italian. The Switzerland reference address database contains only the locality information for the address in Italian. Informatica Address Verification returns the street information in German, which is the database language for the address. In this case, Address Verification returns an extended element result status (EERS) value of E for the street element. An EERS value of E indicates that the output is in the database language because the reference database does not contain the information in the language that you specified.

## The GlobalPreferredDescriptor Attribute

You can configure Informatica Address Verification to specify the output format for street, building, and subbuilding element descriptors. To specify the output format for element descriptors, configure the `GlobalPreferredDescriptor` attribute of the `Result` parameter in `Parameters.xml`.

**Note:** The `GlobalPreferredDescriptor` parameter works only for address element descriptors in Australia and New Zealand addresses and the *Strasse* element descriptor in Germany addresses.

You can specify one of the following values for `GlobalPreferredDescriptor`:

### DATABASE

Returns the element descriptor available in the reference address database. This is the default value. If there is no matching entry in the database, Address Verification copies the input to the output.

### LONG

Returns the expanded form of the element descriptor.

### SHORT

Returns the abbreviated form of the element descriptor.

### PRESERVE\_INPUT

Copies the element descriptor in the input to the output. If the input element descriptor is not an official name, Address Verification returns the corresponding value from the database in the output. For example, when you validate an Australia address that contains BD for boulevard, Address Verification corrects the descriptor to BVD. BVD is the standard short form for boulevard in Australia.

## The ElementAbbreviation Attribute

You can configure the `ElementAbbreviation` attribute of the `Result` element in `Parameters.xml` to specify whether to use abbreviations for element descriptors in result output.

If you set `ElementAbbreviation` to ON, Address Verification uses standard abbreviations for element descriptors if the reference address database contains the corresponding abbreviations.

The default value of `ElementAbbreviation` is OFF.

If you set `GlobalPreferredDescriptor` to a value other than DATABASE, `GlobalPreferredDescriptor` setting takes precedence over the `ElementAbbreviation` setting for the corresponding descriptors.

# Configuring the Element Descriptor Format

You can configure the `ElementAbbreviation` attribute and the `GlobalPreferredDescriptor` attribute of the `Result` element in `Parameters.xml` to specify how Informatica Address Verification handles element descriptors in the result output.

- To use standard abbreviations for address elements, set the `ElementAbbreviation` attribute of the `Result` element to `ON`.

If the reference address database contains abbreviations for element descriptors, Address Verification uses those abbreviations instead of their expanded forms in the result output.

- To specify the element descriptor for addresses in Australia and New Zealand, set the `GlobalPreferredDescriptor` attribute to one of the following values:
  - `DATABASE`
  - `LONG`
  - `SHORT`
  - `PRESERVE_INPUT`

The `GlobalPreferredDescriptor` setting also applies to the `Strasse` element in Germany addresses.

**Note:** If you set `GlobalPreferredDescriptor` to a value other than `DATABASE`, the `GlobalPreferredDescriptor` setting takes precedence over the `ElementAbbreviation` setting.

## The AliasLocality and AliasStreet Attributes

You can configure the `AliasLocality` and `AliasStreet` attributes of the `Result` element in `Parameters.xml` to specify how you want Informatica Address Verification to handle aliases for locality and street names.

You can configure the `AliasLocality` and `AliasStreet` attributes to one of the following values:

### **PRESERVE**

Retains the alias for the street or locality in the output.

### **OFFICIAL**

Returns the street or locality name as mandated by the postal regulations of the country in the output. This is the default value. Based on the postal regulations for the country to which the address belongs, Address Verification returns the postal name or the street or locality alias or vanity name.

### **OFF**

Returns the postal name for the street or locality in the output.

When you validate addresses in the certified mode, set the `AliasStreet` and `AliasLocality` attributes to `OFFICIAL`.

If you want Address Verification to preserve the vanity name or the alias in the validated output, you must set the respective parameters to `PRESERVE`. If you want Address Verification to return the postal name of the locality or street in the validated output, set the respective parameters to `OFF`.

## Configuring AliasStreet and AliasLocality

You can configure Informatica Address Verification to return the postal name or alias or vanity name of a street or locality in the result output.

In `Parameters.xml`, configure the following attributes of the `Result` element:

### AliasStreet

Specifies the output for the street information. Set one of the following values:

- **PRESERVE**. Retains the alias name of the street.
- **OFFICIAL**. Returns either the postal name or the alias name based on the postal regulations of the country.
- **OFF**. Returns the postal name of the street.

The default value is **OFFICIAL**.

### AliasLocality

Specifies the output for the locality information. Set one of the following values:

- **PRESERVE**. Retains the alias name of the locality.
- **OFFICIAL**. Returns either the postal name or the alias name based on the postal regulations of the country.
- **OFF**. Returns the postal name of the locality.

The default value is **OFFICIAL**.

## The RangesToExpand and FlexibleRangeExpansion Attributes

You can configure the `RangesToExpand` attribute of the `Result` element in `Parameters.xml` to specify whether to expand house number ranges in addresses from countries for which individual house numbers are available. The `FlexibleRangeExpansion` attribute of the `Result` element specifies the scope of range expansion in suggestion lists. If you set the `FlexibleRangeExpansion` attribute to **ON**, Informatica Address Verification expands only the initial items in the suggestion list. Address Verification retains the ones at the end of the list as ranges if the number of items exceeds the maximum result count.

If you set the `RangesToExpand` attribute to **ALL**, Address Verification expands the house number ranges whenever individual house numbers are available. If you set `RangesToExpand` to **ONLY\_WITH\_VALID\_ITEMS**, Address Verification expands the ranges only if the reference data contains the complete list of house numbers within the specified range. You can turn off the range expansion by setting the `RangesToExpand` attribute to **NONE**. The default value is **NONE**. When the complete list of house numbers is not available in the reference address database, the **ONLY\_WITH\_VALID\_ITEMS** setting works the same way as **NONE**.

You can use the `RangesToExpand` attribute in conjunction with the `FlexibleRangeExpansion` attribute to achieve the optimum results. If you set `FlexibleRangeExpansion` to **ON**, Address Verification expands only the initial items in a list. Address Verification internal logic decides the number of results to expand and how many to keep as ranges without exceeding the `MaxResultCount`. When you set `FlexibleRangeExpansion` to **on**, the suggestion list might contain both expanded and unexpanded ranges for house numbers and buildings. The number of results and the nature of the suggestion list depend on the values you specify for `MaxResultCount`, `RangesToExpand`, and `FlexibleRangeExpansion`.

## Configuring the Range Attributes

You can configure the following attributes of the `Result` element in `Parameters.xml` to specify how you want Informatica Address Verification to handle number ranges in addresses.

- To specify whether or not to expand the ranges, set the `RangesToExpand` attribute to one of the following values.
  - ALL. Expands the ranges always.
  - ONLY\_WITH\_VALID\_ITEMS. Expands the ranges only when the reference address database contains full list of numbers for the specified range.
  - NONE. Does not expand a range.
- To configure flexible expansion of ranges, set the `FlexibleRangeExpansion` to ON.

`FlexibleRangeExpansion` is set to ON by default. To disable flexible expansion of ranges, set `FlexibleRangeExpansion` to OFF.

## The StandardizeInvalidAddresses Attribute

You can configure Informatica Address Verification to standardize invalid address elements. Standardizing invalid address elements can improve downstream business processes, such as matching and deduplication.

When the `StandardizeInvalidAddresses` attribute is set to ON, Address Verification standardizes address elements, such as the elements listed below, in invalid addresses or addresses that have an Ix process status.

- Street types
- Pre and post directionals
- Delivery service items
- Subbuilding descriptors
- State/province/regions

For example, Address Verification can standardize California to CA.

The default value for `StandardizeInvalidAddresses` is OFF.

## Standardizing Nonvalid Addresses

You can configure Informatica Address Verification to standardize addresses even though the addresses fail to validate.

- In `Parameters.xml`, set the `StandardizeInvalidAddresses` attribute of the `Result` element to ON.

The default value for this attribute is OFF.

## The DualAddressPriority Attribute

Dual addresses are addresses that contain both street address data and delivery service address data. You can configure the `DualAddressPriority` attribute of the `Result` element to specify which address type you

want Informatica Address Verification to validate an input address against if the address contains both types of address data.

You can set the `DualAddressPriority` attribute to `DELIVERY_SERVICE` if you want Address Verification to validate the delivery service part of the input address. If you set `DualAddressPriority` to `STREET`, Address Verification validates the street address part of the input address.

The default value for `DualAddressPriority` is `POSTAL_ADMIN`, which denotes that Address Verification validates addresses based on the postal department's preference for the address type. When you validate addresses in the certified mode, set `DualAddressPriority` to `POSTAL_ADMIN`.

## Configuring the Type of Address to Validate

You can specify the type of address that Address Verification validates when the input address contains both street address data and delivery service address data.

- In `Parameters.xml`, set the `DualAddressPriority` attribute of the `Result` element to one of the following values.
  - `DELIVERY_SERVICE`. Validates the address as a delivery service address.
  - `POSTAL_ADMIN`. Validates the address based on the postal regulations of the country to which the address belongs.
  - `STREET`. Validates the address as a street address.

When you validate addresses in the certified mode, set the `DualAddressPriority` attribute to `POSTAL_ADMIN`.

## Output Standardization

When you validate an address, Address Verification returns the output that meets the specifications from the local postal authorities and the Universal Postal Union (UPU). You can modify the output by configuring the address element standardization options in the `Parameters.xml` file.

The address element-level standardization options include casing, maximum item count, and maximum length of the string. If you do not want to configure these attributes at the address element level, you can set global values for these attributes. If you configure global and address-element level values for `MaxLines`, `MaxLength`, and `Casing` attributes, the address element-level values override the global settings.

