Informatica

Informatica® Address Verification (On-Premises)

6.2.0

Installation and Getting Started Guide

Informatica Address Verification (On-Premises) Installation and Getting Started Guide
6.2.0
January 2022

© Copyright Informatica LLC 1993, 2022

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>. 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.

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/hsqlLicense.html, http://httpunit.sourceforge.net/doc/ license.html, http://jung.sourceforge.net/license.txt , 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://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://srp.stanford.edu/license.txt; http://www.schneier.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.

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: 2022-01-27

Table of Contents

Preface 5

Informatica Resources. 5

Informatica Network. 5

Informatica Knowledge Base. 5

Informatica Documentation. 5

Informatica Product Availability Matrices. 6

Informatica Velocity. 6

Informatica Marketplace. 6

Informatica Global Customer Support. 6

Chapter 1: Installation and Configuration. 7

Supported Platforms and System Requirements. 7

Installing Address Verification. 8

Installing the Informatica Address Verification C/C++-based Package. 9

Installing the Informatica Address Verification Java-based Package. 9

Installing the Informatica Address Verification Microsoft .NET-based Package. 10

Configuring the Engine. 10

Initializing the Intelligent Data Verification Engine (IDVE) framework 14

Reference Databases and Packages. 15

Installing the Reference Databases. 16

Retrieving the Software Version Number. 16

Preface

Read the Informatica Address Verification (On-Premises) Installation and Getting Started Guide to learn how to install and initialize Informatica Address Verification (On-Premises).

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.

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.

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.

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

Installation and Configuration

This chapter includes the following topics:

- Supported Platforms and System Requirements, 7
- Installing Address Verification, 8
- Configuring the Engine, 10
- Initializing the Intelligent Data Verification Engine (IDVE) framework , 14
- Reference Databases and Packages, 15
- Retrieving the Software Version Number, 16

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)

Operating System	Processor Architecture
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.

Each function server, including standby function servers require 300 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.

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 information about the database preloading, see "Configuring the Engine" on page 10.

Installing Address Verification

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.

Before You Install

Consider the following rules and guidelines before you install Address Verification:

- Do not alter the directory structure of the installation package that you extract.
- If you plan to retain an older version of Address Verification on the same machine, ensure that you extract and install the current version of Address Verification in a different location.

Installing the Informatica Address Verification C/C++-based Package

To install the C/C++-based package, complete the following steps:

1. Extract the installation directories and files from the Informatica Address Verification software package.
2. Based on the platform on which you install Address Verification, copy the `.dll` or `.so` file to a shared library path on your machine.

   To find a shared library path on a Windows machine, run the `echo %path%` command from the command prompt.

   To find a shared library path on a UNIX machine, run the `echo $LD_LIBRARY_PATH` command from the command prompt.

Installing the Informatica Address Verification Java-based Package

To use the Java-based version of Informatica Address Verification, you must install a Java Development Kit on the machine that hosts the Address Verification engine. If you want to develop your own applications, you must install a Java platform on the machine.

Windows Installation

To install the Java-based package on a Windows machine, complete the following steps:

1. Extract the installation directories and files from the Informatica Address Verification software package.
2. Copy `IDVE.dll` and `IDVE.jar` to the JRE class path.

   Typically, `C:\Program Files\Java\jre\lib\ext` is saved to the system-wide class path.

   You can explicitly set application-specific class paths with the `-cp` switch.

UNIX Installation

To install the Java-based package on a UNIX machine, complete the following steps:

1. Extract the installation directories and files from the Informatica Address Verification software package.

2. Copy `IDVE.jar` and `IDVE.so` to the JRE class path.

   Typically, `/usr/j2se/jre/lib/ext` is saved to the system-wide class path.

   You can explicitly set application-specific class paths with the `-cp` switch.

Installing the Informatica Address Verification Microsoft .NET-based Package

The Windows package for .NET contains `lib/IDVE_NET.dll`, which is the .NET binding for IDVE, and `doc/IDVE_NET.xml`, which is the API documentation for the .NET DLL.

