



Informatica® Address Verification (On-Premises)

6.4.0

# Migration Guide

© Copyright Informatica LLC 1993, 2023

This software and documentation are provided only under a separate license agreement containing restrictions on use and disclosure. 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.

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation is subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License.

Informatica, the Informatica logo, and any other Informatica-owned trademarks that appear in the document are trademarks or registered trademarks of Informatica LLC in the United States and many jurisdictions throughout the world. A current list of Informatica trademarks is available on the web at <https://www.informatica.com/trademarks.html>. 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 © 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](mailto: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/hsqldbLicense.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/license.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/asl/>; <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.

The information in this documentation is subject to change without notice. If you find any problems in this documentation, report them to us at [infa\\_documentation@informatica.com](mailto:infa_documentation@informatica.com).

Informatica products are warranted according to the terms and conditions of the agreements under which they are provided. INFORMATICA PROVIDES THE INFORMATION IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT.

Publication Date: 2023-08-29

# Table of Contents

|                                                                      |           |
|----------------------------------------------------------------------|-----------|
| <b>Preface .....</b>                                                 | <b>6</b>  |
| Informatica Resources. ....                                          | 6         |
| Informatica Network. ....                                            | 6         |
| Informatica Knowledge Base. ....                                     | 6         |
| Informatica Documentation. ....                                      | 6         |
| Informatica Product Availability Matrices. ....                      | 7         |
| Informatica Velocity. ....                                           | 7         |
| Informatica Marketplace. ....                                        | 7         |
| Informatica Global Customer Support. ....                            | 7         |
| <br><b>Chapter 1: Main Features.....</b>                             | <b>8</b>  |
| Introduction. ....                                                   | 8         |
| Main Features of Address Verification. ....                          | 8         |
| Feature Enhancements in Address Verification 6.x. ....               | 9         |
| Features in Development. ....                                        | 11        |
| Deprecated Features. ....                                            | 11        |
| <br><b>Chapter 2: Getting Started with Address Verification.....</b> | <b>13</b> |
| Getting Started with Address Verification Overview. ....             | 13        |
| Supported Platforms and System Requirements. ....                    | 13        |
| Install Package. ....                                                | 15        |
| JSON Files. ....                                                     | 15        |
| Reference Databases and Packages. ....                               | 16        |
| <br><b>Chapter 3: Changes in Nomenclature.....</b>                   | <b>18</b> |
| Changes in Nomenclature Overview. ....                               | 18        |
| Address Elements. ....                                               | 18        |
| CASS Attributes. ....                                                | 19        |
| Data Structure. ....                                                 | 19        |
| Enrichments. ....                                                    | 20        |
| Geocoding Status Values. ....                                        | 20        |
| Parameters. ....                                                     | 21        |
| Processing Modes. ....                                               | 21        |
| Status Codes. ....                                                   | 22        |
| <br><b>Chapter 4: JSON Schema Highlights.....</b>                    | <b>23</b> |
| Address Verification Job Schema Overview. ....                       | 23        |
| Arrays. ....                                                         | 23        |
| Preferred Script Properties. ....                                    | 24        |
| Preferred Language Properties. ....                                  | 25        |

Status Values. . . . . 26

# Preface

Refer to the Informatica Address Verification (On-Premises) Migration Guide to learn about the differences between Informatica Address Verification 5.x and 6.x. The guide uses Address Verification version 5.16.1 and version 6.1.0 as reference points for 5.x and 6.x behavior. The guide can help you to quickly replicate your Address Verification 5.x experience in the Address Verification 6.x environment.

## Informatica Resources

Informatica provides you with a range of product resources through the Informatica Network and other online portals. Use the resources to get the most from your Informatica products and solutions and to learn from other Informatica users and subject matter experts.

### Informatica Network

The Informatica Network is the gateway to many resources, including the Informatica Knowledge Base and Informatica Global Customer Support. To enter the Informatica Network, visit <https://network.informatica.com>.

As an Informatica Network member, you have the following options:

- Search the Knowledge Base for product resources.
- View product availability information.
- Create and review your support cases.
- Find your local Informatica User Group Network and collaborate with your peers.

### Informatica Knowledge Base

Use the Informatica Knowledge Base to find product resources such as how-to articles, best practices, video tutorials, and answers to frequently asked questions.

To search the Knowledge Base, visit <https://search.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

Use the Informatica Documentation Portal to explore an extensive library of documentation for current and recent product releases. To explore the Documentation Portal, visit <https://docs.informatica.com>.

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

## Informatica Product Availability Matrices

Product Availability Matrices (PAMs) indicate the versions of the operating systems, databases, and types of data sources and targets that a product release supports. You can browse the Informatica 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 and based on real-world experiences from hundreds of data management projects. Informatica Velocity represents the collective knowledge of Informatica consultants who work with organizations around the world to plan, develop, deploy, and maintain successful data management solutions.