The following table describes the `MaxLines`, `MaxLength`, and `Casing` attributes and supported values for different address elements:

| Address Element             | Default Value for Casing | Minimum Recommended value for MaxLength | Value Range Supported for MaxItemCount | Default Value for MaxItemCount |
|-----------------------------|--------------------------|---|--|--------------------------------|
| RecipientLine               | NOCHANGE                 | 25                                      | 1 through 6                            | 6                              |
| DeliveryAddressLine         | NOCHANGE                 | 25                                      | 1 though 6                             | 6                              |
| CountrySpecificLocalityLine | NOCHANGE                 | 25                                      | 1 through 6                            | 6                              |

| Address Element      | Default Value for Casing | Minimum Recommended value for MaxLength | Value Range Supported for MaxItemCount | Default Value for MaxItemCount |
|----------------------|--------------------------|---|--|--------------------------------|
| FormattedAddressLine | NOCHANGE                 | 25                                      | 1 through 19                           | 19                             |
| AddressComplete      | GLOBAL                   | 25                                      | Not Applicable                         | Not Applicable                 |
| Country              | GLOBAL                   | 2                                       | 1 through 3                            | 3                              |
| Locality             | GLOBAL                   | 20                                      | 1 through 6                            | 6                              |
| PostalCode           | GLOBAL                   | 5                                       | 1 through 3                            | 3                              |
| Province             | GLOBAL                   | 2                                       | 1 through 6                            | 6                              |
| Street               | GLOBAL                   | 20                                      | 1 through 6                            | 6                              |
| Number               | GLOBAL                   | 5                                       | 1 through 6                            | 6                              |
| Building             | GLOBAL                   | 25                                      | 1 through 6                            | 6                              |
| SubBuilding          | GLOBAL                   | 25                                      | 1 through 6                            | 6                              |
| DeliveryService      | GLOBAL                   | 25                                      | 1 through 3                            | 3                              |
| Organization         | GLOBAL                   | 25                                      | 1 through 3                            | 3                              |
| Department           | GLOBAL                   | 25                                      | 1 through 3                            | 3                              |
| Contact              | GLOBAL                   | 25                                      | 1 through 3                            | 3                              |
| FirstName            | GLOBAL                   | 20                                      | 1 through 3                            | 3                              |
| MiddleName           | GLOBAL                   | 20                                      | 1 through 3                            | 3                              |
| LastName             | GLOBAL                   | 20                                      | 1 through 3                            | 3                              |
| Title                | GLOBAL                   | 20                                      | 1 through 3                            | 3                              |
| Function             | GLOBAL                   | 20                                      | 1 through 3                            | 3                              |
| Salutation           | GLOBAL                   | 20                                      | 1 through 3                            | 3                              |
| Gender               | GLOBAL                   | 1                                       | 1 through 3                            | 3                              |
| Residue              | GLOBAL                   | 25                                      | 1 through 6                            | 6                              |

At the address element level, you can set one of the following values for the `GlobalCasing` attribute:

#### **NATIVE**

To return the result output in the same casing that the data in the reference address database uses.

#### **UPPER**

To return the result output in uppercase letters.

**LOWER**

To return the result output in lowercase letters.

**MIXED**

To return the result output in mixed case and according to any country-specific casing guidelines.

**NOCHANGE**

To return the result output in the same casing that is used for the input address.

**GLOBAL**

To return the result output according to the value set for the `GlobalCasing` attribute.

## Configuring Output Standardization at Address Element Level

You can configure output standardization attributes at the address element level.

1. In `Parameters.xml`, specify one of the following subelements of the `AddressElementStandardize` element:
  - `RecipientLine`
  - `DeliveryAddressLine`
  - `CountrySpecificLocalityLine`
  - `FormattedAddressLine`
  - `AddressComplete`
  - `Country`
  - `Locality`
  - `PostalCode`
  - `Province`
  - `Street`
  - `Number`
  - `Building`
  - `SubBuilding`
  - `DeliveryService`
  - `Organization`
  - `Department`
  - `Contact`
  - `FirstName`
  - `MiddleName`
  - `LastName`
  - `Title`
  - `Function`
  - `Salutation`
  - `Gender`
  - `Residue`
2. To specify casing, set the `Casing` attribute to one of the following values:



- NATIVE
  - UPPER
  - LOWER
  - MIXED
  - NOCHANGE
  - GLOBAL
3. Set the `MaxLength` attribute to a value that is not less than the minimum recommended value specified in the table in the preceding section.
  4. Set the `MaxItemCount` attribute to a value in the range supported for the address element specified in the first step. For value ranges supported for each of the address elements, see the table in the preceding section.

## CHAPTER 7

# Output Fields

This chapter includes the following topics:

- [Output Fields Overview, 106](#)
- [The Result Element, 106](#)
- [The ResultData Element, 107](#)

## Output Fields Overview

When you process an address, you receive a result output that contains a number of fields such as address elements, address line elements, and enrichment values. The result output also contains process-related information and status and assessment codes.

The `Result.dtd` file defines the output formats and fields. The result output you receive depends on the process parameters and result settings that you configure.

## The Result Element

The `Result` element in `Result.xml` contains the following attributes:

### **ProcessStatus**

Indicates status of the process call. For more information see the *Address Verification Process Status Values* section.

### **ModeUsed**

Indicates the process mode that you specified. The possible values are:

- ADDRESSCODELOOKUP
- BATCH
- CERTIFIED
- COUNTRYRECOGNITION
- FASTCOMPLETION
- INTERACTIVE
- PARSE

**Count**

Indicates the number of results. The value range is between 0 through 100. The default value is 0.

**CountOverflow**

Indicates whether there are more results available when you verify addresses in the interactive or fast completion mode. The possible values are YES and NO. The default value is NO.

**CountryISO3**

Contains the country ISO3 code.

**PreferredScript**

Contains the preferred language you configured. The possible values are:

- ASCII\_EXTENDED
- ASCII\_SIMPLIFIED
- DATABASE
- LATIN
- LATIN\_1
- LATIN\_ALT
- POSTAL\_ADMIN\_PREF
- POSTAL\_ADMIN\_ALT
- PRESERVE\_INPUT

**PreferredLanguage**

Contains the preferred language you configured. The possible values are:

- ALTERNATIVE\_1
- ALTERNATIVE\_2
- ALTERNATIVE\_3
- DATABASE
- ENGLISH
- PRESERVE\_INPUT

## The ResultData Element

The `ResultData` element contains multiple subelements and attributes. The attributes of `ResultData` contain various assessment codes that help you analyze results.

### AddressElements

The `AddressElements` subelement of `ResultData` contains the following elements:

**Key**

Contains the record ID or transaction key. The attributes include:

Type. The possible values are `RECORD_ID` and `TRANSACTION_KEY`. The default value is `RECORD_ID`.

Item. Number of items. The range is 1 through 3. The default value is 1.

### **Country**

Contains the country name or the country code based on the configuration for the `CountryType` attribute in `Parameters.xml`. The attributes include:

Type. Type of country information. The possible values are ABBREVIATION, ISO2, ISO3, ISO\_NUMBER, NAME\_CN, NAME\_DA, NAME\_DE, NAME\_EN, NAME\_ES, NAME\_FI, NAME\_FR, NAME\_GR, NAME\_HU, NAME\_IT, NAME\_JP, NAME\_KR, NAME\_NL, NAME\_PL, NAME\_PT, NAME\_RU, NAME\_SA, and NAME\_SE. The default value is NAME\_EN.

Item. Number of items. The range is 1 through 3. The default value is 1.

### **Locality**

Contains the locality information. The attributes include:

- Type. The type of locality information. The possible values are COMPLETE, NAME, PREFERRED\_NAME, SORTING\_CODE, and ADD\_INFO. The default value is COMPLETE.
- Item. The number of items. The range is 1 through 6. The default value is 1.

### **PostalCode**

Contains the postal code. The attributes include:

- Type. The type of postal code information. The possible values are FORMATTED, UNFORMATTED, BASE, and ADD\_ON. The default value is FORMATTED.
- Item. The number of items. The range is 1 through 3. The default value is 1.

### **Province**

Contains the province information. The attributes include:

- Type. The type of province information. The possible values are COUNTRY\_STANDARD, ABBREVIATION, EXTENDED, and ISO. The default value is COUNTRY\_STANDARD.
- Item. The number of items. The range is 1 through 6. The default value is 1.

### **Street**

Contains the street information. The attributes include:

- Type. The type of street information. The possible values are COMPLETE, COMPLETE\_WITH\_NUMBER, NAME, PRE\_DESCRIPTOR, POST\_DESCRIPTOR, PRE\_DIRECTIONAL, POST\_DIRECTIONAL, and ADD\_INFO. The default value is COMPLETE.
- Item. The number of items. The range is 1 through 6. The default value is 1.

### **Number**

Contains the information such as house or building number. The attributes include:

- Type. The type of number information. The possible values are COMPLETE, NUMBER, DESCRIPTOR, and ADD\_INFO. The default value is COMPLETE.
- Item. The number of items. The range is 1 through 6. The default value is 1.

### **Building**

Contains the building information. The attributes include:

- Type. The type of building information. The possible values are COMPLETE, COMPLETE\_WITH\_SUBBUILDING, NAME, NUMBER, and DESCRIPTOR. The default value is COMPLETE.

- Item. The number of items. The range is 1 through 6. The default value is 1.

#### **SubBuilding**

Contains the subbuilding information. The attributes include:

- Type. The type of building information. The possible values are COMPLETE, NAME, NUMBER, and DESCRIPTOR. The default value is COMPLETE.
- Item. The number of items. The range is 1 through 6. The default value is 1.

#### **DeliveryService**

Contains the delivery service information. The attributes include:

- Type. The type of delivery service information. The possible values are COMPLETE, DESCRIPTOR, NUMBER, and ADD\_INFO. The default value is COMPLETE.
- Item. The number of items. The range is 1 through 3. The default value is 1.

#### **Organization**

Contains the organization information. The attributes include:

- Type. The type of organization information. The possible values are COMPLETE, NAME, DESCRIPTOR, and DEPARTMENT. The default value is COMPLETE.
- Item. The number of items. The range is 1 through 3. The default value is 1.