To install the .NET-based-package on a Windows machine, complete the following tasks:

1. Extract the installation directories and files from the Informatica Address Verification software package.
2. Add a reference to the `IDVE_NET.dll` assembly to your .NET project

   Note: The `IDVE_NET.dll` file requires that the native `IDVE.dll` file is present in the regular .NET DllImport search paths.

Configuring the Engine

Informatica Address Verification uses the Intelligent Data Verification Engine (IDVE) to process the verification jobs that you define.

The `IDVEConfig.json` file provides configuration information for Address Verification engine operations. The file includes default values for several elements.

Before initializing the engine, verify or configure the elements below in the `IDVEConfig.json` file.

Find the following elements under `Core/System`:

NumFunctionServers

Specifies the number of function servers available for processing. Also, determines the number of inputs that the engine can process in parallel.

NumHotStandbyFunctionServers

Specifies the number of standby function servers.

FileSetsHotSwappingEnabled

Enables or disables dynamic switches between two filesets. Supported values are true and false. Default is false.

Note: When enabled, hot swapping increases the maximum memory requirements.

MaxMemoryMegabytes

Specifies the maximum amount of memory available for all processes, including file preloading. The default value is 1024 MB.

Note: 1024 MB is a minimum value. Depending on your workload, you might need to set a higher value.

FunctionServerInitTimeoutSeconds

Specifies the timeout period to successful initialize all function servers. Default is 600 seconds.

FunctionServerCallTimeoutSeconds

Specifies the timeout period to successfully submit a job call to a function server. Default is 60 seconds.

LicenseFilesDirectoryPath

Specifies the directory path to the license files.

Informatica sends you the license key files that you need when you buy a data subscription. Add the license key files at the location that you set in the LicenseFilesDirectoryPath element.

Note: Do not rename any license key file.

FileSetsDirectoryPath

Specifies the directory path to the reference data filesets and to the `FileSetsInfo.json` file.

The `FileSetsInfo.json` file describes the current status of the filesets. Address Verification creates the `FileSetsInfo.json` file in the directory.

Tip: Use absolute paths for all directories in the configuration file.

ExecutablesDirectoryPath

Specifies the directory path to the FunctionServer executable application.

Do not include the function server executable file name in the path.

Find the following element under `ErrorLog`:

FilePath

Specifies the path to the error log target file and the log file name.

ClearAtInit

Clears the log file during initialization. If the log file already exists and includes the content, the element clears the log file. Supported values are true and false. Default is true.

LogInput

Identifies the input issues that a function server instance encounters when processing an address, which prevents the function server from correctly processing the input and returning a result. Default is Disabled.

Includes the following values:

- CompleteInput. The error log contains the full input data that caused the issue.
- OnlySourceID . The error log contains only the "SourceID" property of the input that caused the issue. No potentially sensitive customer data written into the log.
- Disabled. The error log contains only a note that the function server encountered an issue. No further input-specific data attached.

Find the following element under `InitializationLog`:

FilePath

Specifies the path to the initialization log file and the log file name.

LogLevel

Specifies the minimum severity that will be logged to the file. Any events with lower severity does not appear in the log file. Default is Warning.

You can configure one of the following values:

- VeryCriticalError

- CriticalError
- Error
- Warning
- Status
- Info

ClearAtInit

Clears the log file during initialization. If the log file already exists and includes the content, the element clears the log file. Supported values are true and false. Default is true.

Find the following elements under `Functions/AddressVerification`:

MaxNumJobs

Specifies the maximum number of jobs that can exist concurrently.

MaxNumInputs

Specifies the maximum number of addresses that you can submit for verification. The range is from 1 through 1000. Default is 100.

MaxNumResults

Specifies the maximum number of address results that the function can return after execution. The range is from 1 through 100. Default is 20.

MaxNumVariants

Specifies the maximum number of output variants that the function can return after execution. The range is from 1 through 10. Default is 1.

Find the following elements under `FunctionServers/FileSets`:

PreloadingMethod

Specifies how Address Verification preloads the reference address databases to memory. To optimize performance, you can preload reference address databases.