You can find Informatica Velocity resources at <http://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 extend and enhance your Informatica implementations. Leverage any of the hundreds of solutions from Informatica developers and partners on the Marketplace to improve your productivity and speed up time to implementation on your projects. You can find the Informatica Marketplace at <https://marketplace.informatica.com>.

## Informatica Global Customer Support

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

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

<https://www.informatica.com/services-and-training/customer-success-services/contact-us.html>.

To find online support resources on the Informatica Network, visit <https://network.informatica.com> and select the eSupport option.

# CHAPTER 1

## Main Features

This chapter includes the following topics:

- [Introduction, 8](#)
- [Main Features of Address Verification, 8](#)
- [Feature Enhancements in Address Verification 6.x, 9](#)
- [Features in Development, 11](#)
- [Deprecated Features, 11](#)

## Introduction

The transition from Informatica Address Verification 5.x to Informatica Address Verification 6.x provides you with new and updated features that can improve your address verification operations. Use this guide to identify the 5.x features that are obsolete or deprecated in Address Verification 6.x and to identify the alternative features that you can use in 6.x.

**Note:** This guide uses Address Verification version 5.16.1 and version 6.1.0 as reference points for current 5.x and 6.x behavior.

## Main Features of Address Verification

Informatica Address Verification 6.x includes the following features:

### **Support for JSON**

In version 6.x, Address Verification defines address requests and responses in JSON and defines the engine configuration in JSON.

Address Verification 5.x used XML to configure the engine and to define address requests and responses.

### **RESTful interface design**

The interface is modelled according to REST principles, using JSON as the data format. Data is submitted to the engine via Put functions, retrieved from the engine via Get functions, and deleted via the Delete function. Data is processed via the Post function.



### Use of function servers

A function server verifies the address and writes the result back to the job data.

Each function server runs as a separate process, isolating the main process from any issues related to the address processing. If a function server fails, the event is reported back to the caller as an error and a new function server is started.

### Support for hot swapping of file sets

Address Verification supports hot swapping of file sets. Hot swapping allows you to switch from one file set to another in seconds without deinitializing the process.

### Support for UTF-8 and UTF-16

Address Verification 6.x supports UTF-8 and UTF-16 encodings exclusively for input and output. Address Verification sets the encoding to UTF-8 or UTF-16 based on your use of 8-bit or 16-bit API functions.

**Note:** When you use a function to submit or return a string value with UTF-16 characters, append W to the function name. For example, submit `IDVE_PutStringW()`. For functions that do not pass a string value, such as `IDVE_Delete()` and `IDVE_Deinitialize()`, use the function without W.

## Feature Enhancements in Address Verification 6.x

Informatica Address Verification 6.x includes the following feature enhancements:

### Bulk processing of addresses

In Address Verification 6.x, you can submit 1000 addresses in a single call in batch and certified modes.

In Address Verification 5.x, Address Verification read a single address per call in all processing modes.

### Use of arrays

In Address Verification 6.x, the Address Verification engine can operate on an array of inputs, which enables automatic multi-threaded processing of the inputs. Many of the elements you can configure in an address job accept an array of input data options and can return results in an array. In this manner, Address Verification can return more than one output value for a given element.

### Country-specific application of properties

In Address Verification 6.x, you can set address verification properties at the country level in addition to setting the properties at the element level and at the global level.

In Address Verification 5.x, you set properties at the element level and at the global level only.

### Country-specific specification of enrichments

In Address Verification 6.x, you can configure Address Verification to add enrichments globally to all addresses, and you can configure Address Verification to add enrichments on a country-specific basis.

In Address Verification 5.x, Address Verification returned enrichments on a country-specific basis only.

### Support for QuickCapture mode

In Address Verification 6.x, QuickCapture mode replaces fast completion mode. You can use QuickCapture mode to receive address suggestions as you type. In QuickCapture mode, you submit an address in a single line. You can receive up to 100 suggestions for an input address in QuickCapture mode.

### **Enhanced validation status code options**

In Address Verification 6.x, Address Verification overhauls the validation status codes to provide greater clarity.

### **Specification of status codes at the element level**

In Address Verification 6.x, you can configure Address Verification to return a MatchStatus, ResultStatus, PostalRelevance, and ExtendedResultStatus value for individual address elements.

### **Enhanced preferred language options**

In Address Verification 6.x, you can include a list of preferred languages when you submit addresses for verification. Address Verification searches the reference data and returns each address in the first available language based on the order that you specify in the input list.

You can also configure Address Verification to return each address in multiple languages.