#### **Contact**

Contains the contact information. The attributes include:

- Type. The type of contact information. The possible values are COMPLETE, FIRST\_NAME, MIDDLE\_NAME, LAST\_NAME, NAME, TITLE, FUNCTION, SALUTATION, and GENDER. The default value is COMPLETE.
- Item. The number of items. The range is 1 through 3. The default value is 1.

#### **Residue**

Contains superfluous or additional information. The attributes include:

- Type. The type of contact information. The possible values are NECESSARY, SUPERFLUOUS, and UNRECOGNIZED. The default value is UNRECOGNIZED.
- Item. The number of items. The range is 1 through 6. The default value is 1.

## **Province ISO Codes**

You receive the province ISO code in batch, interactive, and fast completion results. Informatica Address Verification returns only the province part, an alphanumeric string of up to three characters, of the ISO 3166-2 code.

You receive province ISO codes in the validated output for the following countries:

- Australia
- Austria
- Bulgaria
- Belarus
- Brazil
- Canada
- Switzerland

- China
- Germany
- Finland
- France
- Greenland
- Haiti
- Hungary
- Indonesia
- Italy
- Jamaica
- Japan
- Moldova
- Madagascar
- Mexico
- Mozambique
- Netherlands
- Oman
- Pakistan
- Philippines
- Poland
- Portugal
- Paraguay
- Romania
- Russian Federation
- Sweden
- Ukraine
- United States
- South Africa
- South Korea

## AddressLines

The `AddressLines` element contains the following subelements:

### **RecipientLine**

Contains the recipient information such as contact or organization name. Informatica Address Verification supports up to six lines of recipient information. The `Line` attribute indicates the number of address lines in `RecipientLine`. The range is from 1 through 6. The default value is 1.

### **DeliveryAddressLine (DAL)**

Contains the delivery information such as street, house number, building, and delivery service information. Address Verification supports up to six lines of delivery address line information. The `Line`

attribute indicates the number of address lines in `DeliveryAddressLine`. The range is from 1 through 6. The default value is 1.

**CountrySpecificLocalityLine (CSLLN)**

Contains information such as locality, postal code, province, and country details. Address Verification supports up to six lines of country specific locality line information. The Line attribute indicates the number of address lines in `CountrySpecificLocalityLine`. The range is from 1 through 6. The default value is 1.

**FormattedAddressLine (FAL)**

Contains unfielded data that includes recipient information, delivery address information, and locality information. Address Verification typically returns the FAL output in the address format of the country to which the address belongs. Even if an address is not valid, Address Verification returns the parsed address output in the FAL. You can use the FAL output to print the address on envelopes or to generate address labels. Address Verification supports up to 19 lines of formatted address line information. The Line attribute indicates the number of address lines in `FormattedAddressLine`. The range is from 1 through 19. The default value is 1.

## AddressComplete

The `AddressComplete` element contains the complete address as one string. Informatica Address Verification separates the address elements with delimiters such as comma or semicolon.

## CHAPTER 8

# Assessment Codes and Return Codes

This chapter includes the following topics:

- [Status Codes, Assessment Codes, and API Return Codes, 112](#)
- [Address Verification Process Status Values, 113](#)
- [Element Positions, 115](#)
- [Element Input Status, 116](#)
- [Element Relevance Value, 116](#)
- [Element Result Status, 117](#)
- [Address Resolution Code, 118](#)
- [Extended Element Result Status, 118](#)
- [Result Percentage Values, 119](#)
- [LanguageISO3 Code, 119](#)
- [Mailability Score, 120](#)
- [Address Types, 120](#)
- [Certified Mode Process Status Values, 124](#)
- [API Return Codes, 126](#)

## Status Codes, Assessment Codes, and API Return Codes

Informatica Address Verification returns status and assessment codes to help you understand and analyze the result output better.

The process status codes provide a high-level summary of the address verification process. The assessment codes that Address Verification returns include element input status, element result status, extended element result status, and element relevance values.

The assessment codes provide granular, element-level details. You can use the assessment codes to assess the following values:

- Quality of the input address and result.



- Probability of successful delivery to the address.
- Postal relevance of address elements.
- Reasons for rejection in case of address that are not valid.
- Level of similarities between the input address and the result.

Apart from the status codes and assessment codes in the result, you also receive return codes that indicate the status of API functions. API function calls provide information about the status of an API call.

## Address Verification Process Status Values

The process status values that `AD_GetResultXML()` or `AD_GetResultParameter()` returns describe the result output quality of a process call to Informatica Address Verification. The process status values provide a high-level description of the status. You can use additional assessment codes such as Element Input Status, Element Result Status, and Extended Element Result Status for specific information on address elements.

| Status Code | Description  |
|-------------|--|
| V4          | Verified. The input data is correct. Address Verification checked all postally relevant elements, and inputs matched perfectly.                      |
| V3          | Verified. The input data is correct, but Address Verification standardized some or all elements, or the input contains outdated names or exonyms.    |
| V2          | Verified. The input data is correct, but Address Verification could not verify some elements because of incomplete reference data.                   |
| V1          | Verified. The input data is correct, but user standardization has adversely affected deliverability. For example, the post code length is too short. |
| C4          | Corrected. Address Verification has checked all postally relevant elements.  |
| C3          | Corrected. Address Verification could not check some elements.   |
| C2          | Corrected, but the delivery status is unclear because of absent reference data.  |
| C1          | Corrected, but the delivery status is unclear because user standardization introduced errors.  |
| I4          | Address Verification could not correct the data completely, but there is a single match with an address in the reference data.                       |
| I3          | Address Verification could not correct the data completely, and there are multiple matches with addresses in the reference data.                     |
| I2          | Address Verification could not correct the data. Interactive mode returns partial suggested addresses.   |
| I1          | Address Verification could not correct the data completely. Batch mode cannot suggest an address.  |
| N7          | Verification error. Verification did not take place because single-line verification is not unlocked.  |
| N6          | Verification error. Verification did not take place because single-line verification is not supported for the destination country.                   |

| Status Code | Description  |
|-------------|--|
| N5          | Verification error. Verification did not take place because the reference database is out of date.   |
| N4          | Verification error. Verification did not take place because the reference data is corrupt or badly formatted.  |
| N3          | Verification error. Verification did not take place because the country data cannot be unlocked.   |
| N2          | Verification error. Verification did not take place because the required reference database is not available.  |
| N1          | Verification error. Verification did not take place because the country is not recognized or not supported.  |
| Q3          | Fast Completion mode. Address Verification can retrieve one or more complete addresses from the address reference data that correspond to the input address.                 |
| Q2          | Fast Completion mode. Address Verification can combine the input address elements and elements from the address reference data to create a complete address.                 |
| Q1          | Fast Completion mode. Address Verification cannot suggest a complete address. To generate a complete address suggestion, add data to the input address.                      |
| Q0          | Fast Completion mode. There is insufficient input data to generate a suggestion.   |
| S4          | Parse mode. The address was parsed perfectly.  |
| S3          | Parse mode. The address was parsed with multiple results.  |
| S2          | Parse mode. Address Verification parsed the address with errors because of the changed element positions.  |
| S1          | Parse mode. There was a parsing error due to an input format mismatch.   |
| RB          | Country recognized from abbreviation. Recognizes ISO two-character and ISO three-character country codes. Can also recognize common abbreviations such as "GER" for Germany. |
| RA          | Country assigned by the ForceCountryISO3 setting.  |
| R9          | Country recognized from the Default CountryISO3 setting.   |
| R8          | Country recognized from the country name.  |
| R7          | Country recognized from the country name, but Address Verification identified errors in the country data.  |
| R6          | Country recognized from territory data.  |
| R5          | Country recognized from province data.   |
| R4          | Country recognized from major town data.   |
| R3          | Country recognized from the address format.  |
| R2          | Country recognized from the script.  |
| R1          | Country not recognized because multiple matches are available.   |

| Status Code | Description   |
|-------------|---|
| R0          | Country not recognized.   |
| A1          | Address code lookup found a partial address or a complete address for the input code. |
| A0          | Address code lookup found no address for the input code.                              |

Address Verification returns the Vx, Cx, and Ix process status codes for BATCH, INTERACTIVE, or CERTIFIED process() calls. Address Verification returns Qx for FASTCOMPLETION mode requestes, Sx for PARSE mode requestes, Rx for COUNTRYRECOGNITION, and Ax for ADDRESSCODELOOKUP.

Address Verification returns Nx process status codes in case of an error. Typically, batch and interactive modes return the same process status for the same input address. However, in case of an incorrect numeric entry, Address Verification returns different codes in batch (I4) and interactive (I3).

In batch, interactive, and certified modes, Address Verification returns parsed output for Ix records. However, ensure that you manually review the Ix results before using the results for any database update.

Address Verification returns country ISO3 codes for countries that are no longer supported. For example, Soviet Union (SUN) or the Netherlands Antilles (ANT).

The CountryISO3 field remains empty if Address Verification does not recognize a country when it verifies an address in process modes other than the country recognition mode.

## Element Positions

For each element, level 1 indicates the status of the first item and level n summarizes the status of items 2 to 6. For example, if the input address contains 3 levels of locality, the assessment value at position 3 points to locality 1, whereas the assessment value at position 4 summarizes the assessment values for locality 2 and 3.