Set one of the following values:

- Map. Address Verification uses the file mapping mechanism of the operating system.

  Note: Map is the default and recommended option.

- Load. Address Verification uses a memory allocation call and loads the data to the allotted memory block.

PreloadingPriorityList

Specifies the order in which Address Verification loads data files into memory. Address Verification loads the database files in the order in which they appear in the list. Each preloaded file consumes a quantity of the available memory. If the maximum memory threshold is reached, further file data will not be preloaded to memory but will remain available for processing.

Includes the following elements:

- Selector. Identifies the types of file data to prioritize during the preload operation.

  The Selector is a string that specifies a Function, Country, Process Type, Process Sub Type, and DataSet ID. For example, the following values for Selector instruct Address Verification to load all data files for batch and interactive verification for Germany in the 000 data set:

  ```
  AddressVerification/DEU/Verify/BatchInteractive/000
  ```

Note: Replace any value between a [/] separator with an asterisk to select all options at that position. For example, the following parameters specify batch and interactive data files for all countries in the 000 data set:

```
AddressVerification/*/Verify/BatchInteractive/000
```

If you use multiple trailing asterisks in the Selector parameter, you can replace them with a single asterisk. For example, `AddressVerification/*/*/*/*` and `AddressVerification/*` have the same meaning.

- PreloadingExtent. Specifies the preloading type for each of the reference address databases.

  Set one of the following values:

  - Full. Address Verification copies the entire reference address database to memory.
  - Partial. Address Verification loads fragments of each data file, starting with the most important data.
  - None. Address Verification does not preload any data.

Address Verification can stop preloading data at any time if the maximum memory threshold is reached. The unloaded reference data remain in the system storage.

IDVEConfig.json Configuration File

The following image shows an example of the `IDVEConfig.json` file in a JSON editor:

```
1  {
2     "Core": {
3        "System": {
4           "NumFunctionServers": 1,
5           "NumHotStandbyFunctionServers": 0,
6           "FileSetsHotSwappingEnabled": false,
7           "MaxMemoryMegabytes": 8000,
8           "FunctionServerInitTimeoutSeconds": 600,
9           "FunctionServerCallTimeoutSeconds": 60,
10          "LicenseFilesDirectoryPath": "C:\\AV6\\Licenses",
11          "FileSetsDirectoryPath": "C:\\AV6\\FileSets",
12          "ExecutablesDirectoryPath": "C:\\AV6"
13       },
14       "ErrorLog": {
15          "FilePath": "C:\\AV6\\ErrorLog.txt",
16          "ClearAtInit": true,
17          "LogInput": "Disabled"
18       },
19       "InitializationLog": {
20          "FilePath": "C:\\AV6\\InitLog.txt",
21          "LogLevel": "Info",
22          "ClearAtInit": true
23       }
24    },
25    "Functions": {
26       "AddressVerification": {
27          "MaxNumJobs": 10,
28          "MaxNumInputs": 20,
29          "MaxNumResults": 20,
30          "MaxNumVariants": 5
31       }
32    },
33    "FunctionServers": {
34       "FileSets": {
35          "PreloadingMethod": "Map",
36          "PreloadingPriorityList": [
37             {
38                "Selector": "AddressVerification/*/*",
39                "PreloadingExtent": "None"
40             }
41          ]
42       }
43    }
44 }
```

Initializing the Intelligent Data Verification Engine (IDVE) framework

Before you can start processing data, you must initialize IDVE. After you process the data, you can call IDVE_Deinitialize() to deinitialize IDVE. You can use the Address Verification function of the IDVE framework to verify the addresses.

Calling Initialization and Deinitialization Functions

Call the IDVE initialization and deinitialization functions in the following sequence:

1. To initialize IDVE, call IDVE_Initialize().

   Pass the configuration information as a string to IDVE_Initialize(). If you use a UTF-16-encoded-string, call IDVE_InitializeW(). You can also pass the configuration information as a file to IDVE_Initialize().

   During initialization, a number of function servers start as separate processes.