In Address Verification 5.x, you specified a single address when you submitted addresses for verification. If the address was unavailable in the preferred language, Address Verification returned the address in the default language.

### **Enhanced preferred script options**

In Address Verification 6.x, Address Verification provides separate options to specify the output script, transliteration standard, and the depth of the Latin character sets for an address in the PreferredScript element.

In Address Verification 5.x, Address Verification preset the options when you configured the PreferredScript parameter.

### **Enhanced geocoding options**

In Address Verification 6.x, you can submit a list of preferred geocoding levels to Address Verification. Address Verification returns the specified geocoordinates based on the order of the geocoding levels that you specify in the input list.

You can also configure Address Verification to return more than one set of geocoordinates in a single request.

In Address Verification 5.x, Address Verification returned a single set of geocoordinates.

### **Enhanced country determination options**

In Address Verification 6.x, you can specify the input fields that you want Address Verification to consider when it searches for country information.

In Address Verification 5.x, Address Verification read a preset group of fields to determine the country. The country determination options in version 6.x replace the Force Country and Default Country options in version 5.x.

### **Enhanced casing options**

In Address Verification 6.x, you can configure Address Verification to preserve the casing style of an input address.

### **Enhanced verification level options**

In Address Verification 6.x, you can specify the desired level and the minimum level to which Address Verification verifies an address.

In Address Verification 5.x, you specified a single desired verification level on the MatchingScope parameter.

### Enhanced alias handling options

In Address Verification 6.x, you can configure Address Verification to return the official name or the alias for all address elements that support alias names.

In Address Verification 5.x, Address Verification provided options to return the official or alias name for street and locality elements only.

### Enhanced alternative handling options

In Address Verification 6.x, Address Verification combines the operations of the MatchingExtendedArchive and MatchingAlternatives parameters into the AlternativeHandling element.

### Reverse geocoding

In Address Verification 6.1.0, you can submit latitude and longitude coordinates as input in an address verification process. Address Verification returns the closest address or addresses to the coordinates that you submit, based on criteria that you define.

To facilitate reverse geocoding, Address Verification introduces the *GeocodeToAddress* process mode. GeocodeToAddress mode requires arrival point reference data and returns addresses with arrival point precision.

## Features in Development

You can verify addresses in Australia, Canada, France, New Zealand, and the United States in certified mode. The Address Verification 6.x engine applies the same certification processes to your address data as the Address Verification 5.16.2 engine. However, formal certification of the 6.x engine in Australia, New Zealand, and the United States is complete.

## Deprecated Features

The following table lists the deprecated features in Informatica Address Verification 6.x:

| Feature Type                         | Deprecated Feature                                                                                                                                                                                                                                                                                                                                          |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Address Elements                     | In 6.x, Address Verification deprecates the following sub-items of an address element: <ul style="list-style-type: none"><li>- The Unformatted sub-item of the PostalCode element.</li><li>- The AddInfo sub-item of the Number element.</li><li>- The CompleteWithSubbuilding sub-item of the Building element.</li></ul>                                  |
| Attributes                           | In 6.x, Address Verification deprecates the following attributes: <ul style="list-style-type: none"><li>- OptimizationLevel</li><li>- DualAddressPriority</li><li>- ElementAbbreviation</li></ul> The StandarzielInvalidAddresses attribute did not appear in Address Verification 6.0-6.2. Address Verification reintroduced the attribute in version 6.3. |
| Country-Specific Address Enrichments | In 6.x, Address Verification deprecates the following address enrichments of the United States: <ul style="list-style-type: none"><li>- CMSAID</li><li>- PMSAID</li></ul>                                                                                                                                                                                   |

| Feature Type                                            | Deprecated Feature                                                                                                                                                                                            |
|---------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Processing Modes                                        | In 6.x, Address Verification deprecates the Country Recognition and Parse processing modes.                                                                                                                   |
| Single-Line Verification in Batch and Interactive Modes | In 6.x, Address Verification uses single-line verification in QuickCapture mode only.                                                                                                                         |
| Status Codes                                            | In 6.x, Address Verification deprecates the following status codes: <ul style="list-style-type: none"> <li>- AddressResolutionCode</li> <li>- Exxx status codes for Enrichments and Certifications</li> </ul> |

## Replaced Features

The following table lists features in Address Verification 5.x that are replaced by similar features in Address Verification 6.x:

| Features in Address Verification 5.x                       | Features in Address Verification 6.x | Description                                                                                                                                                                                                                |
|------------------------------------------------------------|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MatchingExtendedArchive                                    | AlternativeHandling                  | In 6.x, you can use the OutdatedAddresses property of the AlternativeHandling element instead of the MatchingExtendedArchive parameter for handling archives.                                                              |
| KOR_ADDRESS_ID                                             | AlternativeHandling                  | In 6.x, you can use the GetNewAddress option of the OutdatedAddresses property under the AlternativeHandling element instead of the MatchingExtendedArchive parameter to retrieve newer versions of South Korea addresses. |
| RECORD_TYPE address enrichment for United States addresses | AddressType                          | In 6.x, you can use the AddressType property instead of the RECORD_TYPE address enrichment to identify the type of mailbox that an United States address points to.                                                        |