1. PostalCode level 1
2. PostalCode level n (for example, ZIP+4 – Plus 4 addition)
3. Locality level 1
4. Locality level n (for example, Urbanisation, Dependent Locality)
5. Province level 1
6. Province level n (for example, Sub Province)
7. Street level 1
8. Street level n (for example, Dependent Street)
9. Number level 1
10. Number level n
11. Delivery service level 1 (for example, PO Box, GPO, Packstation, Private Bags)
12. Delivery service level n
13. Building level 1
14. Building level n
15. SubBuilding level 1

- 16. SubBuilding level n
- 17. Organization level 1
- 18. Organization level n
- 19. Country level 1 (Mother country)
- 20. Country level n (for example, Territory)

## Element Input Status

The following values are possible for Element Input Status (`ElementInputStatus`, EIS):

- 0. The input address contains no data at this position.
- 1. The data at this position is not available in the reference data.
- 2. The data at this position cannot be checked because reference data is missing.
- 3. The data is incorrect. The reference database suggests that the Number or DeliveryService value is outside the range expected by the reference data. In batch and certified modes, Informatica Address Verification returns the input data at this position without making any correction. In the suggestion list modes such as fast completion and interactive, Address Verification returns suggestions to choose from.
- 4. The data at this position matches the reference data, but with errors.
- 5. The data at this position matches the reference data, but the data element was corrected or standardized. For example:
  - Parsing: Splitting of house number for “MainSt 1”
  - Verification: Replacing an input that is an exonym, or dropping superfluous fielded input that is not valid according to the country reference database
- 6. The data at this position matches the reference data without any error.

In the Parse mode, Address Verification returns the following values:

- 0. The input address contains no data at this position.
- 1. The element at this location was moved to another position.
- 2. The element at this position matched the reference data value but needed to be normalized.
- 3. The data at this position has not been changed.

## Element Relevance Value

You can use the Element Relevance value to understand which of the address elements are relevant for an address according to the local postal guidelines. Informatica Address Verification returns a value of 1 for address elements that are relevant, and 0 for elements that are not relevant.

All address elements with an Element Relevance value of 1 must be present for the local postal authority to consider an address valid. Element Relevance might vary from address to address for countries with different address types, such as rural address and metropolitan addressing. Furthermore, address elements that have

actually been validated against reference data might override the default Element Relevance value defined for that address element.

**Note:** Element Relevance values are available only for addresses with a process status value of Cx or Vx in batch and interactive modes or I3 or I4 values in the interactive mode. Other assessment codes such as Element Input Status, Element Result Status, Extended Element Result Status, and Address Resolution Code always return a value regardless of the process status.

## Element Result Status

Informatica Address Verification sets the Element Result Status (`ElementResultStatus`, ERS) value after verification to indicate whether verification ("verified") or correction ("changed") was possible.

The following values are possible for the address elements in positions 1 through 18:

0. The output address contains no data at this position.
1. The data at this position cannot be found in the reference data. The input data is copied to the output data.
2. Data at this position is not checked but is standardized.
3. Data at this position is checked but does not match the expected reference data. The reference data suggests that the number data is not in the valid range. The input data is copied to the output. The status value applies in batch mode only.
4. Data at this position is copied from the input because the corresponding reference data is not available.
5. Data at this position is valid but not changed because multiple matches exist in the reference data. The status value applies in batch mode only.
6. Verification deleted the input value at this position.
7. Data at this position is valid but contained a spelling error. Verification corrected the error by copying the value from the reference data.
8. Data at this position is valid and updated by adding a value from the reference data. It can also mean that the reference database contains additional data for the input element. For example, verification can add a building or sub-building number if a perfect match is found for the street name or building name.
9. Data at this position is valid but not changed, and the delivery status is not clear. For example, the DPV value is wrong.
- C. Data at this position is valid and verified, but the name data is out of date. Verification changed the name data.
- D. Data at this position is valid and verified but changed from an exonym to an official name.
- E. Data at this position is valid and verified. However, verification standardized the character case or the language. Verification can change the language if the value fully matches a language alternative. For example, verification can change "Brussels" to "Bruxelles" in a Belgium address.
- F. Data at this position is valid, verified, and not changed, due to a perfect match against the reference data

Positions 19 and 20 in the output string relate to country data. The country data values apply to the COUNTRYRECOGNITION process mode also.

The following values are possible for the address elements in positions 19 through 20:

0. The output address contains no data at this position.

1. The country is not recognized.
4. The country is recognized from the DefaultCountryISO3 setting.
5. The country is not recognized because multiple matches are available.
6. The country is recognized from a script.
7. The country is recognized from the address format.
8. The country is recognized from major town data.
9. The country is recognized from province data.
- C. The country is recognized from territory data.
- D. The country is recognized from the country name, but the name contains errors.
- E. The country is recognized from the country name without errors.
- F. The country is recognized from the ForceCountryISO3 setting.

## Address Resolution Code

Address resolution code (`AddressResolutionCode`, ARC) is a 20-character output string similar to the element status fields. Informatica Address Verification returns ARC values for nonvalid addresses with `ix` process status code. ARC describes why Address Verification rejected an address and directs you to possible resolutions. Address Verification generates the following ARC values:

2. Missing element in address.
3. Numeric provided inside element is outside permissible range – for example, wrong numeric inside street name or house number; 100 Main St when house numbers range from 400-800 on Main St.
4. Multiple inputs for the element.
5. Input element ambiguous / multiple matches.
6. Element contradicts other elements. For example, the postal code information and locality information do not match.
7. Three strike rule/too many corrections in combination of several elements.
8. General Postal Authority Rule.

In all other scenarios, Address Verification returns the default ARC value, 0.

## Extended Element Result Status

You can use the Extended element result status (`ExtElementStatus`, EERS) code to understand a result and to see whether more information is available in the database. The EERS code is a 20-character output string similar to the Element Status fields for valid or corrected addresses.

Informatica Address Verification returns the following values for EERS:

1. Data available for the element in the database, but not used in the output.
2. Element unchecked, but changed because of wrong syntax/format.
3. Numeric in element correct, but element changed because of wrong syntax/descriptor.

4. Element correct or unchecked, but moved because of wrong format.
  5. Alternative available in database – for example, language, preferred locality name, alias name.
  6. Unvalidated parts inside element like additional information.
  7. Level change like moving HNO1 to HNO2 or swapping Locality2 with Locality1.
  8. Type changed for fielded input only; for example, moving SubBuilding to Building Level 2.
  9. General Postal Authority Rule.
- A. Dominant match for dual address processing.
  - B. Relevance is only a country-wide default and cannot be trusted.
  - C. Fast Completion Overflow.
  - D. Numeric for range expansion (interpolated).
  - E. Language not available for the country, database language returned.
  - F. Output address is outdated.

In all other scenarios, Address Verification returns the default EERS value, 0.

## Result Percentage Values

Informatica Address Verification returns a result percentage (`ResultPercentage`) value in the output to help you understand how similar a result is to the parsed input. Values close to 100% indicate high similarity.

You can use the result percentage values to filter out results with extensive corrections if your data requirements are stringent. Similarly, you can also use the result percentage values to identify Interactive mode suggestions that have the least deviation from the input.

Informatica discourages using the result percentage values for any other use case scenarios than the two described in the preceding paragraph.

## LanguageISO3 Code

When a result output contains data from the reference address database, Informatica Address Verification specifies the language in the ISO 639 3-letter code in the output. For example, `DEU` for German.

For transliterated output, Address Verification specifies the original language in the output. For example, `JPN` in case of romanized Japanese output.

If the reference address database does not contain language information, Address Verification returns a value of `xxx` in the LanguageISO3 code field.

# Mailability Score

Informatica Address Verification returns a mailability score which indicates the likelihood of delivery to an address on a 0 through 5 scale. You can use the mailability score to make informed decisions about mailing to an address.

Address Verification returns the following mailability scores for verified address outputs.

## **5. Completely Confident**

Indicates that Address Verification checked and verified all relevant elements of the address that you entered.

## **4. Almost Certain**

Indicates either of the following scenarios:

Some of the relevant elements of the address could not be checked due to reference data and the rest of the address have been verified in the process.

All relevant elements have been entered and some of the relevant elements of the address have been corrected in the process with a very high confidence. This only happens if the match was unique and the number of discrepancies was very low.

## **3. Should Be Fine**

Indicates that Address Verification corrected some of the relevant elements of the address during verification. A correction only happens if the match was unique and the number of discrepancies was acceptable.

## **2. Fair Chance**

Indicates that Address Verification could not correct or verify the address because of either of the following cases:

A candidate match could not be made with sufficient confidence

Multiple candidate matches could be found with similar confidence levels (multi-match situation).

Mails to the input address, therefore, have a fair chance of getting delivered.

## **1. Risky**

Indicates that Address Verification could only find partial matches for the input address.

## **0. Undeliverable**

Indicates that the input address is missing too many elements or Address Verification could not verify a majority of the elements as Address Verification could not find matches in the reference data.

# Address Types

You can use the AddressType information in the result output to identify the type of mail box an address points to.

For United States addresses, Informatica Address Verification returns the address type values that the United States Postal Service specifies. The United States Postal Service includes a Record Type value in the reference data for domestic addresses.

New Zealand Posts provides a rural address identifier. Canada Post provides a large volume receiver identifier.



If the reference data does not contain a formal address type designator, Address Verification uses different data elements to assign address types to addresses. Address Verification assigns address type values to addresses from countries that do not define address types based on criteria that Address Verification defined.

Address Verification employs a range of criteria to decide the address type for the following countries:

- Australia
- Canada
- France
- New Zealand

Address Verification has a separate set of criteria for identifying the address type when verifying the United States addresses in the Fast Completion mode.

## Address Types from the United States Postal Service

Informatica Address Verification returns the United States Postal Service address type for a United States address when you perform verification in Batch, Certified, or Interactive mode.

The United States Postal Service provides the following address types for United States addresses:

- F. The address identifies an organization.
- G. The address is a general delivery address. In a general delivery address, the postal code and the recipient data identify the address.
- H. The address identifies a high-rise building. The address contains subbuilding elements such as apartment or suite.
- P. The address identifies a Post Office Box or a delivery service.
- R. The address is a rural route/highway contract address.
- S. The address identifies a street.
- U. Unidentified. The address is not valid, and Address Verification does not assign an address type.

