



Production Validator

Version 2024.1
2024-05-02

Production Validator

InterSystems Version 2024.1 2024-05-02

Copyright © 2024 InterSystems Corporation

All rights reserved.

InterSystems®, HealthShare Care Community®, HealthShare Unified Care Record®, IntegratedML®, InterSystems Caché®, InterSystems Ensemble®, InterSystems HealthShare®, InterSystems IRIS®, and TrakCare are registered trademarks of InterSystems Corporation. HealthShare® CMS Solution Pack™ HealthShare® Health Connect Cloud™, InterSystems IRIS for Health™, InterSystems Supply Chain Orchestrator™, and InterSystems TotalView™ For Asset Management are trademarks of InterSystems Corporation. TrakCare is a registered trademark in Australia and the European Union.

All other brand or product names used herein are trademarks or registered trademarks of their respective companies or organizations.

This document contains trade secret and confidential information which is the property of InterSystems Corporation, One Memorial Drive, Cambridge, MA 02142, or its affiliates, and is furnished for the sole purpose of the operation and maintenance of the products of InterSystems Corporation. No part of this publication is to be used for any other purpose, and this publication is not to be reproduced, copied, disclosed, transmitted, stored in a retrieval system or translated into any human or computer language, in any form, by any means, in whole or in part, without the express prior written consent of InterSystems Corporation.

The copying, use and disposition of this document and the software programs described herein is prohibited except to the limited extent set forth in the standard software license agreement(s) of InterSystems Corporation covering such programs and related documentation. InterSystems Corporation makes no representations and warranties concerning such software programs other than those set forth in such standard software license agreement(s). In addition, the liability of InterSystems Corporation for any losses or damages relating to or arising out of the use of such software programs is limited in the manner set forth in such standard software license agreement(s).

THE FOREGOING IS A GENERAL SUMMARY OF THE RESTRICTIONS AND LIMITATIONS IMPOSED BY INTERSYSTEMS CORPORATION ON THE USE OF, AND LIABILITY ARISING FROM, ITS COMPUTER SOFTWARE. FOR COMPLETE INFORMATION REFERENCE SHOULD BE MADE TO THE STANDARD SOFTWARE LICENSE AGREEMENT(S) OF INTERSYSTEMS CORPORATION, COPIES OF WHICH WILL BE MADE AVAILABLE UPON REQUEST.

InterSystems Corporation disclaims responsibility for errors which may appear in this document, and it reserves the right, in its sole discretion and without notice, to make substitutions and modifications in the products and practices described in this document.

For Support questions about any InterSystems products, contact:

InterSystems Worldwide Response Center (WRC)

Tel: +1-617-621-0700

Tel: +44 (0) 844 854 2917

Email: support@InterSystems.com

Table of Contents

- Production Validator..... 1**
 - 1 Validate a Production 1
 - 1.1 Gather Baseline Production Data 1
 - 1.2 Re-process Extracted Production Data and Compare 2
 - 2 The Visual Comparison Page 5

Production Validator

Production Validator helps you confirm that your production interfaces remain valid when you upgrade your InterSystems IRIS for Health instance. To do so, it extracts HL7 service headers, messages, and operation messages to a temporary database, which is then copied to an upgraded InterSystems IRIS for Health instance and replayed. By comparing the original messages to the messages processed on the upgraded system, you can evaluate and address differences.

You can access the Production Validator in the terminal. The comparisons are presented in a graphical interface on the [Visual Comparison](#) management portal page.

Metrics include the following:

- Messages that were generated but should not have been.
- Messages that are missing.
- Messages that are different in content.

1 Validate a Production

The general workflow for using the Production Validator to validate your production upgrade is as follows:

- Prerequisite: Procure the Production Validator code from InterSystems support.
- On the source system, [gather baseline production data](#).
- On the upgraded system, [re-process extracted production data and compare](#).

1.1 Gather Baseline Production Data

Follow these steps to load Production Validator code and create and export a JSON object describing production data stored in the source system database.

1. Log in to the IRIS terminal associated with the source system, and set the namespace to HSLIB.

```
USER>Set $namespace="HSLIB"  
HSLIB>
```

2. Write enable the HSLIB database to prepare for loading the Production Validator code.

```
HSLIB>write ##class(%ZHSLIB.HealthShareMgr).UpdateReadWrite("HSLIB",0)
```

3. Set a string variable to identify the filename and path to the Production Validator code.

```
HSLIB>set file="<filepath>\<filename_for_production_validator_code>.xml"
```

4. Load Production Validator code in the source system.

```
HSLIB>do $system.OBJ.Load(file,"ck")
```

5. Reset read/write permissions for the HSLIB database.

```
HSLIB>write ##class(%ZHSLIB.HealthShareMgr).UpdateReadWrite("HSLIB",1)
```

- Switch to the namespace associated with the new production system. For example, if the production name is TESTING:

```
USER>Set $namespace="TESTING"
TESTING>
```

- Create a template object for production message comparison.

```
TESTING>set template=##class(EnsLib.InteropTools.HL7.Compare.ProductionExtract).GetServiceList()
```

- Confirm the template contents by writing them in JSON format

```
TESTING>zwrite template.%ToJSON()
```

The terminal should display the expected data in JSON format.