## CHAPTER 2

# Getting Started with Address Verification

This chapter includes the following topics:

- [Getting Started with Address Verification Overview, 13](#)
- [Supported Platforms and System Requirements, 13](#)
- [Install Package, 15](#)
- [JSON Files, 15](#)
- [Reference Databases and Packages, 16](#)

## Getting Started with Address Verification Overview

You can learn how to install Informatica Address Verification 6.x and determine the amount of memory required for installation. You can also specify the data files to use in an address verification job. Address Verification 6.x introduces the DatasetDetermination element, which allows you to choose from the available data files.

## Supported Platforms and System Requirements

Informatica Address Verification is supported on a number of hardware and software platforms. The system resource requirements for Address Verification vary according to your requirements.

### Supported Platforms

Address Verification is developed using the C/C++ programming language. Address Verification provides different software packages to suit the hardware and software environment in which you install Address Verification. The Address Verification software packages contain C/C++, Java, and Microsoft .NET-based APIs.

**Note:** You can model Address Verification implementations for other languages, such as C#, PHP, Perl, Ruby, and Python. Informatica provides technical support for the C/C++, Java, and .NET APIs. Informatica does not provide implementation-specific support.

If you call the Address Verification engine through Java, install a Java Development Kit on the machine that hosts the Address Verification engine.

You can install Address Verification on machines with the following configurations:

| Operating System                | Processor Architecture |
|---------------------------------|------------------------|
| Windows Server 2019             | x64 (64-bit)           |
| Windows Server 2016             | x64 (64-bit)           |
| SUSE Linux Enterprise Server 15 | x64 (64-bit)           |
| RedHat Enterprise Linux 8       | x64 (64-bit)           |
| RedHat Enterprise Linux 7       | x64 (64-bit)           |

## System Requirements

The machine on which you install Address Verification must have a minimum of 1 GB RAM for a C++ installation. Java and Microsoft .NET installations require additional memory.

**Note:** Address Verification acquires memory from the operating system and does not use Java heap memory in engine memory management.

The main address verification functionality is encapsulated in one or more *function servers*. A function server is a running instance of the Informatica data verification engine. Each function server runs as a separate process, isolating the main process from any issues related to address processing. If a function server fails, the event is reported back to the caller as an error, the function server process is terminated, and a new function server is started.

Each function server, including standby function servers require 400 MB of RAM. If you enable hot swapping, the total memory usage doubles for Address Verification. For each job, you need a variable amount of memory based on your configuration settings.

The `IDVE.h` header file contains the functions that you use to define and run the address jobs.

Before you finalize the memory requirements, consider the size of the reference address databases that you require. The complete set of worldwide postal reference databases, including supplementary databases for address enrichments, consumes approximately 55 GB of storage space.

Preloading databases into memory significantly improves the performance of Address Verification. The machine on which you install Address Verification must have sufficient RAM to preload the databases that you require.

As the total size of the worldwide databases is approximately 55 GB, the RAM required to preload all databases and perform address processing is approximately 60 GB. For smaller file sizes, less RAM is required. For example, the worldwide batch and interactive databases consume approximately 8 GB of space, and the geocoding databases consume an additional 4.5 GB. To preload and run the worldwide batch, interactive, and geocoding databases, the maximum amount of RAM that you require is 12 GB. The enrichment databases require 2.3 GB RAM and the quick capture databases require 33.1 GB RAM.

**Tip:** If fully preloading databases is not an option, use solid-state drives to store the reference address databases. Solid-state drives are faster than hard-disk drives and can significantly improve performance, especially when multithreading is used.

You set the database preloading method in the `IDVEConfig.json` file. For more information on database preload settings, see the *Address Verification (On-Premises) Installation and Getting Started Guide*.

# Install Package

You download the installation package from Informatica and extract the package contents to the machine on which you'll run Informatica Address Verification.

## Package Contents

When you extract the Informatica Address Verification software package, you create the following directories:

- `bin`. Contains the executable file for function servers and the sample application binaries.

The folder contains the following sample applications:

- `IDVEConsoleSample`. The native C/C++ sample application.
- `IDVEConsoleSampleJava`. The Java sample application.
- `IDVEConsoleSampleNET`. The .NET sample application.

**Note:** All packages contain the C/C++ and Java sample applications. The Microsoft Windows package includes the `IDVEConsoleSampleNET` sample application.