## Address Type Indicators for the United States Addresses in Fast Completion Mode

When verifying addresses in the fast completion mode, Informatica Address Verification returns the following address type values:

- B. The address identifies a building.
- F. The address identifies an organization.
- L. The address post code identifies the organization as a large volume receiver. The reference data adds or verifies the organization name.  
Address Verification can determine that the address is a large volume receiver in one of the following ways:
  - The address post code identifies the organization as a large volume receiver.
  - The reference data does not contain street or building information.
- P. The address identifies a Post Office Box or a delivery service.
- S. The address identifies a street. S is the default address type. If Address Verification cannot determine the address type from the address data, it returns the default value.
- U. Unidentified. The address is not valid, and Address Verification does not assign an address type.

If an address meets the criteria for more than one address type, Address Verification assigns the first applicable address type from the following list:

L, F, P, B, S

## Address Type Indicators for Australia Addresses

Informatica Address Verification returns the following address type values for Australia addresses:

- B. The address identifies a building.
- F. The address identifies an organization.
- L. The address post code identifies the organization as a large volume receiver. The reference data adds or validates the organization name.  
Address Verification can determine that the address is a large volume receiver in one of the following ways:
  - The address post code identifies the organization as a large volume receiver.
  - The reference data does not contain street or building information.
- P. The address identifies a Post Office Box or a delivery service.
- S. The address identifies a street. S is the default address type. If Address Verification cannot determine the address type from the address data, it returns the default value.
- U. Unidentified. The address is not valid, and Address Verification does not assign an address type.

If an address meets the criteria for more than one address type, Address Verification assigns the first applicable address type from the following list:

L, F, P, B, S

## Address Type Indicators for Canada Addresses

Informatica Address Verification returns the following address types for Canada addresses:

- B. The address identifies a building.
- F. The address identifies an organization. In Canada addresses, the type F addresses are a subset of the type L addresses. Therefore, the address type F also indicates a large volume receiver.
- G. The address is a general delivery address. In a general delivery address, the postal code and the recipient data identify the address. Address Verification uses the delivery record in the reference data to identify the address type.
- L. The address post code identifies the organization as a large volume receiver. The address might or might not contain an organization name.
- P. The address identifies a Post Office Box or a delivery service.
- R. The address identifies a rural route. Address Verification uses the delivery record in the reference data to identify the address type.
- S. The address identifies a street. S is the default address type. If Address Verification cannot determine the address type from the address data, it returns the default value.
- U. Unidentified. The address is not valid, and Address Verification does not assign an address type.

If an address meets the criteria for more than one address type, Address Verification assigns the first applicable address type from the following list:

F, L, P, B, R, S, G.

## Address Type Indicators for France Addresses

Informatica Address Verification returns the following address type values for France addresses:

- B. The address identifies a building.
- F. The address identifies an organization. The address does not include a CEDEX post code.
- G. The address is a general delivery address. The reference data does not contain a match for the street information. However, the reference data contains a match for the CEDEX postal code in the address.
- L. The postal code identifies the organization as a large volume receiver. The address might or might not contain an organization name. The reference data uses the CEDEX postal code to add or validate the organization name.
- P. The address identifies a Post Office Box or a delivery service.
- S. The address identifies a street address. S is the default address type. If Address Verification returns S as the default value if Address Verification cannot determine the address type from the address data.
- U. Unidentified. The address is not valid, and Address Verification does not assign an address type.

If an address meets the criteria for more than one address type, Address Verification assigns the first applicable address type from the following list:

L, F, P, B, S, G.

## Address Type Indicators for New Zealand Addresses

Informatica Address Verification returns the following address type values for New Zealand addresses:

- B. The address identifies a building.
- F. The address identifies an organization.
- L. The address post code identifies the organization as a large volume receiver. The reference data adds or validates the organization name.

Address Verification can determine that the address is a large volume receiver in one of the following ways:

- The address post code identifies the organization as a large volume receiver.
- The reference data does not contain street or building information.
- P. The address identifies a Post Office Box or a delivery service.
- R. The address identifies a rural route. Address Verification uses the delivery record in the reference data to identify the address type.
- S. The address identifies a street. S is the default address type. If Address Verification cannot determine the address type from the address data, it returns the default value.
- U. Unidentified. The address is not valid, and Address Verification does not assign an address type.

If an address meets the criteria for more than one address type, Address Verification assigns the first applicable address type from the following list:

L, F, P, B, R, S

## Address Type Indicators for Addresses from the Rest of the World

Informatica Address Verification returns the following address type values for addresses from countries other than Australia, Canada, France, New Zealand, and the United States:

- B. The address identifies a building.

- F. The address identifies an organization.
- L. The address post code identifies the organization as a large volume receiver. The reference data adds or validates the organization name.  
Address Verification can determine that the address is a large volume receiver in one of the following ways:
  - The address post code identifies the organization as a large volume receiver.
  - The reference data does not contain street or building information.
- P. The address identifies a Post Office Box or a delivery service.
- S. The address identifies a street. S is the default address type. If Address Verification cannot determine the address type from the address data, it returns the default value.
- U. Unidentified. The address is not valid, and Address Verification does not assign an address type.

If an address meets the criteria for more than one address type, Address Verification assigns the first applicable address type from the following list:

L, F, P, B, S

**Note:** Address Verification can return information relevant to the address type on other output elements. Consult the Process Status, Element Input Status, and Element Result Status values.

## Certified Mode Process Status Values

When you validate addresses in the certified mode, Informatica Address Verification returns process status values that are specific for the certified mode you selected. The certified mode status values help you understand and analyze the results you receive in the certified mode.

### AMAS Status Codes

When you validate Australia addresses in the certified mode, Informatica Address Verification returns the following status codes along with the address output.

#### **EAM0**

AMAS output is not available for the input address.

#### **EAM1**

AMAS output for the input address is provided. The address is corrected or validated and DPID for the address is provided.

#### **EMA2**

AMAS output is not provided. The input address could not be corrected or validated, and so no DPID is provided.

### CASS Status Values

When you verify United States addresses in the certified mode, Informatica Address Verification returns the following status values to help you interpret and analyze the verification results.

#### **ECA0**

CASS output is not available for the input address.

**ECA1**

CASS output is not complete for the input address.

**ECA2**

This code is reserved for future use.

**ECA5**

CASS output provided for the input address.

## SendRight Status Codes

When you validate New Zealand addresses in the certified mode, Informatica Address Verification returns the following status values along with the address output.

**ESR0**

SendRight output is not available for the input address.

**ESR1**

SendRight output is provided for the input address.

## SERP Status Codes

When you validate Canada addresses in the certified mode, Informatica Address Verification returns the following status codes to help you interpret and analyze the SERP results.

**ESE0**

SERP output is not available for the input address.

**ESE1**

SERP output is provided.

If you set the process mode to certified and the SERP enrichment to on, Address Verification provides two enrichments: Category and Excluded Flag.

## SNA Status Codes

When you validate France addresses in the certified mode, Informatica Address Verification returns the following status values along with the address output.

**ESN0**

SNA output not available for the input address.

**ESN1**

SNA output for the input address is provided.

# API Return Codes

Informatica Address Verification returns numeric codes to indicate the status of API function calls. All API function calls to Address Verification receive an AD\_I32 (32 bit signed integer) return code value.

- A value of 0 (zero) indicates success.
- A negative value of -10000 or below indicates a very critical error, and further processing is usually impossible. Immediately shut down the whole process as it might be in an instable state.
- Negative values between -1 and -9999 indicate critical errors, and further processing might be impossible.
- A positive value of 1000 and above indicates non-critical errors, and further processing is possible. Return code values between 1 and 999 indicate warnings including possible issues with configuration settings, address input, or output.

The return value must always be checked for by the calling logic. While the return values inform about fundamental errors, the actual verification results are returned through separate API functions.

## Success

On successful completion of an API call, Informatica Address Verification returns a return code of 0 (zero).

## Warnings

Informatica Address Verification returns one of the following return codes when an API call receives an unexpected result:

| Code | Description  |
|------|--|
| 1    | The SetConfig.xml contains at least one corrupt unlock code.   |
| 2    | The SetConfig.xml contains at least one expired or not yet valid unlock code.  |
| 3    | The SetConfig.xml lists at least one database file which was not found.  |
| 4    | The SetConfig.xml lists at least one corrupt database file.  |
| 5    | The SetConfig.xml lists at least one database with a not supported version.  |
| 6    | The SetConfig.xml lists at least one database which is not supported.  |
| 7    | No valid unlock code for a database file.  |
| 8    | The SetConfig.xml lists at least one database more than once.  |
| 9    | The MaxMemoryUsageMB setting in SetConfig.xml is too small to fulfill all preloading settings and/or the CacheSize setting.    |
| 10   | The environmental settings (for example, Operating System) in at least one Unlock Code is incompatible to the current machine. |
| 11   | The SetConfig.xml contains at least one none supported type of unlock code.  |
| 100  | An input element or line which already had content was overwritten.  |

| Code | Description   |
|------|---|
| 101  | The AddressComplete input has too many lines. Extra lines to be ignored for further processing.   |
| 102  | At least one character sequence of a string is not valid. For example, it contains control codes or does violate some constraint. These sequences are replaced by spaces. |
| 200  | The output buffer is too small. The output was written, but truncated.  |
| 201  | At least one character of the output could not be encoded in the chosen encoding. Such characters are replaced by an underscore ('_').                                    |
| 300  | The engine usage period has expired or is not activated yet.  |
| 301  | The unlock code for a database file has expired or is not activated yet.  |
| 400  | Address lines and/or Address Complete was given in input. This part of the input was ignored.   |
| 401  | More than 19 lines were given through FormattedAddressLines or AddressComplete as input. The lines beyond 11 were ignored.  |
| 500  | The MaxResultCount in Parameters.xml was larger than the value in SetConfig.xml, and was reduced to the value in SetConfig.xml.   |
| 900  | No database was found, probably because the path was wrong.   |
| 901  | No database was opened, probably because the path was wrong and/or no valid unlock code was given.  |
| 902  | Error while attempting to open at least one of the extra CASS databases.  |