- Optional. After creating a template object for production message comparison, you can manipulate the object to configure your eventual results in the following ways, by setting element values at the terminal command line.

- By default, the generated template will have `tempDatabase="COMPARE-" <namespace>` (in our example, `COMPARE-TESTING`) and `tempDatabaseDir=$system.Util.ManagerDirectory()._tempDatabase` (in our example, `\mgr\COMPARE-TESTING`). You can put the database elsewhere, potentially with a different name, by modifying the values of the `template.tempDatabase` and `template.tempDatabaseDir` elements.
- By default, everything is exported. You can reduce output size and times by setting the `template.percentage` element to a lower value in the template. Permitted values range from 1 to 100.
- You can set the `template.from` and `template.to` elements in the template; these control the time span of data exported.
- You can skip a particular operation by setting the `exclude` element to 1 in the template. For example, `{"name": "ADT_OPERATION", "exclude": 1}`
- The `workerJobs` element can be specified in the template to control the number of background jobs used to process the data. When not specified, the system default number of worker jobs is used. For example, you can specify `"workerJobs": 0` to not use worker jobs.

- Using the template object, export source production data to the database in `tempDatabaseDir`. In our example, the database is called `COMPARE-TESTING` and it is located in the `<filepath>\mgr\COMPARE-TESTING` directory.

```
TESTING>write ##class(EnsLib.InteropTools.HL7.Compare.ProductionExtract).Run(template)
```

This should return 1, indicating successful data export.

- Export the template object to `tempDatabaseDir` as a JSON file.

```
TEMPLATE>write
template.%ToJSON(##class(%file).NormalizeFilename(template.tempDatabaseDir_"template.json"))
```

This should return 0, indicating successful export.

1.2 Re-process Extracted Production Data and Compare

Follow these steps to generate analogous messages in your upgraded system and compare them to the messages generated in your source system.

Prerequisites

- Ensure that the upgraded system is configured identically to the source system, with the same production and supporting tables already configured.

- If you are upgrading from an InterSystems Cache-based system, rename the Cache.DAT file to IRIS.DAT.
- **IMPORTANT:** Copy your IRIS.DAT and template.json to a *new* location available to the target system.

Steps

1. Log in to the IRIS terminal associated with the source system, and set the namespace to HSLIB.

```
USER>Set $namespace="HSLIB"
HSLIB>
```

2. Production Validator code comes pre-installed in versions 2024.1 and later. If your target system is older than 2024.1, install the Production Validator code on the target system, as follows:

- a. Write enable the HSLIB database to prepare for loading the Production Validator code.

```
HSLIB>write ##class(%ZHSLIB.HealthShareMgr).UpdateReadWrite("HSLIB",0)
```

- b. Set a string variable to identify the filename and path to the Production Validator code.

```
HSLIB>set file="<filepath>\<filename_for_production_validator_code>.xml"
```

- c. Load Production Validator code in the upgraded system.

```
HSLIB>do $system.OBJ.Load(file,"ck")
```

- d. Reset read/write permissions for the HSLIB database.

```
HSLIB>write ##class(%ZHSLIB.HealthShareMgr).UpdateReadWrite("HSLIB",1)
```

- e. Switch to the namespace associated with the new production system. For example, if the production name is TESTING:

```
USER>Set $namespace="TESTING"
TESTING>
```

3. Set a variable to store the path to the template.json file.

```
TESTING>set filename = "<filepath>\template.json"
```

4. Create a JSON object from the template file, using the new variable.

```
TESTING>set template={}.%FromJSONFile(filename)
```

5. Confirm that the new JSON object is a valid object.

```
TESTING>write $isObject(template)
```

This should return 1, indicating that `template` is a valid object.

6. Ensure that the template picks up the appropriate database of the upgraded system.

```
TESTING>set template.namespace = $namespace
TESTING>set template.tempDatabase = "COMPARE-"_$namespace
TESTING>set template.tempDatabaseDir = ##class(%File).GetDirectory(filename)
```

7. Use the JSON object to begin the import of messages from the source system.

```
TESTING>set sc = ##class(EnsLib.InteropTools.HL7.Compare.ProductionExtract).Import(template)
```

8. Ensure that the import succeeded by checking the value of the status code variable `sc`. The value should be 1.

```
TESTING>write sc
```

9. Execute the comparison of messages from the source system to messages from the upgraded system, and store the results in a variable.

```
TESTING>set stats=##class(EnsLib.InteropTools.HL7.Compare.Compare).Run("Source","Target")
```

10. View the comparison results.

```
TESTING>zwrite stats
```

The output includes the following data:

- `compareStartTime`—The beginning of the timespan of the comparison.
 - `count1`—The number of messages in the output from the source system.
 - `count2`—The number of messages in the output from the upgraded system.
 - `missingFiles`—The number of files present in the source system output that are absent in the upgraded system output.
 - `newFiles`—The number of files present in the upgraded system output that are absent in the source system output.
 - `differences`—The number of files that exist in the output of both systems but that have different contents.
 - `kpi`—Percentage representing how close the upgraded system's output is to the output from the source system.
 - `compareEndTime`—The end of the timespan of the comparison.
 - `output`—A link to [the Visual Comparison page](#) in the management portal, which displays a detailed listing of the differences between the two sets of messages.
11. To view the output of the comparison on [the Visual Comparison page](#) in the management portal, click the link in the `output` field. This page displays each message in tabular form with issues highlighted. There is also a link to the visual trace. You can use these tools to identify issues. Certain issues may be inevitable or uncorrectable, and the Visual Comparison page features links to filter some of these out of the output.
 12. Optional. You can insert items to be ignored into a skip list table. Items in this table will not be included in the comparison output. For example, if the only difference between a message from the source system and the corresponding message from the upgrade system is that one is from production and the other is from testing, you might reasonably wish to skip that difference.

```
TESTING>Set rs = ##class(%SQL.Statement).%ExecDirect("insert into
HS_InteropTools_HL7_Compare.Skiplist (Segment,Field) Values (?,?)","<segment_name>","<position>")
```

See `HS.InteropTools.HL7.Compare.Skiplist` in the class reference for additional position options provided by other properties of the `Skiplist` class.

13. Optional. If you added rows to the skip list, recompute and refresh the Visual Comparison page to see the new results.

```
TESTING>do ##class(HS.InteropTools.HL7.Compare.Compare).Recompute("Source","Target")
```


2 The Visual Comparison Page

The Visual Comparison page, located at [Interoperability/Visual Comparison](#) in the management portal, displays tables comparing the output from your source system to the output from your upgraded system.

VisualComparison

1/579 Issues VisualTrace 2 FilterSession FilterServiceOperation

1	MSH	^~\&		OH Epic		HW1 ^ HW1 ^ FACID		Primary	20230118080158	ADTBEDPL	ADT ^ A08 ^ ADT_A01		430804	P	2.6				
	MSH	^~\&		OH Epic2		HW1 ^ HW1 ^ FACID		Primary	20230118080158	ADTBEDPL	ADT		430804	T	2.6				
2	EVN		A08	20230118080158				ADTBEDPL ^ ADT ^ BED ^ PLANNING ^ ^ ^ ^ ^ OH ^ ^ ^ ^ ^ HWP											
	EVN		A08	20230118080158				ADTBEDPL ^ ADT ^ BED ^ PLANNING ^ ^ ^ ^ ^ OH ^ ^ ^ ^ ^ HWP											
3	PID		1		2000043733 ^ ^ ^ EPI ^ MR			^ ONE ^ ^ ^ ^ ^ U		20070118	U		^ ^ ^ ^ ^ US ^ P					100000097717	C
	PID		1		2000043733 ^ ^ ^ EPI ^ MR			^ ONE ^ ^ ^ ^ ^ U		20070118	U		^ ^ ^ ^ ^ US ^ P					100000097717	C
4	PD1					OH HORIZON WEST HOSPITAL ^ ^ HW1													
	PD1					OH HORIZON WEST HOSPITAL ^ ^ HW1													
5	ARV		1	S	OPTOUT			PCP NOTIFICATION											
	ARV		1	S	OPTOUT			PCP NOTIFICATION											
6	PV1		1	IP	HWHEME ^ HWB13 ^ B13 ^ HW1 ^ R ^ ^ ^ ^ HWH ED ^ ^ DEPID	ER				1003108325 ^ OXFORD ^ APRIL ^ L ^ ^ ^ ^ NPI ^ ^ ^ ^ NPI ~ 006184 ^ OXFO									
	PV1		1	IP	HWHEME ^ HWB13 ^ B13 ^ HW1 ^ R ^ ^ ^ ^ HWH ED ^ ^ DEPID	ER				1003108325 ^ OXFORD ^ APRIL ^ L ^ ^ ^ ^ NPI ^ ^ ^ ^ NPI ~ 006184 ^ OXFO									
7	PV2			Priv				HOSP ENC							N				Mec
	PV2			Priv				HOSP ENC							N				Mec

- Each table depicts a single message. You can navigate between messages by clicking the arrow buttons on either side of the issue number displayed above the table.
- Each row represents a single segment of the displayed message, lining up output from the source system, position-by-position directly above output from the upgraded system.
- The first cell of each row is the line number, and the second is the segment name.
- Each cell from the third column onward represents a single position in its segment. You can learn the position number for a cell by looking at that cell's tool tip.
- Differences are highlighted.

For diagnostic purposes, the **Visual Trace...** link takes you to the visual trace of messages that yielded the message currently being displayed.

Two filter links are displayed above the table:

- The **Filter Session...** link removes the currently-displayed message from the comparison output.
- The **Remove all records for <Service|Operation>** link removes all records that match the specified service/operation pairing.