- `etc`. Contains the JSON configuration file.
- `include`. Contains the C/C++ library header file.
- `lib`. Contains C/C++, Java, and .NET library files.
- `src`. Contains the sample application source code.
- `doc`. Contains the JSON schema documents.

**Note:** Sample implementations are available for all APIs from Informatica.

## Installing and Configuring Address Verification

For instructions on installing and configuring Address Verification, see the *Address Verification 6.1.0 (On-Premises) Installation and Getting Started Guide*.

# JSON Files

You receive the following JSON files as part of the Informatica Address Verification package:

- `IDVEConfig.json`
- `AVJob.schema.json`

## The Configuration File

The `IDVEConfig.json` file provides the configuration information for Address Verification engine operations. You verify or configure the general Address Verification settings, such as the maximum number of jobs for Address Verification, memory settings, and filesets and license file settings, in the `IDVEConfig.json` file.

Previously, the `SetConfig.xml` file covered the functionality.

## The Job Schema File

The `AVJob.schema.json` file contains the available schema objects that you can use to verify the addresses. The JSON Schema describes the expected structure of the input and output addresses for the jobs that you define.

Consider the following rules and guidelines when you use the `AVJob.schema.json` file:

- You can use the `AVJob.schema.json` file to verify the parameters of the process, result, output detail, and standardization elements.

Previously, the `Parameters.dtd` file covered the functionality.

- The `AVJob.schema.json` file contains the input properties and the address element fields that you can submit to Address Verification when you verify an address.

Previously, the `InputData.dtd` file covered the functionality.

- The `AVJob.schema.json` file contains the output properties and the address element fields that Address Verification returns for an input address. The result you receive depends on the properties and elements that you use.

Previously, the `Result.dtd` file covered the functionality.

## Reference Databases and Packages

Informatica reference databases are proprietary-format database files that contain reference data for the countries and territories that Informatica supports. The reference databases are read-only and platform-independent.

### Types of Reference Data

Address Verification provides the following types of reference database file:

- Address Code Lookup data
- Batch and Interactive data
- Certified data
- QuickCapture data
- GeocodeToAddress data
- Geocoding data
- CAMEO data
- Supplementary data

### Reading the Database File Names

The database file names have the following format:

`<ISO3>_<Function>_<ProcessType>_<ProcessSubType>_<DataSetID>_<MinVersion>.MD6`

For example, `DEU_ADV_VRF_BIA_001_6_1_0.MD6`

Interpret the file names in the following way:

- ISO3 denotes the three-character ISO country code. For example, DEU for Germany.
- Function denotes the name of the product. In the file name, ADV represents Address Verification.
- ProcessType denotes the type of process. In the file name, VRF represents Verification, ENR represents Enrichment, and RLK represents Reverse Lookup.
- ProcessSubType denotes the process mode for which the file is intended or the type of data that the file contains.



The following list identifies the subtypes:

- ACL for AddressCodeLookup mode
  - BIA for Batch and Interactive modes
  - Cxx for Certified mode
  - QCP for QuickCapture mode
  - GTA for GeocodeToAddress mode
  - GAP for arrival point geocoding
  - GRT for rooftop geocoding
  - GST for street center, locality center, and postal code center geocoding
  - CAM for CAMEO data
  - EN1 for supplementary data
- DataSetID denotes the data set ID value. The supported range is 000 through 999.
  - MinVersion denotes the oldest version of Address Verification with which you can use the file.
- MinVersion has the following format:

`<MajorVersion>_<MinorVersion>_<ReleaseVersion>`

## Installing the Reference Database Files

The *FileSetsDirectoryPath* option in the *IDVEConfig.json* file identifies the root directory for the reference database files and the *FileSetsInfo.json* file. Download the reference database files to a directory named *FileSetA* or *FileSetB* under the root directory. Install the database files directly to *FileSetA* or *FileSetB*. Do not add the files to a subdirectory in either directory.

## Reference Data Selection

When you run an address job, Informatica Address Verification uses the country information in the input addresses and the process mode that the job specifies to select the appropriate reference data file for each address. Address Verification selects the file from the files that are loaded into memory for the current job.

You can additionally use the *DataSet* array on the *DataSetDetermination* parameter to specify the reference data files that Address Verification can use for an address job.

Use a *DataSet* array when more than one reference data file meets the country and process mode requirements that the input address defines. If you configure a *DataSet* array, Address Verification uses only the files that the array identifies in the current address job.