## Errors

Informatica Address Verification returns the following return codes if an API call failed to run because of an error condition:

| Code | Description   |
|------|---|
| 1000 | A pointer parameter was NULL.   |
| 1001 | A function parameter was 0.   |
| 1002 | A NULL pointer to an object was used (not relevant for C API).                                |
| 1003 | Two XMLs, one as a string and the other in a file, were entered though only one was required. |
| 1004 | The output buffer size is not valid.  |
| 1005 | Buffer misalignment; an AD_WCHAR* points to an odd address.                                   |
| 1100 | A parameter is out of range or is not valid.  |
| 1101 | An XML string is not valid.   |

| Code | Description   |
|------|---|
| 1200 | The character sequence of a string is not valid. For example, the string contains control codes or violates some constraint.                            |
| 1201 | The encoding parameter did not match the character size of the API call; for example: UCS2 (16 bit) vs. char (8 bit)                                    |
| 1300 | No <code>SetConfig.xml</code> was entered as parameter for <code>AD_Initialize</code> .   |
| 1301 | The engine has already been initialized.  |
| 1302 | <code>AD_DeInitialize()</code> failed because not all <code>AddressObject</code> s have been released.  |
| 1400 | No <code>AddressObject</code> is available (all <code>AddressObject</code> handles have already been obtained via <code>AD_GetAddressObject()</code> ). |
| 1401 | The passed <code>AddressObject</code> handle is not valid.  |
| 1500 | A database file has not been found.   |
| 1501 | A database file is invalid/corrupt.   |
| 1502 | No valid unlock code for a database file.   |
| 1503 | A database file has a non-supported version.  |
| 1600 | A feature has not been unlocked.  |
| 1700 | The country could not be identified or is not supported.  |
| 1701 | The country is not supported for this processing mode and type of input   |
| 1800 | Results are available, for this reason no <code>AddressObject</code> modification is allowed.   |
| 1801 | XML and direct API calls were used intermixed when setting the input data of an <code>AddressObject</code> .  |
| 1802 | <code>AD_Process()</code> has not been called successfully, no result is available.   |
| 1803 | The attempted operation was invalid, i.e. trying to set incompatible address elements.  |
| 1900 | The result index parameter is out of range (must be $\geq 1$ ).   |
| 1901 | The output buffer is too small to hold the result, no output was written.   |



## Critical Errors

If any critical error occurs, Informatica Address Verification returns one of the following return codes:

| Code  | Description  |
|-------|--|
| -1300 | The engine has not yet been initialized, need to call <code>AD_Initialize()</code> . |
| -1600 | No valid unlock code was given   |
| -1601 | The engine usage period has expired or is not activated yet                          |
| -1602 | A clock inconsistency has been detected  |
| -9900 | A memory allocation request failed   |
| -9901 | A file operation failed  |

Address Verification does not allow any function calls other than `AD_Initialize()` and `AD_DeInitialize()` after it returns a critical error code.

## Very Critical Errors

When a very critical error occurs, Informatica Address Verification does not allow any API calls other than `AD_DeInitialize()`. The following return calls indicate very critical errors:

| Code   | Description  |
|--------|--|
| -10000 | An unknown exception has occurred; this event should never occur |
| -10001 | An internal assertion has failed; this event should never occur  |
| -10002 | An internal error has occurred; this event should never occur    |

## CHAPTER 9

# Frequently Asked Questions

This chapter includes the following topics:

- [How Do I Manage Unlock Codes?, 130](#)
- [How Do I Initialize Informatica Address Verification?, 131](#)
- [How Do I Produce Valid Address Verification XML?, 132](#)
- [How Do I Manage Reference Address Databases?, 132](#)
- [How Do I Assign an Address to an AddressObject, 133](#)
- [How Do I Validate an Address?, 134](#)
- [How Do I Check the Process Mode?, 135](#)
- [How Do I Retrieve a Suggested Correction?, 135](#)
- [How Do I Retrieve the Result Status and Additional Information?, 136](#)
- [How Do I Use the Quick Address Entry Feature of Informatica Address Verification?, 136](#)
- [How Do I Use Informatica Address Verification with Multiple Processor Cores?, 137](#)
- [How Do I Use Address Verification for Master Data Management?, 137](#)
- [How Do I Use Informatica Address Verification in an eBusiness Environment?, 137](#)
- [How Do I Specify Processing Parameters and Input and Result Formats?, 138](#)
- [How Do I Find the Current Post Code for a South Korea Address?, 140](#)
- [How Do I Retrieve the Current Informatica Address Verification Settings?, 140](#)
- [How Do I Determine the Informatica Address Verification Version?, 140](#)

## How Do I Manage Unlock Codes?

You need valid unlock codes to access reference address databases for address validation and address enrichments. Each unlock code can unlock multiple databases that Informatica Address Verification assigns to the unlock code. Address Verification passes the unlock code to the initialization function. The initialization fails if you do not specify any unlock code or if the unlock code you specified is not valid or incorrect.

You can specify multiple unlock codes and pass them one after the other to the initialization function. If there are multiple unlock codes for one country among the unlock codes you specified, Address Verification uses the longest-valid unlock code for that country. For example, you have the following unlock codes:

Code A unlocks DEU and USA validation until 31.12.2014

Code B unlocks CHE and USA validation until 31.12.2015

Code C unlocks CHE and USA geocoding until 31.12.2015

In this case, DEU validation expires on 31.12.2014. For CHE and USA, the validation remains active until 31.12.2015. Note that unlock codes also have valid-from date before which the code remains not valid.

The following code sample shows assigning of unlock codes using C API.

```
AD_Initialize(
    "<?xml version='1.0' encoding='iso-8859-1' ?>\n"
    "<SetConfig>\n"
    "<General /\>\n"
    "<UnlockCode>(Enter Code A here)</UnlockCode>\n"
    "<UnlockCode>(Enter Code B here)</UnlockCode>\n"
    "<UnlockCode>(Enter Code C here)</UnlockCode>\n"
    "<DataBase CountryISO3='USA' Type='GEOCODING' Path='/ADDB' PreloadingType='NONE' /\>\n"
    "<DataBase CountryISO3='CHE' Type='GEOCODING' Path='/ADDB' PreloadingType='NONE' /\>\n"
    "<DataBase CountryISO3='ALL' Type='BATCH_INTERACTIVE' Path='/ADDB'
    PreloadingType='NONE' /\>\n" "</SetConfig>\n",
    NULL,
    NULL,
    NULL
);
```

The following code sample shows assigning of unlock codes using Java API.

```
AddressDoctor.initialize(
    "<?xml version='1.0' encoding='UTF-16LE'?>\n" +
    "<!DOCTYPE SetConfig SYSTEM 'SetConfig.dtd'>\n" +
    "<SetConfig>\n" +
    "<General WriteXMLEncoding='UTF-16LE'/>\n" +
    "// Engine & DB Unlock Code
    "<UnlockCode>(Enter Code A here)</UnlockCode>\n" +
    "<UnlockCode>(Enter Code B here)</UnlockCode>\n" +
    "<UnlockCode>(Enter Code C here)</UnlockCode>\n" +
    "<DataBase CountryISO3='USA' Type='GEOCODING' Path='/ADDB' PreloadingType='NONE'/>\n" +
    "<DataBase CountryISO3='CHE' Type='GEOCODING' Path='/ADDB' PreloadingType='NONE'/>\n" +
    "<DataBase CountryISO3='ALL' Type='BATCH_INTERACTIVE' Path='/ADDB'
    PreloadingType='NONE'/>\n" +
    "</SetConfig>",
    null, null, null);
```

## How Do I Initialize Informatica Address Verification?

If you are using a C language APIs, you must call `AD_Initialize()` or `AD_InitializeW()` functions to initialize Informatica Address Verification. If you are using Java APIs, call the `AddressDoctor.initialize` function. The initialization function evaluates the settings and configures the engine accordingly.

You can call other functions such as `AD_GetAddressObject()` only after the initialization function is successful. If the initialization failed, Address Verification returns an error code of -1300.

The following example shows a sample code for C API.

```
AD_Initialize(
    "<?xml version='1.0' encoding='iso-8859-1' ?>\n"
    "<!DOCTYPE SetConfig SYSTEM 'SetConfig.dtd'>\n"
    "<SetConfig>\n"
    "<General /\>\n"
    "<UnlockCode>(Enter Code here)</UnlockCode>\n"
    "<DataBase CountryISO3='ALL' Type='BATCH_INTERACTIVE' Path='/ADDB'
    PreloadingType='NONE'/>\n"
    "</SetConfig>\n",
    NULL,
    NULL,
    NULL
);
```

Alternatively, you can store the `SetConfig.xml` in an external file. In such cases, you can use the following initialization call.

```
AD_Initialize( NULL,
    "SetConfig.xml",
    NULL,
    NULL
);
```

The following example shows a sample code Java API.

```
// Initialize the Engine using the 'Direct' API
AddressDoctor.initialize(
    "<?xml version='1.0' encoding='UTF-16LE'?>\n" +
    "<!DOCTYPE SetConfig SYSTEM 'SetConfig.dtd'>\n" +
    "<SetConfig>\n" +
    "<General WriteXMLEncoding='UTF-16LE' />\n" +
    "<UnlockCode>(Enter Code here)</UnlockCode>\n" +
    "<DataBase CountryISO3='ALL' Type='BATCH_INTERACTIVE' Path='/ADDB'
PreloadingType='NONE'/>\n" +
    "</SetConfig>",
    null,
    null,
    null
);
```

Alternatively, you can store the `SetConfig.xml` in an external file. In such cases, you can use the following initialization call.

```
// Initialize the Engine using the 'XML' API
AddressDoctor.initialize(
    null,
    "SetConfig.xml",
    null,
    null
);
```

## How Do I Produce Valid Address Verification XML?

Ensure that any XML input to Informatica Address Verification is well-formed and validated against the DTDs provided by Informatica. Note that the sequence of elements in Address Verification XML files is important and you need to validate the structure against the corresponding DTD. However, it is not important that you maintain the sequence for element attributes.