2. To deinitialize IDVE, call IDVE_Deinitialize().

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 Databases

The *FileSetsDirectoryPath* option in the `IDVEConfig.json` file identifies the root directory for the reference database files. 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.

By default, Address Verification looks for the data files in the *FileSetA* directory. During hot swapping, Address Verification also uses the data files available in the *FileSetB* directory.

Retrieving the Software Version Number

To fetch the version number of the software, use the IDVE_GetVersion() or IDVE_GetVersionW() function. Use the IDVE_GetVersion() function to retrieve the version number in the UTF-8 character encoding. Use the IDVE_GetVersionW() function to retrieve the version number in the UTF-16 character encoding. The functions otherwise operate identically.

The Address Verification version number count begins with 6 and the IDVE version number count begins with 1 at launch as IDVE is a new module of the software. You can call the functions at any time.

You can retrieve the following version numbers:

- The version of the IDVE framework. When you provide an empty string URI (""), the function fetches the IDVE framework version number, for example "0.0.2.51140".
- The version of the Address Verification function. When you provide an URI to the function ("AV"), the function fetches the Address Verification version number, for example "6.0.0.51140".

Note: You can retrieve the version number without initializing IDVE.

Retrieving the Software Version Number in the UTF-8 Encoding

The following sample code shows the structure of the IDVE_GetVersion() function:

```
IDVE_EXPORTCALL1 IDVE_StatusCode IDVE_EXPORTCALL2 IDVE_GetVersion(
const char* const kpksURI,
char* const kpsValueBuffer,
const IDVE_U64 ku64ValueBufferSize,
IDVE_U64* const kpu64SizeWritten,
char* const kpsExtStatusMsg
);
```

The following table shows the function definitions of the IDVE_GetVersion() function:

Function	Operation	Comment
const char* const kpksURI	[in]	Pointer to the zero-terminated 7-bit ASCII URI with desired version, for example "" for IDVE or "AV" for AddressVerification. The value might not be NULL.
char* const kpsValueBuffer,	[out]	Pointer to the output buffer that receives the UTF-8 encoded version string and the terminating zero. The value might not be NULL.
const IDVE_U64 ku64ValueBufferSize	[in]	Size of the output buffer in code units, including the terminating zero.
IDVE_U64* const kpu64SizeWritten	[out]	Pointer to an unsigned 64-bit integer value that receives the size of the output written in code units, excluding the terminating zero. The value might be NULL.
char* const kpsExtStatusMsg	[out]	Pointer to a buffer of size IDVE_EXT_STATUS_MSG_BUFFER_SIZE for an optional extended status message. The value might be NULL.

Retrieving the Software Version Number in the UTF-16 Encoding

The following sample code shows the structure of the IDVE_GetVersionW() function:

```
IDVE_EXPORTCALL1 IDVE_StatusCode IDVE_EXPORTCALL2 IDVE_GetVersionW(
const char* const kpksURI,
IDVE_WChar* const kpsValueBuffer,
const IDVE_U64 ku64ValueBufferSize,
IDVE_U64* const kpu64SizeWritten,
char* const kpsExtStatusMsg
);
```

The following table shows the function definitions of the IDVE_GetVersionW() function:

Function	Operation	Comment
const char* const kpksURI	[in]	Pointer to the zero-terminated 7-bit ASCII URI with desired version, for example "" for IDVE or "AV" for AddressVerification. The value might not be NULL.
IDVE_WChar* const kpsValueBuffer	[out]	Pointer to the output buffer that receives the UTF-16 encoded version string and the terminating zero. The value might not be NULL.
const IDVE_U64 ku64ValueBufferSize	[in]	Size of the output buffer in code units, including the terminating zero.
IDVE_U64* const kpu64SizeWritten	[out]	Pointer to an unsigned 64-bit integer value that receives the size of the output written in code units, excluding the terminating zero. The value might be NULL.
char* const kpsExtStatusMsg	[out]	Pointer to a buffer of size IDVE_EXT_STATUS_MSG_BUFFER_SIZE for an optional extended status message. The value might be NULL.