## CHAPTER 3

# Changes in Nomenclature

This chapter includes the following topics:

- [Changes in Nomenclature Overview, 18](#)
- [Address Elements, 18](#)
- [CASS Attributes, 19](#)
- [Data Structure, 19](#)
- [Enrichments, 20](#)
- [Geocoding Status Values, 20](#)
- [Parameters, 21](#)
- [Processing Modes, 21](#)
- [Status Codes, 22](#)

## Changes in Nomenclature Overview

Review the changes in terminology for parameters, status codes, address elements, processing modes, and other elements in Informatica Address Verification 6.x.

## Address Elements

The following table lists the address element name changes between versions 5.x and 6.0:

| Informatica Address Verification 5.x | Informatica Address Verification 6.0 |
|--------------------------------------|--------------------------------------|
| Province                             | AdministrativeDivision               |
| AddressLines                         | PreformattedData                     |
| ReceipientLine                       | PostalReceipientLine                 |
| DeliveryAddressLine                  | PostalDeliveryAddressLine            |
| FormattedAddressLine                 | PostalFormattedAddressLine           |

The following table lists the address element name changes between versions 6.0 and 6.1:

| Informatica Address Verification 6.0 | Informatica Address Verification 6.1 |
|--------------------------------------|--------------------------------------|
| SurName                              | Surname                              |
| TrashNecessary                       | Necessary                            |
| TrashSuperfluous                     | Superfluous                          |
| TrashUnrecognized                    | Unrecognized                         |

## CASS Attributes

The following table lists the name changes for Coding Accuracy Support System (CASS) attributes:

| Informatica Address Verification 5.x | Informatica Address Verification 6.0.0 | Informatica Address Verification 6.1.0 |
|--------------------------------------|----------------------------------------|----------------------------------------|
| DSF2_NOSTATS_INDICATOR               | DSF2NoStatsIndicator                   | DPVNoStatIndicator                     |
| DSF2_NOSTATS_REASON                  | DSF2NoStatsReason                      | DPVNoStatReason                        |
| DSF2_VACANT_INDICATOR                | DSF2VacantIndicator                    | DPVVacantIndicator                     |
| LACS                                 | Lacs                                   | LACSIndicator                          |
| LACSLINK_INDICATOR                   | LacsLinkIndicator                      | LACSLinkIndicator                      |
| LACSLINK_RETURNCODE                  | LacsLinkReturncode                     | LACSLinkReturnCode                     |
| NDD                                  | NDD                                    | DPVNDD                                 |
| SUITELINK_RETURNCODE                 | SuiteLinkReturncode                    | SuiteLinkReturnCode                    |

**Note:** In Informatica Address Verification 5.x, the `Result.dtd` file lists the CASS attributes in uppercase, and words within the attributes are often separated by underscore characters. In version 6.x, the `AVJob.schema.json` file lists the CASS properties in mixed case and omits underscore characters.

The table lists name changes that may cause a user to overlook a CASS property while reviewing the schema file in version 6.1.0.

## Data Structure

In version 5.x, an `AddressObject` is a data structure that stores an input address, validation parameters, and the result addresses, along with enrichment and status code values. In version 6.x, this data structure is called a `Job`.

# Enrichments

The following table lists the enrichment name changes:

| Informatica Address Verification 6.0 | Informatica Address Verification 6.1 |
|--------------------------------------|--------------------------------------|
| ChomeiAzaCode                        | ChoumeiAzaCode                       |
| NewChomeiAzaCode                     | NewChoumeiAzaCode                    |
| CurrentChomeAzaCode                  | CurrentChoumeiAzaCode                |

In version 5.x, Address Verification categorized time zone code and time zone name information as enrichments for United States addresses. In version 5.x, Address Verification presents the time zone enrichment values as global properties in the `AVJob.schema.json` file.

The following table describes the name changes:

| Informatica Address Verification 5.x | Informatica Address Verification 6.x |
|--------------------------------------|--------------------------------------|
| TIME_ZONE_CODE                       | TimeZoneCode                         |
| TIME_ZONE_NAME                       | TimeZoneName                         |

**Note:** TimeZoneCode and TimeZoneName enrichments are available exclusively for United States addresses in version 6.1.

# Geocoding Status Values

The following table lists the geocoding status value name changes:

| Informatica Address Verification 5.x | Informatica Address Verification 6.x |
|--------------------------------------|--------------------------------------|
| EGCN                                 | ER_DATA_NOT_AVAILABLE                |
| EGCU                                 | ER_NOT_UNLOCKED                      |
| EGCC                                 | ER_DATA_CORRUPT                      |
| EGC0                                 | NOTHING_FOUND                        |
| EGC4                                 | POCO_BASE_CENTER                     |
| EGC5                                 | POCO_CENTER                          |
| EGC6                                 | LOCALITY_CENTER                      |
| EGC7                                 | STREET_CENTER                        |

| Informatica Address Verification 5.x | Informatica Address Verification 6.x |
|--------------------------------------|--------------------------------------|
| EGC8                                 | INTERPOLATED                         |
| EGC9                                 | POINT_ARRIVAL_POINT                  |
| EGCA                                 | Deprecated                           |
| EGCB                                 | POINT_ROOFTOP                        |