If you are working with XML files created on different platforms, note that the End-of-Line (EoL) character differs between Windows (CRLF) and Unix (LF). For more information, see the Wikipedia page on Linebreak at <http://wikipedia.org/wiki/Linebreak>.

You can find more information about XML from multiple sources on the Web. <http://wikipedia.org/wiki/XML> contains an introduction to XML. XML functionality is available in most of the Integrated Development Environments (IDEs). You can also find a wide variety of free XML editors with XML validation.

## How Do I Manage Reference Address Databases?

Along with the unlock codes, you must also specify the details of reference address databases including the country name, database type, and file path in `SetConfig.xml`. You can use a virtual ISO code, ALL, to denote

all databases that you have access to. However, you still need to specify details of the databases including the file path in `SetConfig.xml`.

The following snippet from `SetConfig.xml` provides an example of database settings.

```
<SetConfig>
<General WriteXMLEncoding="UTF-16" MaxMemoryUsageMB="2048" MaxAddressObjectCount="10"
MaxThreadCount="2">
</General>
<UnlockCode>Address Validation Unlock Code</UnlockCode>
<UnlockCode>Geocoding Unlock Code</UnlockCode>
<DataBase CountryISO3="USA" Type="CERTIFIED" Path="C:/AddressDoctor/DB/CASS"
PreloadingType="FULL"></DataBase>
<DataBase CountryISO3="CAN" Type="CERTIFIED" Path="C:/AddressDoctor/DB/SERP"
PreloadingType="FULL"></DataBase>
<DataBase CountryISO3="ALL" Type="BATCH_INTERACTIVE" Path="C:/AddressDoctor/DB"
PreloadingType="NONE"></DataBase>
<DataBase CountryISO3="USA" Type="SUPPLEMENTARY" Path="C:/AddressDoctor/DB/Enrichment"
PreloadingType="PARTIAL"></DataBase>
<DataBase CountryISO3="GBR" Type="SUPPLEMENTARY" Path="C:/AddressDoctor/DB/Enrichment"
PreloadingType="PARTIAL"></DataBase>
<DataBase CountryISO3="ALL" Type="GEOCODING" Path="C:/AddressDoctor/DB/Geocoding"
PreloadingType="NONE"></DataBase>
<DataBase CountryISO3="ALL" Type="CAMEO" Path="C:/AddressDoctor/DB/CAMEO"
PreloadingType="NONE"></DataBase>
<DataBase CountryISO3="ALL" Type="FASTCOMPLETION" Path="C:/AddressDoctor/DB"
PreloadingType="NONE"></DataBase>
</SetConfig>
```

The order in which you list the reference database file names in `SetConfig.xml` influences the loading of the database data into memory and can affect the operation of the Address Verification engine. For more information, see ["Reference Database Order in SetConfig.xml" on page 24](#).

In case of conflicting database settings, Informatica Address Verification gives precedence to the first occurrence of that element. When you enter database details in `SetConfig.xml`, enter the Database element with the CountryISO3 attribute set to ALL at the end of the list. If you enter the Database element with the CountryISO3 attribute set to ALL before other Database elements, Address Verification overrides all other database settings with the settings for the database element with the country ISO code set to ALL.

## How Do I Assign an Address to an AddressObject

You must choose an appropriate address format before you assign an address to an `AddressObject`. The quality of result you receive depends on the address format that you use to assign the address to the `AddressObject`.

The address formats can be broadly classified into three: fielded data, partially-fielded data, and unfielded data. Informatica Address Verification returns the most accurate results for fielded data and the least accurate for unfielded data.

The following examples show the API calls for assigning a fielded address to the `AddressObject`.

**Note:** Before you assign an address to the `AddressObject`, call the `AD_ClearData()` function to clear any data that is present.

### Assigning a fielded address using the C direct API

```
AD_SetInputAddressElement( hAOHandle, "Country", 1, NULL, "Canada" );
AD_SetInputAddressElement( hAOHandle, "PostalCode", 1, NULL, "G1R 3X2" );
AD_SetInputAddressElement( hAOHandle, "Locality", 1, NULL, "Toronto" );
AD_SetInputAddressElement( hAOHandle, "DeliveryService", 1, NULL, "PO Box 1827" );
```

### Assigning a fielded address using the Java direct API

```
m_oAO.setInputAddressElement("Country", 1, null, "Canada");
m_oAO.setInputAddressElement("PostalCode", 1, null, "G1R 3X2");
m_oAO.setInputAddressElement("Locality", 1, null, "Toronto");
m_oAO.setInputAddressElement("DeliveryService", 1, null, "PO Box 1827");
```

### Assigning a fielded address using the C XML API

```
AD_SetInputDataXML( hAOHandle,
    "<?xml version='1.0' encoding='ISO-8859-1'?>\n"
    "<!DOCTYPE InputData SYSTEM 'InputData.dtd'\n"
    "<InputData>\n"
    "<AddressElements>\n"
    "<Country Item='1' Type='NAME'>SGP</Country>\n"
    "<Locality Item='1' Type='COMPLETE'>Singapore</Locality>\n"
    "<PostalCode Item='1' Type='FORMATTED'>048624</PostalCode>\n"
    "<Street Item='1' Type='COMPLETE'>Raffles Place</Street>\n"
    "<Number Item='1' Type='COMPLETE'>80</Number>\n"
    "<Building Item='1' Type='COMPLETE'>#50-01 UOB Plaza 1</Building>\n"
    "<Organization Item='1' Type='NAME'>AddressDoctor GmbH</Organization>\n"
    "</AddressElements>\n"
    "</InputData>\n"
);
```

### Assigning a fielded address using the Java XML API

```
m_oAO.setInputDataXML(
    "<?xml version='1.0' encoding='UTF-16'?>"+
    "<!DOCTYPE InputData SYSTEM InputData.dtd>"+
    "<InputData>"+
    "<AddressElements>"+
    "    <Key>4711</Key>"+
    "    <Country Item='1' Type='NAME'>SGP</Country>"+
    "    <Locality Item='1' Type='COMPLETE'>Singapore</Locality>"+
    "    <PostalCode Item='1' Type='FORMATTED'>048624</PostalCode>"+
    "    <Street Item='1' Type='COMPLETE'>Raffles Place</Street>"+
    "    <Number Item='1' Type='COMPLETE'>80</Number>"+
    "    <Building Item='1' Type='COMPLETE'>#50-01 UOB Plaza 1</Building>"+
    "    <Organization Item='1' Type='NAME'>AddressDoctor GmbH</Organization>"+
    "</AddressElements>"+
    "</InputData>");
```

## How Do I Validate an Address?

If you are using C language API, you can validate an address by calling `AD_Process()`. If you are using Java API, call `AddressDoctor.process()` to validate addresses.

Before you can validate an address, you must create an `AddressObject` and add an input address to the object. If you do not specify a process mode, Informatica Address Verification validates addresses in the Batch mode.

The following example shows the function call for C API.

```
AD_Process( hAOHandle );
```

The following example shows the function call for Java API.

```
AddressDoctor.process(m_oAO);
```

## How Do I Check the Process Mode?

You can check the ModeUsed attribute of the Result element to verify the process mode Informatica Address Verification uses to process addresses.

The following example shows a sample code for C API.

```
char sResultParameters[32];
AD_GetResultParameter(hAOHandle, "ModeUsed", sResultParameters, sizeof(sResultParameter));
```

The following example shows a sample code for Java API.

```
System.out.println(m_oAO.getResultParameter("ModeUsed"));
```

## How Do I Retrieve a Suggested Correction?

You can retrieve a suggested correction by calling the AD\_GetResultAddressElement() in the C direct API or m\_oAO.getResultAddressElement() in Java direct API.

You must first run the AD\_Process() function or its equivalent in Java to process an address.

### Retrieving an address suggestion through the C direct API

```
AD_U32 ulNumResults;
size_t stCurResult;
AD_GetResultCount( hAOHandle, &ulNumResults );
for( stCurResult = 1; stCurResult <= ulNumResults; stCurResult++ )
{
    char sStreet[ 256 ];
    AD_U32 ulNumItems;
    size_t stCurItem;
    AD_GetResultAddressElementItemCount( hAOHandle, 1, "Street", &ulNumItems );
    for( stCurItem = 1; stCurItem <= ulNumItems; stCurItem++ )
    {
        AD_GetResultAddressElement( hAOHandle, stCurResult, "Street", stCurItem, "COMPLETE",
        sStreet, sizeof( sStreet ) );
        printf( "Result %u: Street item %u: %s\n", stCurResult, stCurItem, sStreet );
    }
}
```

### Retrieving an address suggestion through the Java direct API

```
int NumResults = m_oAO.getResultCount();
int CurResult;
for (CurResult = 1; CurResult <= NumResults; CurResult++) {
    int NumItems = m_oAO.getResultAddressElementItemCount(CurResult, "Street");
    int CurItem;
    for (CurItem = 1; CurItem <= NumItems; CurItem++) {
        System.out.println(m_oAO.getResultAddressElement(CurResult, "Street", CurItem,
        "COMPLETE"));
    }
}
```

### Retrieving an address suggestion through the C XML API

```
char sResultXML[ 16 * 1024 ];
AD_GetResultXML( hAOHandle, sResultXML, sizeof( sResultXML ) );
```

### Retrieving an address suggestion through the Java XML API

```
String sResultXML = "";
sResultXML = m_oAO.getResultXML();
```

## How Do I Retrieve the Result Status and Additional Information?

If you are using the C direct API, you can call the `AD_GetResultParameter()` function with the `ProcessStatus` value to retrieve the result status. If you are using the Java direct API, you can call the `m_oAO.getResultParameter("ProcessStatus")` function to retrieve the result status.

### Retrieving the result status through the C direct API

```
char sResultParameters[32];
AD_GetResultParameter(hAOHandle, "ProcessStatus", sResultParameters,
sizeof(sResultParameter));
```

To receive specific status of any address element result, call `AD_GetResultDataParameter()` with the required parameters. To receive details of address enrichments, call `AD_GetResultEnrichmentDataParameter()`.

### Retrieving the result status through the Java direct API

```
System.out.println(m_oAO.getResultParameter("ProcessStatus"));
```

In C XML API, the `AD_GetResultXML()` function returns the entire result information without you specifying any parameter. However, some of the settings for the `Result` element in `Parameters.xml` might influence the details in the result output.

## How Do I Use the Quick Address Entry Feature of Informatica Address Verification?

You can use the fast completion process mode of Informatica Address Verification to receive suggestions as you enter an address. You can use the fast completion mode to implement address quality checks and improvements at Point-of-Sales (PoS) systems, Customer Relationship Systems (CRM), Property and Reservation Management (PMS) systems and such.

In the fast completion mode, you can assign incomplete input addresses to an `AddressObject` and receive address suggestions based on partial inputs. Address Verification corrects the input to the extent possible and returns a list of suggestion from which you can choose an address.

When you design an application that uses the Address Verification fast completion mode, you can design the system to call the `AD_Process()` function after each keystroke. For best results, keep the reference databases are easily accessible so that the pick lists appear in real time.

When you enter an address in the fast completion mode, Address Verification returns the first 100 suggestions and indicates an overflow if there are more matches. If there is an overflow indicator, call the `AD_Process()` function again with additional input.



# How Do I Use Informatica Address Verification with Multiple Processor Cores?

On a device that has  $n$  number of processor cores, you can configure Informatica Address Verification to have as many as  $n-1$  address processing threads.

For example, if the device has four cores, you can configure Address Verification with a `MaxThreadCount` of 3 and `MaxObjectCount` of 3.

When you call the initialization function with the maximum thread count of 3, the main thread of the device creates three worker threads and assigns that to Address Verification. Each worker thread then acquires one `AddressObject` handle through `AD_GetAddressObject( &hAOHandle )`; and subsequently keeps repeating the following sequence:

```
AD_SetInputDataXML( hAOHandle, <XML string> );
AD_Process( hAOHandle );
AD_GetResultXML( hAOHandle, sResultXML, sizeof( sResultXML ) );
AD_ClearData( hAOHandle );
```

When you shut down Address Verification, the main thread destroys all worker threads and deinitialize Address Verification through the following function calls:

```
AD_ReleaseAllAddressObjects();
AD_DeInitialize();
```

# How Do I Use Address Verification for Master Data Management?

You can use Informatica Address Verification in batch mode to validate bulk addresses as part of Master Data Management (MDM) or other data integration systems. Batch mode in Address Verification is designed for mass data address quality and enables you to assign addresses to `AddressObject` irrespective of the quality of input address. When you process addresses in batch mode, Address Verification automatically corrects the addresses and returns the single most-likely candidate in the result.

When you design an application for batch processing, call the `AD_Process()` function with the `Mode` attribute of the `Process` element set to `BATCH`. Wherever possible, Address Verification returns a single corrected address with a process code of `Cx` or `Vx`.

To tackle severe address quality challenges, you can also use the `OptimizationLevel` attribute of the `Process` element.

# How Do I Use Informatica Address Verification in an eBusiness Environment?

You can use the interactive processing mode of Informatica Address Verification when you want to integrate Address Verification in an eBusiness environment. You can use the interactive mode of Address Verification

to ensure address quality at point-of-data entry. You can use this mode for online input forms, web-based transactions, and customer-feedback systems.

When you design an application for interactive entry, call the `AD_Process()` function with the `Mode` attribute of the `Process` element set to `INTERACTIVE`.

In the interactive mode, you can assign addresses to an `AddressObject` irrespective of the data quality. Address Verification corrects the input address to the extent possible and returns a list of suggestions from which you can select the most relevant address. If Address Verification identifies one definite match for the input address, Address Verification returns only that address to you.

When you design your system to use the interactive mode, you can design your system to allow users to edit the selected, validated result to make any necessary change before they submit the address to your system.

## How Do I Specify Processing Parameters and Input and Result Formats?

You can specify the processing parameters and input and results formats either at the global level or at address object level. If you are using C API, you can use `AD_SetParametersXML()` or `AD_SetParametersXMLW()` to set parameters at the address-object level. In Java API, use the `m_oAO.setParametersXML` function, where `m_oAO` denotes the address object to which you want to apply the parameters.

The following examples shows C and Java API examples for assigning process parameters and input and results formats at address object level and global level.

### Specifying Parameters at the AddressObject Level Using C API

```
AD_SetParametersXML(hAOHandle,
    "<?xml version='1.0' encoding='iso-8859-1' ?>\n"
    "<!DOCTYPE Parameters SYSTEM 'Parameters.dtd'>\n"
    "<Parameters>\n"
    "<Process Mode='BATCH' />\n"
    "<AddressElementStandardize>\n"
    "<Country Casing='UPPER' />\n"
    "</AddressElementStandardize>\n"
    "</Parameters>\n",
    NULL,
    NULL
);
```

Alternatively, you can call parameters from an external file as shown in the following example.

```
AD_SetParametersXML(hAOHandle,
    NULL,
    NULL,
    "Parameters.xml"
);
```

### Specifying Parameters at the AddressObject Level Using Java API

```
// This code assumes that you have already acquired m_oAO as the active AddressObject
m_oAO.setParametersXML(
    "<?xml version='1.0' encoding='UTF-16LE' ?>\n" +
    "<Parameters>\n" +
    "<Process Mode='BATCH'/>\n" +
    // Java uses UTF-16LE as default encoding for its String method
    "<Input Encoding='UTF-16LE'/>" +
    "<Result Encoding='UTF-16LE'/>" +
    "<AddressElementStandardize> \n" +
    "<Country Casing='UPPER' />\n" +
    "
```

```

    "</AddressElementStandardize> \n" +
    "</Paramaters>",
    null
);

```

Alternatively, you can call parameters from an external file as shown in the following example.

```

// This code assumes that you have already acquired m_oAO as the active AddressObject
m_oAO.setParametersXML(
    null,
    "Parameters.xml"
);

```

Instead of setting attributes for each AddressObject individually, you can specify the parameters at the global level by assigning the values to the initialization function. Assigning parameters at the global level applies global defaults to all AddressObjects that do not have individual parameters set as shown in the preceding examples. The following examples show the global level parameter settings.

### Specifying Parameters at the Global Level Using C API

```

AD_Initialize(
    "<?xml version='1.0' encoding='iso-8859-1' ?>\n"
    "<!DOCTYPE SetConfig SYSTEM 'SetConfig.dtd'\n"
    "<SetConfig>\n"
    "<General /\n"
    "<UnlockCode>(Enter Code here)</UnlockCode>\n"
    "<DataBase CountryISO3='ALL' Type='BATCH_INTERACTIVE' Path='/ADDB'
    PreloadingType='NONE' /\n"
    "</SetConfig>\n",
    NULL,
    "<?xml version='1.0' encoding='iso-8859-1' ?>\n"
    "<!DOCTYPE Parameters SYSTEM 'Parameters.dtd'\n"
    "<Parameters>\n"
    "<Process Mode='BATCH' /\n"
    "<AddressElementStandardize>\n"
    "<Country Casing='UPPER' /\n"
    "</AddressElementStandardize>\n"
    "</Parameters>\n",
    NULL
);

```

Alternatively, you can call the parameters from an external file as shown in the following example:

```

AD_Initialize(
    NULL,
    "SetConfig.xml",
    NULL,
    "Parameters.xml"
);

```

### Specifying Parameters at the Global Level Using Java API

```

AddressDoctor.initialize(
    "<?xml version='1.0' encoding='UTF-16' ?>" +
    "<!DOCTYPE SetConfig SYSTEM 'SetConfig.dtd'" +
    "<SetConfig><General WriteXMLEncoding='UTF-16' /\n" +
    "    <UnlockCode>(Enter Code here)</UnlockCode>" +
    "    <DataBase CountryISO3='ALL' Type='BATCH_INTERACTIVE'" +
    "    Path='/ADDB' PreloadingType='NONE' /\n" +
    "</SetConfig>", null,
    "<?xml version='1.0' encoding='UTF-16' ?>" +
    "<!DOCTYPE SetConfig SYSTEM 'Parameters.dtd'" +
    "<Parameters WriteXMLEncoding='UTF-16'" +
    "    <Input Encoding='UTF-16' /\n" +
    "    <Result Encoding='UTF-16' /\n" +
    "</Parameters>", null);

```

## How Do I Find the Current Post Code for a South Korea Address?

You can use Informatica Address Verification to find the current, five-digit post code for a South Korea address. Korea Post changed the post code structure from six digits to five digits in 2015.

To obtain the current post code, first verify the address in batch or interactive mode. Address Verification returns an address ID value that you can use in address code lookup mode to retrieve the current post code. If you enter a lot-based address or a six-digit post code as the input, set the `MatchingExtendedArchive` attribute of the `Process` element to ON.

For more information on the steps to find the current, street-based address or the five-digit post code for a South Korea address, see [“Extended Archives and South Korea Addresses” on page 50](#).

## How Do I Retrieve the Current Informatica Address Verification Settings?

If you are using C language API, call `AD_GetConfigSettingsXML()` at the Informatica Address Verification global level to generate `GetConfig.xml` that contains the current configuration that `AD_Initialize()` or `AD_InitializeW()` specifies.

To retrieve the default set of parameters at the Address Verification global level, call `AD_GetParametersSettingsXML()` to generate `Parameters.XML` that contains the default set of parameters that `AD_Initialize()` or `AD_InitializeW()` specifies.

Alternatively, you can call `AD_GetParametersXML()` to retrieve the parameters that `AD_SetParametersXML()` or `AD_SetParametersXMLW()` set.

## How Do I Determine the Informatica Address Verification Version?

If you are using C APIs, you can call `AD_GetVersion()` to retrieve the zero-terminated Informatica Address Verification version string in the x.x.x.x format, for example, 5.0.0.251. You can call these functions even before you call `AD_Initialize()` to initialize Address Verification.