## Parameters

The following table lists the parameter name changes:

| Informatica Address Verification 5.x | Informatica Address Verification 6.x |
|--------------------------------------|--------------------------------------|
| GlobalPreferredDescriptor            | DescriptorLength                     |
| MatchingScope                        | VerificationLevel                    |
| MaxLength                            | MaxItemLength                        |
| FlexibleRangeExpansion               | RangeExpansionType                   |

**Note:** In version 5.x, you used the CountryType parameter to specify the language and the character encoding for country information in an output address. Informatica Address Verification split the CountryType parameter into two new parameters in 6.0 and changed one of the parameter names in version 6.1.

The following table lists the parameter name changes from version 5.x to version 6.1:

| Informatica Address Verification 5.x | Informatica Address Verification 6.0.0 | Informatica Address Verification 6.1.0 |
|--------------------------------------|----------------------------------------|----------------------------------------|
| CountryType                          | CountryNameType<br>CountryCodeType     | CountryNameLanguage<br>CountryCodeType |

## Processing Modes

The following table lists the name changes for address processing modes:

| Informatica Address Verification 5.x | Informatica Address Verification 6.x   |
|--------------------------------------|----------------------------------------|
| BATCH                                | Batch                                  |
| COUNTRY RECOGNITION                  | Country Recognition mode is deprecated |

| <b>Informatica Address Verification 5.x</b> | <b>Informatica Address Verification 6.x</b>                 |
|---------------------------------------------|-------------------------------------------------------------|
| GEOCODING                                   | Geocoding                                                   |
| FAST COMPLETION                             | Fast completion is deprecated in favor of QuickCapture mode |
| PARSE-ONLY                                  | Parse-only mode is deprecated                               |
| INTERACTIVE                                 | Interactive                                                 |

## Status Codes

The following table lists the status code name changes:

| <b>Informatica Address Verification 5.x</b> | <b>Informatica Address Verification 6.0.0</b> | <b>Informatica Address Verification 6.1.0</b> |
|---------------------------------------------|-----------------------------------------------|-----------------------------------------------|
| ElementInputStatus                          | MatchStatus                                   | MatchStatus                                   |
| ElementResultStatus                         | ResultStatus                                  | ResultStatus                                  |
| ExtElementStatus                            | ExtendedResultStatus                          | ExtendedResultStatus                          |
| MailabilityScore                            | ResultQuality                                 | ResultQuality                                 |
| ResultPercentage                            | ResultPercentage                              | MatchPercentage                               |

## CHAPTER 4

# JSON Schema Highlights

This chapter includes the following topics:

- [Address Verification Job Schema Overview, 23](#)
- [Arrays, 23](#)
- [Status Values, 26](#)

## Address Verification Job Schema Overview

The structure of the address data that the address job returns depends on the parameters and the fields that you configure in the job. Configure the parameters and the fields according to the contents of the `AVJob.schema.json` file.

Note that you can submit or receive an array of values for some of the elements or properties that you configure.

Additionally, note that the status codes that Informatica Address Verification returns for an address job are streamlined in version 6.x.

## Arrays

Many elements in the `AVJob.schema.json` file have properties that can accept and return an array of values. To find the properties, search the file for `"type": "array"`.

For example the following properties in the `AVJob.schema.json` file can accept an array of values:

### **PreferredLanguage**

You can submit an array of language identifiers. Address Verification returns the address in the first available language based on the order that you specify in the array.

### **CountryDetermination**

You can submit an array of country identifiers that Address Verification can apply to input addresses that do not contain country information. You can also submit an array of input fields that you want Address Verification to consider when it searches for country information. The order in which you specify the values in the property determines the order in which Address Verification searches for the country information.

### **DataSet**

The **DataSet** property under the **DataSetDetermination** element accepts an array of file names that Address Verification can use in an address verification job.

### **CountrySets**

You can submit one or more instances of the **CountrySets** element as an array to set different parameters for addresses in different countries.

### **Countries**

You can add a list of one or more countries as an array to the **Countries** element.

### **Geocoding**

You can use the **Preferred** property to list multiple types of geocoding precision in an array. Address Verification returns the geocoordinates based on the order that you specify in the array.

## Preferred Script Properties

Informatica Address Verification can return addresses from many countries in more than one script. When necessary, Address Verification can transliterate the address into the Latin script that you specify. Use the **PreferredScript** property to specify a preferred script. Find the property under **Parameters > CountrySets > Standardizations** in the `AVJob.schema.json` file.

The **PreferredScript** property supports the following options:

- **Script**. Specifies the policy that Address Verification uses to select the script for an output address.
- **TransliterationType**. Specifies the transliteration standard that Address Verification uses to transliterate the results into Latin when multiple Latin standards are available.
- **LimitLatinCharacters**. Specifies the depth of the Latin character set that Address Verification can use in an output address.

By default, Address Verification returns the addresses in the primary native script that the country uses for the address.

### PreferredScript Process Flow

The process includes the following steps:

1. In the **Script** option, review or update the script policy for the output data.
2. If you select a Latin script, specify a transliteration type to apply to the output.
3. To further define the output, for example to omit special characters that a character set does not support, set the **LimitLatinCharacters** option.

### Script

You can set one of the following values:

#### **NativePrimary**

Returns the address in the primary native script that the country uses for the address. **NativePrimary** is the default value.

#### **NativeAlternative1**

Returns the address in the alternative script that the country uses.

#### **Latin**

Transliterates the address into a Latin script.



**Note:** You can use the options `TransliterationType` and `LimitLatinCharacters` only when you request an output in a Latin script.

**Preserve**

Returns the address in the same script as the input address.

**TransliterationType**

You can set one of the following values:

**Default**

Transliterates the address into a default transliteration type. Default is the default value.

**Alternative1**

Transliterates the address into an alternative transliteration type if the alternative is available for the address.

**LimitLatinCharacters**

You can set one of the following values:

**NoLimit**

Returns the transliterated output in full. `NoLimit` is the default value.

**Latin1**

Limits the Latin characters to the Latin-1 (ISO-8859-1) character set. For example, replaces the following characters with Latin-1 characters:

| Input | Output (Latin-1) |
|-------|------------------|
| ä     | ä                |
| á     | á                |
| à     | à                |
| å     | a                |
| ã     | a                |

**ASCIISimplified**

Changes non-ASCII Latin characters to characters in the ASCII character set. Uses basic conversion, for example ö to o.

**ASCIIExpanding**

Changes non-ASCII Latin characters to characters in the ASCII character set. Uses extended conversion, for example ö to oe.

**Note:** Verify that your database supports the characters that the transliteration can return. For example, a database that supports ASCII characters only will not store all Latin characters.

## Preferred Language Properties

Informatica Address Verification can return address data in more than one language in multiple countries. You can specify the preferred output language for the addresses that you submit. Use the `PreferredLanguage`

property to specify a preferred language. Find the property under **Parameters > CountrySets > Standardizations** in the `AVJob.schema.json` file.

Consider the following rules and guidelines when you configure the PreferredLanguage property:

- You can specify one or more languages on the PreferredLanguage property. Use the three-letter ISO 639 code associated with each language that you specify. Use a comma to separate the countries. Address Verification tries to return an address in the most-preferred language based on the order that you specify in the input list. If the first language that you specified is not present in the reference data, Address Verification tries to return the address in the next language in the list.

For example, when you submit NLD, FRA, and DEU as the list of preferred languages, Address Verification returns the address in Flemish. If the reference data does not contain the address in Flemish, Address Verification returns the address in French if available.

- By default, the input list of preferred languages is empty. When the input list is empty, Address Verification returns the default language.
- You can configure the property to return an address in the language that the input address uses. The PRESERVE option is useful in cases where the input address set contains address records that are not recorded in the default language. Address Verification can preserve the input language if the reference data contains the address information in the input language. To preserve the input language, add PRESERVE as the first entry in the PreferredLanguage property. You can also add language codes after the PRESERVE option.

If Address Verification cannot return an address in the input language, Address Verification tries to return the address in the next language that you specified in the list of preferred languages. If the input list is empty, Address Verification returns the address in the default language.

- You can receive an address in more than one language in a single verification job. To return more than one language at once, add an instance of the Standardizations object for each language and set the PreferredLanguage property for each instance.

For example, to receive Flemish and French languages in the output, add two instances of the Standardizations object and specify NLD and FRA as the preferred language in each instance.

## Status Values

Informatica Address Verification 6.x updates the operation of the following status values:

### Process Status

The Process Status field returns a single alphabetical character in version 6.x and a two-character alphanumeric string in version 5.x.

You can use the Process Status field and the Result Quality field to create a two-character code in version 6.x in the same format as the Process Status field values in version 5.x. Bear in mind that the Process Status and the Result Quality fields in version 6.x are not designed to match or replace the Process Status field outputs in version 5.x.

### Address Element Status Codes

In 6.x, the MatchStatus, ResultStatus, PostalRelevance, and ExtendedResultStatus fields return a discrete status code for every address data element.

In 5.x, the ElementInputStatus, ElementResultStatus, ElementRelevance, and ExtElementStatus fields return a 20-character code in which each character refers to a different address data element.